Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Институт инновационных технологий

Кафедра “Автотранспортная и техносферная безопасность”

составитель

Сабуров Павел Сергеевич

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Курс лекций

Владимир – 2016 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Вторая половина XX века связана с появлением и широким распространением новой методологии исследования сложных объектов и систем. В ее основе лежит метод математического моделирования и реализованные на его основе вычислительные эксперименты. Математические модели использовались и раньше. Они позволяли уже тогда анализировать недоступные или несуществующие объекты и процессы. Например:

1) Планета Уран была открыта путем анализа возмущений орбит трех планет (Леверье).

2) К.Э. Циолковский показал, что для преодоления земного притяжения требуется первая космическая скорость, а не скорость света.

Однако считалось, что методы математического моделирования не пригодны для исследования сложных технических, экономических, биологических и социальных систем. В области техники отсутствие объективных математических методов привело, с одной стороны, к созданию многочисленных частных, так называемых инженерных методик расчета, носивших рецептурный характер, а с другой – к полному безраздельному господству эмпирики (натурных экспериментов).

Недостаточно полная проработка вариантов приводила к субъективным решениям.

Положение начало меняться во второй половине XX в. при развитии средств вычислительной техники, в частности современных ЭВМ, которое дало в руки исследователей новое эффективное средство моделирования сложных систем. В настоящее время не существует объектов, при изучении которых не применялись бы методы математического моделирования. Разработаны и активно используются математические модели технических устройств, модели разнообразных технологических процессов, экономические модели предприятий, регионов и целых государств, экологические модели, модели геологических и геофизических процессов, модели социальных систем, биологические и медицинские модели.

Глава 1. ЦЕЛИ И ЗАДАЧИ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ПРОЦЕССОВ И СИСТЕМ

1.1. ПОНЯТИЕ «МАТЕМАТИЧЕСКАЯ МОДЕЛЬ»

Математическое моделирование позволяет до создания реальной системы (объекта) или возникновения реальной ситуации рассмотреть возможные режимы работы, выбрать оптимальные управляющие воздействия, составить объективный прогноз будущих состояний системы.

Вычислительные эксперименты, проводимые на основе математических моделей, помогают увидеть за частным общее, развить универсальные методы анализа объектов различной физической природы, познать свойства изучаемых процессов и систем.

Наконец, математическое моделирование является основой интенсивно разрабатываемых автоматизированных систем проектирования, управления и обработки данных.

> *Основная задача математического моделирования – выделение законов в природе, обществе и технике и запись их на языке математики.*

Например:

1) Зависимость между массой тела m, действующей на него силой F и ускорением его движения a записывается в форме 2-го закона Ньютона: $F = m \cdot a$;

2) Зависимость между напряжением в электрической цепи U, ее сопротивлением R и силой тока I записывается в виде закона Ома: $I = U/R$.

Существует множество определений математической модели.

Приведем одно из них:

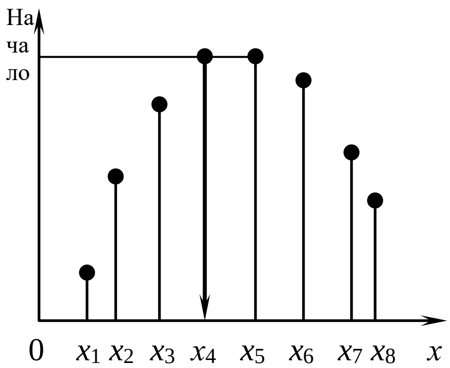
> *Математической моделью некоторого объекта, процесса или явления будем называть запись его свойств на формальном языке с целью получения нового знания (свойств) об изучаемом процессе путем применения формальных методов.*

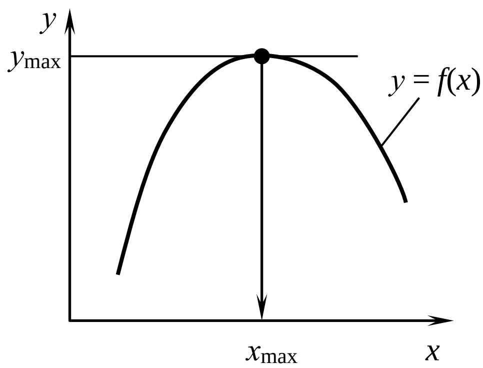
Альтернативой формальному (математическому) подходу является экспериментальный подход. К его недостаткам можно отнести:

1) высокая стоимость подготовки и проведения экспериментов;

2) получение частного знания (знания о конкретном объекте исследования, а не о классе объектов).

Например, пусть требуется определить воздействие x на некоторый процесс или объект, при котором его результирующая характеристика y имеет максимально возможное значение (Рис. 1.1).





а б

Рис. 1.1.

На рис. 1.1. а) показан эмпирический (экспериментальный) подход к решению поставленной задачи, который состоит в экспериментальном определении значения параметра y для нескольких значений входного воздействия x. Среди них найдено наибольшее, и оно принимается за максимум. Как видим из этого рисунка, возможно несколько значений воздействия x (x_4 и x_5), при которых y имеет наибольшее значение, но ни одно из них не является настоящим максимумом, который, возможно, лежит между ними.

Математический подход (рис. 1.1. б) предполагает наличие математической модели процесса типа $y = f(x)$. Взяв производную $\frac{df}{dx}$ и приравняв ее к нулю, получим уравнение, решением которого является точное значение x_{max}, доставляющее максимум функции y.

Схема применения математической модели при решении реальных задач имеет вид, показанный на рис. 1.2.

Рис. 1.2

Модель сложного объекта (процесса, системы) не может быть простой. Из чего следует, что процесс использования математических моделей реальных систем является итерационным процессом, когда последовательно уточняется (дорабатывается) математическая модель и методы решения стоящих задач.

Важнейшей характеристикой моделей является их точность, адекватность действительности. При этом важно иметь в виду, что все модели представляют собой приближенное описание реальных объектов (процессов) и поэтому принципиально неточны. Интегральная оценка модели может быть получена путем сравнения результатов моделирования и экспериментальных данных для конкретных объектов или режимов.

Для оценки значимости совпадения или несовпадения модельных и экспериментальных результатов широко используются методы математической статистики. Вместе с тем не следует переоценивать результаты такой проверки. Хорошее совпадение модельных и экспериментальных данных, вообще говоря, не доказывает точности модели, а лишь подтверждают ее функциональную пригодность для моделирования. Всегда может быть предложена модель, обеспечивающая лучшее совпадение с экспериментом, но не лучшее описание моделируемого объекта или процесса.

1.2. КЛАССИФИКАЦИЯ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ

Существует несколько схем классификации математических моделей. Все они достаточно условны. Одна из таких схем приведена на рис. 1.3.

Рис. 1.3

Все математические модели по использованному формальному языку можно разбить на аналитические и имитационные.

Аналитические – модели, в которых используется стандартный математический язык. Имитационные – модели, в которых использован специальный язык моделирования или универсальный язык программирования.

Аналитические модели могут быть записаны в виде формул или уравнений. Если какой-либо процесс не может быть описан в виде аналитической модели, его описывают с помощью специального алгоритма или программы. Такая модель является имитационной.

Аналитические модели в свою очередь разбиваются на теоретические и эмпирические модели. Теоретические модели отражают реальные структуры и процессы в исследуемых объектах, то есть, опираются на теорию их работы. Эмпирические модели строятся на основе изучения реакций объекта на изменение условий окружающей среды. При этом теория работы объекта не рассматривается, сам объект представляет собой так называемый «черный ящик», а модель – некоторую интерполяционную зависимость. Эмпирические модели могут быть построены на основе экспериментальных данных. Эти данные получают непосредственно на исследуемых объектах или с помощью их физических моделей.

По форме описания аналитические модели подразделяются на линейные и нелинейные.

Если все входящие в модель величины не зависят от времени, то имеем статическую модель объекта или процесса, в противном случае получаем динамическую модель.

В детерминированных моделях все взаимосвязи, переменные и константы заданы точно, что приводит к однозначному определению результирующей функции. Если часть или все параметры, входящие в модель по своей природе являются случайными величинами или случайными функциями, то модель относят к классу стохастических моделей. В стохастических моделях задаются законы распределения случайных величин, что приводит к вероятностной оценке результирующей функции.

Если аналитическое исследование может быть доведено до конца, модели называются аналитически разрешимыми. В противном случае говорят о численно разрешимых аналитических моделях.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что позволяет осуществить математическое моделирование до создания реальной системы, объекта?
2. Что позволяют увидеть вычислительные эксперименты?

3. Сформулируйте основную задачу математического моделирования.
4. Дайте определение математической модели.
5. Какой подход решения научных задач является альтернативным математическому моделированию?
6. Перечислите основные недостатки экспериментального подхода.
7. Что является важнейшей характеристикой математической модели?
8. На какие два вида делятся математические модели?
9. Перечислите виды аналитических математических моделей.
10. Дайте краткую характеристику видов моделей.

1.3. Геометрическое представление математических моделей

Геометрически математическая модель может быть представлена как некоторая ***поверхность отклика****, соответствующая расположению точек $W = W(x)$ в k-мерном факторном пространстве X.*

Наглядно можно представить себе только одномерную и двухмерную поверхности отклика, причем в последнем случае удобно пользоваться топографическим способом изображения рельефа поверхности с помощью линий уровня (изолиний), построенных в двумерном факторном пространстве X. (Рис. 1.4).

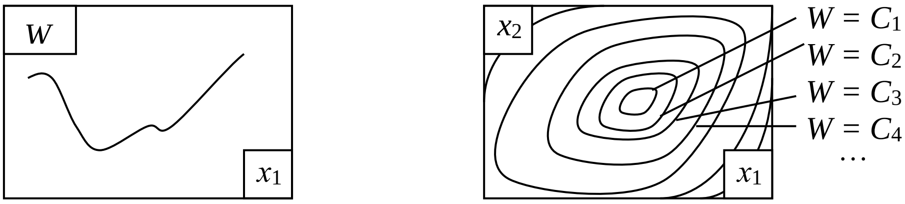


Рис. 1.4

Область, в которой определена поверхность отклика, называется ***областью определения*** *X^*.*

Эта область составляет, как правило, лишь часть полного факторного пространства X ($X^* \subset X$) и выделяется с помощью ограничений, наложенных на управляющие переменные x_i , записанных в виде равенств

$$x_i = C_i\,,\;\; i = 1,\dots, m;$$

$$f_j(x) = C_j\,,\;\; j = 1,\dots, l$$

или неравенств

$$x_{i\,\min} \le x_i \le x_{i\,\max}\,,\;\; i = 1,\dots, k;$$

$$f_j(x) \le C_j\,,\;\; j = 1,\dots, n,$$

При этом функции $f_j(x)$ могут зависеть как одновременно от всех переменных, так и от некоторой их части.

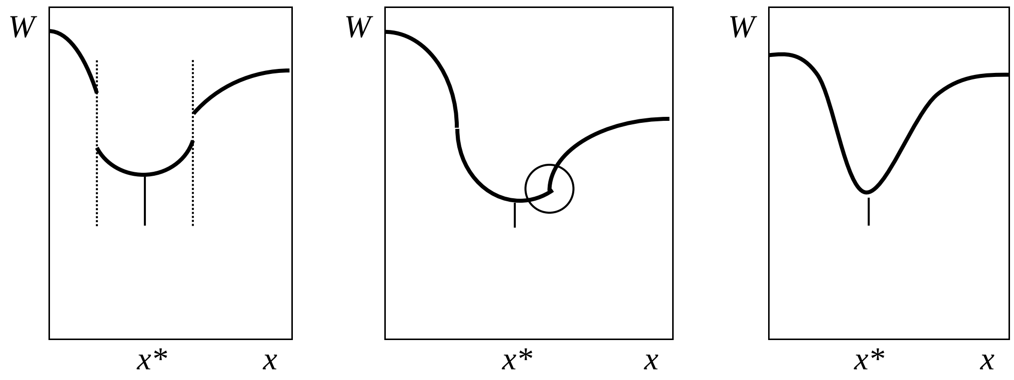
Ограничения типа неравенств характеризуют или физические ограничения на процессы в изучаемом объекте (например, ограничения температуры), или технические ограничения, связанные с условиями работы объекта (например, предельная скорость резания).

Возможности исследования моделей существенно зависят от свойств (рельефа) поверхности отклика, в частности, от количества имеющихся на ней «вершин» и ее контрастности.

Количество вершин (впадин) определяет модальность поверхности отклика.

> *Если в области определения на поверхности отклика имеется одна вершина (впадина), модель называется <u>унимодальной</u>.*

Характер изменения функции при этом может быть различным (Рис. 1.5).



а б в

Рис. 1.5

Модель может иметь разрывы первого рода (см. рис. 1.5. а). Непрерывная унимодальная модель может иметь точки разрыва производной – разрывы второго рода (см. рис. 1.5. б). На рис. 1.5 в показана непрерывно-дифференцируемая унимодальная модель.

Для всех трех случаев, представленных на рис. 1.5, выполняется общее требование унимодальности:

> *Если $W(x^*) = extr\ W$, то из условия $x_1 < x_2 < x^*$ ($x_1 > x_2 > x^*$) следует $W(x_1) < W(x_2) < W(x^*)$, если extr – максимум, или $W(x_1) > W(x_2) > W(x^*)$, если extr – минимум, то есть, по мере удаления от экстремальной точки значение функции $W(x)$ <u>непрерывно</u> падает (растет).*

Наряду с унимодальными бывают полимодальные модели (Рис. 1.6).

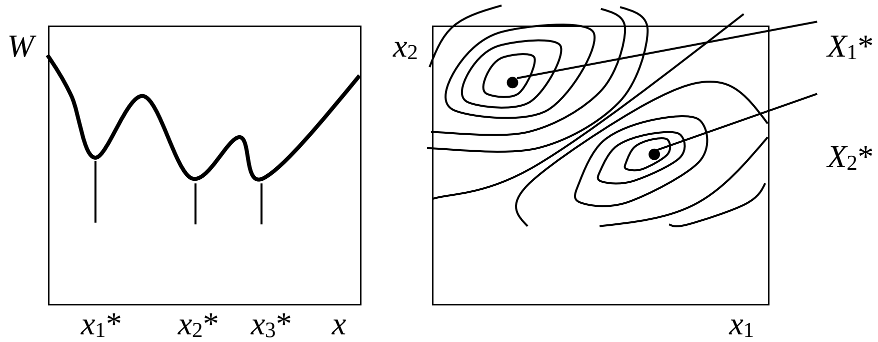


Рис. 1.6

Другим важным свойством поверхности отклика является ее <u>контрастность</u>, показывающая чувствительность результирующей функции к изменению факторов. Контраст-

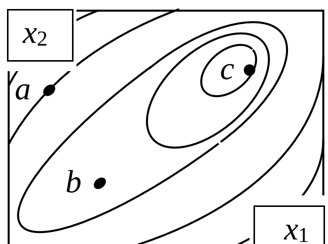


Рис. 1.7

ность характеризуется величинами производных. Продемонстрируем характеристики контрастности на примере двумерной поверхности отклика (Рис. 1.7). Точка *a* расположена на «склоне», характеризующем равную контрастность по всем переменным x_i (i=1,2); точка b расположена в «овраге», в котором различная контрастность по различным переменным (имеем плохую обусловленность функции); точка *c* расположена на «плато», на котором низкая контрастность по всем переменным x_i говорит о близости экстремума.

Глава 2. ТЕОРЕТИЧЕСКИЕ МАТЕМАТИЧЕСКИЕ МОДЕЛИ АНАЛИТИЧЕСКОГО ТИПА

> *Простейшие аналитические модели могут быть <u>заданы явно в виде функции одной или нескольких переменных</u>.*

Обычно в виде функций задаются общие законы природы или общие закономерности, полученные в результате интегрирования дифференциальных уравнений. Примером такой модели может служить знаменитая формула К.Э. Циолковского:

$$\Delta v_{\text{ла}} = v \ln \frac{M_0}{M_{\text{к}}},$$

определяющая приращение скорости ракеты $\Delta v_{\text{ла}}$ при импульсном сжигании топлива через скорость истечения рабочего тела v и отношение начальной M_0 и конечной $M_{\text{к}}$ масс ракеты.

Модель, заданная в явном виде, дает исчерпывающее описание исследуемого объекта. Она позволяет построить зависимость его характеристик от управляющих факторов, взять производные и найти экстремумы модели, определить характеристики модели в окрестности экстремумов и т.д.

Очень удобна графическая интерпретация таких моделей. Однако модели в виде формул могут быть разработаны только для очень простых объектов.

2.1. Построение математической модели сверления лазером

Примером аналитической теоретической модели может служить модель, описывающая глубину отверстия при лазерном сверлении.

Резание и сверление металлов весьма важно для многих областей техники. Значительный интерес представляет создание новых устройств, предназначенных для специальных материалов, а также для тех случаев, когда желательно обеспечить некоторую степень автоматизации указанных процессов. В последнее время для этого были предприняты попытки использования мощных лазеров.

Основная идея состоит в том, чтобы сфокусировать значительную мощность на малой площади поверхности материала, создавая таким образом интенсивный нагрев и испарение с последующим образованием отверстия. При сверлении необходимо постараться обеспечить такие условия процесса, чтобы проделанное отверстие прямо проходило сквозь материал, и избежать, таким образом, затекания расплавленного металла обратно в отверстие и застывания его там.

Построим математическую модель, главная применимость которой – глубокое сверление. При помощи модели попытаемся ответить на вопрос, как быстро можно проделать отверстие, используя пучок излучения высокой мощности, и на какую глубину.

Рассмотрим высокоэнергетический пучок лазерного излучения, сфокусированный на малом участке поверхности металла (Рис. 2.1). Определенная доля энергии поглощается, а остальная часть отражается. Поглощение энергии происходит внутри слоя, толщина которого много меньше миллиметра, вызывает поверхностный нагрев материала и рост температуры поверхности. Температура растет не безгранично. Существует два процесса, ограничивающие рост температуры:

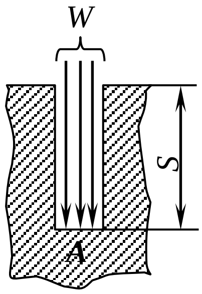


Рис. 2.1

– перенос тепла в глубь материала от нагретых к холодным участкам, обусловленный теплопроводностью;

– испарение. Когда температура материала достигает точки кипения, скрытое тепло поглощается без дальнейшего увеличения температуры в процессе испарения материала.

При удалении пара от поверхности материала в металле образуется выемка.

Задача количественного описания этого процесса и вызывает необходимость математического моделирования.

Будем рассматривать модель, описывающую процесс разрушения материала, при котором вся энергия лазерного излучения используется только для испарения материала.

Этот предельный режим испарения может возникать двумя путями:

– когда энергия поступает на поверхность слишком быстро, так что тепло не успевает распространиться в глубь металла;

– плотность мощности пучка постоянна, а распределение температуры впереди границы области испарения приближается к стационарному.

Предположим, что мощность W распределена по некоторой площади A поверхности; излучение приложено по нормали к поверхности (см. рис. 2.1). За интервал времени Δt поступает энергия $W \cdot \Delta t$. Пусть глубина возникающей выемки равна ΔS, тогда объем испарившегося материала равен $A \cdot \Delta S$. Используя закон сохранения энергии, получим

$$h \cdot \rho \cdot A \cdot \Delta S = W \cdot \Delta t,$$

где h – количество тепла, требуемое для испарения единицы массы материала; ρ – плотность материала.

Преобразуем это выражение и положим $\Delta t \to 0$, получим скорость роста глубины выемки:

$$\frac{dS}{dt} = \frac{W}{A} \cdot \frac{1}{h\rho}.$$

Это уравнение показывает, что для любого материала предельная скорость пропорциональна плотности энерговыделения W/A. Интегрируя это уравнение и полагая $S = 0$ при $t = 0$, найдем глубину выемки в произвольный момент времени t:

$$S(t) = \frac{1}{h\rho A} \int_0^t W dt \qquad (2.1)$$

или

$$S(t) = \frac{E(t)}{h\rho A},$$

где $E(t)$ – полная энергия, выделенная источником за промежуток времени $(0, t)$.

Таким образом, в предельном режиме испарения глубина выемки зависит только от полной энергии, поступившей на поверхность. Формула (2.1) представляет собой теоретическую аналитически-разрешимую динамическую детерминированную модель.

На практике всегда существует перенос некоторого количества тепла в материал за счет теплопроводности. Общая задача движения границы раздела фаз с учетом теплопроводности известна как *задача Стефана.* Ее решение представляет определенные математические трудности.

Контрольные вопросы

11. В виде чего может быть представлена математическая модель геометрически?
12. Что такое область определения математической модели?
13. Какая модель называется унимодальной?
14. Как задаются математические модели аналитического типа?
15. Приведите пример математической модели аналитического типа.
16. Какие задачи позволяет решить модель, заданная в явном виде?
17. Какой предельный режим рассматривается при построении математической модели сверления лазером?
18. Какой закон используется при построении математической модели сверления лазером?

19. Назовите процессы, препятствующие росту температуры при лазерном сверлении.
20. На какие вопросы можно ответить, используя математическую модель сверления лазером?
21. К какому типу принадлежит модель зависимости глубины выемки от длительности импульса?

2.2. Линейные математические модели

Наиболее простыми являются так называемые линейные детерминированные модели. Они задаются в виде линейной формы управляющих переменных (x):

$$W = a_0 + a_1x_1 + \ldots + a_kx_k$$

при линейных ограничениях вида

$$b_{1j}\,x_1 + b_{2j}\,x_2 + \ldots + b_{kj}\,x_k \geq b_j\,,\ \ j = 1,\ldots,\,q_1;$$

$$c_{1j}\,x_1 + c_{2j}\,x_2 + \ldots + c_{kj}\,x_k = c_j\,,\ \ j = 1,\ldots,\,q_2;$$

$$d_{1j}\,x_1 + d_{2j}\,x_2 + \ldots + d_{kj}\,x_k \leq d_j\,,\ \ j = 1,\ldots,\,q_3.$$

Общее число ограничений $m = q_1 + q_2 + q_3$ может превосходить число переменных ($m > k$). Кроме того, обычно вводится условие положительности переменных ($x_i \geq 0$).

Поверхность отклика для линейной модели представляет собой гиперплоскость. Например, рассмотрим линейную модель двух переменных следующего вида:

$$W = -2x_1 - 3x_2 \qquad (2.2)$$

при следующих ограничениях

$$\begin{aligned} &2x_1 + 3x_2 \leq 18; \\ &x_1 - 4x_2 \leq 4; \\ &-2x_1 + x_2 \leq 2; \\ &x_1 \geq 0; \quad x_2 \geq 0. \end{aligned} \qquad (2.3)$$

Область допустимых значений (область определения) *OABCD* для модели (2.2) образована ограничениями (2.3) (Рис. 2.2). Поверхность отклика представляет собой плоский

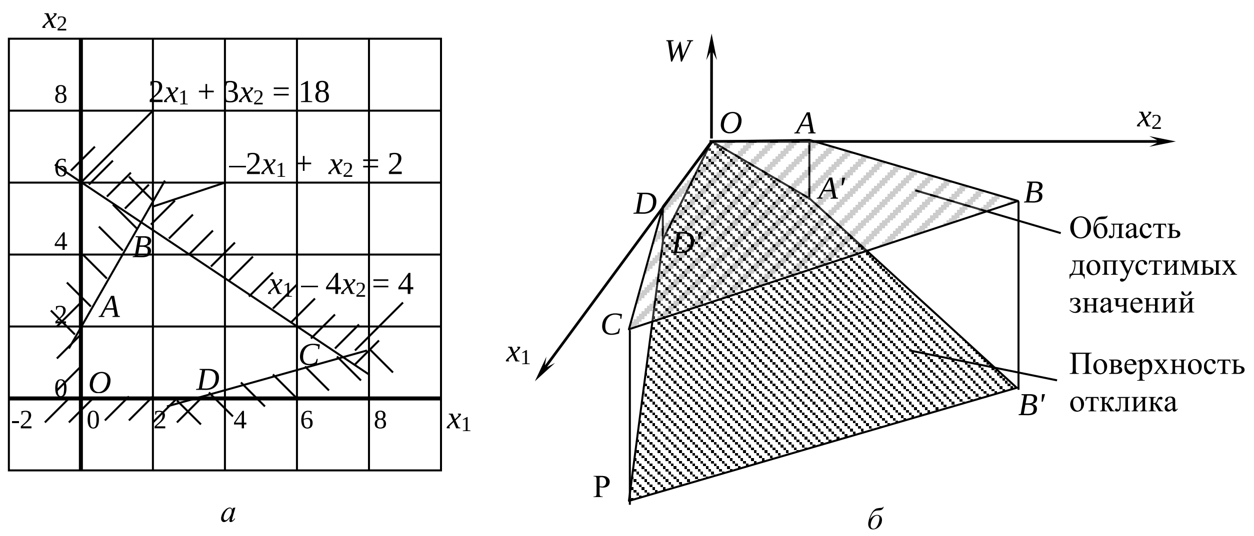


Рис. 2.2

многоугольник *OA'B'C'D'* (рис. 2.2, *б*).

При определенном соотношении ограничений множество допустимых решений может отсутствовать (пусто). Пример такого множества показан на рис. 2.3. Прямые *AC* и *BC* ограничивают область допустимых значений сверху. Третье ограничение отсекает область допустимых значений снизу от прямой *AB.* Таким образом, общей области, удовлетворяющей всем трем ограничениям, нет.

Линейные модели достаточно просты и поэтому, с одной стороны, предполагают существенное упрощение задачи, а с другой – допускают разработку простых и эффективных методов решения.

При исследовании ДЛА линейные модели используются редко и почти исключительно при приближенном описании задач.

Линейные модели могут использоваться при поэтапной аппроксимации нелинейных моделей (линеаризация задачи). Особенно эффективен этот прием при изучении небольших областей исследуемого пространства. Представление отдельных участков нелинейной поверхности отклика линейной моделью лежит в основе большой группы методов оптимизации, так называемых методов с линейной тактикой.

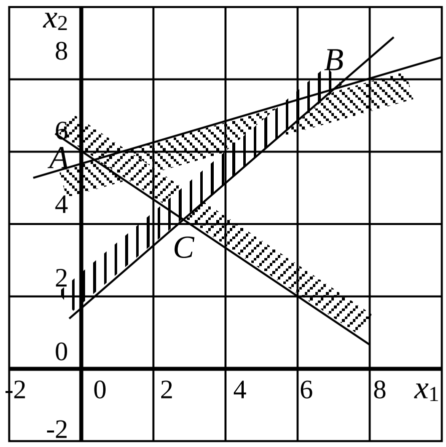


Рис. 2.3

Исследование линейных моделей не представляет труда. В частности влияние каждой из переменных на характеристики модели вида

$$W = a_0 + a_1x_1 + a_2x_2 + \ldots + a_kx_k$$

задается ее коэффициентами:

$$\frac{\partial W}{\partial x_i} = a_i,\ i = 1,\ldots, k.$$

Для нахождения оптимума линейной модели $W_{\text{опт}}$ разработан эффективный симплекс-метод.

К линейным иногда сводятся простейшие модели стоимости, рассматриваемые как совокупность производимых затрат.

Примером такой модели является классическая <u>модель стоимости перевозок (транспортная задача)</u> (Рис. 2.4).

Имеется k пунктов производства ($i = 1,\ldots, k$) и m пунктов потребления ($j = 1,\ldots, m$) некоторого продукта. Количество продукта, произведенного в каждом из k пунктов производства, равно a_i; количество продукта, необходимого в каждом из m пунктов потребления, равно b_j.

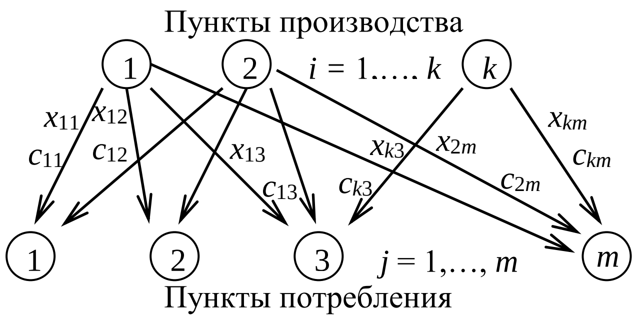


Рис. 2.4

Предполагается равенство общего производства и потребления:

$$\sum_{i=1}^{k} a_i = \sum_{j=1}^{m} b_j\ .$$

Количество продукта, перевозимого из i-го пункта производства в j-й пункт потребления, равно x_{ij}; стоимость перевозки единицы этого продукта – c_{ij}.

Суммарная стоимость перевозок C_Σ задается <u>линейной моделью</u>:

$$C_\Sigma = \sum_{i=1}^{k}\sum_{j=1}^{m} c_{ij}x_{ij}$$

при следующих ограничениях

$$\sum_{j=1}^{m} x_{ij} = a_i;\quad \sum_{i=1}^{k} x_{ij} = b_j;\quad x_{ij} \ge 0.$$

К линейным также относятся модели в виде линейных дифференциальных уравнений (обыкновенных или в частных производных).

Линейное обыкновенное дифференциальное уравнение n-го порядка имеет вид

$$a_n \frac{d^n x(t)}{dt^n} + a_{n-1} \frac{d^{n-1} x(t)}{dt^{n-1}} + \ldots + a_1 \frac{dx(t)}{dt} + a_0 x(t) = f(t)\ . \tag{2.10}$$

Начальные условия записываются как

$$x(0)=C_0,\quad x'(0)=C_1,\quad x''(0)=C_2,...,\quad x^{(n-1)}(0)=C_{n-1}.$$

Линейное дифференциальное уравнение в частных производных имеет вид

$$a_0\frac{\partial\Phi(t)}{\partial t}+a_1\frac{\partial\Phi(t)}{\partial x_1}+a_2\frac{\partial\Phi(t)}{\partial x_2}+...+a_k\frac{\partial\Phi(t)}{\partial x_k}=f(x_1,x_2,...,x_k,t).$$

Модель, заданная в виде дифференциального уравнения в частных производных, включает начальные и граничные условия (условия на границе области определения функции $\Phi(t)$).

2.3. Исследование простейшей математической модели работы газотурбинного двигателя

Газотурбинный двигатель (ГТД) является основной силовой установкой современных самолетов.

Схема ГТД имеет вид, показанный на рис. 2.5.

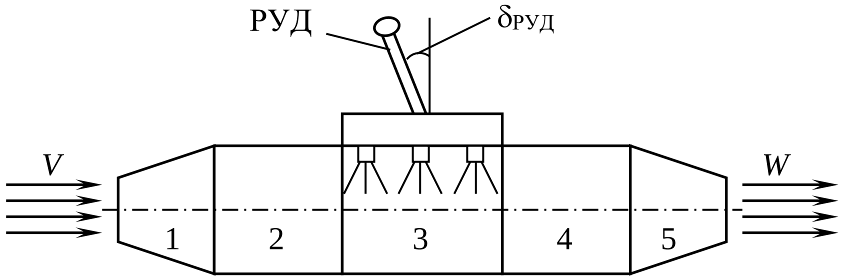


Рис. 2.5

Здесь *1* – входной диффузор; *2* – компрессор; *3* – камера сгорания; *4* – турбина; *5* – выходное сопло.

Цикл работы ГТД включает следующие этапы:

1) Набегающий со скоростью V поток воздуха через диффузор поступает в компрессор.

2) Компрессор, вращаясь на одном валу с турбиной, сжимает воздух, который поступает в камеру сгорания.

3) В камеру сгорания постоянно впрыскивается топливо (керосин), которое смешивается со сжатым воздухом.

4) Газ, образующийся от сгорания, поступает на турбину, которая разгоняет его до скорости W.

5) С этой скоростью газ через сопло выбрасывается в атмосферу.

За счет того, что $W > V$, образуется сила тяги P, которая позволяет самолету осуществлять полет в атмосфере.

Изменение силы тяги осуществляется путем изменения скорости впрыска топлива в камеру сгорания с помощью перемещения ручки управления двигателем (РУД). Перемещение РУД на определенный угол $\delta_{РУД}$ осуществляется либо вручную летчиком, либо с помощью исполнительного устройства по сигналам от САУ полетом. Увеличение значения $\delta_{РУД}$ вызывает возрастание силы P, а уменьшение – убывание этой силы.

ГТД является сложной технической системой, в которой протекает значительное число физических и химических процессов. Двигатель оснащен всевозможными устройствами автоматики, системами поворота и охлаждения турбинных лопаток и т.д. Естественно, математическое описание функционирования ГТД также будет достаточно громоздким, включающим в себя системы дифференциальных уравнений в частных производных, обыкновенных дифференциальных уравнений, трансцендентных функций, алгоритмы цифровой системы управления двигателем. Такие модели используются в процессе проектирования ГТД.

Для решения задач управления полетом используется более простая модель ГТД, представляющая собой зависимость силы тяги P от угла $\delta_{РУД}$ отклонения РУД. Процесс изменения силы тяги описывается обыкновенным дифференциальным уравнением вида:

$$\tau \frac{dP}{dt} + P = k\delta_{РУД}, \qquad (2.11)$$

где $\tau > 0$ – постоянная времени двигателя, зависящая кроме конструктивных характеристик также от температуры окружающего воздуха, его влажности и других внешних факторов; k [кг/град] – коэффициент пропорциональности.

Начальное условие для уравнения (2.11) записывается как

$$P(0) = P_0. \qquad (2.12)$$

Таким образом, уравнение (2.11) совместно с начальным условием (2.12) представляет собой *простейшую математическую модель работы ГТД, записанную в виде обыкновенного дифференциального уравнения* 1-*го порядка.*

Для определения коэффициента пропорциональности k используются градуировочные графики зависимости тяги от угла поворота РУД, построенные на основе экспериментальных данных. Тангенс угла наклона графика равен искомому коэффициенту.

Интегрирование уравнения (2.11) с начальным условием (2.12) позволяет выяснить,

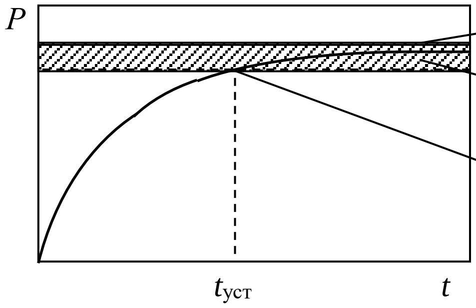


Рис. 2.6

как изменяется сила тяги во времени (рис. 2.6).

При отклонении РУД тяга P нарастает и затем стабилизируется на определенном предельном значении, т.е. ГТД является инерционным объектом.

Предельное значение силы тяги получаем из (2.11), когда скорость ее изменения равна нулю:

$$\frac{dP}{dt} = 0 \quad \Rightarrow \quad P_{\text{пред}} = k\delta_{\text{РУД}}\,. \qquad (2.13)$$

Длительность нарастания зависит от значения постоянной времени двигателя τ. Процесс считается установившимся при $t = t_{уст}$, когда тяга входит в так называемый пятипроцентный коридор от предельного значения силы тяги (рис. 2.6). Чем больше τ, тем инерционнее двигатель и, следовательно, больше $t_{уст}$.

На рис. 2.7 показано поведение силы тяги в зависимости от угла отклонения РУД при $\tau = 0,5$.

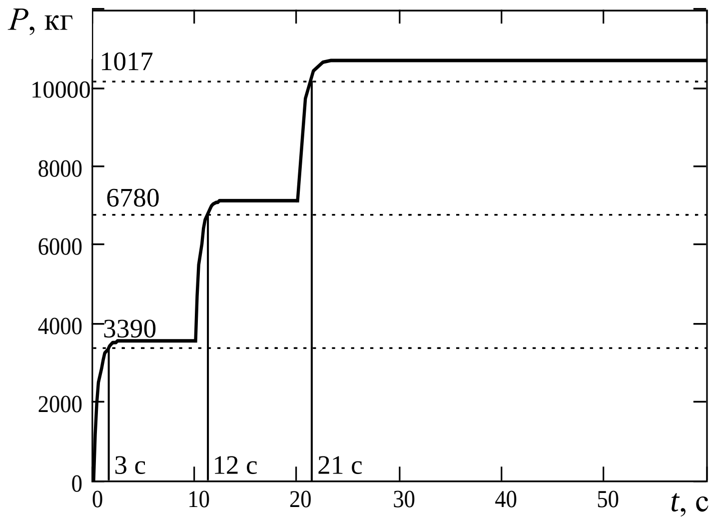


Рис. 2.7

Сила тяги при взлете, когда РУД отклонена на 10°, выходит на установившийся режим на третьей секунде и достигает величины 3390 кг. Через десять секунд после взлета, когда РУД отклонена на 20°, сила тяги устанавливается на величине 6780 кг, и еще через десять секунд, когда РУД отклонена на 30°, сила тяги устанавливается на величине 10170 кг. Предельное значение силы тяги равно

14270 кг.

2.4. Нелинейные детерминированные модели

Нелинейные детерминированные модели обладают бо́льшей точностью и гибкостью. Они могут быть заданы в виде нелинейной функции одной или нескольких переменных или в виде дифференциальных уравнений (обыкновенных или в частных производных). Наиболее распространенными среди нелинейных моделей при описании ДУ и ДЛА являются:

- полиномиальные функции;
- позиномные функции;
- тригонометрические функции;
- экспоненциальные функции;
- обыкновенные дифференциальные уравнения;
- дифференциальные уравнения в частных производных др.

Нелинейные модели могут быть записаны в виде функционала, зависящего от управляющих переменных x и некоторых функций $f(x)$ всех или части этих переменных: $W = W(x,f(\mathrm{x}))$. При этом функции $f(x)$ могут представлять собой функционалы, зависящие от промежуточных функций $f^*(x)$ и т.д. На класс функций $f(x)$, $f^*(x)$ не накладывается никаких ограничений, однако предполагается возможность однозначного перехода от вектора управляющих параметров x к общей характеристике модели W.

Область определения модели может быть ограничена с помощью равенств или неравенств:

$$
\begin{aligned}
&x_i = c_i, && i = 1,\ldots, m;\\
&f(x) = c_j, && j = 1,\ldots, l;\\
&x_{i\,\min} \le x_i \le x_{i\,\max}, && i = 1,\ldots, k;\\
&f_j(x) \le c_j, && j = 1,\ldots, n.
\end{aligned}
$$

По существу под определение нелинейной модели подпадает любое математическое описание ДУ и ДЛА, не укладывающееся в рамки более простых моделей.

2.4.1. Полиномиальные модели

Полиномиальные модели основаны на идее приближенного представления модели конечным числом членов ряда Тейлора:

$$W(x) = W(x_0) + \sum_{i=1}^{k} \frac{\partial W(x_0)}{\partial x_i}(x_i - x_{i0}) + \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k} \frac{\partial^2 W(x_0)}{\partial x_i \partial x_j}(x_i - x_{i0})(x_j - x_{j0}) + \ldots \, .$$

Наиболее простой из моделей этого класса является <u>квадратичная модель</u>:

$$W(x) = a_0 + \sum_{i=1}^{k} a_i x_i + \sum_{\substack{i=1 \\ j \ge i}}^{k} a_{ij} x_i x_j$$

при ограничениях

$$\sum_{i=1}^{k} b_{ij} x_i \ge b_j,\ j = 1,\dots,q_1; \qquad \sum_{i=1}^{k} c_{ij} x_i = c_j,\ j = 1,\dots,q_2; \qquad \sum_{i=1}^{k} d_{ij} x_i \le d_j,\ j = 1,\dots,q_3.$$

Квадратичные модели широко используются для представления экспериментальных данных при идентификации ДЛА и их элементов.

Квадратичные модели используются для аппроксимации отдельных участков поверхности отклика, когда линейное приближение оказывается недостаточным, например, в окрестности экстремума, и лежит в основе нелинейных методов оптимизации. Если квадратичная модель также оказывается недостаточно точной, то используются полиномиальные модели более высоких порядков.

Исследование полиномиальных моделей частично можно осуществить аналитическими методами. Например, аналитически можно определить степень влияния отдельных переменных на характеристики модели.

2.4.2. Позиномные модели

Позиномные модели основаны на представлении модели в виде суммы произведений степенных функций:

$$W(x) = \sum_{j=1}^{m} c_j x_1^{\alpha_{1j}} x_2^{\alpha_{2j}} \dots x_k^{\alpha_{kj}} = \sum_{j=1}^{m} c_j \prod_{i=1}^{k} x_i^{\alpha_{ij}}\ , \tag{2.14}$$

где x_i – управляющие переменные, α_{ij} – произвольные положительные числа, $c_j \ge 0$ – обеспечивает выпуклость модели.

Величины α_{ij}, c_j рассчитываются на основе статистических данных, отражающих опыт производства соответствующих узлов и систем.

Позиномные модели можно использовать для описания стоимости сложных систем.

К позиномным моделям сводится задача выбора геометрических характеристик ряда технических устройств, в том числе элементов ДЛА, например, электромагнитов, силовых ферм и т.д.

Исследование позиномных моделей сложнее, чем моделей полиномиального типа, и осуществляется в основном численными методами. Однако, при $m = 1$ и $x_1 > 0$, $x_2 > 0,\dots, x_k > 0$ в формуле (2.4) существует способ приведения позинома к линейному виду.

В этом частном случае модель (2.4) будет выглядеть в следующем виде:

$$W(x_1, x_2, ..., x_k) = c \cdot x_1^{\alpha_1} x_2^{\alpha_2} ... x_k^{\alpha_k} .$$

Прологарифмируем обе части этого равенства, получим

$$\ln W = \ln c + \alpha_1 \ln x_1 + \alpha_2 \ln x_2 + ... + \alpha_k \ln x_k . \qquad (2.15)$$

Введем обозначения логарифмов переменных W, x_1, x_2,…,x_k и константы c:

$$Y = \ln W; \quad C = \ln c; \quad X_i = \ln x_i; \quad i = 1,...,k.$$

Выражение (2.5) примет линейный вид

$$Y(X_1, X_2, \ldots, X_k) = C + \alpha_1 X_1 + \alpha_2 X_2 + \ldots + \alpha_k X_k.$$

Для поиска оптимальных решений на основе позиномных моделей разработан специальный аппарат – так называемое геометрическое программирование.

Контрольные вопросы

1. С какими значениями величин оперируют детерминированные модели?
2. Как выглядит линейная детерминированная модель в общем виде?
3. Что представляет собой поверхность отклика для линейной модели?
4. Приведите модель стоимости перевозок.
5. Где используются линейные детерминированные модели?
6. Приведите простейшую математическую модель изменения силы тяги ГТД.
7. К какому типу она относится?
8. Где она может быть использована?
9. Приведите модель установившегося процесса горизонтального полета самолета.
10. Что и как можно определить с ее помощью?
11. Какие виды нелинейных математических моделей Вы знаете?
12. Приведите общий вид квадратичного полинома.
13. Приведите формулу позинома.
14. Как привести позином к линейному виду (при каком условии)?

2.4.3. Математическая модель кратчайшего пути

В качестве примера применения нелинейных статических моделей рассмотрим задачу описания двумерного движения точки по ограниченной области (рис. 2.8). Такая задача может возникнуть при определении координат опорных точек движения инструмента на станке с ЧПУ.

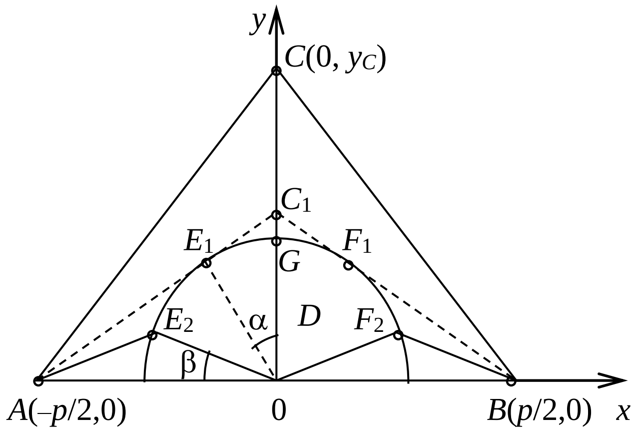


Рис. 2.8

Найдем кратчайший путь от точки A с координатами (x_A, y_A) до точки B с координатами (x_B, y_B) на плоскости, из которой исключена область D, определенная неравенством $x^2 + y^2 \leq R^2$.

Кратчайшим расстоянием между двумя точками на плоскости является соединяющий их отрезок прямой.

Пусть расстояние между точками A и B равно p и центр окружности, ограничивающий область D, лежит посередине между точками A и B. Тогда

$$x_A = -\frac{p}{2};\ y_A = 0;\ x_B = \frac{p}{2};\ y_B = 0.$$

Рассмотрим путь ACB, где точка C имеет координаты $(0, y_C)$, а y_C – достаточно велико, чтобы отрезки AC и CB не пересекались с областью D. Тогда по теореме Пифагора

$$ACB = 2\sqrt{\left(\frac{p}{2}\right)^2 + y_C^2}\ .$$

Отсюда видно, что при убывании y_C путь сокращается. Будем уменьшать y_C до тех пор, пока AC не коснется окружности в точке E_1 ($C \to C_1$). Этот путь является наилучшим среди путей, составленных из двух отрезков прямых линий.

Обозначим через α угол $\angle E_1 0 C_1$, тогда $E_1C_1 + C_1F_1 = 2R\text{tg}\alpha$;

Длина дуги E_1F_1 определяется по формуле $\text{arc}E_1F_1 = 2R\alpha$.

Но $\text{tg}\alpha > \alpha$ для всех $\alpha \in \left(0; \frac{\pi}{2}\right)$.

Следовательно, путь, состоящий из отрезка AE_1, дуги E_1F_1 и отрезка F_1B, является более коротким, чем AC_1B.

На этой стадии решения задачи мы выяснили, что кратчайший путь состоит их двух отрезков прямых линий и дуги окружности.

Для окончательного решения задачи рассмотрим путь AE_2 , дуга E_2F_2 , F_2B, где $\angle A0E_2 = \angle B0F_2 = \beta$.

Длину этого пути обозначим через S. Получим математическую модель пути:

$$S = 2(AE_2 + \text{arc}E_2G) = 2\left[\sqrt{\left(\frac{p}{2} - R\cos\beta\right)^2 + R^2\sin^2\beta} + R\left(\frac{\pi}{2} - \beta\right)\right]. \quad (2.16)$$

$$0 \le \beta \le \beta_{\text{огр}} = \frac{\pi}{2} - \alpha\,, \quad (2.16')$$

где α – угол между прямой E_10 и осью $0Y$.

Ограничение (2.16') вводится потому, что при $\beta > \frac{\pi}{2} - \alpha$ прямая AE_1 пересечет область D, а этого не должно быть.

Задача заключается в определении угла β_0, при котором путь S будет минимальным. Необходимым условием минимума функции $S(\beta)$ является равенство нулю производной:

$$\frac{dS}{d\beta} = 0\,. \quad (2.17)$$

Рассмотрим частный случай:

$$P = 4;\ R = 1.$$

Тогда $\alpha = \frac{\pi}{2} - \arccos\left(\frac{R}{p/2}\right) = \arcsin\left(\frac{R}{p/2}\right) = \arcsin\left(\frac{1}{2}\right) = \frac{\pi}{6}$.

Подставив значения p и R в математическую модель (2.6), получим

$$S = 2\left[\sqrt{(2-\cos\beta)^2 + \sin^2\beta} + \frac{\pi}{2} - \beta\right].$$

Произведя некоторые преобразования, получим

$$S = 2\left(\sqrt{5 - 4\cos\beta} + \frac{\pi}{2} - \beta\right).$$

Возьмем производную по β от этого выражения и приравняем ее к нулю.

$$\frac{4\sin\beta}{2\sqrt{5-4\cos\beta}} - 1 = 0\,.$$

Получили уравнение, решив которое относительно β, найдем значение угла β_0, при котором S минимально. Опустив промежуточные преобразования, получим $\cos\beta = 1/2$. То есть $\beta = \pi/3$.

Чтобы убедиться, что найденное значение является точкой минимума, необходимо исследовать вторую производную от (2.16). Если она больше нуля при $\beta = \beta_0$, то $S(\beta)$ действительно минимальна в этой точке.

Вторая производная от $S(\beta)$ имеет вид

$$\frac{d^2S}{d\beta^2} = -4\frac{2\cos^2(\beta) - 5\cos(\beta) + 2}{\left[5 - 4\cos(\beta)\right]^{3/2}}.$$

Подставив в нее найденное значение $\beta_0 = \pi/3$, получим

$$\left.\frac{d^2S}{d\beta^2}\right|_{\beta_0=\pi/3} = 0.$$

Равенство нулю второй производной требует дополнительного исследования критической точки. Необходимо найти первую, не обращающуюся в нуль, производную. Если она нечетного порядка, функция не имеет в исследуемой точке ни максимума, ни минимума. Если она четного порядка и больше нуля, исследуемая точка является минимумом. Проверим третью производную от $S(\beta)$ по β:

$$\left.\frac{d^3S}{d\beta^3}\right|_{\beta=\frac{\pi}{3}} = -\left.\frac{\left[4\left(4\cos^2(\beta) - 10\cos(\beta) + 13\right)\right]\sin(\beta)}{\sqrt{\left(5 - 4\cos(\beta)\right)^5}}\right|_{\beta=\frac{\pi}{3}} = -2.$$

Отсюда имеем, что при $\beta = \pi/3$ функция $S(\beta)$ не имеет ни максимума, ни минимума. Действительно, из графика функции $S(\beta)$ (рис. 2.9) видно, что на отрезке (2.16') функция (2.16) монотонно убывает. В точке $\beta_0 = \pi/3$, совпадающей с $\beta_{огр}$, кривая имеет точку перегиба. Наименьшее в области определения значение находится на границе этой области. Следовательно, путь AE_1GF_1B действительно кратчайший и его длина равна $S(\pi/3) = 4{,}511$.

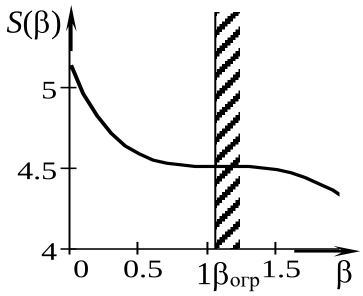


Рис. 2.9

Покажем, что математическая модель (2.6) для любых p и R монотонно убывает на отрезке $0 \le \beta \le \beta_{огр} = \pi/2 - \alpha$ и, следовательно, имеет наименьшее значение при $\beta = \pi/2 - \alpha$. Для этого необходимо показать, что вторая производная от $S(\beta)$ на интересующем нас отрезке не превышает нуля.

Вторая производная от функции (2.6) имеет вид

$$\frac{d^2S}{d\beta^2} = 2R\frac{p\sin(\beta) - \sqrt{p^2 - 4pR\cos(\beta) + 4R^2}}{\sqrt{p^2 - 4pR\cos(\beta) + 4R^2}}.$$

Покажем, что она не превышает нуля:

$$2R\frac{p\sin(\beta) - \sqrt{p^2 - 4pR\cos(\beta) + 4R^2}}{\sqrt{p^2 - 4pR\cos(\beta) + 4R^2}} \le 0.$$

Разделив обе части неравенства на $2R$ и умножив на корень квадратный (это можно сделать, не нарушив неравенства, так как $R > 0$, а корень квадратный представляет собой длину отрезка, т. е. тоже больше нуля), получим

$$p\sin(\beta) \le \sqrt{p^2 - 4pR\cos(\beta) + 4R^2}\ .$$

Возведя обе части в квадрат (на рассматриваемом отрезке $\sin(\beta) > 0$) и произведя некоторые преобразования, получим

$$p^2\cos^2(\beta) \ge 4pR\cos(\beta) - 4R^2.$$

В левой части неравенства $\cos^2(\beta)$ можно заменить его минимальным значением, т.е. нулем, а в правой части – максимальным значением, т.е. единицей. Тогда получим

$$0 \ge 4pR - 4R^2 \quad \text{или} \quad p \ge R.$$

Но p действительно больше R (см. рис. 2.5).

Таким образом, аналитическую модель пути (формула (2.6)) мы использовали для доказательства того, что при $\beta = \pi/2 - \alpha$ путь является кратчайшим. Зная это, можно определить координаты опорных точек движения инструмента на станке с ЧПУ при любых значениях величин p и R:

$$A(-p/2,\ 0);\ E_1(-R\sin(\alpha),\ R\cos(\alpha));\ F_1(R\sin(\alpha),\ R\cos(\alpha));\ B(p/2,\ 0),$$

где $\alpha = \arcsin\left(\dfrac{R}{p/2}\right)$.

Контрольные вопросы

1. К какому типу можно отнести модель кратчайшего расстояния между двумя точками?
2. Является ли найденное значение угла β точкой минимума пути?
3. Является ли путь S при найденном значении угла β кратчайшим?

2.5. Математическая модель в виде обыкновенных дифференциальных уравнений

Математическая модель в виде одного или нескольких обыкновенных дифференциальных уравнений (ОДУ) широко используются при изучении переходных процессов в системах автоматического регулирования (САР), при описании баллистики летательных аппаратов, а также при описании процессов движения (потоки, частицы, механические элементы).

В простейшем случае модель может иметь вид линейного дифференциального уравнения *n*-го порядка:

$$a_n \frac{d^n x(t)}{dt^n} + a_{n-1} \frac{d^{n-1} x(t)}{dt^{n-1}} + \ldots + a_1 \frac{dx(t)}{dt} + a_0 x(t) = f(t)$$

или системы дифференциальных уравнений 1-го порядка

$$\frac{dx_1}{dt} = f_1(t, x_1, \ldots, x_n);$$

$$\frac{dx_2}{dt} = f_2(t, x_1, \ldots, x_n);$$

....

$$\frac{dx_n}{dt} = f_n(t, x_1, \ldots, x_n).$$

Часто встречаются смешанные задачи, а также нелинейные ОДУ.

Модель, заданная в виде дифференциальных уравнений, должна включать в себя необходимый набор начальных условий:

$$x(0) = C_0, \quad x'(0) = C_1, \quad x''(0) = C_2, \ldots, \quad x^{(n-1)}(0) = C_{n-1}$$

или $\qquad x_1(0) = C_1, \quad x_2(0) = C_2, \ldots, \quad x_n(0) = C_n .$

Исследование моделей, заданных в виде обыкновенных дифференциальных уравнений, осуществляется аналитическими и численными методами. Наиболее полными являются аналитические решения, обеспечивающие всесторонний анализ полученных результатов. Но такие решения получены лишь для ограниченного числа дифференциальных уравнений. Численные методы решения позволяют найти лишь конкретные значения изучаемой функции при заданной комбинации исходных данных. Для анализа модели можно использовать некоторую совокупность решений. Однако, очевидно, что результаты анализа в этом случае могут зависеть от выбора этой совокупности.

В качестве простейшего примера математической модели механической системы мо-

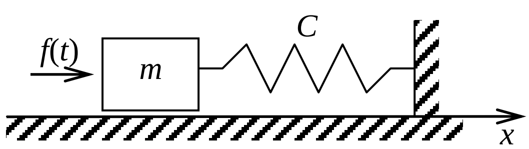


Рис. 2.10

жет быть рассмотрена модель движения груза массой m, закрепленного на вертикальной стенке с помощью пружины жесткостью C и совершающего колебательное движение вдоль оси x в среде с вязкостью ν (Рис. 2.10).

Возмущающая сила, вызывающая колебания, зависит от времени $f(t)$. Наряду с возмущающей силой $f(t)$ на груз действует сила инерции $m\frac{d^2x(t)}{dt^2}$, сила вязкого трения $\nu\frac{dx(t)}{dt}$, усилие пружины $\frac{1}{C}x(t)$. Все эти силы тормозят движение груза.

Согласно принципу Даламбера сумма всех сил, действующих на груз должна равняться нулю:

$$m\frac{d^2x(t)}{dt^2}+\nu\frac{dx(t)}{dt}+\frac{1}{C}x(t)-f(t)=0. \tag{2.18}$$

Начальные условия характеризуют начальное положение и начальную скорость груза:

$$x(0) = x_0;\quad x'(0)=0. \tag{2.19}$$

Уравнение (2.18) совместно с начальными условиями (2.19) представляет собой математическую модель рассматриваемой механической системы.

2.6. Модели, заданные в виде уравнений в частных производных

Ряд задач, связанных с использованием физических полей, приводит к моделям в виде дифференциальных уравнений в частных производных.

> *Особенностью таких задач является то, что изучаемые параметры изменяются не только во времени, но и зависят от координат x, y, z рассматриваемого пространства. Такие модели называются нестационарными. Модели, в которых параметры не зависят от времени, называются стационарными.*

К таким моделям сводятся описания полей температур в элементах конструкции двигателя и полей скоростей при течении жидкости (газа). Уравнениями в частных производных описываются колебания элементов конструкции и поля напряжений, возникающих при работе этих элементов.

Линейное дифференциальное уравнение в частных производных имеет вид

$$a_0\frac{\partial\Phi(t)}{\partial t}+a_1\frac{\partial\Phi(t)}{\partial x_1}+a_2\frac{\partial\Phi(t)}{\partial x_2}+\ldots+a_k\frac{\partial\Phi(t)}{\partial x_k}=f(x_1,x_2,\ldots,x_k,t).$$

Математическая модель, описанная дифференциальными уравнениями в частных производных, должна включать в себя необходимые для решения задачи краевые условия:

1. *Должна быть задана область D, ограниченная поверхностью (на плоскости – кривой) Γ, в которой определяется решение.*
2. *Должны быть заданы условия на границе Γ этой области.*

В случае нестационарного поля эти граничные условия, так же как и сама область могут меняться во времени.

Граничные условия могут быть 1-го, 2-го и 3-го рода:

а) Граничные условия 1-го рода предусматривают задание на границе величины искомой функции:

$$\Phi|_\Gamma = f_1(\Gamma) \text{ – для стационарного поля;}$$

$$\Phi(t)|_\Gamma = f_1(\Gamma, t) \text{ – для нестационарного поля.}$$

б) Граничные условия 2-го рода – предусматривают задание производной искомой функции:

$$\left.\frac{\partial\Phi}{\partial x_i}\right|_\Gamma = f_2(\Gamma) \text{ – для стационарного поля;}$$

$$\left.\frac{\partial\Phi}{\partial x_i}(t)\right|_\Gamma = f_2(\Gamma, t) \text{ – для нестационарного поля.}$$

в) Граничные условия 3-го рода – предусматривают комбинации функции и ее производной:

$$\left.\left(a\Phi + b\frac{\partial\Phi}{\partial x_i}\right)\right|_\Gamma = f_3(\Gamma) \text{ – для стационарного поля;}$$

$$\left.\left(a(t)\Phi(t) + b(t)\frac{\partial\Phi}{\partial x_i}(t)\right)\right|_\Gamma = f_3(\Gamma, t) \text{ – для нестационарного поля.}$$

3. Для нестационарных полей должны быть заданы одно или два начальных условия, характеризующих состояние поля в начальный момент времени:

$$\Phi(x_i)\big|_{t=0} = f_4(x_i);$$

$$\left.\frac{\partial\Phi}{\partial t}(x_i)\right|_{t=0} = f_5(x_i), \quad (i = 1, 2, 3).$$

Здесь x_i – координаты пространства.

Совокупность уравнений и краевых (и начальных) условий полностью определяет модель и позволяет провести ее исследование.

Решение часто задается в виде семейств изолиний Φ = const (Рис. 2.11).

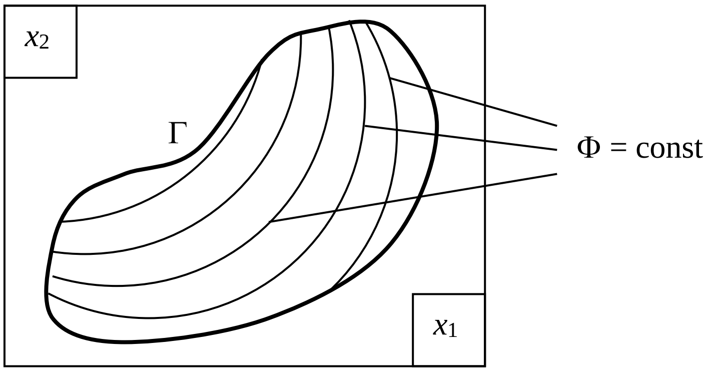


Рис. 2.11

В качестве примера рассмотрим хорошо изолированный металлический пруток, нагреваемый с одной стороны. С другой стороны помещен измеритель температуры (Рис. 2.12). Величина подогрева $x(t)$ в момент времени t является входным сигналом, а измеряемая на другом конце температура $y(t)$ – выходным сигналом.

Обозначим через ξ расстояние от измерителя до точки прутка. Температура в этой точке z будет описываться функцией вида

$$z = z(t, \xi).$$

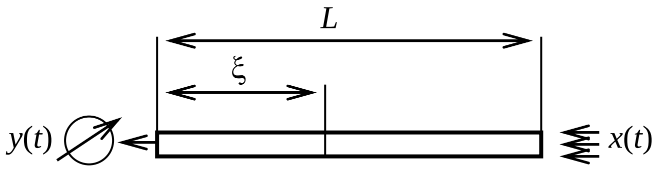


Рис. 2.12

Уравнение теплопроводности для одномерного случая для определения функции z будет иметь вид:

$$\frac{\partial z(t,\xi)}{\partial t} = K\frac{\partial^2 z(t,\xi)}{\partial \xi^2},$$

где K – коэффициент теплопроводности.

Начальным условием в данном случае является начальное распределение температуры (при $t = 0$) по прутку: $z(0, \xi) = \varphi(\xi)$.

Граничные условия определяются тремя условиями:

а) Нагрев прутка на правом конце

$$\left.\frac{\partial z(t,\xi)}{\partial \xi}\right|_{\xi=L} = Kx(t).$$

б) На левом конце подвод тепла отсутствует

$$\left.\frac{\partial z(t,\xi)}{\partial \xi}\right|_{\xi=0} = 0.$$

в) Показания на измерителе температур ($\xi = 0$) в момент времени t определяется следующим выражением

$$z(t,0) = y(t).$$

Таким образом, для вычисления температуры на расстоянии L от измерителя по формуле для $y(t)$ необходимо проинтегрировать дифференциальное уравнение с учетом начальных и граничных условий, т.е. получить функцию $z(t,\xi)$. Затем следует проградуировать измеритель температуры, т.е. определить соответствие между $x(t)$ и $y(t)$, задавая различные значения $x(t)$ и вычисляя $z(t, L)$.

Контрольные вопросы

1. Где используются математические модели в виде обыкновенных дифференциальных уравнений?
2. Что должна включать в себя математическая модель в виде обыкновенных дифференциальных уравнений?
3. Какими методами осуществляется исследование моделей, заданных в виде обыкновенных дифференциальных уравнений?
4. Запишите математическую модель движения груза массой m, закрепленного на вертикальной стенке с помощью пружины жесткостью C и совершающего колебательное движение вдоль оси x в среде с вязкостью ν.
5. Какой принцип используется при построении этой модели?
6. К какому типу относится эта модель?
7. Где используются математические модели в виде дифференциальных уравнений в частных производных?
8. Что является особенностью математических моделей в виде дифференциальных уравнений в частных производных?
9. Что должна включать в себя математическая модель в виде дифференциальных уравнений в частных производных?
10. Какого типа бывают граничные условия?
11. Приведите математическую модель распределения температурного поля в металлическом прутке, нагреваемом с одной стороны.

2.7. Стохастические модели

Точные величины и зависимости, используемые в детерминированных моделях, представляют собой лишь некоторые средние значения (математические ожидания) реальных случайных величин (зависимостей). Так, физические константы, характеризующие материалы и рабочие тела (предел прочности материала σ, теплопроводность λ, плотность ρ и т.д.) меняются в зависимости от партии материала и условий окружающей среды. Всегда имеется определенный разброс размеров деталей l, расходов топлива в системах подачи. Все это приводит к тому, что и результирующие функции, характеризующие процесс, также носят случайный характер. Результаты, полученные с помощью детерминированной модели, представляют собой математические ожидания этих характеристик. При этом конкретные данные для конкретной системы могут существенно отличаться от этих математических ожиданий. Например, ресурс конкретного двигателя может существенно отличаться от среднего ресурса двигателей данного типа. Для учета таких отличий вводятся всевозможные «запасы прочности», призванные гарантировать работоспособность реальных объектов при неблагоприятном стечении обстоятельств.

> *Значительно более полные и объективные результаты можно получить при переходе от детерминированных к стохастическим моделям, то есть при переходе от точно заданных величин к соответствующим случайным величинам.*

При этом константы (σ, λ, ρ, l,…) заменяются случайными величинами ξ_σ, ξ_λ, ξ_ρ, ξ_l,… , подчиненными определенным законам распределения.

Однократное исследование стохастической модели приведет к некоторой случайной величине функции отклика ξ_W, представляющей собой, вообще говоря, ограниченную ценность. Для получения значимых результатов необходимо провести многократное исследование модели и получить распределение результирующей характеристики в интересующем исследователя диапазоне. Поверхность отклика в этом случае представляет собой некий размытый слой переменной плотности.

> *Такой метод исследования стохастической модели получил название метода статистических испытаний или метода Монте-Карло.*

Трудоемкость исследования стохастических моделей существенно выше, чем моделей детерминированных:

1. Значительно возрастает объем исходной информации: замена констант случайными величинами, введение законов распределения этих величин усложняют модель.

2. Для получения распределения результирующей функции необходимо многократное исследование модели.

С другой стороны, полученное при статистическом моделировании распределение характеристик системы дает в руки исследователя чрезвычайно ценную информацию: Такое распределение позволяет оценить не только среднее значение изучаемой величины, но и разброс этих значений, вероятности появления тех или иных значений при конкретном испытании (например, вероятность выхода из строя ДЛА через тот или иной промежуток времени) и их зависимость от различных факторов.

Очень часто используют нормальный или гауссовский закон распределения, для которого плотность вероятности $f(x)$ и функция распределения $P(x)$ задаются следующими соотношениями:

Вероятность того, что случайная величина попадет в интервал $(x,\ x+dx)$:

$$f(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}};$$

Вероятность того, что случайная величина попадет в интервал $(-\infty,\ x)$:

$$P(x)=\frac{1}{\sqrt{2\pi}\sigma}\int\limits_{-\infty}^{x}e^{-\frac{(t-\mu)^2}{2\sigma^2}}dt.$$

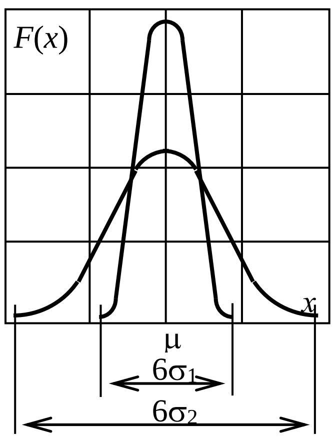


Рис. 2.13

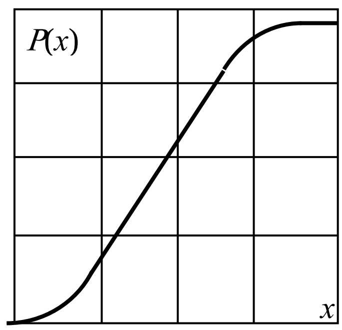


Рис. 2.14

Для случайной величины ξ, распределенной по нормальному закону, $\mu = M(\xi)$, $\sigma = \sigma(\xi)$ (Рис. 2.13, 2.14). Случайная величина распределена в интервале $\mu \pm 3\sigma$. По нормальному закону распределены обычно характеристики материалов, размеры деталей, ресурсы элементов ДЛА.

Наряду с нормальным используются и другие законы распределения случайных величин. Например, равномерное распределение – задает равновероятностные на отрезке $[a,\ b]$ случайные величины. (Рис. 2.15, 2.16). Плотность вероятности и функция распределения при равномерном распределении определяются по формулам:

$$f(x)=\begin{cases}0, & -\infty < x < a;\\ \dfrac{1}{b-a}, & a \le x \le b;\\ 0, & b < x < \infty.\end{cases} \qquad P(x)=\begin{cases}0, & -\infty < x < a;\\ \dfrac{x-a}{b-a}, & a \le x \le b;\\ 1, & b < x < \infty.\end{cases}$$

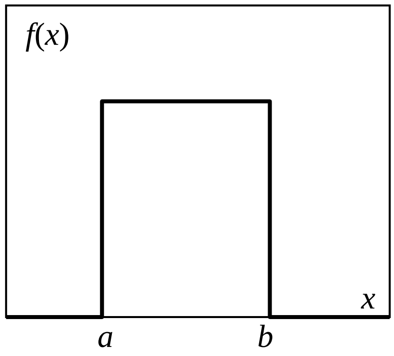


Рис. 2.15

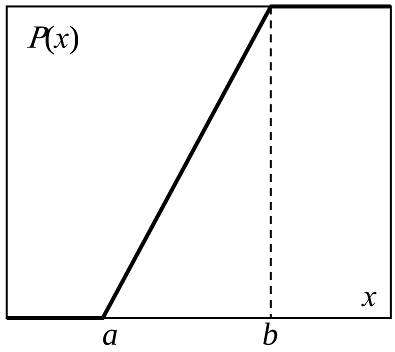


Рис. 2.16

Выбор закона распределения для конкретной случайной величины, входящей в стохастическую модель, может быть обоснован экспериментально или теоретически.

Конкретные параметры распределения (μ, σ,…) всегда определяются на основе экспериментальных данных. Оценка параметров нормального распределения на основе выборки $\{x_i\}$ из n случайных значений величины x дается соотношениями:

$$\mu=\bar{x}=\frac{1}{n}\sum_{i=1}^{n}x_i\,; \qquad \sigma=\sqrt{\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n-1}}\,.$$

При использовании метода статистических испытаний характеристики изучаемой системы оцениваются на основе некоторой ограниченной выборки реализаций. Поэтому важно определить достоверность этой оценки.

Вероятность p пребывания системы в некотором состоянии (например, вероятность того, что время работы элемента ДЛА до первого отказа составит не менее t часов), определяется частотой этого события при моделировании:

$$p \approx \frac{n_+}{n},$$

где n_+ – число реализаций, при которых наблюдалось изучаемое состояние системы (время работы ДЛА до первого отказа превысило t); n – общее число реализаций.

Эта оценка является приближенной, так как определяется на основе ограниченной выборки. Отношение $\frac{n_+}{n}$ называется выборочной статистикой.

Ошибка моделирования определяется отклонением выборочной статистики от вероятности

$$\delta = \left| \frac{n_+}{n} - p \right|.$$

Можно показать, что эта ошибка удовлетворяет неравенству

$$\delta \le \sqrt{\frac{p(1-p)}{\alpha n}}, \qquad (2.20)$$

Здесь p – вероятность рассматриваемого состояния; α – вероятность невыполнения оценки (2.20) (уровень риска). Доверительная вероятность выполнения этой оценки равна 1–α.

Из (2.20) следует, что погрешность стохастического моделирования обратно пропорциональна $\sqrt{n}$. То есть увеличение точности при стохастическом моделировании требует значительного увеличения числа реализаций. Для уменьшения погрешности в 10 раз необходимо увеличить число реализаций (а значит и время счета) в 100 раз. Поэтому метод статистических испытаний не может дать решения с очень высокой степенью точности. Считается, что допустимая ошибка может составлять 1-5% максимальной величины, полученной при моделировании.

Величина ошибки зависит также от вероятности p оцениваемого состояния и допустимого уровня риска α. Обычно α задают на одном из фиксированных уровней (α = 0,005; 0,01; 0,025; 0,05; 0,1 …).

Контрольные вопросы

1. Что представляют собой величины, входящие в стохастическую модель?
2. Что представляет собой поверхность отклика моделей, исследуемых методом статистических испытаний?
3. В чем заключается метод Монте-Карло?
4. Какие трудности возникают при исследовании стохастических моделей?
5. Какую информацию дает в руки исследователя полученное при статистическом исследовании распределение характеристик системы?
6. Какие законы распределения случайной величины Вы знаете?
7. Как выглядит плотность распределения для нормального закона?
8. Как выглядит плотность распределения для закона равной вероятности?
9. Как определяются оценки математического ожидания и дисперсии случайной величины?
10. Что такое выборочная статистика?
11. Почему она называется «выборочная»?
12. От чего зависит погрешность стохастического моделирования?

Глава 3. ЭМПИРИЧЕСКИЕ МАТЕМАТИЧЕСКИЕ МОДЕЛИ

3.1 Идентификация эмпирических математических моделей

Переход к эмпирическим моделям предполагает заведомый отказ от аналитических методов исследования. Поэтому эмпирические модели более разнообразны и включают в себя различные по форме математические зависимости.

При разработке эмпирической математической модели предполагается использование экспериментальных данных, полученных при испытаниях объектов. Результаты таких испытаний всегда представляют собой наборы величин, характеризующих работу объекта или системы при различных сочетаниях управляющих параметров.

Наиболее эффективным средством представления результатов экспериментов в системах математического моделирования являются эмпирические модели.

При построении эмпирической модели обычно предполагается, что физическая теория работы объекта отсутствует или по тем или иным причинам не может быть использована.

Объект идентификации представляет собой так называемый «черный ящик» с некоторым числом регулируемых (или, по крайней мере, измеряемых) входов x и одним или несколькими наблюдаемыми (измеряемыми) выходами (Рис. 3.1).

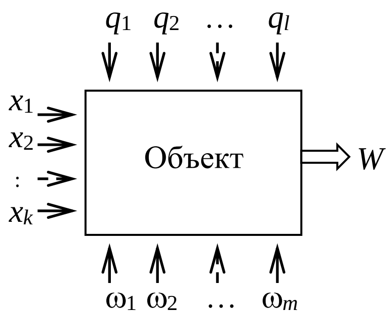


Рис. 3.1

Здесь x_i – управляющие переменные; ω_i – неопределенности (шумы); q_i – ограничения; W – характеристическая функция.

Задачей идентификации является построение модели объекта по результатам наблюдений его реакции на возмущения внешней среды.

При этом необходимо учитывать ошибки, возникающие при измерении характеристик объекта.

Требуется построить зависимость (модель)

$$W = f(x),$$

которая описывает характеристики изучаемой системы.

Это уравнение называется уравнением регрессии и описывает поверхность (гиперповерхность) отклика, характеризующую эмпирическую модель.

Обычно предполагается, что имеющиеся экспериментальные данные дают достаточно информации для воссоздания математического описания объекта.

На рис. 3.2 показано решение задачи идентификации для некоторого набора данных, полученное с помощью линейной регрессионной зависимости: $W = a + bx$.

Идентификацию модели начинают с выбора формы модели, т.е. вида функции $f(x)$.

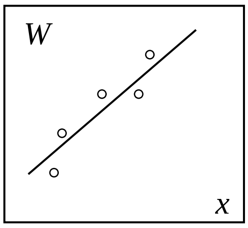


Рис. 3.2

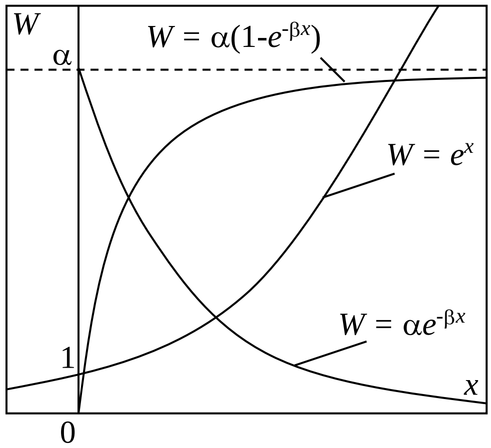


Рис. 3.3

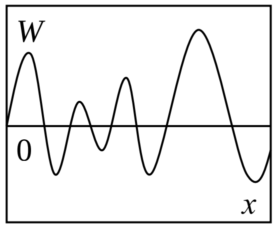


Рис. 3.4

При этом на практике может встретиться два случая:

1) Форма математической модели известна заранее, а задача идентификации сводится к определению коэффициентов этой модели. Так, описание ряда затухающих или развивающихся процессов дается зависимостями экспоненциального типа (Рис. 3.3). Задача исследования является определение коэффициентов α, β.

2) Форма математической модели заранее неизвестна. В этом случае для идентификации модели используются отрезки бесконечных рядов, а задача заключается в определение числа членов ряда и коэффициентов при этих членах. Модель может быть представлена в виде

$$W = \sum_{i=1}^{k}\beta_{0i} f_0(x_i) + \sum_{i=1}^{k}\beta_{1i} f_1(x_i) + \ldots + \sum_{i=1}^{k}\beta_{li} f_l(x_i),$$

где $f_q(x_i)$ – некоторые заданные функции; β_{qi} – коэффициенты регрессии; $q = 0, 1,\ldots, l$.

В одномерном случае ($k = 1$) уравнение принимает вид

$$W = \beta_0 f_0(x) + \beta_1 f_1(x) + \ldots + \beta_l f(x).$$

Конкретный вид модели зависит от выбора функций $f_q(x)$, по которым производится разложение W. Например, при описании колебательных процессов удобно использовать ряд Фурье $W = \alpha_0 + \sum_{k=1}^{\infty}(\alpha_k \cos kx + \beta_k \sin kx)$ (Рис. 3.4).

Часто в качестве функций $f_0(x)$, $f_1(x)$, $f_2(x)$,…, $f_l(x)$ выступают степенные функции x^0, x^1, x^2,…, x^l. Если ограничиться первыми членами разложения, то уравнения сведутся к линейным, квадратичным и другим полиномиальным моделям. Однако пока остается не ясным, сколько членов ряда обеспечивает наилучшее описание изучаемого процесса.

Обычно берут количество экспериментальных точек значительно больше, чем количество коэффициентов регрессии. В этом случае нельзя построить поверхность отклика, проходящую через все экспериментальные точки. Да этого и не требуется. При этом, однако, можно построить приближенную модель, обеспечивающую в некотором смысле наилучшее совпадение с экспериментальными данными.

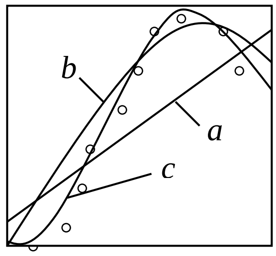


Рис. 3.5

Например, прямая a построена по 10-ти экспериментальным точкам методом наименьших квадратов (Рис. 3.5); кривая b – квадратичная модель; c – полиномиальная модель 3-го порядка достаточно хорошо соответствует исходному экспериментальному материалу, хотя проходит не через все экспериментальные точки.

Таким образом, для любой экспериментальной выборки могут быть предложены различные модели идентификации. Конкретная форма модели зависит от выбора функций $f_q(x)$ и количества членов ряда.

Сама постановка задачи идентификации включает в себя элемент неопределенности, возможность множественности решений. Важно выбрать лучшее или, по крайней мере, достаточно хорошее из этих решений.

Для оценки точности модели естественно использовать величины отклонений, полученных в эксперименте величин W_j и их оценок Wm_j, предсказанных моделью

$$\varepsilon_j = W_j - Wm_j. \tag{3.1}$$

Исключительное распространение получил метод наименьших квадратов отклонений реальных значений оцениваемой величины от значений, предсказанных моделью.

Специальные методы планирования эксперимента позволяют существенно повысить объем получаемой информации, улучшают характеристики эмпирических моделей, а также упрощают процедуру обработки экспериментальных данных. Однако на практике очень часто приходится иметь дело с неорганизованным (пассивным) экспериментом. Связано это, по крайней мере, с тремя причинами:

1) Исследователь может только наблюдать входы системы, но не может их регулировать, что полностью исключает возможность планирования эксперимента (типичная ситуация: астроном – галактика).

2) Неизвестны диапазоны возможного изменения переменных (входов), что затрудняет планирование эксперимента и исключает возможность использования ряда эффективных методов планирования.

3) Приходится строить модели идентификации на основе уже полученных ранее беспорядочных экспериментальных данных.

3.2. Использование метода наименьших квадратов

В качестве простого примера построения модели методом наименьших квадратов рассмотрим задачу восстановления математического описания некоторого процесса по результатам эксперимента.

Предполагается, что процесс описывается одномерным уравнением 2-го порядка

$$W = a_0 + a_1x + a_2x^2, \quad 0 \le x \le 6.$$

Считаем, что величина x измеряется точно, а W – с ошибкой ε, имеющей нормальное распределение с нулевым математическим ожиданием и единичной дисперсией $M(\varepsilon) = 0$, $\sigma^2(\varepsilon) = 1$.

Выборка десяти случайных пар ($x, \tilde{W}$) представлена в табл. 3.1 в графах 2 и 3.

Таблица 3.1

№	x	$\tilde{W}$	Wm	ε
1	2	3	4	5
1	4,8608	9,28	8,848	0,432
2	4,2396	9,40	8,821	0,579
3	2,7792	7,88	7,460	0,420
4	0,5988	1,86	2,039	-0,179
5	3,2136	7,77	8,056	-0,286
6	4,5156	8,73	8,874	-0,144
7	5,9340	8,33	8,118	0,212
8	1,5852	5,16	4,994	0,166
9	4,4880	7,28	8,872	$\boxed{-1,592}$
10	4,0932	9,22	8,767	0,453

Метод наименьших квадратов заключается в том, что неизвестные (искомые) коэффициенты a_0, a_1, a_2 должны минимизировать функцию, представляющую собой сумму квадратов невязок ε_j:

$$G = \sum_{j=1}^{10} \varepsilon_j = \sum_{j=1}^{10} (Wm_j - \tilde{W}_j)^2 .$$

Минимум некоторой функции, как известно, находится в точке (a_0^*, a_1^*, a_2^*), где все частные производные этой функции по переменным a_0, a_1, a_2 равны нулю.

Для определения частных производных, распишем функцию G через ее предполагаемый вид:

$$G = \sum_{j=1}^{10} (a_0 + a_1 x_j + a_2 x_j^2 - \tilde{W}_j)^2 .$$

Возьмем от функции G производные по a_0, a_1, a_2 :

$$\frac{\partial G}{\partial a_0} = \sum_{j=1}^{10} \left[2(a_0 + a_1 x_j + a_2 x_j^2 - \tilde{W}_j) \cdot 1 \right];$$

$$\frac{\partial G}{\partial a_0} = \sum_{j=1}^{10} \left[2(a_0 + a_1 x_j + a_2 x_j^2 - \tilde{W}_j) \cdot x_j \right];$$

$$\frac{\partial G}{\partial a_0} = \sum_{j=1}^{10} \left[2(a_0 + a_1 x_j + a_2 x_j^2 - \tilde{W}_j) \cdot x_j^2 \right].$$

Приравняв эти выражения к нулю и произведя некоторые преобразования, получим систему линейных алгебраических уравнений третьего порядка с тремя неизвестными, коэффициенты которой вычисляются по известным данным из табл. 3.1:

$$\begin{cases} a_0 \cdot 10 + a_1 \sum\limits_{j=1}^{10} x_j + a_2 \sum\limits_{j=1}^{10} x_j^2 = \sum\limits_{j=1}^{10} \tilde{W}_j; \\ a_0 \cdot \sum\limits_{j=1}^{10} x_j + a_1 \sum\limits_{j=1}^{10} x_j^2 + a_2 \sum\limits_{j=1}^{10} x_j^3 = \sum\limits_{j=1}^{10} \tilde{W}_j x_j; \\ a_0 \cdot \sum\limits_{j=1}^{10} x_j^2 + a_1 \sum\limits_{j=1}^{10} x_j^3 + a_2 \sum\limits_{j=1}^{10} x_j^4 = \sum\limits_{j=1}^{10} \tilde{W}_j x_j^2. \end{cases}$$

Решая полученную систему, получим a_0 = –0,161; a_1 = 3,929; a_2 = –0,427.

Таким образом, математическая модель будет иметь вид

$$Wm = -0{,}161 + 3{,}929\, x - 0{,}427 x^2. \qquad (3.2)$$

Проверим адекватность модели методом Фишера. Для этого заполним четвертый и пятый столбцы таблицы 3.1, подставляя в математическую модель (3.2) и затем в формулу (3.1) значения x_j из первого столбца.

Определим число степеней свободы системы по формуле

$$f_s = n - m - 1,$$

где n = 10 – количество экспериментальных точек; m = 3 – количество неизвестных коэффициентов. То есть f_s = 6.

Выборочная дисперсия вычисляется по формуле

$$s^2(\varepsilon) = \frac{1}{f_s} \sum_{j=1}^{10} (\tilde{W}_j - W_{mj})^2 = \frac{1}{6} \sum_{j=1}^{10} (\tilde{W}_j - W_{mj})^2 = 0{,}607 .$$

Критерий Фишера вычисляется по формуле

$$F = \frac{s^2(\varepsilon)}{\sigma^2(W)} = 0{,}607\,.$$

По статистическим таблицам при 5%-м уровне риска (α = 0,05) находим пороговое значение критерия Фишера

$$F_{f,\alpha} = F_{6;0,05} = 2{,}01\,.$$

Так как полученное значение F меньше критического (порогового), гипотеза об адекватности модели реальному процессу принимается.

Контрольные вопросы

1. Что является исходным материалом при построении эмпирической модели?
2. Как используется физическая теория работы объекта при построении эмпирической модели?
3. Что при этом представляет собой объект идентификации?
4. Сформулируйте задачу идентификации.
5. Что такое уравнение регрессии?
6. С чего начинается процесс идентификации?
7. От чего зависит конкретная форма модели?
8. Перечислите причины проведения непланируемого эксперимента.
9. В чем заключается метод наименьших квадратов?

3.3. Статистические методы проверки адекватности математических моделей

Если имеются или могут быть получены необходимые и достоверные экспериментальные данные, для проверки адекватности моделей можно использовать методы математической статистики.

Математически задача проверки адекватности модели формулируется как задача проверки предположения о том, что значение отклика модели W_m отличается от реального отклика системы W не более чем на заданную величину ε:*

$$|W(x) - W_m| = |\varepsilon| \le \varepsilon^*, \quad x \in X^*\,. \tag{3.3}$$

Однако, истинное значение отклика системы никогда неизвестно. Полученный в результате эксперимента отклик $\tilde{W}$ в силу неконтролируемого дрейфа системы, разброса характеристик ее элементов и, наконец, просто ошибок измерения представляет собой случайную величину, отличающуюся от W. Поэтому при сравнении результатов математического и физического экспериментов $(W_{m\,i}, \widetilde{W}_i)$ будет получена совокупность случайных величин $\{\varepsilon_i\}$: $\widetilde{W}_i(x) - W_{m\,i}(x) = \varepsilon_i$, $i = 1,\ldots,n$, среди которых могут оказаться как величины, удовлетворяющие условию (3.3), так и не удовлетворяющие ему.

Можно ли считать, что полученные отклонения ($\varepsilon_i > \varepsilon^*$) объясняются случайными причинами или их наличие должно быть признано существенным, что приводит к отказу от проверяемой модели. Для решения этого вопроса на основе выборки случайных величин $\{\varepsilon_i\}$ строят статистические критерии, по которым оценивают адекватность модели.

Гипотеза об адекватности модели действительности (гипотеза H_0) может быть сформулирована как предположение о том, что полученная совокупность $\{\varepsilon_i\}$ не дает оснований отказаться от рассматриваемой модели. Иными словами, модель удовлетворяет заданной точности ε^*.

Альтернативная гипотеза H_1 состоит в том, что модель не отвечает заданным требованиям (3.3) и, следовательно, должна быть отвергнута.

Так как выборка $\{\varepsilon_i\}$ случайна, решение о выборе одной из гипотез H_0 или H_1 носит вероятностный характер. При этом может быть допущена ошибка первого рода, состоящая в отказе от правильной модели (принимается H_1, когда верна H_0), или ошибка второго рода, состоящая в принятии ошибочной модели (принимается H_0, когда верна H_1). Вероятность ошибки первого рода обозначают через α, второго рода – β. Принято называть α риском разработчика, β – риском потребителя. Разумеется, желательно минимизировать как α, так и β. Однако, при заданном объеме экспериментальной выборки уменьшение α влечет за собой увеличение β.

На практике α задается на определенном уровне (α = 0,05; 0,01; 0,005; 0,001), при этом в 100α% случаев правильная модель отвергается.

Величина $1-\beta$ характеризует вероятность отказа от ошибочной модели, называется мощностью критерия и является мерой его эффективности.

Выбор вероятностей ошибок α и β при проверке конкретной модели зависит от ответственности решений, принимаемых на основе моделирования.

Например, если модель предназначена для управления двигателем летательного аппарата, необходимо в первую очередь минимизировать β, так как в данном случае принятие неверной модели, а значит, возможность ошибочных решений при управлении представляет больший вред, чем отказ от правильной модели.

Для оценки гипотезы об адекватности модели существует несколько критериев:

1) Критерий согласия χ^2 Пирсона.

2) Критерий Смирнова-Колмогорова.

3) Критерий Фишера и др.

При использовании критерия χ^2 проверке подлежит гипотеза о том, что рассматриваемая модель адекватна исследуемой системе с вероятностью p (например, $p = 0{,}95$). Это значит, что при n независимых испытаниях np значений ε_i должно удовлетворять условию (3.3) и лишь в $(1-p)n$ случаях это условие может быть нарушено.

В результате случайного эксперимента для этих событий будут получены частоты ν_1 и ν_2: $\nu_1 \approx pn$; $\nu_2 \approx (1-p)n$; $(\nu_1 + \nu_2 = n)$.

Частоты ν_1 и ν_2 отличаются от точных вероятностных оценок или из-за несоответствия модели действительности (заданная вероятность p не соблюдается), или из-за случайных отклонений.

Для оценки предположения о том, что отклонения ν_1 и ν_2 от соответствующих вероятностей случайны, строится функция

$$U^* = \frac{(\nu_1 - pn)^2}{pn} + \frac{[\nu_2 - (1-p)n]^2}{(1-p)n},$$

представляющая собой сумму квадратов отклонений, нормированных на соответствующие вероятности.

Полученное значение U^* сравнивается с табличным значением при заданном уровне риска α. Если U^* превышает пороговое значение $\chi^2_{1,\alpha}$, модель должна быть отвергнута, и принимается гипотеза H_1. Если $U^* \le \chi^2_{1,\alpha}$, экспериментальные данные не противоречат гипотезе об адекватности модели, и принимается гипотеза H_0.

Необходимым условием использования критерия χ^2 является многочисленность экспериментальных данных (не меньше 20).

Критерий Смирнова-Колмогорова основан на максимальном значении отклонений

$$S = \sup\{\varepsilon_i\} = \sup\{\tilde{W}_i - W_{m\,i}\}.$$

Для заданной экспериментальной выборки строится вспомогательная функция

$$\lambda_n^* = S\sqrt{n},$$

которая сравнивается с пороговым значением $\lambda_{n,\alpha}$, определенным по таблицам распределения функции Смирнова-Колмогорова.

При $\lambda_n^* > \lambda_{n,\alpha}$ модель должна быть отвергнута, а при $\lambda_n^* \le \lambda_{n,\alpha}$ экспериментальные данные не противоречат гипотезе об адекватности модели.

Критерий Смирнова-Колмогорова целесообразно использовать при относительно малых выборках, когда критерий χ^2 оказывается неэффективным.

Критерий Фишера осуществляется путем анализа дисперсий. Если дисперсия, характеризующая ошибку эксперимента $\sigma^2(W)$, известна, вычисляется выборочная дисперсия $S^2(\varepsilon)$ и составляется F-отношение:

$$F_{f_S,\infty} = \frac{S^2(\varepsilon)}{\sigma^2(W)}.$$

Полученную величину F-отношения сравнивают с пороговым значением критерия Фишера $F_{f\,s,\infty,\,\alpha}$ при заданном уровне риска α.

При $F_{f\,s,\infty} \le F_{f\,s,\infty,\,\alpha}$ полученная величина $S^2(\varepsilon)$ может быть объяснена случайным разбросом экспериментальных данных и, следовательно, нет оснований для отказа от проверяемой модели.

Если $F_{f\,s,\infty} > F_{f\,s,\infty,\,\alpha}$, полученное расхождение результатов моделирования и экспериментальных данных зна́чимо и, следовательно, модель должна быть отвергнута как недостаточно точная.

Контрольные вопросы

1. Сформулируйте задачу проверки адекватности модели.
2. Что означает понятие «адекватность математической модели»?
3. В чем заключается ошибка первого рода?
4. В чем заключается ошибка второго рода?
5. Какие критерии проверки адекватности математической модели Вы знаете?
6. Охарактеризуйте каждый из этих критериев.

3.4. Идентификация параметров математической модели силы резания токарной операции

Построим математическую модель силы резания при обработке круглой детали на токарном станке (Рис. 3.6).

Сила резания P описывается математической моделью в виде позинома

$$P = C S^{\alpha} V^{\beta} t^{\gamma}, \tag{3.4}$$

где S – продольная подача; V – скорость резания; t – глубина резания; C, α, β, γ – неизвестные параметры.

Формула (3.4) является справочной. Для определения неизвестных параметров воспользуемся методом наименьших квадратов.

Пусть проведено n экспериментов, результаты которых сведены в таблицу 3.2.

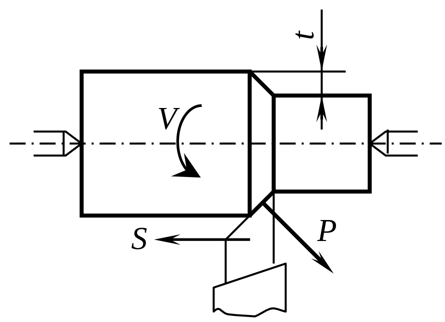


Рис. 3.6

Таблица 3.2

№	S(мм/об)	V(мм/с)	t(мм)	P(Кг)
1	S_1	V_1	t_1	P_1
2	S_2	V_2	t_2	P_2
⋮	⋮	⋮	⋮	⋮
n	S_n	V_n	t_n	P_n

Для упрощения решения поставленной задачи прологарифмируем выражение (3.4):

$$\ln P = \ln C + \alpha \ln S + \beta \ln V + \gamma \ln t .$$

Введем обозначения $\overline{P} = \ln P$; $\overline{C} = \ln C$; $\overline{S} = \ln S$; $\bar{t} = \ln t$.

Тогда формула (3.4) преобразуется к линейному виду:

$$\overline{P} = \overline{C} + \alpha \overline{S} + \beta \overline{V} + \gamma \bar{t} . \tag{3.5}$$

Метод наименьших квадратов сведется к минимизации функции

$$G = \sum_{i=1}^{n} (\overline{P}_i - \overline{P}_{mi})^2 = \sum_{r=1}^{n} (\overline{P}_i - \overline{C} - \alpha \overline{S}_i - \beta \overline{V}_i - \gamma \bar{t}_i)^2 \to \min ,$$

где $\overline{P}_i$ – логарифмы экспериментальных значений силы резания, взятых из табл. 3.2; $\overline{P}_{mi}$ – логарифмы силы резания, предсказанные с помощью математической модели (3.5), $\overline{S}_i, \overline{V}_i, \bar{t}_i$ – логарифмы экспериментальных значений подачи, скорости и глубины резания, взятых из той же табл. (3.2), $\overline{C}$ – логарифм неизвестного параметра C.

Возьмем производные от функции G по $\overline{C}, \alpha, \beta, \gamma$ и приравняем их к нулю:

$$\frac{\partial G}{\partial C} = -2\sum_{i=1}^{n}(\overline{P_i} - \overline{C} - \alpha\overline{S_i} - \beta\overline{V_i} - \gamma\bar{t}_i) = 0;$$

$$\frac{\partial G}{\partial \alpha} = -2\sum_{i=1}^{n}(\overline{P_i} - \overline{C} - \alpha\overline{S_i} - \beta\overline{V_i} - \gamma\bar{t}_i)\overline{S_i} = 0;$$

$$\frac{\partial G}{\partial \beta} = -2\sum_{i=1}^{n}(\overline{P_i} - \overline{C} - \alpha\overline{S_i} - \beta\overline{V_i} - \gamma\bar{t}_i)\overline{V_r} = 0;$$

$$\frac{\partial G}{\partial \gamma} = -2\sum_{i=1}^{n}(\overline{P_i} - \overline{C} - \alpha\overline{S_i} - \beta\overline{V_i} - \gamma\bar{t}_r)\bar{t}_i = 0.$$

Разделим обе части уравнений на –2; вынесем $\overline{C}$, α, β, γ за знак суммы; перенесем члены, не зависящие от $\overline{C}$, α, β, γ, в правую часть:

$$\begin{aligned}
&n\overline{C} + \alpha\sum_{i=1}^{n}\overline{S_i} + \beta\sum_{i=1}^{n}\overline{V_i} + \gamma\sum_{i=1}^{n}\bar{t}_i = \sum_{i=1}^{n}\overline{P_i};\\
&\overline{C}\sum_{i=1}^{n}\overline{S_i} + \alpha\sum_{i=1}^{n}\overline{S_i}^2 + \beta\sum_{i=1}^{n}\overline{V_i}\overline{S_i} + \gamma\sum_{i=1}^{n}\bar{t}_i\overline{S_i} = \sum_{i=1}^{n}\overline{P_i}\overline{S_i};\\
&\overline{C}\sum_{i=1}^{n}\overline{V_i} + \alpha\sum_{i=1}^{n}\overline{S_i}\overline{V_i} + \beta\sum_{i=1}^{n}\overline{V_i}^2 + \gamma\sum_{i=1}^{n}\bar{t}_i\overline{V_i} = \sum_{i=1}^{n}\overline{P_i}\overline{V_i};\\
&\overline{C}\sum_{i=1}^{n}\bar{t}_i + \alpha\sum_{i=1}^{n}\overline{S_i}\bar{t}_i + \beta\sum_{i=1}^{n}\overline{V_i}\bar{t}_i + \gamma\sum_{i=1}^{n}\bar{t}_i^2 = \sum_{i=1}^{n}\overline{P_i}\bar{t}_i.
\end{aligned} \tag{3.6}$$

Получили систему линейных алгебраических уравнений четвертого порядка, коэффициентами которой являются суммы произведений логарифмов экспериментальных данных. Решив полученную систему, найдем искомые значения коэффициентов $\overline{C}$, α, β, γ линейной модели (3.5).

Для определения параметров исходной модели (3.4) необходимо для коэффициента C (<u>только для него</u>) проделать операцию, обратную логарифмированию – потенцирование: $C = e^{\overline{C}}$. Коэффициенты α, β, γ получаются непосредственно из решения системы (3.6).

Если в распоряжении исследователя имеются экспериментальные данные, для проверки адекватности математической модели действительности можно использовать методы математической статистики. Рассматриваемый ниже метод пригоден при изучении любых математических моделей. Однако конкретный анализ проводится на примере построенной модели силы резания при точении с помощью критерия согласия χ^2, предложенного Пирсоном.

Гипотеза H_0 формулируется как предположение о том, что отклонение ε экспериментальных данных от значений, предсказанных моделью (3.4), с вероятностью p (доверительная вероятность) укладываются в некоторый толерантный интервал ±ε*. Если это предположение правильно, то в толерантный интервал ($P \pm \varepsilon^*$) должно укладываться np отклонений $\varepsilon_i = |P_i - P_{mi}|$. Вне толерантного интервала должно оказаться $(1-p)n$ отклонений. Для ограниченной случайной выборки из n наблюдений эти события будут наблюдаться с частотой ν_1 и ν_2, лишь приближенно совпадающие с соответствующими вероятностями:

$$\nu_1 \approx pn; \;\; \nu_2 \approx (1-p)n; \;\; \nu_1 + \nu_2 = n.$$

Необходимо установить, можно ли объяснить эти отклонения случайными причинами (в этом случае можно принять гипотезу H_0) или же они не случайны – статистически значимы (в этом случае нужно принять альтернативную гипотезу H_1).

Для этого вычисляется некоторая величина U, называемая статистикой:

$$U = \frac{(\nu_1 - np)^2}{np} + \frac{\left(\nu_2 - (1-p)n\right)^2}{(1-p)n}.$$

Эту величину нужно сравнить с пороговым значением χ^2-критерия ($\chi^2_{1,\alpha}$) при принятом уровне риска α. Если $U \le \chi^2_{1,\alpha}$, наблюдаемые отклонения частот от соответствующих вероятностей можно объяснить случайностью и нет оснований для отказа от нуль-гипотезы H_0. Если $U > \chi^2_{1,\alpha}$, то или произошло маловероятное событие $(1-p)$, или наблюдаемые отклонения не случайны. В этом случае принимается гипотеза H_1.

Вывод о правильности гипотезы H_1, вообще говоря, не требует безоговорочного отказа от проверяемой модели:

1) Можно изменить исходные предположения с тем, чтобы увеличить толерантный интервал $\pm\varepsilon^*$ или уменьшить доверительную вероятность p. При этом умéньшатся отклонения ν_1 и ν_2 от соответствующих вероятностей, и проверка может привести к принятию гипотезы H_0. В этом случае моделью можно пользоваться, но нужно признать, что ее точность оказалась ниже, чем первоначально предполагалось.

2) Можно уменьшить уровень риска α (то есть вероятность отказа от правильной модели в результате неудачного эксперимента). Это приводит к увеличению порогового значения $\chi^2_{1,\alpha}$. Это, в свою очередь, может изменить оценку значения U. Однако нужно помнить, что при этом увеличивается риск признать правильной ошибочную модель.

3) Можно потребовать увеличения объема выборки, что, разумеется, приведет к увеличению точности оценки модели и уменьшению риска ошибок.

При проверке адекватности моделей действительности всегда рассматривается случай, когда за пределами толерантного интервала оказалось больше точек, чем ожидалось ($\nu_1 < pn$; $\nu_2 > (1-pn)n$). В противном случае опасений за точность модели не возникает, однако можно предположить, что величина толерантного интервала задана необоснованно большой. Если в результате проверки по критерию χ^2 в этом случае будет получена величина $U > \chi^2_{1,\alpha}$, то завышение толерантного интервала (или занижение доверительной вероятности *p*) статистически значимо, и необходимо уменьшить ε* или увеличить *p*. В обоих случаях нужно признать, что модель оказалась точнее, чем ожидалось.

Контрольные вопросы

1. Приведите общий вид математической модели силы резания при точении.
2. Как привести модель, заданную в виде позинома, к линейному виду?
3. Каким методом найдены параметры линейной модели?
4. В чем заключается этот метод?
5. Как перейти от линейной модели к позиному?
6. Сформулируйте нуль-гипотезу проверки построенной модели на адекватность.
7. Что такое доверительная вероятность?
8. Перечислите меры, которые можно применить в случае неадекватности построенной математической модели.
9. В каком случае можно не проверять модель на адекватность?

3.5. Выбор оптимальной эмпирической модели

Принцип наименьших квадратов позволяет найти наилучшую модель идентификации для исследуемой экспериментальной выборки с заданным уравнением регрессии вида

$$W = \sum_{i=1}^{k} \beta_{0i} f_0(x_i) + \sum_{i=1}^{k} \beta_{1i} f_1(x_i) + \ldots + \sum_{i=1}^{k} \beta_{li} f_l(x_i).$$

Если имеются достаточно веские основания для выбора формы этого уравнения, никаких проблем не возникает. Однако, в большинстве случаев конкретная форма модели заранее неизвестна и может, вообще говоря, быть различной.

На первый взгляд может показаться, что более сложная модель (увеличение степени полинома) всегда обеспечивает получение бо́льшей точности. На самом деле это не так. При переходе к полиномам более высокой степени можно, конечно, получить лучшее согласие регрессионной кривой с экспериментальными данными. Для $m = n$ это согласие будет абсолютным, но при этом получится худшее согласие с истинным характером процесса $W(x)$. Дело в том, что экспериментальные данные представляют собой случайные величины и содержат лишь ограниченную информацию о характере $W(x)$. Увеличение степени полинома целесообразно лишь до тех пор, пока из экспериментальной выборки извлекается надежная информация. Таким образом, возникает проблема выбора формы модели.

Подход к решению этой проблемы основан на статистическом исследовании уравнений регрессии.

1) Метод всех возможных регрессий основан на последовательном изучении всех возможных моделей ($m < n$), из которых отбирается лучшая модель.

Метод представляется мало пригодным для анализа сложных систем, так как отличается высокой трудоемкостью.

2) Метод исключения предполагает исследование наиболее полной (в пределах разумного) модели и последовательную проверку на значимость всех ее членов. При этом для каждого из членов модели вычисляется величина критерия Фишера F. На основе полученного множества $\{F_i\}$ выбирается член уравнения регрессии, соответствующий минимальному значению критерия F_i . Если это минимальное значение меньше критического при выбранном уровне риска ($F_i < F_{\text{кр } \alpha}$), то соответствующий член исключается из регрессионного уравнения как несущественный, после чего все коэффициенты регрессии пересчитываются заново и вновь осуществляется проверка их значимости.

Если $F_i > F_{\text{кр } \alpha}$, то все члены модели существенны и уравнение регрессии остается в первоначальном виде. Однако, если это произошло уже на первом шаге исследования, сто́ит рассмотреть целесообразность усложнения первоначальной модели.

Трудоемкость этого метода меньше, чем метода всех возможных регрессий.

3) Метод включений по существу противоположен методу исключений и предусматривает последовательное включение в модель новых членов с проверкой их статистической значимости.

Трудоемкость этого метода существенно меньше трудоемкости рассмотренных выше методов.

Существуют и некоторые другие методы подбора оптимального уравнения регрессии.

Общим недостатком всех рассмотренных ранее методов является использование для оценки модели того же экспериментального материала, на основе которого эта модель построена.

4) Иной подход основан на использовании регуляризации. При этом подходе все экспериментальные данные разбиваются на две части: обучающую (n_1) и проверочную (n_2). Первая из них используется для определения коэффициентов регрессии модели, вторая – для оценки модели в целом.

Оптимальные по этому подходу модели мало чувствительны к небольшим изменениям исходных данных.

Число точек обучающей последовательности должно быть, по крайней мере, на единицу больше числа коэффициентов регрессии ($n_1 > m+1$). Для повышения достоверности результатов этот запас должен быть существенно увеличен ($n_1 \geq (2\ldots3)m$). Проверочная последовательность должна включать в себя хотя бы одну точку.

В ряде случаев в качестве критерия регуляризации удобно использовать критерий несмещенности, обеспечивающий наименьшее изменение модели при изменении состава обучающей последовательности. При этом весь экспериментальный массив разбивается на две одинаковые по величине последовательности ($n_1 = n_2$), каждая из которых поочередно используется в качестве обучающей. В результате их использования определяются две независимые, одинаковые по форме модели $W_m^*(n_1)$ и $W_m^{**}(n_2)$. Оптимальная модель ищется по всем точкам выборки:

> *Критерий регуляризации всегда имеет четко выраженный минимум, что обеспечивает объективное выделение модели оптимальной сложности.*

3.6. Использование критерия Фишера для проверки значимости высших степеней математической модели

Критерий Фишера может быть использован для сравнения точности двух (или нескольких) конкурирующих моделей.

Пусть рассматриваются две модели изучаемой системы (ω_1, ω_2), приводящие к двум различным множествам значений функции отклика: W_{m1}, W_{m2}.

Будем считать, что модель ω_2 более подробна и предположительно более точна, чем ω_1. Для каждой из моделей может быть составлена остаточная сумма квадратов:

$$S_1 = \sum_{i=1}^{n}(W_i - W_{m1i})^2; \;\; f_{S_1} = n - m_1 - 1;$$

$$S_2 = \sum_{i=1}^{n}(W_i - W_{m2i})^2; \;\; f_{S_2} = n - m_2 - 1$$

и подсчитаны средние квадраты этих сумм (выборочные дисперсии):

$$s_1^2 = \frac{S_1}{f_{S_1}} = \frac{\sum_{i=1}^{n}(W_i - W_{m1i})^2}{n - m_1 - 1}; \;\; s_2^2 = \frac{S_2}{f_{S_2}} = \frac{\sum_{i=1}^{n}(W_i - W_{m2i})^2}{n - m_2 - 1}.$$

Для сравнения моделей подсчитывается так называемая дополнительная сумма квадратов SS, связанная с дополнительными данными, введенными в модель ω_2 , и характеризующаяся внесенными в нее уточнениями; а также число степеней свободы этой дополнительной суммы квадратов:

$$SS = S_1 - S_2; \;\; f_{SS} = f_{S1} - f_{S2} = m_2 - m_1.$$

Средний квадрат дополнительной суммы определяется соотношением

$$s_{SS}^2 = \frac{SS}{f_{SS}} = \frac{\sum_{i=1}^{n}(W_i - W_{m1i})^2 - \sum_{i=1}^{n}(W_i - W_{m2i})^2}{m_2 - m_1}.$$

Если известна дисперсия экспериментальных данных $\sigma^2(W)$, то роль дополнительной информации, содержащейся в модели ω_2, оценивается путем сравнения F-отношения с пороговым (критическим) значением критерия Фишера:

$$F_1 = \frac{s_{SS}^2}{\sigma^2(W)}.$$

Если дисперсия экспериментальных данных $\sigma^2(W)$ неизвестна, сравнение проводится с оценкой дисперсии для упрощенной модели s_1^2:

$$F_2 = \frac{s_{SS}^2}{s_1^2}.$$

Если полученное значение критерия Фишера значимо:

$$F_1 > F_{m_2 - m_1, \alpha} \text{ или } F_2 > F_{m_2 - m_1, n - m_1 - 1, \alpha},$$

то дополнительная информация, заложенная в модели ω_2, существенна, и модель ω_2 действительно отличается от модели ω_1. В противном случае уточнения, вносимые моделью ω_2, неразличимы на фоне шума; с точки зрения точности модели равноценны, и предпочтение должно быть отдано более простой модели ω_1.

В частном случае полиномиальных моделей, представляющих собой конечные отрезки бесконечных рядов, этим методом можно проверить целесообразность включения в модель членов рада с более высокими степенями.

Рассмотрим пример, приведенный в п. 3.2 (лекция 7) с построенной моделью в виде полинома второй степени. Проанализируем целесообразность использования для данной выборки кубической модели типа $W_m = a_0 + a_1x + a_2x^2 + \boxed{a_3x^3}$.

В результате расчетов методом наименьших квадратов можно найти коэффициенты такой модели:

$$W_m = -0{,}878 + 4{,}98x - 0{,}768x^2 + 0{,}032x^3.$$

Вычислим остаточную сумму квадратов:

$$S_3 = \sum_{i=1}^{10}(W_i - W_{mi})^2 = 4{,}537.$$

Напомним, что остаточная сумма квадратов для модели второго порядка имела значение $S_2 = 4{,}393$.

Дополнительная сумма квадратов

$$SS = S_2 - S_3 = 0{,}144; \quad f_{SS} = m_3 - m_2 = 4 - 3 = 1.$$

Средний квадрат дополнительной суммы

$$s_{SS}^2 = \frac{SS}{f_{SS}} = 0{,}144.$$

Критерий Фишера

$$F = \frac{s_{SS}^2}{s_2^2} = \frac{0{,}144}{0{,}607} = 0{,}2206.$$

По статистическим таблицам распределения определяем критическое (пороговое) значение критерия Фишера при количестве степеней свободы $f = n - m_2 - 1 = 10 - 3 - 1 = 6$ и уровне риска $\alpha = 0{,}05$

$$F_{f_{SS}, f, \alpha} = 5{,}59.$$

В связи с тем, что полученное из расчета значение критерия Фишера меньше критического, можно считать, что член третьего порядка не добавляет существенной информации и, следовательно, он является незначимым. Переход к модели третьего порядка нецелесообразен.

Контрольные вопросы

1. Обеспечивает ли получение бо́льшей точности более сложная модель?

2. Перечислите методы выбора оптимальной модели.
3. На чем основан метод всех возможных регрессий?
4. На чем основан метод исключения?
5. На чем основан метод включений?
6. На чем основан подход регуляризации?
7. Опишите критерий проверки значимости высших степеней математической модели.

Глава 4. МАТЕМАТИЧЕСКИЕ МОДЕЛИ ТЕОРИИ ПРИНЯТИЯ РЕШЕНИЙ

4.1. Общие сведения о теории принятия решений

Принятие решений является основой любой деятельности человека.

Простейшая схема принятия решений включает в себя некоторую цель и совокуп-

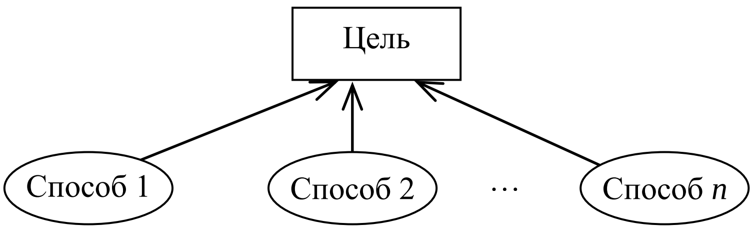


Рис. 4.1

ность способов ее достижения (Рис. 4.1).

> *Под* <u>*целью*</u> *будем понимать конкретный конечный результат, который необходимо получить путем выбора и реализации тех или иных способов деятельности.*
>
> *При этом в любом процессе принятия решений обязательно присутствует* <u>*субъект принятия решений*</u>*, который в общем случае представляется группой лиц, ответственных за целеполагание, формирование вариантов способов действий и, главное, за выбор конкретного решения.*

При формировании оптимальных решений обязательным является наличие <u>критериев оптимальности решений</u> или по-другому <u>целевых функций</u>.

> *Критерием оптимальности называется математическое выражение, позволяющее количественно оценить степень достижения поставленной цели при выборе того или иного решения.*

Задача ПР называется <u>однокритериальной</u>, если выбираемое решение служит достижению одной цели. Например, выбор управленческого решения по производственной программе предприятия, позволяющего получить максимум прибыли (цель) от реализации продукции.

Во многих ситуациях ПР объективно присутствует несколько целей.

Задачи ПР, удовлетворяющих нескольким целям, называются <u>многокритериальными</u> задачами. Например, при выборе проектных решений по новому пассажирскому самолету требуется обеспечить максимальное число пассажиров (цель 1) при минимальном расходе топлива (цель 2).

Отметим, что если в однокритериальных задачах возможно получение единственного оптимального решения (Рис. 4.2 а), то в многокритериальных ЗПР такая возможность отсутствует (Рис. 4.2 б).

В многокритериальных задачах возможно получение совокупности компромиссных

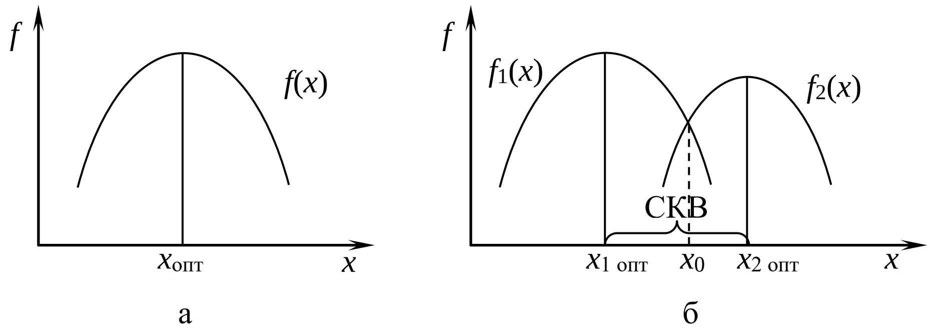


Рис. 4.2

вариантов (СКВ) решений на интервале $[x_{1\ опт}, x_{2\ опт}]$.

4.2. Общая математическая модель формирования оптимальных решений

В математических моделях принятия решений в качестве нового знания выступает оптимальное решение, которое в наилучшем смысле соответствует достижению поставленной цели (целей).

Введем в рассмотрение n-мерный вектор $X = (x_1, x_2, \ldots, x_n)$, определяющий количественные характеристики формируемого решения.

Обозначим через a, b, c вектора соответствующих размерностей, описывающие количественные характеристики неконтролируемых факторов.

Для оценки эффективности различных вариантов решений будем использовать специальным образом сформированную функцию:

$$W = f(c, X),$$

которая называется критерием оптимальности решений или целевой функцией задачи ПР.

Тогда выбор оптимального решения $X_{опт}$ будем осуществлять, исходя из требования $W(x) \to \max_{x \in X}(\min)$.

Множество X должно быть допустимым с точки зрения учета условий принятия решений (ограничений).

Пусть ЛПР обладает для достижения цели вектором ресурсов b. Представим в виде вектор-функции $\varphi(a, X)$ фактический расход ресурсов при использовании вектора решений X и вектора некоторых факторов a.

Тогда $\varphi(a, X) \le b$ есть ограничение.

Во многих задачах ПР учитывается условие $X \ge 0$.

Таким образом, общая математическая модель формирования оптимальных решений может быть представлена в следующем виде:

$$W = f(c, X) \to \max; \qquad (4.1)$$

$$\varphi(a, X) \le b; \qquad (4.2)$$

$$X \ge 0. \qquad (4.3)$$

Постановка задачи в этом случае выглядит следующим образом:

Найти значение вектора X, доставляющего максимум (минимум) критерию оптимальности решений (4.1) и удовлетворяющего при этом условиям (4.2) и (4.3).

Математическая модель ПР (4.1) – (4.3) является однокритериальной моделью.

Если ЛПР должен учитывать m целей, то, формализуя их в виде критериев оптимальности, получим:

$$\begin{array}{l} W_1 = f_1(c_1, X) \to \max; \\ W_2 = f_2(c_2, X) \to \min; \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ W_m = f_m(c_m, X) \to \text{extr}; \end{array} \qquad (4.4)$$

где c_1, c_2, …, c_m – вектора неконтролируемых факторов.

Математическая модель (4.4), (4.2), (4.3) является многокритериальной моделью.

В реальных задачах ПР ограничения вида (4.2) могут включать в себя как неравенства вида «≤», «≥», «=», так и их различные сочетания.

4.3. Построение и решение оптимизационной задачи принятия решения (Задача о баке)

Пусть требуется выбрать геометрические размеры цилиндрического бака объемом V из условия минимального расхода материала на его изготовление.

Для построения математической модели введем в рассмотрение вектор проектных решений $X = (r, h)$, где $2r$, h – диаметр и высота бака (Рис. 4.3).

Если предположить, что бак изготавливается сваркой из трех деталей, то расход материала при произвольном векторе решений X будет равен площади поверхности бака:

$$S = 2\pi r^2 + 2\pi rh \to \min_{r,h}. \qquad (4.5)$$

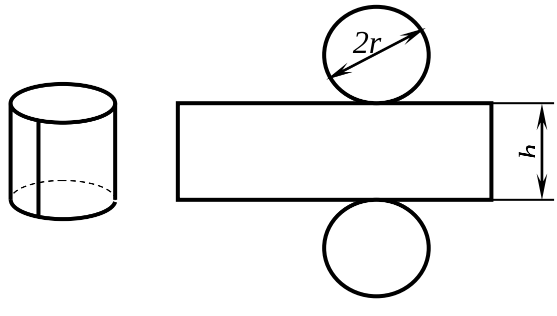


Рис. 4.3

Согласно условиям задачи выражение (4.5) является целевой функцией (критерий оптимальности проектных решений).

Условие того, что бак должен иметь объем заданного значения V, представим в виде:

$$\pi r^2 h = V. \tag{4.6}$$

На компоненты вектора решений X необходимо наложить дополнительные условия:

$$R > 0,\ \ h > 0. \tag{4.7}$$

Выражения (4.5) – (4.7) описывают нелинейную однокритериальную модель формирования оптимальных решений, при $n = 2$, $m = 1$.

Пусть бак должен иметь минимальную трудоемкость его изготовления. Если считать трудоемкости изготовления крышки, дна и боковой стенки достаточно малыми величинами, то затраты времени на изготовление бака будут пропорциональны длине свариваемых швов:

$$T = c(4\pi r + h) \to \min_{r,h}, \tag{4.8}$$

где c – затраты времени на сварку единицы длины.

Выражения (4.5), (4.8), (4.6), (4.7) описывают двухкритериальную нелинейную модель формирования оптимальных решений.

При построении математической модели в этой задаче принятия решений были использованы известные геометрические закономерности.

Аналитическое решение задачи ПР возможно, если соответствующая математическая модель включает в себя ограничения типа равенств, то есть имеет вид:

$$W = f(c, X) \to \underset{X}{\mathrm{extr}};$$
$$\varphi(a, X) = b;$$
$$-\infty < X < \infty.$$

Такие задачи решаются обычно классическими методами условной оптимизации, которые предусматривают построение функции Лагранжа вида

$$\begin{gathered} L(x_1, x_2, ..., x_n, \lambda_1, \lambda_2, ..., \lambda_m) = \\ f(c, x_1, ..., x_n) + \sum_{j=1}^{m} \lambda_j [\varphi_j(a, x_1, ..., x_n) - b_j], \end{gathered} \tag{4.9}$$

где λ_1, λ_2, … , λ_m – неопределенные множители Лагранжа.

Точки экстремума этой функции определяются из решения системы уравнений вида

$$\begin{gathered} \frac{\partial L}{\partial x_i} = 0,\ \ i = 1, ..., n; \\ \frac{\partial L}{\partial \lambda_j} = 0,\ \ j = 1, ..., m. \end{gathered} \tag{4.10}$$

Решая эту систему, получим решение вида

$$x_i^{\text{опт}} = \psi_i(a,b,c),\ i = 1,...,n;$$
$$\lambda_j = \lambda_j(a,b,c),\quad j = 1,...,m. \tag{4.11}$$

Используем этот метод для решения однокритериальной задачи (4.8), (4.6) (без учета (4.5), (4.7)).

Функция Лагранжа имеет вид:

$$L(r,h,\lambda) = c(4\pi r + h) + \lambda(\pi r^2 h - V)\ .$$

Система уравнений (4.17) относительно переменных r, h, λ:

$$\frac{\partial L}{\partial r} = 4\pi c + 2\lambda\pi r h = 0;$$
$$\frac{\partial L}{\partial h} = c + \lambda\pi r^2 = 0;$$
$$\frac{\partial L}{\partial \lambda} = \pi r^2 h - V = 0.$$

Имеем систему алгебраических уравнений, решая которую, получим значения неизвестных r, h (λ находить необязательно):

$$r = \sqrt[3]{\frac{V}{2\pi^2}};\quad h = \sqrt[3]{4\pi V}\ ;\quad \lambda = -c\sqrt[3]{\frac{4\pi}{V^2}}\ .$$

Таким образом, оптимальные размеры бака, найденные с помощью аналитического метода условной оптимизации, не зависят от затрат времени c на сварку единицы длины, но зависят от требуемого объема бака V. Требование (4.8) при этих значениях r и h выполняется, то есть трудоемкость будет минимальной.

Недостатками этого метода являются:

1) Не учитываются в явном виде условия неотрицательности (4.7).

2) Система уравнений (4.10) позволяет получить решение в форме (4.11) только для простых функций (4.5), (4.6).

Контрольные вопросы

1. Что включает в себя простейшая схема принятия решений?
2. Что такое цель?
3. Что такое критерий оптимальности?
4. Что такое однокритериальная ЗПР?
5. Что такое многокритериальная ЗПР?
6. Возможно ли получение единственного оптимального решения в многокритериальных задачах?

7. Напишите общий вид математической модели формирования оптимальных решений.
8. Сформулируйте задачу принятия решений.
9. Запишите критерий минимального расхода материала для задачи о баке.
10. Запишите критерий минимальной трудоемкости для задачи о баке.
11. Запишите общий вид функции Лагранжа.
12. Перечислите недостатки аналитического метода условной оптимизации.

4.4. Многокритериальные задачи принятия решений

Во многих задачах принятия решений имеется несколько целей, которые хочет достичь ЛПР. Такие задачи сводятся к многокритериальным задачам вида:

$$\begin{aligned}
&W_1 = f_1(X) \to \underset{X}{extr};\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&W_k = f_k(X) \to \underset{X}{extr};\\
&\varphi(X) \le 0; \quad X \ge 0,
\end{aligned}$$

где $X\,(x_1, x_2, \ldots x_n)$ – вектор решений.

Наибольшее распространение на практике решения таких задач получил подход, связанный с работами итальянского математика-экономиста Викторио Парето. Он обеспечивает ЛПР возможность гибкого принятия решений. При оптимизации по Парето строится <u>множество «неулучшаемых» решений, изменение каждого из которых ухудшает значение целевых функций $f_1(X), f_2(X), \ldots, f_k(X)$.</u>

Рассмотрим наиболее распространенную на практике двухкритериальную задачу оптимизации вида:

$$\begin{aligned}
&W_1 = f_1(x_1, x_2) \to \max;\\
&W_2 = f_2(x_1, x_2) \to \max;
\end{aligned} \qquad (4.12)$$

$$\begin{aligned}
&\varphi(x_1, x_2) \le 0;\\
&x_1 \ge 0; \quad x_2 \ge 0.
\end{aligned} \qquad (4.13)$$

Условия (4.13) определяют множество допустимых решений и образуют на плоскости x_1Ox_2 некоторую область, каждой точке C которой соответствует точка C^* в пространстве значений критериев W_1OW_2 (Рис. 4.4). Ее координаты $W_1^{C*}; W_2^{C*}$ вычисляются по формулам (4.12) при $x_1 = x_1^C;\ x_2 = x_2^C$:

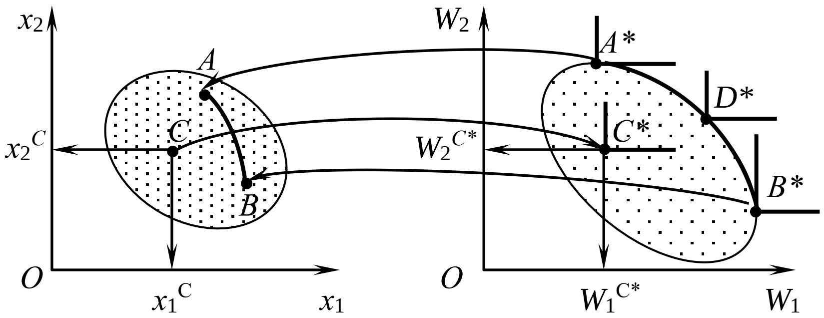


Рис. 4.4

$$W_1^{C^*} = f_1(x_1^C, x_2^C); \quad W_2^{C^*} = f_2(x_1^C, x_2^C).$$

Рассмотрим в множестве значений критериев четыре точки A^*, B^*, D^* и C^* (см. рис. 4.4). Точка A^* является оптимальной для критерия $W_2 = f(x_1,x_2)$, так как в этой точке критерий W_2 имеет максимальное значение. Аналогично точка B^* является оптимальной для критерия $W_1 = f(x_1, x_2)$. Точка C^* является «заведомо плохой» точкой, она не является оптимальной ни для одного критерия, так как в области значений критериев можно найти «более лучшую» точку D^* такую, что $W_1^{D^*} > W_1^{C^*}$; $W_2^{D^*} > W_2^{C^*}$. Для точек A^*, B^*, D^* более «лучших» точек в пространстве значений критериев не существует. Такие точки составляют множество решений, оптимальных по Парето в пространстве значений критериев. В нашем случае это точки кривой $A^*D^*B^*$. Для выделения «лучших» (неулучшаемых) точек используется понятие конуса K_i с вершиной в точке (W_1^i, W_2^i) (Рис. 4.5). Уравнения этого конуса имеют вид: $W_1 \geq W_1^i$; $W_2 \geq W_2^i$.

Правило выделения «лучших» точек:

Если в конусе K_i лежит хотя бы одна точка (W_1^j, W_2^j), то она является более предпочтительной, чем точка (W_1^i, W_2^i) (см. рис. 4.5).

Тогда все точки множества значений критериев, для которых соответствующие конусы являются пустыми, являются парето-оптимальными решениями в пространстве значений критериев.

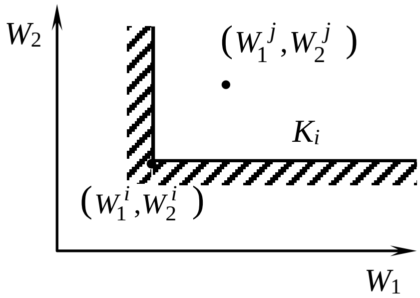


Рис. 4.5

Для нашего примера конусы, построенные во всех точках кривой $A^*D^*B^*$ (см. рис. 4.4), являются пустыми. Строя обратное отображение этих точек в пространство решений X, можно получить множество искомых решений (кривая AB на рис 4.4), оптимальных по Парето. Такое множество называется множеством компромиссов, множеством эффективных точек или множеством Парето. Построив множество компромиссов, ЛПР выбирает в нем из неформальных соображений некоторую точку, которая является наилучшим компромиссом, по мнению ЛПР.

4.5. Построение решений, оптимальных по Парето (Двухкритериальная задача о баке)

Вернемся к рассмотренной в п. 4.5 задаче о баке, описанную формулами (4.5) – (4.8), и попробуем найти паретооптимальное ее решение. Для этого введем параметры $\alpha_1 > 0$ и $\alpha_2 > 0$, удовлетворяющие условию $\alpha_1 + \alpha_2 = 1$, и построим линейную свертку критериев (4.5) и (4.8):

$$F(X, \alpha_1, \alpha_2) = \alpha_1 S(X) + \alpha_2 T(X). \tag{4.14}$$

Положим $\alpha_1 = 1$, тогда α_2 обращается в ноль, свертка (4.14) принимает вид $F(X) = S(X)$ и задача превращается в однокритериальную. Решив эту задачу, найдем оптимальную точку X_S для обеспечения минимального расхода материала. Теперь положим $\alpha_1 = 0$, тогда $\alpha_2 = 1$, свертка (4.14) примет вид $F(X) = T(X)$. Решив эту однокритериальную задачу, найдем оптимальную точку X_T, обеспечивающую минимальную трудоемкость изготовления бака. Если провести аналогию с рис. 7.4, то X_S соответствует точке A, X_T соответствует точке B. Чтобы определить промежуточные точки (остальные компромиссные решения), введем обозначение $\alpha = \alpha_1$. Тогда $\alpha_2 = 1 - \alpha$. Формула (4.14) примет вид $F(X, \alpha) = \alpha S(X) + (1-\alpha)T(X)$. Или:

$$F(r, h, \alpha) = \alpha S(r, h) + (1-\alpha)T(r, h).$$

Функция Лагранжа запишется в виде

$$L(r, h, \lambda) = F(r, h, \alpha) + \lambda\varphi(r, h, \alpha).$$

Распишем функцию Лагранжа подробнее:

$$L(r, h, \lambda) = \alpha(2\pi r^2 + 2\pi rh) + (1-\alpha)C(4\pi r + h) + \lambda(\pi r^2 h - V) \to \min_{r,h,\lambda}.$$

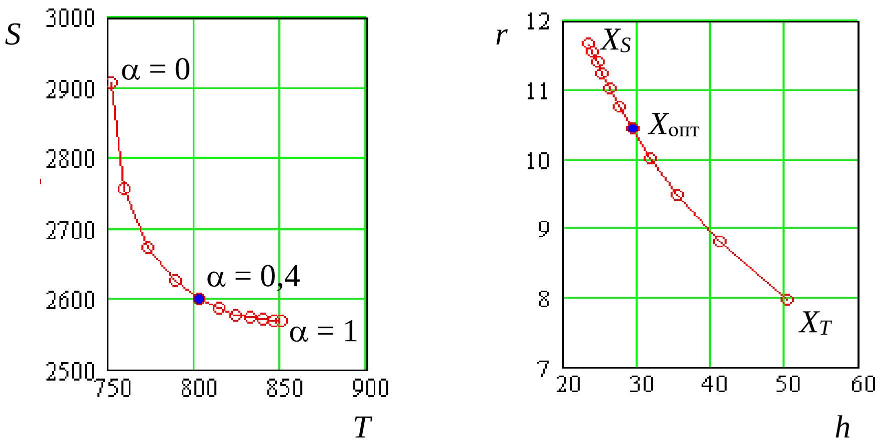
Чтобы найти минимум функции Лагранжа, нужно взять от нее производные по искомым переменным r, h, λ и приравнять их к нулю.

$$\begin{aligned} &\frac{\partial L}{\partial r} = 2\pi\alpha(2r + h) + 4\pi(1-\alpha)C + 2\pi\lambda rh = 0; \\ &\frac{\partial L}{\partial h} = 2\alpha\pi r + (1-\alpha)C + \pi\lambda r^2 = 0; \\ &\frac{\partial L}{\partial \lambda} = \pi r^2 h - V = 0. \end{aligned} \tag{4.15}$$

Получили систему трех алгебраических уравнений, решив которую найдем зависимость r, h, λ от α. Задавая α от 0 до 1, получим множество решений, оптимальных по Парето.

Поскольку аналитически решить систему (4.15) довольно сложно, можно воспользоваться любым численным методом, задавая предварительно значения α с любым приемлемым шагом.

Для примера эта задача была решена с шагом 0,1 в пакете *MathCad*. На рис. 4.6 пока-



зано полученное множество паретооптимальных решений.

Рис. 4.6

ЛПР выбрал из этого множества точку при α = 0,4, при котором S = 2601; T = 802; r = 10,44; h = 29,2. Эта точка устроила его потому, что при дальнейшем увеличении α S уменьшается уже незначительно и T имеет наименьшее значение из всех последующих.

Контрольные вопросы

1. Какие решения называются паретооптимальными?
2. Сформулируйте правило выделения лучших точек.
3. Что такое множество компромиссных решений?
4. Как получить множество компромиссных решений?
5. Запишите функцию Лагранжа для двухкритериальной задачи о баке.
6. Как найти минимум функции Лагранжа?

4. МОДЕЛИРОВАНИЕ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ

4.1. Имитационное моделирование

Имитационное моделирование (англ. Simulation) – самый мощный инструмент исследования сложных систем, управление которыми связано с принятием решений в условиях неопределённости [15].

Имитационное моделирование – есть процесс конструирования модели реальной системы и постановки экспериментов на этой модели с целью:

– либо понять поведение реальной системы;
– либо оценить (в рамках ограничений, накладываемых некоторым критерием или совокупностью критериев) различные стратегии, обеспечивающие функционирование данной системы.

Имитировать (англ. Simulate) – значит вообразить, постичь суть явления, не прибегая к экспериментам на реальном объекте.

Имитационное моделирование является экспериментальной и прикладной методологией, имеющей целью:

– описать поведение системы;
– построить теории и гипотезы, которые могут объяснить наблюдаемое поведение системы;
– использовать эти теории для предсказания будущего поведения системы, т.е. тех воздействий, которые могут быть вызваны изменениями в системе или изменениями способов её функционирования.

Имитационное моделирование получило первоначальный толчок в ходе реализации авиакосмических программ. В настоящее время имитационное моделирование успешно используется во всех областях знаний, что следует из анализа полезности методов исследования в научной работе (табл. 4.1) [15].

Таблица 4.1

Полезность методов исследования в научной работе (по данным США)

Методы	Относительная ценность
Теория вероятностей и статистические оценки	0,182
Экономический анализ	0,150
Имитационное моделирование	0,143
Линейное программирование	0,120
Управление запасами	0,097
Теория массового обслуживания	0,085
Сетевые модели	0,072
Модели замены	0,042
Теория игр	0,040
Динамическое программирование	0,031
Методы поиска	0,020
Нелинейное программирование	0,018
	1,000

ЭВМ открывает широкие возможности исследования систем путём имитационного моделирования.

Смысл и возможности имитационного моделирования могут быть продемонстрированы на следующем примере. Имеется очередь покупателей к прилавку магазина. Интервал времени прибытия покупателей в магазин составляет от одной до десяти минут, а время обслуживания покупателя – от одной до шести минут. Требуется определить среднее время, которое покупатель проводит в магазине (включая ожидание и обслуживание) и коэффициент простаивания продавца.

Решение этой задачи аналитическими методами у большинства людей вызывает затруднение. С помощью метода имитационного моделирования подобную проблему может решить практически каждый. Для её решения требуется поставить искусственный эксперимент, имитирующий процесс прибытия покупателей в магазин и процесс обслуживания. Для реализации модели в данном случае требуется: десять фишек, на каждой из которых номер от 1 до 10; игральный кубик (6 положений); таблица (табл. 4.2).

Проведение эксперимента с данной моделью включает в себя следующие действия:

– вытягиваем фишку, номер на которой определяет интервал прибытия в магазин очередного покупателя (первый покупатель пришёл к открытию магазина);
– бросаем игральный кубик и получаем время его обслуживания;
– полученные результаты заносим в таблицу (первые три колонки табл. 4.2);
– обрабатываем результаты эксперимента и заполняем оставшиеся колонки таблицы.

Таблица 4.2

Имитационное моделирование работы магазина

Покупатель	Интервал прибытия, мин.	Время обслуживания, мин.	Текущее модельное время, мин.	Начало обслуживания, мин.	Конец обслуживания, мин.	Время пребывания покупателя в магазине, мин.	Время простоя продавца, мин.
1	-	1	0	0	1	1	0
2	3	4	3	3	7	4	2
3	7	4	10	10	14	4	3
4	3	2	13	14	16	3	0
5	9	1	22	22	23	1	6
…	…	…	…	…	…	…	…

Всего: 13 11

В результате получаем:

среднее время пребывания покупателя в магазине составляет

$$t_{ср} = \frac{1+4+4+3+1}{5} = \frac{13}{5} = 2{,}6 \quad мин.;$$

коэффициент простаивания продавца

$$k_{пр} = \frac{0+2+3+0+6}{23} = \frac{11}{23} = 0{,}478\,.$$

Для получения статистически значимых результатов число экспериментов должно быть не менее ста.

В настоящее время широчайшие возможности для имитационного моделирования предоставляет вычислительная техника.

Достоинства и недостатки имитационного моделирования [15]. Имитационные модели представляют собой модели типа “чёрный ящик”. Это значит, что они обеспечивают выдачу выходного сигнала, если на их взаимодействующие подсистемы поступает входной сигнал. Поэтому для получения необходимой информации или результатов необходимо осуществлять “прогон” имитационных моделей, а не “решать” их.

Применять имитационное моделирование целесообразно при наличии любого из следующих условий:

1. Не существует законченной математической постановки задачи, либо ещё не разработаны аналитические методы решения сформулированной математической модели.

2. Аналитические методы имеются, но математические процедуры столь сложны и трудоёмки, что имитационное моделирование даёт более простой способ решения задачи.

3. Аналитические решения существуют, но их реализация невозможна вследствие недостаточной математической подготовки имеющегося персонала. В этом случае следует сопоставить затраты на проектирование, испытания и работу на имитационной модели с затратами, связанными с приглашением специалистов со стороны.

4. Кроме оценки определённых параметров желательно осуществлять на имитационной модели наблюдение за ходом процесса в течение определённого периода.

5. Имитационное моделирование может оказаться единственной возможностью вследствие трудностей постановки экспериментов и наблюдения явлений в реальных условиях.

6. Имитационное моделирование даёт возможность полностью контролировать время изучаемого процесса, поскольку явление может быть замедлено или ускорено по желанию.

7. Широчайшие возможности в сфере создания тренажёров.

Недостатки имитационного моделирования:

1. Недостаточное математическое изящество.

2. Разработка хорошей имитационной модели обходится дорого и требует много времени.

3. Имитационная модель в принципе не точна, и мы не в состоянии измерить степень этой неточности. Это затруднение может быть преодолено лишь частично путём анализа чувствительности модели к изменению определённых параметров.

4. Результаты, которые даёт имитационная модель, обычно являются численными. В связи с этим возникает опасность “обожествления чисел”, т.е. приписывания им большей значимости, чем они на самом деле имеют.

5. Имитационные модели не способны формировать своё собственное решение в том виде, в каком это имеет место в аналитических моделях, а могут служить лишь в качестве средства для анализа поведения системы в условиях, которые определяются экспериментатором.

Приведённые соображения показывают, что, хотя имитационное моделирование является чрезвычайно ценным и полезным методом решения сложных задач, этот метод не панацея для решения всех проблем.

Структура имитационных моделей. Имитационная модель представляет собой комбинацию следующих составляющих: компонент, параметров, переменных, функциональных зависимостей, ограничений и целевых функций.

Компоненты – составные части модели. Обычно они соответствуют элементам реальной системы или её подсистемы.

Параметры – величины, которые оператор (исследователь), работающий на модели, может задать произвольно. Параметры, после того как они установлены, являются постоянными величинами, не подлежащими изменению.

Переменные – величины, которые принимают только значения, определяемые видом заданной функции. Разделяются на экзогенные – входные, независимые, существующие вне системы, и эндогенные – выходные, зависимые, порождаемые системой, представляющие собой переменные состояния или выходные переменные.

Функциональные зависимости – соотношения, описывающие поведение переменных и параметров в пределах компонента или выражающие связи между компонентами системы. Эти соотношения или операционные характеристики по своей природе являются либо детерминированными, либо стохастическими. Оба типа соотношений обычно выражаются в форме математического уравнения, которое устанавливает зависимость между эндогенными переменными и экзогенными, и строятся на основе гипотез или с помощью статистического, либо математического анализа.

Ограничения – величины, которые или устанавливают пределы изменения значений переменных, или ограничивают условия распределения и расходования тех или иных средств (энергии, запасов, времени и т.п.). Ограничения бывают искусственными и естественными. *Искусственные ограничения* вводятся разработчиком, их можно изменять (например, требования, предъявляемые к системе). *Естественные ограничения* присущи системе и определяются законами природы.

Целевая функция или функция критерия – это точное отображение целей либо задач системы и необходимых правил их выполнения. Различают два типа целей: сохранение и приобретение. *Цели сохранения* связаны с сохранением или поддержанием каких-либо ресурсов (временных, материальных, энергетических и т.п.) или состояний (комфорта, безопасности, уровня занятости и т.п.). *Цели приобретения* связаны с приобретением новых ресурсов (прибыли, персонала, заказчиков и т.п.) или достижением определённых состояний, к которым стремится организация или руководитель (захват части рынка и т.п.). Выражение для целевой функции должно быть однозначным определением целей и задач, с которыми должны соизмеряться принимаемые решения.

Критерий – мерило оценки, правило или вид проверки, при помощи которого составляется правильное суждение о чём-либо. Функция критерия (целевая функция) направлена на оптимизацию или удовлетворение заданного критерия и должна быть составной частью модели.

Искусство моделирования состоит в способности анализировать проблему, выделять из неё путём абстрагирования её существенные черты, выбирать и должным образом модифицировать основные предположения, а затем отрабатывать и совершенствовать модель до тех пор, пока она не станет давать полезные для практики результаты [15].

Конструирование модели по Моррису рекомендует выполнить следующие действия.

1. Разложить общую задачу исследования системы на ряд более простых задач.

2. Чётко сформулировать цели.

3. Подыскать аналоги.

4, Рассмотреть специальный численный пример, соответствующий данной задаче.

5. Выбрать определённые обозначения.

6. Записать очевидные соотношения.

7. Если полученная модель поддаётся математическому описанию, расширить её. В противном случае упростить:

– превратить переменные величины в константы;
– исключить некоторые переменные или объединить их;
– предложить линейную зависимость между исследуемыми величинами;
– ввести более жёсткие предположения и ограничения;
– наложить на систему более жёсткие граничные условия.

Процесс имитации исследуемой системы включает ряд этапов, взаимосвязь которых отражена на блок-схеме (рис. 4.1).

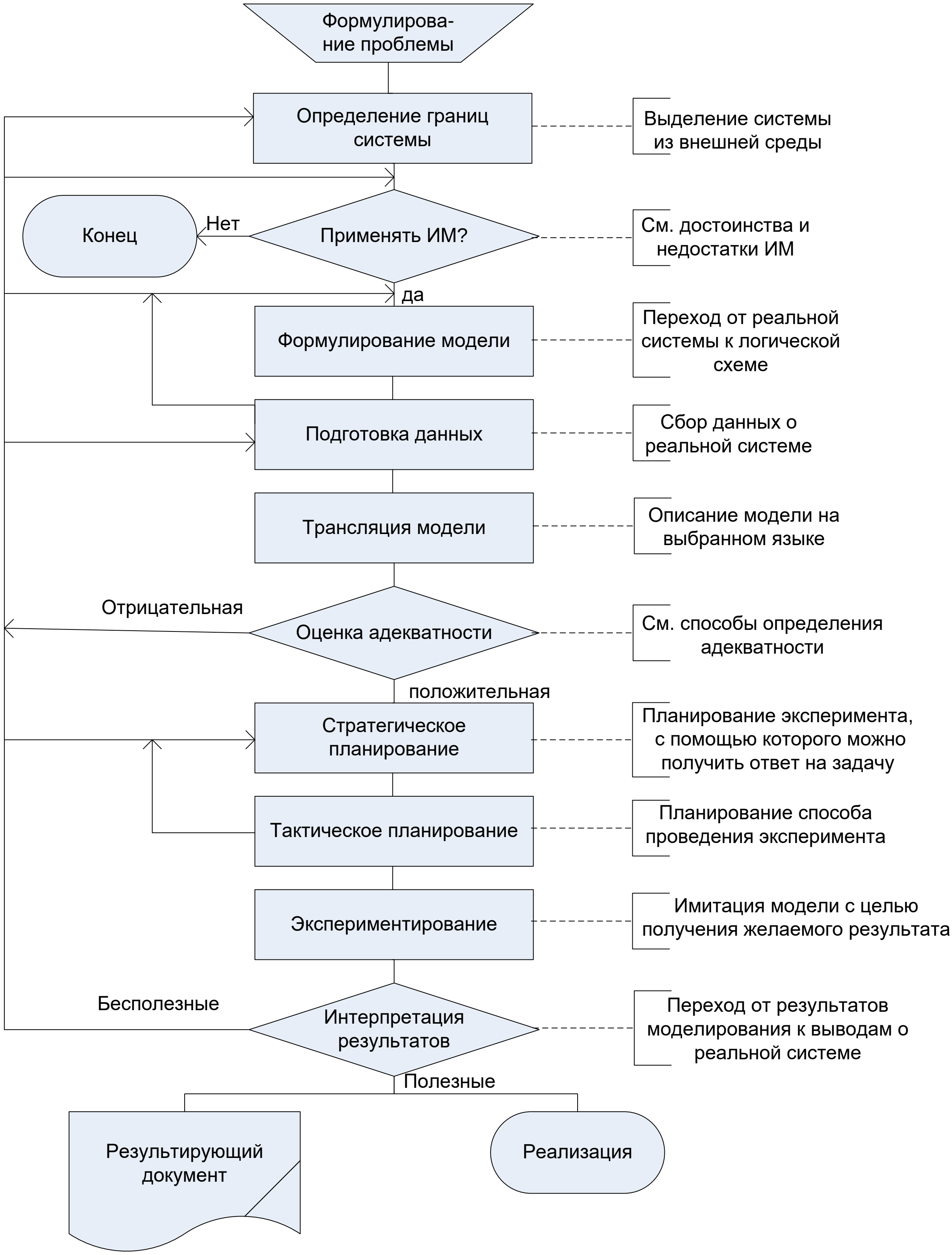


Рис. 4.1. Блок-схема процесса имитации

Определения системы – установление границ, ограничений и измерителей эффективности системы, подлежащей изучению. Наблюдается тенденция имитировать избыточное число деталей. Во избежание такого положения следует строить модель, ориентированную на решение вопросов, на которые требуется найти ответы, а не имитировать реальную систему во всех подробностях.

Формулирование модели (абстрагирование) – переход от реальной системы к некоторой логической схеме.

Подготовка данных – отбор данных, необходимых для построения модели, и представление их в соответствующей форме.

Трансляция модели – описание модели на выбранном языке моделирования.

Оценка адекватности – повышение до приемлемого уровня степени уверенности, с которой можно судить относительно корректности выводов о реальной системе, полученных на основании обращения к модели. Это важнейший этап. Проверка, выполненная без должной тщательности, может привести к катастрофическим последствиям. Способы оценки имитационной модели:

- верификация – убеждение экспериментатора, что модель ведёт себя так, как было задумано;
- проверка соответствия между поведением модели и поведением реальной системы;
- проблемный анализ – формулирование статистически значимых выводов на основе данных, полученных путём машинного моделирования.

Стратегическое планирование – планирование эксперимента, который должен дать необходимую информацию. При этом выделяют два типа задач:

1. Определение сочетания параметров, которое оптимизирует переменную отклика.

2. Объяснение соотношения между переменной отклика и контролируемыми в системе факторами.

Тактическое планирование – определение способа проведения каждой серии испытаний, предусмотренных планом эксперимента. При этом решаются следующие задачи:

1. Определение начальных условий в той мере, в какой они влияют на достижение установившегося режима.

2. Возможно большее уменьшение дисперсии решений при одновременном сокращении необходимых размеров выборки.

Экспериментирование – процесс осуществления имитации с целью получения желаемых данных и анализ чувствительности. Имитационное моделирование идеально подходит для анализа чувствительности благодаря тому, что экспериментатор здесь может успешно контролировать весь ход эксперимента.

В отличие от экспериментирования с реальными системами пользователь модели, располагая возможностями абсолютного контроля над своей моделью, может варьировать по желанию любой параметр и судить о поведении модели по наблюдаемым результатам.

Интерпретация – построение выводов по данным, полученным путём имитации.

Реализация – практическое использование модели и (или) результатов моделирования.

Документирование – регистрация хода осуществления проекта и его результатов, а также документирование процесса создания и использования модели.

В заключение отметим рекомендуемое распределение времени проектирования модели: 25 % – на постановку задачи; 20% – сбор и анализ данных; 30% – разработка модели; 25% – на реализацию.

4.2. Среда и функциональная структура языка моделирования GPSS

GPSS (General Purpose Simulation System) – общецелевая система моделирования, представляет собой язык и машинную программу [16]. Машинная программа интерпретирует модель, написанную на языке GPSS, предоставляя тем самым пользователю возможность проведения экспериментов с этой моделью на ЭВМ.

GPSS предназначена для создания моделей систем массового обслуживания. Модели на GPSS компактны, что является следствием того, что в GPSS встроено максимально возможное число логических программ, необходимых для моделирования систем.

Организация и принцип функционирования GPSS. GPSS построен в предположении, что моделью системы является описание её элементов и логических правил их взаимодействия в процессе функционирования моделируемой системы. Далее предполагается, что для моделируемых систем можно выделить небольшой набор абстрактных элементов, называемых объектами. Причём набор логических правил тоже ограничен и может быть описан небольшим числом стандартных операций.

Основой GPSS являются программы, описывающие функционирование конечного набора объектов, и специальная диспетчеризирующая программа – *симулятор*, которая выполняет следующие функции:

– обеспечение заданных маршрутов продвижения динамических объектов, называемых далее *транзактами* (сообщениями);
– планирование событий, происходящих в модели, путём регистрации времени наступления каждого события и выполнения их в нарастающей временной последовательности;
– регистрацию статистической информации о функционировании модели;
– продвижение модельного времени в процессе моделирования системы.

Объекты GPSS подразделяются на 7 категорий и 14 типов (табл. 4.3) и позволяют, с одной стороны, описать их взаимодействие сравнительно несложными наборами операций, с другой – достаточно просто и наглядно представить процесс функционирования исследуемой системы, формализованной в виде ***Q***–схемы [9].

Таблица 4.3

Категории и типы объектов GPSS

Категории GPSS	Типы объектов GPSS
1. Динамическая	1. Транзакты
2. Операционная	2. Блоки
3. Аппаратная	3. Устройства 4. Ключи 5. Памяти
4. Вычислительная	6, 7. Переменные: арифметические, булевские 8. Функции
5. Статистическая	9. Очереди 10. Таблицы
6. Запоминающая	11. Ячейки 12. Матрицы ячеек
7. Группирующая	13. Списки пользователя 14. Группы

Динамическими объектами являются транзакты (сообщения), которые представляют собой единицы исследуемых потоков и производят ряд определённых действий, продвигаясь по фиксированной структуре, представляющей собой совокупность объектов других категорий.

Операционные объекты, т.е. блоки, задают логику функционирования модели системы и определяют пути движения транзактов между объектами аппаратной категории.

Объекты аппаратной категории – это абстрактные элементы (устройства, логические ключи и памяти), на которые может быть декомпозировано оборудование реальной системы. Воздействуя на эти объекты, транзакты могут изменять их состояние и влиять на движение других транзактов.

Вычислительная категория служит для описания таких ситуаций в процессе моделирования, когда связи между компонентами моделируемой системы наиболее просто и компактно выражаются в виде математических (аналитических и логических) соотношений. Для этих целей в качестве объектов вычислительной категории введены арифметические и булевские переменные и функции.

К *статистическим объектам* относятся очереди и таблицы, вводимые для оценки характеристик поведения системы.

Основными объектами являются блоки и транзакты.

Блоки используются для описания функций моделируемой системы и управляют движением транзактов. Практически все изменения состояний модели системы возникают в результате поступления транзактов в соответствующие блоки и выполнения подпрограмм, связанных с этими блоками. После выполнения соответствующей подпрограммы транзакт либо продолжает движение, либо задерживается на некоторое время в блоке.

Транзакты представляют собой описание динамических процессов в реальных системах. Они могут описывать реальные физические объекты, например автомобили в очереди у бензозаправочной колонки. Кроме того, транзакты могут описывать и нефизические (программные) объекты, например, канальные программы, порядок выбора в коммутационных схемах и т.д. Транзакты можно «генерировать» и «уничтожать» в процессе моделирования системы. Работа GPSS модели заключается в перемещении транзактов от блоков к блокам.

В процессе моделирования системы одни объекты взаимодействуют с другими, в результате чего происходят изменения атрибутов и преобразование арифметических или логических их значений. Транзакты моделируют прохождение по системе соответствующих единиц исследуемого потока.

Каждому объекту соответствуют *атрибуты*, описывающие его состояние в данный момент времени. Значения атрибутов могут быть арифметическими или логическими. Большая часть атрибутов недоступна для программиста. Атрибуты, которые надо адресовать, называются *стандартными числовыми атрибутами* (СЧА) (табл. 4.4).

Т а б л и ц а 4.4

Стандартные числовые атрибуты

Типы объ-ектов	Символи-ческое обо-значе-ние	Диапазон из-менения	Описание
Транзакты	PF PH PB PL PR M1 MP	$\pm(2^{31}-1)$ $\pm(2^{15}-1)$ $\pm(2^{7}-1)$ $\pm(2^{24}-1)$ 0…127 $\pm(2^{31}-1)$ $\pm(2^{31}-1)$	Текущее значение параметра (формат F – «слово», H – «полуслово», B – «байт», L – «плавающая точка») транзакта, обрабатываемого в данный момент. Приоритет обрабатываемого транзакта. Транзактное время. Параметрическое транзактное время.
Блоки	N W	$2^{24}-1$ $2^{15}-1$	Счетчик входов в блок Счетчик текущего содержимого блока
Памяти	F FR FC FT S R SR SA SC ST SM	Логические 1 или 0 0…999 $2^{31}-1$ $2^{31}-1$ $2^{31}-1$ $2^{31}-1$ 0…999 $2^{31}-1$ $2^{31}-1$ $2^{31}-1$ $2^{31}-1$	Состояние устройства Коэффициент использования (в долях от 1000) Счетчик входов Среднее время/транзакт Текущее содержимое памяти Число свободных единиц памяти Коэффициент использования (в долях от 1000) Среднее содержимое памяти Счетчик входов Среднее время, транзакт Максимальное содержимое памяти
Переменные	V FV BV	$\pm(2^{31}-1)$ 10^{-78} … 10^{75} 1 или 0	Значение арифметической переменной Переменные с плавающей запятой Значение булевой переменной
Функции	FN	$\pm(2^{31}-1)$	Значение функции

Продолжение табл. 4.4

Типы объ-ектов	Символи-ческое обо-значе-ние	Диапазон из-менения	Описание
Очереди	Q QA QM QC QZ QT	$2^{31}-1$ $2^{31}-1$ $2^{31}-1$ $2^{31}-1$ $2^{31}-1$ $2^{31}-1$	Текущая длина очереди Средняя длина очереди Максимальная длина очереди Общее число элементов в очереди Число элементов, не задержанных в очереди Среднее время пребывания в транзакта в очереди

	QX	$2^{31}-1$	Среднее время пребывания в очереди без учета нулевых входов
Таблицы	TB TC TD	$\pm(2^{31}-1)$ $2^{31}-1$ $2^{31}-1$	Среднее значение аргумента таблицы Счетчик входов в таблицу Среднее квадратичное отклонение аргумента
Ячейки	XF XH XB XL	$\pm(2^{31}-1)$ $\pm(2^{15}-1)$ $\pm(2^{7}-1)$ $\pm(2^{24}-1)$	Текущее содержание ячейки (формат F – «слово», H – «полуслово», B – «байт», L – «плавающая точка»)
Матрицы ячеек	MX(a,b) MH(a,b) MB(a,b) ML(a,b)	$\pm(2^{31}-1)$ $\pm(2^{15}-1)$ $\pm(2^{7}-1)$ $\pm(2^{24}-1)$	Текущее содержание матрицы: a – строка, b – столбец. (Формат X – «слово», H – «полуслово», B – «байт», L – «плавающая точка»)
Группы	G	$2^{15}-1$	Число членов группы в текущий момент
Список пользователя	CA CH CM CC CT	$2^{15}-1$ $2^{15}-1$ $2^{15}-1$ $2^{31}-1$ $2^{31}-1$	Среднее число элементов в списке Текущее число элементов в списке Максимальное число элементов в списке Общее число элементов в списке Среднее время пребывания элементов в списке
Системные атрибуты	RN1…RN8 C1 AC1 TG	0…0,999999 0…999 $2^{31}-1$ $2^{31}-1$ $2^{31}-1$	Случайное число, используемое как аргумент функции Случайное число в других случаях Текущее значение относительного времен Текущее значение абсолютного времен Содержание счетчика завершения

Следующие СЧА не связаны только с одним отдельным объектом, а используются моделью в целом: *C1* – текущее значение относительного времени с начала процесса моделирования; K_n или n – положительная константа; RN_x – одно из восьми случайных чисел ($1 \le x \le 8$), значения которых находятся в пределах 0…999.

У каждого блока имеется два СЧА: W_n – счётчик входов в блок или ожидающий счётчик, который содержит в себе номер текущего транзакта, находящегося в блоке n, N_n – общий счётчик транзактов, поступивших в блок с начального момента моделирования или с момента обнуления. Оба счётчика меняют своё содержимое автоматически.

При написании программы в среде GPSS следует помнить об ограничениях на количество объектов в программе, которое накладывается версией языка GPSS (табл. 4.5).

Таблица 4.5

Ограничение на количество объектов

Тип объекта	Стандартное число объектов
Транзакты	1200
Блоки	1000
Устройства	300
Памяти	300
Ключи	1000
Очереди	300

В зависимости от назначения блоки подразделяются на несколько групп.

1. Блоки, осуществляющие модификацию атрибутов транзактов: а) временная задержка *ADVANCE*; б) генерация и уничтожение транзактов *GENERATE*, *TERMINATE*, *SPLIT*, *ASSEMBLE*; в) синхронизация движения нескольких транзактов *MATCH*, *GATHER*; г) изменение параметров транзактов *ASSIGN*, *INDEX*, *MARK*; д) изменение приоритета *PRIORITY*.

2. Блоки, изменяющие последовательность продвижения транзактов (блоки передачи управления): *TRANSFER*, *LOOP*, *TEST*, *GATE*.

3. Блоки, связанные с группирующей категорией: *JOIN*, *REMOVE*, *EXAMITE*, *SCAN*, *ALTER*.

4. Блоки, организующие использование объектов аппаратной категории: а) устройства (технические средства) *SEIZE*, *RELEASE*, *FAVAIL*, *PREEMPT*, *RETURN*, *FUNAVAIL*; б) памяти (запоминающие устройства) *ENTER*, *LEAVE*, *SAVAIL*, *SUNAVAIL*; в) ключи (логические переключатели) *LOGIC*.

5. Блоки, сохраняющие необходимые значения для дальнейшего использования: *SAVEVALUE*, *MSAVEVALUE*.

6. Блоки, обеспечивающие получение статистических результатов: а) очереди *QUEUE*, *DEPART*; б) статистические таблицы *TABULATE*, *TABLE*.

7. Специальные блоки *BUFFER*, *PRINT*, *EXECUTE*, *COUNT'X'*, *CHANGE*, *TRACE*, *UNTRACE*, *SELECT'X'*, *HELP*.

8. Блоки для организации цепей *LINK*, *UNLINK*.

9. Вспомогательные блоки *WRITE*, *SAVE*, *LOAD*, *REPORT*, *UPDATE*.

При моделировании в среде GPSS систем массового обслуживания совершаются события, то есть изменения, которые являются следствием движения транзактов по системе.

В GPSS все события делятся на две категории: основные и вспомогательные.

Основное событие – это такое событие, время возникновения которого можно запланировать заранее, то есть рассчитать его до фактического возникновения. К основным событиям относятся: приход заявок на обслуживание и окончание обслуживания.

Вспомогательное событие – это такое событие, время возникновения которого невозможно запланировать заранее. Они являются зависимыми, возникающими как следствие основных событий. К вспомогательным событиям относятся поступления заявок на обслуживание обслуживающими аппаратами («захват» обслуживающих аппаратов).

Список основных событий приведён в таблице 4.6, которая раскрывает логику функционирования системы моделирования GPSS. Для того чтобы вызвать событие, необходимо выполнить соответствующие логические операции обработки событий. Вторая колонка таблицы 4.6 как раз содержит список логических операций обработки двух видов основных событий.

Последовательность событий, которая происходит в модели, отслеживается с помощью таймера модельного времени, определяющего текущее время модели – модельное время. Таймер модельного времени в начале моделирования устанавливается в нулевое значение и в дальнейшем корректируется автоматически в соответствии с логикой, предписанной моделью.

Планирование основных событий производится в два этапа:

1. Розыгрыш случайного числа, соответствующего интервалу времени (прибытия или обслуживания).

2. Это значение временного интервала прибавляется к текущему значению таймера модельного времени. Сумма этих значений указывает на момент времени в будущем, когда фактически произойдёт событие.

Таблица 4.6

Список основных событий и действий, которые они вызывают

Основные события	Действия, которые они вызывают (вспомогательные события и планирование)
Приход заявки	1. Планирование следующего прихода. 2. Проверка состояния обслуживающего аппарата. Канал свободен? НЕТ: поступление заявки в очередь. ДА: поступление заявки на обслуживание; это вызывает: а) переход обслуживающего аппарата из свободного состояния в занятое; б) планирование события окончания обслуживания.

Окончание обслуживания	Проверка состояния очереди. Есть ли в очереди заявка, ожидающая обслуживания? НЕТ: переход обслуживающего аппарата из занятого состояния в свободное; ДА: поступление заявки на обслуживание; это вызывает: а) продвижение заявки в очереди; б) планирование события окончания обслуживания.

Здесь в машинную модель вносится допущение. В процессе моделирования системы модель как бы знает, когда появится следующая заявка и когда произойдёт окончание обслуживания обслуживаемой заявки. В реальной системе эти моменты неизвестны. Тем не менее, можно считать, что при длительном моделировании модель становится адекватной реальной системе в статистическом смысле.

Таймер модельного времени регистрирует только целые значения. Это означает, что события могут возникать только в «целые» моменты времени. Единицу времени, которая может быть отмечена таймером, определяет разработчик модели, которая ему удобна для того, чтобы правильно отразить события реальной системы в модели. Разработчик должен следить за тем, чтобы все данные, связанные со временем, были выражены через определённую минимальную единицу времени.

Система моделирования GPSS является интерпретатором для «следующего события». Иначе говоря, после того как модель полностью скорректирована в данный момент модельного времени, таймер продвигается к ближайшему значению времени, в которое происходит следующее событие (точнее, к следующему моменту времени, на которое запланировано возникновение события). Интервал модельного времени пропускается, если на этом интервале нет событий. Практически это означает, что время «прогона» модели не зависит от единицы времени, выбранной разработчиком модели.

Общая логическая блок-схема функционирования модели в среде моделирования GPSS приведена на рис. 4.2.

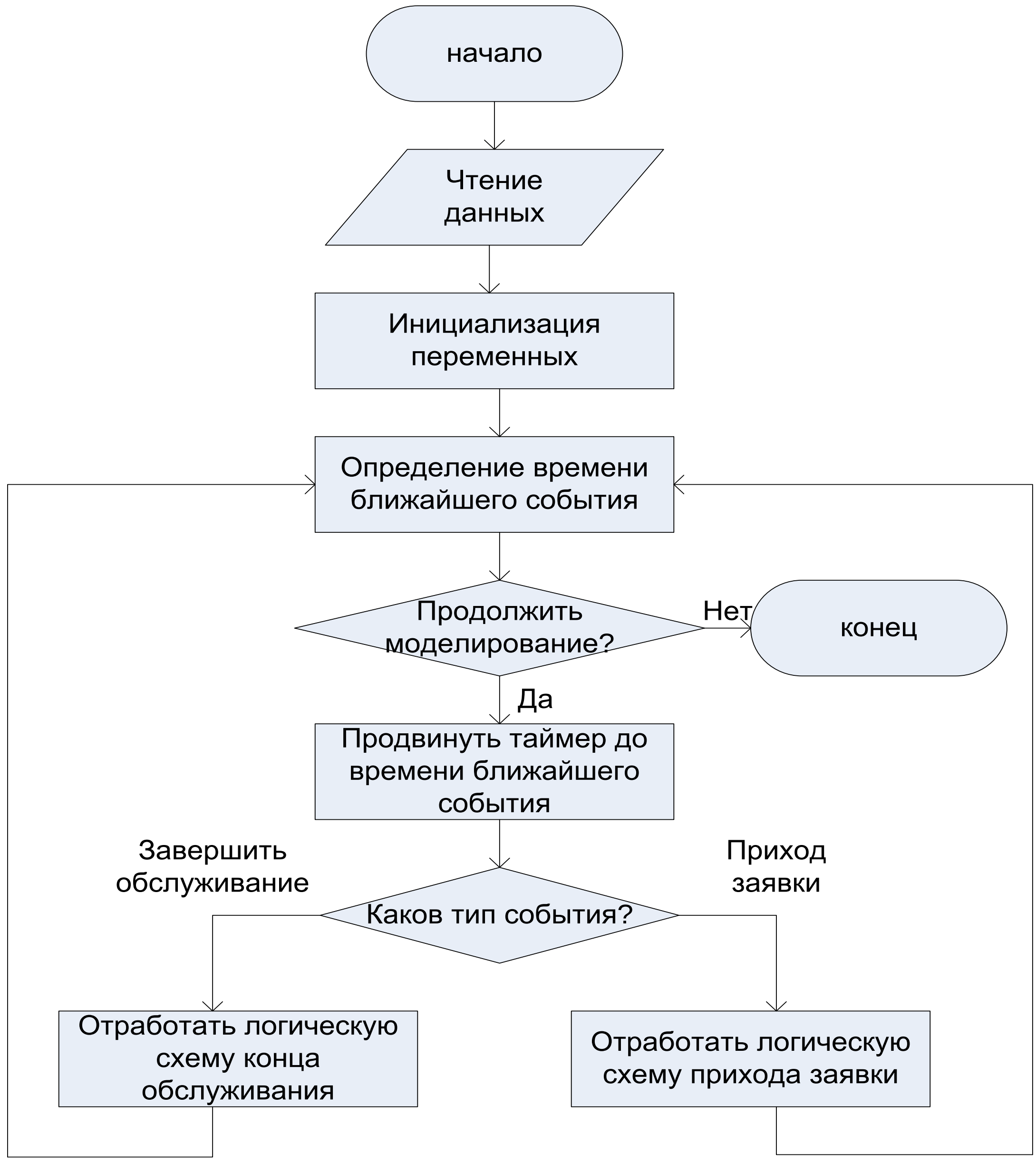


Рис. 4.2. Общая логическая блок-схема моделирования в среде GPSS

Завершение моделирования в среде GPSS происходит либо по времени, либо по поступлению заданного числа транзактов.

Особенности построения и работы моделей в среде GPSS. Модель на GPSS строится следующим образом [6, 9, 16].

1. Выбираются необходимые блоки из допустимого множества существующих в GPSS блоков.

2. Выбранные блоки выстраивают в блок-схему в соответствии с взаимодействием элементов в реальной системе для того, чтобы в процессе использования модели они (т.е. машинные эквиваленты блоков модели) как бы взаимодействовали друг с другом. Блок-схема представляет собой набор фигур с ха-

рактерным очертанием блоков, соединённых между собой линиями. Каждый блок имеет свой графический аналог, с помощью которого отображается пространственная конструкция модели.

3. После того как модель создана, взаимодействие между блоками модели аналогично взаимодействию элементов моделируемой реальной системы.

4. После разработки блок-схемы модели создаётся текст программы. Эта процедура является, как правило, чисто механической.

Выстроенные в блок-схему блоки означают продвижение по ней транзактов. Конфигурация блок-схемы GPSS модели отражает направления, по которым происходит движение транзактов. В модели существует большое число транзактов, однако в один момент времени двигается только один транзакт. Каждое успешное вхождение транзакта в блок вызывает обращение к соответствующей подпрограмме. Следовательно, выполнение модели заключается в последовательном обращении к подпрограммам, являющемся следствием входа в определённые блоки перемещающихся транзактов.

В общем случае модель на GPSS состоит из нескольких независимых сегментов. В процессе моделирования активным является тот из сегментов, в котором находится перемещающийся в настоящий момент транзакт. Когда он блокируется, начинает двигаться следующий транзакт, и может так случиться, что этот следующий транзакт принадлежит другому сегменту модели. Таким образом происходит переключение активности между сегментами.

При описании схем алгоритмов посредством языковых средств GPSS используются следующие типы строк: 1) строки описания блоков; 2) строки описания объектов; 3) управляющие строки; 4) строки комментарии (наличие символа «*» в 1-й позиции, указывает на то, что строка содержит только комментарии). Каждый блок блок-схемы GPSS модели определяется с помощью отдельной строки, имеющей следующий формат:

2 *Поле метки* 6	8 *Поле операции* 18	19 *Поле операндов* 31
Символический адрес блока	*Наименование типа блока*	*Поле операндов A,B,C,D,E,F,G*

Запись начинается с 1-й позиции, а поля разделяются пробелами, Отсутствие метки также обозначается пробелом.

Все блоки GPSS определяются наименованием типа блока и операндами *A, B, C, D, E, F, G*, задающими выполняемые функции. Кроме того, каждый блок может иметь символический адрес, записываемый в поле метки блока.

Поле метки расположено со 2 по 6 позиции экрана, поле операции (наименование блока) с 8 по 18, поле операндов с 19 позиции и до первого пробела в тексте. Комментарии, как правило, размещаются после 31 позиции.

Тип блока указывает функциональное назначение этого блока, а операнды задают информацию для его действия. Если некоторые операнды не используются, это отмечается запятой. В качестве аргументов в поле операндов используются: а) некоторые СЧА; б) положительные константы K_n; в) специальные обозначения, определяющие режим; г) метки, имена.

Для задания дополнительной информации симулятору GPSS требуются операторы контроля управления.

Оператор SIMULATE обеспечивает “прогон” имитационных моделей. Допускаются две формы оператора: *SIMULATE* и *SIMULATE m,* где *m* – число минут машинного времени, по истечении которого моделирование прекращается.

Оператор START показывает симулятору GPSS, что все входные данные получены и можно начинать счёт.

Поле *A* определяет число транзактов, которые должны пройти через систему до выдачи окончательной статистики, называемой счётчиком числа завершений и обозначаемой *TG1*. Счёт заканчивается в тот момент, когда $TG1 < 0$. Значение *TG1* уменьшается на величину, заданную в поле *A* блока *TERMINATE*, при каждом входе сообщения в этот блок.

Поле *B* может быть использовано для блокирования выдачи статистики в конце счёта. Для этого нужно записать в поле *B* мнемоническое обозначение *NP*. При наличии такой записи в поле *B* сбор статистической информации в процессе счёта не прекращается, блокируется только печать собранной информации по окончании счёта. Если поле *B* пусто, печать происходит как обычно.

Поле *C* можно использовать для задания интервала промежуточных выдач статистик. Счётчик интервала уменьшается на то же число единиц, что и счётчик завершения. Когда значение этого счётчика становится равным нулю или отрицательным, то производится выдача статистики в обычном формате и восстановление счётчика, после чего счёт продолжается.

Поле *D* может содержать указание на то, что при выдаче информации (как в конце выдачи промежуточных результатов, так и в конце счёта) нужно распечатывать списки текущих событий, будущих событий, список прерываний, а также списки пользователя. Значение «1» в поле *D* указывает на то, что пользователю нужна печать списков. В противном случае поле оставляют незаполненным.

Пользователю может понадобиться провести серию просчётов, т.е. несколько просчётов один за другим. Для облегчения задания такого режима введены три управляющих оператора, в различной степени возвращающих модель к исходному состоянию: *RESET*, стирающий всю накопленную статистику; *CLEAR*, который стирает всю статистику и все транзакты; *JOP*, стирающий всю статистику, все транзакты и все блоки.

Оператор RESET. Результатом действия этого оператора является то, что стирается вся статистическая информация, накопленная в процессе моделирования. Счётчики числа входов в блоки сбрасываются до «0», но содержимое счётчиков ожидания не изменяется. Коэффициенты использования устройств сбрасываются до «0», счётчики числа входов в устройство устанавливаются в соответствие с текущим состоянием устройства. Счётчик равен «0», если устройство свободно, «1» – если оно занято или обслуживание сообщения прервано, и «2» – если устройство занято и обслуживает прерывание. Коэффициенты использования памятей также сбрасываются до «0», а счётчики числа входов в память устанавливаются в соответствие с текущим содержимым каждой памяти. Коэффициенты использования очередей сбрасываются до «0», и счётчики числа входов устанавливаются равными текущей длине соответствующих очередей. Счётчики всех нулевых задержек сбрасываются до «0», а максимальные значения длины очереди устанавливаются равными текущей длине очереди. Все счётчики числа наблюдений интервалов таблиц, все средние значения и т.д. также сбрасываются до «0». Содержимое ячеек *SAVEVALUE* и состояние логических переключателей не изменяются. Датчик случайных чисел не возвращается к исходному числу. Пользователь может также задать объекты (устройства, памяти, очереди и др.), атрибуты которых останутся без изменения после выполнения операций, соответствующих оператору *RESET*.

Оператор CLEAR. В результате действия этого оператора стирается вся накопленная в процессе моделирования статистика и из системы удаляются все транзакты. Счётчики числа входов и счётчики ожидания всех блоков сбрасываются до «0». Коэффициенты использования всех устройств и счётчики числа входов сбрасываются до «0» и все устройства переводятся в состояние «не занято». Коэффициенты использования памятей, счётчики числа входов и текущее содержимое сбрасываются до «0» для всех памятей. Коэффициенты использования очередей, счётчики числа входов, счётчики числа нулевых задержек, текущая и максимальная длина очереди сбрасываются до «0». Стирается вся информация в таблицах и ячейках *SAVEVALUE*. Все логические переключатели устанавливаются в «0», из системы удаляются все транзакты, а абсолютное и относительное время устанавливается в «0». Затем запускаются все блоки *GENERATE*. Начальные интервалы задержки, записанные в поле *C* блока *GENERATE*, отсчитываются после завершения операций оператора *CLEAR*. Счётчикам пределов, заданным в поле *D*, присваиваются их исходные значения. Работа этих счётчиков также начинается после выполнения операций оператора *CLEAR*. Операции оператора *CLEAR* производятся в момент его появления, причём действие оператора *CLEAR* на следующего за ним оператора не распространяется. Перед первым оператором *START* модели оператор *CLEAR* не нужен. Датчик случайных чисел не возвращается к исходному числу.

Оператор JOP. Этот оператор помещается между последовательными рабочими вычислениями, являющимися частями одной работы. Он выполняет все функции оператора *CLEAR*, а также стирает все описания блоков, функций, переменных, таблиц и памятей. Перед первой работой оператор *JOP* не нужен. Оператор *JOP* возвращает датчик случайных чисел к исходному числу.

Оператор END. Этот оператор указывает на конец модели.

В языке GPSS используется символическая и косвенная адресация. Любым объектам программы (блокам, устройствам, памятям) можно дать символические имена. Нумерацию блоков производит симулятор. Разным объектам можно дать одно и то же имя.

Возможности языка расширяет косвенная адресация и косвенное описание, в которых используются индексы, применяемые в качестве номера *n* аргумента блока и в качестве индекса *n* СЧА, указываемого в некоторой из строк описания. Для записи индекса используется знак «$» и число после него. Индекс определяется значением параметра, номер которого записан после $.

Косвенная адресация и косвенное описание могут быть использованы совместно и порознь практически для любых аргументов различных типов блоков.

4.3. Система программного обеспечения GPSS/PC

Система программного обеспечения GPSS/PC предназначена для имитационного моделирования систем массового обслуживания. В системе GPSS/PC языком моделирования является язык GPSS.

Наиболее распространенным методом описания систем является составление блок-диаграмм. Блок-диаграмма – это графическое представление операций, происходящих внутри системы. Другими словами, блок-диаграмма описывает взаимодействие событий внутри системы. Линии, соединяющие блоки, указывают маршруты потоков сообщений или описывают последовательность выполняемых событий. В случае нескольких вариантов действий от блока отходят несколько линий. Если же к блоку подходят несколько линий, то это означает, что выполняемая операция является общей для двух или более последовательностей блоков. Выбор логических путей может основываться на статистических или логических условиях, действующих в момент выбора.

В GPSS/PC имеется определенное количество типов блоков для задания объектов и операций над ними. Каждому блоку соответствует графическое изображение на блок-диаграмме. Стрелки между блоками указывают маршруты потоков сообщений.

Далее, чтобы применить язык моделирования GPSS, каждый блок блок-диаграммы заменяется соответствующим оператором GPSS/PC.

4.3.1. Объекты GPSS/PC

Объекты в моделируемой системе предназначены для различных целей. Выбор объектов в конкретной моделируемой системе зависит от характеристик модели и, в некоторых случаях, от специалиста, составляющего модель. Совершенно не обязательно, чтобы в одной модели участвовали все типы объектов. Обязательным является лишь то, что в каждой модели должны быть блоки и сообщения, иначе просчитать ее будет невозможно. Какие объекты, помимо сообщений и блоков, будут включены в модель, будет зависеть от того, какие именно используются блоки и операнды блоков (которые могут повлечь появление операторов описания других объектов).

4.3.1.1. Сообщения

Сообщения (транзакты) – это динамические объекты GPSS/PC. Они создаются в определенных точках модели, продвигаются интерпретатором через блоки, а затем уничтожаются. Сообщения являются аналогами единиц-потоков в реальной системе. Они могут представлять собой различные элементы даже в одной системе. Практически все изменения состояния моделей происходят в результате входа сообщений в блоки и последующего выполнения подпрограмм системы моделирования, связанных с соответствующими типами блоков.

Каждое продвижение считается событием, которое должно происходить в конкретный момент времени. Интерпретатор GPSS/PC автоматически определяет моменты наступления событий. В тех случаях, когда событие не может произойти, хотя момент его наступил (например, при попытке занять устройство, когда оно уже занято), сообщение прекращает продвижение до снятия блокирующего условия.

Сообщения нумеруются последовательно, начиная с номера 1. Параметры сообщений принимают значения из множества целых чисел. Каждое сообщение имеет один или более параметров. Параметры нумеруются. Сообщение может обращаться только к своим параметрам. Номера параметров используются для ссылок на значения, присвоенные параметрам.

Сообщениям может присваиваться приоритет, используемый системой в ходе моделирования.

4.3.1.2. Блоки

После того как система описана, исходя из операций, которые она выполняет, ее нужно описать на языке GPSS/PC, используя блоки, которые выполняют соответствующие операции в модели.

В блоках могут происходить события четырёх основных типов:

1) создание или уничтожение сообщений;
2) изменение числового атрибута объекта;
3) задержка сообщения на определенный период времени;
4) изменение маршрута сообщения в модели.

4.3.1.3. Одноканальные устройства

Аналогами обслуживающих приборов и реальных систем в GPSS/PC являются объекты типа "оборудование". К объектам этого типа относятся одноканальные устройства и многоканальные устройства. Одноканальные устройства (в дальнейшем их будем называть устройствами) представляют собой оборудование, которое в любой момент времени может быть занято только одним сообщением. Интерпретатор записывает информацию о том, какое сообщение в настоящий момент занимает устройство. Если другое сообщение попытается захватить устройство, то это сообщение задерживается до тех пор, пока устройство не освободится. Программа автоматически подсчитывает общее время занятости устройства. Это значение позволяет определить коэффициент использования устройства. Подсчитывается также общее число сообщений, занимавших устройство, что позволяет вычислить среднее время занятости устройства одним сообщением.

4.3.1.4. Многоканальные устройства

В GPSS/PC многоканальные устройства представляют объекты типа "оборудование" для параллельной обработки, они могут быть использованы несколькими сообщениями одновременно. Многоканальные устройства используются для представления физического оборудования, например, зрительного зала театра, стоянки автомобилей и, в некоторых случаях, основной памяти в системах для обработки данных. Пользователь определяет емкость каждого многоканального устройства, используемого в модели, а интерпретатор ведет учет числа единиц многоканальных устройств, занятых в каждый момент времени. Если сообщение пытается занять больше единиц многоканального устройства, чем свободно в данный момент, обработка этого сообщения задерживается до того момента, пока в многоканальном устройстве освободится достаточный объем.

Программа автоматически ведет подсчет числа сообщений, входящих в многоканальное устройство. Определяется также среднее число единиц многоканальных устройств, занятых одним сообщением, и среднее время пребывания сообщения в многоканальном устройстве. Эти статистические данные выдаются в конце счета и позволяют определить, насколько эффективно используются в системе объекты параллельной обработки и достаточна ли их емкость.

4.3.1.5. Очереди

В любой системе движение потока сообщений может быть задержано из-за недоступности оборудования. Например, требуемые устройства могут быть уже заняты или многоканальные устройства, в которые нужно войти, уже заполнены. В этом случае задержанные сообщения ставятся в "очередь" – еще один тип объектов GPSS/PC. Учет этих очередей составляет одну из основных функций интерпретатора GPSS/PC.

Пользователь может определить специальные точки в модели, в которых нужно собирать статистику об очередях. Тогда интерпретатор GPSS/PC автоматически будет собирать статистику об очередях (длину очереди, среднее время пребывания в очереди и т.д.). Число задержанных сообщений и продолжительность этих задержек определяется только в этих заданных точках. Интерпретатор также автоматически подсчитывает в этих точках общее число сообщений, поступающих в очередь. Интерпретатор подсчитывает среднее время пребывания сообщения в очереди (для каждой очереди), а также максимальное число сообщений в очереди.

4.3.1.6. Модельное время

Чтобы обеспечить правильную временную последовательность событий в модели, организованы часы, хранящие значения текущего момента в модели. Все отрезки времени моделируемой системы измеряются целыми значениями. В отли-

чие от обычных часов, измеряющих время в определенных единицах, обычно в секундах, часы в GPSS/PC меняют свое значение только для того, чтобы указать время наступления ближайшего события. Например, если текущее значение часов модели равно 2, а очередное событие должно наступить в момент времени 7, то значение часов увеличивается сразу на 5 единиц. Отметим, что единицы времени в модели не обязательно должны быть конкретными единицами времени, такими как секунда или час. Основной единицей времени в модели можно выбрать любую единицу, которая позволит получить необходимую точность моделирования. Важно помнить, единицы времени выбираются исходя из требований пользователя к точности моделирования.

4.3.1.7. Статистика GPSS/PC

Поскольку целью построения любой модели является исследование моделируемой системы, интерпретатор GPSS/PC автоматически собирает стандартную статистику по каждому типу объектов, занятых в модели.

Стандартная статистика используется и при отладке модели, и при оценке самой системы.

4.3.2. Кодирование операторов GPSS/PC

4.3.2.1. Стандартные числовые атрибуты

В процессе моделирования интерпретатор GPSS/PC автоматически регистрирует и корректирует информацию, касающуюся различных элементов, используемых в модели. Большая часть информации доступна только интерпретатору. Однако к некоторым атрибутам объектов может обращаться и программист, манипулируя их значениями согласно логике модели. Такие атрибуты называются стандартными числовыми атрибутами (СЧА). Каждый объект GPSS/PC имеет свой набор СЧА. Кроме СЧА объектов, существуют еще системные числовые атрибуты, к которым пользователь может обращаться в модели, но не может изменять их значение. Имя СЧА состоит из двух частей. Первая часть указывает групповое имя, идентифицирующее тип объекта и тип информации об объекте. Вторая часть идентифицирует конкретного члена группы.

Групповое имя состоит из одной-двух букв, фиксированных для информации об объектах определенного типа, например, Q – ссылка на текущее значение длины очереди, QA – целая часть среднего значения очереди и т.д. Объекты GPSS/PC могут быть идентифицированы с помощью числовых или символьных имен. Если объект идентифицирован с помощью номера, то ссылка на его стандартный числовой атрибут записывается как СЧАj, где j –

номер объекта (целое число). При символической идентификации объекта ссылка на его стандартный атрибут записывается как СЧА$<имя>, где <имя> – символьное имя объекта.

К системным числовым атрибутам относятся следующие величины:

RNj – число, вычисляемое j датчиком случайных чисел (где 1<=j<=7). Все датчики генерируют последовательность равномерно распределенных случайных чисел. Это целое число изменяется от 0 до 999 включительно;

C1 – текущее значение условного времени. Автоматически изменяется программой и устанавливается в 0 управляющими операторами CLEAR или RESET;

AC1 – текущее значение абсолютного времени. Автоматически изменяется программой. Эта величина не меняется под действием управляющего оператора RESET и устанавливается в 0 лишь под действием оператора CLEAR;

TG1 – число, равное текущему значению счетчика завершений. Сообщения, вошедшие в блоки TERMINATE с ненулевым операндом A, уменьшают значение этого счетчика на число, равное значению операнда A;

XN1 – возвращает номер активного сообщения;

Z1 – возвращает размер свободной оперативной памяти в байтах;

M1 – время пребывания в модели сообщения, обрабатываемого программой в данный момент. Эта величина может изменяться блоком MARK. Это время вычисляется следующим образом:

```
     |¯               ¯|   |¯               ¯|   |¯                ¯|
     |  Время прохож-  |   |  Текущее значе- |   |  Отметка времени |
M1 = |  дения   сооб-  | = |  чение абсо-    | - |  обрабатываемого |
     |  щением модели  |   |  лютного услов- |   |  в данный момент |
     |_               _|   |  ного времени   |   |  сообщения       |
                           |_               _|   |_                _|
```

PR – приоритет обрабатываемого в данный момент сообщения. Каждое сообщение GPSS/PC имеет уровень приоритета, присваиваемый ему в момент создания. Значение приоритета (0-127) присваивается в блоке GENERATE или SPLIT. По умолчанию приоритет равен 0. При обращении к стандартному числовому атрибуту PR получают значение приоритета сообщения, обрабатываемого в данный момент.

Сообщения имеют следующие СЧА:

Pj – значение параметра j текущего сообщения;

MBj – флаг синхронизации: 1, если сообщение в блоке j принадлежит тому же семейству, что и текущее сообщение, 0 – в противном случае;

MPj (j<>1) – значение времени, равное разности абсолютного модельного времени и содержимого j-го параметра текущего сообщения;

Запись времени прохождения в параметре сообщения – MPj (j<>1): в поле А блока MARK может быть задан номер j параметра сообщения. При прохождении сообщения через такой блок MARK, текущее значение абсолютного условного времени записывается в j-м параметре сообщения. Впоследствии, при обращении, значение стандартного числового атрибута MPj (j<>1) вычисляется следующим образом:

```
        |-            -|   |-            -|     |-            -|
        |              |   |              |     |              |
        |Время прохож- |   |Текущее значе-|     |Значение j-го |
 MPj =  |дения, запи-  | = |чение абсо-   |  -  |параметра об- |
 j<>1   |санное в па-  |   |лютного услов-|     |рабатываемого |
        |раметре       |   |ного времени  |     |в настоящий   |
        |_            _|   |_            _|     |момент        |
                                                |сообщения     |
                                                |_            _|
```

Блоки имеют следующие СЧА:

Nj – общее число сообщений, которое должно войти в j-й блок. Подсчёт ведется программой автоматически. Например, N$MET1 – счетчик числа входов в блок MET1. Этот счетчик изменяется при каждом входе сообщения в блок MET1;

Wj – текущее число сообщений, которое находится в блоке j. Значение этого счетчика подсчитывается автоматически. Например, W$MET2 – счетчик текущего числа сообщений в блоке MET2.

Стандартные числовые атрибуты могут использоваться в качестве операндов практически в любом типе блоков. Также значение любого СЧА может входить в большинство операторов описания объектов. Особенностью СЧА является то, что они обеспечивают пользователю доступ к характеристикам состояния системы в процессе моделирования.

4.3.2.2. Типы операторов GPSS/PC

После того как блок-диаграмма составлена, она должна быть записана в форме, удобной для ввода в ЭВМ. Для этого информация об объектах GPSS/PC записывается в виде операторов в определенном формате. Операторы GPSS/PC делятся на следующие типы:

- операторы описания блоков;
- операторы описания данных и контроля управления;
- команды GPSS/PC.

4.3.2.3. Кодирование операторов GPSS/PC

При использовании GPSS/PC для ссылок на числа, блоки или объекты применяются имена. Имя представляет собой алфавитно-цифровую последователь-

ность, длиной до 20 символов, начинающуюся с буквы. Допустимо применение символов только латинского алфавита. В состав имени могут включаться также символы подчеркивания "_". Именами не могут быть коды операторов GPSS/PC, ключевые слова или коды стандартных числовых атрибутов. При трансляции GPSS/PC присваивает именам уникальные номера, начиная со стартового номера 10000.

Для принудительного присваивания именам нужных номеров, необходимо перед использованием имен с помощью оператора EQU присвоить именам соответствующие номера.

4.3.3. Блоки, ориентированные на сообщения

Практически все изменения состояния в моделях GPSS/PC происходят в результате входа сообщений в блоки и выполнения подпрограмм, связанных с этими блоками.

4.3.3.1. Блок GENERATE

Блок GENERATE является источником потока сообщений в модели. В данном блоке производится подготовка сообщений и запуск их в модель через интервалы времени, заданные пользователем. Кроме задания правильной временной последовательности, пользователь может в блоке GENERATE задать некоторую информацию об атрибутах сообщений.

Блок GENERATE имеет следующий формат записи:

GENERATE [<A>],[<B>],[<C>],[<D>],[<E>]

В поле А указывается время, которое определяет интервал между моментами генерации сообщений блоком GENERATE. Операнд А может быть именем, положительным целым числом или непосредственно СЧА. Нельзя использовать в качестве операнда параметры сообщения.

В поле В задается модификатор, который изменяет значения интервала генерации сообщений по сравнению с интервалом, указанным в поле А. Операнд В может быть именем, положительным целым числом или непосредственно СЧА. Нельзя использовать в качестве операнда параметры сообщения.

При вычислении разницы значений, заданных в полях А и В, получается нижняя граница интервала, а при вычислении суммы - верхняя граница. После генерации очередного сообщения выбирается число из полученного интервала, и это будет значение времени, через которое следующее сообщение выйдет из блока GENERATE.

Пример: GENERATE 25,10

В этом случае генерация сообщений производится по равномерному закону из интервала – (15,35).

Следует обратить внимание на то, что сообщения генерируются с заданными интервалами только в том случае, если у блоков, следующих за блоком GENERATE (например: GATE, TEST, SEIZE или ENTER), не выставлены блокирующие условия. Каждое последующее сообщение формируется только тогда, когда сообщение из блока GENERATE входит в следующий блок.

Из-за возможных воздействий на модель при изменении заданного интервала генерации сообщений нежелательно, чтобы после блока GENERATE следовал блок, создающий блокирующее условие.

В поле C задается начальная задержка. Начальная задержка относится к моменту формирования первого сообщения в блоке GENERATE как при первом просчете модели, так и после выполнения операции CLEAR. Начальная задержка – это момент времени, в который первое сгенерированное сообщение должно выйти из блока GENERATE; поля A и B на задержку сообщения влияния не имеют. Начальная задержка может быть меньше, равна или больше среднего времени, заданного в поле A. Операнд C может быть именем, положительным целым числом или непосредственно СЧА. Нельзя использовать в качестве операнда параметры сообщения.

В поле D задается предел генерации. Эта величина представляет собой максимальное число сообщений, которое будет создано в блоке GENERATE. Операнд D может быть именем, положительным целым числом или непосредственно СЧА. Нельзя использовать в качестве операнда параметры сообщения. Если поле D пусто, блок генерирует неограниченное число сообщений. Предел генерации инициализируется повторно операцией CLEAR.

Поле E определяет приоритет сообщений. Операнд E может быть именем, положительным целым числом или непосредственно СЧА. Нельзя использовать в качестве операнда параметры сообщения. Если поле E не задано, приоритет по умолчанию равен 0.

При повторном описании блока GENERATE при помощи нового оператора описания блока интерпретатор GPSS просматривает все находящиеся в данный момент модели сообщения и проверяет, есть ли среди них сообщения, связанные с повторно описываемым блоком GENERATE (таких сообщений может и не быть, если данный блок уже создал заданное число сообщений). Эти сообщения, если они есть, уничтожаются. Операнды нового блока GENERATE заменяют операнды предыдущего блока GENERATE, и затем создается новое сообщение, используя спецификации нового блока.

При использовании блока GENERATE необходимо помнить, что сообщение не должно входить в блок GENERATE. Если сообщение пытается это сделать, возникает ошибка выполнения.

Если в операторе описания блока GENERATE в поле D было задано максимальное число генерируемых блоком сообщений, и заданное число сообщений уже вышло из этого блока, т.е. блок уже закончил работу, то в процессе моделирования этот блок может быть снова запущен только в одном из двух случаев:

1) выполнены операции, заданные оператором CLEAR, и производится повторный запуск всех блоков GENERATE;

2) блок GENERATE описан повторно.

4.3.3.2. Блок TERMINATE

Блок TERMINATE имеет следующий формат записи:

TERMINATE [<A>]

Блок TERMINATE удаляет из модели входящие сообщения.

В поле A задается число единиц, на которое этот блок изменяет содержимое счетчика завершений, определяющего момент окончания моделирования. Операнд A может быть именем, положительным целым числом или СЧА. По умолчанию значение, определяемое полем A, равно 0. Если поле A пусто, то сообщение уничтожается, а содержимое счетчика не изменяется.

Когда пользователь подготавливает модель, он задает время счета, указывая в операторе START значение счетчика завершений. Поскольку различные пути сообщений в модели имеют различные смысловые значения, каждый блок TERMINATE может либо уменьшать, либо не уменьшать содержимое счетчика завершений. Если содержимое счетчика уменьшилось до 0, счет завершается.

Стандартным числовым атрибутом, связанным с описываемым оператором, является TG1 – число, равное текущему значению счетчика завершений. TG1 возвращает содержимое счетчика завершений, которое уменьшается блоком TERMINATE при заданном операнде A. Эта величина первоначально задается оператором START и указывает на завершение моделирования, когда становится равной 0.

4.3.3.3. Блок ADVANCE

Блок ADVANCE имеет следующий формат записи:

ADVANCE <A>,[<B>]

Блок ADVANCE задерживает продвижение сообщения на заданный период времени. В поле A задается среднее время пребывания сообщения в блоке ADVANCE. Содержимое поля A может быть именем, любым целым числом, в том числе и 0 или СЧА.

В поле В указывается способ модификации среднего значения, заданного в поле А. Операнд В может быть именем, положительным целым числом или СЧА.

4.3.3.4. Блок ASSIGN

Блок ASSIGN имеет следующий формат записи:

ASSIGN <A>,<B>,[<C>]

Блок ASSIGN заменяет, увеличивает или уменьшает текущее значение параметра сообщения на заданное значение.

В поле А задается номер параметра, которому присваивается значение. Операнд А может быть именем, положительным целым числом, СЧА и следующими за ними знаками +, – . Если значение параметра нужно увеличить или уменьшить, то справа в поле А ставится знак сложения или вычитания.

Поле В определяет значение, которое следует добавить или вычесть из значения параметра, заданного аргументом поля А, или заменить его. Если такой параметр не существует, то он создается. Операнд В может быть именем, любым целым числом или СЧА.

Пример: ASSIGN 2000+,–3

В этом примере значение поля В, равное –3, добавляется к значению параметра с номером 2000, который задан операндом А. Если такой параметр в сообщении отсутствует, то он создается со значением, равным 0, до того, как будет произведено добавление. Тогда значение параметра сообщения становится равным –3.

4.3.3.5. Блок MARK

Блок MARK имеет следующий формат записи:

MARK [<A>]

Блок MARK либо заменяет значение отметки времени сообщения на текущее значение абсолютного условного времени (операнд А не определен), либо записывает значение условного времени в заданный параметр сообщения (при использовании операнда А).

Поле А содержит номер параметра, в который записывается значение абсолютного условного времени. Если такого параметра не существует, то он создается. Операнд А может быть именем, положительным целым числом или СЧА.

Исходное значение времени создания сообщения может быть заменено на текущее значение абсолютного условного времени при прохождении сообщения через блок MARK. Каждое сообщение имеет следующие два стандартных числовых атрибута, связанных с временем прохождения участков модели данным сообщением:

1) M1 - время прохождения сообщением модели;

2) MPj (j<>1) - промежуточное время прохождение сообщением участка модели.

4.3.4. Блоки, изменяющие порядок прохождения блоков сообщениями

Обычно интерпретатор пытается продвинуть сообщение к следующему по номеру блоку. Однако существуют блоки, которые позволяют изменять номер следующего блока.

4.3.4.1. Блок TRANSFER

Блок TRANSFER имеет следующий формат:

TRANSFER [<A>],[<B>],[<C>],[<D>]

Блок TRANSFER является основным средством, позволяющим направить сообщение к любому блоку модели.

Поле A задает режим выбора следующего блока, к которому должно перейти сообщение.

Существуют следующие режимы работы блока TRANSFER:

- безусловный (пробел);
- статистический (.);
- BOTH.

Кроме того, операнд A может быть дробным числом, именем, положительным целым числом или СЧА.

Поля B и C задают возможные значения номеров следующих блоков или их положение. Использование значений описано при рассмотрении определенных режимов выбора. Операнды могут быть именем, положительным целым числом или СЧА. Если поле B пусто, ассемблер записывает в нем номер блока, следующего за блоком TRANSFER.

Безусловный режим выбора. Если операнд A пропущен, то блок TRANSFER работает в безусловном режиме. Входящее в блок TRANSFER сообщение переходит к блоку, указанному в поле B. Если сообщение в этот блок войти не может, попытка направить сообщение к какому-либо другому блоку не производится.

Статистический режим выбора. Когда операнд A не является зарезервированным словом, блок TRANSFER работает в статистическом режиме выбора.

Значение аргумента, записанного после точки (.) в поле A, рассматривается как трехзначное число, показывающее (в частях от тысячи), какой процент входящих в блок сообщений следует направить к блоку, указанному в поле C. Остальные *сообщения направляются к блоку, указанному в поле B, или к следующему по номеру блоку, если операнд B пропущен. Для каждого сообщения выбирается один из двух возможных вариантов; после того как выбор сделан, второй вариант для этого сообщения не рассматривается.*

Пример: BCD TRANSFER .709,BLK1,BLK2

Из общего числа сообщений, входящих в блок BCD, в среднем 0,709 будут пытаться войти в блок BLK2. Остальные 0,291 будут пытаться войти в блок BLK1.

Режим BOTH. Если в поле A стоит зарезервированное слово BOTH, блок TRANSFER работает в режиме BOTH.

В этом режиме каждое входящее сообщение сначала пытается перейти к блоку, указанному в поле B. Если это сделать не удается, сообщение пытается перейти к блоку, указанному в поле C. Если сообщение не сможет перейти ни к тому, ни к другому блоку, оно остается в блоке TRANSFER и будет повторять в том порядке попытки перехода при каждом просмотре списка текущих событий до тех пор, пока не сможет выйти из блока TRANSFER.

4.3.4.2. Блок TEST

Блок TEST имеет следующий формат:

TEST <X> <A>,<B>,[<C>]

Блок TEST определяет номер следующего блока для вошедшего в него сообщения в зависимости от того, выполняется требуемое условие или нет. Блок управляет потоком сообщений, проверяя выполнение алгебраических отношений между значениями СЧА, заданных в полях A и B.

Операнды A и B – сравниваемые величины, которые могут быть именем, любым целым числом или СЧА.

Если в полях A и B заданы СЧА, то они должны быть записаны как СЧАj (например : очередь Q1). Необходимо присвоить им номера в начале программы с помощью оператора EQU.

Во вспомогательном поле операции оператора описания блока TEST <X> – записывается один из шести условных операторов:

- 'L' – меньше. Отношение истинное, если значение аргумента поля A меньше значения аргумента поля B;

- 'LE' – меньше или равно. Отношение истинное, если значение аргумента поля A меньше или равно значению аргумента поля B;

- 'E' – равно. Отношение истинное, если значения обоих аргументов равны;

- 'NE' – не равно. Отношение истинное, если значения аргументов полей А и В не равны;

- 'G' – больше. Отношение истинное, если значение аргумента поля А больше значения аргумента поля В;

- 'GE' – больше или равно. Отношение истинное, если значение аргумента поля А больше или равно значению аргумента поля В.

Если отношение СЧА, заданных в полях А и В, истинно, сообщение переходит к следующему блоку. Если отношение ложно, сообщение переходит к блоку, номер которого задан полем С.

С – номер блока для входящего сообщения, если отношение величин, заданных в полях А и В, ложно. Операнд С может быть именем, положительным целым числом или СЧА.

Блок TEST может работать в двух режимах:

1) в режиме безусловного входа. Если в поле С задан номер следующего блока, сообщения никогда не задерживаются на входе блока TEST. Если заданное в блоке TEST отношение истинно, то сообщение пытается перейти к следующему по номеру блоку. Если отношение ложно, сообщение пытается перейти к блоку, заданному полем С. Выбор следующего блока производится только один раз (в момент входа сообщения в блок TEST);

2) в режиме условного входа. Если поле С блока TEST пусто (т.е. не указан альтернативный выход), сообщения не могут войти в блок TEST до тех пор, пока условия не изменятся таким образом, что отношение будет истинно. Если отношение истинно, сообщение входит в блок TEST и пытается перейти к следующему по номеру блоку.

4.3.5. Блоки для обработки сообщений, принадлежащих одному семейству

Блоки GENERATE являются основным средством создания сообщений и ввода их в модель. Вход сообщений в блок GENERATE не допускается. Помимо блока GENERATE, для создания сообщений используется также блок SPLIT, который создает заданное число копий вошедшего в блок сообщения. Эти копии принадлежат к тому же семейству, что и породившее их сообщение.

Блок TERMINATE является основным средством уничтожения сообщений и удаления их из модели. Для удаления сообщений, принадлежащих к одному семейству, может быть также использован блок ASSEMBLE.

4.3.5.1. Блок SPLIT

Блок SPLIT имеет следующий формат:

SPLIT <A>,[<B>],[<C>]

Блок SPLIT выполняет функцию копирования входящего в него сообщения, которое называется исходным или порождающим.

В поле А задается число создаваемых копий. Операнд А может быть именем, положительным целым или СЧА. Если вычисленное значение аргумента поля А равно нулю, то блок SPLIT не выполняет никаких операций. После создания копий сообщение пытается перейти к следующему по номеру блоку. Все копии формируются в момент входа порождающего сообщения в блок SPLIT.

Поле В задает номер следующего блока, к которому переходят копии исходного сообщения, причем значение вычисляется для каждой копии отдельно. Операнд В может быть именем, положительным целым, СЧА.

В поле С может быть задан номер параметра, используемого для присвоения копиям последовательных номеров. Операнд С может быть именем, положительным целым, СЧА.

Каждая новая копия становится членом семейства сообщений, порожденного одним исходным сообщением, которое было создано блоком GENERATE. Сообщения, принадлежащие к одному семейству, объединяются интерпретатором в список.

4.3.5.2. Блок ASSEMBLE

Блок ASSEMBLE имеет следующий формат:

ASSEMBLE <A>

Блок ASSEMBLE объединяет заданное число сообщений, принадлежащих к одному семейству, в одно сообщение (т.е. осуществляет сборку заданного числа сообщений). После сборки из блока ASSEMBLE выходит только одно сообщение, которое переходит в следующий по номеру блок. В одном и том же блоке ASSEMBLE возможна одновременная сборка сообщений нескольких семейств. Когда сообщение входит в блок ASSEMBLE, интерпретатор просматривает семейство, к которому принадлежит это сообщение, и проверяет, есть ли другое сообщение из того же семейства в данном блоке ASSEMBLE.

Поле А задает число сообщений, участвующих в сборке. Операнд А может быть именем, положительным целым, СЧА.

4.3.5.3. Блок MATCH

Блок MATCH имеет следующий формат:

MATCH <A>

Блок MATCH используется для синхронизации движения двух сообщений, принадлежащих к одному семейству, без удаления этих сообщений из модели.

Блоки MATCH не объединяют синхронизируемые сообщения. Синхронизация осуществляется путем подбора пар сообщений из одного семейства и задержки этих сообщений до тех пор, пока оба сообщения из одной пары не поступят в заданные точки модели. Сообщения никогда не задерживаются в блоке MATCH. Сообщения, для которых выполнилось условие синхронизации, переходят к следующему по номеру блоку. В одной паре блоков MATCH могут одновременно находиться в состоянии синхронизации пары сообщений из различных семейств. Возможна также одновременная синхронизации пар сообщений из одного семейства в нескольких блоках MATCH.

Поле A задает имя или номер другого блока MATCH, называемого "сопряженным блоком MATCH". Если такого блока нет, происходит останов по ошибке. Операнд A может быть именем, положительным целым, СЧА.

Допускается использование блока MATCH в качестве сопряженного самому себе.

Стандартным числовым атрибутом, связанным с описываемым оператором является:

MB<номер блока> – флаг синхронизации. Возвращает 1, если сообщение, находящееся в блоке <номер блока>, принадлежит к тому же семейству, что и текущее.

4.3.6. Списки пользователя

В GPSS/PC имеется тип списков сообщений, названных списками пользователя, которые дают возможность удалять сообщения из списков текущих событий и переводить их во временно неактивное состояние. Впоследствии эти сообщения возвращаются в список текущих событий.

4.3.6.1. Блок LINK

Блок LINK имеет следующий формат записи:

LINK <A>,<B>,[<C>]

Блок LINK удаляет сообщение из списка текущих событий и помещает его в список пользователя.

Поле A задает номер списка пользователя, в который будет помещено вошедшее сообщение. Операнд A может быть именем, положительным целым, СЧА.

Поле В задает алгоритм упорядочивания списка пользователя. Операнд В может быть LIFO, FIFO, именем, целым или СЧА.

Допустимые операнды:

- FIFO – вошедшее сообщение помещается в конец списка пользователя;
- LIFO – вошедшее сообщение помещается в начало списка пользователя;
- *номер параметра – входящие сообщения располагаются в списке пользователя в соответствии со значением указанного параметра. Сообщения располагаются по возрастанию значения этого параметра. Чем меньше значение параметра, тем ближе сообщение к началу списка, чем больше значение параметра, тем ближе сообщение к концу списка.*

Поле С, указывающее альтернативный выход, используется при описании различных ситуаций, возникающих в очередях. Операнд С может быть именем, положительным целым, СЧА.

Если поле С пусто, индикатор, связанный с заданным списком пользователя, устанавливается в единицу. Это приводит к тому, что все входящие сообщения безусловно заносятся в список пользователя, определенный полем А в том порядке, какой задан полем В.

Если поле С не пустое, проверяется индикатор списка пользователя. Если индикатор списка установлен в единицу, вошедшее сообщение заносится в список пользователя в том порядке, какой задан в поле В. Если индикатор списка установлен в "0", он переводится в единицу, и вошедшее сообщение переходит к блоку, заданному в поле С.

Пример: LINK HOLD,FIFO

В этом примере вошедшее сообщение помещается в конец списка пользователя с именем HOLD.

Стандартными числовыми атрибутами, связанными с описываемым оператором, являются:

CA<номер списка> – среднее число сообщений в списке пользователя с именем <номер списка>;

CC<номер списка> – общее число сообщений в списке пользователя с именем <номер списка>;

CH<номер списка> – текущее число сообщений в списке пользователя с именем <номер списка>;

CM<номер списка> – максимальное число сообщений в списке пользователя с именем <номер списка>;

CT<номер списка> – среднее время пребывания сообщения в списке пользователя с именем <номер списка>, которое рассчитывается следующим образом:

```
                               вычисленный временной интервал
     CT<номер списка> = ------------------------------
                                  общее число входов
```

4.3.6.2. Блок UNLINK

Блок UNLINK имеет следующий формат записи:

UNLINK [<X>]   <A>,<B>,[<C>],[<D>],[<E>],[<F>]

Блок UNLINK удаляет сообщения из списка пользователя. Указатели отношения, записываемые во вспомогательном поле операции <X>, определяют, какое явное условие следует рассматривать. Если указатель отношения не задан, предполагается отношение равенства (E).

Указатели отношения следующие:

- 'G' – больше. Отношение истинно, если значение параметра, заданного в поле D, больше значения, заданного в поле E;

- 'GE' – больше или равно. Отношение истинно, если значение параметра, заданного в поле D, больше или равно значению, заданному в поле E;

- 'L' – меньше. Отношение истинно, если значение параметра, заданного в поле D, меньше значения, заданного в поле E;

- 'LE' – меньше или равно. Если значение параметра, заданного в поле D, меньше или равно значению, заданному в поле E, то отношение истинно;

- 'E' – равно. Отношение истинно, если значение параметра, заданного в поле D, равно значению, заданному в поле E;

- 'NE' – не равно. Отношение истинно, если значение параметра, заданного в поле D, не равно значению, заданному в поле E.

Поле A задает номер списка пользователя, из которого удаляются одно или несколько сообщений. Операнд A может быть именем, положительным целым или СЧА.

В поле B указывается номер блока, к которому переходят удаленные сообщения. Операнд B может быть именем, положительным целым, СЧА.

Поле C задает счетчик числа удаленных сообщений. Операнд C может быть "ALL", именем, положительным целым или СЧА. Значение СЧА является числом удаляемых сообщений. Может быть задано "ALL", что означает удаление всех сообщений.

Пример использования блока UNLINK: UNLINK   HOLD, REENTRY, 1

Первое сообщение из списка пользователя с именем HOLD помещается в блок с именем REENTRY. Оно заносится в список текущих событий за сообщением с таким же приоритетом. Входящее сообщение переходит к следующему блоку.

Стандартными числовыми атрибутами, связанными с описываемым оператором являются те же, что и для LINK.

4.3.7. Устройства

Устройства используются при моделировании систем для имитации работы оборудования единичной емкости, например, процессора, каналов передачи данных, устройств ввода-вывода, линии связи и т.д. Такое оборудование в любой момент времени может обрабатывать только одно сообщение.

4.3.7.1. Блок SEIZE

Блок SEIZE имеет следующий формат:

SEIZE <A>

Операнд А может быть именем, положительным целым или СЧА. Свободный блок SEIZE позволяет вошедшему в него сообщению занять указанное устройство. Блок SEIZE задерживает сообщение, если устройство занято или находится в состоянии недоступности.

В поле А задается имя занимаемого устройства.

Стандартными числовыми атрибутами, связанными с оператором SEIZE, являются:

F<номер устройства> – возвращает 1, если устройство занято, 0 - если свободно;

FC<номер устройства> – общее число входов в устройство;

FI<номер устройства> – возвращает 1, если устройство обслуживает прерывание, 0 в противном случае;

FR<номер устройства> – коэффициент использования устройства, выражается в тысячных долях и возвращает целое число от 0 до 1000;

FT<номер устройства> – среднее время занятости устройства одним сообщением;

FV<имя устройства> – возвращает 1, если устройство доступно и 0 – если недоступно.

4.3.7.2. Блок RELEASE

Блок RELEASE имеет следующий формат:

RELEASE <A>

Блок RELEASE предназначен для освобождения устройства тем сообщением, которым оно было занято.

Если сообщение обрабатывается устройством, то с помощью блока RELEASE оно освобождает устройство и переходит к следующему последовательному блоку.

Операция освобождения выполняется немедленно после входа сообщения в блок RELEASE. В поле А задается номер освобождаемого устройства. Операнд А может быть именем, положительным целым или СЧА.

Стандартными числовыми атрибутами, связанными с описываемым оператором, являются те же, что и для SEIZE.

4.3.8. Многоканальные устройства

При моделировании систем объекты GPSS/PC типа "многоканальные устройства" используются для имитации работы объектов реальных систем, параллельно обрабатывающих поступающие заявки.

4.3.8.1. Оператор описания многоканального устройства

Оператор описания многоканального устройства имеет следующий формат:

<NAME> STORAGE <A>

Оператор STORAGE определяет емкость многоканального устройства в текущей модели.

Поле метки определяет имя многоканального устройства. Поле может быть именем. Операнд А определяет объем многоканального устройства. Операнд А может быть положительным целым.

Стандартными числовыми атрибутами, связанными с описываемым оператором, являются:

- S<номер многоканального устройства> – текущее содержимое. Возвращает емкость заполненной части устройства;
- SA<номер многоканального устройства> – среднее содержимое. Возвращает целую часть среднего заполнения устройства;
- SC<номер многоканального устройства> – счетчик числа входов в многоканальное устройство;
- SE<номер многоканального устройства> – незаполненность устройства. Возвращает 1, если устройство использовалось не полностью, и 0 – в противном случае;
- SF<номер многоканального устройства> – заполненность устройства. Возвращает 1, если устройство используется полностью, 0 – в противном случае;

- SR<номер многоканального устройства> – коэффициент использования многоканального устройства, выраженный в долях тысячи. Возвращает целое в интервале от 0 до 1000;

- SM<номер многоканального устройства> – максимально занятая емкость устройства;

- ST<номер многоканального устройства> – среднее время пребывания сообщения в устройстве;

- SV<номер многоканального устройства> – доступность устройства. Возвращает 1, если устройство доступно, 0 – в противном случае.

Блоки, связанные с описываемым оператором: ENTER, LEAVE.

4.3.8.2. Блок ENTER

Блок ENTER имеет следующий формат записи:

ENTER <A>,[<B>]

Блок ENTER позволяет вошедшему сообщению использовать многоканальное устройство. Сообщение может быть задержано на входе в блок, если многоканальное устройство заполнено, или имеющейся емкости недостаточно, или устройство в данный момент недоступно.

В поле A указывается номер или имя многоканального устройства, куда входит сообщение. Операнд может быть именем, положительным целым или СЧА.

В поле B содержится число занимаемых единиц многоканального устройства. Если поле B пусто, то предполагается, что занимается одна единица. Если это значение равно нулю, то сообщение никогда не задерживается на входе, а блок рассматривается как нерабочий. Операнд может быть именем, положительным целым или СЧА.

Стандартными числовыми атрибутами, связанными с описываемым оператором, являются те же, что и для STORAGE и:

- *R<имя многоканального устройства> – емкость незаполненной части устройства.*

4.3.8.3. Блок LEAVE

Блок LEAVE имеет следующий формат:

LEAVE <A>,[<B>]

Блок LEAVE освобождает определенное число единиц многоканального устройства.

Занятый объем многоканального устройства уменьшается на число освобождаемых единиц. Оставшаяся емкость многоканального устройства увеличивается на ту же величину. Счетчик числа входов не изменяется. Недоступность многоканального устройства не влияет на блок LEAVE.

Поле А блока LEAVE определяет номер или имя многоканального устройства. Операнд может быть именем, положительным целым или СЧА.

Поле В - число освобождаемых единиц многоканального устройства. Операнд может быть именем, положительным целым или СЧА. Если это поле пусто, предполагается 1. Число освобождаемых единиц не должно превышать текущее содержимое многоканального устройства.

Стандартные числовые атрибуты, связанные с описываемым оператором, те же, что и для ENTER.

4.3.8.4. Блоки GATE SNE, GATE SF, GATE SNF, GATE SE, GATE SV

Блоки GATE имеют следующий формат:

GATE <X> <A>,<B>

Блоки GATE позволяют управлять движением сообщений в модели в соответствии со значением логических операторов, связанных с многоканальным устройством (в поле X). К этим операторам относятся:

- SNEj – многоканальное устройство j не пусто;
- SEj – многоканальное устройство j пусто;
- SNFj – многоканальное устройство j заполнено;
- SFj – многоканальное устройство j не заполнено;
- SNVj – многоканальное устройство j недоступно;
- SVj – многоканальное устройство j доступно.

Блок GATE может задержать сообщение на входе, если не задан альтернативный выход.

Поле А определяет имя многоканального устройства, состояние которого проверяется. Операнд А может быть именем, положительным целым числом или СЧА.

Поле В – альтернативный блок, если логический атрибут имеет значение "ложь". Операнд В может быть именем, положительным целым или СЧА. Если поле В определено, то оно должно содержать метку блока, допустимую для текущей модели.

4.3.9. Очереди

В GPSS/PC объекты типа "очередь" вводятся для сбора статистических данных. Статистика об очередях собирается в моменты входа сообщения в блок QUEUE (вход в очередь) или в блок DEPART (выход из очереди). С объектом "очередь" связаны два типа блоков: QUEUE, DEPART.

4.3.9.1. Блок QUEUE

Формат записи блока QUEUE:

QUEUE <A>,[<B>]

Блок QUEUE увеличивает длину очереди.

В поле А задается номер или имя очереди, к длине которой добавляются единицы. Операнд может быть именем, положительным целым или СЧА.

Поле В определяет число единиц, на которое увеличивается текущая длина очереди. Если поле В пусто, то прибавляется единица. Операнд может быть именем, положительным целым или СЧА.

Стандартные числовые атрибуты, связанные с описываемым оператором, следующие:

- Q – текущая длина очереди;
- QA – средняя длина очереди;
- QC – общее число входов в очередь;
- QM – максимальная длина очереди;
- QT – среднее время ожидания в очереди;
- QX – среднее время ожидания в очереди для сообщений с ненулевым временем задержки;
- QZ – число входов в очередь с нулевым временем задержки.

4.3.9.2. Блок DEPART

Блок DEPART имеет следующий формат:

DEPART <A>,[<B>]

Блок DEPART служит для уменьшения длины очереди.

В поле А задается номер или имя очереди, длину которой нужно уменьшить. Операнд может быть именем, положительным целым или СЧА.

В поле В задается число единиц, на которое уменьшается длина очереди. Это число не должно превышать текущую длину очереди. Если поле В пусто, длина очереди уменьшается на единицу. Операнд может быть именем, положительным целым или СЧА.

4.3.10. Операторы контроля управления

4.3.10.1. Оператор START

Оператор START имеет следующий формат записи:

START <A>,[<B>],<C>,[<D>]

Оператор START используется для инициирования начала моделирования.

В поле А задается значение счетчика завершений, определяющего момент окончания прогона модели. В процессе счета сообщения будут входить в блоки

TERMINATE. В поле А блока TERMINATE может задаваться число единиц, вычитаемое из счетчика завершений, заданного в поле А оператора START, при входе сообщения в блок TERMINATE. Операнд А может быть положительным целым.

Поле В – операнд вывода статистики. Операнд может быть "NP" либо опущен. Задание "NP" в поле В приводит к блокировке вывода статистики. По умолчанию выводится стандартная статистика.

Поле С не используется. Сохраняется по аналогии с описаниями других версий GPSS.

Поле D определяет необходимость вывода содержимого списков текущих и будущих событий. Операнд D может быть положительным целым.

Моделирование идет до тех пор, пока счетчик завершения моделирования, определенный операндом А, не достигнет 0. Для уменьшения счетчика используется блок TERMINATE.

Моделирование может закончиться раньше, если достигнута граница времени, определенная оператором SIMULATE, или если нажата клавиша [Esc], или если обнаружены ошибочные условия.

Операнды В и D используются для контроля неотформатированной статистики. Если не используется операнд В, то выводится неотформатированная статистика. Если операнд D не равен 0, то выводятся списки текущих и будущих событий. В других случаях они не выводятся.

Стандартный числовой атрибут, связанный с описываемым оператором TG1 – счетчик завершения.

4.3.10.2. Оператор RESET

Формат записи оператора: RESET

Оператор RESET сбрасывает в ноль статистику и атрибуты системы. Действие оператора RESET можно описать следующим образом:

- значение относительного условного времени (C1) устанавливается в ноль;

- значение абсолютного условного времени (AC1) остается неизменным;

- все датчики псевдослучайных чисел остаются неизменными.

Счетчики блоков (Nj) сбрасываются в ноль.

4.3.10.3. Оператор CLEAR

Формат записи оператора: CLEAR

Оператор CLEAR сбрасывает всю накопленную статистику, удаляет все сообщения из модели и устанавливает отсчет сгенерированных сообщений для блока GENERATE, начиная с 1.

Когда выполняется оператор CLEAR:

- содержимое всех блоков становится равным 0;
- все сообщения удаляются из модели;
- текущие счетчики устанавливаются в 0;
- системное время устанавливается в 0;
- устройства незаняты и доступны;
- многоканальные устройства становятся свободными и доступными;
- общие счетчики устанавливаются равными 0;

- минимальные и максимальные значения устанавливаются равными текущему содержимому очередей, списков пользователя и многоканальных устройств;

- состояние датчиков псевдослучайных чисел не изменяется;

- внутренний счетчик генерируемых сообщений в блоке GENERATE устанавливается в 0.

4.3.10.4. Оператор EQU

Оператор предназначен для присвоения числовых значений именам, используемым в модели.

Оператор имеет следующий формат:

<NAME> EQU   <X>

Здесь NAME – имя в поле метки оператора;

X – выражение.

Имена удаляются из тела программы, находящейся в буфере, после присвоения им числовых значений. Операторы EQU также не включаются в тело модели в буфере редактора GPSS/PC. Однако повторным определением и вводом новых операторов EQU имена могут быть переопределены. В поле метки операторов EQU не могут использоваться метки блоков.

4.3.10.5. Оператор SIMULATE

Оператор предназначен для задания верхней временной границы моделирования. Время – реальное, измеряется в минутах.

Оператор имеет следующий формат:

SIMULATE <A>

Здесь A – предел времени моделирования в минутах.

Пример: SIMULATE 120

В примере задается лимит времени, равный 2 часам. Если реальное время прогона модели достигает предельного значения, то моделирование завершается.

4.4. Руководство пользователя СПО GPSS/PC

В системе GPSS/PC языком моделирования является язык GPSS/PC.

GPSS/PC может эксплуатироваться на ПЭВМ типа IBM PC/XT-AT и совместимых с ними машин в среде операционной системы MS-DOS. Для функционирования GPSS/PC необходимо наличие на диске следующих файлов:

GPSSPC.EXE – головной модуль системы, он осуществляет запуск СПО GPSS/PC;

SETTINGS.GPS – файл настройки системы, используется головным модулем системы. Перед запуском системы должен располагаться на том же диске и в том же каталоге (подкаталоге), что и головной модуль;

STARTUP.GPS – текстовый файл для автоматического запуска команд GPSS/PC. С помощью любого текстового редактора в файл могут быть внесены любые команды GPSS/PC. В дальнейшем, после запуска, система GPSS/PC считывает данные этого файла и отрабатывает каждую встретившуюся в нем команду GPSS/PC;

GPSSREPT.EXE – программа-Редактор выходной статистики GPSS/PC;

Примечание: *если СПО GPSS/PC эксплуатируется на ПЭВМ IBM PC в среде операционной системы MS-DOS ниже версии 5.0, то на том же устройстве, что и GPSS/PC, должны располагаться следующие файлы*:

- *NG.EXE*;
- *NG.INI*;
- *GPTUT.LZH*;
- *GPSSTUT.NG*.

Обычно они находятся в подкаталоге TUT. Запуск GPSS/PC в этом случае осуществляется с помощью файла GPSS.BAT.

Взаимодействие пользователя с системой осуществляется в режиме диалога. Для этого в системе предусмотрен ряд средств, к которым относятся:

- команды GPSS/PC;
- виртуальные окна;
- редактор исходных текстов моделей на языке GPSS;
- редактор выходной статистики GPSS/PC.

4.4.1. Команды GPSS/PC

Команды GPSS/PC необходимы для построения программы модели и интерактивного взаимодействия с моделью. В противоположность операторам описания данных и контроля управления или блокам программы, команды не являются частью языка GPSS/PC. Для спецификации команды достаточно указать несколько первых символов этой команды, обеспечивающих ее уникальность.

4.4.1.1. Команда @<FILESPEC>

Эта команда считывает текст модели из файла с именем FILESPEC в рабочий буфер GPSS/PC.

Формат команды:

@<FILESPEC>

GPSS/PC открывает текстовый файл с именем FILESPEC и начинает считывать в рабочий буфер системы. Если в тексте модели встречается ошибка, то издается звуковой сигнал, текст диагностического сообщения отображается в окне данных (Data Window), и продолжается считывание модели в рабочий буфер GPSS/PC. При этом ошибочная строка не включается в текст считываемой модели. Можно прервать процесс считывания модели, нажав клавишу [Esc]. Можно временно приостановить процесс считывания файла, нажав любую клавишу. Вторичное нажатие клавиши приводит к возобновлению считывания файла.

4.4.1.2. Команда CONTINUE

Эта команда предназначена для возобновления прерванного процесса моделирования.

Формат команды:

CONTINUE

Команда возобновляет процесс моделирования, который был ранее прерван или остановлен командой STOP. Моделирование считается остановленным, когда встречается условие, установленное командой STOP или STEP. Команда CONTINUE продолжает процесс моделирования, но не исключает условие STOP, т.е., если это условие встретится вновь в модели, то оно снова сработает. Условие STOP может быть исключено из модели опцией OFF команды STOP.

Моделирование считается прерванным, если была нажата клавиша [Esc]. В этом случае команда CONTINUE продолжает процесс моделирования.

Команда CONTINUE продолжает процесс моделирования, если счетчик завершения является положительным числом, или, в противном случае, прекращает процесс моделирования. При возобновлении процесса моделирования осуществляется перенумерация блоков модели, если были вставлены новые блоки или удалены некоторые из них; переопределяется размещение блоков и затем начинается процесс планирования сообщений. Генераторы случайных чисел не сбрасываются.

4.4.1.3. Команда DELETE

Команда DELETE удаляет блоки модели из рабочего буфера GPSS/PC.

Формат команды:

DELETE <A>,[<B>]

A – номер первого удаляемого блока;

B – номер последнего удаляемого блока.

Операнды A и B могут быть положительными десятичными числами.

Пример: DELETE 122,145

Эта команда удаляет все блоки, номера которых содержатся в интервале 122÷145.

Если операнд B отсутствует, то удаляется только один блок с номером A.

Для удаления текста всей модели из рабочего буфера GPSS/PC можно указать номера границ блоков, заведомо большие реально существующих.

4.4.1.4. Команда DISPLAY

Команда DISPLAY выводит в окне данных весь или часть текста модели, содержащейся в рабочем буфере GPSS/PC.

Формат команды:

DISPLAY [<A>],[<B>]

A – номер первой отображаемой строки программы;

B – номер последней отображаемой строки программы.

Операнды A и B могут быть положительным десятичным числом. Для отображения всей программы применяется команда DISPLAY без операндов.

Если отсутствует операнд A, но имеется операнд B, то отображение начинается с первого блока программы и до блока с номером B. Если используется только операнд A, то отображается только одна строка программы с номером A.

Для приостановки процесса прокрутки текста на экране дисплея необходимо нажать любую клавишу. Повторное нажатие клавиши вызывает продолжение прокрутки.

4.4.1.5. Команда EDIT

Команда EDIT позволяет модифицировать программу, находящуюся в рабочем буфере модели.

Формат команды:

EDIT <A>

A – номер строки программы. Операнд может быть положительным десятичным числом.

Команда EDIT сначала восстанавливает копию указанного оператора из программы, находящейся в рабочем буфере GPSS/PC. Можно сделать изменения в этой копии, включая и номер блока. После нажатия клавиши [Enter] копия блока помещается в соответствующее место текста модели, находящейся в рабочем буфере. В ходе модификации блока модели одновременно осуществляется синтаксический анализ модифицируемого блока. При выявлении ошибки выдается звуковой сигнал. Команда EDIT запрещает ввод ошибочного блока программы модели.

В режиме TYPE-OVER MODE осуществляется изменение строки программы без вставки новых символов. Если нажать клавишу [Ins], то EDIT переходит в режим INSERT MODE, что дает возможность вставлять новые символы в модифицируемую строку.

EDIT использует следующие специальные клавиши или комбинации клавиш:

[Ins] – переключить в режим вставки (INSERT MODE) и обратно (TYPE-OVER MODE);
[<-] – переместить курсор влево на один символ;
[->] – переместить курсор вправо на один символ;
[Ctrl],[<-] – переместить курсор влево на одно слово;
[Ctrl],[->] – переместить курсор вправо на одно слово;
[Del] – удалить один символ;
[End] – переместить курсор к концу строки;
[Esc] – выйти из EDIT без изменения строки.

4.4.1.6. Команда RENUMBER

Эта команда перенумеровывает все блоки программы модели, находящиеся в рабочем буфере GPSS/PC.

Формат команды:

RENUMBER [<A>],[<B>]

A – номер первого блока. Может быть положительным десятичным числом, содержащим не более 6 знаков.

B – шаг (приращение) нумерации. Может быть положительным десятичным числом, содержащим не более 6 знаков.

Пример: RENUMBER 10,.1

Эта команда нумерует блоки в следующем порядке: 10.1, 10.2, и т.д. Если операнды A и B не используются, то по умолчанию нумерация начинается с номера 10 с шагом 10.

4.4.1.7. Команда SAVE

Команда переписывает текст модели, содержащийся в рабочем буфере GPSS/PC в указанный текстовый файл MS DOS.

Формат команды:

SAVE <A>,[<B>],[<C>]

A – имя файла, в который записывается программа модели;

B – номер блока модели, начиная с которого осуществляется запись программы в файл;

C – номер блока модели, по какой включительно осуществляется запись программы в файл.

При отсутствии операндов B и C в файл записывается весь текст программы модели. При отсутствии операнда B и наличии операнда C в файл записывается текст модели, начиная с первого оператора и по оператор, указанный операндом C. При отсутствии операнда C и наличии операнда B в файл записывается только один оператор программы модели с номером, указанным операндом B. Если будет использована команда END с несохраненной рабочей моделью, то GPSS/PC выдает предупреждающее сообщение с возможностью предотвратить выход из системы.

4.4.1.8. Команда STEP

Команда задает возможность прерывания процесса моделирования при прохождении указанного количества блоков.

Формат команды:

STEP <A>

А – счетчик прохождения блоков. Операнд может быть положительным целым.

Пример: STEP 1

Процесс моделирования прерывается при прохождении одного блока модели.

Когда процесс моделирования прерывается, GPSS/PC переходит в состояние, вызываемое клавишей [Esc]. В верхней части экрана появляется сообщение трассировки. Когда используется команда STEP в модели должно быть активное сообщение. Модель, которая стартовала с использованием команды STEP, не может быть завершена при нулевом счетчике завершения. Модель завершится лишь при прохождении указанного в операнде А команды STEP количества блоков.

Кода команда STEP выполняется, то

- счетчик завершения не устанавливается;
- если необходимо, перенумеровываются блоки;
- все блоки, генерирующие сообщения, приводятся в исходное состояние;
- генераторы случайных чисел не сбрасываются.

4.4.1.9. Команда STOP

Команда устанавливает или снимает условие прерывания моделирования.

Формат команды:

STOP [<A>],[<B>],[<C>]

А – номер сообщения. Операнд может быть положительным целым;

В – номер блока. Операнд может быть положительным целым;

С – флаг состояния команды. Операнд может быть ON или OFF. По умолчанию ON.

Пример: STOP 100,52

Эта команда устанавливает условие прерывания модели при входе сообщения с номером 100 в блок с номером 52.

Команда STOP с опцией ON устанавливает условие прерывания моделирования, но не стартует модель. Для запуска моделирования должна использоваться последовательность команд START, STEP, CONTINUE.

Когда выполняется прерывание по условию команды STOP, система переходит в состояние, вызываемое нажатием клавиши [Esc]. В верхней части экрана высвечивается трассируемое сообщение. Команда CONTINUE позволяет выйти из прерывания и продолжить моделирование, однако условие прерывания, введенное ранее командой STOP, остается включенным. Для отключения условия прерывания необходимо войти в команду STOP с флагом OFF. Если отсутствует операнд A команды STOP, то любой транзакт может вызвать условие прерывания. Если отсутствует операнд B команды STOP, то любой блок модели удовлетворяет условию прерывания. Команда STOP без операндов вызывает немедленное прерывание процесса моделирования.

4.4.1.10. Команда USERCHAINS

Команда отображает содержимое списков пользователя на экране дисплея.

Формат команды:

USERCHAINS

Команда USERCHAINS переходит в окно данных и отображает в нем все члены списков пользователя.

4.4.1.11. Команда WINDOW

Команда WINDOW предназначена для открытия виртуальных окон GPSS/PC.

Формат команды:

WINDOW <A>,[<B>]

Операнд A позволяет Вам выбрать окно из набора окон GPSS/PC. Операнд может быть: BLOCKS, DATA, FACILITIES, MATRICES, POSITIONS, STORAGES, TABLES.

Операнд B определяет объект, наиболее близко расположенный к левому верхнему углу окна, в случае одновременного отображения в окне нескольких объектов одного типа.

В окнах таблиц и матриц, как правило, отображается один объект за один раз. При наличии нескольких объектов типа таблиц и матриц полный просмотр можно осуществить с помощью управляющих клавиш [PgUp], [PgDn], [End]. Операнд В не используется для окон DATA и POSITIONS.

4.4.1.12. Команда END

Эта команда предназначена для завершения работы с системой GPSS/PC.

Формат команды:

END

В результате выполнения команды END система завершает работу и возвращает управление DOS. Если в буфере редактора GPSS/PC проводились корректировки и не было сохранения буфера, то система выдает соответствующее предупреждающее сообщение с рекомендацией подумать, не стоит ли сохранить в файле содержимое буфера, прежде чем завершить работу с системой.

При работе с командами GPSS/PC можно использовать функциональные клавиши. Они запрограммированы следующим образом:

F1 – DISPLAY	F5 – RENUMBER
F2 – EDIT	F6 – STEP
F3 – SAVE	F7 – STOP
F4 – DELETE	F8 – END

4.4.2. Виртуальные окна и редактор исходных текстов моделей

Под виртуальным окном (имеется шесть графических окон и одно псевдографическое окно) понимается отображение информации о состоянии отдельных объектов на экране дисплея.

GPSS/PC предоставляет в распоряжение пользователя семь виртуальных окон:

- псевдографическое окно данных (DATA WINDOW);
- графическое окно блоков (BLOCKS WINDOW);
- графическое окно устройств (FACILITIES WINDOW);
- графическое окно многоканальных устройств (STORAGES WINDOW);
- графическое окно матриц (MATRICES WINDOW);
- графическое окно позиций (POSITIONS WINDOW);
- графическое окно таблиц (TABLES WINDOW).

Графические окна матриц, позиций и таблиц используются очень редко, поэтому работа с ними не рассматривается.

Все виртуальные окна GPSS/PC могут быть открыты с помощью команды WINDOWS. Кроме того, окна могут быть открыты путем одновременного нажатия на клавиатуре ПЭВМ клавиши [Alt] и клавиши, соответствующей первому символу в имени окна.

Пользователь может активно вмешиваться в процесс отображения информации в виртуальном окне, выдавая различные команды из набора команд GPSS/PC и/или используя специальные поля команд, имеющиеся в ряде виртуальных окон GPSS/PC.

Вы можете взаимодействовать с меню окон с помощью устройства управления позицией курсора окна (см. рис 4.5). Выбор нужной команды производится с помощью перемещения курсора окна в нужную часть поля команд и последующего нажатия клавиши [Ins]. Перемещение курсора окна осуществляется с помощью клавиш управления курсором (стрелки влево, вправо, вверх, вниз). Если вы хотите отказаться от выполнения выбранной команды, нажмите клавишу [Esc].

Использование управляющих клавиш позволяет осуществлять просмотр информации, не умещаемой в одном окне. Так, клавиша [PgUp] осуществляет листание информации в окне назад. Клавиша [PgDn] осуществляет листание информации в окне вперед.

В окнах можно запросить трассировку прохождения сообщений по блокам модели в процессе имитации, нажав клавиши [Alt] и [L]. Это вызовет появление в верхней части окна строки (см. рис. 4.5), на которой отображается следующая информация:

- Time – текущее модельное время;
- XACT – номер участвующего в трассировке сообщения;
- leaves – номер пройденного сообщением блока;
- enters – номер блока, в который входит сообщение;
- ASSIGN – тип последнего блока.

При повторном нажатии клавиш [Alt] и [L] строка трассировки исчезает.

При вводе команд, не ориентированных на работу в текущем окне, система автоматически открывает нужное окно, меняя картинку на экране дисплея.

При появлении в окнах лишней информации (это может быть диаграмма устройства, имя которого было исправлено) или при выдаче ошибки системы (System error) необходимо выйти в DOS, перезапустить систему и повторить прогон моделирования.

4.4.2.1. Окно данных

Окно данных (DATA WINDOW) предназначено для создания, и модификации исходных текстов моделей, с помощью встроенного в систему GPSS/PC строчного Редактора исходных текстов. Смена страниц отображаемой информации производится по запросу системы "Press space bar for move, any other key to resume" путём нажатия клавиши "пробел".

На рис. 4.3 изображен фрагмент модели в окне данных. Окно данных представляет собой 20 строк экрана дисплея, ограниченных сверху и снизу рамкой. Сразу под верхней чертой рамки окна располагается строка, предназначенная для отображения системой сообщений о тяжелых ошибках, обусловивших прекращение, либо невозможность начала выполнения, запрошенных пользователем действий (на рис.4.3 она пустая).

Следующие 18 строк окна данных предназначены для отображения данных, вводимых с клавиатуры, либо выдаваемых системой в ответ на вводимые команды. Сюда же система выводит данные информационного и диагностического характера.

Двадцатая строка окна предназначена для выдачи системой сообщений-запросов о необходимости перехода к следующей странице, при постраничной организации работы окна.

```
______________________________ GPSS/PC _______________________________
100 ;*********Use POSACME.GPS for the POSITION.GPS file.*********
102 ;
104 ;
106 ;           TIME DEFINITIONS
108 ;
110 ;
112 ;
114 MOVE_TIME EQU 4
116 BODY_INTERARRIVALS EQU 60
118 PRIMER_TIME EQU       15
120 PAINTER_TIME EQU       22
122 CUST_INTERARRIVALS EQU 60
124 PRIMER_RESET_TIME EQU 40
126 PAINTER_RESET_TIME EQU 18
128 ;
130 ;
132 ;                 Color Definitions
134 ;
******Press space bar for more, any key to resume.******
______________________________________________________________________
136 ;                                <--------   Командная строка
Reading ACME.GPS                 <------------   Строка состояния
```

Рис. 4.3. Фрагмент программы в окне данных

Сразу под нижней чертой рамки окна расположена командная строка окна данных. Через командную строку осуществляется ввод в систему команд

GPSS/PC и строк текстов модели программ пользователя. В этой же строке производится корректировка ранее введенных в буфер редактора GPSS/PC строк программ моделей пользователя. При выполнении системой команды @ в командной строке высвечиваются последовательно считываемые из файлов строки программ моделей пользователя.

Данные, попадающие в командную строку, при считывании их системой из файла, подвергаются синтаксическому контролю. При обнаружении ошибок в поле данных окна отображается соответствующая диагностическая информация.

При вводе данных с клавиатуры система осуществляет контроль правильности ввода информации в каждую позицию командной строки.

Ниже командной строки, с отступом в одну строку, располагается строка состояния – служебная строка, предназначенная для отображения различной информации, зависящей от выполняемых системой функций. Так, при считывании данных из файла модели пользователя в буфер строчного Редактора системы, в строке состояния отображается спецификация считываемого файла (см. рис. 4.3). При работе со строчным Редактором GPSS/PC, в строке состояния высвечивается информация о выполняемых Редактором командах GPSS/PC. В этой же строке отображается информация о начале, продолжении, прерывании или завершении процесса имитации.

В процессе имитации в правом конце нижней черты рамки окна данных высвечивается мерцающий индикатор имитации.

4.4.2.2. Создание и модификация моделей в окне данных с помощью редактора исходных текстов моделей

После запуска GPSS/PC на экране высвечивается окно данных. С помощью команды @ пользователь может считать в буфер Редактора файл, содержащий нужную программу модели. Впоследствии он может ее многократно корректировать и запускать на выполнение. (см. команды GPSS/PC). После внесения изменений в программу необходимо записать ее на жесткий диск, а затем запустить на выполнение командой @<FILESPEC>.

В случае создания новой программы с помощью встроенного редактора GPSS/PC пользователь последовательно набирает строки программы в командной строке окна данных и вводит их в буфер GPSS/PC клавишей ввода. Редактор GPSS/PC, добавляя новые строки в буфер, ориентируется на их номера. При этом возможно добавление строк в любое место программы.

Возможность указания номеров в виде десятичных дробных чисел позволяет вносить изменения в программу в больших объемах.

Строка описания оператора GPSS/PC состоит из следующей последовательности полей:

- номер строки. Начинается с первой позиции строки. Содержимым поля может быть десятичное число из символов, в том числе десятичное дробное число, в последнем случае десятичная точка рассматривается как один из семи символов. Пример: 105.7;

- поле метки. Используется в зависимости от типа операции. Содержимым поля является некоторое имя. В операторах описания данных поле метки обязательно должно быть заполнено;

- поле операции;

- поле операндов. Содержимое этих полей, их наличие и количество зависит от типа операции;

- поле комментариев. Необязательное поле. Содержит информацию, поясняющую назначение оператора. В данной версии допускается запись комментариев с использованием прописных и строчных букв латинского алфавита. Отделяется от поля операндов символом ";". Допускается запись комментариев с начала строки. В этом случае в первой позиции строки ставится символ ";" или "*".

Если при записи оператора пропускаются необязательные операнды, то их отсутствие отмечается символом ",".

Пример строки программы:

40 MET ADVANCE 10,5 ;OBRABOTKA

Встроенный в систему GPSS/PC редактор исходных текстов модулей осуществляет переход к следующему полю строки оператора, при вводе в конце текущего поля пробела или запятой.

Строка описания оператора может содержать до 79 символов.

Завершение ввода строки описания оператора отмечается нажатием клавиши [Enter].

При использовании для создания программ модулей встроенного редактора GPSS/PC последний, управляя перемещением курсора по полям вводимой строки, выдает в начале каждого поля строки подсказку, поясняющую назначение поля:

начало строки отмечается символом ">";

символ "L" отмечает поле метки;

символ "V" отмечает поле операции;

символ ";" отмечает поле комментариев;

символ "X" отмечает поле описания выражения в операторах описания переменных и блоков;

символами "A", "B", "C", "D", "E", "F", "G" отмечаются поля операндов.

Ввод описания очередного оператора в состав программы модели осуществляется нажатием клавиши [Enter] . Отказ от ввода описания очередного оператора в программу модели и удаление его из буфера редактора выполняется при нажатии клавиши [Esc].

С помощью клавиш [<--] или [Backspace] осуществляется возврат курсора на одну позицию влево с удалением ранее введенного символа.

4.4.2.3. Окно блоков

Окно блоков (BLOCKS WINDOW) предназначено для графического отображения блоков программ моделей на языке GPSS. На рис. 4.4 приведен пример окна блоков. Если на ПЭВМ имеется дисплей с адаптером EGA, то блоки представляются в виде принятых в GPSS/PC изображений блоков. В противном случае, они представляются как прямоугольники.

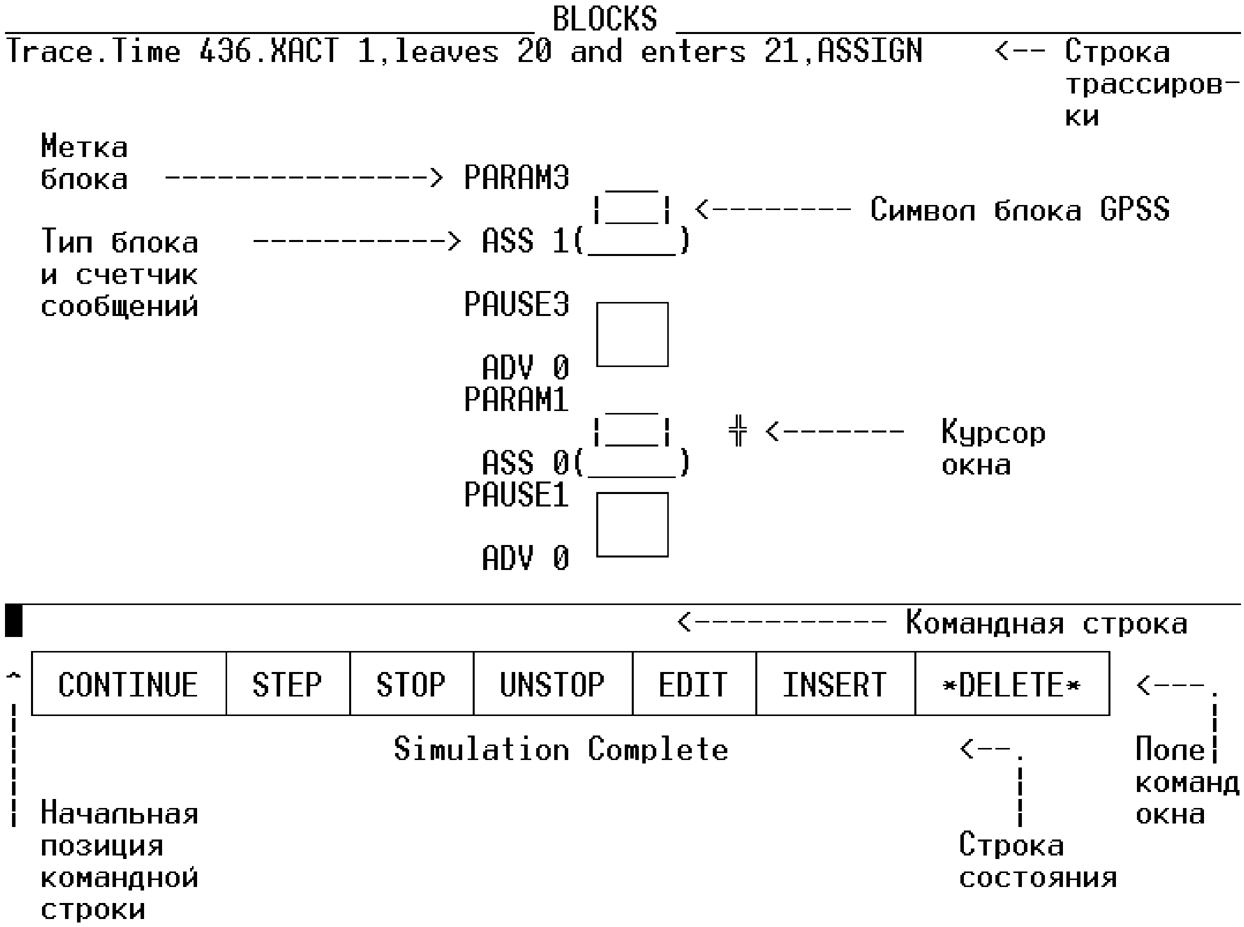


Рис. 4.4. Фрагмент модели в окне блоков

Каждый блок в текущей модели представлен как ячейка информации в окне блоков. Каждая такая ячейка содержит изображение блока, первые три символа из названия блока, метку блока и один из двух возможных счетчиков сообщений. Цвет представления блока зависит от числа сообщений, находящихся в блоке.

Блоки располагаются в окне сверху-вниз и слева-направо. Окно блоков имеет заголовок "BLOCKS" , расположенный посередине верхней рамки окна. При цветном мониторе цвет представления блока изменяется по мере накопления в нем сообщений. Обычно блок без сообщений зеленого цвета, с одним сообщением - белого цвета, от 2 до 9 сообщений - коричневого цвета, больше 10 сообщений - красного цвета. Это сделано для того, чтобы привлечь ваше внимание к переполнению областей модели.

Когда сообщение входит в блок, его изображение мелькает с высшей яркостью его нормального цвета, который определяется числом сообщений, находящихся в блоке.

Меню окна блоков состоит из семи команд, которые могут быть выбраны:

- CONTINUE – продолжает моделирование после прерывания работы модели;

- STEP – для создания условий прерывания работы модели после прохождения сообщением одного блока модели;

- STOP – устанавливает условие останова в выбранном блоке. Попытка любого сообщения войти в этот блок приводит к останову моделирования;

- UNSTOP – удаляет все условия останова;

- EDIT – редактирование оператора GPSS, связанного с выбранным блоком. Команда может быть прервана нажатием клавиш [Esc].

- INSERT – подготовка к вставке блока непосредственно после выбранного блока;

- DELETE – удаление выбранного блока из текущей модели и сохраняемой программы.

С помощью клавиши [Esc] процесс моделирования с отображением динамики моделирования в окне блоков можно прервать. Продолжение моделирования - команда CONTINUE.

Команды STOP, EDIT, INSERT, DELETE из строки команд требуют выбора блока перед их использованием. Выбор блока осуществляется путем наложения курсора окна на изображение блока с последующим нажатием клавиши [Ins]. После выбора объекта необходимо курсор переместить в позицию поля команд, соответствующую выбранной команде, и нажать клавишу [Ins].

При использовании команды EDIT в командной строке высвечивается строка программы, соответствующая выбранному блоку. Пользователь может откорректировать строку и ввести новое ее содержимое в буфер Редактора, нажав клавишу ввода.

4.4.2.4. Окно устройств

Окно устройств (FACILITIES WINDOW) предназначено для графического отображения блоков программ моделей на языке GPSS. В окне может быть представлено до 4-х устройств. На рис. 4.5 приведен пример окна устройств.

Окно устройств имеет заголовок "FACILITIES", расположенный посередине верхней рамки окна.

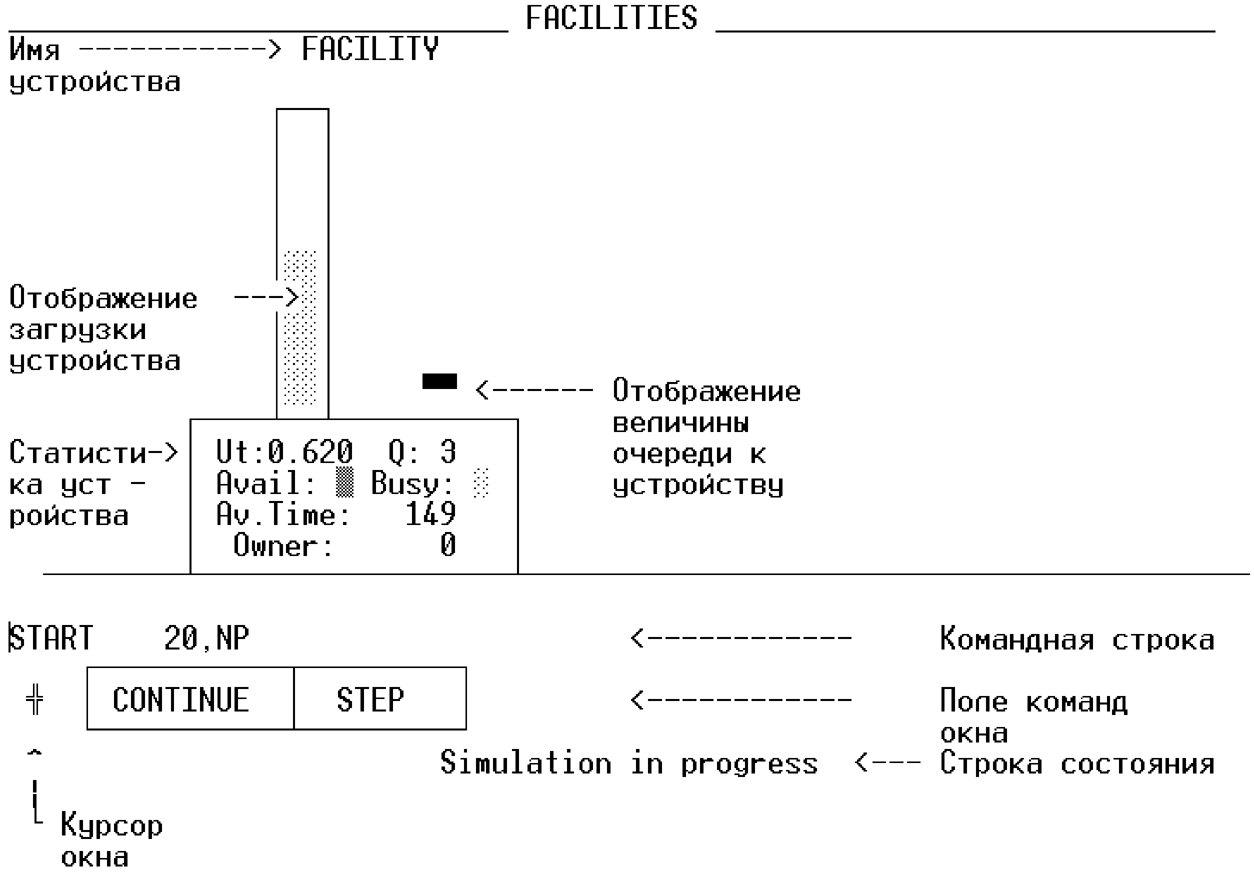


Рис. 4.5. Окно устройств

Каждое устройство (см. рис. 4.5) представлено информацией, которая включает следующие элементы:

1) метка - имя или номер устройства;

2) вертикальный столбец, представляющий загрузку устройства с момента последнего выполнения оператора RESET;

3) вертикальный столбец, представляющий число сообщений из цепи устройств;

4) в статистике устройства содержится информация:

- Ut: – процент загрузки устройства с момента последнего выполнения оператора RESET;

- Q: – количество задержанных, прерванных и ожидающих сообщений;

- Avail: – индикатор доступности. Заштрихованный прямоугольник означает, что устройство доступно;

- Busy: – индикатор занятости. Заштрихованный прямоугольник означает, что устройство занято;

- Av.Time: – среднее время пребывания в устройстве;

- Owner: – номер сообщения, занимающего устройство.

Цвет вертикального столбца, представляющего загрузку устройства с момента последнего выполнения оператора RESET, изменяется в зависимости от загрузки устройства. Обычно зеленый цвет присутствует при небольшой загрузке (0,25% использования устройства), белый (25÷50%), коричневый (50÷75%) и красный (75÷100%).

Цвет вертикального столбца, представляющего отображение величины очереди к устройству, изменяется в зависимости от длины очереди. Обычно зеленый представляет маленькую очередь (0÷9 сообщений), белый (10÷19), коричневый (20÷99) и красный (100 и более). Красный цвет дает вам понять, что достигнуты максимальные значения каких-либо статистических данных.

Меню окна устройств (см. поле команд окна) состоит из двух команд, которые могут быть выбраны:

- CONTINUE – продолжает моделирование после прерывания работы модели;
- STEP – для создания условий прерывания работы модели после прохождения сообщением одного блока модели.

4.4.2.5. Окно многоканальных устройств

Окно многоканальных устройств (STORAGES WINDOW) дает графическое представление о многоканальных устройствах GPSS, фигурирующих в текущей модели. В окне может быть представлено до 4-х устройств. Окно многоканальных устройств имеет заголовок "STORAGES" , расположенный посередине верхней рамки окна. На рис. 4.6 приведен пример окна многоканальных устройств.

Каждое многоканальное устройство представлено информацией, которая включает следующие элементы:

1) метка – имя или номер многоканального устройства;

2) левый вертикальный столбец, представляющий загрузку многоканального устройства с момента последнего выполнения оператора RESET;

3) правый вертикальный столбец, представляющий текущую загрузку многоканального устройства;

4) прямоугольник со статистикой устройства, включающий следующие данные:

- Ut: – процент загрузки многоканального устройства с момента последнего выполнения оператора RESET;
- Q: – количество задержанных или ожидающих сообщений;

- Avail: – индикатор доступности. Заштрихованный прямоугольник означает, что многоканальное устройство доступно;

- Part Used: – процент загрузки многоканального устройства в текущий момент времени;

- Content: – загрузка многоканального устройства в текущий момент модельного времени.

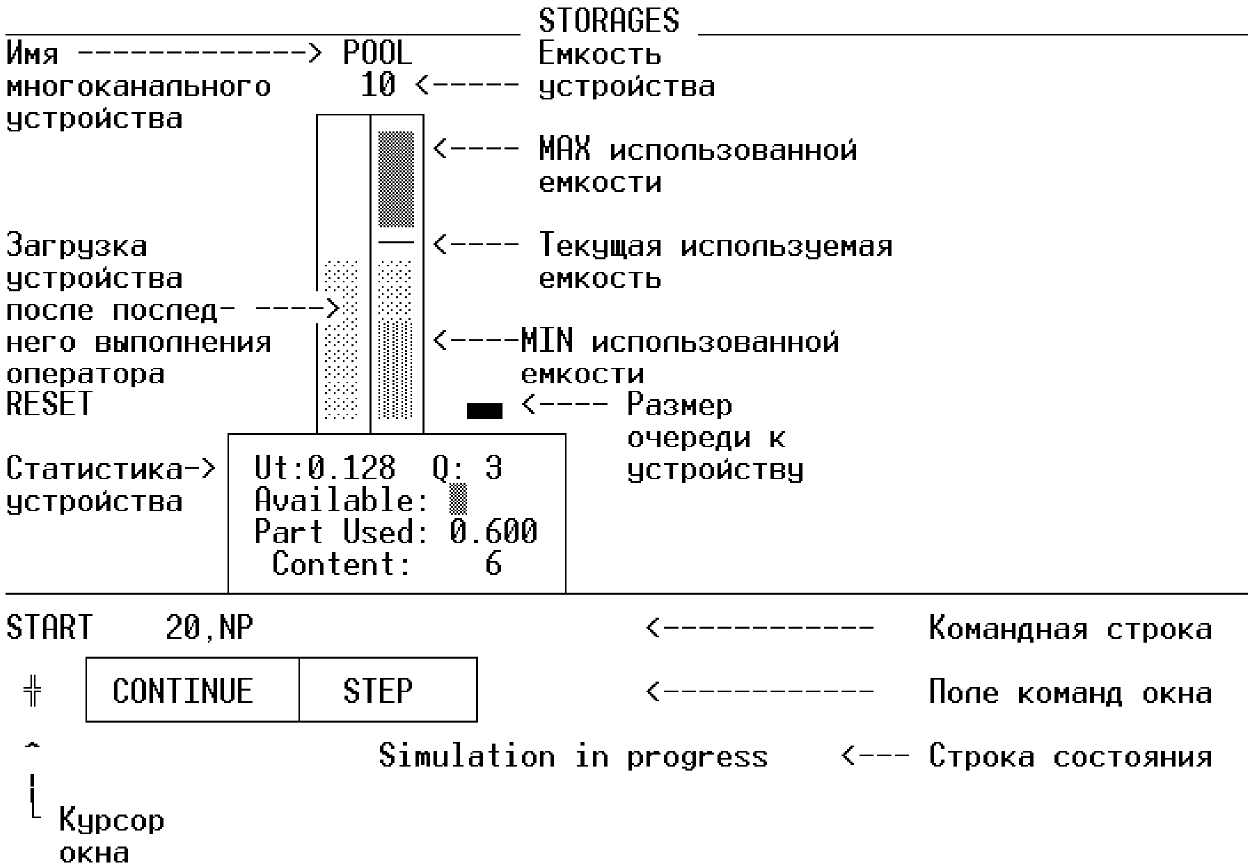


Рис. 4.6. Окно многоканальных устройств

Цвета, определяющие загрузку многоканального устройства и очередь к нему, те же, что и в окне FACILITIES.

Меню окна STORAGES состоит из двух команд, которые аналогичны командам окна FACILITIES.

4.4.3. Стандартная выходная статистика

Файл стандартной выходной статистики создается по умолчанию в не отформатированном виде (файл REPORT.GPS). Форматирование и вывод файла статистики на устройства отображения либо в файл осуществляются Редактором стандартной выходной статистики GPSS/PC. Имя программы Редактора – GPSSREPT.EXE. Файл REPORT.GPS создается после выполнения моделирования. Если вы хотите сохранить результаты моделирования, то необходимо провести форматирование и записать результаты в файл на диске. После каждого прогона программы модели содержимое файла REPORT.GPS обновляется.

4.4.3.1. Меню редактора выходной статистики

Запуск Редактора выходной статистики осуществляется из DOS с помощью файла GPSSREPT.EXE.

При этом на экране дисплея появится меню. Мерцающий курсор указывает, какую характеристику вы определяете. Для того чтобы перейти к следующей характеристике, нажмите клавишу [Enter]. В меню программы GPSSREPT необходимо указать спецификацию устройства либо файла, в который будет помещен отформатированный файл выходной статистики (по умолчанию SCRN:).

Другая характеристика определяет спецификацию неотформатированного файла выходной статистики REPORT.GPS. Её изменять не надо. Эта характеристика задается в файле настройки системы. В позиции (SCRN:) можно указать:

- пробел – для вывода отформатированного файла статистики на экран;
- имя файла – для вывода отформатированного файла статистики в файл на диске;
- PRN: – для вывода отформатированного файла статистики на устройство печати.

После того как указаны характеристики REPORT.GPS и SCRN:, необходимо нажать клавишу [Space] для начала процесса форматирования. Если вам необходимо вернуться в DOS, нажмите клавишу [Esc].

При выводе отформатированного файла статистики на экран программа GPSSREPT предоставляет возможность его постраничного просмотра. Переход к очередной странице осуществляется нажатием клавиши [PgDn]. При этом на экран будет выводиться первая строка следующей страницы текущего подраздела, а затем первая страница следующего подраздела.

4.4.3.2. Описание элементов файла статистики

Отформатированный файл статистики состоит из подразделов, содержащих стандартную статистику об объектах GPSS\PC, используемых в данной модели (FACILITY, QUEUE, STORAGE и т.д.). Начинается файл статистики с заголовка, который берется из поля комментария, расположенного перед началом программы. Заголовок появляется на каждой странице файла статистики. После заголовка автоматически вставляется подзаголовок, который содержит имя неотформатированного файла статистики, номер версии GPSS\PC, серийный номер, дату и время моделирования.

Пример: GPSS\PC Report file TEST (V2) 06-24-1989 21:57:38

Далее следует строка, содержащая основную информацию о результатах работы модели. Например:

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES	FREE MEMORY
0	10850	24	1	1	15850

Элементы статистики, представленные в этой строке имеют следующее содержание:

- START TIME – абсолютное системное время в момент начала моделирования. Оно эквивалентно абсолютному системному времени после последнего применения операторов RESET или CLEAR;

- END TIME – абсолютное время, когда счетчик завершений принимает значение 0;

- BLOCKS – количество блоков, использованных в текущей модели, к моменту завершения моделирования;

- FACILITIES – количество устройств, использованных в модели, к моменту завершения моделирования;

- STORAGES – количество многоканальных устройств, использованных в текущей модели, к моменту завершения моделирования;

- FREE MEMORY – количество байтов памяти, доступной для дальнейшего использования.

Затем в файле статистики следует информация об именах, которые просматривает GPSS/PC в ходе моделирования. Информация об именах имеет следующий вид:

NAME	VALUE	TYPE
MOTOR	10001	2

Поле NAME отмечает имена, содержащиеся в программе модели.

Поле VALUE определяет числовое значение, соответствующее имени. Система устанавливает начальный номер равным 10000.

Поле TYPE равно 0, если значение имени устанавливает пользователь; равно 2, если значение имени устанавливает система; 3 – если имя является именем блока.

Далее описываются блоки текущей модели в виде:

```
LINE  LOC  BLOCK TYPE   ENTRY COUNT  CURRENT COUNT  RETRY
 90    1   GENERATE         333            0           0
```

Поле LINE определяет номер строки в рабочей модели, связанный с блоком GPSS/PC.

Поле LOC определяет имя или номер этого блока.

Поле BLOCK TYPE определяет тип блока GPSS/PC.

Поле ENTRY COUNT определяет количество транзактов, вошедших в данный блок, после последнего выполнения блоков RESET или CLEAR, или с начала работы программы.

Поле CURRENT COUNT определяет количество транзактов, находящихся в данном блоке в конце моделирования.

Поле RETRY определяет количество транзактов, ожидающих специальных условий, зависящих от состояния данного блока.

Если в модели используются объекты типа "устройство", то далее в файле статистики идет информация об этих объектах. Например:

```
FACILITY  ENTRIES  UTIL.  AVE.TIME  AVAILABLE  OWNER  PEND  INTER
 TELLER     254    0.996  395.67        1        291    0     0
  RETRY      DELAY
    0          78
```

Поле FACILITY определяет номер или имя объекта типа "устройство".

Поле ENTRIES определяет количество раз, когда устройство было занято или прервано после последнего выполнения блоков RESET или CLEAR или с начала работы программы.

Поле UTIL. определяет часть периода моделирования, в течение которого устройство было занято.

Поле AVE.TIME определяет среднее время занятости устройства одним сообщением в течение периода моделирования после последнего выполнения операторов RESET или CLEAR.

Поле AVAILABLE определяет состояние готовности устройства в конце периода моделирования. Оно равно 1, если устройство готово и 0 – если не готово.

Поле OWNER определяет номер последнего сообщения, занимавшего устройство. 0 означает, что устройство не занималось.

Поле PEND определяет количество сообщений, ожидающих устройство, находящееся в "режиме прерывания".

Поле INTER определяет количество сообщений, прерывающих устройство в данный момент (счетчик сообщений в списке прерывания).

Поле RETRY определяет количество сообщений, ожидающих специальных условий, зависящих от состояния объекта типа "устройство".

Поле DELAY определяет количество сообщений, ожидающих занятия устройства. Сюда входят также сообщения, ожидающие освобождения устройства в "режиме прерывания" (блок PREEMPT).

В случае использования в модели объектов типа "очередь", далее следует информация об этих объектах. Например:

```
QUEUE  MAX CONT. ENTRIES ENTRIES(0) AVE.CONT. AVE.TIME  AVE.(-0) RETRY
 MOD   78   78    332        1        36.24   11613.51  11848.6    0
```

Поле QUEUE определяет имя или номер объекта типа "очередь".

Поле MAX определяет максимальное содержимое объекта типа "очередь" в течение периода моделирования, который начинается с начала работы программы или с последнего оператора RESET или CLEAR.

Поле CONT определяет текущее содержимое объекта типа "очередь" в конце периода моделирования.

Поле ENTRIES определяет общее количество входов в очередь в течение периода моделирования (счетчик входов).

Поле ENTRIES(0) определяет общее количество входов в очередь с нулевым временем ожидания (счетчик "нулевых" входов).

Поле AVE.CONT определяет среднее значение содержимого очереди.

Поле AVE.TIME определяет среднее время, проведенное в очереди с учетом всех входов в очередь.

Поле AVE.(–0) определяет среднее время, проведенное в очереди без учета "нулевых" входов в очередь.

Поле RETRY определяет количество сообщений, ожидающих специальных условий, зависящих от состояния объекта типа "очередь".

Если в модели использовались объекты типа "многоканальное устройство", то далее в файле статистики идет информация об этих объектах. Например:

```
STORAGE  CAP. REMAIN MIN MAX ENTRIES AVL. AVE.C. UTIL. RETRY DELAY
 MOTOR    3     3     0   1    50     1    0.99  0.331   0     0
```

Поле STORAGE определяет имя или номер объекта типа "многоканальное устройство".

Поле CAP. определяет емкость многоканального устройства, заданную оператором STORAGE.

Поле REMAIN определяет число единиц свободной емкости многоканального устройства в конце периода моделирования.

Поле MIN определяет минимальное количество используемой емкости многоканального устройства за период моделирования.

Поле MAX определяет максимальное количество используемой емкости многоканального устройства за период моделирования.

Поле ENTRIES определяет количество входов в многоканальное устройство за период моделирования.

Поле AVL. определяет состояние готовности многоканального устройства в конце периода моделирования. 1 означает, что многоканальное устройство готово, 0 – не готово.

Поле AVE.C определяет среднее значение занятой емкости за период моделирования.

Поле UTIL. определяет часть периода моделирования, в течение которого многоканальное устройство использовалось.

Поле RETRY определяет количество сообщений, ожидающих специальных условий, зависящих от состояния многоканального устройства.

Поле DELAY определяет количество сообщений, ожидающих возможности входа в блок ENTER.

Далее в файле статистики выводятся списки пользователя. Например:

```
USER CHAIN  CHAIN SIZE  RETRY  AVE.CONT  ENTRIES   MAX  AVE.TIME
 TAXILINE       202       0    100.70     252      203  40289.50
```

Поле USER CHAIN определяет номер или имя объекта типа "список пользователя".

Поле CHAIN SIZE определяет количество сообщений в списке пользователя в конце периода моделирования.

Поле RETRY определяет количество сообщений, ожидающих наступления специальных условий, связанных с состоянием объекта типа "список пользователя".

Поле AVE.CONT определяет среднее содержимое списка пользователя в течение периода моделирования.

Поле ENTRIES определяет общее количество сообщений, помещаемых в список пользователя в течение периода моделирования.

Поле MAX определяет максимальное количество транзактов в списке пользователя за период моделирования.

Поле AVE.TIME определяет среднее время пребывания транзакта в списке пользователя.

Список литературы.

1. Гмурман, В.Е. Введение в теорию вероятностей и математическую статистику [Текст] / В.Е. Гмурман. – М.: Наука, 1997. – 380 с.
2. Кочетков, Е.С. Теория вероятностей и математическая статистика [Текст] / Е.С. Кочетков, С.О. Смерчинская, В.В. Соколов. – М.: ИНФРА, 2003. – 240 с.
3. Ашихмин, В.Н. Введение в математическое моделирование [Текст] / В.Н. Ашихмин, М.Б. Гитман, Ч.Э. Келлер и др.; под ред. В.П. Трусова. – М.: Логос, 2004. – 440 с.
4. Костомаров, Д.П. Вводные лекции по численным методам [Текст] / Д.П. Костомаров. – М.: Логос, 2004. – 184 с.
5. Семакин И.Г. Информационные системы и модели [Текст] / И.Г. Семакин, Е.К. Хеннер. – М.: Бином, 2005. – 303 с.
6. Бережная, Е.В. Математические методы моделирования экономических систем [Текст] / Е.В. Бережная, В.И. Бережной. – М.: Финансы и статистика, 2005. – 432 с.
7. Гмурман, В.Е. Введение в теорию вероятностей и математическую статистику [Текст] / В.Е. Гмурман. – М.: Наука, 1997. – 380 с.
8. Ясницкий, Л.Н. Введение в искусственный интеллект [Текст] / Л.Н. Ясницкий. – М.: Академия, 2005. – 176 с