Ясюкович Э.И.

Конспект лекций
по дисциплине "Моделирование систем"

Содержание

УМК

Введение

Широкое внедрение автоматизированных систем управления (АСУ), созданных с применением экономико-математических методов и средств компьютерной техники, требует всестороннего изучения методики их анализа и синтеза.

В настоящее время трудно найти область человеческой деятельности, в которой в той или иной степени не использовались бы методы моделирования. Без использования моделирования практически невозможно провести исследование АСУ на всех этапах его разработки, начиная с обследования объекта управления и составления технического задания на проектирование и кончая внедрением системы в эксплуатацию. Именно моделирование является средством, позволяющим без капитальных затрат решить проблемы построения больших систем.

В настоящей дисциплине рассматриваются основные классы моделей и методы моделирования, применяемые при разработке АСУ, принципы построения моделей больших систем, методы формализации и алгоритмизации, а также методы практической реализации моделирующих алгоритмов и программ.

В.1.Основные определения

В основе моделирования систем используется теория моделирования.

Теория моделирования - это теория замещения одних объектов (оригиналов) другими объектами (моделями).

Модель - это образ какого-либо объекта, процесса или явления, используемый в качестве его заменителя.

Модели делятся на физические и математические.

Физическая модель - это модель, которая сохраняет физическую природу оригинала. Например, модель самолета для исследования параметров крыла в аэродинамической трубе.

Математическая модель технического объекта (системы) - это совокупность математических объектов (компоненты, параметры, переменные, матрицы, множества, функциональные зависимости, ограничения, целевые функции и т.п.) и отношений между ними, которая адекватно отображает свойства системы.

Компоненты - это такие составные части модели или системы, которые при соответствующем объединении образуют систему. Иногда компонентами системы считают ее элементы или подсистемы.

Параметрами модели являются величины, характеризующие исследуемую систему и выбираемые исследователем произвольно.

Переменные модели - это величины, которые могут принимать некоторые значения.

В математических моделях используются переменные двух видов: экзогенные и эндогенные.

Экзогенные переменные - это независимые переменные (входные воздействия, воздействия внешней среды и внутренние параметры).

Эндогенные переменные - это выходные (зависимые) переменные или характеристики системы или переменные состояния.

Если необходимо описать входы и выходы системы, то используют входные и выходные переменные.

Функциональные зависимости описывают поведение переменных и параметров в пределах компоненты или же выражают соотношения между компонентами системы. Эти соотношения по своей природе являются либо детерминированными, либо стохастическими и выражаются обычно в виде алгоритмов и устанавливают зависимость между переменными состояния и экзогенными переменными.

Детерминированные соотношения - это соотношения, которые однозначно описываются входными и внешними параметрами системы.

Стохастические соотношения учитывают влияние случайных факторов.

Ограничения представляют собой установленные пределы изменения значений переменных. Ограничения могут вводиться либо разработчиком, либо устанавливаться самой системой вследствие присущих ей свойств.

Целевая функция (функция критерия) представляет собой функцию, позволяющую оценить качество функционирования системы. Выражения целевой функции должно однозначно определять цели и задачи, с которыми должны соизмеряться принимаемые решения.

Система - это группа или совокупность компонентов, объединенных некоторой формой регулярного взаимодействия или взаимозависимости и выполняющих заданные функции.

При построении модели большую роль играют гипотезы, т.е. научные предположения, основывающиеся на небольшом количестве опытных данных или наблюдений.

В процессе формулирования и проверки правильности гипотез большое значение имеет аналогия. Аналогия - это суждение о каком-либо частном сходстве двух объектов.

Моделирование представляет собой процесс замены системы (объекта или физического процесса) моделью, имеющей такие же свойства, с целью получения информации об этой системе путем проведения экспериментов с полученной моделью.

Математическое моделирование представляет собой процесс замены системы математической моделью и реализации этой модели на компьютере на основе разработанного алгоритма.

Методы математического моделирования, используемые для исследования характеристик процесса функционирования систем, можно разделить на аналитические и имитационные.

Для аналитических моделей характерно, что процессы функционирования элементов записываются в виде некоторых функциональных соотношений или логических условий. Наиболее полное исследование можно провести, если известны явные зависимости, связывающие искомые характеристики с параметрами системы и начальными условиями. Такие зависимости удается получить только для сравнительно простых систем. При усложнении систем их исследование аналитическими методами становится затруднительным. Поэтому в данном случае, желая использовать аналитические методы, обычно идут на существенное упрощение модели. Однако исследования на упрощенных моделях позволяют получить лишь ориентировочные результаты.

Имитационная модель воспроизводит процесс функционирования системы во времени. Сущность метода имитационного моделирования состоит в реализации на ЭВМ процесса функционирования системы и получении по исходным данным сведений о ее состоянии в определенные моменты времени.

Основным преимуществом имитационных моделей по сравнению с аналитическими является возможность решения более сложных задач. Имитационные модели позволяют достаточно точно учитывать: наличие дискретных и непрерывных элементов; нелинейные характеристики элементов системы; случайные воздействия и т.д., которые при аналитическом моделировании создают непреодолимые трудности. Поэтому имитационное моделирование является наиболее эффективным, а часто и единственным практически доступным, методом исследования систем.

Кибернетическое моделирование позволяет не выявлять подобие физических процессов, происходящих в реальной системе. Система в этом случае рассматривается как черный ящик, имеющий ряд входов и выходов. Моделирование же производится на основе функции, представленной в виде некоторых операторов связи между входом и выходом.

В.2.Философские аспекты теории подобия и моделирования

Практически во всех областях человеческой деятельности в той или иной степени используются различные методы моделирования. Особенно это относится к сфере управления различными объектами и системами, где основными являются процессы принятия решений на основе получаемой информации. При этом под *объектом* понимается все то, на что направлена человеческая деятельность. Разработка методологии моделирования позволяет упорядочить информацию об объектах, которые взаимодействуют между собой и внешней средой.

Научно-техническое развитие в любой области идет по пути: наблюдение и эксперимент – теоретические исследования – организация производственных процессов.

Диалектический путь познания истины (объективной реальности): "от живого созерцания к абстрактному мышлению и от него к практике".

С точки зрения философии моделирование – эффективное средство познания природы. Процесс моделирования предполагает наличие:

- объекта исследования;
- исследователя, перед которым поставлена конкретная задача;
- модели, созданной для получения информации об объекте и необходимой для решения поставленной задачи.

При этом исследователь по отношению к модели является экспериментатором. Хотя эксперимент в данном случае проводится не с реальным объектом, а с моделью. Такой инструмент для инженера - системотехника – это инструмент непосредственного решения организационно-технических задач.

Любой эксперимент может иметь существенное значение в конкретной области науки только при специальной его обработке и обобщении. Единичный эксперимент не может быть решающим для подтверждения гипотезы, проверки теории. Борьба за правильную оценку места эксперимента в научном познании составляет часть той борьбы, которая ведется в теории познания между материализмом и идеализмом во всех его разновидностях. Поэтому инженеры (исследователи и практики) должны быть знакомы с элементами современной методологии теории познания и не должны забывать о том, что именно экспериментальное исследование, опыт, практика являются критерием истины.

В.3.Моделирование и эргономика

Эргономика представляет собой науку, занимающуюся исследованием и проектированием систем, включающих как технические звенья, так и человека. Т.е. объектом исследования в эргономике являются человеко-машинные системы.

Научно-техническая революция, основанная на использовании компьютерной техники, совершила переворот в представлениях о трудовой деятельности человека. В настоящее время никого не удивляет сообщение о том, что компьютер управляет процессами проектирования, разработки технологии, изготовлением деталей, сборки изделий и т.д.

Во всех эти случаях компьютер работает по программам, разработанным на основе соответствующих математических моделей и алгоритмов. Поэтому очевидно, что моделирование, особенно математическое, играет значительную роль в эргономике.

В.4.Значение моделирования в экономии топливно-энергетических ресурсов, улучшения экологической обстановки и снижения уровня вредного влияния машин на экологическую среду

Основное назначение моделирования состоит в предоставлении в распоряжение исследователя (конструктора) инструмента для получения информации об особенностях функционирования уже существующих объектов и систем с целью их совершенствования или проектируемых для выбора их оптимальных конструкций. Оптимальные режимы функционирования систем должны обеспечивать минимальный расход топливно-энергетических ресурсов, не ухудшать экологическую обстановку и вести к снижению уровня вредного влияния машин на экологическую среду. Неоценимую роль в проектировании таких конструкций машин и их режимов функционирования играют математические модели, которые позволяют еще на начальных стадиях проектирования в первом приближении выбрать рациональные параметры проектируемых объектов.

В.5.Основные понятия моделирования систем. Классификация методов и видов моделирования

Основное назначение математических моделей заключается в формировании структуры (синтезе) систем и в воспроизведении процессов функционирования последних для оценки их качества.

Методы моделирования не могут быть абсолютными, т.е. модель не может воспроизводить все стороны процесса функционирования изучаемых систем, т.к. абсолютное подобие - это замена системы той же системой, а мы от этого сознательно уходим.

На практике для решения задач реализации подобия применяется ***полное и локальное*** моделирование, реализуемое соответственно при полном и неполном подобии.

Полное подобие - это подобие между всеми элементами, процессами и функциями моделируемого объекта и его модели.

Неполное подобие - это подобие между частью элементов, процессов и функций объекта и его модели.

Полное и локальное моделирование представляет собой приближенные методы, т.к. некоторые факторы могут моделироваться количественно неточно или совсем не входят в модель.

По видам моделирование классифицируется следующим образом, рис 1.1.

В зависимости от характера изучаемых процессов в системе все виды моделирования могут быть разделены на:

- детерминированные и стохастические;
- статические и динамические;
- дискретные, непрерывные и дискретно-непрерывные.

Детерминированное моделирование отображает детерминированные процессы, т.е. процессы, в которых предполагается отсутствие всяких случайных воздействий.

Рис. 1.1. Классификация видов моделирования систем

Стохастическое моделирование отображает вероятностные процессы и события. В этом случае анализируется ряд реализаций случайного процесса и оцениваются средние характеристики.

Статическое моделирование используется для описания объекта в какой-либо момент времени.

Динамическое моделирование отражает поведение объекта во времени.

Дискретное моделирование служит для описания дискретных процессов.

Непрерывное моделирование позволяет отразить непрерывные процессы в системах.

Дискретно-непрерывное моделирование используется в тех случаях, когда необходимо учесть наличие в системе как дискретных, так и непрерывных процессов.

В зависимости от формы представления объекта можно выделить *мысленное и реальное* моделирование.

Мысленное моделирование используется, когда процессы функционирования системы практически не реализуемы с помощью модели в заданном интервале времени, либо существуют в условиях, невозможных для их физического создания. К мысленному моделированию относится наглядное, символьное и математическое.

При реальном моделировании используется возможность исследования различных характеристик либо на реальном объекте целиком, либо на его части.

В.6.Классификация математических моделей

Различают математические модели *структурные и функциональные.*

Структурные модели относятся к конструкторскому аспекту проектирования. Они отражают структурные свойства системы, такие как геометрическую форму, размеры, взаимное расположение ее элементов в пространстве.

Структурные модели чаще всего представляются в виде графов, матриц инциденций и смежности и т.п.

Функциональные модели связаны с функциональным аспектом проектирования систем и отражают закономерности процессов их функционирования. Эти модели представляются в виде систем уравнений, описывающих соответствующие процессы: электрические, механические, тепловые, процессы преобразования информации и т.д. Функциональные модели применяют на этапах оценки работоспособности систем, структура которых предварительно синтезируется с помощью структурных моделей.

В зависимости от сложности моделируемой системы в иерархии функциональных моделей выделяют три укрупненных уровня - *микроуровень, макроуровень, и метауровнь*.

На *микроуровне* используют математические модели, описывающие физическое состояние и процессы в сплошных средах. Для построения таких моделей применяют аппарат уравнений математической физики. Например, дифференциальные уравнения в частных производных (уравнения электродинамики, теплопроводности, упругости, газовой динамики и_т.д.). В качестве зависимых переменных на микроуровне могут использоваться фазовые переменные, такие как напряженности полей, электрические потенциалы, давления, температуры, концентрации частиц, плотности токов, механические напряжения и деформации. Независимыми переменными в этих моделях являются время и пространственные координаты.

На *макроуровне* математические модели описывают процессы в отдельных элементах: деталях, дискретных электрорадиоэлементах, участках полупроводниковых кристаллов. В качестве фазовых переменных используются электрические напряжения, токи, силы, скорости, температуры, расходы и т.п.

Функциональные модели на макроуровне представляют собой системы алгебраических или обыкновенных дифференциальных уравнений.

На *метауровне* математические модели описывают информационные процессы, протекающие в системах. Для построения математических моделей используют теорию автоматического управления, математическую логику, теорию конечных автоматов, теорию массового обслуживания.

Математические модели на метауровне представляются в виде систем обыкновенных дифференциальных уравнений, систем логических уравнений, имитационных моделей систем массового обслуживания.

В.7.Формы представления математических моделей

Для представления моделей используют следующие основные формы:

1.*Инвариантная* – запись модели с помощью традиционного математического языка без учета метода решения уравнений модели;

2.*Алгоритмическая* – запись уравнений модели и выбранного метода решения в форме алгоритма;

Среди алгоритмических моделей особое место занимают имитационные модели, предназначенные для имитации физических или информационных процессов при различных входных воздействиях. Примерами имитационных моделей являются модели электронных схем в виде систем обыкновенных дифференциальных уравнений или модели систем массового обслуживания, предназначенные для имитации процессов прохождения заявок через систему и т.п.

3.*Аналитическая* – запись модели в виде аналитического решения ее исходных уравнений. В этой форме модель представляет собой явные выражения выходных параметров как функций внутренних и внешних параметров;

4.*Схемная (графическая)* форма - представление модели на некотором графическом языке (на языке графов, эквивалентных схем, диаграмм и т.п.). Модели, представленные в этой форме, предварительно переводятся на язык инвариантных или алгоритмических форм.

В.8.Критерии оценки эффективности применения методов моделирования

При моделировании весьма актуальным является вопрос о его эффективности. Эффективность моделирования оценивается рядом следующих критериев:

1.Точностью и достоверностью результатов моделирования;

2.Временем построения и работы с моделью;

3.Затратами машинных ресурсов (времени и памяти);

4.Стоимостью разработки и эксплуатации модели.

Точность моделирования оценивается путем сравнения результатов моделирования и натурного эксперимента.

Время моделирования складывается из времени ввода и вывода данных, проведения вычислений и обращения к внешним запоминающим устройствам. Большое влияние на затраты машинного времени при проведении имитационных экспериментов оказывает рациональное планирование эксперимента.

Вообще, эффективность моделирования определяется разработкой научных основ моделирования и развитием средств вычислительной техники.

В.9.Требования к математическим моделям

Основными требованиями, предъявляемыми к математическим моделям, являются требования адекватности, универсальности и экономичности.

Адекватность. Модель считается адекватной, если она отражает свойства объекта с приемлемой точностью. Точность определяется как степень совпадения значений выходных параметров модели и объекта.

Пусть $\boldsymbol{e_j}$ -относительная погрешность модели по j-му выходному параметру:

$$e_j = \frac{y_{rj} - y_j}{y_j}, \qquad (2.1)$$

где y_{rj} - j-й выходной параметр, рассчитанный с помощью модели; y_j - тот же выходной параметр, имеющий место в моделируемом объекте.

Погрешность модели e_m по совокупности учитываемых выходных параметров оценивается одной из норм вектора $e = (e_1, e_2,...,e_n)$, например

$$e_m = max(e_j),\ j = 1,...,n. \tag{2.2}$$

$$e_m = \sqrt{\sum_{j=1,n} e_j^2} \tag{2.3}$$

Точность модели зависит от внешних факторов. Пусть $e_{пред}$ - предельная допустимая погрешность. Тогда в пространстве внешних параметров можно выделить область, в которой выполняется условие

$$e_m < e_{пред}. \tag{2.4}$$

Эту область называют областью адекватности (ОА) модели.

Если для каждого выходного параметра ввести индивидуальное значение $e_{предj}$, то можно определить ОА, в которой одновременно выполняются все n условий

$$|e_j| <= e_{предj}. \tag{2.5}$$

Пример ОА приведен на рис.2.1.

В общем случае ОА может иметь произвольную сложную и неудобную в использовании форму. Поэтому на практике вместо истинных ОА применяют те или иные их аппроксимации.

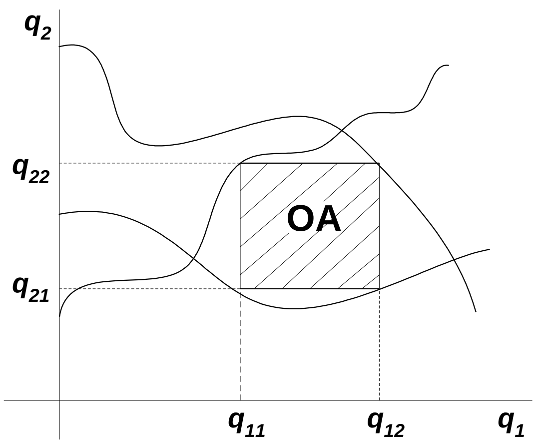

Рис. 2.1. Пример области адекватности

Наиболее просто представляются области, имеющие форму гиперпараллелепипеда, который задается неравенствами вида:

$$q_{k1} <= q_k <= q_{k2},\ k = 1,..., p, \tag{2.6}$$

где p - размерность пространства внешних параметров.

Область адекватности может быть задана также в виде участков гиперплоскостей, областей гиперсфер и т.д.

Определение ОА для конкретных моделей представляет собой сложную процедуру, требующую больших вычислительных затрат. Последние быстро растут с увеличением размерности пространства внешних параметров.

Знание ОА позволяет правильно выбрать модель из числа имеющихся и тем самым повышать достоверность результатов моделирования.

Пример. Для одной из возможных макромоделей логического элемента транзисторно-транзисторной логики (ТТЛ), реализующего функцию И-НЕ, ОА выражается следующими неравенствами:

$$4.47 \text{ В} <= E <= 5.89 \text{ В};$$

$$3.90 \text{ нс} <= t_{ф.вх.} <= 11.2 \text{ нс};$$

$$0 <= N <= 3;$$

$$3 \text{ нс} <= t_{вх} <= 6 \text{ нс},$$

где E - напряжение питания; $t_{ф.вх.}$ - длительность фронта входного сигнала; N - коэффициент нагружения; $t_{вх}$ - длительность входного сигнала.

Универсальность модели определяется числом и составом учитываемых в ней внешних факторов и выходных параметров. Увеличение числа учитываемых внешних факторов расширяет применимость модели, но существенно удорожает работу по ее отработке (определению области адекватности).

Набор выходных параметров для большинства объектов достаточно стабилен. Например, для макромоделей логических элементов БИС выходными параметрами являются уровни выходного напряжения в состояниях логического *"0"* и *"1"*, запасы помехоустойчивости, задержка распространения сигнала, рассеиваемая мощность.

Экономичность модели характеризуется затратами вычислительных ресурсов для ее реализации, т.е. затратами машинного времени и памяти, необходимых для реализации модели. Эти затраты зависят как от особенностей выбранной модели, так и от выбранного метода ее решения.

Анализ приведенных требований показывает, что требования широких областей адекватности и высокой степени универсальности, с одной стороны, и высокой экономичности, с другой, являются противоречивыми. Это обстоятельство обусловливает использование нескольких моделей для анализа систем одного и того же типа.

1. Принципы моделирования больших систем

1.1.Понятие большой системы. Подсистемы и элементы

Развитие науки и совершенствование техники ведет к созданию все более сложных технических устройств и систем, повышается быстродействие технологических агрегатов, растет уровень механизации и автоматизации управления, создаются автоматизированные системы управления (АСУ) на базе автоматики, телемеханики, электроники и вычислительной техники. Телемеханика - наука об управлении и контроле на расстоянии с передачей по каналу связи кодированных электрических и радиосигналов, несущих управляющую или контрольную информацию. Обычно используется один канал.

Объектами телемеханического управления и контроля могут служить:

- технологические процессы,
- машины,
- устройства,
- биологические системы и др.

Телемеханика, как отрасль техники, используется:

- в энергетике,
- на газо- и нефтепроводах,
- на атомных и электростанциях,
- на некоторых химических предприятиях,
- в системах сбора информации.

При проектировании таких сложных объектов (больших систем) возникают задачи, требующие исследования закономерностей их функционирования. При доводке таких систем становится практически недопустимым натурный эксперимент из-за колоссального роста затрат времени и средств. Это требует не только повышения точности инженерных расчетов, но и значительного расширения круга используемых методов, особенно математического моделирования.

Под большой системой понимается объект, содержащий большое количество взаимосвязанных и взаимодействующих подсистем и элементов и сложноорганизованное управление.

Основные особенности большой системы:

1. Наличие подсистем и большого количества элементов,
2. Сложный характер связей между отдельными элементами,
3. Сложность функций, выполняемых системой,
4. Наличие управления, как правило, сложноорганизованного,
5. Взаимодействие с окружающей средой,
6. Воздействия случайных факторов,
7. Высокая степень автоматизации в системе и применение микропроцессоров в качестве основного управляющего звена.

При исследованиях больших систем они разбиваются на ряд самостоятельных взаимосвязанных и взаимодействующих подсистем. Подсистема характеризуется частью свойств системы и рассматривается как определенный аспект описания системы.

Структурно подсистемы состоят из элементов. Под элементом понимается часть подсистемы (системы), выполняющая некоторые функции, например

элемент: триггер;
подсистема: оперативная память;
система: компьютер.

1.2.Структура, функции, параметры состояния и характеристики системы

При рассмотрении больших систем важным понятием является понятие структуры системы.

Под структурой системы понимают организацию системы из отдельных элементов с их взаимосвязями, которые определяются распределением функций, выполняемых системой, и целей, которые должны быть достигнуты в процессе ее функционирования.

Под функцией системы понимается ее свойство, приводящее к достижению цели. Изучение системы осуществляется путем изучения ее свойств и функций.

В связи с этим для изучения структуры системы и ее свойств используются два основных подхода: структурный и функциональный.

При структурном подходе изучается состав элементов системы и связи между ними.

При функциональном подходе изучаются отдельные функции, т. е. алгоритмы функционирования системы.

1.3.Свойства и состояния системы

Свойства системы оцениваются некоторыми качественными и количественными характеристиками ее элементов, подсистем либо системы в целом. Для количественной характеристики вводятся числа, выражающие отношения между ее реальным значением и эталоном. Качественные характеристики системы находятся, например, с помощью экспертных оценок.

Функционирование системы означает переход ее из одного состояния в другое, т. е. движение в пространстве состояний Z.

Состояние системы определяется ее параметрами состояния.

При описании системы используются также внутренние, внешние, входные и выходные параметры.

1.4.Принципы системного подхода в моделировании.

При синтезе и анализе больших систем получил развитие системный подход, который существенно отличается от классического (индуктивного) подхода.

Индуктивный подход предусматривает рассмотрение системы путем перехода от частного к общему и синтез (конструирование) системы путем слияния ее компонентов, разрабатываемых раздельно.

Системный подход предполагает последовательный переход от общего к частному. При системном подходе в понятие системы в качестве составляющих входят такие понятия как структура, функция, состояние, элемент, управление и другие. В связи с этим системный подход служит методом комплексного изучения сложных объектов и процессов с точки зрения устройства и взаимосвязи их подсистем, функционирования подсистем и объекта в целом, а также характера взаимодействия с окружающей внешней средой.

Таким образом, что при классическом (индуктивном) подходе движение происходит от частного к общему и создаваемая модель (система) образуется путем суммирования отдельных ее компонентов. При этом не учитывается возникновение нового системного эффекта.

1.5.Методы получения математических моделей

Методы получения моделей делятся на теоретические и экспериментальные.

Теоретические методы основаны на изучении физических закономерностей, протекающих в системе, определении соответствующего этим закономерностям математического описания, обосновании и принятии упрощающих предложений, выполнении необходимых выкладок и приведении результата к принятой форме представления модели. Теоретические методы используются в большинстве случаев для получения математических моделей на микроуровне. Эти методы базируются на основе уравнений математической физики - интегральных, интегро-дифференциальных или дифференциальных уравнений в частных производных.

Экспериментальные методы основаны на использовании внешних проявлений свойств объекта, фиксируемых во время эксплуатации однотипных объектов или при проведении целенаправленных экспериментов.

1.6.Технология моделирования больших систем

Технология моделирования больших систем от идеи конкретной задачи до получения результатов включает в себя следующие этапы:

1. Формальное описание системы.
2. Построение моделирующего алгоритма.
3. Реализация алгоритма (модели) на компьютере.
4. Планирование расчетного эксперимента.
5. Экспериментирование с моделью
6. Обработка результатов расчетного эксперимента.
7. Принятие решения.

При системном подходе большая система на основе принципа декомпозиции разбивается на подсистемы и элементы так, чтобы была возможна их формализация. Для формализации систем используется следующий математический аппарат:

- булевы схемы,
- марковские процессы,
- вероятностные автоматы,
- математический аппарат систем массового обслуживания,
- математический аппарат агрегативных систем,
- математический аппарат динамических систем и т. д.

Формальное описание системы заканчивается представлением ее в виде набора типовых блоков и построением математической модели.

1.7.Аналитические и имитационные модели

Математические модели играют большую роль в процессе проектирования и исследования больших систем. При этом используются аналитические и имитационные модели.

Исторически первым сложился аналитический подход к исследованию систем, когда ЭВМ использовалась в качестве вычислителя по аналитическим зависимостям. Моделирование больших систем с помощью аналитических моделей наталкивается на значительные затруднения, которые приводят к необходимости существенного упрощения моделей, что часто приводит к получению недостоверных результатов.

Поэтому наряду с аналитическими моделями большое внимание уделяется имитационным моделям. Перспективность имитационного моделирования как метода исследования больших систем возрастает с повышением быстродействия и объема оперативной памяти компьютеров, развитием математического обеспечения.

1.8.Основные подходы к построению математических моделей систем

Задачи формализации и алгоритмизации.

Под формализацией понимается процесс построения математической модели, описывающей функционирование системы в течение продолжительного времени.

Формализации любого реального процесса (системы) предшествует изучение структуры составляющих его элементов. В результате этого появляется так называемое <u>содержательное описание</u> процесса, которое представляет собой попытку четко изложить закономерности, характерные для исследуемого процесса, и позволяют выполнить постановку прикладной задачи.

Результаты содержательного описания являются исходным материалом для последующих этапов формализации: построения формализованной математической схемы и математической модели для него.

1.9.Понятие формализованной математической схемы

<u>Формализованная математическая схема</u> – это промежуточное звено от содержательного к формальному описанию процесса функционирования системы с учетом воздействия внешней среды. Т.е. имеет место цепочка: <описательная модель – математическая схема – математическая (аналитическая или (и) имитационная) модель>.

Формализованная математическая схема разрабатывается не всегда, а лишь тогда, когда из-за сложности исследуемого процесса или трудоемкости формализации некоторых его элементов непосредственный переход от содержательного описания к математической модели оказывается невозможным или нецелесообразным.

Представляется формализованная схема в виде укрупненных (обобщенных) математических зависимостей – уравнений в общем виде.

Исходной информацией при построении математических моделей служат данные о назначении и условиях работы исследуемой или проектируемой системы. Эта информация определяет основную цель моделирования системы и позволяет сформулировать требования к разрабатываемой модели (аналитической или имитационной).

Каждая конкретная система характеризуется набором свойств, под которыми понимаются величины, отражающие поведение моделируемого объекта (реальной системы) и учитывающие условия его функционирования во взаимодействии с внешней средой. При построении математической модели необходимо решить вопрос о полноте системы. В то же время не следует забывать и о задаче упрощения модели, которая помогает выделить основные свойства, отбросив второстепенные.

Отнесение свойств системы к основным или второстепенным существенно зависит от цели моделирования (например, анализ вероятностно - временных характеристик, синтез структуры системы и т.д.).

1.10.Величины, используемые для формализации системы

Модель объекта (системы) можно представить в виде множества величин, описывающих процесс функционирования реальной системы и образующих в общем случае следующие подмножества:

1.Совокупность входных воздействий на систему

$$x_i \in X,\ i=1,n_x \tag{2.7}$$

2.Совокупность воздействий внешней среды

$$v_l \in V,\ l=1,n_v \tag{2.8}$$

3.Совокупность внутренних (собственных) параметров системы

$$h_k \in H,\ k=1,n_h \tag{2.9}$$

4.Совокупность выходных характеристик системы

$$y_j \in Y,\ j=1,n_y \tag{2.10}$$

В перечисленных подмножествах можно выделить управляемые и неуправляемые переменные.

В общем случае параметры $\boldsymbol{x_i}$, $\boldsymbol{v_l}$, $\boldsymbol{h_k}$, $\boldsymbol{y_j}$ являются элементами непересекающихся подмножеств и содержат как детерминированные, так и стохастические составляющие.

При моделировании системы входные воздействия, воздействия внешней среды и внутренние параметры системы являются независимыми (экзогенными) переменными, а ее выходные параметры - зависимыми (эндогенными) переменными.

Процесс функционирования системы описывается во времени оператором F_s, который в общем случае преобразует экзогенные переменные в эндогенные в соответствии с соотношением вида:

$$\vec{y}(t) = F_s(\vec{x}, \vec{v}, \vec{h}, t). \tag{2.11}$$

Совокупность зависимых выходных характеристик системы от времени $y_j(t)$ для всех видов $j = 1, n_x$ называется выходной траекторией $y(t)$. Зависимость (2.11) называется законом функционирования системы и обозначается F_s.

Закон функционирования F_s в общем случае может быть задан в виде функции, функционала, логических условий, в алгоритмической и табличной формах или в виде словесного правила соответствия.

Для описания и исследования систем важным понятием является алгоритм функционирования $\boldsymbol{A_s}$. Под $\boldsymbol{A_s}$ понимается метод получения выходных характеристик с учетом входных воздействий, воздействий внешней среды и соответствующих параметров системы.

Один и тот же закон функционирования $\boldsymbol{F_s}$ системы может быть реализован различными способами, т.е. с помощью множества различных алгоритмов функционирования.

Соотношения (2.11) являются математическим описанием поведения объекта или системы моделирования во времени, т.е. отражает его динамические свойства. Поэтому математические модели такого типа называют динамическими моделями.

Другой тип моделей, называемых статическими моделями, представляет собой соотношения между двумя подмножествами свойств моделируемого объекта *Y и {X, V, H}*, что в векторной форме может быть записано в виде:

$$\vec{y} = F_s(\vec{x},\vec{v},\vec{h}). \tag{2.12}$$

Соотношения (1) и (2) могут быть заданы различными способами: аналитически(с помощью формул), графически, таблично и т.д. Такие соотношения в ряде случаев могут быть получены через свойства системы в конкретные моменты времени, называемые состояниями. Состояния системы характеризуются векторами

$$z' = (z_1', z_2', \ldots, z_k'); \tag{2.13}$$

$$z'' = (z_1'', z_2'', \ldots, z_k''), \tag{2.14}$$

где $z_1' = z_1(t')$, $z_2' = z_2(t')$, $, z_k' = z_k(t')$ в момент $t' \in (t_o,T)$;
$z_1'' = z_1(t'')$, $z_2'' = z_2(t'')$, $, z_k'' = z_k(t'')$ в момент $t'' \in (t_o, T), k = 1,\ldots,n_z$.

Если рассматривать процесс функционирования системы как последовательную смену состояний $z_1(t')$, $z_2(t)$, …, $z_k(t)$, то они могут быть интерпретированы как координаты точек в *k*-мерном фазовом пространстве. Причем каждой реализации процесса будет соответствовать некоторая фазовая траектория.

Совокупность всех возможных значений состояний *{z}* называется пространством состояний *Z* объекта моделирования, причем $z_k \in Z$.

Состояния системы в момент времени $t_o < t^* \le T$ полностью определяются:

- начальными условиями $z^O = (z_1^O, z_2^O, \ldots, z_k^O)$,

где $z_1^O = z_1(t_O)$, $z_2^O = z_2(t_O)$, …, $z_k^O = z_k(t_O)$;

- входными воздействиями *x(t)*;
- внутренними параметрами *h(t)*;
- воздействиями внешней среды *v(t)*,

которые имели место за промежуток времени $(t^* - t^O)$, с помощью двух векторных уравнений

$$\vec{z}(t) = \phi(\vec{z}^o, \vec{x}, \vec{v}, \vec{h}, t); \tag{2.15}$$

$$\vec{y}(t) = F(\vec{z}, t). \tag{2.16}$$

Первое уравнение по начальному состоянию z^O и экзогенным (независимым) переменным *x, v, h* определяет вектор-функцию *z(t)*, а второе по полученному значению состояний *z(t)* – эндогенные переменные на выходе системы *y(t)*. Таким образом, цепочка уравнений объекта <вход – состояния – выход> позволяет определить характеристики системы

$$\vec{y}(t) = F[\phi(\vec{z}^{o}, \vec{x}, \vec{v}, \vec{h}, t)]. \tag{2.17}$$

В общем случае время в модели системы на интервале времени моделирования ***(0, T)*** может рассматриваться как непрерывное, так и дискретное, т.е. квантованное на отрезке длиной $\Delta \boldsymbol{t}$ временных единиц каждый. Тогда $T = m\,\Delta t$, где $m = 1, \ldots, m_T$ – число интервалов дискретизации.

Рассмотренные математические соотношения представляют собой математические схемы общего вида и позволяют описать широкий класс систем. Однако в практике моделирования объектов в области системотехники и системного анализа на первоначальных этапах исследования системы рационально использовать типовые математические схемы:

- дифференциальные уравнения;
- конечные вероятностные автоматы;
- системы массового обслуживания и некоторые другие.

Типовые математические схемы имеют преимущества простоты и наглядности при существенном сужении возможностей применения.

Для описания больших информационных систем, к которым относятся АСУ, наиболее перспективным является применение агрегативных моделей.

Агрегативные модели (системы) позволяют описать широкий круг объектов исследования. При агрегативном описании сложный объект (система) разбивается на конечное число частей (подсистем), сохраняя при этом связи, обеспечивающие взаимодействие частей.

Таким образом, при построении математических моделей систем можно выделить следующие основные подходы:

1.Непрерывно-детерминироавнный (например, дифференциальные уравнения);
2.Дискретно-детерминированный (конечные автоматы);
3.Дискретно-стохастический (вероятностные автоматы);
4.Непрерывно-стохастический (системы массового обслуживания);
5.Обобщенный или универсальный (агрегативные системы).

2. Непрерывно-стохастические модели (Q – схемы)

2.1.Характеристика непрерывно-стохастических моделей.

Непрерывно-стохастический подход рассмотрим на основе систем массового обслуживания, которые будем называть Q – схемами. Системы массового обслуживания представляют собой класс математических схем, разработанных в теории массового обслуживания для формализации процессов функционирования систем, которые по своей сути являются процессами обслуживания.

В качестве процесса обслуживания могут быть представлены процессы функционирования экономических, технических, производственных и других систем. Например, потоки поставок продукции некоторому предприятию; потоки заявок на обработку информации северу от удаленных терминалов и т.д. Процессы обслуживания рассматриваются также в телефонных службах, на автозаправочных станциях, в аэропортах при управлении взлетом и посадкой самолетов, в торговой сети при обслуживании покупателей и т.д.

Характерным для работы таких объектов является стохастический характер процесса их функционирования, т.е. случайное появление заявок на обслуживание и завершение обслуживания в случайные моменты времени.

Рассмотрим основные понятия массового обслуживания, используемые в Q-схемах при аналитическом и имитационном подходах.

2.2.Понятие прибора обслуживания

В любом элементарном акте обслуживания можно выделить две основные составляющие:

1.Ожидание обслуживания заявкой;

2.Собственно обслуживания заявки.

Это можно изобразить в виде некоторого i-го прибора обслуживания $П_i$ (рис.2.1), состоящего из

накопителя заявок $Н_i$,

канала обслуживания $К_i$.

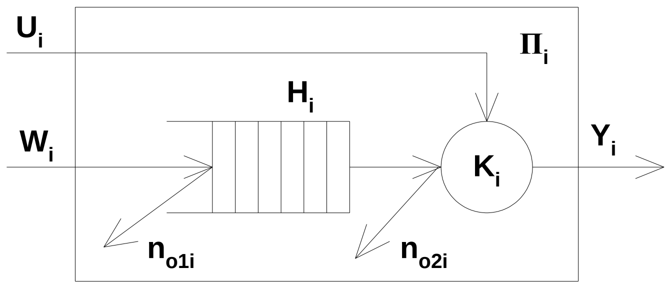

Рис.2.1. Прибор обслуживания заявок

В накопителе заявок может одновременно находится $l_i = \overline{0, L_i}$ заявок, где L_i – емкость i-го накопителя.

2.3.Понятие потока событий. Потоки событий однородные и неоднородные

На каждый прибор обслуживания $П_i$ поступают потоки событий:

- в накопитель $Н_i$ потоки заявок ω_i ;
- в канал $К_i$ – поток обслуживаний U_i .

Потоком событий называется последовательность событий, происходящих одно за другим в случайные моменты времени.

Различают потоки однородных и неоднородных событий.

Поток событий называется однородным, если он характеризуется только моментами наступления этих событий (вызывающими моментами) и задаются последовательностью

$$\{t_n\} = \{0 \le t_1 \le t_2 \ldots \le t_n \le \ldots\}, \tag{3.1}$$

где t_n – момент наступления *n*-го события.

Однородный поток событий может быть задан также в виде последовательности промежутков времени между *n*-м и (*n*-1)-м событиями $\{\tau_n\}$, которая однозначно связана с последовательностью вызывающих моментов $\{t_n\}$ следующим образом:

$$\tau_n = t_n - t_{n-1}, \tag{3.2}$$

Например, при n = 1 и $t_o = 0$, $\tau_1 = t_1$;

при n = 2 $\tau_2 = t_2 - t_1$. и т.д.

Потоком неоднородных событий называется последовательность $\{t_n, f_n\}$, где t_n – вызывающие моменты; f_n – набор признаков события (наличие приоритета; возможность обслуживания тем или иным типом канала и т.д.).

Заявки, обслуженные каналом K_i, и заявки, покинувшие прибор Π_i по различным причинам необслуженными (например, из-за переполнения накопителя), образуют выходной поток заявок $y_i \in Y$.

2.4.Понятие состояния

Процесс функционирования прибора обслуживания Π_i можно представить как процесс изменения состояний его элементов во времени $z_i(t)$. Переход в новое состояние для прибора Π_i означает изменение количества заявок, которые в нем находятся (в канале $К_j$ и накопителе $Н_i$).

Вектор состояний для прибора Π_i имеет вид

$$z_i = (z_i^H, z_i^K), \tag{3.3}$$

где Z_i^H - состояние накопителя (0 – накопитель пустой, n – в накопителе n заявка, $\mathbf{L^н}$ – накопитель заполнен);

$Z_i^к$ - состояние канала (0 – канал свободен, 1 – канал занят, 2 – канал заблокирован).

2.5.Многоканальные и многофазные, разомкнутые и замкнутые СМО

При моделировании систем, имеющих более сложные структурные связи и алгоритмы поведения, для формализации используются не отдельные приборы обслуживания, а Q-схемы, образуемые композицией многих элементарных приборов обслуживания $\mathbf{\Pi_i}$ или их элементов.

Если ***k*** различных приборов обслуживания соединены параллельно, то имеет место многоканальное обслуживание (многоканальная Q-схема), рис. 2.2.

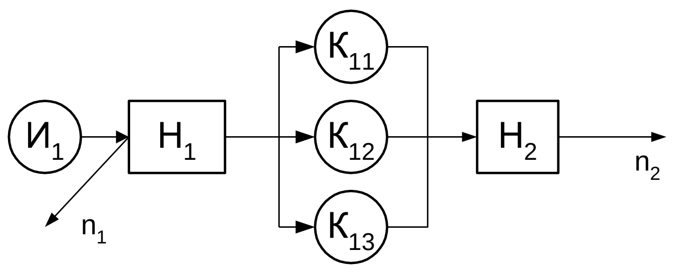

Рис.2.2. Трехканальная СМО

Если приборы $\mathbf{\Pi_i}$ и их параллельные композиции соединены последовательно, то имеет место многофазное обслуживание (многофазная Q-схема), рис 2.3.

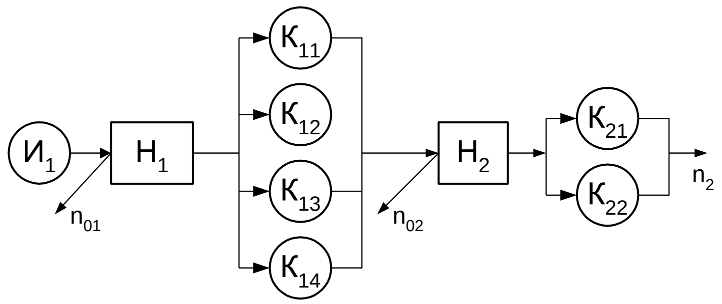

Рис.2.3. Двухфазная СМО

Связи между элементами Q-схемы изображают в виде стрелок (линий потока, отражающих направление движения заявок). Различают разомкнутые и замкнутые Q-схемы.

В замкнутых Q-схемах имеются обратные связи, по которым заявки перемещаются в направлении, обратном движению вход-выход, рис.2.4.

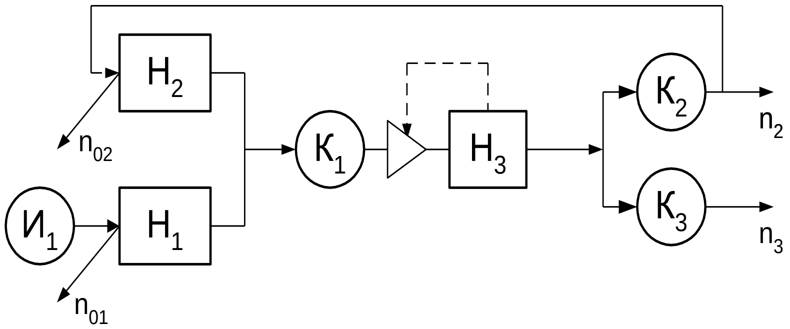

Рис.2.4.СМО замкнутой структуры

В разомкнутой Q-схеме выходной поток обслуженных заявок не может снова поступить на какой-либо элемент, т.е. обратная связь отсутствует, рис.2.2, 2.3.

Для задания Q-схемы можно использовать оператор сопряжения ***R***, отражающий взаимосвязь элементов структуры системы. При этом необходимо описать также алгоритмы функционирования системы (**А**), которые определяют набор правил поведения в ней заявок в различных неоднозначных ситуациях. В зависимости от места возникновения таких ситуаций различают: алгоритмы ожидания заявок в накопителе $\mathbf{H_i}$ и алгоритмы обслуживания заявок в канале $\mathbf{K_i}$.

2.6.Понятие приоритета. Приоритеты статические и динамические, относительные и абсолютные

В реальных системах заявки неоднородны, Они могут различаться, например, по важности. Неоднородность заявок учитывается с помощью введения классов приоритетов.

Приоритеты бывают статические и динамические, а также относительные и абсолютные.

Статические приоритеты назначаются заранее и не зависят от состояния Q-схемы.

Динамические приоритеты возникают при моделировании в зависимости от возникших ситуаций.

В зависимости от правил выбора заявок из накопителя $\mathbf{H_i}$ на обслуживание каналом $\mathbf{K_i}$, можно выделить относительные и абсолютные приоритеты.

Относительный приоритет означает, что поступающая в накопитель $\mathbf{H_i}$ заявка с более высоки приоритетом ожидает окончания обслуживания предыдущей заявки каналом $\mathbf{K_i}$ и после этого занимает канал.

Абсолютный приоритет означает, что заявка с более высоким приоритетом, поступающая в накопитель $\mathbf{H_i}$, прерывает обслуживание каналом $\mathbf{K_i}$ заявки с более низким приоритетом и сама занимает канал.

2.7.Задание непрерывно-стохастических моделей

Q-схема, описывающая процесс функционирования системы массового обслуживания любой сложности, однозначно задается в виде

$$\boldsymbol{Q = \langle W, U, H, Z, R, A \rangle,} \qquad (2.4)$$

где $\boldsymbol{W}$ – подмножество входных потоков;

$\boldsymbol{U}$ – подмножество потоков обслуживания;

$\boldsymbol{H}$ – подмножество собственных параметров (например, емкость накопителя, параметры входного потока и потоков обслуживания и т.д.);

$\boldsymbol{Z}$ – подмножество параметров состояния;

$\boldsymbol{R}$ – оператор сопряжения, отражающий взаимосвязь элементов структуры (источников, накопителе, каналов и т.д.) между собой;

$\boldsymbol{A}$ – алгоритм обслуживания заявок.

2.8.Общие принципы построения моделирующих алгоритмов Q-схем

Характерной особенностью процесса функционирования Q-схем является его стохастический характер.

Для формализации реальной системы с помощью Q-схемы, необходимо построить ее структуру (структурную схему). В качестве элементов такой структуры используются элементы трех типов: **И** – источник, **Н** – накопитель, **К** – канал обслуживания заявок.

Рассмотрим пример структуры системы, представленной в виде Q-схемы, рис. 2.5.

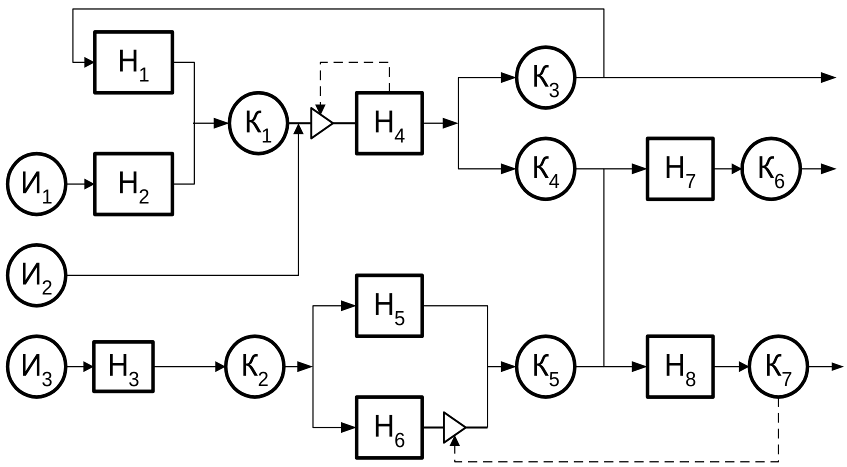

Рис. 2.5. Структура системы, представленной в виде Q-схемы

На данной схеме кроме связей, отражающих движения заявок в Q-схеме (сплошная линия), показаны управляющие связи.

В качестве управляющих связей могут использоваться обратные связи блокировок обслуживающих каналов (по входу и по выходу) и клапаны. На схеме клапаны изображены в виде треугольников, а управляющие связи показаны пунктирными линиями.

Блокировка канала по входу означает, что этот канал отключается от входящего потока заявок.

Блокировка канала по выходу указывает, что заявка, уже обслуженная блокированным каналом, остается в этом канале до момента снятия блокировки (открытия клапана).

Если перед накопителем нет клапана, то при переполнении накопителя будут иметь место потери заявок. Тогда помимо выходящего потока обслуженных заявок можно говорить о потоке потерянных заявок.

Q-схему считают заданной, если определены:

1.Поток событий (входящие потоки заявок и потоки обслуживаний для каждого накопителя и канала);

2.Структура системы (число фаз $L^ф$; число каналов обслуживания $L^к$; число накопителей $L^н$ каждой из фаз обслуживания и связи между источниками, накопителями и каналами);

3.Алгоритмы функционирования системы (дисциплина ожидания заявок в накопителях и выбора их на обслуживания в каналы, а также правила ухода заявок из накопителей и каналов).

3.9.Требования, предъявляемые к моделирующим алгоритмам Q -схем

Моделирующие алгоритмы должны отвечать следующим основным требованиям:

1.Обладать универсальностью относительно структуры, алгоритмов функционирования и параметров системы;

2.Обеспечивать одновременную и независимую работу элементов системы;

3.Укладываться в приемлемые затраты ресурсов компьютера для реализации машинного эксперимента;

4.Иметь возможность разбиения на достаточно автономные логические части, т.е. обладать блочной структурой;

5.Гарантировать выполнение рекурентного правила: событие, происшедшее в момент времени t_k, может моделироваться только после того, как промоделированы все события, произошедшие в момент времени $t_{k-1} < t_k$.

2.9.Принципы построения моделирующих алгоритмов Q -схем: Δt и δz

Существуют два основных принципа построения моделирующих алгоритмов Δt и δz. При построении моделирующего алгоритма Q-схемы по принципу Δt, т.е. алгоритма с детерминированным шагом, для построения адекватной модели необходимо определить минимальный интервал времени между соседними событиями $\Delta t' = \min\{u_i\}$ во входящих потоках и потоках обслуживания и принять шаг моделирования равным Δt'.

В моделирующих алгоритмах построенных по принципу δz, т.е. в алгоритмах со случайным шагом, элементы Q-схемы при моделировании просматриваются только в моменты особых состояний, т.е. в моменты появления заявок из источника или изменения состояний каналов. В этом случае длительность шага Δt является переменной и зависит как от особенностей самой системы, так и от воздействий внешней среды.

2.10.Синхронный и асинхронный способы реализации моделирующих алгоритмов, построенных по принципу δz

Моделирующие алгоритмы со случайным шагом могут быть реализованы синхронным и асинхронным способами.

При синхронном способе один из элементов Q-схемы (источник, накопитель или канал) выбирается в качестве ведущего и по нему синхронизируется весь процесс моделирования.

При асинхронном способе ведущий элемент не используется, а очередному моменту моделирования (просмотру элементов Q-схемы) может соответствовать любое особое состояние всего множества элементов (источников, накопителей и каналов).

2.11.Принципы реализации моделирующих алгоритмов Q-схем по принципу Δt.

Реализация моделирующих алгоритмов Q-схем может быть выполнена с помощью пакетов прикладных программ, созданных на базе алгоритмических языков общего назначения (C++, Pascal,Visual Basc, Java и других), с помощью пакетов MathCad, MatLab, Mathematica и т.д., либо с помощью специализированных языков имитационного моделирования: Simula, Simscript, GPSS и др.

Для имитации процесса функционирования Q-схем на компьютере целесообразно организовать следующие массивы:

1.Массив ***Z*** - текущих значений состояний $z_{k,j}$ соответствующих каналов $K_{k,j}$;

2.Массив ***T*** - моментов времени окончания обслуживания заявок каналами;

3.Массив ***ZN*** – текущих значений состояний накопителей;

Процедура моделирования процесса обслуживания каждым элементарным каналом $\boldsymbol{K_{k,j}}$ сводится к следующему:

1.Моделируется случайное число – длительность обслуживания – с законом распределения, соответствующим обслуживанию данного канала $\boldsymbol{K_{k,j}}$;;

2.Вычисяется время окончания обслуживания $t_{k,j}$;

3.Фиксируется состояние канала $Z_{k,j}$;:

$\boldsymbol{Z_{k,j}} = 1$, если канал занят (обслуживает заявку);

$\boldsymbol{Z_{k,j}} = 0$, если канал свободен;

$\boldsymbol{Z_{k,j}} = 2$, если канал заблокирован.

При поступлении заявки в накопитель $\boldsymbol{H_i}$ к его содержимому добавляется единица, т.е. $\boldsymbol{z_{,i} = z_{,i} + 1}$, а при уходе заявки из накопителя $\boldsymbol{H_i}$ например, на обслуживание в канал, его состояние уменьшается на единицу $\boldsymbol{z_{,i} = z_{,i} - 1}$.

Рассмотрим процесс моделирования системы массового обслуживания на примере Q-схемы, приведенной на рисунке 2.6.

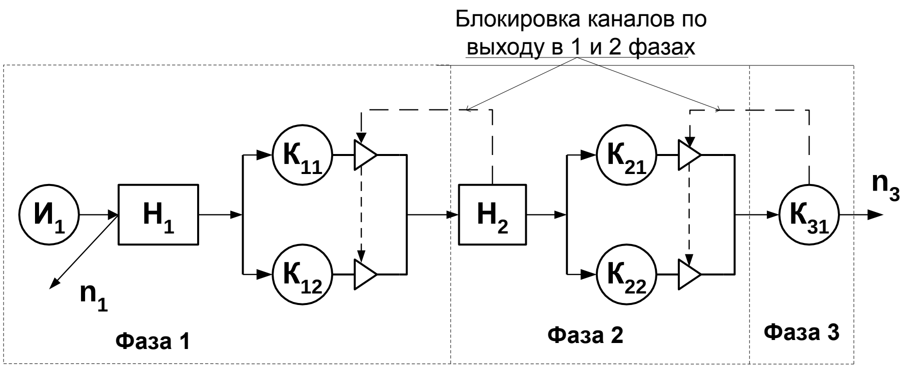

Рис.2.6. Q-схема общего вида

На рис. 2.6 представлена трехфазная Q-схема с блокировкой каналов по выходу в 1-й и 2-й фазах обслуживания.

В качестве выходящих потоков рассматриваются: поток n_1 потерянных заявок из накопителя H_1 и поток обслуженных заявок n_3 из канала K_{31}.

Для имитационной модели данной Q-схемы примем следующие переменные и уравнения:

Эндогенные (зависимые) переменные:

р – вероятность потери заявок.

Экзогенные (независимые) переменные:

t_m – момент времени появления очередной заявки из источника;

t_{kj} – момент времени окончания обслуживания каналом K_{kj} очередной заявки, $k = 1, 2, 3; j=1,2$,

t_n – системное время.

Вспомогательные переменные

zn_i – состояние накопителя H_i;

$z_{k,j}$ – состояние канала $K_{k,j}$, $k=1, 2, 3; j=1, 2$.

Параметры

L_k – емкость k-го накопителя.

Переменные состояния:

n_1 – число потерянных заявок в накопителе H_1;

n_3 – число обслуженных заявок, т.е вышедших из 3-й фазы;

Уравнение модели:

$$p = \frac{n_1}{n_1 + n_3} = \frac{n_1}{N}. \quad (2.4)$$

2.12.Схема моделирующего алгоритма СМО по принципу Δt

Схема моделирующего алгоритма, построенного по принципу Δt, приведен на рис. 2.7.

Все накопители и каналы системы в процессе моделирования просматриваются, начиная с обслуживающего канала последней фазы по направлению к источнику заявок 1-й фазы.

Процедура обслуживания заявок каналами СМО реализуется также на основе функции *Rand*, которая выдает длительность интервала обслуживания очередной заявки каналом $K_{k,j}$. Эта процедура использует генератор псевдослучайных чисел с законом распределения, соответствующим данному каналу обслуживания.

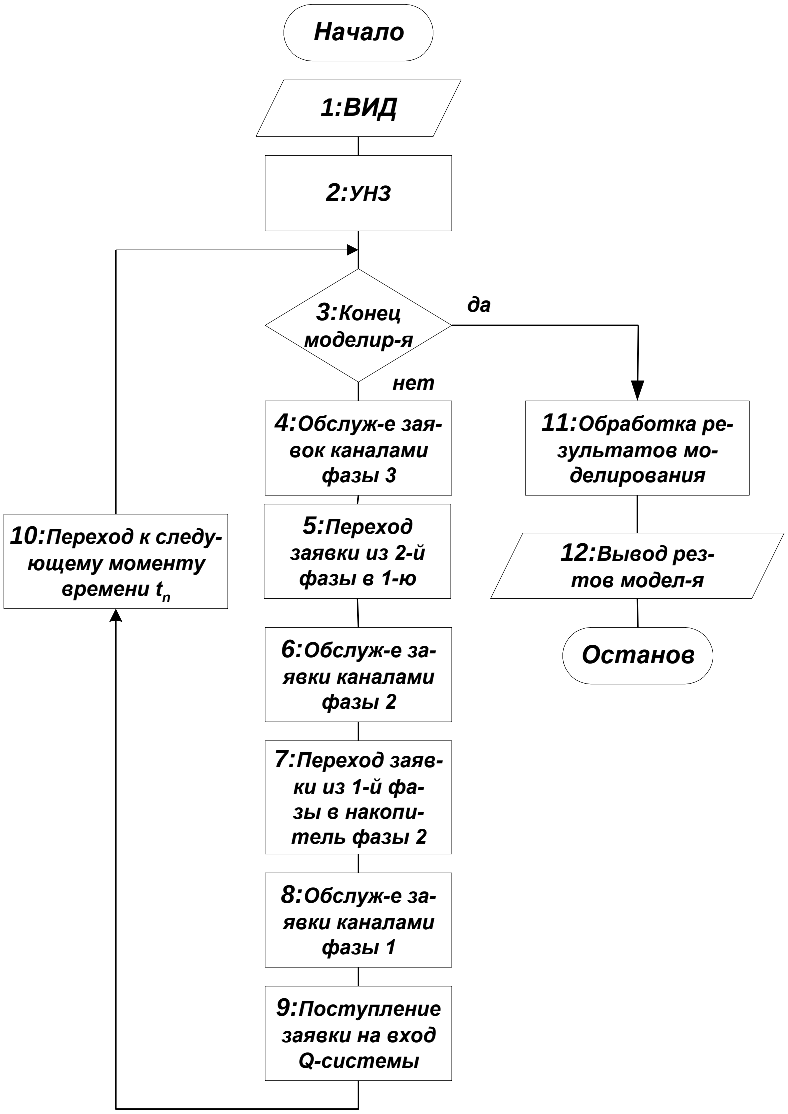

Рис. 2.7.Укрупненная схема моделирующего алгоритма

Блок 10 схемы алгоритма служит для расчета системного времени t_n с помощью оператора: $t_n = t_{n-1} + \Delta t$.

Для определения момента остановки моделирования, по числу реализаций или по длине интервала времени T, проводится проверка соответствующих условий (блок 3).

Алгоритм функционирует следующим образом.

После ввода исходных данных (блок ВИД) и установки начальных значений переменных (блок УНЗ), если условие окончания моделирования не выполняется, то производится имитация обслуживания заявок каналом $K_{3,1}$ 3-й фазы Q-схемы, (блок 4).

Схема алгоритма блока имитации обслуживания заявок каналом фазы 3 (блок 4) приведена на рис. 2.8.

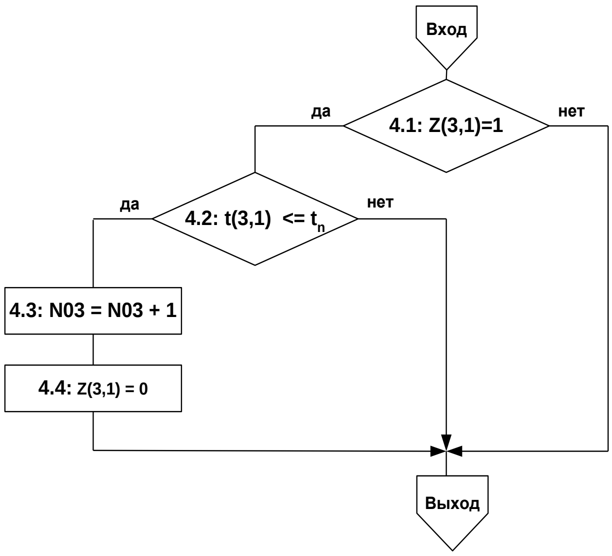

Рис. 2.8. Схема алгоритма блока имитации обслуживания заявок каналом фазы 3

Алгоритм работает следующим образом. Анализируется состояние канала $\boldsymbol{K_{3,1}}$. И если обслуживание в этом канале закончилось (операторы 4.1, 4.2), то фиксируется выход из системы очередной обслуженной заявки (оператор 4.3) и производится освобождение канала $\boldsymbol{K_{3,1}}$ (оператор 4.4).

Переменная t_n в операторе 4.2 определяет системное время.

Работа каналов 2-й фазы Q-схемы во взаимодействии ее с каналом третьей фазы представлена схемой алгоритма рис. 2.9:

Приведенный алгоритм работает следующим образом. Организуется последовательный просмотр каналов 2-й фазы (операторы 1, 9, 10). При просмотре каждого канала определяется, имеются ли в j-ом канале 2-й фазы заявки, ожидающие обслуживания в канале $К_{3,1}$ (операторы 2, 3).

Если в момент t_n такие заявки имеются и канал $К_{3,1}$ свободен (оператор 4), то в соответствии с дисциплиной обслуживания (по очереди или по приоритетам) выбирается одна из заявок и: имитируется ее обслуживание (оператор 6); фиксируется занятие канала 3-й фазы (оператор 7) и освобождается j-й канал 2-й фазы (оператор 8).

Если же канал $К_{3,1}$ занят (оператор 4), то фиксируется блокировка канала 2-й фазы (оператор 5).

Затем имитируется взаимодействие заявок в накопителе $\mathbf{H_2}$ и каналах 2-й фазы. Алгоритм этого процесса приведен на рис. 2.10.

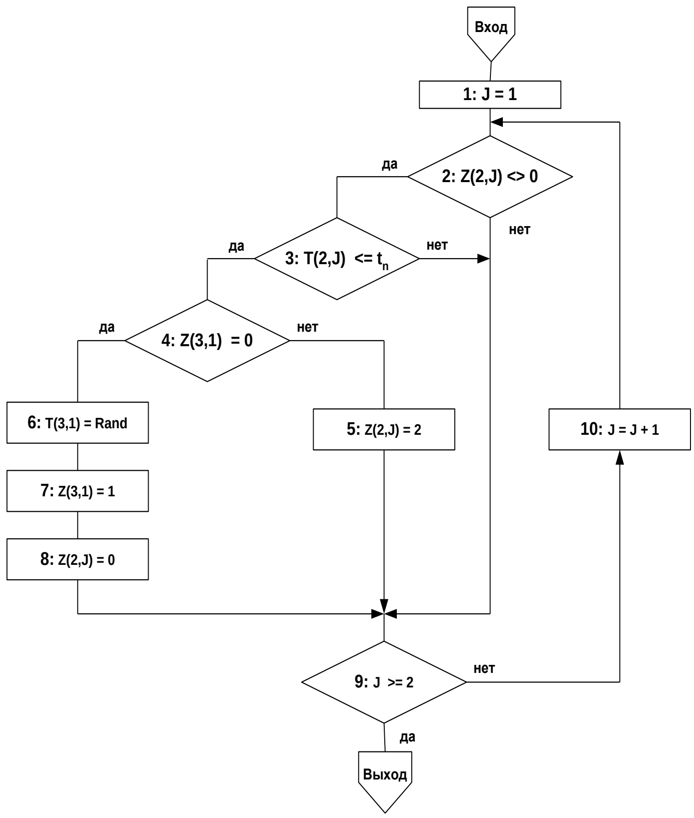

Рис. 2.9. Схема алгоритма блока имитации обслуживания заявок каналами фазы 2

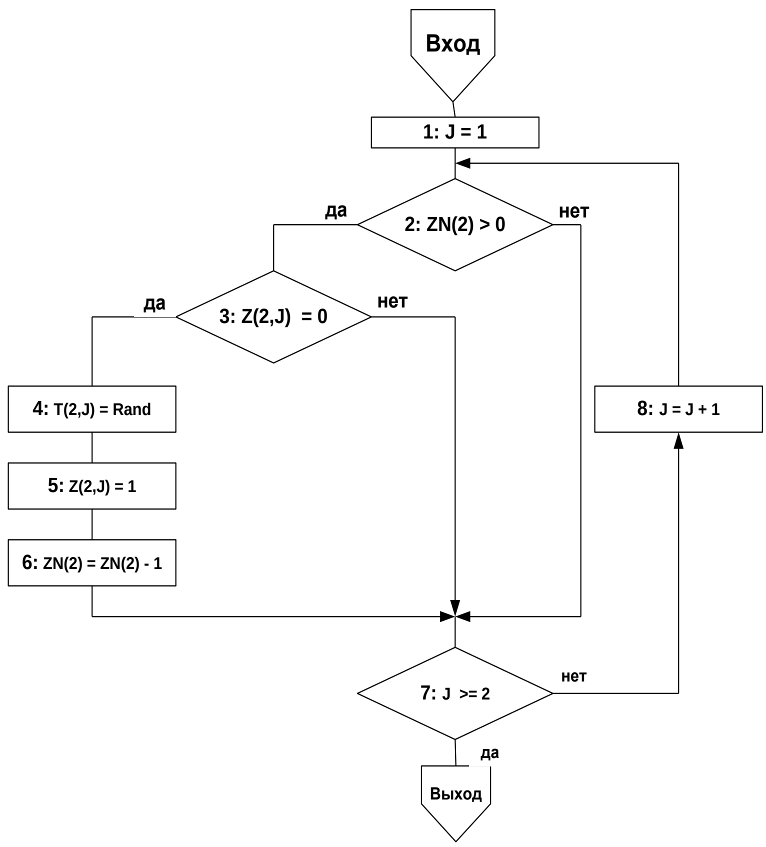

Рис. 2.10. Схема алгоритма взаимодействия накопителя и каналов фазы 2

Работа схемы начинается с проверки наличия заявок в накопителе H_2. Если в нем имеются заявки (оператор 2) и свободные каналы 2-й фазы (оператор 3), то имитируется обслуживание очередной заявки одним из свободных каналов этой фазы (операторы 4, 5) и освобождается место в накопителе H_2 (оператор 6).

Следующим работает блок взаимодействия каналов 1-й фазы и накопителя 2-й фазы. Алгоритм такого взаимодействия приведен на рис. 2.11.

На рис. 2.11 приняты следующие обозначения:

LH(2), ZN(2) – массивы емкостей и количества заявок в накопителях 1-й и 2-й фаз;

Индекс *j* используется для обозначения элементов 1-й фазы, *i* – второй.

Работа схемы начинается с проверки наличия заявок в канале $K_{1,j}$ 1-й фазы, требующих обслуживания в момент t_n (операторы 3,4).

Если во второй фазе свободных каналов нет (оператор 5), но в накопителе имеются свободные места (оператор 6), то моделируется размещение заявки в накопитель H_2 (оператор 7) и освобождается *j-й* канал 1-й фазы (оператор 8). Если в накопителе H_2 свободных мест нет, то фиксируется блокировка канала 1-й фазы (оператор 9). При наличии свободных каналов во 2-й фазе выполняется обслуживание заявки (оператор 10), фиксируется занятость *i-го* канала 2-й фазы (оператор 11) и освобождение *j-го* канала 1-й фазы (оператор 12).

Затем операторы внутреннего цикла выполняются повторно, т.к. из первой фазы во вторую одновременно могут переместиться две заявки.

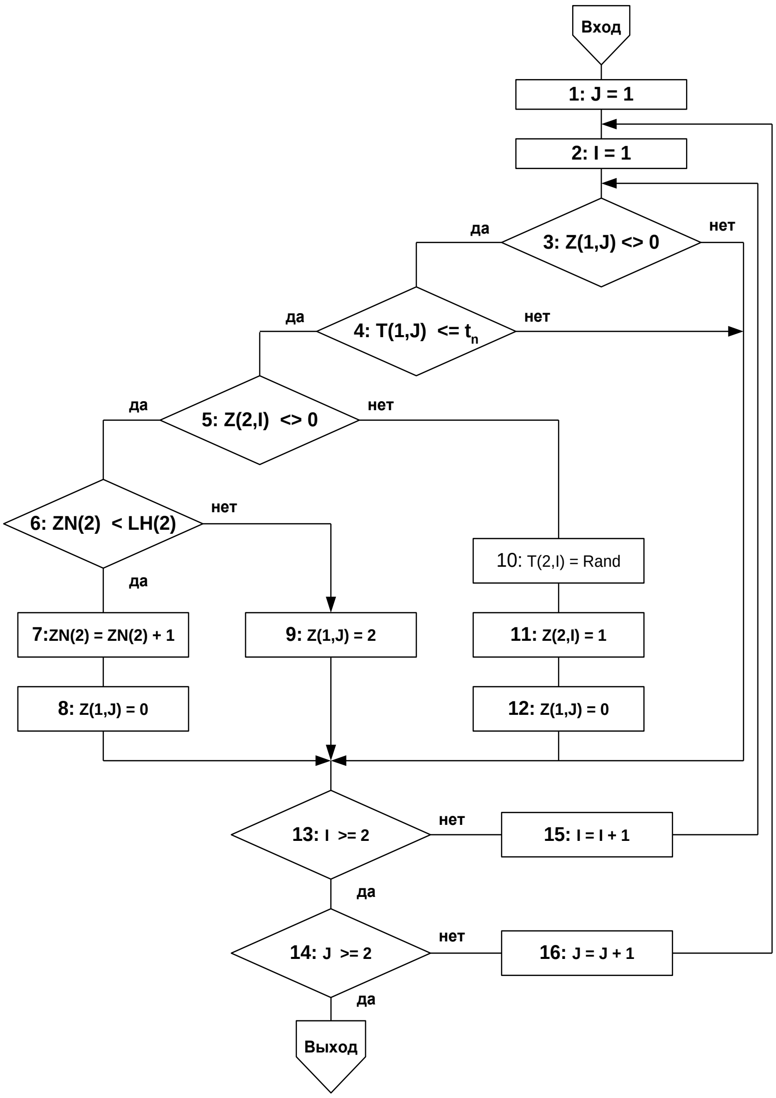

Рис. 3.11. Схема алгоритма взаимодействия и каналов фазы 1 и накопителя H_2

Далее вступает в работу блок имитации взаимодействия заявок в накопителе $\boldsymbol{H_1}$ и каналов 1-й фазы. Схема алгоритма этого блока представлена на рис 2.12.

В начале этого блока проверяется наличие заявок в накопителе $\boldsymbol{H_1}$ (оператор 2) и возможность обслуживания их ***j-м*** каналом, т.е. свободен ли ***j-й*** канал (оператор 3).

Если в накопителе $\boldsymbol{H_1}$ имеются заявки и один из каналов $\boldsymbol{K_{1,j}}$ свободен, то имитируется обслуживание заявки ***j-м*** каналом 1-й фазы (оператор 4), фиксируется занятие ***j-го*** канала (оператор 5) и освобождение одного места в накопителе $\boldsymbol{H_1}$ (оператор 6).

Последним вступает в работу блок взаимодействия источника ***И*** и накопителя 1-й фазы $\boldsymbol{H_1}$ с учетом занятости каналов этой фазы, рис. 2.13.

В данной схеме алгоритма операторы 2, 6, 7, 8, 9 являются вспомогательными операторами циклов.

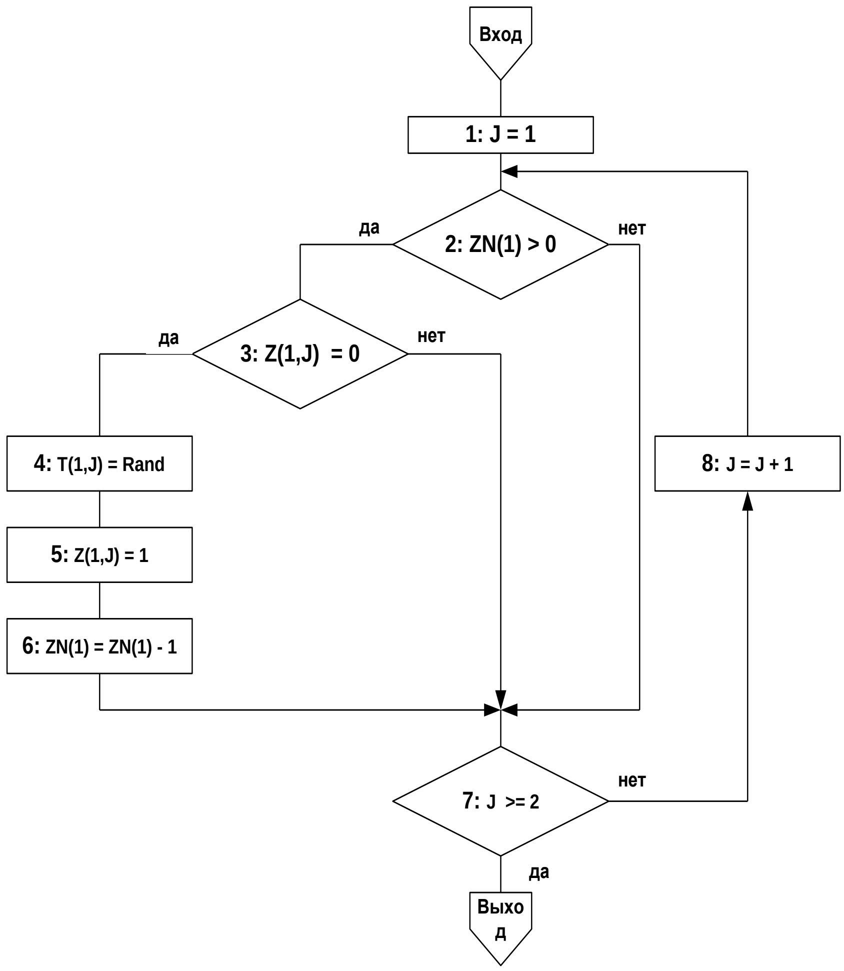

Рис. 2.12. Схема алгоритма взаимодействия заявок
в накопителе $\boldsymbol{H_1}$ и каналах 1-й фазы.

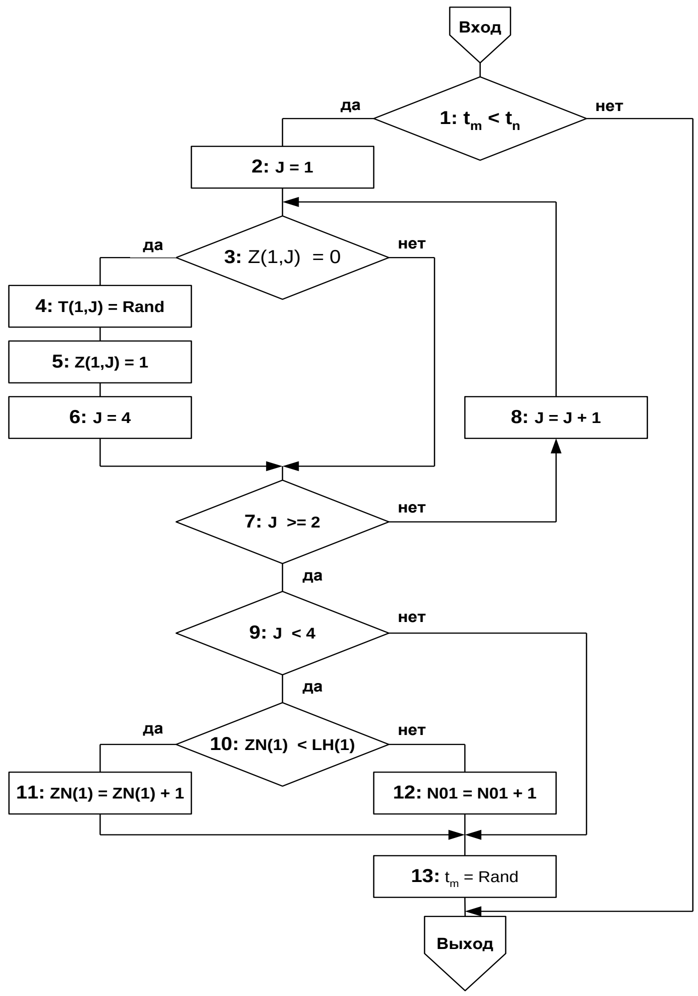

Рис.2.13.Схема алгоритма взаимодействия источника ***И*** и накопителя $\boldsymbol{H_1}$ первой фазы

Если в момент времени $t_n > t_m$ поступила заявка из источни ка ***И*** (оператор 1), то при наличии свободного канала $\boldsymbol{K_{1,j}}$ (оператор 3) начнется ее обслуживание одним из каналов.

Если же оба канала заняты, то при наличии места в накопителе $\boldsymbol{H_1}$ (оператор 10) заявка будет поставлена в очередь в накопитель $\boldsymbol{H_1}$. При отсутствии места в накопителе $\boldsymbol{H_1}$ заявка будет потеряна (оператор 12).

После отработки операторов 11 или 12 управление передается блоку 13, который вычисляет момент $\mathbf{t_m}$ поступления следующей заявки в Q-схему.

После отработки всех боков алгоритма (рис.3.8…3.13) будет выполняться блок 10 рис.3.4, который увеличивает системное время t_n на величину дискретизации времени Δt. Затем управление передается блоку 3 рис.2.7, который после набора необходимой статистики производит обработку и выдачу результатов моделирования (блоки 11 и 12 рис. 2.7).

2.12.Построение синхронного моделирующего алгоритма Q-схем по принципу δz

Рассмотрим особенности построения моделирующего алгоритма рассмотренной ранее Q-схемы, рис.3.6, по принципу ***δz***.

Построим синхронный моделирующий алгоритм. В качестве синхронизирующего элемента выберем выберем источник заявок ***И***. В этом случае моменты времени t_n - просмотра элементов СМО будут совпадать с моментами времени t_m – поступления заявок в систему.

В момент времени t_n на вход 1-й фазы Q-схемы поступает очередная заявка из источника И. На протяжении времени t_{n-1} - t_n в Q-схеме могли произойти изменения состояний ее накопителей и каналов.

Все эти изменение состояний k-й фазы необходимо промоделировать до поступления в нее заявки из (k-1)-й фазы. (*)

Синхронный моделирующий алгоритм, схема которого представлена на рис. 2.14, работает следующим образом:

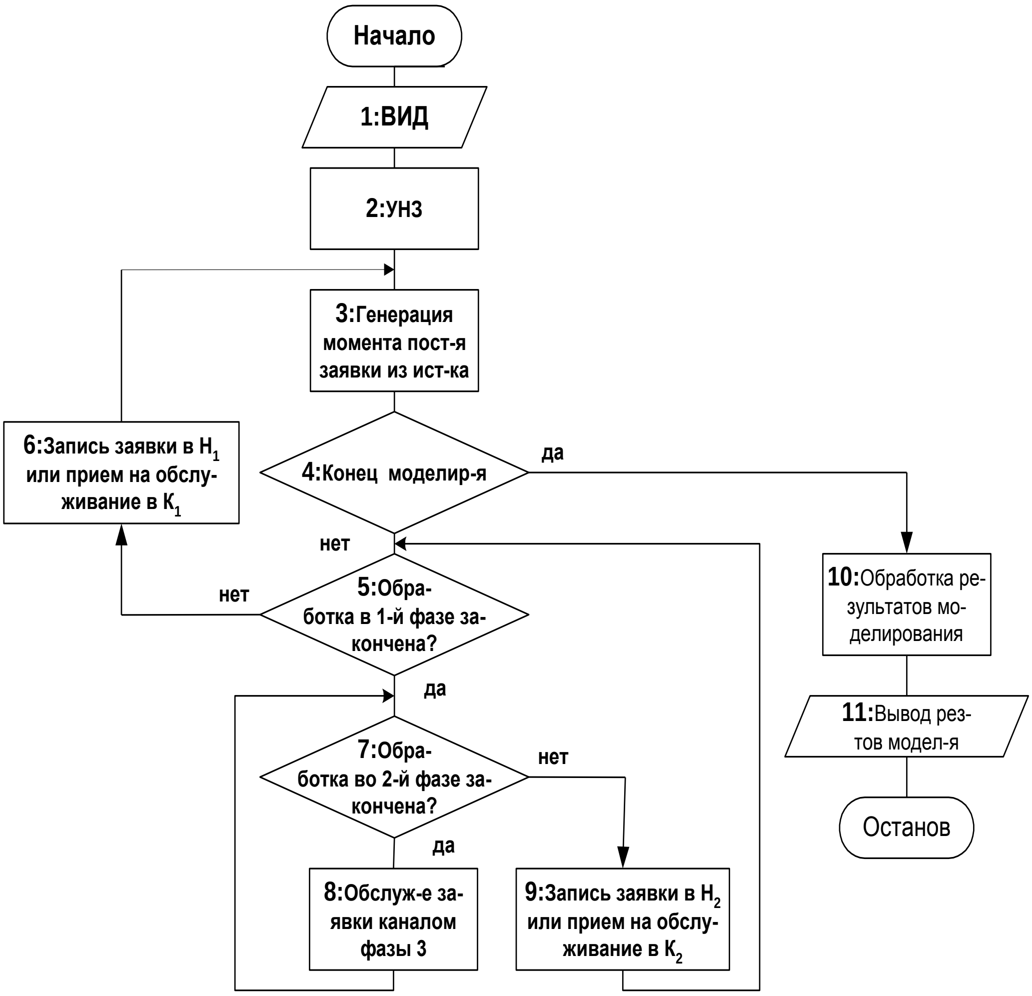

Рис. 2.14. Укрупненная схема синхронного моделирующего алгоритма Q-схемы

В момент времени t_n источник ***И*** генерирует заявку, блок 3 рис 3.14. Затем блок 5 в силу *условия* **(*)** должен передать управление блоку 7, а блок 7, по этой же причине, – блоку 8, который выполнит обслуживание заявки каналом K_3. После отработки блока 8 управление передается блоку 7, который по выходу "нет" передаст управление блоку 9, т.к. во второй фазе обработка не закончена. После блока 9 управление получит блок 5, который в свою очередь передаст управление блоку 6, т.к. обработка в 1-й фазе не закончена, поскольку во второй фазе могли освободиться каналы. После этого управление получает блок 6 генерации момента поступления заявок.

Работа большинства блоков этой схемы аналогична рассмотренной схеме моделирующего алгоритма по принципу Δt. Поэтому более подробно рассмотрим только взаимодействие синхронизирующего элемента ***И*** с остальной частью Q-схемы. Т.е. рассмотрим работу блока 6, имитирующего перемещение заявки из входного потока в накопитель H_1 или прием на обслуживание в один из каналов 1-й фазы, рис 2.15.

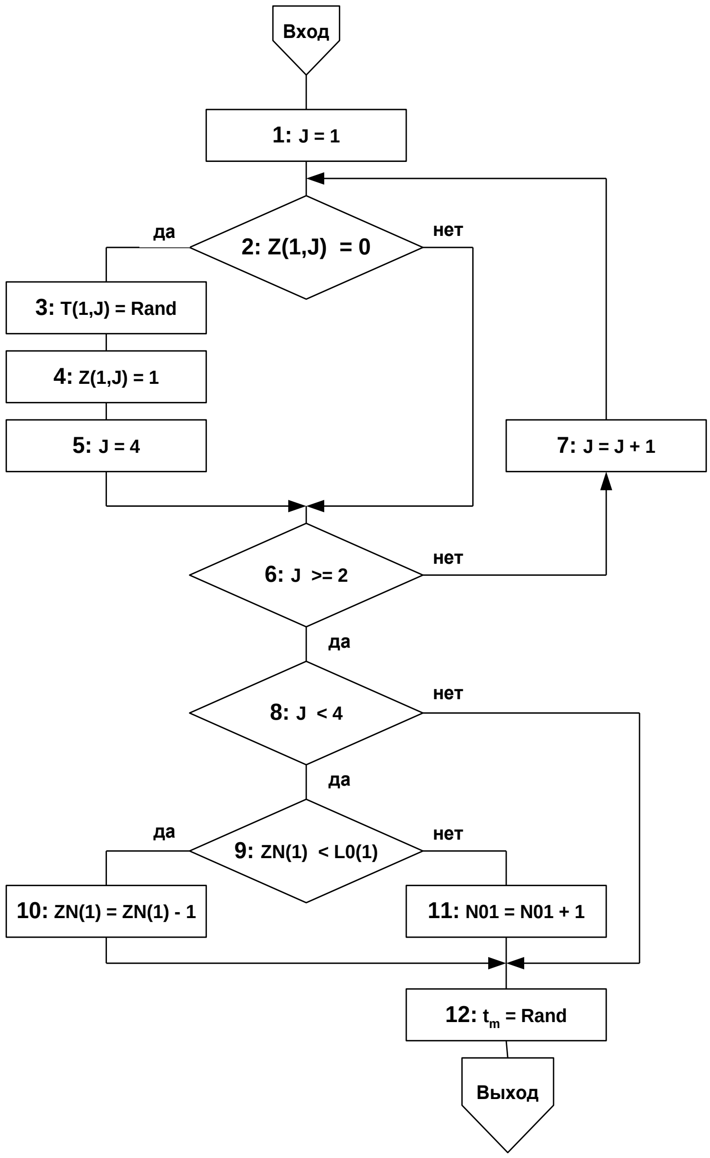

Рис 2.15. Укрупненная схема имитации перемещения заявок из входного потока в канал 1-й фазы

2.13.Построение асинхронного моделирующего алгоритма Q-схем по принципу δz

В данной схеме алгоритма дополнительно проверяется наличие свободных каналов в 1-й фазе (оператор 2). Если среди каналов 1-й фазы $K_{1,j}$ есть свободные, то выбирается один из них и имитируется обслуживание, т.е. определяется время окончания обслуживания в этом канале (оператор 3). Затем фиксируется, что этот канал занят (оператор 4) и осуществляется переход к следующему шагу моделирования.

Если же оба канала 1-й фазы заняты, то проверяется условие о наличии свободных мест в накопителе H_1 (оператор 9). Если такие места есть, то имитируется запись заявки в накопитель H_1 (оператор 10), в противном случае фиксируется потеря заявки (оператор 11).

Рассмотрим особенности построения асинхронного моделирующего алгоритма на примере той же Q-схемы, рис.3.6, который отличается от синхронного отсутствием ведущего (синхронизирующего) элемента. В этом случае переход к очередному шагу моделирования производится в моменты особых состояний: моменты окончания обслуживания заявок каналами или моменты поступления их из источника.

При таком принципе построения моделирующего алгоритма целесообразно процесс изменения состояний элементов Q-схемы рассматривать в направлении, противоположном направлению движения заявок в системе.

Такой алгоритм в плане просмотра состояний элементов Q-схемы аналогичен детерминированному моделирующему алгоритму, рассмотренному ранее, рис.3.7. Отличие состоит лишь в том, что отсчет системного времени производится следующим образом:

$$t_n=min\{min(t_н), min(t_{k,j}), min(t_s), t_m\} \qquad .5)$$

т.е. время очередного шага определяется как минимальное из:

ближайшего времени окончания хранения заявок в накопителях - $min(t_н)$;

минимального времени окончания обслуживания заявок в каналах Q-схемы - $min(t_{k,j})$;

ближайшего момента завершения пребывания заявки в системе - $min(t_s)$;

ближайшего времени поступления очередной заявки из источника в систему - t_m;

Укрупненная схема асинхронного циклического моделирующего алгоритма приведена на рис. 2.16.

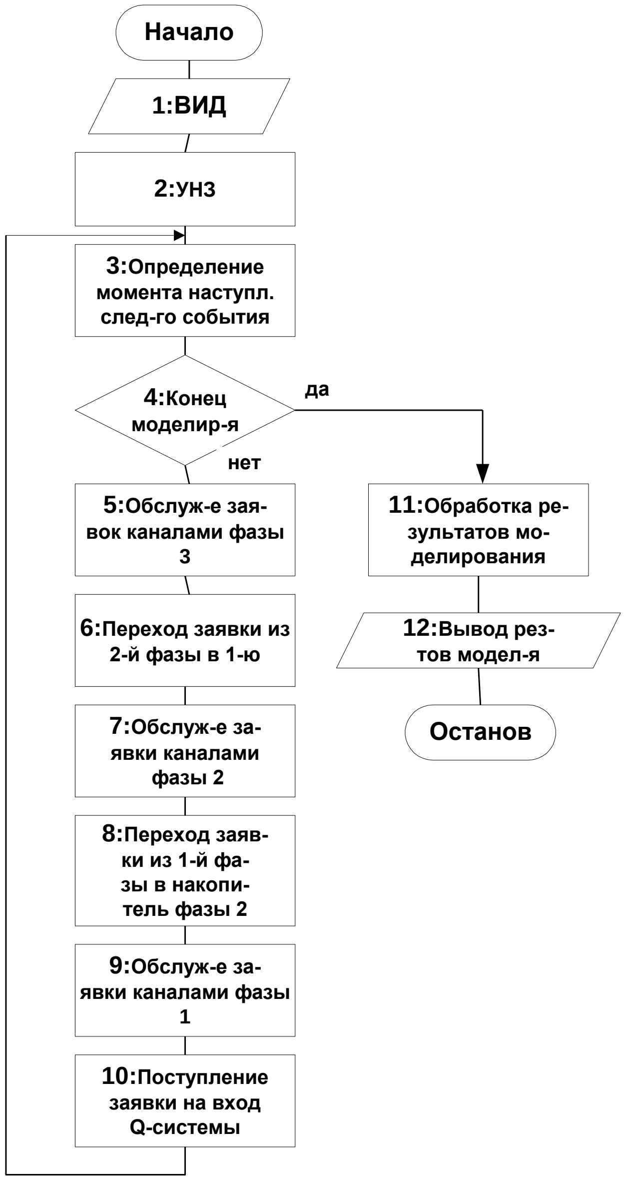

Рис 2.16.Укрупненная схема асинхронного циклического моделирующего алгоритма

3. Статистическое моделирование на ЭВМ

3.1.Моделирование случайных процессов

Одним из основных вопросов при моделирования случайных и статистическом моделировании является формирование последовательностей случайных чисел. Известны три основных способа генерации случайных чисел:

- аппаратный (физический);
- табличный (файловый);
- алгоритмический (программный).

Рассмотрим эти способы.

Аппаратный способ. Случайные числа генерируются специальной электронной приставкой – генератором (датчиком) случайных чисел, представляющей собой внешнее устройство компьютера. При этом в качестве физического эффекта, лежащего в основе таких генераторов, используются шумы в электронных и полупроводниковых приборах, явление распада радиоактивных элементов и т.д.

Характерным для этого способа является то, что он не гарантирует качество последовательности случайных чисел непосредственно во время моделирования системы, а также не позволяет повторно получить при моделировании одинаковые последовательности чисел.

Табличный способ. Этот способ предполагает размещение случайных чисел, оформленных в виде таблиц, во внешней или оперативной памяти компьютера.

Недостатки этого способа заключаются в в ограниченном размере таблиц и в потере быстродействия моделирования при считывании таблиц с внешних накопителей.

Алгоритмический способ. Данный способ основан на формировании случайных чисел с помощью специальных алгоритмов и программ.

При дискретном моделировании случайных воздействий имитация сводится к генерированию равномерно распределенных на интервале (0;1) случайных чисел и их последующему функциональному преобразованию.

К генераторам псевдослучайных чисел предъявляются следующие требования:

1.равномерный закон распределения;
2.статистическая независимыми;
3.возможность воспроизведения;
4.удовлетворение условию неповторяемости;
5.возможность реализации на компьютерах с ограниченными ресурсами.

Рассмотрим некоторые алгоритмы получения последовательностей псевдослучайных равномерно распределенных чисел, которые нашли применение в практике статистического моделирования систем.

Алгоритм, основанный на методе срединных квадратов

Берется $2n$-разрядное число x_i меньшее 1:

$x_i = 0,a_1a_2...a_{2n}$

Число x_i возводится в квадрат:

$x^2_i = 0,b_1b_2...b_{4n}$

Отбираются $2n$ средних разрядов

$x_{i+1} = 0,b_{n+1}b_{n+2}...b_{3n}$,

которые и образуют очередное число псевдослучайной последовательности.

Пример. $x_0 = 0,2152$,

$x^2_0 = 0,46\underline{6311}04$,

$x_1 = 0,6311$;

$x^2_1 = 0,39\underline{8287}21$; $x_2 = 0,8287$ и т.д.

Недостаток метода заключается в наличии корреляции между числами последовательности, а также то, что в ряде случаев наблюдается отсутствие случайности, например

$x_0 = 0,25$ $x^2_0 = 0,06\underline{2500}00$; $x_1 = 0,25$ и т.д.

Алгоритм, основанный на методе срединных произведений

берутся два произвольных n-разрядных числа x1 и x2;
перемножаются x1 и x2;
- из произведения $x_1 \cdot x_2$ выбирается n средних разрядов
- результат есть число x_3.

Алгоритм, основанный на методе вычетов.

Два числа назовем конгруэнтными по модулю q, если разность этих чисел делится на q без остатка. Символически это записывается в виде:

$$x = y \cdot mod(q) \tag{3.1}$$

Выражение $y \cdot mod \cdot q$ обозначает остаток от деления числа y на q. Рекурентное соотношение

$$x_{n+1} = k \cdot x_n \cdot mod(q) \tag{3.2}$$

порождает последовательность чисел $\{x\}$, распределенных приблизительно по равномерному закону.

Качество этой последовательности тем выше, чем больше число q.

К недостаткам данного метода относятся:

а) интервал распределения отличен от $[0;1]$;

б) имеется довольно сильная корреляционная зависимость в последовательности $\{x\}$.

Первый недостаток устраняется простым преобразованием

$$x_{ni}^{'} = \frac{x_n}{10^{\{\lg(q)\}+1}} \tag{3.3}$$

здесь $\{lg(q)\}$ - означает целую часть от $lg(q)$.

Для уменьшения корреляции специальным образом выбирают числа x_1 и k.

В частности можно использовать следующий алгоритм:

$$x_1 = 31415926$$

$$x_{n+1} = (1000 \cdot x_1 + 231) \; mod \; 10^8$$

$$x^*_{n+1} = (x_{n+1} / 10^8)$$

Мультипликативный метод

В основе этого метода лежит следующее рекуррентное соотношение:

$$a_i = (A \cdot a_{i-1} + C) \;\; mod \; M,$$

где a_i, a_{i-1} –очередное и предыдущее случайные числа, соответственно; A, c – константы; M – достаточно большое целое положительное число.

В данном методе чем больше M, тем длиннее неповторяемая последовательность псевдослучайных чисел.

Алгоритм Д. Лемера

Пусть задано $2n$-разрядное число x_o и постоянный множитель c. Тогда алгоритм Д. Лемера можно сформулировать в виде последовательности следующих шагов:

1) число x_o возвести в квадрат;
2) выбрать $2n$ последних цифр из x^2_o, результат присвоить x_o';
3) найти произведение $c \cdot x_o'$ и из него выделить первые $2n$ цифр, результат присвоить x_o'';
4) x_o'' возвести в квадрат;
5) выделить первые $2n$ цифр из $(x_o'')^2$, результат присвоить x_o''';

6) умножить x_o''' на c;
7) из произведения $c \cdot x_o'''$ выбрать последние $2n$ цифр, результат присвоить переменной x_o'''';
8) сложить x_o'' и x_o''''. Полученная сумма и является случайным числом x_1.
Для получения x_2 с x_1 необходимо проделать те же операции, и т.д.

Проверка качества псевдослучайных числовых последовательностей.

Результаты моделирования систем, полученные методом статистического моделирования, существенно зависят от качества используемых псевдослучайных чисел. Поэтому генераторы случайных чисел перед использованием их в моделировании должны пройти тщательное тестирование, которое включает проверки на: равномерность, стохастичность и независимость.

Проверка равномерности

Проверка равномерности псевдослучайных чисел может быть выполнена по гистограмме и с использованием косвенных признаков.

Проверка по гистограмме. Интервал $(0;1)$ разбивается на m равных частей. Тогда каждое из чисел x_i с вероятностью $p_j = 1/m$, $j = 1,\ldots,m$ попадает в один из подинтервалов.

В каждый интервал попадает N_j чисел. Относительная частота попадания случайных чисел в каждый из подинтервалов равна N_j/N, где N – общее количество случайных чисел.

Вид гистограммы представлен на рис. 3.1, где пунктирная линия соответствует теоретическому значению p_j, а сплошная – экспериментальному N_j/N.

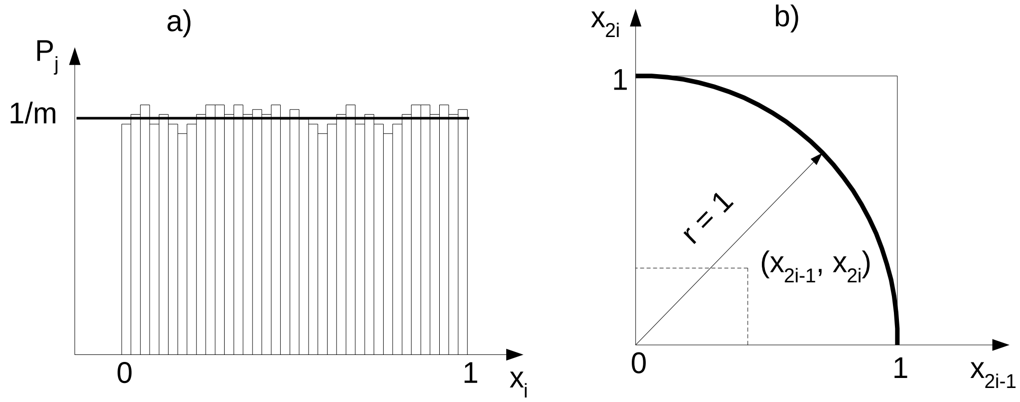

Рис.3.1. Проверка равномерности числовой последовательности

Если числа x_i принадлежат псевдослучайной равномерно распределенной последовательности, то при достаточно большом N экспериментальная гистограмма (сплошная линия, рис. 3.1,а) приблизится к теоретической $1/m$.

Оценка степени такого приближения может быть проведена с использованием критериев согласия. При этом обычно принимается $m = 200\ldots500$, $N = (100\ldots1000)\cdot m$.

Проверка равномерности по косвенным признакам. Генерируемая последовательность чисел разбивается на две последовательности

$x_1\ x_3\ x_5\ \ldots x_{2i-1}$
$x_2\ x_4\ x_6\ \ldots x_{2i},\ \ i = 1,\ldots,N/2.$

Затем производится следующий эксперимент. Если выполняется условие

$$x^2_{2i-1} + x^2_{2i} < 1,\ i=1,\ldots,N, \qquad (3.4)$$

то фиксируется наступление некоторого события и в счетчик событий добавляется единица.

После $N/2$ опытов, когда сгенерировано N чисел, в счетчике будет некоторое число $k \le N/2$.

Геометрически условие (3.4) означает, что точка

(x_{2i-1}, x_{2i}), $i = 1,\dots,N$ находится внутри четверти круга радиусом $r = 1$, что иллюстрируется рис. *1,b)*.

Относительная частота попадания точки в четверть круга $f = k/(2\cdot N) = 2\cdot k/n$.

В общем случае точка (x_{2i-1}, x_{2i}) всегда попадает внутрь единичного квадрата. Тогда теоретическая вероятность попадания этой точки в четверть круга

$$p_k = S_{1/4круга}/S_{квадрата} = (\pi\cdot r^2/4)/(1\cdot 1) = \pi/4. \quad (5.5) \tag{3.5}$$

Если числа последовательности равномерны, то в силу закона больших чисел теории вероятностей при больших N относительная частота

$$2\cdot k/n \to \pi/4. \tag{3.6}$$

Проверка стохастичности

Проверка стохастичности последовательностей псевдослучайных чисел чаще всего проводится *методом комбинаций* и *методом серий*.

Сущность метода комбинаций сводится к определению закона распределения длин участков между единицами (нулями) в n-разрядном двоичном числе x_i.

Теоретически закон появления j единиц в l разрядах двоичного числа x_i описывается (исходя из независимости двоичных разрядов) биномиальным законом распределения

$$p(j,l) = C^j_l\cdot p^l(1), \tag{3.7}$$

где $p(j,l)$ – вероятность появления j единиц в l разрядах числа x_i;

$p(1) = p(0) = 0,5$ – вероятность появления единицы (нуля) в любом разряде числа x_i.

$C^j_l = l!/(j!\cdot(l-j)!)$ – число сочетаний из l по j.

Тогда теоретическое ожидаемое число появления случайных чисел x_i с j единицами в проверяемых l разрядах будет равно

$$n_j = N\cdot C^j_l\cdot p^l(1). \tag{3.8}$$

Здесь N – количество опытов.

После нахождения теоретических и экспериментальных вероятностей $p(j,l)$ или чисел n_j гипотеза о стохастичности проверяется с помощью критериев согласия.

При анализе стохастичности методом серий последовательность разбивается на элементы первого и второго рода (a и b)

$$x_i = \begin{cases} a, & если \quad x_i < p, \\ b & в \quad противном \quad случае. \end{cases} \tag{3.9}$$

где $0 < p < 1$.

Серией называется отрезок последовательности $\{x_i\}$, состоящий из элементов одного и того же рода, следующих друг за другом. Число элементов (a или b) в отрезке называется длиной серии (l или j). После разбиения получают последовательность

$$\dots aabbaaabaabbbab \dots$$

Т.к. случайные числа a и b в данной последовательности независимы, то теоретическая вероятность появления серии длиной j в последовательности длиной l в N опытах определяется формулой Бернули

$$p(j,l) = C^j_l\cdot p^j\,(1-p)^{l-j}, \quad j = 1,\dots,l;\ l = 1,\dots,N. \tag{3.10}$$

Под опытом понимается генерация числа x_i и проверка условия $x_i < p$.

При экспериментальной проверке оценивается частота появления серий длиной j. В результате получаются теоретическая и экспериментальная зависимости $p(j,l)$, сходимость которых проверяется по известным критериям согласия.

Проверка независимости

Проверка независимости элементов псевдослучайных равномерно распределенных чисел проводится на основе вычисления корреляционного момента.

Случайные величины ξ и η называются независимыми, если закон распределения каждой из них не зависит от того, какое значение приняла другая. Следовательно, независимость элементов последовательности может быть проведена путем введения в рассмотрение последовательности $\{y_i\} = \{x_{i+\tau}\}$, где τ – величина сдвига последовательности.

При любом $\tau \neq 0$ для достаточно больших N с доверительной вероятностью β справедливо соотношение

$$|\overline{\rho}_{\xi\eta}(\tau)| \leq \beta\sqrt{1/N} \tag{3.11}$$

где $\rho_{\xi\eta}(\tau)$ – коэффициент корреляции

$$\overline{\rho}_{\xi\eta}(\tau) = \frac{\frac{1}{N-\tau}\sum_{i=1}^{n-\tau} x_i \cdot x_{i+\tau} - \frac{1}{(N-\tau)^2}\sum_{i=1}^{N-\tau} x_i \cdot \sum_{i=1}^{N-\tau} x_{i+\tau}}{\sqrt{D[x_i]\cdot D[x_{i+\tau}]}},$$

з
десь

$$D[x_i] = \frac{1}{N-\tau}\sum_{i=1}^{N-\tau} x_i^2 - \frac{1}{(N-\tau)^2}\left(\sum_{i=1}^{N-\tau} x_i\right)^2$$

$$D[x_{i+\tau}] = \frac{1}{N-\tau}\sum_{i=1}^{N-\tau} x_{i+\tau}^2 - \frac{1}{(N-\tau)^2}\left(\sum_{i=1}^{N-\tau} x_{i+\tau}\right)^2$$

Если найденное значение $\rho_{\xi\eta}(\tau)$ находится в указанных пределах, то с вероятностью β можно утверждать, что последовательность чисел $\{\boldsymbol{x_i}\}$ удовлетворяет гипотезе корреляционной независимости.

Важными характеристиками качества генератора псевдослучайных равномерно распределенных последовательностей является длина периода $\boldsymbol{p}$ и длина отрезка апериодичности $\boldsymbol{L}$. Параметр $\boldsymbol{p}$ определяет количество чисел повторяющихся групп в последовательности $\{\boldsymbol{x_i}\}$, а параметр $\boldsymbol{L}$ - максимальный отрезок, в пределах которого все числа $\boldsymbol{x_i}$ не повторяются.

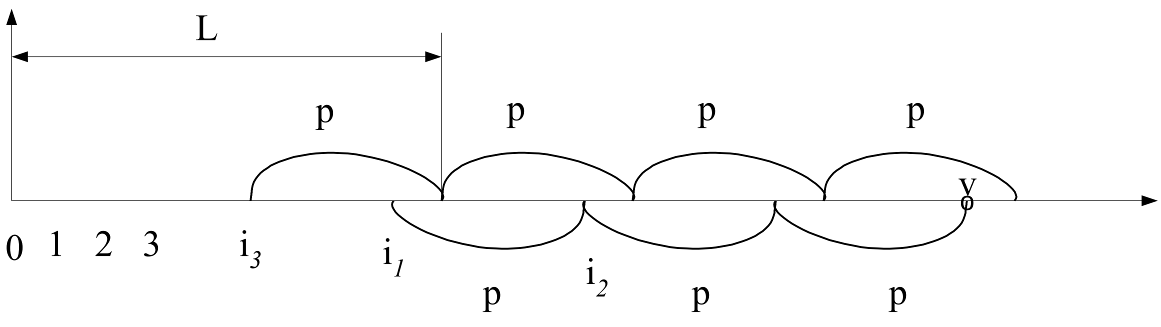

Рис.3.2. Схема для определения периода и длины отрезка апериодичности последовательности случайных чисел

Моделируется v чисел. Число x_v запоминается. Затем снова начиная с x_o моделируются числа и определяется период p, путем сравнения x_v с x_i : $p = i_2 - i_1$.

Для определения L фиксируется минимальный номер $i - i_3$, при котором $x_i = x_{p+i}$. и определяется длина отрезка апериодичности

$$L = i_3 + p. \tag{3.12}$$

Формирование случайных чисел с заданным законом распределения

Для получения последовательностей псевдослучайных чисел с заданным законом распределения чаще всего используют последовательности случайных чисел, равномерно распределенных в интервале *[0,1]*.

Пусть для величины z, имеющей заданное распределение, известна функция плотности распределения *f(z)*. Тогда, используя свойство функции распределения принимать значение от *0* до *1* при изменении аргумента от -∞ до +∞, каждому значению x_i случайной величины *X*, равномерно распределенной в интервале *[0;1]*, ставим в однозначное соответствие значение z_i, величины *Z*, т.е.

$$x_i = \int_{-\infty}^{z_i} f(z)dz \tag{3.13}$$

Формирование последовательности случайных чисел, равномерно распределенных в интервале [a,b] с функцией плотности распределения

$$f(z) = \begin{cases} \dfrac{1}{b-a}, & при \quad a \le z \le b \\ 0, & при \quad z < a; \quad z > b \end{cases} \tag{3.14}$$

Для формирования равномерно распределенных псевдослучайных чисел в интервале *[a,b]* достаточно случайное число x_i из интервала *[0;1]* привести к интервалу *[a,b]* и сдвинуть на величину *a:*

$$z_i = (b-a)\cdot x_i + a. \tag{3.15}$$

При этом теоретическое значение математического ожидания последовательности случайных равномерно распределенных чисел

$$m = \frac{a+b}{2} \tag{3.16}$$

$$D = \frac{(b-a)^2}{12}. \tag{3.17}$$

Формирование последовательности случайных чисел с показательным распределением.

Показательное распределение случайной величины имеет следующую функцию плотности распределения

$$f(z) = \lambda \cdot e^{-\lambda \cdot z} \tag{3.18}$$

где $z > 0$.

Подставим выражение (5.18) в (5.13), тогда

$$x_i = \lambda \cdot \int_0^{z_i} e^{-\lambda \cdot z} \cdot dz \tag{3.19}$$

$$x_i = 1 - e^{-\lambda \cdot z_i} \tag{3.20}$$

$$z_i = -\frac{1}{\lambda} \ln(1 - x_i)\,. \tag{3.21}$$

Так как $(1 - x_i)$ случайное равномерно распределенное в интервале *[0;1]* число, то

$$z_i = -\frac{1}{\lambda} \ln x_i\,. \tag{3.22}$$

Теоретическое значение математического ожидания последовательности случайных чисел с показательным распределением

$$m = \frac{1}{\lambda}, \tag{3.23}$$

а дисперсия

$$D = \frac{1}{\lambda^{2}} \tag{3.24}$$

Формирование последовательности случайных чисел, распределенных по нормальному закону с параметрами m и σ.

Функция плотности распределения последовательности случайных чисел, распределенных по нормальному закону имеет вид

$$f(z') = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot e^{-\frac{(x-m)^2}{2 \cdot \sigma^2}} \tag{3.25}$$

В силу центральной предельной теоремы случайная величина

$$z = \sum_{1}^{N} x_i \tag{3.26}$$

при достаточно большом N будет иметь распределение, близкое к нормальному.

Если x_i некоррелированные величины, то

$$m[z] = \sum_{1}^{N} m_{xi} = N \frac{a+b}{2}, \tag{3.27}$$

$$D[z] = \sum_{1}^{N} D_{xi} = N \frac{(a+b)^2}{12} \tag{3.28}$$

Используя последние выражения для заданного N можно определить границы ***[a; b]*** такие, чтобы Z имела заданные значения параметров m и σ, решив систему уравнений:

$$m^* = \frac{N \cdot (a+b)}{2} \quad ; \quad \sigma^* = \frac{(b-a) \cdot \sqrt{N}}{2 \cdot \sqrt{3}} \tag{3.29}$$

откуда:

$$a = \frac{m^* - \sigma^* \sqrt{3N}}{N}; \qquad b = \frac{m^* + \sigma^* \sqrt{3N}}{N}. \tag{3.30}$$

Для получения псевдослучайных чисел, равномерно распределенной на интервале ***[a; b]*** можно использовать преобразование (3.15).

Формирование последовательности случайных чисел, подчиняющихся распределению Пуассона (Приближенный метод).

Во многих задачах (анализ вызовов на АТС, излучение электронов из накаленного катода, анализ микробов в воздухе и т,д.) приходится иметь дело со случайными величинами, распределенными по своеобразному закону, который называется законом Пуассона.

Случайная величина *X* распределена по закону Пуассона, если вероятность того, что она примет определенное значение m из ряда чисел

0, 1, 2, ..., m, ...,

выражается формулой

$$P_m = \frac{\alpha^m}{m!} \cdot e^{-\alpha}, \tag{3.31}$$

где α - некоторая положительная величина, называемая параметром закона Пуассона (может быть и нецелой).

Ряд распределения случайной величины по закону Пуассона, имеет вид, табл. 3.1

Таблица 3.1 – Распределение случайной величины закону Пуассона

x_m	0	1	2	...	m	...
P_m	$e^{-\alpha}$	$\frac{\alpha}{1!}e^{-\alpha}$	$\frac{\alpha^2}{2!}e^{-\alpha}$	...	$\frac{\alpha^m}{m!}e^{-\alpha}$	...

На рис.3.3 приведены многоугольники распределения случайной величины ***X***, распределенной по закону Пуассона, соответствующие различным значениям параметра α.

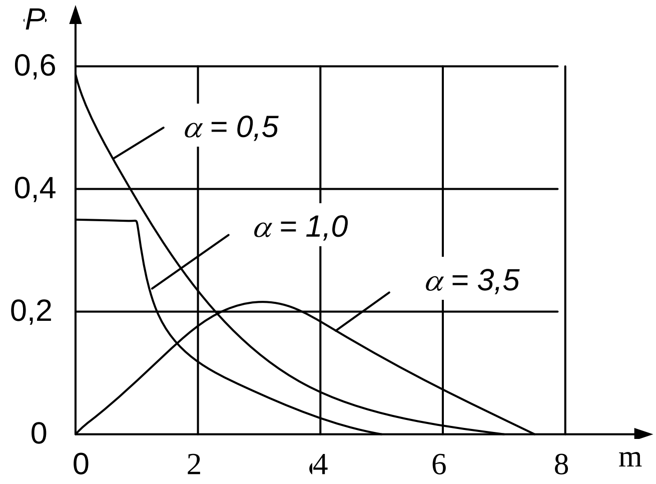

Рис. 3.3

Для формирования такой последовательности может быть использована следующая процедура. Берется произведение равномерно распределенных в интервале ***[0;1]*** чисел x_j. Причем число сомножителей ***m*** выбирается таким, чтобы выполнялось неравенство

$$\prod_{j=1}^{m} x_j < e^{-\lambda} \tag{3.32}$$

где m - параметр моделируемого распределения Пуассона.

Если условие (5.32) выполняется, то можно утверждать, что число $z_j = m - 1$ будет представлять случайное число, принадлежащее последовательности чисел, распределенных по закону Пуассона с параметром m.

Моделирование случайных воздействий

При моделировании систем существенное внимание уделяется учету случайных факторов и воздействий, называемых случайными объектами. При этом используются случайные события, случайные векторы и случайные процессы.

Простейшими случайными объектами при статистическом моделировании систем являются случайные события. Рассмотрим особенности их моделирования.

Пусть имеется последовательность случайных чисел $\{x_i\}$ случайной величины ξ, равномерно распределенной в интервале ***(0,1)***. Необходимо реализовать случайное событие ***A***, наступающее с заданной вероятностью ***p***. Определим событие ***A*** как состоящее в том, что выбранное значение x_i случайной величины ξ удовлетворяет неравенству

$$x_i \le p. \tag{3.33}$$

Процедура моделирования в этом случае состоит в выборе значений x_i и сравнении их с p. Если условие (5.33) выполняется, то исходом испытания является событие *A*. Следовательно, моделирование случайных событий сводится к моделированию дискретных случайных величин.

Рассмотрим случайную величину ξ, называемую индикатором события A, которая равна 1 при наступлении события A и 0 при наступлении противоположного события $\bar{A}$.

Распределение ξ задается таблицей

$$\begin{pmatrix} 1 & 0 \\ p & 1-p \end{pmatrix}$$

Для осуществления каждого испытания надо найти случайное число $\boldsymbol{x_i}$ и проверить неравенство (3.33).

Моделирование случайных процессов

При моделировании некоторых систем, например динамических, входные воздействия или воздействия внешней среды могут быть представлены в виде случайных процессов с заданными корреляционной функцией либо спектральной плотностью.

В основу алгоритмов формирования случайных процессов может быть положено линейное преобразование стационарной последовательности $x[n]$ независимых случайных чисел, имеющих нормальное распределение. Рассмотрим некоторые алгоритмы формирования реализаций дискретных случайных процессов.

Алгоритм формирования реализаций дискретного случайного процесса с экспоненциальной корреляционной функцией вида

$$R_q(\tau) = \sigma_E e^{-\alpha E|\tau|}$$

где σ_E – среднее квадратичное отклонение случайной величины;

α_E – коэффициент корреляционной связи;

τ - аргумент ($\tau = t_1 - t$)

может представлена в виде

$$q[n] = a_o \cdot x_N[n] + b_1 \cdot q[n-1], \; n=1, 2, \ldots$$

где $a_0 = \sigma_E \sqrt{1-b_1^2}$

$$b_1 = e^{-\alpha Eh}.$$

Здесь x_N – нормально распределенная случайная величина с параметрами $m_x = 0$, $\alpha_x = 1$; h – шаг дискретизации независимой переменной t.

Алгоритм формирования реализаций случайного процесса с экспоненциально-косинусной корреляционной функцией вида

$$R_q(\tau) = \sigma_E^2 e^{-\alpha E|\tau|} Cos(\beta_E \cdot \tau)$$

где α_E, β_E – коэффициенты корреляционной связи

имеет вид

$$q[n]=a_o \cdot x_N[n]+a_1 \cdot q[n-1]+b_1 \cdot q[n-1]+b_2 \cdot q[n-2], \; n=0, 1, 2, \ldots$$

где $a_o = \sigma_E \cdot b_o$; $a_1 = \sigma_E / b_o$.

$$b_o = \sqrt{\frac{c_1 + \sqrt{c_1^2 - 4c_0^2}}{2}}$$

$$b_1 = 2e^{-\alpha Eh}.Cos(\beta_E \cdot h) \qquad b_2 = e^{-2\alpha Eh}$$

$$c_o = e^{-\alpha Eh}(e^{-2\alpha Eh}-1)Cos(\beta_E \cdot h)$$

$$c_1 = 1 - e^{-4\alpha Eh}$$

3.2.Регрессионный и корреляционный анализ

Обработка результатов эксперимента

Регрессионный анализ дает возможность построить модель наилучшим образом соответствующую набору данных, полученных в ходе машинного (расчетного) или натурного эксперимента.

Рассмотрим основные положения линейного регрессионного анализа.

В результате вероятностного эксперимента фиксируются некоторые случайные величины, например X и Y, которые являются функциями одного и того же аргумента. Это обстоятельство создает между ними некоторую связь, которую принято называть статистической зависимостью.

Статистической зависимостью между случайными величинами X и Y, соответствующими двухмерной случайной величине, (X,Y) называется правило F, которое каждому числу x из числового множества X ставит в соответствие условное распределение составляющей Y.

$$\rightarrow F(y|x).$$

Зависимости между случайными величинами изучают с точки зрения предсказания (прогноза) значений одной случайной величины по наблюдаемым значениям другой.

В практических приложениях при исследовании зависимости между СВ X и Y ограничиваются исследованием зависимости между X и условным математическим ожиданием

$$M(Y|X = x) = \int_{-\infty}^{\infty} yf(y \mid x)dy.$$

Зависимости такого рода называются *регрессионными*. Задача нахождения наилучшей оценки уравнения регрессии Y на X практически совпадает с задачей сглаживания экспериментальных зависимостей (задача подбора эмпирических формул по экспериментальным данным). При решении такой задачи считается, что между математическими ожиданиями M(X) и M(Y) переменных X и Y существует функциональная зависимость вида $f(x, a, b, \ldots)$, причем общий вид ее может быть известен из физической сущности задачи.

Методы решения задачи подбора эмпирических формул и оценки уравнения регрессии одинаковы. Однако круг задач, рассматриваемых при нахождении эмпирических функций регрессии более широк.

Для решения задачи нахождения эмпирических уравнений регрессии $\bar{y} = f(x,a,b,\ldots)$ применяется метод наименьших квадратов.

Рассмотрим методику построения математической модели объекта по заданным экспериментальным данным используя эмпирическую зависимость вида

$$y=a \cdot x^b. \tag{1}$$

В модели (1) неизвестными являются коэффициент a и показатель степени b. Для вычисления указанных неизвестных сначала прологарифмируем выражение (1)

$$ln(y) = ln(a) + b \cdot ln(x) \tag{2}$$

Коэффициент a и показатель степени b найдем с помощью метода наименьших квадратов при условии

$$F =\sum(e_i^2) = \sum(y_i - y_{ri})^2 = min. \tag{3}$$

Здесь y_{ri} - расчетные значения y, полученные по составленной модели; e_i^2 – квадратичная ошибка.

Подставив (2) в (3) получим

$$F = \sum[ln(y_i) – ln(a) - b \cdot ln(x_i)]^2. \tag{4}$$

Введем обозначения $c=ln(a)$. Тогда получим

$$F = \sum[ln(y_i) – c - b \cdot ln(x_i)]^2. \tag{5}$$

Найдем частные производные от F по c и по b и приравняем их нулю

$$\partial F/\partial c = -2\cdot\sum[\ln(y_i) - c - b\cdot\ln(x_i)] = 0;$$

$$\partial F/\partial b = -2\cdot\sum\{[\ln(y_i) - c - b\cdot\ln(x_i)]\cdot\ln(x_i)\} = 0,$$

или

$$\sum\ln(y_i) - n\cdot c - b\cdot\ln(x_i) = 0;$$

$$\sum\ln(y_i)\cdot\ln(x_i) - c\cdot\sum\ln(x_i) - b\cdot\sum[\ln(x_i)]^2 = 0,$$

откуда

$$n\cdot c - b\cdot\sum\ln(x_i) = \sum\ln(y_i); \qquad (6)$$

$$c\cdot\sum\ln(x_i) - b\cdot\sum[\ln(x_i)]^2 = \sum\ln(y_i)\cdot\ln(x_i).$$

Решив полученную систему уравнений получим формулы для вычисления коэффициентов b и c.

Коэффициент a вычисляется по формуле

$$a = \exp(c).$$

Пример 1.

Пусть требуется построить математическую модель объекта по заданным экспериментальным данным, приведенным в табл.3.1. Вид эмпирической зависимости $\boldsymbol{y = a\cdot x^b}$.

Таблица 3.1 - Исходные (экспериментальные) данные

x	y					$y_{ср}$
1	-125,1	-124,7	-125,0	-124,9	-125,2	-124,98
2	-203,2	-202,8	-203,1	-203,0	-203,3	-203,08
3	-269,8	-269,4	-269,7	-269,6	-269,9	-269,68
4	-330,0	-329,6	-329,9	-329,8	-330,1	-329,88
5	-385,7	-385,3	-385,6	-385,5	-385,8	-385,58
6	-438,2	-437,8	-438,1	-438,0	-438,3	-438,08
7	-488,2	-487,8	-488,1	-488,0	-488,3	-488,08
8	-536,0	-535,6	-535,9	-535,8	-536,1	-535,88
9	-582,0	-581,6	-581,9	-581,8	-582,1	-581,88
10	-626,6	-626,2	-626,5	-626,4	-626,7	-626,48

Для решения задачи воспользуемся программой Excel, в частности ее функциями и мастером диаграмм.

Найдем средние значения для y, которое запишем в столбец $y_{ср}$ в табл.1

Построим график зависимости $y_{ср} = f(x)$, Рис.3.4.

Рис. 3.4

Построим таблицу 3.2 с исходными данными для решения системы нормальных уравнений (6). При этом в столбец Y запишем абсолютные значения Yср, т.к., как известно, логарифм отрицательного числа на существует.

В строке "Сумма" запишем вычисленные суммы соответствующих столбцов.

Таблица 3.2. - Данные для решения системы уравнений (6)

	X	*Y*	*Ln(x)*	*Ln(y)*	*[Ln(x)]^2*	*Ln(x)·Ln(y)*	*Yr*
	1	124,980	0,000	4,828	0,000	0,000	124,989
	2	203,080	0,693	5,314	0,480	3,683	203,048
	3	269,680	1,099	5,597	1,207	6,149	269,691
	4	329,880	1,386	5,799	1,922	8,039	329,857
	5	385,580	1,609	5,955	2,590	9,584	385,624
	6	438,080	1,792	6,082	3,210	10,898	438,119
	7	488,080	1,946	6,190	3,787	12,046	488,041
	8	535,880	2,079	6,284	4,324	13,067	535,861
	9	581,880	2,197	6,366	4,828	13,988	581,915
	10	626,480	2,303	6,440	5,302	14,829	626,456
Сумма	55	3983,600	15,104	58,856	27,650	92,283	3983,602

Подставим исходные данные из строки "Сум" табл.2 в систему уравнений (6). Получим:

Таблица 3.3 - Матрица коэффициентов при неизвестных

10	15,104
15,104	27,650

Таблица 3.4 - Правые части

58,856
92,283

Используя функцию МОБР найдем обратную матрицу от матрицы табл.5.

Таблица 3.5 - Обратная матрица

0,572	-0,312
-0,312	0,207

Умножим обратную матрицу на столбец свободных членов, табл.4. Получим следующее решение системы (6)

c 4,828

b 0,700

Значение коэффициента ***a*** найдем по формуле

$a = Exp(c)$

Искомые коэффициент а и показатель степени *b*

a 124,989

b 0,700

Таким образом, получена математическая модель:

$$y= 124,989 \cdot X^{0,7} \quad (7)$$

Рассчитанные значения с помощью модели (7) приведены в столбце $\boldsymbol{y_r}$ таблицы 2. Заменим знаки в $\boldsymbol{y_r}$ на обратные, т.к. исходные значения были отрицательные. Результаты запишем в табл.6. Как видно из таблицы, степень совпадения расчетных и экспериментальных данных высокая, т.к. погрешность не превышает 0,0114%.

Таблица 3.6

X	*Y*	*Yr*	*Погрешность*
1	-124,980	-124,989	0,0072%
2	-203,080	-203,048	0,0158%
3	-269,680	-269,691	0,0041%
4	-329,880	-329,857	0,0070%
5	-385,580	-385,624	0,0114%
6	-438,080	-438,119	0,0089%
7	-488,080	-488,041	0,0080%
8	-535,880	-535,861	0,0035%
9	-581,880	-581,915	0,0060%
10	-626,480	-626,456	0,0038%

Пример 2. Найти коэффициенты регрессии используя метод наименьших квадратов, если заданы результаты эксперимента в следующем виде

Таблица 3.7

x_i	1	2	3	4
y_i	2	4	5	7

Вид эмпирической зависимости задан

$y = b_0 + b_1 \cdot x$

Решение.

Коэффициент b_0 и b_1 найдем с помощью метода наименьших квадратов при условии

$F = \sum(e_i^2) = \sum(y_i - y_{ri})^2 = \sum(y_i - b_0 - b_1 \cdot x_i)^2 = min.$ (7)

Найдем частные производные $\partial F/\partial b_0$ и $\partial F/\partial b_1$ и приравняем их к нулю

$\partial F/\partial b_0 = 2 \cdot \sum(y_i - b_0 - b_1 \cdot x_i) = 0;$

$\partial F/\partial b_1 = 2 \cdot \sum(y_i - b_0 - b_1 \cdot x_i) \cdot x_i = 0;$

Раскрыв знак суммы и произведя преобразования получим систему двух линейных уравнений, называемых нормальными уравнениями:

$$n \cdot b_0 + b_1 \cdot \sum x_i = \sum y_i \qquad (8)$$

$$b_0 \cdot \sum x_i + b_1 \cdot \sum x^2_i = \sum x_i y_i$$

Построим на рис.3.5 уравнение регрессии Y на X по результатам опытов, приведенным в табл. 3.7

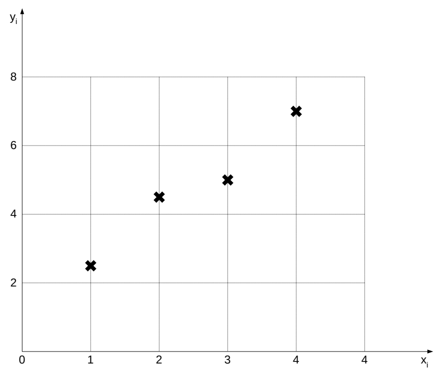

Рис.3.5

Анализируя расположения точек, заключаем, они группируются вокруг прямой линии. Поэтому построим уравнение регрессии линейного вида. Для этого сначала вычислим необходимые суммы, табл.2.

Таблица 3.8

x_i	y_i	$x_i\, y_i$	x^2_i	y^2_i
1	2,25	2,25	1	5,06
2	4,25	8,50	4	18,06
3	5,00	15,00	9	25,00
4	7,00	28,00	16	49,00
10	*18,50*	*53,75*	*30*	*97,13*

Подставим найденные суммы в систему (8), тогда получим

$$4 \cdot b_0 + 10 \cdot b_1 = 18{,}5$$

$$10 \cdot b_0 + 30 \cdot b_1 = 53{,}75$$

Решая эту систему уравнений получим

$\cdot b_0 = 0.875$

$\cdot b_1 = 1.5$

Тогда уравнение регрессии получит вид
$y = 0{,}875 = 1{,}5 \cdot x$

Построение полиномиальных регрессионных моделей систем

Для проведения регрессионного анализа необходимо выполнение следующих условий:

1) Результаты эксперимента $y_1, y_2, \ldots, y_n$ представляют собой независимые, нормально распределенные случайные величины;

2) Независимые переменные $x_1, x_2, \ldots, x_n$ измеряются с пренебрежительно малой ошибкой по сравнению с ошибкой в определении y;

3) При проведении эксперимента с объемом выборки N при условии, что каждый опыт повторен m раз, выборочные дисперсии $s_1^2, s_2^2, \ldots, s_n^2$ должны быть однородны. Нормальное распределение случайной величины y может быть проверено при помощи известных статистических критериев (например критерия χ^2).

После того, как сделан вывод о возможности применения к экспериментальным данным метода регрессионного анализа, необходимо выбрать класс функциональных зависимостей, среди которых ищется требуемая статистическая модель процесса (функция отклика) вида

$y = \varphi(x_1, x_2 \ldots, x_k).$

Чаще всего для этих целей выбирают следующие степенные полиномы:

$$\hat{y} = b_0 + \sum_{j=1}^{k} b_j \cdot x_j + \sum_{u,j=1}^{k} b_{uj} \cdot x_u \cdot x_j \cdot + \cdot \sum_{j=1}^{k} b_{jj} \cdot x_j^2 + \ldots, \qquad (7)$$

Поскольку регрессионные модели вида (7) являются нелинейными, то определение коэффициентов b_0, b_j, b_{uj}, b_{jj} сопряжено со значительными вычислительными трудностями. Поэтому, с целью упрощения вычислений вводится фиктивная переменная $x_0 = 1$, а члены второго порядка заменяются линейными соотношениями следующего вида:

$x_1^2 = x_{k+1}, \quad x_2^2 = x_{k+2}, \ldots x_k^2 = x_{2k},$

$x_1 \cdot x_2 = x_{2k+1}, \quad x_2 \cdot x_3 = x_{2k+2}, \ldots x_{k-1} \cdot x_k = x_{2k+(k-1)},$

Аналогичным образом линейными членами можно заменить члены любого порядка и свести исходную регрессионную модель (7) к линейной множественной модели вида

$$y = b_0 \cdot x_0 + b_1 \cdot x_1 + \ldots + b_k \cdot x_k \qquad (8)$$

Введением вектора-строки $X = \{x_0, x_1 , ..., x_k\}$ и вектора-столбца коэффициентов $B^T = \{b_0, b_1 , ..., b_k\}$ уравнения (8) запишется в виде

$$Y = X \cdot B^T$$

Коэффициенты уравнения (8) определяются методом наименьших квадратов из условия:

$$\Phi = [Y\text{-}Y_r]\cdot[Y\text{-}Y_r]^T = \sum(y_i - y_i)^2 = min, \tag{9}$$

где N- объем выборки из всей совокупности значений исследуемых параметров;

$Y^T = [y_1, y_1 ... y_1]$ - вектор наблюдений.

Приравнивая нулю частные производные от этой квадратичной формы, взятые по переменным $b_0, b_1 , ..., b_k$ получим систему нормальных уравнений

$$X^T \cdot Y = X^T \cdot X \cdot B, \tag{10}$$

где X - матрица независимых переменных.

Вектор-столбец искомых коэффициентов регрессии B может быть определен непосредственно решением системы уравнений (10)

$$B = (X^T \cdot X)^{-1} X^T \cdot Y. \tag{11}$$

Обычно систему (10) решают каким-либо численным методом, поскольку это более предпочтительно с точки зрения затрат машинного времени и вычислительных погрешностей, чем реализация соотношения (11).

После определения коэффициентов регрессии b_j $(j=1,2, ..., k)$ необходимо произвести статистический анализ уравнения регрессии, состоящий в оценке значимости полученных коэффициентов регрессии и проверке адекватности полученной модели исходным данным. Однако, как показано в [2], оценка значимости коэффициентов регрессионной модели не имеет смысла в случае коррелированности коэффициентов регрессии, что имеет место в данном случае, поскольку каждый коэффициент учитывает вклад в значение выхода y нескольких входных переменных (факторов). Поэтому проверку значимости стоит производить только при обработке результатов экспериментов, построенных по определенному плану, обеспечивающему некоррелированность коэффициентов регрессии.

Адекватность представления результатов эксперимента полиномом заданной степени проверяется по критерию Фишера. Для этого составляется отношение (12)

$$F = \frac{S^2_{ост}}{S^2_{воспр}}, \tag{12}$$

где $S^2_{воспр} = \frac{\sum_{i=1}^{N} S_i^2}{N}$ - дисперсия воспроизводимости;

$S^2_{ост} = \frac{\sum_{i=1}^{N} (\tilde{Y}_i - \bar{Y}_i)^2}{N - l}$ - остаточная дисперсия.

Здесь $\boldsymbol{l}$ - число коэффициентов в уравнении регрессии (8)

Приведем программу на VBA вычисления коэффициентов модели. Она предназначена для решения задачи регрессионного анализа при следующих условиях проведения эксперимента:

1) количество факторов равно 2 (x_1 и x_2), которые принимают целые положительные значения от 1 до 5 включительно;

2) эксперимент проводится при всех возможных комбинациях значений x_1 и x_2.

3) преполагается, что статистическая модель процесса может быть описана полиномом не выше 4-го порядка;

4) значения коэффициентов регрессии b_j $(j=0,1, ..., n)$ определяются на основе решения системы уравнений 10.

В программе использованы следующие обозначения:

n – количество коэффициентов;

sp – степень полинома;

y – массив, в котором располагается вектор наблюдений выходной переменной *y* во всех экспериментальных точках. Размерность вектора N = 25 (5х5);

x – (матрица независимых переменных) - двухмерный массив, в котором хранятся значения x_0, x_1,x_2 и всевозможные комбинации x_1 и x_2, определяемые в соответствии со степенью *sp* аппроксимирующего полинома.

xt – матрица, полученная транспонированием матрицы *x*.

xy – вектор-столбец (матрица), являющийся произведением *XT* и *Y*.

korm – матрица, полученная в результате перемножения *XT* и *X*.

kregr – матрицы коэффициентов регрессии b_j $(j=0,1,\ ...,\ n)$;

ypr – вектор предсказанных значений выхода с помощью модели процесса.

Программа имеет следующий вид:

```
Public Sub regm()
Dim x(25, 15) As Variant, xt(15, 25) As Variant, xy(15) As Variant
Dim korm(15, 15) As Variant, kregr(15) As Variant, y(25) As Variant
Dim ypr(25) As Variant

sp = 1
With Worksheets(1)
For i = 1 To 5
y(i) = .Cells(i + 1, 2)
y(i + 5) = .Cells(i + 1, 3)
y(i + 10) = .Cells(i + 1, 4)
y(i + 15) = .Cells(i + 1, 5)
y(i + 20) = .Cells(i + 1, 6)
Next i
End With
    For i = 1 To 25
    x(i, 1) = 1
Worksheets(3).Cells(i, 17) = y(i)
Worksheets(3).Cells(i, 1) = x(i, 1)
    Next i
 k = 0
 For i = 1 To 5
  For j = 1 To 5
   k = k + 1
   x(k, 2) = i
   x(k, 3) = j
Worksheets(3).Cells(k, 2) = x(k, 2)
Worksheets(3).Cells(k, 3) = x(k, 3)
  Next j
 Next i
n = 3
 If sp >= 2 Then
            k = 0
            For i = 1 To 5
             For j = 1 To 5
               k = k + 1
               x(k, 4) = i * i
               x(k, 5) = j * j
               x(k, 6) = i * j
Worksheets(3).Cells(k, 4) = x(k, 4)
Worksheets(3).Cells(k, 5) = x(k, 5)
Worksheets(3).Cells(k, 6) = x(k, 6)
             Next j
            Next i
            n = 6
           If sp >= 3 Then
```

```
                k = 0
                For i = 1 To 5
                 For j = 1 To 5
                  k = k + 1
                  x(k, 7) = i * i * i
                  x(k, 8) = j * j * j
                  x(k, 9) = i * i * j
                  x(k, 10) = i * j * j
Worksheets(3).Cells(k, 7) = x(k, 7)
Worksheets(3).Cells(k, 8) = x(k, 8)
Worksheets(3).Cells(k, 9) = x(k, 9)
Worksheets(3).Cells(k, 10) = x(k, 10)
                 Next j
                Next i
                n = 10
            If sp >= 4 Then
                     k = 0
                     For i = 1 To 5
                      For j = 1 To 5
                      k = k + 1
                      x(k, 11) = i * i * i * i
                      x(k, 12) = j * j * j * j
                      x(k, 13) = i * i * i * j
                      x(k, 14) = j * j * j * i
                      x(k, 15) = i * i * j * j
Worksheets(3).Cells(k, 11) = x(k, 11)
Worksheets(3).Cells(k, 12) = x(k, 12)
Worksheets(3).Cells(k, 13) = x(k, 13)
Worksheets(3).Cells(k, 14) = x(k, 14)
Worksheets(3).Cells(k, 15) = x(k, 15)
                      Next j
                     Next i
                     n = 15
                    Else
                   End If
                 Else
              End If
           Else
         End If

For i = 1 To n
For j = 1 To 25
xt(i, j) = x(j, i)
Next j
Next i

With Worksheets(2)
For i = 1 To n
For j = 1 To n
s = 0
For k = 1 To 25
s = s + xt(i, k) * x(k, j)
Next k
korm(i, j) = s
.Cells(i, j) = s
Next j
Next i
End With

For i = 1 To n
s = 0
For j = 1 To 25
s = s + xt(i, j) * y(j)
```

```
Next j
xy(i) = s
Worksheets(2).Cells(i, n + 2) = s
Next i

Select Case n
Case 3: Range("A21:C23").Select
        Selection.FormulaArray = "=MINVERSE(R[-20]C:R[-18]C[2])"
Case 6: Range("A21:F26").Select
        Selection.FormulaArray = "=MINVERSE(R[-20]C:R[-15]C[5])"
Case 10: Range("A21:J30").Select
        Selection.FormulaArray = "=MINVERSE(R[-20]C:R[-11]C[9])"
Case 15: Range("A21:O35").Select
        Selection.FormulaArray = "=MINVERSE(R[-20]C:R[-6]C[14])"
End Select

For i = 1 To n
kregr(i) = 0
For j = 1 To n
kregr(i) = kregr(i) + Worksheets(2).Cells(i + 20, j) * xy(j)
Next j
Worksheets(2).Cells(i + 20, n + 2) = kregr(i)
Next i

s1 = 0
For i = 1 To 25
s = 0
For j = 1 To n
s = s + x(i, j) * kregr(j)
ypr(i) = s
Next j
Worksheets(2).Cells(i, n + 3) = ypr(i)
s1 = s1 + (y(i) - s) ^ 2
Next i
dos = s1 / (25 - n)
Worksheets(2).Cells(17, 1) = "Остаточная дисперсия"
Worksheets(2).Cells(17, 4) = dos
End Sub
```

Приведенная программа использует три листа MS Excel.

4. Планирование, обработка и анализ результатов моделирования

Общие вопросы планирования эксперимента. Цели и задачи планирования.

Цель любого эксперимента (натурного или расчетного) заключается в получении информации, т.е. опытных данных, об исследуемой системе. Опытные данные могут накапливаться либо путем пассивного наблюдения, либо с помощью активного эксперимента.

В случае активного эксперимента эти данные получают по заранее спланированной программе – плану эксперимента

В общем случае исследуемая система (объект) характеризуется:

1.Входными параметрами x_i, $i=1,...,n_x$;

2.Выходными параметрами y_j, $j=1,...,n_y$;

3.Внутренними параметрами h_k, $k=1,...,n_h$;

4.Параметры возмущающих воздействий v_l, $l=1,..., n_v$.

Параметры возмущающих воздействий оказывают влияние на систему и проявляют себя как случайные величины или случайные функции.

Одной из основных задач активного эксперимента является выявление взаимосвязей между входными (x_i) и выходными (y_i) параметрами системы и представление их в виде математической модели вида

$$y_i = f_j(x_1, x_2, ..., x_n).$$

Входные параметры x_i называют факторами. Каждый фактор имеет область определения (область допустимых значений). Комбинация факторов рассматривается как факторное пространство. Область возможных комбинаций факторов определяется планом эксперимента.

Факторы должны удовлетворять следующим требованиям:

1.Управляемость (т.е. возможность установки требуемых значений);

2.Однозначность (трудно управлять фактором, который в свою очередь является функцией других факторов);

3.Независимость, т.е. возможность установки любого уровня фактора вне зависимости от уровней других факторов.

Математическая обработка результатов эксперимента

Математическая обработка результатов спланированного эксперимента позволяет получить уравнение регрессии (математическую модель), которое чаще представляется в виде полинома следующего вида:

$$y = a_0 + \sum_{i=1}^{n_1} a_i \cdot x_i + \sum_{i=1}^{n_2} a_{ij} \cdot x_i \cdot x_j + \sum_{i=1}^{n_3} a_{ii} \cdot x_i^2 + ... \tag{4.1}$$

где a_o, a_i , a_{ij}, a_{ii} – коэффициенты регрессии.

Для правильного выбора степени полинома необходимо учитывать требования простоты и адекватности.

Модели низких степеней являются более простыми.

Адекватность модели оценивается путем сопоставления результатов, полученных с помощью модели, и опытных данных. Такая проверка выполняется с помощью соответствующих критериев, например с помощью критерия Фишера, вычисляемого по формуле

$$F = \frac{S_{adk}^2}{S_y^2} \tag{4.2}$$

где $S^2_{адк}$ - дисперсия адекватности;

S^2_y - дисперсия, характеризующая ошибку опыта.

Модель считается адекватной, если значение $\boldsymbol{F}$, полученное по формуле (4.2), не превышает соответствующего табличного значения при заданном уровне значимости. Уровень значимости представляет собой достаточно малое значение вероятности, соответствующее событиям, которые в данных условиях эксперимента можно считать практически невозможными. Обычно при решении инженерных задач уровень значимости принимается равным 0.05 (5%).

Полный факторный эксперимент

Полным факторным экспериментом (ПФЭ) называется такой эксперимент, при реализации которого определяется значение функции цели для всех возможных сочетаний уровней варьирования факторов.

Если имеется k факторов, каждый из которых может устанавливаться на q уровнях, то для осуществления ПФЭ необходимо поставить q^k опытов.

Наибольшее распространение получили эксперименты, в которых факторы варьируются на двух уровнях, т.е. эксперименты 2^k.

Планирование, проведение и обработка результатов ПФЭ состоит из следующих обязательных этапов:

1.Выбор факторов и кодирование их уровней;
2.Составление плана-матрицы эксперимента;
3.Реализация плана эксперимента;
4.Проверка воспроизводимости параллельных опытов;
5.Расчет коэффициентов регрессии линейной модели;
6.Проверка адекватности линейной модели;
7.Оценка значимости коэффициентов регрессии.

Планирование эксперимента начинается с выбора влияющих факторов и их кодирования.

После выбора факторов для каждого из них устанавливается основной уровень (r_o) (среднее, исходное значение) и интервал варьирования. Затем определяются:

- верхний уровень фактора ($r_в$) – путем прибавления интервала варьирования к основному уровню;
- нижний уровень фактора ($r_н$) – путем вычитания интервала варьирования из основного уровня.

Кодирование факторов выполняется в соответствии с формулами

$$\begin{cases} x_в = \dfrac{r_в - r_0}{\Delta r} = +1; \\ x_н = \dfrac{r_н - r_0}{\Delta r} = -1. \end{cases} \tag{4.3}$$

Здесь $r_в$, $r_н$ – соответственно верхний и нижний уровни натуральных факторов; Δr – интервал варьирования.

Результаты кодирования уровней факторов могут быть представлены в виде таблицы 4.1.

Таблица 4.1 – Кодирование факторов

Наименование уровня фактора	Обозначение	Значение
Основной	x_o	0
Верхний $(r_в - r_o)/\Delta r$	$x_в$	+1
Нижний $(r_н - r_o)/\Delta r$	$x_н$	-1

Интервал варьирования рекомендуется выбирать таким образом, чтобы он не превышал удвоенной средней квадратичной ошибки в определении данного фактора.

Кодирование факторов необходимо для перевода их натуральных значений в безразмерные величины. Такой перевод выполняется для того, чтобы иметь возможность построить стандартную ортогональную матрицу планирования эксперимента.

2.Составление план–матрицы эксперимента

План-матрица эксперимента представляет собой таблицу, состоящую из строк и столбцов. В столбцах таблицы содержатся кодированные значения факторов, а в строках – опыты в виде уровней факторов и их произведений. Составляется план-матрица эксперимента в строгом соответствии с принятой формой уравнения регрессии (4.1). Например, для уравнения регрессии вида

$$y = a_0 + a_1 \cdot x_1 + a_2 \cdot x_2 + a_{12} \cdot x_1 \cdot x_2 \qquad (4.4)$$

при варьировании двух факторов (x_1 и x_2) на двух уровнях (*+1 и –1*) план-матрица будет иметь следующий вид:

Таблица 4.2 – Расширенная план-матрица эксперимента

Номер опыта	x_1	x_2	$x_1 \cdot x_2$
1	*2*	*3*	*4*
1	-1	-1	+1
2	+1	-1	-1
3	-1	+1	-1
4	+1	+1	+1

В столбцах 2 и 3 табл. 4.2 записаны уровни факторов, в столбце 4 – их произведение. Матрицу типа табл. 4.2 называют расширенной матрицей планирования, т.к. в нее введен столбец (4) произведения уровней факторов ($\boldsymbol{x_1 \cdot x_2}$), позволяющий оценить коэффициент регрессии при взаимодействии факторов.

Для записи результатов эксперимента в матрицу планирования добавляются столбцы результатов, столбцы 5, 6 в табл. 4.3.

Таблица 4.3 – Расширенная матрица планирования со столбцами результатов эксперимента

Номер опыта	x_1	x_2	$x_1 \cdot x_2$	y_{n1}	y_{n2}	$y_n = (y_{n1} + y_{n2})/2$
1	*2*	*3*	*4*	*5*	*6*	*7*
1	-1	-1	+1			
2	+1	-1	-1			
3	-1	+1	-1			
4	+1	+1	+1			

При этом для обеспечения возможности проверки воспроизводимости каждый опыт эксперимента проводится дважды. В столбец 7 этой таблицы записываются вычисленные средние значения параллельных опытов.

3.Проверка воспроизводимости параллельных опытов при одинаковом их числе на каждом сочетании уровней факторов осуществляется по критерию Кохрена:

$$\sigma = \frac{S^2_{y\max}}{S^2_y} \le \sigma_t(0.05; k1; k2) \qquad (4.5)$$

где $S_y^2 = \dfrac{1}{m-1}\sum_{j=1}^{m}(y_{uj} - \bar{y}_u)^2$ – дисперсия воспроизводимости, характеризующая рассеивание результатов, т.е. ошибку опыта на *u*– м сочетании уровней факторов;

$\bar{y}_u = \dfrac{1}{m}\sum_{j=1}^{m} y_{ju}$ - среднее значение функции отклика на *u*-ой строке матрицы планирования;

$S^2_{y\max}$ – наибольшая из дисперсий S_y^2 в строках плана;

$\sigma_t(0,05;\ k1;\ k2)$- табличное значение критерия Кохрена при 5%-ом уровне значимости;

0.05 – уровень значимости (значение вероятности, отвечающее событиям, которые в данных условиях эксперимента можно считать практически невозможными);

$k1 = n$ – количество опытов в плане эксперимента;

$k2 = m - 1$ – число степеней свободы каждой оценки;

m – число параллельных опытов.

Процесс считается воспроизводимым, если выполняется неравенство (4.5). Если же проверка по критерию Кохрена не выполняется, то следует обратить внимание на опыт с максимальным значение S_y^2 и, возможно, повторить его. В случае воспроизводимости процесса переходят к расчету коэффициентов регрессии:

$$a_0 = \frac{1}{n}\sum_{u=1}^{n}\overline{y}_u; \tag{4.6}$$

$$a_i = \frac{1}{n}\sum_{u=1}^{n}x_{iu}\cdot\overline{y}_u; \tag{4.7}$$

$$a_{ij} = \frac{1}{n}\sum_{u=1}^{n}x_{iu}\cdot x_{ju}\cdot\overline{y}_u; \tag{4.8}$$

После расчета коэффициентов строят регрессионную модель эксперимента в виде (4.4).

Далее необходимо проверить значимость коэффициентов регрессии. Погрешность определения указанных коэффициентов оценивается дисперсией S_{ai}^2. При этом дисперсии оценок всех коэффициентов регрессии одинаковы. Величина S_{ai}^2 зависит только от ошибки (дисперсии) воспроизводимости эксперимента S_y^2 и числа опытов

$$S_{a1}^2 = \frac{S_y^2}{n\cdot m}. \tag{4.9}$$

Проверка значимости коэффициентов регрессии выполняется с помощью t-критерия Стьюдента, расчетное значение которого определяется соотношением

$$t_i = \frac{|a_i|}{S_{ai}}. \tag{4.10}$$

Полученное значение t_i для каждого коэффициента регрессии сравнивается с табличным $t(0.05;\ k)$, определяемым при принятом уровне значимости и числе степеней свободы $k = n\cdot(m-1)$, с которым определяется дисперсия воспроизводимости.

Если $t_i < t(0.05;\ k)$, то коэффициент a_i незначим и член уравнения регрессии, включающий этот коэффициент исключается из математической модели.

Если же $t_i > t(0.05;\ k)$, то коэффициент a_i значим и его следует сохранить в регрессионной модели. В этом случае значение коэффициента a_i больше ошибки опыта, которую можно оценить величиной доверительного интервала Δa_i, определяемого по формуле

$$\Delta a_i = \pm t\cdot S_{ai}. \qquad 4.11)$$

Далее необходимо привести полученную линейную модель эксперимента в виде (5.6).

Затем необходимо проверить адекватность полученной модели с помощью критерия Фишера. Согласно этой проверке модель считается адекватной, если имеет место неравенство

$$F = \frac{S_{adk}^2}{S_y^2} \le F(0{,}05; k_3; \mathrm{k}_2) \tag{4.12}$$

где $\mathrm{S}_{\text{адк}}^2 = \dfrac{\sum\limits_{u=1}^{n}\left(\bar{y}_u - y_{pu}\right)^2}{n-k-1}$ - дисперсия адекватности;

y_{pu} - расчетное значение функции отклика в u -м опыте;

n- количество опытов, строк в матрице планирования;

$F(0{,}05;\ k_3;\ k_2)$ - табличное значении критерия Фишера при 5% -ом уровне зависимости;

$k_3 = n - k - 1$ - число степеней свободы дисперсии адекватности;

При увеличении числа факторов порядок построения плана эксперимента и статистических оценок не изменится, а лишь несколько усложнится.

В табл. 4.4 приведена матрица планирования ПФЭ 2^k для k=2, 3, 4, 5.

Таблица 4.4 – Матрица планирования ПФЭ 2^k для k=2, 3, 4, 5

ПФЭ 2^k				№ опыта	x_0	x_1	x_2	x_3	x_4	x_5	Y
2^5	2^4	2^3	2^2								
1	2	3	4	5	6	7	8	9	10	11	12
				1	*+1*	*+1*	*+1*	+1	*+1*	*+1*	y_1
			2^2	*2*	*+1*	*-1*	*+1*	+1	*+1*	*+1*	Y_2
				3	*+1*	*+1*	*-1*	+1	*+1*	*+1*	Y_3
		2^3		*4*	*+1*	*-1*	*-1*	+1	*+1*	*+1*	Y_4
				5	+1	+1	+1	-*1	*+1*	*+1*	Y_5
				6	+1	-1	+1	-1	*+1*	*+1*	Y_6
				7	+1	+1	-1	-1	*+1*	*+1*	Y_7
	2^4			8	+1	-1	-1	-1	*+1*	*+1*	Y_8
				9	*+1*	*+1*	*+1*	*+1*	*-1*	*+1*	Y_9
				10	*+1*	*-1*	*+1*	*+1*	*-1*	*+1*	y_{10}
				11	*+1*	*+1*	*-1*	*+1*	*-1*	*+1*	y_{11}
				12	*+1*	*-1*	*-1*	*+1*	*-1*	*+1*	y_{12}
				13	*+1*	*+1*	*+1*	*-1*	*-1*	*+1*	y_{13}
				14	*+1*	*-1*	*+1*	*-1*	*-1*	*+1*	y_{14}
				15	*+1*	*+1*	*-1*	*-1*	*-1*	*+1*	y_{15}
				16	*+1*	*-1*	*-1*	*-1*	*-1*	*+1*	y_{16}
2^5				17	+1	+1	+1	+1	+1	-1	y_{17}
				18	+1	-1	+1	+1	+1	-1	y_{18}
				19	+1	+1	-1	+1	+1	-1	y_{19}
				20	+1	-1	-1	+1	+1	-1	Y_{20}
				21	+1	+1	+1	-1	+1	-1	Y_{21}
				22	+1	-1	+1	-1	+1	-1	Y_{22}
				23	+1	+1	-1	-1	+1	-1	Y_{23}
				24	+1	-1	-1	-1	+1	-1	Y_{24}
				25	+1	+1	+1	+1	-1	-1	Y_{25}
				26	+1	-1	+1	+1	-1	-1	Y_{26}
				27	+1	+1	-1	+1	-1	-1	Y_{27}
				28	+1	-1	-1	+1	-1	-1	Y_{28}
				29	+1	+1	+1	-1	-1	-1	Y_{29}
				30	+1	-1	+1	-1	-1	-1	Y_{30}
				31	+1	+1	-1	-1	-1	-1	Y_{31}
				32	+1	-1	-1	-1	-1	-1	Y_{32}

Дробный факторный эксперимент

С увеличением числа факторов количество опытов в полном факторном эксперименте резко возрастает. Так при трех факторах необходимо поставить 2^3 = 8 опытов, при восьми – 256 опытов. Поэтому целесообразно сократить число опытов за счет информации, которую несут эффекты взаимодействия факторов и которая для построения линейного плана несущественна.

Рассмотрим ПФЭ 2^2, план которого приведен в табл.4.2. На основе этого плана можно построить модель

$$y= a_o + a_1 \cdot x_1 + a_2 \cdot x_2 + a_{12} \cdot x_1 \cdot x_2. \tag{4.13}$$

Однако, если процесс описывается линейной моделью, то достаточно определить только три коэффициента a_o, a_1, a_2, а квадратичный коэффициент a_{12} будет достаточно малым. Если включит в матрицу планирования вместо столбца $x_1 \cdot x_2$ фактор x_3, то получится матрица планирования для трех факторов, линейная модель которой

$$y = a_o + a_1 \cdot x_1 + a_2 \cdot x_2 + a_3 \cdot x_3 \qquad (4.14)$$

Коэффициенты этой модели рассчитываются по формулам 4.6…4.8, где коэффициент a_3 совпадает со значением a_{12}.

В результате сокращения числа опытов имеет место значительная потеря информации по сравнению с ПФЭ. Например, ПФЭ с тремя факторами содержит 8 опытов, а не 4. Однако, поскольку эти опыты нужны для построения линейной модели, то парными взаимодействиями факторов можно пренебречь, предполагая, что основные эффекты более значимы по сравнению с парными взаимодействиями. В табл. 4.5 приведены 4 опыта, необходимые для оценки влияния трех факторов.

Таблица 4.5 – План-матрица эксперимента

№ опыта	x_0	x_1	x_2	$x_1 \cdot x_2 = x_3$	y
1	+1	+1	+1	+1	y_1
2	+1	-1	+1	-1	y_2
3	+1	+1	-1	-1	y_3
4	+1	-1	-1	+1	y_4

Приведенный план представляет собой половину ПФЭ или <u>*дробную реплику*</u>.

При организации планирования эксперимента наиболее распространены <u>*регулярные дробные реплики*</u>, которые получают делением числа опытов соответствующего ПФЭ на число, кратное 2. Эти реплики условно обозначают 2^{k-p}, где k – число факторов; p – число линейных эффектов, приравниваемых эффектам взаимодействия. Составляют дробные реплики заменой некоторых эффектов взаимодействия новыми независимыми переменными.

Например, ПФЭ $2^6 = 64$ опыта; его полуреплика содержит $2^{6-1} = 32$ опыта; четвертьреплика - $2^{6-2} = 16$ опытов и т.д. Например, дробный факторный эксперимент, приведенный в табл. 4.8, представляет собой полуреплику 2^{3-1}, которая построена в результате отождествления

$$x_3 = x_1 \, x_2. \qquad (4.15)$$

Равенство (4.16) называют генерирующим отношением.

Построим матрицу планирования полуреплики 2^{4-1}. Результаты представим в табл.4.6.

Таблица 4.6 – План-матрица эксперимента

№ опыта	x_0	x_1	x_2	x_3	x_4	y
1	+1	+1	+1	+1	+1	y_1
2	+1	-1	+1	+1	-1	y_2
3	+1	+1	-1	+1	-1	y_3
4	+1	-1	-1	+1	+1	y_4
5	+1	+1	+1	-1	-1	y_{55}
6	+1	-1	+1	-1	+1	y_6
7	+1	+1	-1	-1	+1	y_7
8	+1	-1	-1	-1	-1	y_8

Здесь $x_4 = x_1 \, x_2 \, x_3$.

С помощью матрицы 4.6 можно построить линейную модель вида

$$y = a_o + a_1 \cdot x_1 + a_2 \cdot x_2 + a_3 \cdot x_3 + a_4 \cdot x_4 \qquad (4.16)$$

Свойства полного и дробного факторных экспериментов

Полному и дробному факторным экспериментам присущи следующие свойства:

1.Симметричность относительно центра эксперимента:

$$\sum_{u=1}^{N} x_{ui} = 0,$$

т.е. сумма элементов любого столбца матрицы планирования ровна нулю.

2.Нормировка: $\sum_{u=1}^{N} x_{ui}^2 = N$,

т.е. сумма квадратов элементов любого столбца ровна числу опытов.

3.Ортогональность: $\sum_{u=1}^{N} x_{ui} \cdot x_{uj} = 0, i \neq j$,

т.е. сумма почленных произведений любых двух столбцов равна нулю.

4.Ротатабельность – способность математической модели, полученной в результате полного и дробного экспериментов, предсказывать значение параметра y с одинаковой точностью на равных расстояниях от центра эксперимента, независимо от направления.

Свойства 1 и 2 следуют из способа построения матрицы планирования, т.к. факторы равны +1 и -1.

Построение математической модели в натуральных единицах

Полученные ранее регрессионные математические модели являются нормированными, т.к. значения факторов в них могут принимать значения от –1 до +1. Для практического использования математической модели удобнее, когда факторы в ней принимают натуральные значения.

Пересчет коэффициентов модели к натуральным значениям факторов производится на основании выражения нормирования факторов

$$x_i = \frac{\lambda_i - \overline{\lambda}_i}{\Delta\lambda},$$

где x_i – нормированное значение фактора;

λ_i – нижний или верхний уровень фактора;

$\overline{\lambda}_i$ - нулевой уровень фактора;

$\Delta\lambda_i$ – интервал варьирования фактора.

Откуда

$$\lambda_i = \overline{\lambda}_i + x_i \cdot \Delta\lambda.$$

<u>*Пример 4.1*</u>. Пусть требуется исследовать процесс износа деталей некоторого агрегата в зависимости от конечной температуры и скорости его нагрева. Известно, что температура изменяется от *250°С* до *450°С*, а скорость нагрева – от *2К/мин* до *10К/мин*.

<u>Решение</u>

Очевидно, что в задаче фигурируют два независимых фактора: температура нагрева и скорость нагрева агрегата. Введем обозначения факторов и выполним их кодирование, табл. 4.7.

Таблица 4.7 - Кодирование факторов

Интервал варьирования и уровни факторов	Температура нагрева, x_1, *°С*	Скорость нагрева агрегата, x_2, *К/мин*
Интервал варьирования, Δx_i	50	2
Нулевой уровень, $x_{i0}=0$	350	6
Нижний уровень, $x_i=-1$	300	4
Верхний уровень, $x_i=1$	400	8

Составим план-матрицу эксперимента. В данном примере варьируются два фактора x_1 и x_2 на двух уровнях: +1, -1. Разработанная расширенная план-матрица приведена в табл.4.8. Строки этой таблицы соответствуют различным опытам, а столбцы – значениям факторов.

Таблица 4.8.- Расширенная план-матрица эксперимента

Номер опыта	x_1	x_2	$x_1{\cdot}x_2$
1	2	3	4
1	-1	-1	+1
2	+1	-1	-1
3	-1	+1	-1
4	+1	+1	+1

В столбцах 2 и 3 табл.4.6 записаны уровни варьируемых факторов, в столбце 4 – их произведение. Столбец 4 используется для учета эффекта взаимодействия факторов.

В процессе реализации плана эксперимента заполняются столбцы 5 и 6 табл. 4.9. Причем эксперимент проводится два раза. В столбце 7 записывается вычисленное среднее значение двух серий эксперимента y_{n1} и y_{n2}, характеризующих износ деталей агрегата.

Таблица 4.9 - Результаты опытов

Номер опыта	x_1	x_2	$x_1{\cdot}x_2$	y_{n1}	y_{n2}	$y_n = (y_{n1} + y_{n2})/2$
1	2	3	4	5	6	7
1	-1	-1	+1	27,0	28,0	27,5
2	+1	-1	-1	15,9	17,1	16,5
3	-1	+1	-1	22,1	22,9	22,5
4	+1	+1	+1	13,4	13,6	13,5

Проверим воспроизводимость данного процесса. Для этого вычислим значение критерия Кохрена.

Рассчитаем значение оценок дисперсии в каждой точке плана по формуле

$$S_u^2 = \frac{\Delta^2}{2},$$

где Δ - разность между значениями функции отклика в параллельных опытах

$$\Delta = y_{n1} - y_{n2};$$

$$S^2_1 = (27.0 + 28.0)^2 / 2 = 0.50;$$

$$S^2_2 = (15.9 + 17.1)^2 / 2 = 0.72;$$

$$S^2_3 = (22.1 + 22.9)^2 / 2 = 0.32,$$

$$S^2_4 = (13.4 + 13.6)^2 / 2 = 0.02;$$

Проверим воспроизводимость данного процесса. Для этого вычислим значение критерия Кохрена по формуле 4.5.

$$\sigma = 0.72/(0.5+0.72+0.32+0.02) = 0.4615$$

Сравнивая вычисленное значение с табличным σ(0,05;4, 1) = 0.9065, заключаем, что рассматриваемый процесс воспроизводим.

Дисперсия воспроизводимости (ошибка эксперимента)

$$S^2_y = (0.5+0.72+0.32+0.02)/4 = 0.39.$$

Построим линейную модель эксперимента, вычислив коэффициенты регрессии:

$$a_0 = (27.5+16.5+22.5+13.5)/4 = 20;$$

$$a_1 = (-27.5+16.5-22.5+13.5)/4 = -5;$$

$$a_2 = (-27.5-16.5+22.5+13.5)/4 = -2;$$

a_{12} = (27.5-16.5-22.5+13.5)/4 = 0.5;

Тогда математическая модель примет вид

$$y_p = 20 - 5 \cdot x_1 - 2 \cdot x_2 \quad (4.17)$$

или с учетом коэффициента a_{12}

$$y_p = 20 - 5 \cdot x_1 - 2 \cdot x_2 + 0{,}5 \cdot x_1 \cdot x_2. \quad (4.18)$$

Рассчитаем значения y_p по полученным формулам. Результаты сведем в табл. 4.10, в столбце 8 по формуле 4.18 и в столбце 9 по формуле 4.19.

Таблица 4.10.

Результаты опытов и расчетов по модели

Номер опыта	x_1	x_2	$x_1 \cdot x_2$	y_{n1}	y_{n2}	$y_n = (y_{n1} + y_{n2})/2$	y_p	y^*_p
1	2	3	4	5	6	7	8	9
1	-1	-1	+1	27,0	28,0	27,5	27,0	27,5
2	+1	-1	-1	15,9	17,1	16,5	17,0	16,5
3	-1	+1	-1	22,1	22,9	22,5	23,0	22,5
4	+1	+1	+1	13,4	13,6	13,5	13,0	13,5

Проверим адекватность построенной модели условиям эксперимента, воспользовавшись критерием Фишера (4.10). Для этого сначала вычислим значение дисперсии адекватности $S^2_{адк}$ по формуле (4.11).

$S^2_{адк}$ =[(27-27.5)^2+(17-16.5)^2+(23-22.5)^2 +

(13-13.5)^2]/(4-2-1) = 1/(4-2-1) = 1.

Тогда значение критерия Фишера

F = 1,0 / 0.39 = 2.564.

Табличное значение критерия Фишера

F(0,05; 1; 4) = 7.7086.

Значит F = 2.564 < *F(0,05; 1; 4)* = 7.7086. Следовательно, можно утверждать, что адекватность рассмотренной модели обеспечивается.

Проверим значимость коэффициентов регрессии по критерию Стьюдента

$\Delta a_i = t(0{,}05; f_y)S_y / \sqrt{n}$;

t(0,05; 4) =2.7764.

$\Delta a_i = 2.7764\ \sqrt{0.39} / \sqrt{4} = 0.8669$.

Поскольку коэффициенты регрессии a_0, a_1, a_2, > Δa_i, то все эти коэффициенты значимы.

<u>*Пример 4.2*</u>. Пусть требуется исследовать систему массового обслуживания рис.4.1 на основе данных расчетного эксперимента

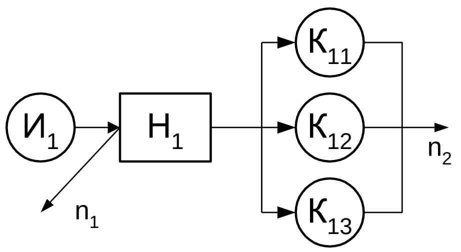

Рис. 4.1. Однофазная СМО

Требуется построить регрессионную математическую модель СМО на основе планирования эксперимента для исследования влияния интенсивностей входного потока (λ) и потока обслуживания (μ) на вероятность отказа. Модель построить в кодированных и натуральных единицах.

Известно, что $\boldsymbol{\lambda}$ изменяется в пределах (4…12 ед/мин, $\boldsymbol{\mu}$ - (0,2…0,8) ед/мин.

Решение

В задаче фигурируют два независимых фактора: λ и μ. Введем обозначения факторов и выполним их кодирование, табл. 4.11.

Таблица 4.11 - Кодирование факторов

Интервал варьирования и уровни факторов	Интенсивность λ, x_1	Интенсивность μ, x_2
Интервал варьирования, Δx_i	4,0	0.3
Нулевой уровень, $x_{i0}=0$	8,0	0.5
Нижний уровень, $x_i=-1$	4,0	0,2
Верхний уровень, $x_i=1$	12,0	0,8

Составим план-матрицу эксперимента. В данном примере варьируются два фактора x_1 и x_2 на двух уровнях: +1, -1. Разработанная расширенная план-матрица приведена в табл.4.12. Строки этой таблицы соответствуют различным опытам, а столбцы – значениям факторов.

Таблица 4.12 - Расширенная план-матрица эксперимента

Номер опыта	x_1	x_2	$x_1 \cdot x_2$
1	2	3	4
1	-1	-1	+1
2	+1	-1	-1
3	-1	+1	-1
4	+1	+1	+1

В столбцах 2 и 3 табл.4.12 записаны уровни варьируемых факторов, в столбце 4 – их произведение. Столбец 4 используется для учета эффекта взаимодействия факторов.

В процессе реализации плана эксперимента заполняются столбцы 5 и 6 табл. 4.13. Причем эксперимент проводится два раза. В столбце 7 записывается вычисленное среднее значение двух серий эксперимента y_1 и y_2, характеризующих износ деталей агрегата.

Таблица 4.13 - Результаты опытов

Номер опыта	x_1	x_2	$x_1 \cdot x_2$	y_1	y_2	$y_n = (y_1 + y_2)/2$
1	2	3	4	5	6	7
1	-1	-1	+1	0,480	0,460	0,470
2	+1	-1	-1	0,780	0,760	0,770
3	-1	+1	-1	0,0	0,040	0,020
4	+1	+1	+1	0,710	0,68	0,695

Рассчитаем значение оценок дисперсии S_u^2 в каждой точке плана по формуле

$$S^2{}_1 = (0.48 - 0.46)^2 / 2 = 0.00020;$$

$$S^2{}_2 = (0.78 - 0.76)^2 / 2 = 0.00020;$$

$$S^2{}_3 = (0.00 - 0.04)^2 / 2 = 0.00080,$$

$$S^2{}_4 = (0.71 - 0.68)^2 / 2 = 0.00045;$$

Проверим воспроизводимость данного процесса. Для этого вычислим значение критерия Кохрена по формуле 4.5.

$$\sigma = 0.0008/(0.0002+0.0002+0.0008+0.00045) = 0.4848$$

Сравнивая вычисленное значение с табличным $\sigma(0,05;4;\ 1) = 0.9065$, заключаем, что рассматриваемый процесс воспроизводим.

Дисперсия воспроизводимости (ошибка эксперимента)

$S^2_y = (0.0002+0.0002+0.0008+0.00045)/4 = 0.000403.$

Построим линейную модель эксперимента, вычислив коэффициенты регрессии:

$$a_0 = (0.47+0.77+0.02+0.695)/4 = 0{,}489;$$

$$a_1 = (-0.47-0.77+0.02+0.695)/4 = -0{,}131;$$

$$a_2 = (-0.47+0.77-0.02+0.695)/4 = 0{,}244;$$

$$a_{12} = (0.47-0.77-0.02+0.695)/4 = 0.094.$$

Тогда математическая модель примет вид

$$y = 0{,}48875 – 0{,}13125 \cdot x_1 – 0{,}24375 \cdot x_2; \qquad (4.19)$$

или с учетом коэффициента $\boldsymbol{a_{12}}$

$$y = 0{,}48875 – 0{,}13125 \cdot x_1 – 0{,}24375 \cdot x_2 + 0.09375 \cdot x_1 \cdot x_2; \qquad (4.21)$$

Рассчитаем значения ***y*** по полученной модели и запишем результаты в табл.4.14.

Таблица 4.14 - Результаты опытов и расчета по модели

Номер опыта	x_1	x_2	$x_1 \cdot x_2$	y_1	y_2	$y_n = (y_1 + y_2)/2$	y_p *по 4.20*	y_p *по 4.21*
1	2	3	4	5	6	7	8	9
1	-1	-1	+1	0,48	0,46	0,470	0,376	0,470
2	+1	-1	-1	0,78	0,76	0,770	0,864	0,770
3	-1	+1	-1	0,0	0,04	0,020	0,114	0,020
4	+1	+1	+1	0,71	0,68	0,695	0,601	0,695

Проверим адекватность построенной модели условиям эксперимента, воспользовавшись критерием Фишера (4.10). Для этого сначала вычислим значение дисперсии адекватности $S^2_{адк}$ по формуле (4.11)

$$S^2_{адк} = 0.78/(4-2-1) = 0.78.$$

Тогда значение критерия Фишера

$$F = 0.78/0.39 = 2.0.$$

Табличное значение критерия Фишера

$$F(0{,}05;\ 1;\ 4) = 7.7086.$$

Значит $F = 2.0 < F(0{,}05;\ 1;\ 4) = 7.7086$. Следовательно, можно утверждать, что адекватность рассмотренной модели обеспечивается.

Проверим значимость коэффициентов регрессии по критерию Стьюдента

$$\Delta a_i = t(0{,}05;\ f_y)S_y/\sqrt{n};$$

$$t(0{,}05;\ 4) = 2.132.$$

$$\Delta a_i = 2.132\ \sqrt{0.0004125}/\sqrt{4} = 0.022$$

Поскольку коэффициенты регрессии a_0, a_1, a_2, $> \Delta a_i$, то все они значимы.

5. Инструментальные средства реализации моделей

5.1 Введение

Процессы функционирования различных систем и сетей связи могут быть представлены той или иной совокупностью систем массового обслуживания (СМО) - стохастических, динамических, дискретно-непрерывных математических моделей. Исследование характеристик таких моделей может проводиться либо аналитическими методами, либо путем имитационного моделирования [1 - 6].

Имитационная модель отображает стохастический процесс смены дискретных состояний СМО в непрерывном времени в форме моделирующего алгоритма. При его реализации на ЭВМ производится накопление статистических данных по тем атрибутам модели, характеристики которых являются предметом исследований. По окончании моделирования накопленная статистика обрабатывается, и результаты моделирования получаются в виде выборочных распределений исследуемых величин или их выборочных моментов. Таким образом, при имитационном моделировании систем массового обслуживания речь всегда идет о статистическом имитационном моделировании [5, 6].

Сложные функции моделирующего алгоритма могут быть реализованы средствами универсальных языков программирования (Паскаль, Си), что предоставляет неограниченные возможности в разработке, отладке и использовании модели. Однако подобная гибкость приобретается ценой больших усилий, затрачиваемых на разработку и программирование весьма сложных моделирующих алгоритмов, оперирующих со списковыми структурами данных. Альтернативой этому является использование специализированных языков имитационного моделирования [5 - 7].

Специализированные языки имеют средства описания структуры и процесса функционирования моделируемой системы, что значительно облегчает и упрощает программирование имитационных моделей, поскольку основные функциии моделирующего алгоритма при этом реализуются автоматически. Программы имитационных моделей на специализированных языках моделирования близки к описаниям моделируемых систем на естественном языке, что позволяет конструировать сложные имитационные модели пользователям, не являющимся профессиональными программистами.

Одним из наиболее эффективных и распространенных языков моделирования сложных дискретных систем является в настоящее время язык GPSS [1, 4, 7]. Он может быть с наибольшим успехом использован для моделирования систем, формализуемых в виде систем массового обслуживания. В качестве объектов языка используются аналоги таких стандартных компонентов СМО, как заявки, обслуживающие приборы, очереди и т.п. Достаточный набор подобных компонентов позволяет конструировать сложные имитационные модели, сохраняя привычную терминологию СМО.

На персональных компьютерах (ПК) типа IBM/PC язык GPSS реализован в рамках пакета прикладных программ GPSS/PC [8]. Основной модуль пакета представляет собой интегрированную среду, включающую помимо транслятора со входного языка средства ввода и редактирования текста модели, ее отладки и наблюдения за процессом моделирования, графические средства отображения атрибутов модели, а также средства накопления результатов моделирования в базе данных и их статистической обработки. Кроме основного модуля в состав пакета входит модуль создания стандартного отчета GPSS/PC, а также ряд дополнительных модулей и файлов.

5.2. Общие сведения о GPSS/PC

Исходная программа на языке GPSS/PC, как и программа на любом языке программирования, представляет собой последовательность операторов. Операторы GPSS/PC записываются и вводятся в ПК в следующем формате:

номер_строки имя операция операнды ; комментарии.

Все операторы исходной программы должны начинаться с номера_строки - целого положительного числа от 1 до 9999999. После ввода операторов они располагаются в исходной программе в соответствии с нумерацией строк. Обычно нумерация производится с некоторым шагом, отличным от 1, чтобы иметь возможность добавления операторов в нужное место исходной программы.

Некоторые операторы удобно вводить, не включая их в исходную программу. Такие операторы вводятся без номера строки.

В настоящем издании при описании формата операторов и в примерах моделей номера строк будут опускаться для лучшей читаемости текста.

Отдельные операторы могут иметь имя для ссылки на эти операторы в других операторах. Если такие ссылки отсутствуют, то этот элемент оператора не является обязательным.

В поле операции записывается ключевое слово (название оператора), указывающее конкретную функцию, выполняемую данным оператором. Это поле оператора является обязательным. У некоторых операторов поле операции включает в себя также вспомогательный операнд.

В полях операндов записывается информация, уточняющая и конкретизирующая выполнение функции, определенной в поле операции. Эти поля в зависимости от типа операции содержат до семи операндов, расположенных в определенной последовательности и обозначаемых обычно первыми буквами латинского алфавита от A до G. Некоторые операторы вообще не имеют операндов, а в некоторых операнды могут быть опущены, при этом устанавливаются их стандартные значения (по умолчанию). При записи операндов используется позиционный принцип: пропуск операнда отмечается запятой.

Необязательные комментарии в случае их присутствия отделяются от поля операндов точкой с запятой. Комментарии не могут содержать букв русского алфавита.

Операторы GPSS/PC записываются, начиная с первой позиции, в свободном формате, т.е. отдельные поля разделяются произвольным количеством пробелов. При вводе исходной программы в интегрированной среде GPSS/PC размещение отдельных полей операторов с определенным количеством интервалов между ними производится автоматически.

Каждый оператор GPSS/PC относится к одному из четырех типов: операторы-блоки, операторы определения объектов, управляющие операторы и операторы-команды.

Операторы-блоки формируют логику модели. В GPSS/PC имеется около 50 различных видов блоков, каждый из которых выполняет свою конкретную функцию. За каждым из таких блоков стоит соответствующая подпрограмма транслятора, а операнды каждого блока служат параметрами этой подпрограммы.

Операторы определения объектов служат для описания параметров некоторых объектов GPSS/PC (о самих объектах речь пойдет дальше). Примерами параметров объектов могут быть количество каналов в многоканальной системе массового обслуживания, количество строк и столбцов матрицы и т.п.

Управляющие операторы служат для управления процессом моделирования (прогоном модели). Операторы-команды позволяют управлять работой интегрированной среды GPSS/PC. Управляющие операторы и операторы-команды обычно не включаются в исходную программу, а вводятся непосредственно с клавиатуры ПК в процессе интерактивного взаимодействия с интегрированной средой.

После трансляции исходной программы в памяти ПК создается так называемая текущая модель, являющаяся совокупностью разного типа объектов, каждый из которых представляет собой некоторый набор чисел в памяти ПК, описывающих свойства и текущее состояние объекта. Объекты GPSS/PC можно разделить на семь классов: динамические, операционные, аппаратные, статистические, вычислительные, запоминающие и группирующие.

Динамические объекты, соответствующие заявкам в системах массового обслуживания, называются в GPSS/PC транзактами. Они "создаются" и "уничтожаются" так, как это необходимо по логике модели в процессе моделирования. С каждым транзактом может быть связано произвольное число параметров, несущих в себе необходимую информацию об этом транзакте. Кроме того, транзакты могут иметь различные приоритеты.

Операционные объекты GPSS/PC, называемые блоками, соответствуют операторам-блокам исходной программы. Они, как уже говорилось, формируют логику модели, давая транзактам указания: куда идти и что делать дальше. Модель системы на GPSS/PC можно представить совокупностью блоков, объединенных в соответствии с логикой работы реальной системы в так называемую блок-схему. Блок-схема модели может быть изображена графически, наглядно показывая взаимодействие блоков в процессе моделирования.

Аппаратные объекты GPSS/PC - это абстрактные элементы, на которые может быть расчленено (декомпозировано) оборудование реальной системы. К ним относятся одноканальные и многоканальные устройства и логические переключатели. Многоканальное устройство иногда называют памятью.

Одноканальные и многоканальные устройства соответствуют обслуживающим приборам в СМО. Одноканальное устройство, которое для краткости далее будем называть просто устройством, может обслуживать одновременно только один транзакт. Многоканальное устройство (МКУ) может обслуживать одновременно несколько транзактов. Логические переключатели (ЛП) используются для моделирования двоичных состояний логического или физического характера. ЛП может находиться в двух состояниях: включено и выключено. Его состояние может изменяться в процессе моделирования, а также опрашиваться для принятия определенных решений.

Статистические объекты GPSS/PC служат для сбора и обработки статистических данных о функционировании модели. К ним относятся очереди и таблицы.

Каждая очередь обеспечивает сбор и обработку данных о транзактах, задержанных в какой-либо точке модели, например перед одноканальным устройством. Таблицы используются для получения выборочных распределений некоторых случайных величин, например времени пребывания транзакта в модели.

К вычислительным объектам GPSS/PC относятся переменные (арифметические и булевские) и функции. Они используются для вычисления некоторых величин, заданных арифметическими или логическими выражениями либо табличными зависимостями.

Запоминающие объекты GPSS/PC обеспечивают хранение в памяти ПК отдельных величин, используемых в модели, а также массивов таких величин. К ним относятся так называемые сохраняемые величины и матрицы сохраняемых величин.

К объектам группирующего класса относятся списки пользователя и группы. Списки пользователя используются для организации очередей с дисциплинами, отличными от дисциплины "раньше пришел - раньше обслужен". Группы в данном издании рассматриваться не будут.

Каждому объекту того или иного класса соответствуют числовые атрибуты, описывающие его состояние в данный момент модельного времени. Кроме того, имеется ряд так называемых системных атрибутов, относящихся не к отдельным объектам, а к модели в целом. Значения атрибутов всех объектов модели по окончании моделирования выводятся в стандартный отчет GPSS/PC. Большая часть атрибутов доступна программисту и составляет так называемые стандартные числовые атрибуты (СЧА), которые могут использоваться в качестве операндов операторов исходной программы. Все СЧА в GPSS/PC являются целыми числами.

Каждый объект GPSS/PC имеет имя и номер. Имена объектам даются в различных операторах исходной программы, а соответствующие им номера транслятор присваивает автоматически. Имя объекта представляет собой начинающуюся с буквы последовательность букв латинского алфавита, цифр и символа "подчеркивание". При необходимости имени любого объекта, кроме имени блока, можно поставить в соответствие любой номер с помощью оператора описания EQU, имеющего следующий формат:

имя EQU номер.

Блокам присваиваются их порядковые номера в исходной программе (не путать с номерами строк!).

Для ссылки на какой-либо стандартный числовой атрибут некоторого объекта соответствующий операнд оператора исходной программы записывается одним из следующих способов:

СЧА$имя

СЧАj

где СЧА - системное обозначение (название) конкретного стандартного числового атрибута данного объекта; имя - имя объекта; j - номер объекта; $ - символ-разделитель.

Прогон текущей модели, т.е. собственно моделирование, выполняется с помощью специальной управляющей программы, которую называют симулятором (от английского SIMULATE -

моделировать, имитировать). Работа GPSS-модели под управлением симулятора заключается в перемещении транзактов от одних блоков к другим, аналогично тому, как в моделируемой СМО перемещаются заявки, соответствующие транзактам.

В начальный момент времени в GPSS-модели нет ни одного транзакта. В процессе моделирования симулятор генерирует транзакты в определенные моменты времени в соответствии с теми логическими потребностями, которые возникают в моделируемой системе. Подобным же образом транзакты покидают модель в определенные моменты времени в зависимости от специфики моделируемой системы. В общем случае в модели одновременно существует большое число транзактов, однако в каждый момент времени симулятор осуществляет продвижение только какого-либо одного транзакта.

Если транзакт начал свое движение, он перемещается от блока к блоку по пути, предписанному блок-схемой. В тот момент, когда транзакт входит в некоторый блок, на исполнение вызывается подпрограмма симулятора, соответствующая типу этого блока, а после ее выполнения, при котором реализуется функция данного блока, транзакт "пытается" войти в следующий блок. Такое продвижение транзакта продолжается до тех пор, пока не произойдет одно из следующих возможных событий:

1) транзакт входит в блок, функцией которого является удаление транзакта из модели;

2) транзакт входит в блок, функцией которого является задержка транзакта на некоторое определенное в модели время;

3) транзакт "пытается" войти в следующий блок, однако блок "отказывается" принять его. В этом случае транзакт остается в том блоке, где находился, и позднее будет повторять свою попытку войти в следующий блок. Когда условия в модели изменятся, такая попытка может оказаться успешной, и транзакт сможет продолжить свое перемещение по блок-схеме.

Если возникло одно из описанных выше условий, обработка данного транзакта прекращается, и начинается перемещение другого транзакта. Таким образом, выполнение моделирования симулятором продолжается постоянно.

Проходя через блоки модели, каждый транзакт вносит вклад в содержимое счетчиков блоков. Значения этих счетчиков доступны программисту через СЧА блоков: W - текущее содержимое блока и N - общее количество входов в блок.

Каждое продвижение транзакта в модели является событием, которое должно произойти в определенный момент модельного времени. Для того, чтобы поддерживать правильную временную последовательность событий, симулятор имеет таймер модельного времени, который автоматически корректируется в соответствии с логикой, предписанной моделью.

Таймер GPSS/PC имеет следующие особенности:

1) регистрируются только целые значения (все временные интервалы в модели изображаются целыми числами);

2) единица модельного времени определяется разработчиком модели, который задает все временные интервалы в одних и тех же, выбранных им единицах;

3) симулятор не анализирует состояние модели в каждый следующий момент модельного времени (отстоящий от текущего на единицу модельного времени), а продвигает таймер к моменту времени, когда происходит ближайшее следующее событие.

Значения таймера доступны программисту через системные СЧА C1 (относительное время) и AC1 (абсолютное время).

Центральной задачей, выполняемой симулятором, является определение того, какой транзакт надо выбрать следующим для продвижения в модели, когда его предшественник прекратил свое продвижение. С этой целью симулятор рассматривает каждый транзакт как элемент некоторого списка. В относительно простых моделях используются лишь два основных списка: список текущих событий и список будущих событий.

Список текущих событий включает в себя те транзакты, планируемое время продвижения которых равно или меньше текущего модельного времени (к последним относятся транзакты, движение которых было заблокировано ранее). Он организуется в порядке убывания приоритетов транзактов, а в пределах каждого уровня приоритета - в порядке поступления транзактов.

Список будущих событий включает в себя транзакты, планируемое время продвижения которых больше текущего времени, т.е. события, связанные с продвижением этих транзактов, должны произойти в будущем. Этот список организуется в порядке возрастания планируемого времени продвижения транзактов.

Симулятор GPSS/PC помещает транзакты в зависимости от условий в модели в тот или иной список и переносит транзакты из списка в список, просматривает списки, выбирая следующий транзакт для обработки, корректирует таймер модельного времени после обработки всех транзактов в списке текущих событий.

5.2. Основные блоки GPSS/PC и связанные с ними объекты

Блоки, связанные с транзактами

С транзактами связаны блоки создания, уничтожения, задержки транзактов, изменения их атрибутов и создания копий транзактов.

Для создания транзактов, входящих в модель, служит блок GENERATE (генерировать), имеющий следующий формат:

имя GENERATE A,B,C,D,E

В поле A задается среднее значение интервала времени между моментами поступления в модель двух последовательных транзактов. Если этот интервал постоянен, то поле B не используется. Если же интервал поступления является случайной величиной, то в поле B указывается модификатор среднего значения, который может быть задан в виде модификатора-интервала или модификатора-функции.

Модификатор-интервал используется, когда интервал поступления транзактов является случайной величиной с равномерным законом распределения вероятностей. В этом случае в поле B может быть задан любой СЧА, кроме ссылки на функцию, а диапазон изменения интервала поступления имеет границы A-B, A+B.

Например, блок

GENERATE 100,40

создает транзакты через случайные интервалы времени, равномерно распределенные на отрезке [60;140].

Модификатор-функция используется, если закон распределения интервала поступления отличен от равномерного. В этом случае в поле B должна быть записана ссылка на функцию (ее СЧА), описывающую этот закон, и случайный интервал поступления определяется, как целая часть произведения поля A (среднего значения) на вычисленное значение функции.

В поле C задается момент поступления в модель первого транзакта. Если это поле пусто или равно 0, то момент появления первого транзакта определяется операндами A и B.

Поле D задает общее число транзактов, которое должно быть создано блоком GENERATE. Если это поле пусто, то блок генерирует неограниченное число транзактов до завершения моделирования.

В поле E задается приоритет, присваиваемый генерируемым транзактам. Число уровней приоритетов неограничено, причем самый низкий приоритет - нулевой. Если поле E пусто, то генерируемые транзакты имеют нулевой приоритет.

Транзакты имеют ряд стандартных числовых атрибутов. Например, СЧА с названием PR позволяет ссылаться на приоритет транзакта. СЧА с названием M1 содержит так называемое резидентное время транзакта, т.е. время, прошедшее с момента входа транзакта в модель через блок GENERATE. СЧА с названием XN1 содержит внутренний номер транзакта, который является уникальным и позволяет всегда отличить один транзакт от другого. В отличие от СЧА других объектов, СЧА транзактов не содержат ссылки на имя или номер транзакта. Ссылка на СЧА транзакта всегда относится к активному транзакту, т.е. транзакту, обрабатываемому в данный момент симулятором.

Важными стандартными числовыми атрибутами транзактов являются значения их параметров. Любой транзакт может иметь неограниченное число параметров, содержащих те или иные числовые значения. Ссылка на этот СЧА транзактов всегда относится к активному транзакту и имеет вид Pj или P$имя, где j и имя - номер и имя параметра соответственно. Такая ссылка возможна только в том случае, если параметр с указанным номером или именем существует, т.е. в него занесено какое-либо значение.

Для присваивания параметрам начальных значений или изменения этих значений служит блок ASSIGN (присваивать), имеющий следущий формат:

имя ASSIGN A,B,C

В поле A указывается номер или имя параметра, в который заносится значение операнда B. Если в поле A после имени (номера) параметра стоит знак + или -, то значение операнда B добавляется или вычитается из текущего содержимого параметра. В поле C может быть указано имя или номер функции-модификатора, действующей аналогично функции-модификатору в поле B блока GENERATE.

Например, блок

ASSIGN 5,0

записывает в параметр с номером 5 значение 0, а блок

ASSIGN COUNT+,1

добавляет 1 к текущему значению параметра с именем COUNT.

Для записи текущего модельного времени в заданный параметр транзакта служит блок MARK (отметить), имеющий следующий формат:

имя MARK A

В поле A указывается номер или имя параметра транзакта, в который заносится текущее модельное время при входе этого транзакта в блок MARK. Содержимое этого параметра может быть позднее использовано для определения транзитного времени пребывания транзакта в какой-то части модели с помощью СЧА с названием MP.

Например, если на входе участка модели поместить блок

MARK MARKER ,

то на выходе этого участка СЧА MP$MARKER будет содержать разность между текущим модельным временем и временем, занесенным в параметр MARKER блоком MARK.

Если поле A блока MARK пусто, то текущее время заносится на место отметки времени входа транзакта в модель, используемой при определении резидентного времени транзакта с помощью СЧА M1.

Для изменения приоритета транзакта служит блок PRIORITY (приоритет), имеющий следующий формат:

имя PRIORITY A,B

В поле A записывается новый приоритет транзакта. В поле B может содержаться ключевое слово BU, при наличии которого транзакт, вошедший в блок, помещается в списке текущих событий после всех остальных транзактов новой приоритетной группы, и список текущих событий просматривается с начала. Использование такой возможности будет рассмотрено ниже.

Для удаления транзактов из модели служит блок TERMINATE (завершить), имеющий следующий формат:

имя TERMINATE A

Значение поля A указывает, на сколько единиц уменьшается содержимое так называемого счетчика завершений при входе транзакта в данный блок TERMINATE. Если поле A не определено, то оно считается равным 0, и транзакты, проходящие через такой блок, не уменьшают содержимого счетчика завершений.

Начальное значение счетчика завершений устанавливается управляющим оператором START (начать), предназначенным для запуска прогона модели. Поле A этого оператора содержит начальное значение счетчика завершений (см. разд. 3). Прогон модели заканчивается, когда содержимое счетчика завершений обращается в 0. Таким образом, в модели должен быть хотя бы один блок TERMINATE с непустым полем A, иначе процесс моделирования никогда не завершится.

Текущее значение счетчика завершений доступно программисту через системный СЧА TG1.

Участок блок-схемы модели, связанный с парой блоков GENERATE-TERMINATE, называется сегментом. Простые модели могут состоять из одного сегмента, в сложных моделях может быть несколько сегментов.

Например, простейший сегмент модели, состоящий всего из двух блоков GENERATE и TERMINATE и приведенный на рис. 1, в совокупности с управляющим оператором START моделирует процесс создания случайного потока транзактов, поступающих в модель со средним интервалом в 100 единиц модельного времени, и уничтожения этих транзактов. Начальное значение счетчика завершений равно 1000. Каждый транзакт, проходящий через блок TERMINATE, вычитает из счетчика единицу, и таким образом моделирование завершится, когда тысячный по счету транзакт войдет в блок TERMINATE. При этом точное значение таймера в момент завершения прогона непредсказуемо. Следовательно, в приведенном примере продолжительность прогона устанавливается не по модельному времени, а по количеству транзактов, прошедших через модель.

```
GENERATE 100,40
TERMINATE 1
START 1000
```

Рис. 1

Если необходимо управлять продолжительностью прогона по модельному времени, то в модели используется специальный сегмент, называемый сегментом таймера.

```
GENERATE 100,40
TERMINATE
GENERATE 100000
TERMINATE 1
START 1
```

Рис. 2

Например, в модели из двух сегментов, приведенной на рис. 2, первый (основной) сегмент выполняет те же функции, что и в предыдущем примере. Заметим, однако, что поле А блока TERMINATE в первом сегменте пусто, т.е. уничтожаемые транзакты не уменьшают содержимого счетчика завершений. Во втором сегменте блок GENERATE создаст первый транзакт в момент модельного времени, равный 100000. Но этот транзакт окажется и последним в данном сегменте, так как, войдя в блок TERMINATE, он обратит в 0 содержимое счетчика завершений, установленное оператором START равным 1. Таким образом, в этой модели гарантируется завершение прогона в определенный момент модельного времени, а точное количество транзактов, прошедших через модель, непредсказуемо.

В приведенных примерах транзакты, входящие в модель через блок GENERATE, в тот же момент модельного времени уничтожались в блоке TERMINATE. В моделях систем массового обслуживания заявки обслуживаются приборами (каналами) СМО в течение некоторого промежутка времени прежде, чем покинуть СМО. Для моделирования такого обслуживания, т.е. для задержки транзактов на определенный отрезок модельного времени, служит блок ADVANCE (задержать), имеющий следующий формат:

имя ADVANCE A,B

Операнды в полях А и В имеют тот же смысл, что и в соответствующих полях блока GENERATE. Следует отметить, что транзакты, входящие в блок ADVANCE, переводятся из списка текущих событий в список будущих событий, а по истечении вычисленного времени задержки возвращаются назад, в список текущих событий, и их продвижение по блок-схеме продолжается.

Если вычисленное время задержки равно 0, то транзакт в тот же момент модельного времени переходит в следующий блок, оставаясь в списке текущих событий.

Например, в сегменте, приведенном на рис. 3, транзакты, поступающие в модель из блока GENERATE через случайные интервалы времени, имеющие равномерное распределение на отрезке [60;140], попадают в блок ADVANCE. Здесь определяется случайное время задержки транзакта, имеющее равномерное распределение на отрезке [30;130], и транзакт переводится в список будущих событий. По истечении времени задержки транзакт возвращается в список текущих событий и входит в блок TERMINATE, где уничтожается. Заметим, что в списке будущих событий, а значит и в блоке ADVANCE может одновременно находиться произвольное количество транзактов.

```
GENERATE 100,40
ADVANCE 80,50
TERMINATE 1
```

Рис. 3

В рассмотренных выше примерах случайные интервалы времени подчинялись равномерному закону распределения вероятностей. Для получения случайных величин с другими распределениями в GPSS/PC используются вычислительные объекты: переменные и функции.

Как известно, произвольная случайная величина связана со случайной величиной R, имеющей равномерное распределение на отрезке [0;1], через свою обратную функцию распределения. Для некоторых случайных величин уравнение связи имеет явное решение, и значение случайной величины с заданным распределением вероятностей может быть вычислено через R по формуле. Так, например, значение случайной величины E с показательным (экспоненциальным) распределением с параметром d вычисляется по формуле:

E= -(1/d)*ln(R)

Напомним, что параметр d имеет смысл величины, обратной математическому ожиданию E, а, следовательно, 1/d - математическое ожидание (среднее значение) случайной величины E.

Для получения случайной величины R с равномерным распределением на отрезке [0;1] в GPSS/PC имеются встроенные генераторы случайных чисел. Для получения случайного числа путем обращения к такому генератору достаточно записать системный СЧА RN с номером генератора, например RN1. Правда, встроенные генераторы случайных чисел GPSS/PC дают числа не на отрезке [0;1], а целые случайные числа, равномерно распределенные от 0 до 999, но их нетрудно привести к указанному отрезку делением на 1000.

Проще всего описанные вычисления в GPSS/PC выполняются с использованием арифметических переменных. Они могут быть целыми и действительными. Целые переменные определяются перед началом моделирования с помощью оператора определения VARIABLE (переменная), имеющего следующий формат:

имя VARIABLE выражение

Здесь имя - имя переменной, используемое для ссылок на нее, а выражение - арифметическое выражение, определяющее переменную. Арифметическое выражение представляет собой комбинацию операндов, в качестве которых могут выступать константы, СЧА и функции, знаков арифметических операций и круглых скобок. Следует заметить, что знаком операции умножения в GPSS/PC является символ # (номер). Результат каждой промежуточной операции в целых переменных преобразуется к целому типу путем отбрасывания дробной части, и, таким образом, результатом операции деления является целая часть частного.

Действительные переменные определяются перед началом моделирования с помощью оператора определения FVARIABLE, имеющего тот же формат, что и оператор VARIABLE. Отличие действительных переменных от целых заключается в том, что в действительных переменных все промежуточные операции выполняются с сохранением дробной части чисел, и лишь окончательный результат приводится к целому типу отбрасыванием дробной части.

Арифметические переменные обоих типов имеют единственный СЧА с названием V, значением которого является результат вычисления арифметического выражения, определяющего переменную. Вычисление выражения производится при входе транзакта в блок, содержащий ссылку на СЧА V с именем переменной.

Действительные переменные могут быть использованы для получения случайных интервалов времени с показательным законом распределения. Пусть в модели из примера на рис. 3 распределения времени поступления транзактов и времени задержки должны иметь показательный закон. Это может быть сделано так, как показано на рис. 4.

```
TARR FVARIABLE-100#LOG((1+RN1)/1000)
TSRV FVARIABLE -80#LOG((1+RN1)/1000)
GENERATE V$TARR
ADVANCE V$TSRV
TERMINATE 1
```

Рис. 4

Переменная с именем TARR задает выражение для вычисления интервала поступления со средним значением 100, вторая переменная с именем TSRV - для вычисления времени задержки со средним значением 80. Блоки GENERATE и ADVANCE содержат в поле А ссылки на соответствующие переменные, при этом поле В не используется, так как в поле А содержится случайная величина, не нуждающаяся в модификации.

Большинство случайных величин не может быть получено через случайную величину R с помощью арифметического выражения. Кроме того, такой способ является достаточно трудоемким, так как требует обращения к математическим функциям, вычисление которых требует десятков машинных операций. Другим возможным способом является использование вычислительных объектов GPSS/PC типа функция.

Функции используются для вычисления величин, заданных табличными зависимостями. Каждая функция определяется перед началом моделирования с помощью оператора определения FUNCTION (функция), имеющего следующий формат:

имя FUNCTION A,B

Здесь имя - имя функции, используемое для ссылок на нее; А - стандартный числовой атрибут, являющийся аргументом функции; В - тип функции и число точек таблицы, определяющей функцию.

Существует пять типов функций. Рассмотрим вначале непрерывные числовые функции, тип которых кодируется буквой С. Так, например, в определении непрерывной числовой функции, таблица которой содержит 24 точки, поле В должно иметь значение С24.

При использовании непрерывной функции для генерирования случайных чисел ее аргументом должен быть один из генераторов случайных чисел RNj. Так, оператор для определения функции показательного распределения может иметь следующий вид:

EXP FUNCTION RN1,C24

Особенностью использования встроенных генераторов случайных чисел RNj в качестве аргументов функций является то, что их значения в этом контексте интерпретируются как дробные числа от 0 до 0,999999.

Таблица с координатами точек функции располагается в строках, следующих непосредственно за оператором FUNCTION. Эти строки не должны иметь поля нумерации. Каждая точка таблицы задается парой Xi (значение аргумента) и Yi (значение функции), отделяемых друг от друга запятой. Пары координат отделяются друг от друга символом "/" и располагаются на произвольном количестве строк. Последовательность значений аргумента Xi должна быть строго возрастающей.

Как уже отмечалось, при использовании функции в поле В блоков GENERATE и ADVANCE вычисление интервала поступления или времени задержки производится путем умножения операнда А на вычисленное значение функции. Отсюда следует, что функция, используемая для генерирования случайных чисел с показательным распределением, должна описывать зависимость y=-ln(x), представленную в табличном виде. Оператор FUNCTION с такой таблицей, содержащей 24 точки для обеспечения достаточной точности аппроксимации, имеет следующий вид:

```
EXP FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915 .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9
.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
```

Вычисление непрерывной функции производится следующим образом. Сначала определяется интервал (Xi;Xi+1), на котором находится текущее значение СЧА-аргумента (в нашем примере - сгенерированное значение RN1). Затем на этом интервале выполняется линейная интерполяция с использованием соответствующих значений Yi и Yi+1. Результат интерполяции усекается (отбрасыванием дробной части) и используется в качестве значения функции. Если функция служит операндом В блоков GENERATE или ADVANCE, то усечение результата производится только после его умножения на значение операнда А.

Использование функций для получения случайных чисел с заданным распределением дает хотя и менее точный результат за счет погрешностей аппроксимации, но зато с меньшими вычислительными затратами (несколько машинных операций на выполнение линейной интерполяции). Чтобы к погрешности аппроксимации не добавлять слишком большую погрешность усечения, среднее значение при использовании показательных распределений должно быть достаточно большим (не менее 50). Эта рекомендация относится и к использованию переменных.

Функции всех типов имеют единственный СЧА с названием FN, значением которого является вычисленное значение функции. Вычисление функции производится при входе транзакта в блок, содержащий ссылку на СЧА FN с именем функции.

Заменим в примере на рис. 4 переменные TARR и TSRV на функцию EXP (рис. 5).

Поскольку в обеих моделях используется один и тот же генератор RN1, интервалы поступления и задержки, вычисляемые в блоках GENERATE и ADVANCE, должны получиться весьма близкими, а может быть и идентичными. При большом количестве транзактов, пропускаемых через модель (десятки и сотни тысяч), разница в скорости вычислений должна стать заметной.

```
EXP FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915 .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE 100,FN$EXP
ADVANCE 80,FN$EXP
TERMINATE 1
```

Рис. 5

Особенностью непрерывных функций является то, что они принимают "непрерывные" (но только целочисленные) значения в диапазоне от Y1 до Yn , где n - количество точек таблицы. В отличие от них дискретные числовые функции, тип которых кодируется буквой D в операнде В оператора определения функции, принимают только отдельные (дискретные) значения, заданные координатами Yi в строках, следующих за оператором определения FUNCTION. При вычислении дискретной функции текущее значение СЧА-аргумента, указанного в поле А оператора FUNCTION, сравнивается по условию <= последовательно со всеми значениями упорядоченных по возрастанию координат Xi до выполнения этого условия при некотором i. Значением функции становится целая часть соответствующего значения Yi.

Если последовательность значений аргумента таблицы с координатами точек функции представляет числа натурального ряда (1,2,3,...,n), то такую дискретную функцию с целью экономии памяти и машинного времени удобно определить как списковую числовую функцию (тип L).

Пусть в модели на рис. 5 заявки, моделируемые транзактами, с равной вероятностью 1/3 должны относиться к одному из трех классов (типов) 1,2 и 3, а среднее время задержки обслуживания заявок каждого типа должно составлять соответственно 70, 80 и 90 единиц модельного времени. Это может быть обеспечено способом, показанным на рис. 6.

В блоке ASSIGN в параметр TYPE каждого сгенерированного транзакта заносится тип заявки, получаемый с помощью дискретной функции CLASS. Аргументом функции является генератор случайных чисел RN1, а координаты ее таблицы представляют собой обратную функцию распределения дискретной случайной величины "класс заявки" с одинаковыми вероятностями каждого из трех значений случайной величины.

Поле А блока ADVANCE содержит ссылку на списковую функцию MEAN, аргументом которой служит параметр TYPE входящих в блок транзактов. В зависимости от значений этого параметра (типа заявки) среднее время задержки принимает одно из трех возможных значений функции MEAN: 70, 80 или 90 единиц.

```
EXP FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915 .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
CLASS FUNCTION RN1,D3
333,1/.667,2/1,3
MEAN FUNCTION P$TYPE,L3
1,70/2,80/3,90
GENERATE 100, FN$EXP
ASSIGN TYPE, FN$CLASS
ADVANCE FN$MEAN, FN$EXP
TERMINATE 1
```

Рис. 6

Следует отметить, что в данном примере можно было бы не использовать параметр TYPE и обойтись одной дискретной функцией, возвращающей с равной вероятностью одно из трех возможных значений среднего времени задержки. Однако использование параметров дает некоторые дополнительные возможности, которые будут рассмотрены позже.

Транзакты могут входить в модель не только через блок GENERATE, но и путем создания копий уже существующих транзактов в блоке SPLIT (расщепить), имеющем следующий формат:

имя SPLIT A,B,C

В поле А задается число создаваемых копий исходного транзакта (родителя), входящего в блок SPLIT. После выхода из блока SPLIT транзакт-родитель направляется в следующий блок, а все транзакты-потомки поступают в блок, указанный в поле В. Если поле В пусто, то все копии поступают в следующий блок.

Транзакт-родитель и его потомки, выходящие из блока SPLIT, могут быть пронумерованы в параметре, имя или номер которого указаны в поле С. Если у транзакта-родителя значение этого параметра при входе в блок SPLIT было равно k, то при выходе из блока оно станет равным k+1, а значения этого параметра у транзактов-потомков окажутся равными k+2, k+3 и т.д.

Например, блок SPLIT 5,MET1,NUM

создает пять копий исходного транзакта и направляет их в блок с именем MET1. Транзакт-родитель и потомки нумеруются в параметре с именем NUM. Если, например, перед входом в блок значение этого параметра у транзакта-родителя было равно 0, то при выходе из блока оно станет равным 1, а у транзактов-потомков значения параметра NUM будут равны 2, 3, 4, 5 и 6.

5.4. Блоки, связанные с аппаратными объектами

Все примеры моделей, рассматривавшиеся выше, пока еще не являются моделями систем массового обслуживания, так как в них не учтена основная особенность СМО: конкуренция заявок на использование некоторых ограниченных ресурсов системы. Все транзакты, входящие в эти модели через блок GENERATE, немедленно получают возможность "обслуживания" в блоке ADVANCE, который никогда не "отказывает" транзактам во входе, сколько бы транзактов в нем не находилось.

Для моделирования ограниченных ресурсов СМО в модели должны присутствовать аппаратные объекты: одноканальные или многоканальные устройства. Одноканальные устройства создаются в текущей модели при использовании блоков SEIZE (занять) и RELEASE (освободить), имеющих следующий формат:

имя SEIZE A

имя RELEASE A

В поле А указывается номер или имя устройства. Если транзакт входит в блок SEIZE, то устройство, указанное в поле А, становится занятым и остаётся в этом состоянии до тех пор, пока этот же транзакт не пройдёт соответствующий блок RELEASE, освобождая устройство. Если устройство, указанное в поле А блока SEIZE, уже занято каким-либо транзактом, то никакой другой транзакт не может войти в этот блок и остаётся в предыдущем блоке. Транзакты, задержанные (заблокированные) перед блоком SEIZE, остаются в списке текущих событий и при освобождении устройства обрабатываются с учетом приоритетов и очередности поступления.

Каждое устройство имеет следующие СЧА: F - состояние устройства (0 - свободно, 1 - занято); FR - коэффициент использования в долях 1000; FC - число занятий устройства; FT - целая часть среднего времени занятия устройства.

Воспользуемся блоками SEIZE и RELEASE для моделирования одноканальной СМО с ожиданием (рис. 7). Теперь блок ADVANCE находится между блоками SEIZE и RELEASE, моделирующими занятие и освобождение устройства с именем SYSTEM, и поэтому в нем может находиться только один транзакт. Транзакты, выходящие из блока GENERATE в моменты занятости устройства, не смогут войти в блок SEIZE и будут оставаться в блоке GENERATE, образуя очередь в списке текущих событий.

```
EXP FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915      .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE 100,FN$EXP
SEIZE SYSTEM
ADVANCE 80,FN$EXP
RELEASE SYSTEM
TERMINATE 1
```

Рис. 7

Для моделирования захвата (прерывания) одноканального устройства вместо блоков SEIZE и RELEASE используются соответственно блоки PREEMPT (захватить) и RETURN (вернуть). Блок PREEMPT имеет следующий формат:

имя PREEMPT A,B,C,D,E

В поле А указывается имя или номер устройства, подлежащего захвату. В поле В кодируется условие захвата. Если это поле пусто, то захват возникает, если обслуживаемый транзакт сам не является захватчиком. Если же в поле В записан операнд PR, то захват возникает, если приоритет транзакта-захватчика выше, чем приоритет обслуживаемого транзакта.

Поля C, D и E определяют поведение транзактов, обслуживание которых было прервано. Поле C указывает имя блока, в который будет направлен прерванный транзакт. В поле D может быть указан номер или имя параметра прерванного транзакта, в который записывается время, оставшееся этому транзакту до завершения обслуживания на устройстве. При отсутствии операнда в поле E прерванный транзакт сохраняет право на автоматическое восстановление на устройстве по окончании захвата. Если же в поле E указан операнд RE, то транзакт теряет такое право.

Блок RETURN имеет единственный операнд А, содержащий имя или номер устройства, подлежащего освобождению от захвата.

Блоки PREEMPT и RETURN могут быть использованы для моделирования СМО с абсолютными приоритетами. В простейших случаях, при одном уровне захвата, в блоке PREEMPT используется единственный операнд А. При этом прерванный транзакт переводится симулятором из списка будущих событий в так называемый список прерываний устройства, а по окончании захвата устройства возвращается в список будущих событий с предварительно вычисленным временем занятия устройства для продолжения обслуживания.

Для создания в модели многоканальных устройств (МКУ) они должны быть предварительно определены с помощью операторов определения STORAGE (память), имеющих следующий формат:

имя STORAGE A

Здесь имя - имя МКУ, используемое для ссылок на него; A - емкость (количество каналов обслуживания) МКУ, задаваемая константой.

Для занятия и освобождения каналов обслуживания МКУ используется пара блоков ENTER (войти) и LEAVE (покинуть), имеющих следующий формат:

имя ENTER A,B

имя LEAVE A,B

В поле А указывается номер или имя МКУ, в поле В - число каналов МКУ, занимаемых при входе в блок ENTER или освобождаемых при входе в блок LEAVE. Обычно поле В пусто, и в этом случае по умолчанию занимается или освобождается один канал.

При входе транзакта в блок ENTER текущее содержимое МКУ увеличивается на число единиц, указанное в поле В. Если свободная емкость МКУ меньше значения поля В, то транзакт не может войти в блок ENTER и остается в предыдущем блоке, образуя очередь в списке текущих событий.

При входе транзакта в блок LEAVE текущее содержимое МКУ уменьшается на число единиц, указанное в поле В. Не обязательно освобождается такое же число каналов МКУ, какое занималось при входе данного транзакта в блок ENTER, однако текущее содержимое МКУ не должно становиться отрицательным.

Многоканальные устройства имеют следующие СЧА: S - текущее содержимое МКУ; R - свободная емкость МКУ; SR - коэффициент использования в долях 1000; SA - целая часть среднего содержимого МКУ; SM - максимальное содержимое МКУ; SC - число занятий МКУ; ST - целая часть среднего времени занятия МКУ.

Воспользуемся блоками ENTER-LEAVE и оператором STORAGE для моделирования двухканальной СМО с ожиданием (рис. 8). Если текущее содержимое МКУ с именем STO2 меньше 2, т.е. в блоке ADVANCE находится один или ни одного транзакта, то очередной транзакт, поступающий в модель через блок GENERATE, может войти в блок ENTER и затем в блок ADVANCE. Если же текущее содержимое МКУ равно 2, то очередной транзакт остается в блоке GENERATE, образуя очередь в списке текущих событий. По истечении задержки одного из двух обслуживаемых транзактов в блоке ADVANCE и после входа его в блок LEAVE первый из заблокированных транзактов сможет войти в блок ENTER.

```
STO2 STORAGE 2
EXP FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915 .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE 100,FN$EXP
ENTER STO2
ADVANCE 160,FN$EXP
LEAVE STO2
TERMINATE 1
```

Рис. 8

К аппаратным объектам относятся также логические переключатели (ЛП), которые могут находиться в двух состояниях: "включено" и "выключено". В начале моделирования все ЛП находятся в состоянии "выключено". Отдельные переключатели могут быть установлены в начальное состояние "включено" с помощью оператора INITIAL (инициализировать), имеющего следующий формат:

INITIAL LS$имя

INITIAL LSj

Здесь имя и j - соответственно имя и номер ЛП, устанавливаемого в начальное состояние "включено".

Для включения, выключения и инвертирования логических переключателей в процессе моделирования служит блок LOGIC (установить ЛП), имеющий следующий формат:

имя LOGIC X A

В поле А указывается имя или номер ЛП. Вспомогательный операнд X указывает вид операции, которая производится с логическим переключателем при входе транзакта в блок: S - включение, R - выключение, I - инвертирование. Например:

LOGIC S 9

LOGIC R FLAG

Логические переключатели имеют единственный СЧА с названием LS. Значение СЧА равно 1, если ЛП включен, и 0, если он выключен.

5.5. Блоки для сбора статистических данных

Два последних примера в предыдущем параграфе представляют собой законченные модели одноканальной и многоканальной СМО с ожиданием. Однако такие модели разрабатываются обычно для исследования различных характеристик, связанных с ожиданием заявок в очереди: длины очереди, времени ожидания и т.п., а в приведенных примерах очередь транзактов образуется в списке текущих событий и недоступна исследователю. Для регистрации статистической информации о процессе ожидания транзактов в модели должны присутствовать статистические объекты: очереди или таблицы.

Объекты типа очередь создаются в модели путем использования блоков - регистраторов очередей: QUEUE (стать в очередь) и DEPART (уйти из очереди), имеющих следующий формат:

имя QUEUE A,B

имя DEPART A,B

В поле А указывается номер или имя очереди, а в поле В – число единиц, на которое текущая длина очереди увеличивается при входе транзакта в блок QUEUE или уменьшается при входе транзакта в блок DEPART. Обычно поле В пусто, и в этом случае его значение по умолчанию принимается равным 1.

Для сбора статистики о транзактах, заблокированных перед каким-либо блоком модели, блоки QUEUE и DEPART помещаются перед и после этого блока соответственно. При прохождении транзактов через блоки QUEUE и DEPART соответствующим образом изменяются следующие СЧА очередей: Q - текущая длина очереди; QM - максимальная длина очереди; QA - целая часть средней длины очереди; QC - общее число транзактов, вошедших в очередь; QZ - число транзактов, прошедших через очередь без ожидания (число "нулевых" входов); QT - целая часть среднего времени ожидания с учетом "нулевых" входов; QX - целая часть среднего времени ожидания без учета "нулевых" входов.

Дополним приведенную на рис. 7 модель одноканальной СМО блоками QUEUE и DEPART (рис. 9). Теперь транзакты, заблокированные перед блоком SEIZE из-за занятости устройства SYSTEM, находятся в блоке QUEUE, внося свой вклад в статистику о времени ожидания, накапливаемую в статистическом объекте типа "очередь" с именем LINE. При освобождении устройства первый из заблокированных транзактов войдет в блок SEIZE и одновременно в блок DEPART, прекращая накопление статистики об ожидании этого транзакта.

```
EXP FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915 .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3 .92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE 100,FN$EXP
QUEUE LINE
SEIZE SYSTEM
DEPART LINE
ADVANCE 80,FN$EXP
RELEASE SYSTEM
TERMINATE 1
```

Рис. 9

Очень часто исследователя интересует не только среднее значение времени ожидания в очереди, но и дисперсия этого времени, а также статистическое распределение выборки времени ожидания, представляемое обычно графически в виде гистограммы. Имея такое распределение, можно оценить вероятность того, что время ожидания превысит или не превысит некоторое заданное значение. Для сбора и обработки данных о выборочном распределении времени ожидания в очереди служат статистические объекты типа Q-таблица.

Для создания в модели такой таблицы она должна быть предварительно определена с помощью оператора определения QTABLE (Q-таблица), имеющего следующий формат:

имя QTABLE A,B,C,D

Здесь имя - имя таблицы, используемое для ссылок на нее; A - номер или имя очереди, распределение времени ожидания в которой необходимо получить; B - верхняя граница первого частотного интервала таблицы; C - ширина частотных интервалов; D - количество частотных интервалов.

Диапазон всевозможных значений времени ожидания в очереди, указанной в поле A, разбивается на ряд частотных интервалов, количество которых указано в поле D. Первый из этих интервалов имеет ширину от минус бесконечности до величины, указанной в поле B, включительно. Второй интервал включает значения, большие, чем величина первой границы в поле B, но меньшие или равные B+C, и т.д. Все промежуточные интервалы имеют одинаковую ширину, указанную в поле C. Наконец, последний интервал включает все значения, большие, чем последняя граница. Значения операндов B, C и D должны задаваться целыми константами. Операнд B может быть неположительным, хотя для Q-таблицы это не имеет смысла, так как время не может быть отрицательным. Операнды C и D должны быть строго положительными.

При прохождении транзакта через блоки QUEUE и DEPART его время ожидания фиксируется, и к счетчику частотного интервала таблицы, в который попало это время, добавляется 1. Одновременно в таблице накапливается информация для вычисления среднего значения и среднеквадратического отклонения (корня из дисперсии) времени ожидания. По окончании моделирования среднее значение и среднеквадратическое отклонение времени ожидания, а также счетчики попаданий в различные частотные интервалы выводятся в стандартный отчет GPSS/PC.

Таблицы, как и другие объекты GPSS/PC, имеют СЧА: TC - общее число транзактов, вошедших в очередь, связанную с таблицей; TB - целая часть среднего времени ожидания в очереди; TD - целая часть среднеквадратического отклонения времени ожидания в очереди.

Дополним модель из примера на рис. 9 оператором QTABLE для получения распределения времени ожидания в очереди с именем LINE (рис. 10).

```
WTIME QTABLE LINE,50,50,10
EXP FUNCTION RN1,C24
```

```
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915 .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE 100,FN$EXP
QUEUE LINE
SEIZE SYSTEM
DEPART LINE
ADVANCE 80,FN$EXP
RELEASE SYSTEM
TERMINATE 1
```

Рис. 10

Оператор определения таблицы с именем WTIME разбивает ось времени на 10 частотных интервалов. Первый интервал включает значения от 0 до 50, второй - от 50 до 100, третий - от 100 до 150 и т.д. Последний, десятый, интервал включает значения, превышающие 450. Если, например, время ожидания некоторого транзакта в очереди составило 145 единиц модельного времени, то к счетчику третьего частотного интервала будет добавлена 1. Следует заметить, что информация в таблицу с именем WTIME заносится автоматически, при входе транзактов в блоки QUEUE и DEPART, и никаких специальных мер для этого принимать не требуется.

Таблицы в GPSS/PC могут использоваться в более общем случае не только для табулирования времени ожидания в очереди, но и для получения выборочных распределений произвольных СЧА любых объектов модели. Для определения таблиц служит оператор TABLE (таблица), формат которого совпадает с форматом оператора QTABLE. Отличие состоит лишь в том, что в поле A оператора TABLE записывается стандартный числовой атрибут, выборочное распределение которого необходимо получить, а операнды B, C и D определяют разбиение на частотные интервалы диапазона всевозможных значений этого СЧА.

Занесение информации в таблицу, определяемую оператором TABLE, уже не может быть выполнено симулятором автоматически, как в случае Q-таблиц. Для этого используется специальный блок TABULATE (табулировать), имеющий следующий формат:

имя TABULATE A

В поле A указывается номер или имя таблицы, определенной соответствующим оператором TABLE. При входе транзакта в блок TABULATE текущее значение табулируемого аргумента таблицы, указанного в поле A оператора TABLE, заносится в нее в соответствии с заданным в операторе TABLE разбиением области значений аргумента на частотные интервалы. Одновременно корректируются текущие значения СЧА таблицы: счетчик входов в таблицу TC, среднее время ожидания TB и среднеквадратическое отклонение времени ожидания TD.

Пусть, например, в модели многоканальной СМО, приведенной на рис. 8, надо получить распределение времени пребывания заявок в системе, включающего время ожидания в очереди и время обслуживания. Это может быть обеспечено способом, показанным на рис. 11.

Оператор TABLE определяет таблицу с именем TTIME, аргументом которой служит СЧА M1 - время пребывания транзакта в модели. В рассматриваемой модели значение СЧА M1 одновременно будет являться временем пребывания транзакта в СМО в том случае, если занесение информации в таблицу производить перед выходом транзакта из модели. Поэтому блок TABULATE, заносящий информацию о времени пребывания каждого транзакта в модели в таблицу TTIME, располагается перед блоком TERMINATE. Диапазон возможных значений времени пребывания транзакта в модели разбит в операторе TABLE на 12 частотных интервалов, ширина которых (кроме последнего) равна 100 единицам модельного времени.

```
TTIME TABLE M1,100,100,12
STO2 STORAGE 2 EXP FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915 .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
```

```
GENERATE 100,FN$EXP
ENTER STO2
ADVANCE 160,FN$EXP
LEAVE STO2
TABULATE TTIME
TERMINATE 1
```

Рис. 11

5.6. Блоки, изменяющие маршруты транзактов

В приведенных выше примерах транзакты, выходящие из любого блока, всегда поступали в следующий блок. В более сложных моделях возникает необходимость направления транзактов к другим блокам в зависимости от некоторых условий. Эту возможность обеспечивают блоки изменения маршрутов транзактов.

Блок TRANSFER (передать) служит для передачи входящих в него транзактов в блоки, отличные от следующего. Блок имеет девять режимов работы, из которых рассмотрим здесь лишь три наиболее часто используемых. В этих трех режимах блок имеет следующий формат:

имя TRANSFER A,B,C

Смысл операндов в полях A, B и C зависит от режима работы блока.

В режиме безусловной передачи поля A и C пусты, а в поле B указывается имя блока, к которому безусловным образом направляется транзакт, вошедший в блок TRANSFER. Например:

TRANSFER ,FINAL

В режиме статистической передачи операнд A определяет вероятность, с которой транзакт направляется в блок, указанный в поле C. С вероятностью 1-A транзакт направляется в блок, указанный в поле B (в следующий, если поле B пусто).

Вероятность в поле A может быть задана непосредственно десятичной дробью, начинающейся с точки. Например, блок

TRANSFER .75,THIS,THAT

с вероятностью 0,75 направляет транзакты в блок с именем THAT, а с вероятностью 0,25 - в блок с именем THIS.

Если же поле A начинается не с десятичной точки и не содержит одного из ключевых слов - признаков других режимов работы блока, то его значение рассматривается как количество тысячных долей в вероятности передачи. Например, предыдущий блок TRANSFER можно записать также в следующем виде:

TRANSFER 750,THIS,THAT

В режиме логической передачи в поле A записывается ключевое слово BOTH (оба). Транзакт, поступающий в блок TRANSFER, сначала пытается войти в блок, указанный в поле B (или в следующий блок, если поле B пусто), а если это не удается, т.е. блок B отказывает транзакту во входе, то в блок, указанный в поле C. Если и эта попытка неудачна, то транзакт задерживается в блоке TRANSFER до изменения условий в модели, делающего возможным вход в один из блоков B или C, причем при одновременно возникшей возможности предпочтение отдается блоку B. Например:

TRANSFER BOTH,MET1,MET2

Блок TEST (проверить) служит для задержки или изменения маршрутов транзактов в зависимости от соотношения двух СЧА. Он имеет следующий формат:

имя TEST X A,B,C

Вспомогательный операнд X содержит условие проверки соотношения между СЧА и может принимать следующие значения: L (меньше); LE (меньше или равно); E (равно); NE (не равно); GE (больше или равно); G (больше). Поле A содержит первый, а поле B - второй из сравниваемых СЧА. Если проверяемое условие A X B выполняется, то блок TEST пропускает транзакт в следующий блок. Если же это условие не выполняется, то транзакт переходит к блоку, указанному в поле C, а если оно пусто, то задерживается перед блоком TEST.

Например, блок

TEST LE P$TIME,C1

не впускает транзакты, у которых значение параметра с именем TIME больше текущего модельного времени. Блок

TEST L Q$LINE,5,OUT

направляет транзакты в блок с именем OUT, если текущая длина очереди LINE больше либо равна 5.

Для задержки или изменения маршрута транзактов в зависимости от состояния аппаратных объектов модели служит блок GATE (впустить), имеющий следующий формат:

имя GATE X A,B Вспомогательный операнд X содержит код состояния проверяемого аппаратного объекта, а в поле A указывается имя или номер этого объекта. Если проверяемый объект находится в заданном состоянии, то блок GATE пропускает транзакт к следующему блоку. Если же заданное в блоке условие не выполняется, то транзакт переходит к блоку, указанному в поле B, а если это поле пусто, то задерживается перед блоком GATE.

Операнд X может принимать следующие значения: U (устройство занято); NU (устройство свободно); I (устройство захвачено); NI (устройство не захвачено); SE (МКУ пусто); SNE (МКУ не пусто); SF (МКУ заполнено); SNF (МКУ не заполнено); LS (ЛП включен), LR (ЛП выключен).

Например, блок GATE SNE BUF3

отказывает во входе транзактам, поступающим в моменты, когда в МКУ с именем BUF3 все каналы обслуживания свободны. Блок

GATE LR 4,BLOK2 направляет транзакты в блок с именем BLOK2, если в момент их поступления ЛП с номером 4 включен.

Блоки рассматриваемой группы используются при моделировании различных СМО с потерями заявок. Воспользуемся, например, блоками TRANSFER для моделирования двухканальной СМО с отказами и повторными попытками (рис. 12).

```
STO2  STORAGE 2
EXP   FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915    .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE 100,FN$EXP
ENT1 TRANSFER BOTH,,REFUS
ENTER STO2
ADVANCE 160,FN$EXP
LEAVE STO2
TERMINATE 1
REFUS TRANSFER .1,,OUT
 ADVANCE 250,FN$EXP
 TRANSFER ,ENT1
OUT   TERMINATE 1
```

Рис. 12

Транзакты, поступающие в модель, попадают в блок TRANSFER с именем ENT1, работающий в логическом режиме. Если в момент поступления транзакта в МКУ STO2 хотя бы один канал свободен, то блок TRANSFER направит транзакт в следующий блок, т.е. в блок ENTER. Если же в момент поступления оба канала МКУ заняты, и поэтому блок ENTER отказывает во входе, то транзакт будет направлен в блок TRANSFER с именем REFUS, работающий в статистическом режиме. С вероятностью 0,9 транзакты из этого блока передаются в следующий блок, задерживаются в нем на случайное время и с помощью блока TRANSFER, работающего в безусловном режиме, передаются вновь на вход модели в блок с именем ENT1. С вероятностью 0,1 транзакты из блока с именем REFUS передаются в блок TERMINATE с именем OUT для уничтожения.

Следует заметить, что для уничтожения транзактов, получивших отказ в обслуживании, понадобился отдельный блок TERMINATE для фиксации в стандартном отчете количества потерянных транзактов с помощью счетчика блока с именем OUT (СЧА N$OUT).

Для моделирования той же СМО может быть использован также блок TEST (рис. 13). В этом варианте модели транзакт проходит в блок ENTER, если текущее число занятых каналов (СЧА S$STO2) меньше 2.

```
STO2 STORAGE 2
EXP    FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915          .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE 100,FN$EXP
ENT1 TEST L S$STO2,2,REFUS
ENTER STO2
ADVANCE 160,FN$EXP
LEAVE STO2
TERMINATE 1
REFUS TRANSFER .1,,OUT
ADVANCE 250,FN$EXP
TRANSFER ,ENT1
OUT    TERMINATE 1
```

Рис. 13

При использовании блока GATE модель принимает вид, показанный на рис. 14. В этом варианте транзакт проходит в блок ENTER, если условие "МКУ STO2 не заполнено" истинно.

```
STO2 STORAGE 2 EXP FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915          .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE 100,FN$EXP
ENT1 GATE SNF STO2,REFUS
ENTER STO2
ADVANCE 160,FN$EXP
LEAVE STO2
TERMINATE 1
REFUS TRANSFER .1,,OUT
ADVANCE 250,FN$EXP
TRANSFER ,ENT1
OUT    TERMINATE 1
```

Рис. 14

5.7. Блоки, работающие с памятью

Для хранения в памяти отдельных числовых значений и массивов таких значений используются сохраняемые величины и матрицы сохраняемых величин.

Сохраняемые величины могут использоваться в модели для хранения исходных данных, которые надо изменять при различных прогонах модели, промежуточных значений и результатов моделирования. В начале моделирования все сохраняемые величины устанавливаются равными 0. Для установки отличных от 0 начальных значений сохраняемых величин используется оператор INITIAL, имеющий следующий формат:

INITIAL X$имя, значение

INITIAL Xj, значение

Здесь имя и j - соответственно имя и номер сохраняемой величины, а значение - присваиваемое ей начальное значение (константа).

Для изменения сохраняемых величин в процессе моделирования служит блок SAVEVALUE (сохранить величину), имеющий следующий формат:

имя SAVEVALUE A,B

В поле A указывается номер или имя сохраняемой величины, в которую записывается значение операнда B. Если в поле A после имени (номера) сохраняемой величины стоит знак + или -, то значение операнда B добавляется или вычитается из текущего содержимого сохраняемой величины. Например:

SAVEVALUE 5,Q$LINE

SAVEVALUE NREF+,1

Сохраняемые величины имеют единственный СЧА с названием X, значением которого является текущее значение соответствующей сохраняемой величины.

Изменим пример на рис. 14 таким образом, чтобы исходные данные модели (средний интервал поступления транзактов и среднее время обслуживания) были заданы сохраняемыми величинами, а результат моделирования (количество потерянных транзактов) фиксировался также в сохраняемой величине. Такая модель будет иметь вид, показанный на рис. 15.

Матрицы сохраняемых величин дают возможность упорядочить сохраняемые значения в виде матриц m*n, где m - число строк, n - число столбцов матрицы. Каждая матрица должна быть перед началом моделирования определена с помощью оператора MATRIX (определить матрицу), имеющего следующий формат:

имя MATRIX A,B,C Поле A оператора не используется и сохранено в GPSS/PC для совместимости со старыми версиями GPSS. В полях B и C указываются соответственно число строк и столбцов матрицы, задаваемые константами, причем общее число элементов, равное произведению B на C, не должно превышать 8191. Например, оператор

MTAB MATRIX ,10,2 определяет матрицу с именем MTAB, содержащую десять строк и два столбца.

```
INITIAL X$TARR,100
INITIAL X$TSRV,160
STO2 STORAGE 2
EXP FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915     .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE X$TARR,FN$EXP
ENT1 GATE SNF STO2,REFUS
ENTER STO2
ADVANCE X$TSRV,FN$EXP
LEAVE STO2
OUT   TERMINATE 1
REFUS TRANSFER .1,,COUT
ADVANCE 250,FN$EXP
TRANSFER ,ENT1
COUT SAVEVALUE NREF+,1
TRANSFER ,OUT
```

Рис. 15

В начале моделирования элементы всех определенных матриц устанавливаются равными 0. Для установки отличных от 0 начальных значений отдельных элементов матриц используется оператор INITIAL, имеющий следующий формат:

INITIAL MX$имя(a,b),значение

INITIAL MXj(a,b),значение

Здесь имя и j - соответственно имя и номер матрицы; a и b - номера соответственно строки и столбца, задаваемые константами; значение - присваиваемое элементу матрицы начальное значение, задаваемое также константой.

Для изменения значений элементов матриц в процессе моделирования служит блок MSAVEVALUE (сохранить значение элемента матрицы), имеющий следующий формат:

имя MSAVEVALUE A,B,C,D

В поле A указывается имя или номер матрицы, после которого, как и в блоке SAVEVALUE, может стоять знак + или -. В полях B и C указываются номера соответственно строки и столбца, определяющие изменяемый элемент матрицы. В поле D указывается величина, используемая для изменения заданного элемента матрицы. Например:

MSAVEVALUE 5,3,2,X1

MSAVEVALUE MTAB+,PROW,PCOL,1

Матрицы имеют единственный СЧА с названием MX, ссылка на который записывается в следующем виде: MX$имя(a,b)

MXj(a,b)

Здесь имя и j - соответственно имя и номер матрицы; a и b - номера соответственно строки и столбца, задаваемые константами или ссылками на СЧА параметров транзактов. Например:

MX5(2,1)

MX$MTAB(P$ROW,P$COL)

5.8. Блоки для работы со списками пользователя

Так как заблокированные транзакты находятся в списке текущих событий, то при большом количестве таких транзактов симулятор расходует слишком много времени на просмотр этого списка с целью выбора очередного транзакта для продвижения. Для экономии машинного времени заблокированные транзакты целесообразно помещать в так называемые списки пользователя и оставлять их там до тех пор, пока не выполнятся условия, позволяюшие дальнейшее продвижение этих транзактов. Кроме того, размещение ожидающих транзактов в списках пользователя позволяет организовать различные дисциплины очередей, от личные от дисциплины "раньше пришел - раньше обслужен", реализованной в списке текущих событий.

Списки пользователя представляют собой некоторые буферы, куда могут временно помещаться транзакты, выведенные из списка текущих событий. В отличие от списков текущих и будущих событий транзакты вводятся в списки пользователя и выводятся из них не автоматически, а в соответствии с логикой модели с помощью специальных блоков.

Для ввода транзактов в список пользователя служит блок LINK (ввести в список), который может быть использован в двух режимах: условном и безусловном. Ограничимся рассмотрением лишь безусловного режима, в котором блок LINK имеет следующий формат:

имя LINK A,B

В поле A задается имя или номер списка пользователя, в который безусловным образом помещается транзакт, вошедший в блок. Поле B определяет, в какое место списка пользователя следует поместить этот транзакт. Если в поле B записано ключевое слово FIFO, то транзакт помещается в конец списка, если LIFO - в начало списка. В других случаях транзакты упорядочиваются в соответствии с вычисленным значением поля B, где обычно записывается один из СЧА транзактов, таких как PR, M1 или P. Если поле B содержит СЧА PR, то транзакты упорядочиваются по убыванию приоритета. В остальных случаях производится упорядочение по возрастанию указанного СЧА.

Например, блок

LINK 5,FIFO

помещает транзакты в список пользователя с номером 5 в порядке их поступления в блок. Блок LINK BUFER,P$ORDER помещает транзакты в список пользователя с именем BUFER, упорядочивая их по возрастанию параметра с именем ORDER.

Условия, при которых транзакт помещается в список пользователя, в безусловном режиме проверяются средствами, предусмотренными разработчиком модели. Например, направить транзакт в список пользователя в случае занятости устройства можно так, как показано на рис. 16. Если устройство с именем FAC4 занято, то блок GATE не впускает транзакт в блок SEIZE, а направляет его в блок LINK с именем WAIT, и транзакт вводится в конец списка пользователя с именем BUFER.

```
....................
GATE NU FAC4,WAIT
SEIZE FAC4
....................
WAIT LINK BUFER,FIFO
....................
```

Рис. 16

Для вывода одного или нескольких транзактов из списка пользователя и помещения их обратно в список текущих событий служит блок UNLINK (вывести из списка), имеющий следующий формат:

имя UNLINK X A,B,C,D,E,F

В поле A указывается имя или номер списка пользователя. Поле B содержит имя блока, в который переходят выведенные из списка пользователя транзакты. В поле C указывается число выводимых транзактов или ALL для вывода всех находящихся в списке транзактов.

Операнды в полях D и E вместе со вспомогательным операндом X определяют способ и условия вывода транзактов из списка пользователя. Если поля D и E пусты, то и операнд X не используется, а транзакты выводятся с начала списка пользователя. Если поле D содержит ключевое слово BACK, то поле E и вспомогательный операнд X не используются, а транзакты выводятся с конца списка. В остальных случаях значение поля D интерпретируется как номер параметра транзактов, находящихся в списке пользователя, а из списка выводится заданное число тех транзактов, у которых значение этого параметра по отношению к значению операнда в поле E удовлетворяет условию, заданному вспомогательным операндом X. Операнд X принимает те же значения, что и в блоке TEST.

В поле F указывается имя блока, куда переходит транзакт, выходящий из блока UNLINK, если из списка пользователя не выведен ни один транзакт. Если это поле пусто, то выводящий транзакт переходит в следующий блок независимо от количества выведенных транзактов.

Например, блок

UNLINK 5,NEXT,1

выводит из списка пользователя с номером 5 один транзакт с начала списка и направляет его в блок с именем NEXT. Блок

UNLINK BUFER,ENT1,1,BACK

выводит из списка пользователя с именем BUFER один транзакт с конца списка и направляет его в блок с именем ENT1. Блок

UNLINK E P$UCH,MET2,ALL,COND,P$COND,MET3

выводит из списка пользователя, номер которого записан в параметре UCH выводящего транзакта, и направляет в блок с именем MET2 все транзакты, содержимое параметра COND которых равно содержимому одноименного параметра выводящего транзакта. Если таких транзактов в списке не окажется, то выводящий транзакт будет направлен в блок с именем MET3, в противном случае - к следующему блоку.

Следует отметить следующие особенности выполнения блока UNLINK. Во-первых, если поля D и E содержат ссылки на СЧА транзактов, то поле D вычисляется относительно транзактов в списке пользователя, а поле E - относительно активного транзакта. Во-вторых, после вывода транзактов из списка симулятор продолжает или начинает продвижение транзакта с наивысшим приоритетом, а при равенстве приоритетов отдает предпочтение транзакту-инициатору вывода.

Каждый список пользователя имеет следующие СЧА: CH - текущая длина списка; CA - средняя длина списка (целая часть); CM - максимальная длина списка; CC - общее число транзактов, вошедших в список; CT - целая часть среднего времени пребывания транзакта в списке.

Воспользуемся рассмотренными блоками для моделирования многоканальной СМО с ожиданием транзактов в списке пользователя (рис. 17). Если МКУ с именем STO2 не заполнено, блок GATE впускает вновь прибывший транзакт в блок ENTER, и в МКУ занимается один канал. Если же МКУ заполнено, то блок GATE направляет транзакт в блок LINK с именем WAIT, помещающий транзакт в конец списка пользователя с именем BUFER, моделирующего очередь к МКУ. Каждый транзакт, покидающий МКУ по завершении обслуживания и освобождающий один канал, проходит блок UNLINK и выводит один транзакт с начала списка (если список не пуст), направляя его в блок с именем ENT1 на занятие канала в МКУ.

```
STO2 STORAGE 2
EXP FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915          .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE 100,FN$EXP
GATE SNF STO2,WAIT
ENT1 ENTER
STO2 ADVANCE 160,FN$EXP
LEAVE STO2
UNLINK BUFER,ENT1,1
TERMINATE 1
WAIT LINK BUFER,FIFO
```

Рис. 17

Заметим, что для изменения дисциплины обслуживания на "позже пришел - раньше обслужен" достаточно или заменить в поле B блока LINK FIFO на LIFO, или записать в поле D блока UNLINK операнд BACK. Следует также обратить внимание на то, что блоки QUEUE-DEPART для сбора статистики об ожидающих транзактах не используются, так как почти все те же данные можно получить из статистики о списке пользователя.

Рассмотрим еще один пример, иллюстрирующий использование списков пользователя для организации нестандартных дисциплин обслуживания. Пусть в одноканальной СМО с ожиданием требуется организовать такую дисциплину, при которой приоритет отдается заявкам с наименьшим временем обслуживания. Такая модель будет иметь вид, показанный на рис. 18.

В параметр TSRV поступающих в модель транзактов в блоке ASSIGN записывается случайное время обслуживания, вычисляемое с использованием функции EXP. Если устройство SYSTEM свободно, то блок GATE впускает транзакт в блок SEIZE, и устройство занимается на время P$TSRV. Если же в момент поступления транзакта устройство занято, то блок GATE направляет транзакт в блок LINK, который вводит транзакт в список пользователя LINE, упорядочивая транзакты по возрастанию времени обслуживания, записанного в параметре P$TSRV. Блок UNLINK по освобождении устройства выводит с начала списка транзакт с наименьшим временем обслуживания, обеспечивая тем самым заданную дисциплину.

```
EXP FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915          .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE 100,FN$EXP
ASSIGN TSRV,80,EXP
```

```
GATE NU
SYSTEM,WAIT
SFAC SEIZE SYSTEM
ADVANCE P$TSRV
RELEASE SYSTEM
UNLINK LINE,SFAC,1
TERMINATE 1
WAIT LINK LINE,P$TSRV
```

Рис. 18

3. УПРАВЛЯЮЩИЕ ОПЕРАТОРЫ GPSS/PC

Для управления прогоном модели используются управляющие операторы GPSS/PC. С одним из них - оператором START - мы уже сталкивались при рассмотрении блока TERMINATE. Оператор START (начать) имеет следующий формат:

START A,B,C,D

Поле A содержит константу, задающую начальное значение счетчика завершений. В поле B может быть записано ключевое слово NP - признак подавления формирования стандартного отчета по завершении моделирования. Если поле B пусто, то по окончании прогона модели формируется отчет со стандартной статистической информацией о всех объектах модели (см. разд. 5). Поле C не используется и сохранено для совместимости со старыми версиями GPSS. Поле D может содержать 1 для включения в отчет списков текущих и будущих событий. Если поле D пусто, то выдача в отчет содержимого этих списков не производится.

Оператор SIMULATE (моделировать) устанавливает предел реального времени, отводимого на прогон модели. Если прогон не завершится до истечения этого времени, то он будет прерван принудительно с выдачей накопленной статистики в отчет.

Оператор SIMULATE имеет единственный операнд A, содержащий предельное время моделирования в минутах, задаваемое константой. Оператор размещается перед оператором START, начинающим лимитированный прогон.

Оператор RMULT (установить значения генераторов) позволяет перед началом прогона установить начальные значения генераторов случайных чисел RN, определяющие генерируемые ими последовательности. Поля A-G оператора могут содержать начальные значения генераторов соответственно RN1-RN7, задаваемые константами. Начальные значения генераторов, не установленные операторами RMULT, совпадают с номерами генераторов.

Оператор RESET (сбросить) сбрасывает всю статистическую информацию, накопленную в процессе прогона модели. При этом состояние аппаратных, динамических и запоминающих объектов, а также генераторов случайных чисел сохраняется, и моделирование может быть возобновлено с повторным сбором статистики. Оператор не имеет операндов.

С оператором RESET связано различие между относительным (СЧА C1) и абсолютным (СЧА AC1) модельным временем. Таймер относительного времени C1 измеряет модельное время, прошедшее после последнего сброса статистики оператором RESET, а таймер абсолютного времени AC1 - модельное время, прошедшее после начала первого прогона модели. Если не использовалось ни одного оператора RESET, то значения этих таймеров совпадают. Оператор RESET устанавливает таймер C1 в ноль и не влияет на таймер AC1.

Оператор RESET используется обычно при моделировании нестационарных процессов, когда требуется собрать статистику по отдельным интервалам стационарности или исключить влияние переходного периода на собираемую статистическую информацию.

Пусть, например, в модели, приведенной на рис. 18, необходимо отбросить статистику, собираемую на первой тысяче транзактов. Это может быть сделано способом, показанным на рис. 19.

Первый оператор START начинает прогон модели длиной 1000 транзактов (переходный период). Поскольку статистика, накопленная на этом периоде, не используется, в поле В оператора указан признак подавления формирования отчета NP. Оператор RESET сбрасывает накопленную статистику, не изменяя состояния модели. Второй оператор START начинает основной прогон модели с формированием отчета по завершении прогона.

```
EXP FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915 .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE 100,FN$EXP
ASSIGN TSRV,80,EXP
GATE NU
SYSTEM,WAIT
SFAC SEIZE SYSTEM
ADVANCE P$TSRV
RELEASE SYSTEM
UNLINK LINE,SFAC,1
TERMINATE 1
WAIT LINK LINE,P$TSRV
START 1000,NP
RESET
START 10000
```

Рис. 19

Оператор CLEAR (очистить) очищает модель, подготавливая ее к повторному прогону. При этом сбрасывается вся накопленная в предыдущем прогоне статистика, из модели удаляются все транзакты, и она приводится к исходному состоянию, как перед первым прогоном. Устанавливаются в ноль сохраняемые величины и матрицы, что следует учитывать при использовании этих объектов для хранения исходных данных. Исключение составляют генераторы случайных чисел, которые не возвращаются к своим начальным значениям, что позволяет повторить прогон модели на новой последовательности случайных чисел. Оператор не имеет операндов.

Оператор CLEAR используется обычно для организации нескольких независимых прогонов модели на разных последовательностях случайных чисел. Перед повторением прогона можно при необходимости переопределить отдельные объекты модели, например емкости многоканальных устройств.

Пусть, например, требуется повторить прогон модели, приведенной на рис. 17, три раза при емкости МКУ, равной 1, 2 и 3. Это может быть выполнено так, как показано на рис. 20. После каждой очистки модели оператором CLEAR оператор STORAGE устанавливает новое значение емкости МКУ с именем STO2.

Оператор END (закончить) завершает сеанс работы с GPSS/PC и возвращает управление в операционную систему. Оператор не имеет операндов.

```
STO2 STORAGE 1
EXP FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915 .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE 100,FN$EXP
GATE SNF STO2,WAIT
ENT1 ENTER STO2
ADVANCE 160,FN$EXP
LEAVE STO2
```

```
UNLINK BUFER,ENT1,1
TERMINATE 1
WAIT LINK BUFER,FIFO
START 10000
CLEAR
STO2 STORAGE 2
START 10000
CLEAR
STO2 STORAGE 3
START 10000
```

Рис. 20

Как правило, управляющие операторы не включаются в исходную программу, т.е. не имеют номеров строк, а вводятся пользователем непосредственно с клавиатуры ПК.

5.9. Некоторые приемы конструирования GPSS-моделей

Косвенная адресация

В рассматривавшихся до сих пор примерах моделей ссылки на различные объекты GPSS/PC производились исключительно по данным им произвольным именам. Такая адресация объектов удобна, когда речь идет о небольшом числе объектов каждого типа. Если же число объектов некоторого типа велико, то во избежание пропорционального роста количества блоков в модели используют ссылки на эти объекты по их номерам с использованием так называемой косвенной адресации.

Идея косвенной адресации заключается в том, что каждый транзакт в некотором своем параметре содержит номер того или иного объекта, а в полях блоков, адресующихся к объектам, записывается ссылка на этот параметр транзакта. Проиллюстрируем применение косвенной адресации на примере следующей модели.

```
EXP FUNCTION RN1, C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915 .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3 .92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
CLASS FUNCTION RN1,D3
.333,1/.667,2/1,3
MEAN FUNCTION P$TYPE,L3
1,70/2,80/3,90
PRIOT VARIABLE 4-P$TYPE
STO2 STORAGE 2
WTIME QTABLE LINE,50,50,10
TTIME TABLE M1,100,100,12
GENERATE 100,FN$EXP
ASSIGN TYPE,FN$CLASS
PRIORITY V$PRIOT
QUEUE LINE
QUEUE P$TYPE
ENTER STO2
DEPART P$TYPE
DEPART LINE
ADVANCE FN$MEAN,FN$EXP
LEAVE STO2
```

```
TABULATE TTIME
TERMINATE 1
```

Рис. 21

Пусть на вход моделируемой многоканальной СМО с двумя каналами обслуживания поступает пуассоновский поток заявок со средним интервалом поступления 100 единиц модельного времени. Каждая заявка с равной вероятностью 1/3 относится к одному из трех классов: 1, 2 или 3, а среднее время обслуживания заявок каждого типа составляет соответственно 70, 80 и 90 единиц модельного времени. Чем меньше среднее время обслуживания заявки, тем выше ее приоритет. Необходимо построить модель, позволяющую оценить средние значения времени ожидания заявок каждого типа, а также распределения общего времени ожидания в очереди и общего времени пребывания в системе. Такая модель имеет вид, показанный на рис. 21.

Переменная PRIOT служит для вычисления приоритета транзакта как функции его класса, содержащегося в параметре с именем TYPE. Транзакты класса 1 (P$TYPE=1) получат приоритет 3, транзакты класса 2 - приоритет 2 и транзакты класса 3 - приоритет 1.

В блоке ASSIGN в параметр TYPE транзактов записывается класс заявки, разыгрываемый с помощью функции CLASS. В следующем блоке PRIORITY с помощью переменной PRIOT определяется приоритет транзактов, первоначально равный 0 (отсутствует поле E в блоке GENERATE).

Далее каждый транзакт "отмечается" в блоках QUEUE в двух очередях. Очередь с именем LINE является общей для транзактов всех классов. Входя в следующий блок QUEUE, транзакт отмечается в очереди с номером 1, 2 или 3 в зависимости от класса заявки, записанного в параметре TYPE. Аналогичным образом фиксируется уход из очередей в блоках DEPART. Таким образом, в модели создается четыре объекта типа "очередь": одна очередь с именем LINE и три с номерами 1, 2 и 3. При этом три последние очереди создаются одной парой блоков QUEUE-DEPART! В этом и заключается эффект косвенной адресации.

Как уже отмечалось ранее, каждому имени объекта симулятор сам ставит в соответствие некоторый номер. При ссылках на объекты одного и того же типа одновременно по именам и номерам, как это имеет место в рассматриваемом примере, существует опасность параллельной адресации к одному и тому же объекту вместо двух разных или, наоборот, к двум разным объектам вместо одного. Так, в рассматриваемой модели мы, вообще говоря, не знаем, какой именно номер поставит симулятор в соответствие имени очереди LINE. Если этот номер будет от 1 до 3, то это приведет к ошибке, так как в модели окажется не четыре очереди, а три, причем в одну из них будет заноситься информация как обо всех транзактах, так и дополнительно о транзактах одного из трех классов. Как избежать такой ситуации?

К счастью, в большинстве случаев об этом можно не заботиться, поскольку симулятор ставит в соответствие именам объектов достаточно большие номера, начиная с 10000. При необходимости же можно воспользоваться оператором EQU, о котором уже говорилось выше, и самостоятельно сопоставить имени объекта желаемый номер. Например, в рассматриваемой модели для того, чтобы очередь с именем LINE имела номер 4, достаточно записать оператор:

```
LINE EQU 4
```

5.10. Обработка одновременных событий

Так как модельное время в GPSS целочисленно, то оказывается вполне вероятным одновременное наступление двух или более событий, причем вероятность этого тем больше, чем крупнее выбранная единица модельного времени. В некоторых случаях одновременное наступление нескольких событий, или так называемый временной узел, может существенно нарушить логику модели.

Рассмотрим, например, еще раз модель на рис. 14. Здесь может образоваться временной узел между событиями "поступление транзакта на вход модели" и "завершение обслуживания в МКУ". Если непосредственно перед завершением обслуживания были заняты оба канала МКУ, то обработка временного узла зависит от последовательности транзактов, соответствующих событиям, в списке текущих событий.

Предположим, что первым в списке расположен транзакт, освобождающий канал МКУ. Тогда вначале будет обработан этот транзакт, т.е. событие "завершение обслуживания в МКУ", причем условие "МКУ STO2 не заполнено", проверяемое в блоке GATE, станет истинным. Затем будет обработан транзакт, поступивший на вход модели, в блок GATE с именем ENT1, из блока GENERATE или из блока TRANSFER в безусловном режиме. При этом транзакт будет впущен в блок ENTER, и МКУ в тот же момент модельного времени снова окажется заполненным. Такая ситуация при обработке временного узла представляется естественной.

Предположим теперь, что первым в списке текущих событий расположен транзакт, поступающий на вход модели. Так как условие "МКУ STO2 не заполнено" ложно, то блок GATE направит этот транзакт в блок с именем REFUS. Таким образом, в модели будет зафиксирован отказ в обслуживании, хотя в этот же момент модельного времени, после обработки транзакта, освобождающего канал, МКУ станет доступным.

Порядок расположения транзактов, соответствующих рассматриваемым событиям, в списке текущих событий случаен, и в среднем в половине случаев временной узел будет обрабатываться не так, как нужно. В результате статистика, связанная с отказами, окажется искаженной.

Для правильной обработки временного узла надо обеспечить такой порядок расположения транзактов в списке текущих событий, чтобы транзакт, освобождающий МКУ, всегда располагался первым. Этого можно добиться, управляя приоритетами транзактов (рис. 22).

```
STO2 STORAGE 2
EXP    FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915          .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE 100,FN$EXP
ENT1 GATE SNF STO2,REFUS
ENTER STO2
PRIORITY 1
ADVANCE 160,FN$EXP
LEAVE STO2
TERMINATE 1
REFUS TRANSFER .1,,OUT
ADVANCE 250,FN$EXP
TRANSFER ,ENT1
OUT TERMINATE 1
```

Рис. 22

Транзакты, поступающие в модель через блок GENERATE, имеют нулевой приоритет. Такой же приоритет имеют транзакты, получившие отказ в обслуживании, направленные в блок с именем REFUS и затем повторно поступающие в блок с именем ENT1. Те же транзакты, что поступают на обслуживание, повышают приоритет до 1 в блоке PRIORITY, и после выхода из блока ADVANCE возвращаются из списка будущих в список текущих событий, располагаясь в начале списка. Таким образом, нужный порядок транзактов обеспечивается, и временной узел будет обработан правильно.

Опасность неверной обработки временных узлов характерна для моделей со списками пользователя. Рассмотрим, например, еще раз модель на рис. 18. Здесь также возможен временной узел между событиями "приход транзакта" и "завершение обслуживания транзакта".

Пусть первым в списке текущих событий располагается вновь пришедший транзакт. Так как устройство с именем SYSTEM занято, то блок GATE направит этот транзакт в блок LINK, и он будет введен в список пользователя с именем LINE. Затем будет обработан транзакт, освобождающий устройство. Проходя через блок UNLINK, он выведет транзакт с начала списка пользователя и направит его в список текущих событий, где тот продвинется в блок SEIZE, занимая устройство SYSTEM.

Если же первым в списке текущих событий располагается транзакт, освобождающий устройство, то он выведет первый из ожидающих транзактов из списка пользователя в список текущих событий, где тот расположится после вновь пришедшего транзакта. Поэтому первым будет обработан вновь пришедший транзакт, который пройдет через блок GATE и займет устройство "без очереди". Транзакт-очередник, который был выведен из списка пользователя, "застрянет" перед блоком SEIZE и после очередного освобождения устройства займет его, нарушая, в свою очередь, логику работы модели.

Проведенный анализ показывает, что для правильной обработки временного узла необходимо обеспечить такой порядок расположения транзактов в списке текущих событий, чтобы первым всегда располагался вновь пришедший транзакт. В рассматриваемом случае этого можно добиться, используя блок PRIORITY с операндом BU (рис. 23).

Перед освобождением устройства обслуженный транзакт проходит через блок PRIORITY, который, оставляя неизменным приоритет транзакта PR, переводит его в конец списка текущих событий. При новом просмотре списка в случае наличия временного узла начинает обрабатываться вновь поступивший транзакт. Так как устройство еще занято, он направляется блоком GATE в список пользователя. При повторной обработке обслуженного транзакта тот освобождает устройство и выводит очередной транзакт из списка пользователя. Таким образом, правильная обработка временного узла обеспечивается и в этом случае.

```
EXP FUNCTION RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915      .7,1.2/.75,1.38/.8,1.6/.84,1.85/.88,2.12/.9,2.3
.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9 .99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
GENERATE 100,FN$EXP
ASSIGN TSRV,80,EXP
GATE NU SYSTEM,WAIT
SFAC SEIZE SYSTEM
ADVANCE P$TSRV
PRIORITY PR,BU
RELEASE SYSTEM
UNLINK LINE,SFAC,1
TERMINATE 1
WAIT LINK LINE,P$TSRV
```

Рис. 23

5.11 Команды GPSS/PC и технология работы с пакетом

5.11.1. Загрузка интегрированной среды

Пакет GPSS/PC включает в себя два основных модуля: модуль GPSSPC.EXE, представляющий интегрированную среду, в которой производится ввод, редактирование, отладка и выполнение модели, и модуль GPSSREPT.EXE, предназначенный для получения стандартного отчета GPSS/PC. Загрузка обоих модулей производится обычным образом из командной строки MS DOS или из программы-оболочки Norton Commander.

После загрузки интегрированной среды на экране появляется "заставка" с названием пакета: начинается так называемый сеанс работы с GPSS/PC. Затем заставка гасится, и появляется экран, разделенный на две части: большая верхняя часть содержит так называемое окно данных, меньшая нижняя часть - окно команд. Окно данных в начальный момент пусто, в окне команд в верхней командной строке высвечен символ "приглашения" > , сигнализирующий о готовности системы принимать команды.

5.11.2. Ввод новой модели

Если исходная программа с моделью еще не введена и не записана на диске, то необходимо ввести ее с клавиатуры. Ввод производится в командную строку. Сначала вводится номер строки очередного оператора и нажимается клавиша Пробел. Курсор автоматически перемещается к началу следующего поля - поля имени, и в позиции курсора высвечивается символ L , сигнализирующий о том, что вы находитесь в поле имени (LABEL - метка). Если оператор имеет имя, необходимо ввести его и нажать клавишу Пробел, в противном случае - сразу нажать клавишу Пробел. В любом случае курсор переходит к началу следующего поля - поля операции, о чем сигнализирует символ V (VERB - глагол) в позиции курсора. Необходимо ввести название оператора и нажать клавишу Пробел. Очень удобным является то, что название оператора не обязательно вводить полностью: как только транслятор распознает оператор по нескольким первым буквам, он после нажатия клавиши Пробел сам дополнит его до полного названия.

При синтаксической ошибке в операторе под командной строкой появляется указатель на место ошибки, причем ошибочный символ не вводится. Необходимо в этом случае повторить ввод символа.

Аналогичным образом вводятся поля операндов, при этом в позиции курсора высвечивается обозначение текущего поля (A, B, ... ,G). Для перехода к следующему полю операндов вводится запятая, для перехода к полю комментариев - Пробел. При переходе курсора в поле комментариев в позиции курсора высвечивается символ ; , сигнализирующий о возможности начать ввод комментария.

По окончании ввода последнего поля операндов или комментария следует нажать клавишу Enter, при этом введенный оператор транслируется и отображается в окне данных, а командная строка очищается, и в ее первой позиции снова появляется символ "приглашения".

По мере ввода новых операторов окно данных заполняется, и по окончании ввода в нем находится вся исходная программа в последовательности ввода, необязательно совпадающей с последовательностью нумерации строк. Для отображения в окне данных исходной программы в последовательности нумерации строк необходимо ввести в командную строку команду DISPLAY (отобразить). Эта команда, как и все остальные команды GPSS/PC, вводится без номера строки. С помощью команды DISPLAY можно также вывести в окно данных отдельную строку, указав ее номер в поле A команды, или последовательность строк, указав начальный и конечный номера в полях A и B соответственно.

5.11.3. Редактирование текста модели

Удалить строки из исходной программы можно командой DELETE (удалить), указав в полях A и B начальный и конечный номера удаляемой последовательности. Для удаления одной строки достаточно ввести лишь поле A.

При необходимости вставить в текст новый оператор, поместив его между уже введенными операторами, достаточно ввести его с промежуточным номером строки. Вы можете перенумеровать строки, введя команду RENUMBER (перенумеровать), в поле A которой указывается номер первой строки, а в поле B - шаг перенумерации.

Отредактировать содержимое строки можно с помощью команды EDIT (редактировать), в поле A которой указывается номер редактируемой строки. При вводе такой команды в командной строке появляется редактируемая строка. Подводя курсор к нужным позициям строки, вы можете внести в нее необходимые изменения. По окончании редактирования следует нажать клавишу Enter, и отредактированная строка перенесется в окно данных, заменив в исходной программе первоначальную строку с этим номером. Вы можете убедиться в этом, введя команду DISPLAY.

Если редактируемый оператор короткий, а изменений в нем много, то редактирование удобнее произвести, введя измененный оператор с тем же номером строки.

5.11.4. Запись и считывание модели с диска

Если работа с моделью предполагается и по окончании данного сеанса, то после ввода и редактирования исходную программу имеет смысл записать на диск. Для этого необходимо ввести команду SAVE (сохранить), в поле A которой указывается имя файла, в который будет записана модель. Файл должен иметь расширение .GPS.

Записав модель в файл, вы сможете в следующем сеансе работы с GPSS/PC не вводить ее заново с клавиатуры, а считать с диска, введя команду @спецификация_файла, где спецификация_файла - полное имя файла, которое вы дали исходной программе в команде SAVE, включающее расширение .GPS. При выполнении команды @ операторы исходной программы по мере их считывания из файла транслируются и выводятся в окно данных.

5.11.5. Прогон модели и наблюдение за моделированием

После того, как исходная программа модели введена с клавиатуры или считана с диска и оттранслирована, в памяти ПК создалась текущая модель, и теперь можно выполнить ее прогон. Для этого в командную строку необходимо ввести управляющий оператор START, указав в поле A соответствующее начальное значение счетчика завершений. После нажатия клавиши Enter оператор START переносится в окно данных, и прогон модели начинается. Об этом сигнализирует сообщение

Simulation in Progress ,

появляющееся в нижней строке командного окна - строке состояния, а также так называемый индикатор моделирования, мигающий в правой стороне нижней части окна данных.

Если прогон модели достаточно длинный, то можно наблюдать за процессом моделирования, открывая те или иные графические окна. Это производится путем нажатия клавиши Alt одновременно с символьной клавишей с первой буквой названия окна.

Например, после нажатия клавиш Alt+B в верхней части экрана на месте окна данных появляется окно блоков (BLOCKS), изображающее динамику продвижения транзактов через блок-схему модели. Рядом с каждым блоком выводится текущее число транзактов в нем, которое обновляется в процессе моделирования. Нажав клавиши Alt+N, вы можете за менить эту информацию на общее число транзактов, прошедших через каждый блок. Блок, в котором находится активный транзакт, выделен повышенной яркостью (на цветных мониторах - другим цветом).

Нажав клавиши Alt+F, вы можете наблюдать окно устройств (FACILITIES), в котором наглядно отображена информация о текущем состоянии каждого устройства модели: его использовании, занятости, очереди к нему.

Аналогичную информацию о многоканальных устройствах можно получить, нажав Alt+S и открыв окно памятей (STORAGES).

Если в модели используются статистические таблицы, то, нажав клавиши Alt+T, вы откроете окно таблиц (TABLES) с гистограммой распределения соответствующего атрибута модели, обновляющейся в процессе моделирования. Над гистограммой выводятся также текущие значения среднего и среднеквадратического отклонения табулируемого атрибута.

Если в модели используются матрицы, то, нажав клавиши Alt+M, вы откроете окно матриц (MATRICES), в котором можно наблюдать обновляющиеся в процессе моделирования значения элементов матриц.

Находясь в любом из перечисленных окон, вы можете путем нажатия клавиш Alt+L включить трассировку активного транзакта. При этом в верхней части окна появляется строка, содержащая информацию о текущем модельном времени, номере активного транзакта и его продвижении через блок-схему модели. Отключить трассировку можно повторным нажатием этих же клавиш.

Перемещение внутри окна любого типа к тому или иному объекту этого типа осуществляется путем нажатия клавиш управления курсором PgUp, PgDn и End. Возвращение в окно данных производится путем нажатия клавиш Alt+D.

Следует заметить, что наблюдение графических окон и особенно строки трассировки существенно замедляет моделирование, и при длинных прогонах моделей этой возможностью не следует злоупотреблять.

Открытие того или иного окна может быть выполнено также с помощью команды WINDOW (окно), в поле A которой указывается имя окна, однако удобнее это делать так, как описано выше.

Кроме графических окон внутри любого из них, кроме окна данных, может быть открыто до четырех микроокон. Микроокна открываются и закрываются командой MICROWINDOW (микроокно), имеющей следующий формат:

MICROWINDOW A,B,C ; комментарий

В поле A указывается номер микроокна - константа 1, 2, 3 или 4. Поле B содержит наблюдаемую величину - любой СЧА модели. Поле C определяет состояние микроокна в результате выполнения команды: ON - открыто, OFF - закрыто. Если поле C пусто, то по умолчанию команда открывает заданное микроокно. В поле комментария может быть задано название микроокна длиной до восьми символов.

При открытии любого окна заданные микроокна с обновляющейся в процессе моделирования информацией появляются в правой части соответствующего окна. Микроокно имеет форму прямоугольника с названием над рамкой, если оно было задано в комментарии к команде MICROWINDOW.

В процессе моделирования можно также наблюдать одновременно до двух графиков зависимостей любых СЧА модели от модельного времени. Для этого необходимо до запуска модели ввести одну или две команды PLOT (начертить), имеющие следующий формат:

PLOT A,B,C,D ; комментарий

В поле A указывается аргумент зависимости - любой СЧА модели.

Поле B должно содержать максимальное значение этого СЧА, определяющее масштаб изображения по оси Y. Операнд B задается константой, значение которой должно быть не менее 13. Поля C и D определяют начальное и конечное значения модельного времени, определяющие масштаб изображения по оси X. Эти операнды также задаются константами. В поле комментария может быть задан заголовок графика длиной до 34 символов.

График обновляется при каждом изменении модельного времени, если оно попадает в диапазон, заданный операндами C и D. Указанный в поле A СЧА-аргумент вычисляется относительно первого транзакта, обрабатываемого после изменения модельного времени.

Процесс моделирования можно прервать, нажав одну из клавиш Esc или Home. При этом в строке состояния командного окна появляется сообщение о номере активного транзакта, обрабатываемого симулятором в момент прерывания. Вы можете узнать значения интересующих вас стандартных числовых атрибутов модели в момент прерывания, введя команду SHOW (показать), операндом которой служат отдельные СЧА или выражения из них. Значение заданного в команде СЧА или выражения выводится в окно данных или другое активное окно. Введя команду EVENTS (события), можно увидеть в окне данных содержимое списков текущих и будущих событий. Команда USERCHAINS (списки пользователя) позволяет просматривать в окне данных содержимое списков пользователя. Обе последние команды не имеют операндов.

Инициировать прерывание моделирования можно также с помощью команды STOP (остановить), имеющей следующий формат:

STOP A,B,C

В поле A указывается номер транзакта, вызывающего прерывание, задаваемый константой. Если это поле пусто, то прерывание вызывается любым транзактом. В поле B задается имя или номер блока, при входе в который происходит прерывание. Если этот операнд опущен, то прерывание происходит при входе в любой блок. В поле C указывается ON для установки условия прерывания и OFF для снятия этого условия (по умолчанию ON).

Например, команда

STOP 100,MET1

устанавливает условие прерывания моделирования при входе транзакта с номером 100 в блок с именем MET1. Команда

STOP 2

будет вызывать прерывание при каждом продвижении транзакта с номером 2, а команда

STOP ,CHAIR

при каждом входе любого транзакта в блок с именем CHAIR. Наконец, команда

STOP без операндов

будет вызывать прерывание при каждом продвижении любого транзакта, а команда

STOP ,,OFF

снимает все условия прерывания, установленные ранее другими командами STOP.

Прервав моделирование, можно также воспользоваться командой STEP (выполнить шаг) для пошагового выполнения модели с целью ее отладки. Операнд в поле A команды задает количество входов активного транзакта в блоки, которое производится при каждом выполнении команды. Обычно этот операнд равен 1, и каждое выполнение команды STEP приводит к продвижению активного транзакта к следующему блоку. Отладку с использованием команды STEP удобно проводить, находясь в окне блоков.

Для продолжения моделирования после прерывания следует ввести в командную строку команду CONTINUE (продолжить).

Команды STEP и CONTINUE могут не только вводиться в командную строку с клавиатуры, но и выбираться из меню команд, появляющемся в командном окне при активизации любого графического окна. Выбор производится подводом крестообразного курсора в прямоугольную область нужной команды и нажатием клавиши Ins. В окне блоков меню команд предоставляет также некоторые дополнительные возможности [8].

Команды STEP, CONTINUE, а также любые другие часто используемые команды удобно загрузить на функциональные клавиши F1-F10. Для этого после ввода загружаемой команды с клавиатуры необходимо нажать клавиши Ctrl+Fn, где n - номер выбранной функциональной клавиши. После загрузки команды на функциональную клавишу для ее выполнения достаточно нажатия этой клавиши.

5.11.6. Получение и интерпретация стандартного отчета

По завершении прогона модели раздается звуковой сигнал, и в строке состояния появляются сообщения

Writing REPORT.GPS Simulation Complete Reporting ... , сигнализирующие о том, что моделирование закончено и в данный момент производится создание отчета о прогоне модели. Затем система переходит в состояние ожидания дальнейших команд.

Отчет, создаваемый по завершении моделирования, записывается в файл со стандартным именем REPORT.GPS. Это имя может быть изменено командой REPORT (создать отчет), имеющей следующий формат:

REPORT A,B В поле A указывается спецификация файла, в который должен быть выведен отчет. Если поле B содержит ключевое слово NOW, то отчет создается немедленно после ввода команды.

Необходимо иметь ввиду, что отчет, создаваемый автоматически по завершении прогона модели или командой REPORT, является неформатированным, т.е. непригодным для непосредственного просмотра. Для форматирования и создания стандартного отчета GPSS/PC необходимо завершить сеанс и выполнить программу форматирования отчета. Выход из интегрированной среды (завершение сеанса) производится путем ввода управляющего оператора END (закончить). При этом производится выход в MS DOS или в программу-оболочку Norton Commander.

Для форматирования отчета необходимо загрузить модуль форматирования GPSSREPT.EXE. После его загрузки на экране появляется "заставка" с названием модуля, двумя окнами в нижней части экрана и сообщениями-подсказками. В левом окне выведено имя файла, в котором находится неформатированный отчет (по умолчанию это файл REPORT.GPS). В правом окне выведено обозначение устройства, куда должен быть выведен форматированный отчет (по умолчанию это экран дисплея SCRN:). Форматированный отчет может быть также выведен на печать или на диск. Для этого в правое окно надо ввести обозначение PRN: или имя файла на диске соответственно. Для переключения окон используется клавиша Enter. Для создания отчета на выбранном устройстве следует нажать клавишу Пробел, для выхода из программы - клавишу Esc.

Если содержимое окон по умолчанию оставлено без изменения, то после нажатия клавиши Пробел на экране появляется отчет о последнем прогоне модели, выполненном перед завершением сеанса работы с модулем GPSSPC.EXE. Отчет содержит следующую информацию:

1) общие сведения о модели и ее прогоне, включающие модельное время начала (START_TIME) и конца (END_TIME) прогона, количество блоков в модели (BLOCKS), количество устройств (FACILITIES), количество многоканальных устройств (STORAGES), объем памяти, остававшейся свободной при прогоне модели (FREE_MEMORY);

2) сведения об именах объектов модели, включающие для каждого имени идентификатор (NAME), присвоенное ему числовое значение (VALUE) и тип имени: 0, если числовое значение имени присвоено пользователем с помощью оператора EQU; 1, если числовое значение имени присвоено системой; 2, если имя является именем блока;

3) сведения о блоках модели, включающие для каждого блока номер строки исходной программы (LINE), номер или имя блока (LOC), название блока (BLOCK_TYPE), количество транзактов, прошедших через блок (ENTRY_COUNT), текущее количество транзактов в блоке в момент завершения моделирования (CURRENT_COUNT), количество транзактов, заблокированных перед блоком в момент завершения моделирования (RETRY);

4) сведения об устройствах модели, включающие для каждого устройства его имя или номер (FACILITY), количество занятий устройства (ENTRIES), коэффициент использования (UTIL.), среднее время на одно занятие (AVE._TIME) и ряд других данных;

5) сведения о многоканальных устройствах модели, включающие для каждого МКУ его имя или номер (STORAGE), емкость (CAP.), количество свободных каналов в момент завершения моделирования (REMAIN.), наименьшее (MIN.) и наибольшее (MAX.) количество занятых каналов в процессе моделирования, количество занятий МКУ (ENTRIES), среднее количество занятых каналов (AVE.C.), коэффициент использования (UTIL.) и ряд других данных;

6) сведения об очередях модели, включающие для каждой очереди ее имя или номер (QUEUE), максимальную длину очереди в процессе моделирования (MAX.), текущую длину очереди в момент завершения моделирования (CONT.), общее количество транзактов, вошедших в очередь в процессе моделирования (ENTRIES), и количество "нулевых" входов в очередь (ENTRIES(0)), среднюю длину очереди (AVE.CONT.), среднее время ожидания в очереди с учетом всех транзактов (AVE.TIME) и без учета "нулевых" входов (AVE.(-0));

7) сведения о статистических таблицах модели, включающие для каждой таблицы ее имя или номер (TABLE), среднее значение (MEAN) и среднеквадратическое отклонение (STD.DEV.) табулируемой величины, границы частотных интервалов (RANGE), частоты (FREQUENCY) и накопленные частоты в процентах (CUM.%) попадания наблюдений в эти интервалы;

8) сведения о списках пользователя модели, включающие для каждого списка его имя или номер (USER_CHAIN), количество транзактов в списке в момент завершения моделирования (CHAIN_SIZE), среднее количество транзактов в списке (AVE.CONT), общее количество транзактов, вошедших в список в процессе моделирования (ENTRIES), максимальное количество транзактов, находившихся в списке (MAX), среднее время пребывания транзакта в списке (AVE.TIME);

9) сведения о логических переключателях модели, включающие для каждого ЛП его имя или номер (LOGICSWITCH) и состояние ЛП в момент завершения моделирования: 1 - "включен", 0 - "выключен";

10) сведения о сохраняемых величинах модели, включающие для каждой сохраняемой величины ее имя или номер (SAVEVALUE) и значение в момент завершения моделирования (VALUE);

11) сведения о матрицах модели, включающие для каждой матрицы ее имя или номер (MATRIX), а также список всех элементов матрицы в формате: "строка" (ROW), "столбец" (COLUMN), "значение" (VALUE).

Если в операторе START задан вывод в отчет списков текущих и будущих событий, то отчет включает в себя также сведения о транзактах, находившихся в момент завершения моделирования в этих списках. Сведения о транзактах размещаются в отчете в соответствии с размещением транзактов в каждом списке.

Информация о списке текущих событий включает в себя для каждого транзакта его номер (XACT_NUMBER), приоритет (PRI), резидентное время транзакта (M1), номер текущего блока (CURRENT), номер следующего блока (NEXT), а также перечень всех параметров транзакта в формате: "параметр" (PARAMETER), "значение" (VALUE).

Информация о списке будущих событий включает для каждого транзакта те же данные, однако вместо резидентного времени транзакта (M1) выводится запланированное время выхода транзакта из списка будущих событий (BDT).

Разумеется, сведения об объектах того или иного типа появляются в отчете только в том случае, если в модели присутствует хотя бы один объект данного типа. Кроме того, включением в отчет сведений об объектах разных типов можно управлять с помощью так называемого установочного файла SETTINGS.GPS [8]. В отчетах о прогоне моделей, включающих в себя другие, не рассматривавшиеся здесь объекты GPSS/PC, появляется соответствующая информация и об этих объектах.

На рис. 24 приведен отчет о прогоне модели примера на рис. 21.

START_TIME	END_TIME	BLOCKS	FACILITIES	STORAGES	FREE_MEMORY
0	14617	12	0	1	274320

LINE	LOC	BLOCK_TYPE	ENTRY_COUNT	CURRENT_COUNT	RETRY
80	1	GENERATE	150	0	0
90	2	ASSIGN	150	0	0
100	3	PRIORITY	150	0	0
110	4	QUEUE	150	0	0
120	5	QUEUE	150	0	0
130	6	ENTER	150	0	0
140	7	DEPART	150	0	0
150	8	DEPART	150	0	0
160	9	ADVANCE	150	0	0
170	10	LEAVE	150	0	0
180	11	TABULATE	150	0	0
190	12	TERMINATE	150	0	0

QUEUE	MAX	CONT.	ENTRIES	ENTRIES(0)	AVE.CONT.	AVE.TIME	AVE.(-0)
1	1	0	54	48	0.02	6.07	54.67
2	1	0	42	35	0.01	4.14	24.86
3	1	0	54	49	0.02	6.22	67.20
LINE	2	0	150	132	0.06	5.59	46.56

STORAGE	CAP.	REMAIN.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.
STO2	2	2	0	2	150	1	0.66	0.328

TABLE	MEAN	STD.DEV.	RETRY	RANGE	FREQUENCY	CUM.%
WTIME	5.59	25.23	0			
				- 50	144	96.00
				50 - 100	3	98.00
				100 - 150	1	98.67

```
    150 -   200       2    100.00

TTIME   69.48   70.88   0   -
   -  100     117     78.00
100 -   200      23     93.33
200 -   300       8     98.67
300 -   400       2    100.00
```

Рис. 24

Отчет выводится на экран постранично. Для вывода очередной страницы необходимо нажать клавишу Пробел, для прекращения вывода отчета - клавишу Esc. По окончании вывода отчета на экране появляется сообщение

[SPACE] for another report Any other key to end

Для создания отчета на другом устройстве или другого отчета надо нажать клавишу Пробел, для выхода из программы GPSSREPT - любую другую клавишу.

Помимо отчета отдельные результаты моделирования могут быть

также выведены в базу данных GPSS/PC [8] с помощью команд RESULT. Однофакторный дисперсионный анализ и получение доверительных интервалов для выведенных в базу данных характеристик модели могут быть выполнены с помощью команды ANOVA. Рассмотрение этих команд выходит за рамки данного издания.

6 .Обобщенные (комбинированные) модели (А-схемы)

Существуют общие подходы к формальному описанию процессов функционирования систем. Наиболее известным из них является подход, предложенный Н.П.Бусленко. Это подход позволяет описать поведение непрерывных и дискретных, детерминированных и стохастических систем. По сравнению с рассмотренными выше данный подход является обобщенным (универсальным) и базируется на понятии *агрегативной системы*, представляющей собой формальную схему общего вида, которую будем называть А-схемой.

При агрегативном описании сложный объект (система) разбивается на конечное число частей (подсистем), сохраняя при этом связи, обеспечивающие их взаимодействие. Подсистемы в свою очередь могут разбиваться на элементы, если они представляются более удобными для математического описания.

В качестве элемента А-схемы выступает агрегат. Связь между агрегатами системы и с внешней средой осуществляется с помощью оператора сопряжения $\boldsymbol{R}$.

Каждый агрегат характеризуется следующими множествами:

1.моментов времени $\boldsymbol{T}$;

2.входных сигналов $\boldsymbol{X}$;

3.вы ходных сигналов $\boldsymbol{Y}$;

4.состояний $\boldsymbol{Z}$ в каждый момент времени $\boldsymbol{t}$;

5.собственными (внутренними) параметрами $\boldsymbol{H}$.

Состояние агрегата в момент времени $\boldsymbol{t}$ обозначается $\boldsymbol{z(t)}$, а входные и выходные сигналы $\boldsymbol{x(t)}$ и $\boldsymbol{y(t)}$ соответственно. Переход агрегата из состояния $\boldsymbol{z(t_1)}$ в состояние $\boldsymbol{z(t_2)}$ происходит за малый интервал времени, т.е. имеет место скачок δz. Такие переходы определяются входными сигналами $\boldsymbol{x(t)}$ и собственными (внутренними) параметрами самого агрегата $\boldsymbol{h(t)}$.

Последовательность входных сигналов, поступающих в А-схему, называют входным сообщением или $\boldsymbol{x}$- сообщением, а последовательность выходных сигналов - выходным сообщением или $\boldsymbol{y}$-сообщением.

Большие системы ввиду их сложности не могут быть формализованы в виде математических схем одиночных агрегатов. Поэтому их формализуют некоторой конструкцией из отдельных агрегатов $\boldsymbol{A_n}$, которую называют агрегативной системой или А-схемой. Для описания реальной системы в виде А-схемы необходимо иметь описание как отдельных агрегатов $\boldsymbol{A_n}$, так и связей между ними.

Рассмотрим А-схему, структурная схема которой приведена на рис 6.1.

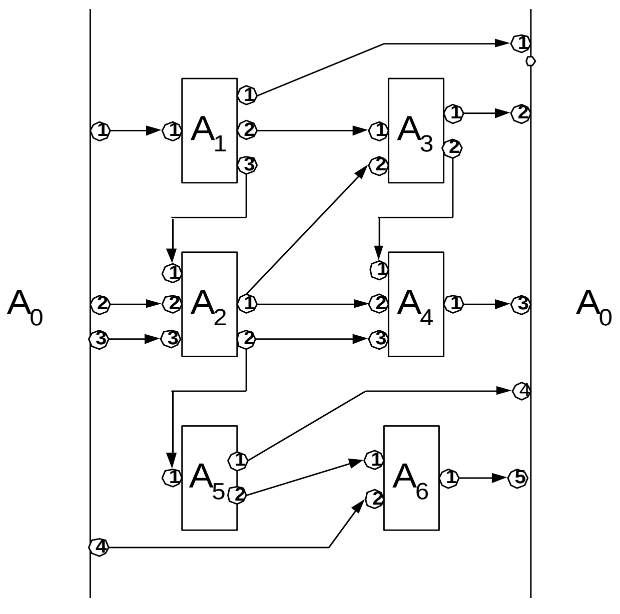

Рис.6.1. Структура агрегативной системы

Внешняя информация в приведенной схеме поступает от внешних объектов, не являющихся элементами рассматриваемой схемы. Внутренняя информация вырабатывается агрегатами самой ***A***-схемы. Обмен информацией между а_схемой и внешней средой происходит через агрегаты, называемые полюсами ***A***-схемы.

Различают входные полюсы, на которые поступают ***x***-сообщения (агрегаты A_1, A_2, A_3) и выходные полюсы ***A***-схемы, на которых формируются ***y***-сообщения (агрегаты A_3, A_4, A_5, A_6). Агрегаты, не являющиеся полюсами, называются внутренними.

Для построения формального представления ***A***-схемы необходимо выбрать удобные способы математического описания взаимодействия между агрегатами. Взаимодействие ***A***-схемы с внешней средой рассматривается как обмен сигналами между с внешней средой и элементами ***A***-схемы. Поэтому внешнюю среду можно представить в виде фиктивного элемента системы A_0.

Сигнал, выдаваемый ***A***-схемой во внешнюю среду, принимается элементом A_0 как входной сигнал. Сигнал, поступающий в ***A***-схему из внешней среды, является выходным сигналом элемента A_0.

Каждый элемент ***A***-схемы характеризуется множеством входных контактов $\{x_i^{(n)}\}$ и множеством выходных контактов $\{y_j^{(k)}\}$.

Полученная пара множеств $\{x_i^{(n)}\}$, $\{y_j^{(k)}\}$ является математической моделью элемента A_n, использующейся для формального описания сопряжения его с другими элементами ***A***-схемы и внешней средой.

Введем оператор R – сопряжения элементов (агрегатов) в A-схему.

$$Y_j^{(k)} = R(X_i^{(n)})$$

Оператор R отражает связь входных и выходных контактов, связанных с элементарным каналом. Задать оператор сопряжения можно в виде таблицы, в которой на пересечении строк с номерами элементов (агрегатов) ***n*** и столбцов с номерами входных контактов ***i*** располагаются пары чисел ***k***, ***l***, указывающих номер элемента ***k*** и номер контакта ***l***, с которым соединен контакт ***i***.

Если в A-схеме к контакту $X_i^{(n)}$ не подключен никакой элементарный канал, то оператор R не определен на этом контакте. Рассмотрим оператор сопряжения, для A-схемы, приведенной на рис 6.1

Таблица 6.1 - Оператор сопряжения для A-схемы рис.6.1

n	i (вх. контакты)				
(эл-т)	1 (э-лт,к-т)	2	3	4	5
j↓ 0	1, 1	3, 1	4, 1	5, 1	6, 1
1	0, 1				
2	1, 3	0, 2	0, 3		
3	1, 2	2, 1			
4	3, 2	2, 1	2, 2		
5	2, 2				
6	5, 2	0, 4			

Если столбцы и строки такой таблицы пронумеровать двойными индексами ***n***, ***i*** и ***k***, ***l*** соответственно, и на пересечении помещать ***1*** для контактов ***n***, ***i*** и ***k***, ***l***, соединенных элементарным каналом, и ***0*** в противном случае, то получим матрицу смежности ориентированного графа, вершинами которого являются контакты агрегатов, а дуги - элементарные каналы A-схемы.

Моделирование СМО с использованием агрегативных моделей

Рассмотри процесс перехода от содержательного описания системы к A-схеме на примере ***Q***-схемы, приведенной на рис. 3.6. Для представления этой системы в виде A-схемы будем использовать пять типов агрегатов:

A^E – внешняя среда;

A^H – накопитель;

A^K – канал;

A^P – распределитель;

A^C – сумматор.

Функции агрегатов A^H и A^K соответствуют функциям таких элементов Q-схемы, как накопитель H и канал K. Агрегат A^E позволяет формировать взаимодействие между агрегатами A-схемы и внешней средой E. Вспомогательные агрегаты A^P и A^C служат для синхронизации работы агрегатов A-схемы в соответствии с принятыми дисциплинами постановки в очередь и обслуживания заявок.

Агрегаты A^C позволяют передать сигналы с выходных контактов нескольких агрегатов на один и тот же входной контакт другого агрегата.

Тогда структура A-схемы будет иметь вид, приведенный на рис.6.2.

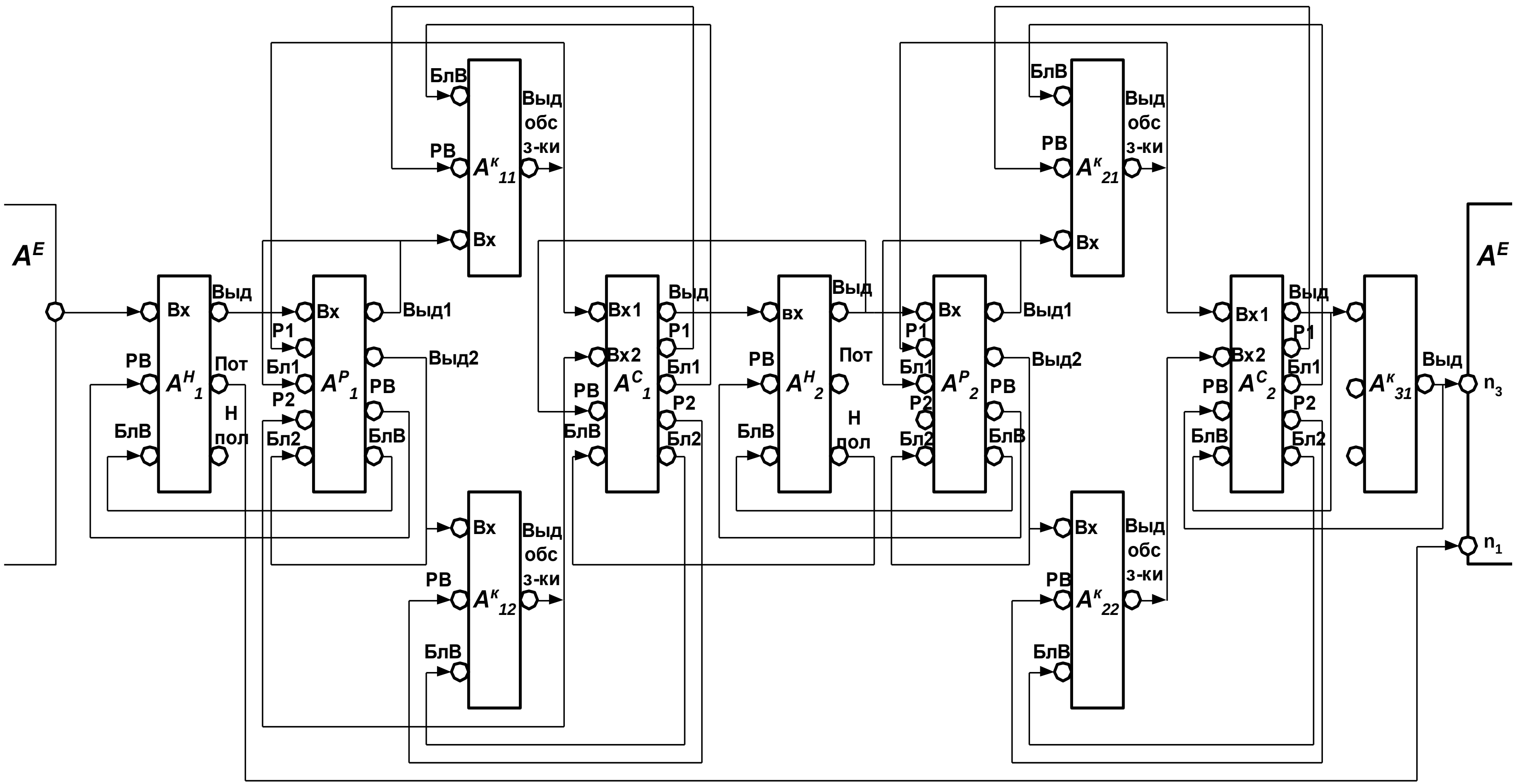

Рис.6.2. А-схема общего вида

Опишем работу отдельных агрегатов.

1.*Агрегат* A^E – внешняя среда.

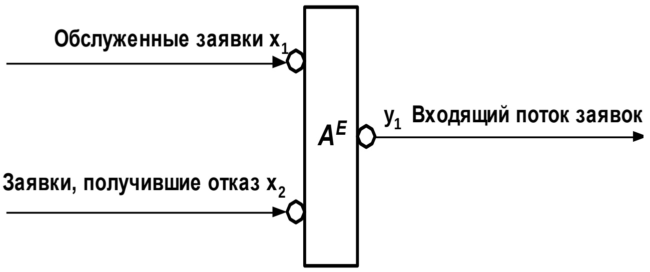

Рис. 6.3

Здесь x_1, x_2 – входные контакты; y_1 – выходные контакты.

На вход x_1 поступают обслуженные заявки, на вход x_2 – заявки, получившие отказ. С выхода y_1 снимают заявки через промежутки времени, распределенные по закону распределения входящего потока заявок.

2.*Агрегат* A^K – канал.

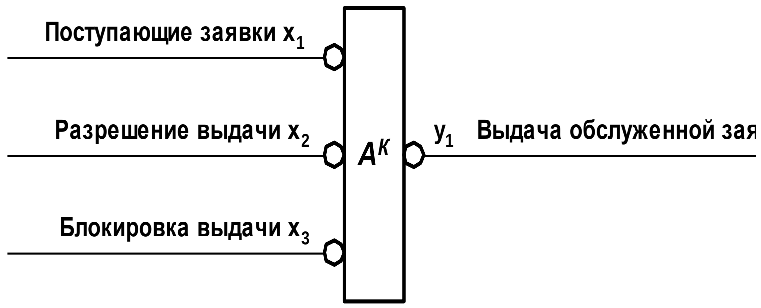

Рис. 6.4

Здесь x_1 – сигнал поступления заявок на обслуживание; x_2 – разрешение на выдачу обслуженной заявки; x_3 – блокировка выдачи обслуженной каналом заявки; y_1 – сигнал выдачи обслуженной каналом заявки.

Вектор состояний агрегата

$$z^{(k)}(t) = (z_1^{(k)}(t), z_2^{(k)}(t), z_3^{(k)}(t))$$

где $z_1^{(k)}(t)$ – время, оставшееся до окончания обслуживания заявки, которая находится в канале.

$$z_2^{(k)}(t) = \begin{cases} 1, \text{ если канал заблокирован,} \\ 0 \text{ в противном случае.} \end{cases}$$

$$z_3^{(k)}(t) = \begin{cases} 1, \text{ если заявка находится в канале,} \\ 0 \text{ в противном случае.} \end{cases}$$

Если время обслуживания заявки в канале истекло, т.е.

$z_1^{(k)}(t) \le 0$, но ее выдача из канала запрещена, т.е. $z_2^{(k)}(t) = 1$, то заявка остается в канале до тех пор, пока не придет сигнал $z_2^{(k)}(t) = 0$.

3.*Агрегат* A^H – накопитель.

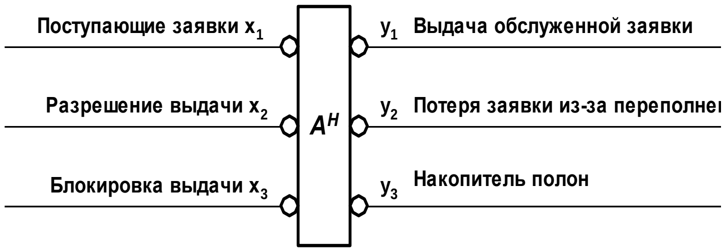

Рис. 6.5

Здесь x_1, x_2, x_3 соответствуют входам агрегата A^K.

С выхода y_1 выдается заявка, стоящая в очереди в накопителе первой.

С выхода y_2 выдается заявка, потерянная из-за переполнения накопителя.

С выхода y_3 выдается сигнал о заполнении накопителя.

Внутреннее состояние агрегата A^H описывается вектором

$$z^{(н)}(t) = (z_1^{(н)}(t), z_2^{(н)}(t)),$$

где $z_1^{(н)}(t)$ – число заявок в накопителе;

$$z_2^{(н)}(t) = \begin{cases} 1, \text{ если установлена блокировка,} \\ 0 \text{ в противном случае.} \end{cases}$$

4.*Агрегат* A^P – распределитель.

Распределитель служит для разделения поступающего на вход x_1 потока заявок по двум направлениям: выход y_1 и выход y_2 в соответствии с алгоритмом взаимодействия накопителя и канала.

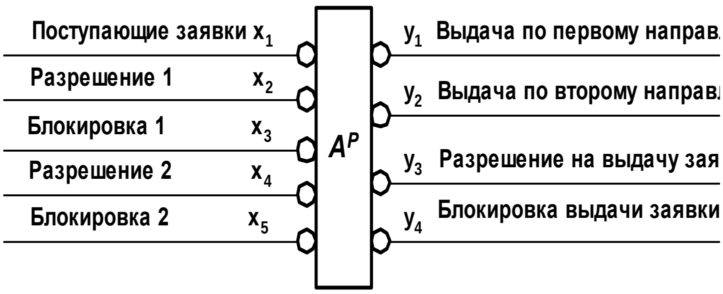

Рис. 6.6

В рассматриваемом примере, рис.6.2, заявка, поступающая в A^P, передается через выход y_1, если соответствующий ему агрегат A^K свободен для обслуживания заявки. В противном случае заявка выдается через выход y_2, если есть сигнал на входе «Разрешение 2».

Информация о занятости агрегатов A^K, на которые поступают заявки с выходов y_1 и y_2 агрегата A^P передается на входные контакты x_2 - x_5 агрегата A^P.

Если оба агрегата A^K не могут принять заявки от агрегата A^P, то на выходной контакт y_4 агрегата A^P выдается сигнал, запрещающий передачу заявки агрегату A^P. Как только один из агрегатов A^K освободится (о чем выдается соответствующий сигнал агрегату A^P на входы x_2, x_4), сигнал с контакта y_3 агрегата A^P разрешает посылаемому агрегату A^H пересылку заявки через агрегат A^P в агрегат A^K.

Внутреннее состояние агрегата A^P определяется вектором

$$z^{(p)}(t) = (z_1^{(p)}(t), z_2^{(p)}(t)),$$

где $z_1^{(p)}(t)$ – число заявок в накопителе;

$$z_1^{(p)}(t) = \begin{cases} 1, \text{ если разрешена передача заявки по выходу 1,} \\ 0 \text{ в противном случае.} \end{cases}$$

$$z_2^{(p)}(t) = \begin{cases} 1, \text{ если разрешена передача заявки по выходу 2,} \\ 0 \text{ в противном случае.} \end{cases}$$

5.*Агрегат* A^C – сумматор.

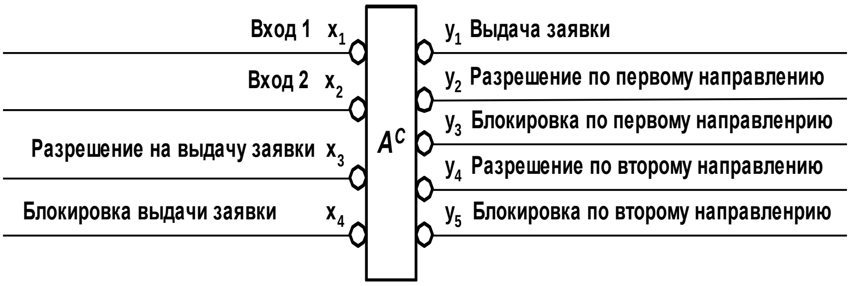

Рис. 6.7

Агрегат A^C выполняет функции, обратные агрегату A^P, т.е. избирательно суммирует поступающие поступающие заявки от двух посылающих агрегатов A^K и передает их на вход принимающего агрегата (A^H или A^K) через выход y_1.

При поступлении на вход x_1 или x_2 заявки от передающих агрегатов, агрегат A^C опрашивает принимающие агрегаты. Для этого на вход x_3 агрегата A^C подается разрешающий сигнал из принимающего агрегата, или на вход x_4 запрещающий.

Для приема заявки агрегат A^C в случае готовности ее принять передает разрешающий сигнал с выхода y_2 или y_4 на соответствующий вход посылающего агрегата. В случае неготовности принять заявку агрегат A^C передает сигнал блокировки с выхода y_3 или y_5 на соответствующий вход посылающего агрегата.

Укрупненная схема моделирующего алгоритма системы, приведенной на рис. 3.6 и представленной в виде *A*-схемы рис. 6.2, приведена на рис. 6.8.

В основу моделирования положен принцип δz особых состояний. Обработка каждого особого состояния выполняется блоками 6 и 12.

Схема алгоритма блока обработки особого состояния приведена на рси.6.9. Работа такого блока к выбору типа агрегата (A^E, A^K, A^H, A^P или A^C), для которого реализуется дальнейшее продвижение при моделировании.

Схема алгоритма, имитирующего работу блока A^E (воздействие внешней среды на систему) приведена на рис 6.10. В начале определяется какое событие имело место:

- поступил входной сигнал (закончилось обслуживание заявки);
- выдан сигнал из блока A^E, т.е. поступила заявка из входного потока в *A*-схему.

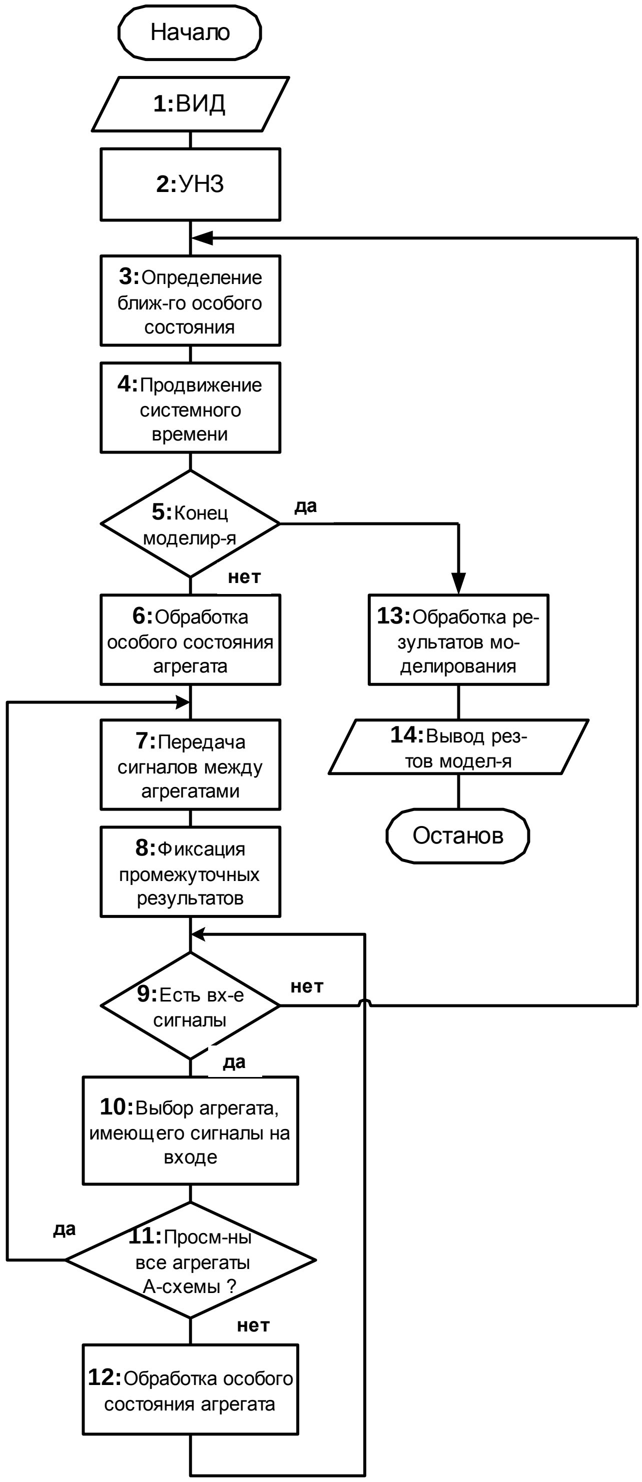

Рис. 6.8. Укрупненная схема моделирующего алгоритма А-схемы

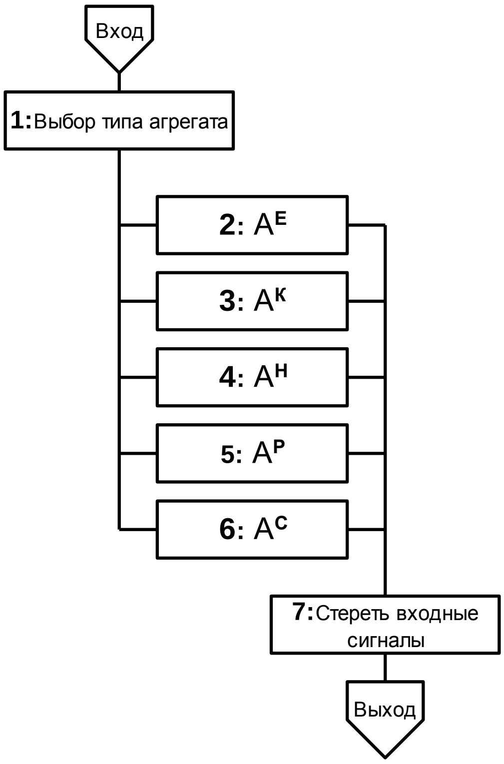

Рис.6.9. Схема алгоритма блока 6 рис. 6.8

При наступлении времени выдачи заявки она выдается в ***A***-схему (блок 2 рис.6.10) и генерируется интервал времени до момента поступления новой заявки (блок 2 рис.6.10).

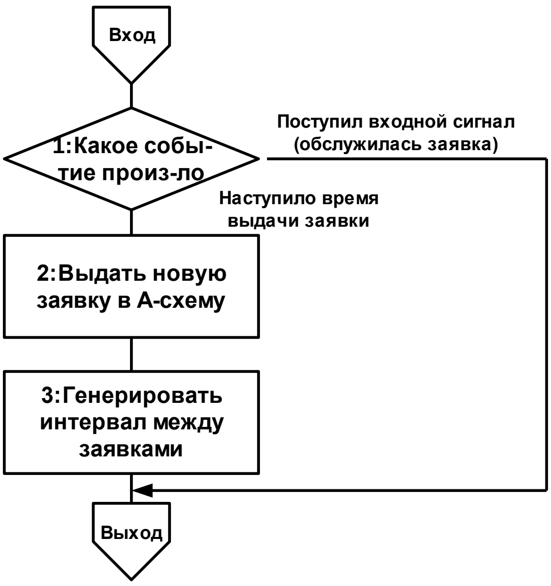

Рис.6.10. Схема алгоритма блока A^E

Схема алгоритма, имитирующего работу агрегата $\boldsymbol{A^K}$, приведена на рис. 6.11. Работа схемы данного алгоритма полностью соответствует описанию процесса функционирования агрегата $\boldsymbol{A^K}$, рассмотренного ранее.

Схема алгоритма, имитирующего работу агрегата $\boldsymbol{A^H}$, приведена на рис. 6.12. Работа схемы данного алгоритма полностью соответствует описанию процесса функционирования агрегата $\boldsymbol{A^H}$, рассмотренного ранее.

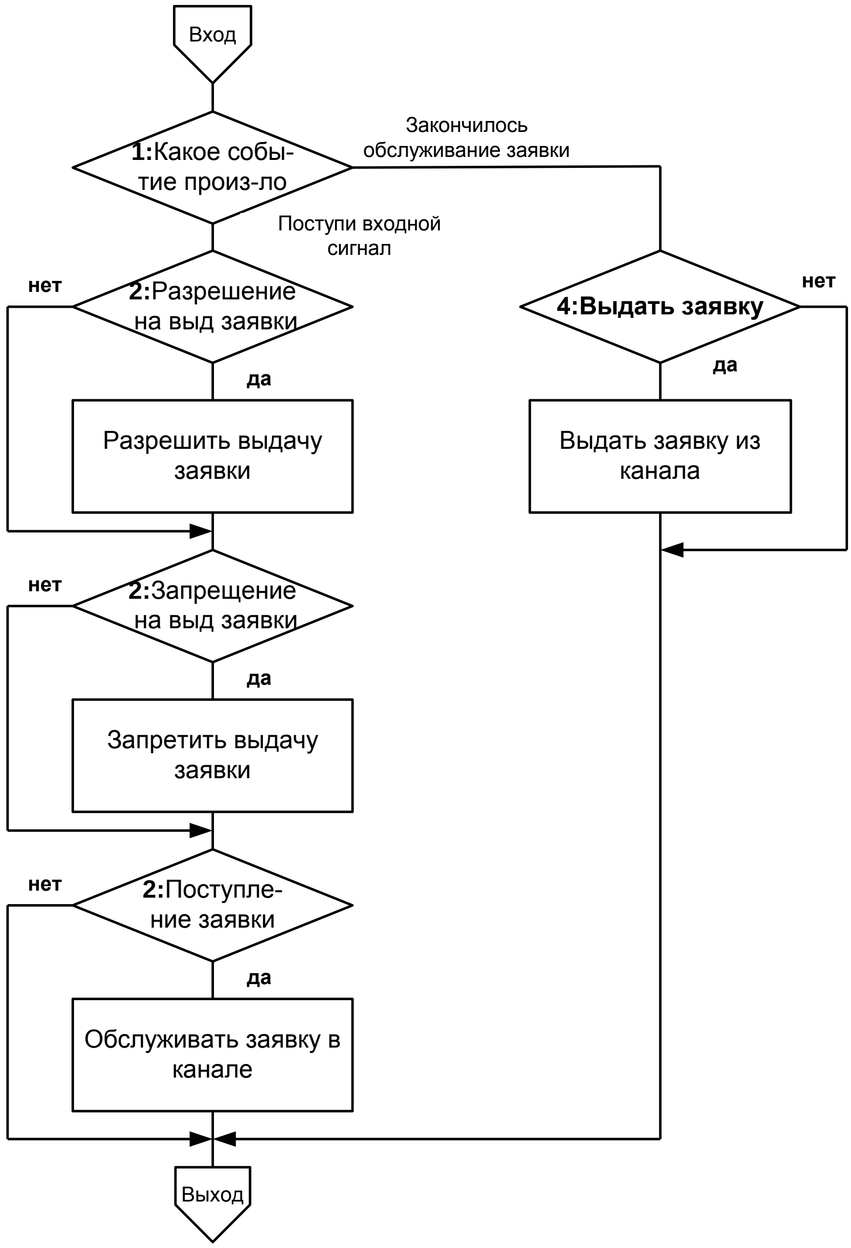

Рис.6.11. Схема алгоритма блока $\boldsymbol{A^K}$

Рис.6.12. Схема алгоритма блока A^H

Схема алгоритма, имитирующего работу агрегата A^P, выполняющего вспомогательные функции сопряжения агрегатов, представлена на рис. 6.13.

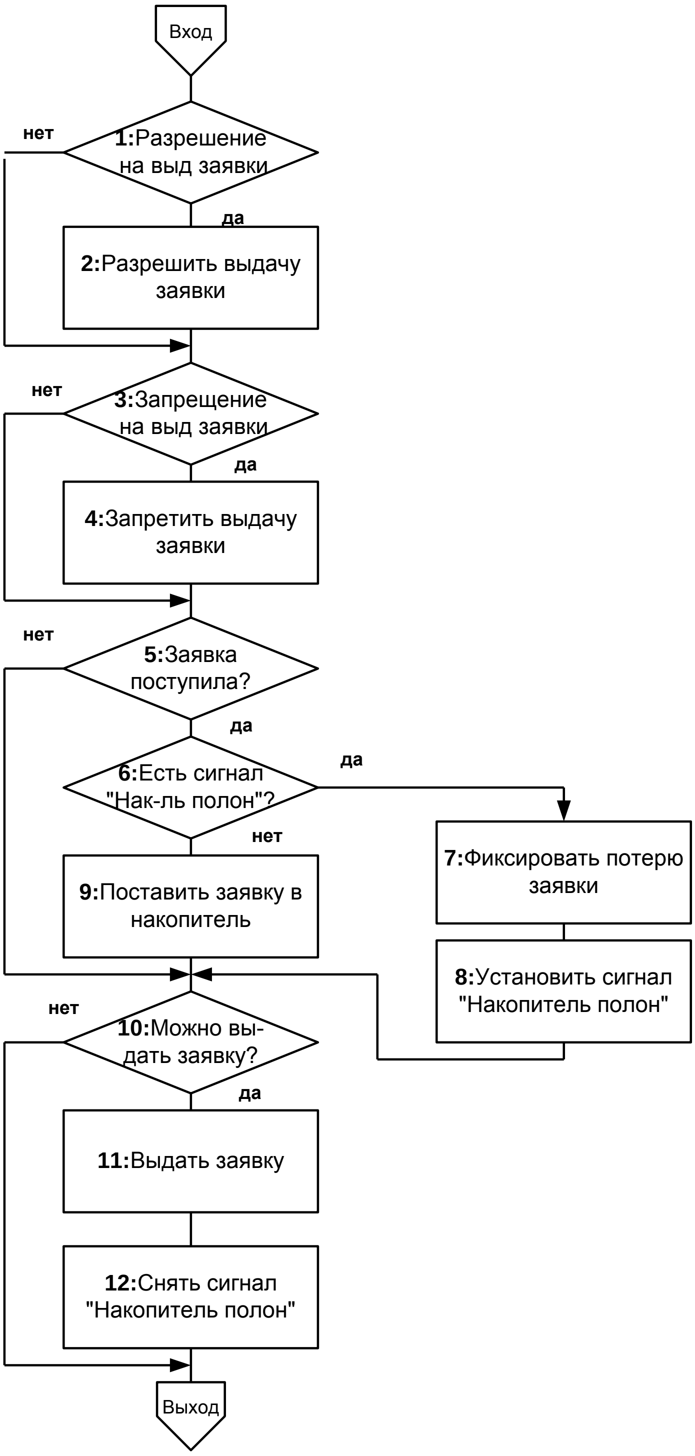

Рис.6.13. Схема алгоритма блока $\boldsymbol{A}^{\boldsymbol{P}}$

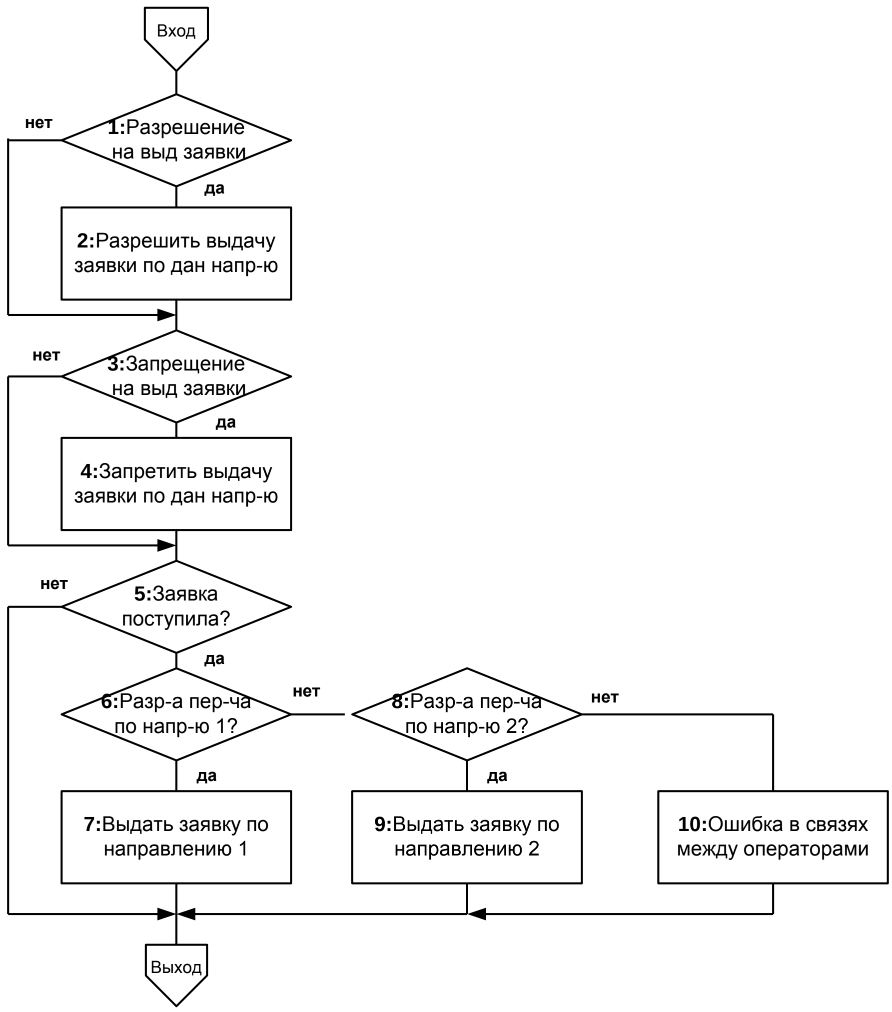

Рис.6.14. Схема алгоритма блока

Блоки 1 и 2 схемы алгоритма рис.6.13 выполняют анализ входного сигнала "Разрешение на выдачу заявки" или" и " Разрешение на выдачу заявки" или", соответственно. Блок 10 формирует сигнал "Ошибка в связях между операторами" на основе сигналов принимаемых от принимающего агрегата.

Схема алгоритма, имитирующего работу агрегата $\boldsymbol{A^C}$, выполняющего вспомогательные функции сопряжения агрегатов, представлена на рис. 6.15.

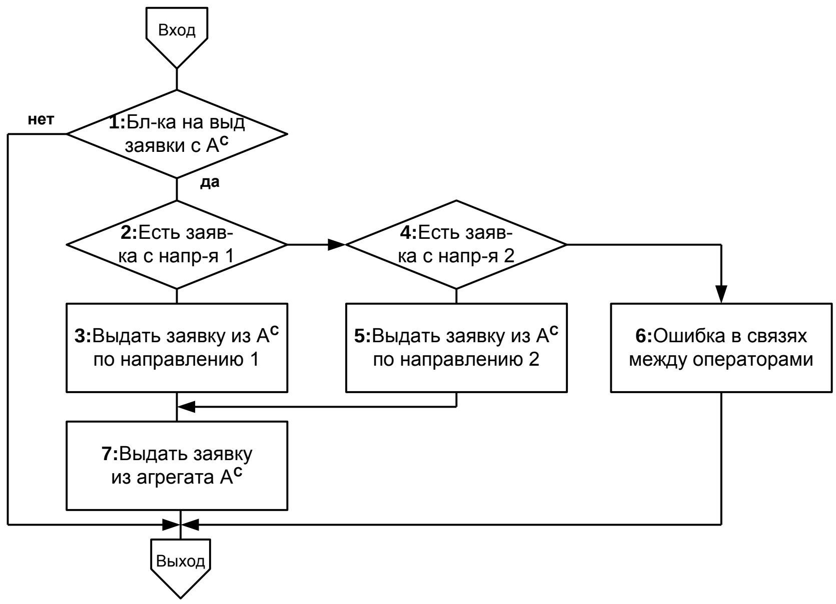

Рис.6.15. Схема алгоритма блока $\boldsymbol{A^C}$

7. Моделирование при исследовании и проектировании АСОИ

Математические методы и модели, используемые при исследовании и проектировании АСОИ

При проектировании АСОИ математические методы исследования процессов их функционирования находят широкое применение. С появление таких систем моделирования как MATLAB и ее расширения Simulink, а также систем SolidWorks, Kosmos и других позволяет значительно сократить время отработки моделей и выполнения расчетных исследований, повысить точность и качество проектирования.

Приведем краткую характеристику некоторых систем моделирования.

Уже первые версии системы MATLAB обладали мощными средствами. В области математических вычислений:

- матричные, векторные, логические операторы;
- элементарные и специальные функции;
- полиномиальная арифметика;
- многомерные массивы;
- массивы записей;
- массивы ячеек.

В области реализации численных методов:

- дифференциальные уравнения;
- вычисление одномерных и двумерных квадратур;
- поиск корней нелинейных алгебраических уравнений;
- оптимизация функций нескольких переменных;
- одномерная и многомерная интерполяция.

В области программирования:

- свыше 500 встроенных математических функций;
- ввод/вывод двоичных и текстовых файлов;
- применение программ, написанных на Си и ФОРТРАН;
- автоматическая перекодировка процедур MATLAB в тексты программ на языках Си и С++;
- типовые управляющие структуры.

В области визуализации и графики:

- возможность создания двумерных и трехмерных графиков;
- осуществление визуального анализа данных.

Эти средства сочетались с открытой архитектурой систем, позволяющей изменять уже существующие функции и добавлять свои собственные. Входящая в состав MATLAB программа Simulink дает возможность имитировать реальные системы и устройства, задавая их моделями, составленными из функциональных блоков. Simulink имеет обширную и расширяемую пользователями библиотеку блоков и простые средства задания и изменения их параметров.

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

Simulink является достаточно самостоятельным инструментом MATLAB и при работе с ним совсем не требуется знать сам MATLAB и остальные его приложения. С другой стороны доступ к функциям MATLAB и другим его инструментам остается открытым и их можно использовать в Simulink. Часть входящих в состав пакетов имеет инструменты, встраиваемые в Simulink (например, LTI-Viewer приложения Control System Toolbox – пакета для разработки систем управления). Имеются также дополнительные библиотеки блоков для разных областей применения (например, Power System Blockset – моделирование электротехнических устройств, Digital Signal Processing Blockset – набор блоков для разработки цифровых устройств и т.д).

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

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

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

Систему MatLab (Matrix Laboratory – матричная лаборатория) можно отнести к среде проектирования инженерных приложений.

MatLab – это средство математического моделирования, обеспечивающее проведение исследований практически во всех известных областях науки и техники. При этом структура пакета позволяет эффективно сочетать оба основных подхода к созданию модели: аналитический и имитационный.

Пользователь может создавать средствами MatLab собственный графический интерфейс, отвечающий как его вкусам, так и требованиям решаемой задачи.

MatLab содержит богатейшую библиотеку функций, количество которых достигает 800. Для удобства поиска библиотека функций разбита на разделы. Функции, которые носят общий характер, входят в состав ядра MatLab, другие включены в состав соответствующих специализированных разделов. Эти разделы называются Toolboxes (наборы инструментов) и содержит 30 наборов инструментов. В их число входят как стандартные средства (решение дифференциальных и алгебраических уравнений, интегральное исчисление, символьные вычисления и т.д.), так и нетрадиционные: средства цифровой обработки изображений, поиска решений на основе нечеткой логики, аппарат построения и анализа нейронных сетей, средства финансового анализа и ряд других.

Раздел вычислительных приложений содержат:

- пакет вычислительной математики, содержащий элементарные и специальные функции, функции решения систем линейных уравнений и уравнений в частных производных;
- пакет символьной математики Symbolic Math Toolbox, позволяющий выполнять символьные аналитические преобразования;
- пакет статистической обработки, аппроксимации и сглаживания экспериментальных данных, аппроксимации на основе сплайнов.
- пакет нечеткой логики и операций на размытых множествах;

Система управления и идентификации представляет важнейший раздел, связанный с проектированием современных технических систем с обратной связью. Включает базовые пакеты идентификации параметров линейных динамических моделей, синтеза и анализа систем управления, а также набор блоков для проектирования нелинейных систем.

Раздел обработки сигналов и изображений содержит средства для проектирования цифровых фильтров и цифровой обработки сигналов, а также импульсной декомпозиции сигналов изображений

Система передачи сигналов и связи - предназначена для расчета и моделирования телекоммуникационных систем. Содержит средства для анализа систем связи и коммуникаций, а также наборы блоков для моделирования таких систем.

Финансовые приложения - содержат средства поддержки финансовой аналитики, анализа временных финансовых рядов и регрессионного анализа.

Раздел моделирования приложений содержит наборы специализированных блоков для моделирования динамики аэрокосмических летательных аппаратов, механических и электрических систем, а также имитации виртуальной реальности.

Раздел подготовки проектной документации содержит генераторы отчетов MatLab.

Кроме того, имеются средства взаимодействия с популярными офисными продуктами компании Microsoft – MS Word и MS Excel.

Excel Link - позволяет использовать MS Excel как процессор ввода-вывода MATLAB. Для этого достаточно установить в Excel как add-in функцию поставляемый Math Works файл excllinkxla. В Excel нужно набрать Сервис > Надстройки > Обзор, выбрать файл в каталоге \matlabrl2\toolbox\exlink и установить его. Теперь при каждом запуске Excel появится командное окно MATLAB, а панель управления Excel дополнится кнопками getmatrix, putmatrix, evalstring. Для закрытия MATLAB из Excel достаточно набрать =MLC1ose() в любой ячейке Excel;

MATLAB Compiler - компилятор для программ, создаваемых на языке программирования системы MATLAB. Транслирует коды этих программ в программы на языке Си++. Применение компилятора обеспечивает возможность создания исполняемых кодов (полностью законченных программ), время выполнения которых для программ с большим числом циклических операций уменьшается в 10-15 раз. Может интегрироваться в среду Microsoft Visual Studio и использоваться вместе с Visual C++.

Real Time Windows Target и Workshop - подсистема имитационного моделирования в реальном масштабе времени;

Neural Networks Toolbox - пакет прикладных программ, содержащих средства для построения нейронных сетей, базирующихся на поведении математического аналога нейрона;

Fuzzy Logic Toolbox - пакет прикладных программ теории нечетких (размытых) множеств;

Symbolic Math Toolbox - пакет решения задач в символьном (аналитическом) виде;

Statistics Toolbox - пакет прикладных программ по статистике;

Optimization Toolbox - пакет прикладных программ для решения оптимизационных задач и систем нелинейных уравнений;

Partial Differential Equations Toolbox - пакет прикладных программ, содержащий функции для решения систем дифференциальных уравнений в частных производных;

Control System Toolbox - пакет для моделирования, анализа и проектирования непрерывных и дискретных систем автоматического управления;

Robust Control Toolbox - включает средства для проектирования и анализа многопараметрических устойчивых систем управления;

Stateflow - пакет моделирования событийно-управляемых систем, основанный на теории конечных автоматов;

LMI (Linear Matrix Inequality) Control - обеспечивает интегрированную среду для постановки и решения задач линейного программирования;

Пакет System Identification содержит средства для создания математических моделей динамических систем на основе наблюдаемых входных и выходных данных;

Пакет Image Processing предоставляет ученым, инженерам и даже художникам широкий спектр средств для цифровой обработки и анализа изображений;

Power System Blockset - пакет моделирования мощных энергетических (в основном электротехнических) систем, таких как линии передачи, силовые ключи, регуляторы напряжения и тока, устройства управления электродвигателями различного типа и нагревательными системами;

Система визуального моделирования Simulink

Среди наборов инструментов система Simulink занимает особое место. Она содержит большое количество функций, а также использует многие функции, входящие в состав пакета MatLab. Simulink – это раздел моделирования и проектирования систем, который содержит набор инструментальных средств моделирования динамических систем, систем реального времени, событийного моделирования, подготовки кодов для микропроцессоров и т.д.

В Simulink используются в основном файлы трех типов:

- M-файлы – (с расширением .m) – файлы, содержащие тексты программ на языке MatLab. В виде m-файлов реализованы все библиотечные функции MatLab;

- Mdl- файлы (с расширение .mdl) - файлы моделей;

- Mat-файлы - (с расширение .mat) – файлы, содержащие данные в двоичном коде, доступ к которым возможен либо из командного окна MatLab, либо и помощью специальных средств Simulink.

Этапы построения модели в подсистеме Simulink

Перед построение модели необходимо загрузить систему MatLab и запустить подсистему Simulink.

Подсистему Simulink можно загрузить из основного окна MatLab Command Window путем ввода команды

>> simulink

или щелкнув левой кнопкой мыши на кнопке запуска этой подсистемы. Откроется окно Simulink Library Browser – система просмотра библиотек Simulink. В верхней части этого окна две крайние левые кнопки служат для создания новой и открытия существующей модели соответственно. После щелчка на первой кнопке появится пустое окно *untitles* для построения новой модели.

Процесс построения модели Simulink включает в себя компоновку и задания необходимых параметров. Компоновка заключается в выборе из библиотек Simulink необходимых блоков, размещении их в окне *untitles* и задании межблочных связей. Затем для каждого блока необходимо установить соответствующие параметры, отвечающие требованиям моделируемой системы.

Библиотека блоков Simulink

Основным строительным элементом в процессе построения в пакете Simulink является блок. Блок представляет собой систему типа >>вход-состояние-выход>> или просто >>вход-выход>> и может быть как простым, так и составным.

Каждый блок из любой библиотеки блоков пакета Simulink является классом. Как только блок буде перенесен из библиотеки в окно построения модели, этот блок становится экземпляром данного класса. В этом экземпляре можно изменять значения параметров блока в зависимости от требований, предъявляемых к моделируемой системе.

Для перемещения блока в окно модели необходимо курсор мыши установить на этом блок и нажав левую кнопку мыши перетащить в нужное место этого окна.

Чтобы установить параметры блока необходимо выполнить двойной щелчок левой кнопкой мыши на пиктограмме этого блока. В открывшемся окне параметров выбранного блока следует установить необходимые параметры.

Модели, разрабатываемы средствами Simulink, называют S-моделями. S-модель может иметь иерархическую структуру, т.е состоять из моделей более низкого уровня, Число уровней иерархии практически не ограничено.

В ходе моделирования имеется возможность наблюдать за процессами, происходящими в системе с помощью специальных >>смотровых окон>>. Кроме этого, существует возможность включения в состав модели средств анимации.

Библиотека Simulink является открытой и может пополняться пользователем за счет разработки собственных блоков.

Библиотека блоков Simulink представляет собой набор визуальных объектов, из которых можно собирать произвольную конструкцию. Для блоков существует возможность настройки их внутренних параметров и внешнего оформления (размер, цвет, имя и т.д.).

Основная библиотека блоков разбита на следующие разделы:

1. Continuous – включает непрерывные элементы, такие как Integrator, Derivative (Дифференциатор), State-Space (Состояния) и другие, задаваемые с помощью передаточных функций;

2. Discrete – содержит дискретные блоки: для моделирования дискретных систем;

3. Functions & Tables – содержит блоки, позволяющие реализовать функции, отсутствующие в библиотеке математических функций, и работать с табличными функциями, например Fcn (функция пакета Matlab), функции линейной интерполяции и другие;

4. Math – содержит блоки, реализующие математические функции;

5. Nonlinear – содержит блоки, реализующие нелинейные функции, такие как Relay (Релейное звено), Switch (ключ) и другие весьма опасные функции, если они используются в правых частях решаемых дифференциальных уравнений;

6. Signals & Systems – сигналы и системы: для создания подсистем и управления сигналами; содержит такие функции, как Mux (Мультиплексор), DeMux (Демультиплексор), In1 (Входной сигнал), Out1 (Выходной сигнал) и другие

7. Sinks – регистрирующие устройства, содержит средства отображения сигналов, возникающих на выходе блоков: Scope (осциллограф), To File (вывод результатов в файл), Stop (Остановка выполнения);

8. Sources – источники сигналов и воздействий, такие как Signal Generator (Генератор сигналов), Random Number (Генератор случайных чисел), Repeating Sequence (Генератор пилообразных сигналов), Clock (Часы) и т.д.;

9. Subsystems – блоки подсистем.

Моделирование в системе SIMULINK

Для моделирования поведения динамических систем, к которым относятся экипажи подвижного состава, используются ЭВМ. Существует большое количество алгоритмических языков, на которых может быть выполнено решение задачи. Выбор того или иного языка программирования зависит от многих условий. Часто решающую роль оказывает удобство программирования, наличие проверенных математических методов, легкость представления результатов моделирования. Такими особенностями обладает пакет MATLAB, содержащий в своем составе инструмент визуального моделирования SIMULINK.

SIMULINK сочетает в себе наглядность аналоговых машин и точность цифровых вычислительных машин. SIMULINK обеспечивает пользователю доступ ко всем возможностям пакета MATLAB, в том числе к большой библиотеке численных методов.

Подготовка задачи для моделирования в SIMULINK проводится в следующей последовательности:

Выбор расчетной схемы.

Составление системы уравнений, описывающих исследуемый процесс.

Приведение системы к виду, удобному для решения (разрешение относительно старших производных).

Определение начальных условий.

Составление структурной схемы.

Моделирование возмущающих функций.

Определение исходных данных.

Составление модели в среде SIMULINK.

Включение средств визуализации.

Тестирование.

Решение.

Анализ результатов.

Отчет.

Рассмотрим пример моделирования в среде SIMULINK вертикальных колебаний экипажа

Математическая модель вынужденных колебаний двухмассовой системы

Для исследования влияния основных параметров экипажа на вертикальные колебания используют упрощенную модель с двумя степенями свободы, в которой две массы связаны упругими и диссипативными связями (рис.1). Такая модель описывает вертикальные колебания рельсовых экипажей с двухъярусным подвешиванием: магистральных локомотивов (электровозов и тепловозов) и пассажирских вагонов.]

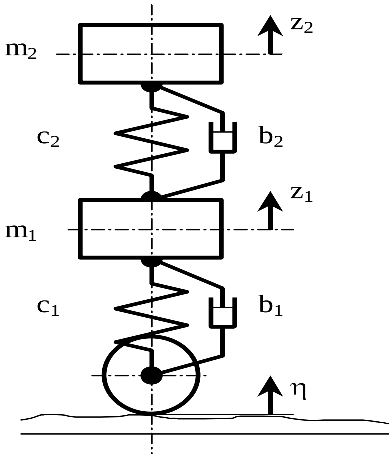

Рис.1. Расчетная схема

Уравнения движения рассматриваемой системы при наличии возмущения со стороны пути описывается следующими дифференциальными уравнениями:

$$m_1 \cdot \ddot{z}_1 + b_1 \cdot \dot{z}_1 + b_2 \cdot (\dot{z}_1 - \dot{z}_2) + c_1 \cdot z_1 + c_2 \cdot (z_1 - z_2) = b_1 \cdot \dot{\eta} + c_1 \cdot \eta \; ; \quad (1)$$

$$m_2 \cdot \ddot{z}_2 + b_2 \cdot (\dot{z}_2 - \dot{z}_2) + c_2 \cdot (z_2 - z_1) = 0 \, .$$

В уравнениях (1) введены следующие обозначения:

m_1 – обрессоренная масса тележки;

m_2 – масса кузова, приведенная к одной тележке;

c_1, b_1 – жесткость и демпфирование в первом ярусе подвешивания;

c_2, b_2 – жесткость и демпфирование во втором ярусе подвешивания;

$\eta(t)$ – возмущение со стороны пути;

$z_i\, ,\, \dot{z}_i\, ,\, \ddot{z}_i$ – обобщенные координаты и их производные по времени:

Преобразуем уравнения движения к виду:

$$\ddot{z}_1 = -\frac{1}{m_1}(b_1 \cdot \dot{z}_1 + b_2 \cdot (\dot{z}_1 - \dot{z}_2) + c_1 \cdot z_1 + c_2 \cdot (z_1 - z_2) - b_1 \cdot \dot{\eta} - c_1 \cdot \eta)$$

$$\ddot{z}_2 = -\frac{1}{m_2}(b_2 \cdot (\dot{z}_2 - \dot{z}_2) + c_2 \cdot (z_2 - z_1)) \quad (2)$$

В качестве возмущения используем неровность проф. Н.Н.Кудрявцева. Неровность хорошо описывает изменение прогиба вдоль рельсового звена. Модель неровности представляет собой сумму полуволны синусоиды частотой □ и трех полуволн синусоиды частотой 3□, уложенные на длине рельсового звена L. Амплитуды неровностей A_1 A_2 выбираются в зависимости от типа и состояния пути.

$$\eta(t) = \left| A_1 \cdot Sin(\omega \cdot t) + A_1 \cdot Sin(3 \cdot \omega \cdot t) \right|, \quad (3)$$

где

$\omega = \dfrac{\pi}{L} \cdot V$ – частота возмущения;

V - скорость движения.

Построим описанную выше модель в среде SIMULINK.

Модель вынужденных колебаний двухмассовой системы в системе Simulink

При запуске SIMULINK открываются два окна:

- - пустое рабочее окно – заготовка для создания новой модели (*untitled*);
- - окно библиотеки SIMULINK, содержащей наборы основных разделов (*Library: simulink*).

Далее можно раскрыть необходимые разделы библиотеки SIMULINK (рис. 2).

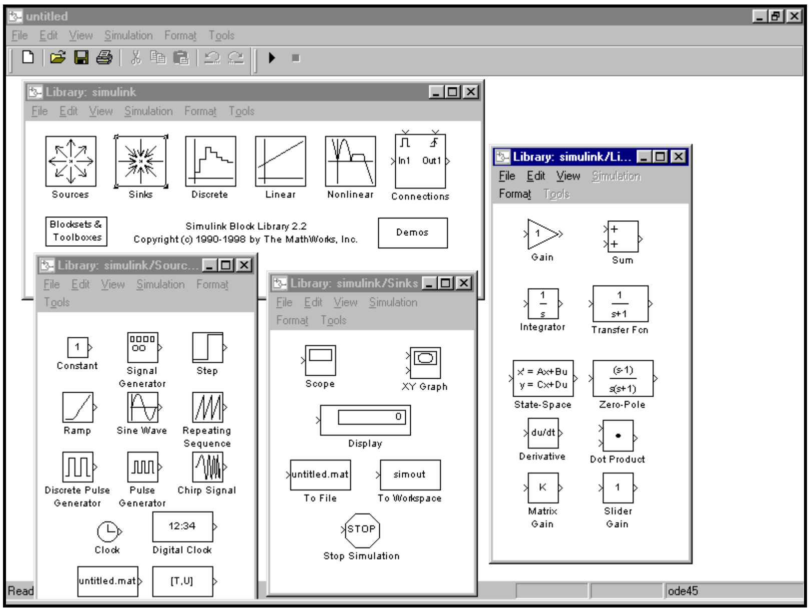

Рис. 2. Пример начала работы в SIMULINK

Поскольку моделируемая система довольно проста, покажем реализацию модели одного уровня (без вложенных подсистем).

Процесс построения модели в системе SIMULINK представляет последовательность выбора необходимых блоков из соответствующих библиотек, и соединение их связями.

Обычно для моделирования динамической системы используют уравнения движения в виде (2).

Построение модели каждого уравнения начинаем с сумматора, имеющего столько входов, сколько членов содержит правая часть уравнение. Для первого уравнения это сумматор *Sum1* (рис. 3). Входы сумматора могут иметь как положительное значение, так и отрицательное (рис. 4). Используемые блоки рекомендуется именовать для облегчения последующей проверки и анализа.

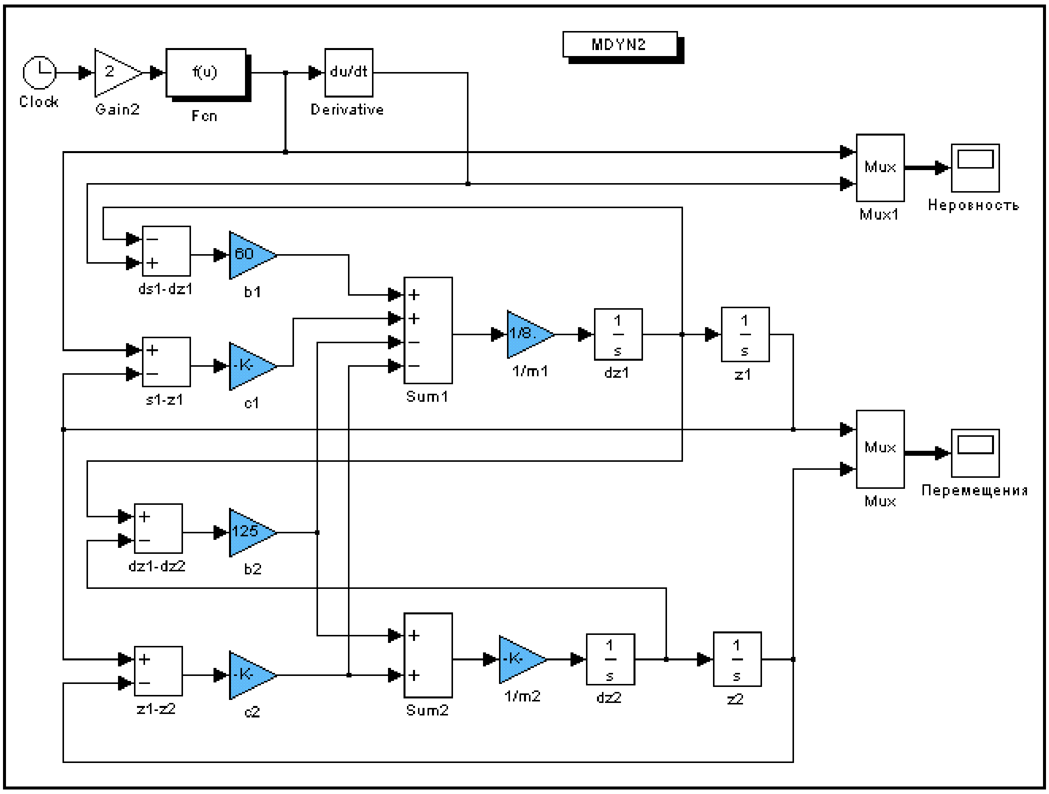

Рис. 3. Блок-схема модели

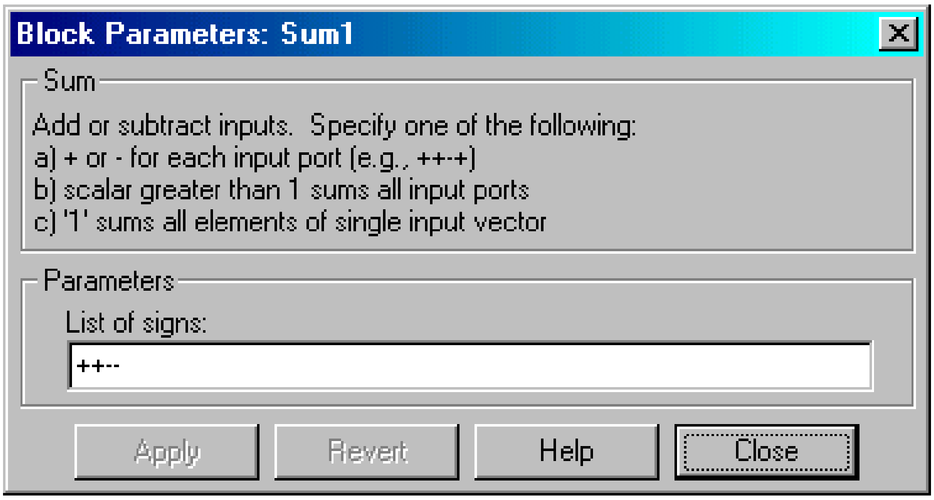

Рис. 4. Настройка сумматора

К выходу сумматора подключаем линейный преобразователь (множитель *1/m1*), на выходе которого получаем значение второй производной. Далее включаем последовательно два интегратора (*dz1*, *z1*), на выходе которых получаем значения первой производной и самой переменной.

В нашем случае имеем два уравнения, соответственно процесс повторяем для второго уравнения.

Далее устанавливаем связи между входами и выходами соответствующих блоков, применяя, где необходимо, дополнительные линейные преобразователи и сумматоры.

Для моделирования возмущения используем функциональный блок *Fcn*, в котором в аналитическом виде запишем зависимость перемещения неровности от времени. Блок имеет вход, на который через линейный преобразователь подается от "таймера" независимая переменная - время.

Для вычисления производной перемещения неровности используется блок дифференцирования.

После установки всех связей определяем необходимые коэффициенты в используемых преобразователях.

Для моделирования использованы следующие исходные данные:

амплитуда 1-й гармоники неровности	$A_1 = 0.005$	м;
амплитуда 2-й гармоники неровности	$A_2 = 0.002$	м;
длина рельсового звена	$L = 25$	м;
масса первого тела	$m_1 = 8.82$	т;
жесткость первого яруса	$c_1 = 7000$	кН/м;
демпфирования первого яруса	$b_1 = 60$	кН с/м;
масса второго тела	$m_2 = 25.8$	т;
жесткость второго яруса	$c_2 = 2600$	кН/м;
демпфирования второго яруса	$b_2 = 125$	кН с/м.

Зададим значения коэффициентов в соответствующих блоках (рис. 5).

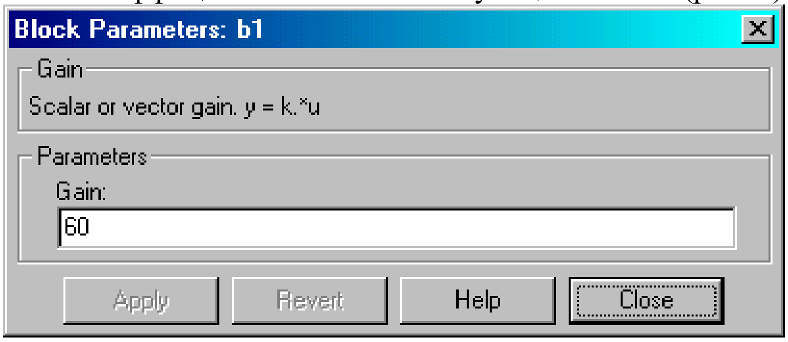

Рис. 5. Задание коэффициента линейного преобразователя

Настроим интеграторы, зададим нулевые начальные условия (рис. 6).

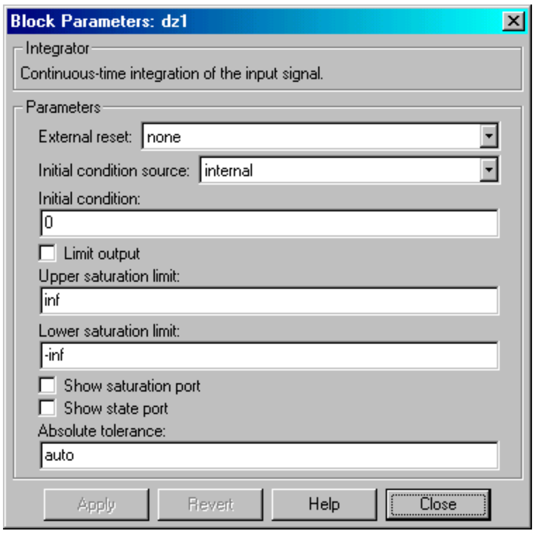

Рис. 6. Панель настойки интегратора

Установим параметры интегрирования через меню (рис. 7).

Начало интегрирования – 0, конец интегрирования – 10. Используем метод *ODE45* с переменным шагом интегрирования.

В соответствие с формулой (3) запишем функцию возмущения в блоке *Fcn* (рис. 8).

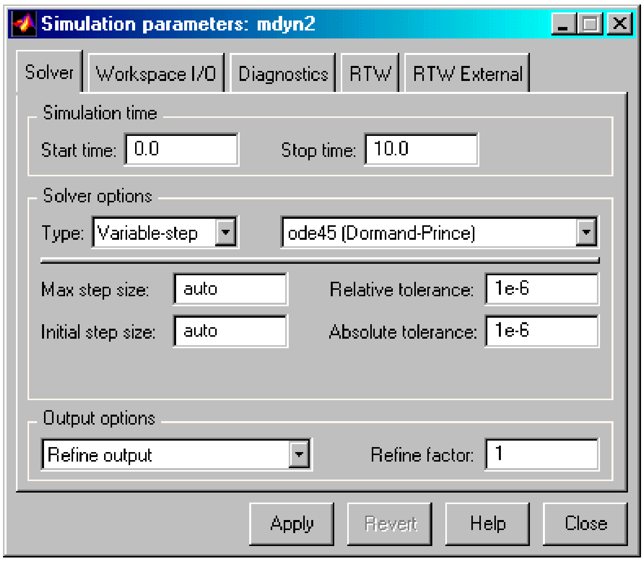

Рис. 7. Параметры интегрирования

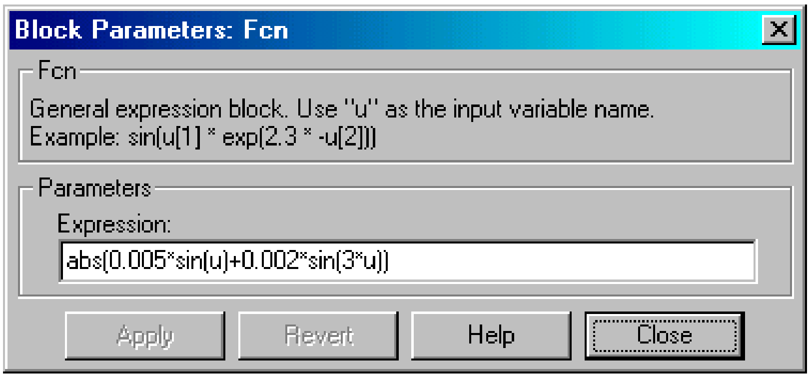

Рис. 8. Функция возмущения

Для наблюдения за процессами в модели установим "осциллограф" - блок *Scope*, обозначенный *"Перемещения"*, для отображения изменения переменных во времени.

Модель готова.

Приступаем к тестированию.

Проведем моделирование для заданных исходных данных.

Результаты моделирования представим в графическом виде (рис. 9).

При тестировании оцениваем соответствие модели поставленной задачи.

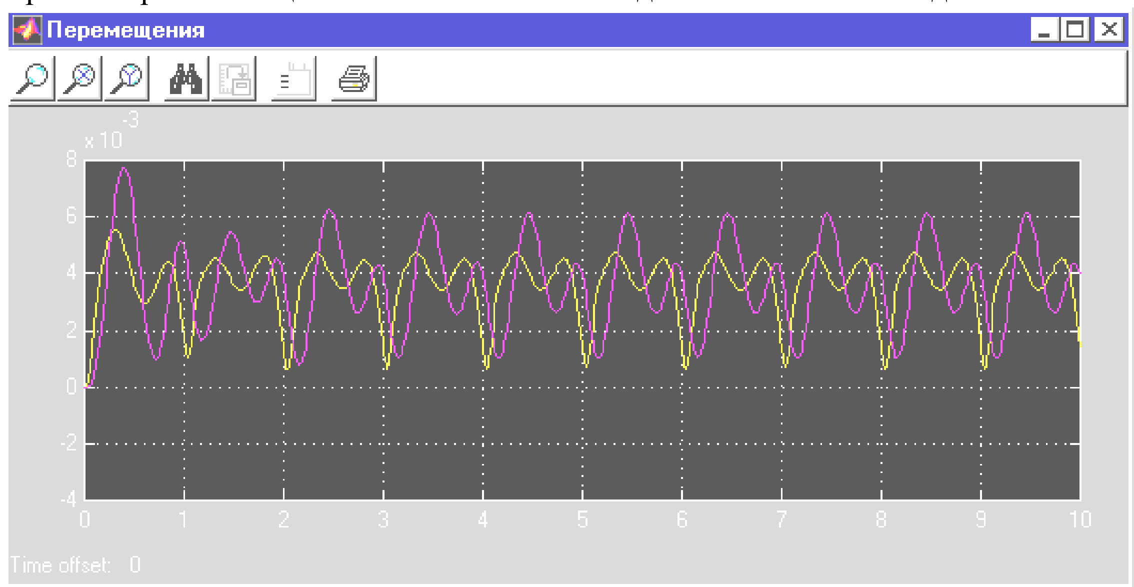

Рис. 9. Результаты моделирования – перемещения

Использование подсистем

Для моделирования простых систем достаточно использовать подход описанный выше. При увеличении размерности для увеличения наглядности и удобства отладки выделяют логические блоки – подсистемы. SIMULINK поддерживает многоуровневые системы. Для вложенных подсистем используют блок Subsystem (рис. 10). При раскрытии такого блока выводится окно с моделью соответствующей подсистемы (рис. 11, 12). Для передачи входных и выходных переменных используются так называемые порты подсистемы входные и выходные. В приведенном примере используются многоканальные порты – через один порт передается переменная и ее производная. Для создания «сложной переменной» (на блок-схеме такие переменные показаны жирными линиями) применяется блок *Mux*, для разделения – блок *Demux*.

Для большой разветвленной модели предпочтительно исходные данные собрать в одном месте, что облегчает настройку при исследовании влияния параметров системы. Один из вариантов – использование модуля присвоения начальных значений переменным MATLAB и использование в модулях SIMULINK имен этих переменных. В модели, показанной на рис. 10, в модуле *START* используется ссылка на подпрограмму в которой осуществляется присвоение начальных значений (рис. 13).

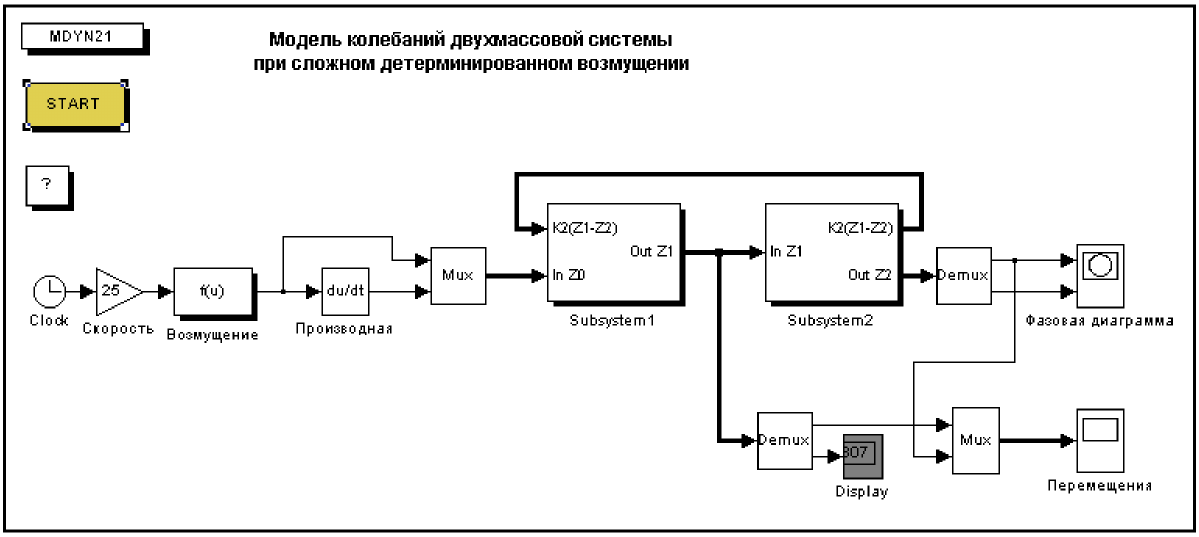

Рис. 10. Модель системы

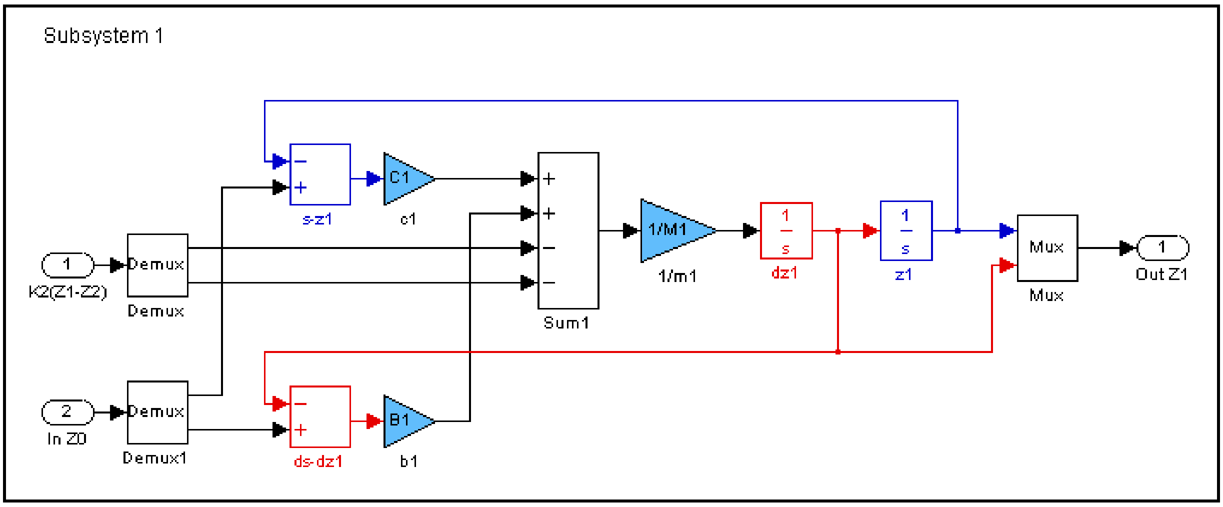

Рис. 11. Подсистема, описывающая колебания первого тела

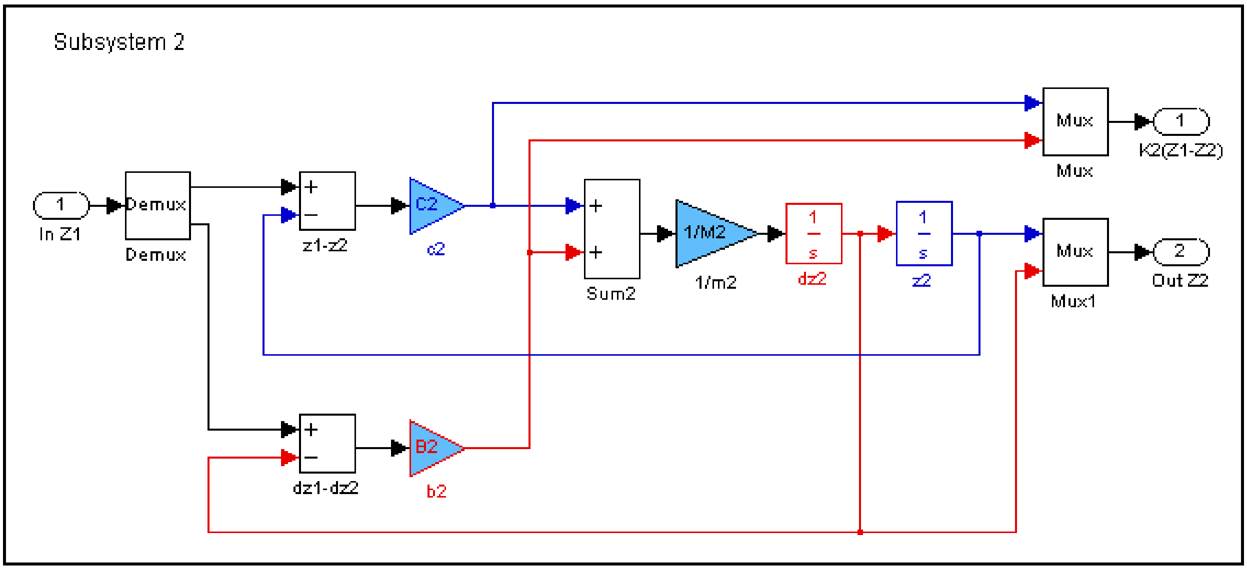

Рис. 12. Подсистема, описывающая колебания второго тела

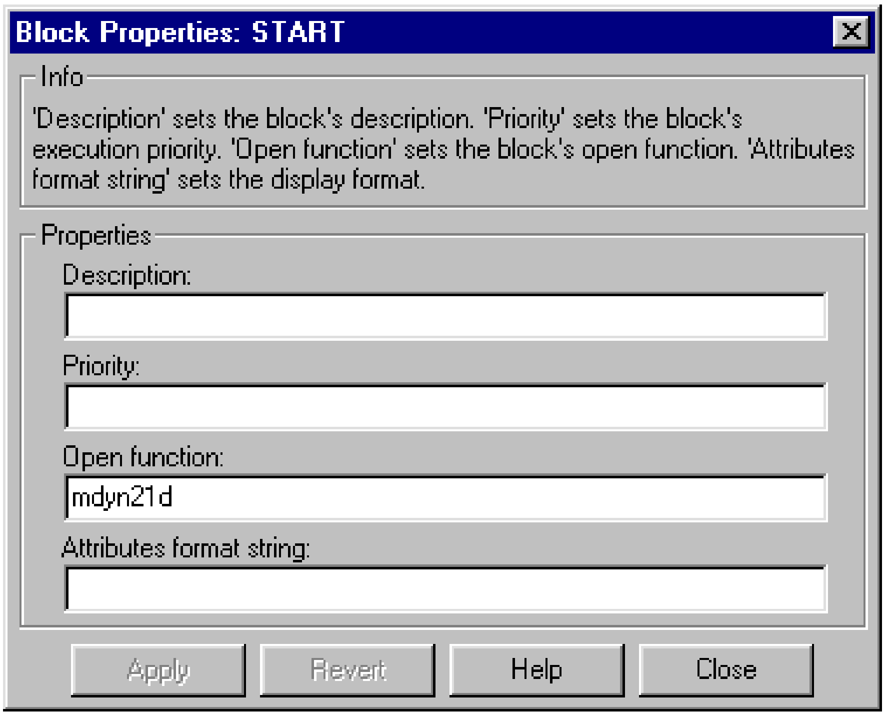

Рис. 13. Настройка запускающей программы

Подпрограмма задания начальных значений – обычный m-файл. Для данного примера в файле *MDYN21.m* задаются следующие значения:

```
% MDYN21
% Start programm
%
echo on
A1=0.005;
A2=0.002;
L=25;
p=pi/L;
M1=8.82;
C1=7000;
B1=60;
M2=25.8;
C2=2600;
B2=125;
```

```
echo off
x0 = [0; 0; 0; 0;];
[t,x]=sim('mdyn21',10,simset('InitialState',x0));
```

В приведенном тексте подпрограммы знак "точка с запятой" в конце строки запрещает вывод значений переменных.

Для отображения введенных данных применены операторы «*echo on – echo off*». Выделенный фрагмент подпрограммы выводится в управляющее окно MATLAB (рис 14).
Для изменения исходных данных используется текстовый редактор.

Запуск моделирующей программы в этом примере осуществляется двойным нажатием на блок *START*.

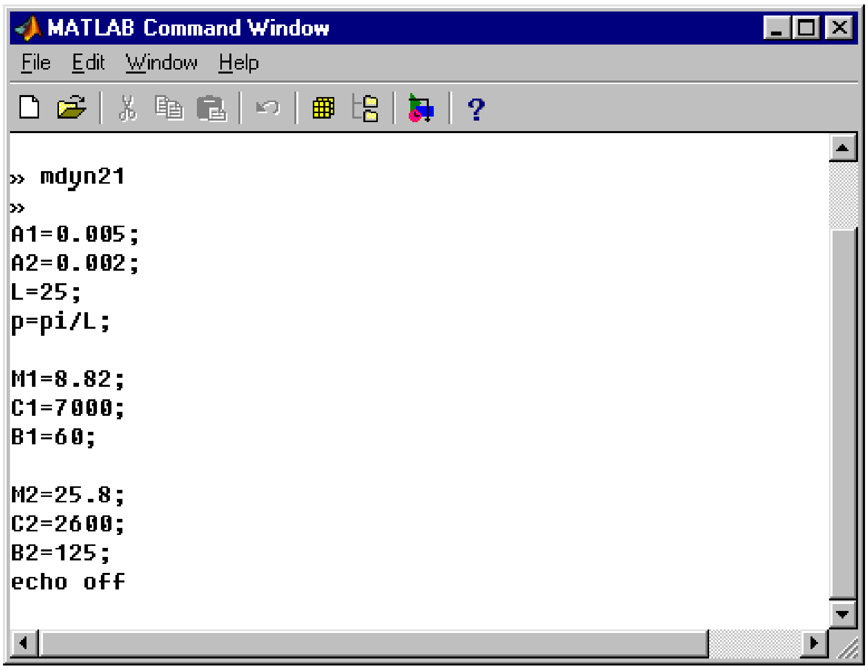

Рис. 14. Вывод в окне управляющей программы MATLAB

Добавим в модель информационный блок, содержащий краткое описание модели – блок *Info*, обозначенный символом «**?**». Содержание блока показано на рис. 15.

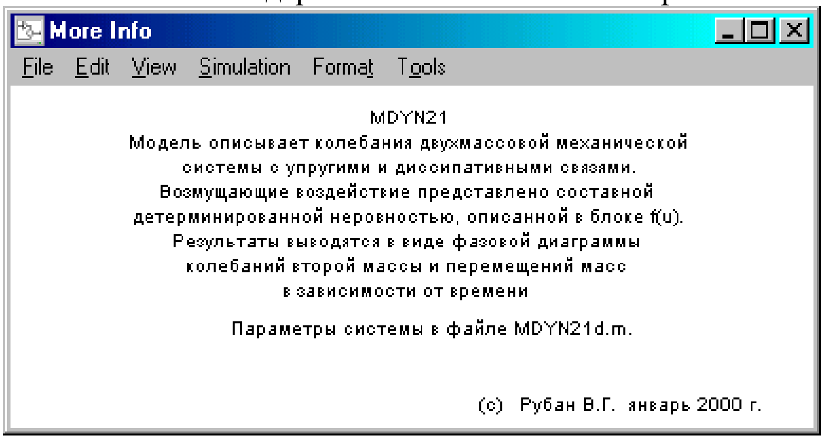

Рис. 15. Информационный блок

Для отображения фазовой диаграммы введем блок отображения *XY_Graph*, обозначенный на блок-схеме "*фазовая диаграмма*". Результаты вывода показаны на рис. 16.

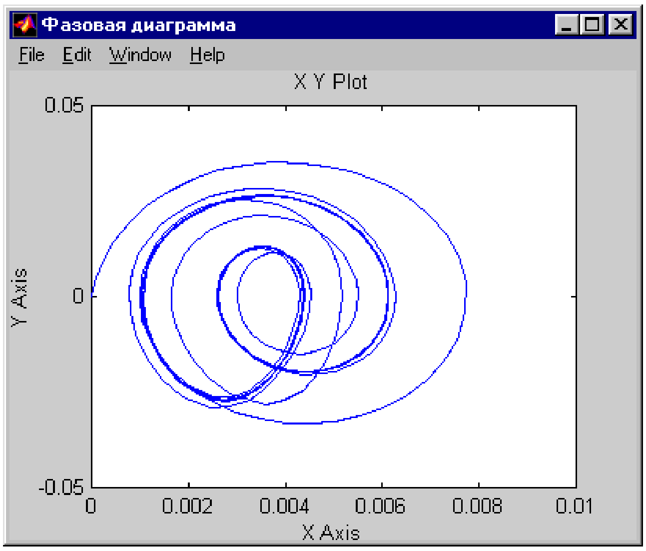

Рис. 16. Результаты моделирования – фазовая диаграмма

Использование подпрограмм пользователя

Для расширения возможностей моделирования в среде SIMULINK предусмотрена возможность подключения подпрограмм пользователя написанных на языке MATLAB. В предыдущей модели заменим функции неровности модулем "*MATLAB Function*" - "*Неровность*". Модель определяет ссылку на подпрограмму пользователя, в которой описана функция неровности от пути и ее производная (рис. 17). Такой подход позволяет использовать проверенные модули как при программировании в пакете MATLAB, так и при моделировании в среде SIMULINK.

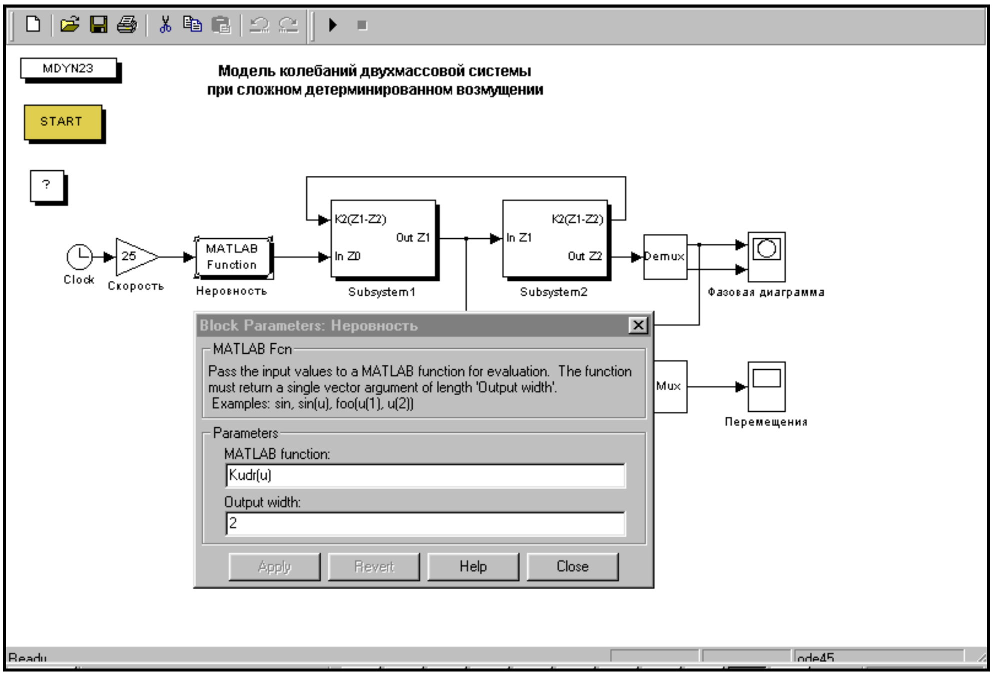

Рис. 17. Модель и настройка блока функции MATLAB

Рекомендуемая литература (основная)

1.Советов Б.Я., Яковлев С.А. Моделирование систем: Учебник для ВУЗов по специальности АСУ.- М.:Высшая школа, 2003.

2.Бусленко Н.П. Моделирование сложных систем. - М. :Наука, 1978.

3.Рябов В.Ф., Советов Б.Я., Яковлев С.А. Машинное имитационное моделирование при проектировании больших систем /Учеб. пособие. - Л.:Мир, 1980.-272 с.

4.Шеннон Р. Имитационное моделирование: искусство и наука /Пер. с англ. - М.:Мир, 1978.

5.Максимей И.В. Математическое моделирование больших систем: Учеб. пособие для спец. "Прикладная математика". - Мн.:Выш. школа, 1985.-119 с.

6.Советов Б.Я., Яковлев С.А. Моделирование систем. Курсовое проектирование. - М.:Выс. школа, 1988.

7.Советов Б.Я., Яковлев С.А. Моделирование систем: Лабораторный практикум: Учебн. пособие для ВУЗов по спец. "Автом. сист. обраб. инф. и управл." - М.:Высш. школа, 1989. - 80 с.:ил.

8.Герасимович А.И. Математическая статистика: Учеб. пособие для инж. техн. и экон. спец. ВТУЗов. - 2-е изд., перераб. и доп. - МН.:Выш. школа, 1983.- 279 с.

9.Микулик Н.А., Рейзина Г.Н. Решение технических задач по теории вероятностей и математической статистике. - Мн.:Выш. школа, 1991.

10.Иванов В.М. Случайные числа и их применение. - М.:Финансы и статистика, 1984. - 111 с.

11.Справочник по теории вероятностей и математической статистике. /Под ред. В.С. Королюка.- Киев: Наукова Думка, 1978.

12.Полонников Р.И., Никандров А.В. Методы оценки показателей надежности программного обеспечения. - СПб. :Политехника, 1992. - 78 с.:ил.

13.Спиридонов А.А. Планирование эксперимента при исследовании технологических процессов. – М., :Машиностроение, 1981. – 181 с.

14.Микулик Н.А., Рейзина Решение технических задач по теории вероятностей и математической статистике. – Мн. :Выш. шк. – 1991.

Дополнительная литература к теме 6

1. Шакин В.Н., Воробейчиков Л.А., Шибанов С.Е., Семенова Т.И. Моделирование систем и сетей связи: Учебное пособие/МИС.- М., 1988.

2. Игельник Б.М., Лившиц В.М., Шибанов С.Е. Аналитическое моделирование систем связи: Учебное пособие/МИС. - М., 1989.

3. Шакин В.Н., Лившиц В.М. Принципы построения локальных сетей и анализ их характеристик: Учебное пособие для слушателей ФПКП/ МИС. - М., 1990.

4. Методические указания по использованию средств имитационного моделирования систем и сетей связи для слушателей ФПКП/ Л.А.Воробейчиков, В.Н.Шакин, С.Е.Шибанов/МИС. - М., 1990.

5. Шеннон Р. Имитационное моделирование систем - искусство и наука: Пер. с англ. - М.: Мир, 1978.

6. Максимей И.В. Имитационное моделирование на ЭВМ. - М.: Радио и связь, 1988.

7. Шрайбер Т.Дж. Моделирование на GPSS: Пер. с англ. - М.: Машиностроение, 1980.

8. GPSS/PC General Purpose Simulation. Reference Manual. - Minuteman software. P.O. Box 171. Stow, Massachusetts 01775, 1986.

Дополнительная литература к теме 7

1 Ануфриев И.Е. Самоучитель MatLab 5.3/6x – СПб.: БХВ–Петербург, 2002-736.; ил.

2 Потемкин В.Г. MatLab 6 : средства проектирования инженерных приложений. – М.:Диалог – МИФИ, - 2003. – 448 с.

3 Бенькович Е.С., Колесов Ю.Б., Сениченков Ю.Б. Практическое моделирование динамических систем – Саб.:БХВ-Перербург, 2002. – 464 с.:ил.

4 Гуляев А. Виртуальное моделирование в среде MatLab: учебный курс. – СПб: Питер, 2000. – 432 с.:ил.

5 Новгородцев А.Б. Расчет электрических цепей в MatLab: учебный курс. – СПб: Питер, 2004. – 250 с.:ил.

6 Потемкин В.Г. MatLab 6: среда проектирования инженерных приложений. – М.:ДИАЛОГ-МИФИ, 2003. -448 с.

7 Леоненков А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. - СПб.: БХВ-Петербург,2003, - 736 с.:ил.

8 Рубан В.Г. Моделирование в среде Simulink / www.mathmod.narod.ru