

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

В.А. РОМАНЕНКО

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ СИСТЕМ ВОЗДУШНОГО ТРАНСПОРТА

Рекомендовано редакционно-издательским советом федерального государственного автономного образовательного учреждения высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева» в качестве учебного пособия для студентов, обучающихся по основной образовательной программе высшего образования по направлению подготовки 24.04.04 Авиационное

© Самарский университет, 2018

ISBN 978-5-7883-1287-3

САМАРА
Издательство Самарского университета
2018

УДК 004.9 (075) + 629.7 (075)

ББК 32.973я7 + 68.53я7

Р 691

Рецензенты: д-р экон. наук, проф. Гераськин М.И.,
д-р экон. наук., проф. Хайтбаев В.А.

Романенко, Владимир Алексеевич

Р691 Компьютерное моделирование систем воздушного транспорта: учеб. пособие / *В.А. Романенко.* – Электрон. текст. дан. (1,79 Мб). – Самара: Издательство Самарского университета, 2018. – 1 опт. компакт-диск (CD-ROM). – Систем. требования: ПК Pentium, Adobe Acrobat Reader. – Загл. с титул. экрана.

ISBN 978-5-7883-1287-3

В электронном учебном пособии рассмотрены вопросы имитационного компьютерного моделирования систем воздушного транспорта. Изложены теоретические основы математического и компьютерного моделирования. Описана общецелевая система моделирования GPSS World. Пособие содержит сборник заданий на лабораторные работы по моделированию систем воздушного транспорта.

Содержание пособия отвечает требованиям рабочей программы по дисциплине «Компьютерное моделирование систем воздушного транспорта», которая изучается студентами, обучающимися в магистратуре по направлению подготовки 24.04.04 Авиастроение (магистерская программа «Авиастроение и авиаперевозки»).

Пособие подготовлено на кафедре организации и управления перевозками на транспорте Самарского университета.

УДК 004.9 (075) + 629.7 (075)

ББК 32.973я7 + 68.53я

Редактор М.С. Сараева
Компьютерная верстка М.С. Сараевой

Подписано для тиражирования 12.11.2018.

Объем издания 1,79 Мб.

Количество носителей 1 диск.

Тираж 10 экз.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С. П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

443086, Самара, Московское шоссе, 34.

Изд-во Самарского университета.
443086, Самара, Московское шоссе, 34.

Оглавление

ВВЕДЕНИЕ	6
1 ЗАДАЧИ МОДЕЛИРОВАНИЯ СИСТЕМ	9
1.1 Понятие модели и моделирования. Классификация моделей.....	9
1.2 Задачи исследования систем.....	10
1.3 Основные этапы и уровни моделирования.....	12
2 МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ.....	14
2.1 Математические модели.....	14
2.2 Критерии адекватности моделей.....	16
2.3 Классификация математических моделей.....	19
2.4 Сравнительный анализ аналитических и имитационных моделей.....	21
2.5 Модельное время.....	23
2.6 Способы получения случайных чисел.....	24
2.7 Сущность и цели планирования эксперимента	26
2.8 Стратегическое планирование эксперимента.....	27
2.9 Тактическое планирование имитационного эксперимента	33
2.10 Программные средства имитационного моделирования..	38
3 СИСТЕМА ИМИТАЦИОННОГО КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ GPSS WORLD.....	41
3.1 Общее описание системы имитационного моделирования GPSS World.....	41
3.2 Структура модели на языке GPSS.....	47

3.3 Объекты вычислительной и запоминающей категории.....	48
3.4 Организация поступления транзактов в модель и удаления транзактов из нее	52
3.5 Изменение значений параметров транзактов.....	56
3.6 Занятие и освобождение одноканального устройства	58
3.7 Имитация обслуживания посредством задержки во времени.....	58
3.8 Регистратор очереди.....	60
3.9 Статистические таблицы	62
3.10 Построение моделей систем с многоканальными устройствами	64
3.11 Управление продвижением транзактов.....	65
3.12 Работа с копиями транзактов.....	67
3.13 Функции	69
3.14 Язык PLUS	72
4 ЗАДАЧИ МОДЕЛИРОВАНИЯ СИСТЕМ ВОЗДУШНОГО ТРАНСПОРТА. СБОРНИК ЗАДАНИЙ НА ЛАБОРАТОРНЫЕ РАБОТЫ.....	77
4.1 Задание на лабораторную работу и пример её выполнения.....	77
4.2 Задания на лабораторные работы	86
СПИСОК ЛИТЕРАТУРЫ.....	98

ВВЕДЕНИЕ

Имитационное компьютерное моделирование является мощным универсальным инструментом исследования и оценки эффективности сложных систем, поведение которых зависит от воздействия случайных факторов. Методы имитационного моделирования находят широкое применение для решения задач анализа и оптимизации технических и организационно-технических систем воздушного транспорта, ярким примером которых является аэропорт, представляющий собой совокупность большого числа взаимосвязанных между собой служб и комплексов, действующих в условиях влияния разнообразных стохастических факторов и объединенных выполнением общей цели, в качестве которой выступает достижение оптимальных коммерческих показателей при обеспечении необходимых безопасности и регулярности перевозок. Настоящее пособие посвящено описанию методов построения имитационных моделей, имеющих непосредственное отношение к авиатранспортным системам, с применением инструментальных средств современной мощной системы моделирования GPSS World.

Первый раздел пособия посвящен общим вопросам имитационного моделирования. Даны определения понятиям модели и моделирования. Приведена классификация моделей по различным признакам. Рассмотрены задачи исследования систем, основные этапы и уровни моделирования.

Во втором разделе изложены теоретические основы математического и компьютерного моделирования. Проведен сравнительный анализ аналитических и имитационных моделей. Перечислены наиболее широко применяемые критерии адекватности моделей. Рассмотрена классификация математических моделей. Описаны подходы к решению задач, неизбежно встающих перед исследователем, применяющим методы имитационного моделирования: получение случайных чисел, стратегическое и тактиче-

ское планирование имитационного эксперимента, установление связи между «реальным» и модельным временем. Дана краткая характеристика современных программных средств имитационного моделирования, как широко используемых, так и менее известных.

В третьем разделе описана общецелевая система моделирования GPSS World, позволяющая моделировать как дискретные, так и непрерывные процессы и обладающая повышенной гибкостью, благодаря включению в ее состав алгоритмического языка Plus. Возможности системы позволяют автоматизировать весь цикл исследований от разработки модели до выработки рекомендаций за счет новых функций планирования экспериментов и обработки статистики. Система GPSS World максимально ориентирована на использование современных компьютерных технологий, обеспечивающих высокую интерактивность и визуальное представление информации. В пособии рассмотрены состав системы моделирования GPSS World и назначение ее основных компонентов, структура реализованной в системе модели. Детально изложены вопросы использования важнейших блоков, составляющих имитационную модель. Рассмотрены задачи построения моделей систем с одно- и многоканальными обслуживающими устройствами, использования в моделях статистических таблиц, функций и Plus-операторов. Описание сопровождается примерами.

Четвертый раздел пособия представляет собой сборник заданий на лабораторные работы по моделированию систем воздушного транспорта, в частности аэропортовых систем наземного обслуживания и обеспечения воздушных перевозок. Сборник включает 10 заданий. Задание первой лабораторной работы сопровождается подробным руководством и примером выполнения. Для остальных работ, наряду с заданием, приводятся краткие пояснения, достаточные для самостоятельного выполнения работы студентами с помощью материалов настоящего пособия.

Содержание пособия соответствует требованиям рабочей программы по дисциплине «Компьютерное моделирование систем воздушного транспорта», которая изучается студентами,

обучающимися в магистратуре по направлениям подготовки 24.04.04 «Авиастроение» (магистерская программа «Авиастроение и авиаперевозки») и 25.04.02 «Техническая эксплуатация авиационных электросистем и пилотажно-навигационных комплексов» (магистерская программа «Моделирование и оптимизация процессов эксплуатации систем и комплексов авиационной техники»).

1 ЗАДАЧИ МОДЕЛИРОВАНИЯ СИСТЕМ

1.1 Понятие модели и моделирования.

Классификация моделей

Моделирование – метод исследования систем, основанный на том, что рассматриваемая система заменяется на модель и проводится исследование модели с целью получения информации об изучаемой системе.

Модель исследуемой системы – некоторая другая система, которая ведет себя с точки зрения целей исследования аналогично исследуемой системе.

При моделировании реальная система заменяется некоторой упрощенной копией или схемой, которая отражает существенные черты исследуемой системы, позволяет разобраться в механизме ее работы, дает возможность предсказать ее изменение. Одной и той же системе могут соответствовать разные модели.

Классификация моделей (по форме отображения исследуемых систем):

1. материальные
 - 1.1. физические
 - 1.2. химические
 - 1.3. биологические
 - 1.4. аналоговые
2. идеальные
 - 2.1. знаковые
 - 2.1.1. графические (схематические)
 - 2.1.2. логико-описательные
 - 2.1.3. математические
 - 2.1.4. компьютерные
 - 2.2. мысленные.

Физическое моделирование предполагает использование в качестве модели либо самой исследуемой системы, либо другой системы с той же или подобной физической природой (макет-

ный или опытный образец). Пример – продувка моделей самолетов в аэродинамических трубах.

Аналоговая модель – это модель, представляющая исследуемую систему аналогом, который ведет себя так же, как реальная система, но не выглядит таковой. Пример может служить изучение механических колебаний с помощью электрической схемы, описываемой теми же дифференциальными уравнениями.

Математическое моделирование – процесс установления соответствия реальной системы некоторой математической модели и исследование этой модели, позволяющее получить характеристики реальной системы. Математическая модель – приближенное описание реального процесса, выраженное с помощью математических соотношений.

Компьютерное моделирование предполагает использование математической модели системы, представленной в виде программы на ЭВМ или компьютерной модели, позволяющей проводить с ней вычислительные эксперименты.

1.2 Задачи исследования систем

Система (S) – совокупность элементов со связями и целью функционирования, отличной от целей функционирования составляющих ее элементов.

Элемент (s) – некоторый объект (материальный, энергетический, информационный), обладающий рядом свойств, обеспечивающих выполнение некоторых функций, внутреннее строение которого для целей исследования не представляет интереса.

Связь (l) между элементами – процесс их взаимодействия, важный для целей исследования.

Формализованное обозначение системы:

$$S = \{\{s\}, \{l\}, F\},$$

где $\{s\}$ – множество априорно выделенных элементов системы; $\{l\}$ – комплекс как взаимных связей между элементами, так и связей между элементами и внешней средой; F – функция, описывающая цель функционирования системы.

Внешняя среда – все, что оказывает воздействие на систему, и на что система оказывает воздействие.

Структура системы – ее расчленение (декомпозиция) на элементы или группы элементов с указанием связей между ними, неизменное в течение времени исследования и дающее точное представление о системе.

Различают структуру с равноправно входящими в нее элементами и иерархическую структуру.

Иерархией называется структура с наличием подчиненности одних элементов другим, когда воздействия в одном из направлений оказывают гораздо большее влияние на элемент, чем в другом.

Состояние системы – множество характеристик элементов системы, изменяющихся во времени и важных для целей ее функционирования.

Процесс (динамической) системы – множество значений состояний системы, изменяющихся во времени.

Цель функционирования системы – задача получения желаемого состояния системы.

Определение цели системы влечет:

1) необходимость постановки локальных целей для ее элементов;

2) целенаправленное вмешательство в процесс функционирования системы, называемое управлением.

Классификация систем по различным признакам:

1) По числу состояний:

– динамические – системы, состояние которых меняется с изменением времени;

– статические.

2) По типу состояний:

– дискретные – системы, состояние которых, т. е. значение ее атрибутов, изменяется скачкообразно, в дискретные моменты времени (человек вошел/вышел в магазин);

– непрерывные – системы, состояние которых изменяется непрерывно (температура закипающего чайника);

– комбинированные (непрерывно-дискретные) – если часть атрибутов, описывающих состояние системы, меняется непре-

рывно, а часть дискретно (например, поведение капли из крана – «висит», «летит», «упала»).

3) по условиям переходов из одного состояния в другое:

– стохастические (вероятностные) – если при одних и тех же начальных условиях результаты функционирования системы будут различаться;

– детерминированные;

4) по виду элементов:

– системы типа «объект» – элементами системы являются предметы;

– системы типа «процесс» – элементы системы – операции.

Классы задач исследования систем:

1) задачи анализа, направленные на изучение свойств функционирования системы в зависимости от ее структуры;

2) задачи синтеза, связанные с выбором структуры и значений параметров по заданным свойствам системы.

Во многих практических исследованиях единственно возможным способом их решения оказывается имитационное моделирование процессов функционирования системы на ЭВМ.

1.3 Основные этапы и уровни моделирования

Этапы моделирования и их сущность:

1. Постановка задачи. Определение цели анализа и пути его достижения и выработки общего подхода к исследуемой проблеме. Необходимо глубокое понимание существа поставленной задачи.

2. Изучение теоретических основ и сбор информации об объекте оригинала. Подбирается или разрабатывается подходящая теория. Устанавливаются причинно-следственные связи между переменными, описывающими систему. Определяются входные и выходные данные, принимаются упрощающие предположения.

3. Формализация. Выбор системы условных обозначений, запись с их помощью отношений между составляющими исследуемой системы в виде математических выражений.

4. Выбор метода решения. Выбирается имеющийся или разрабатывается специальный метод.

5. Реализация модели. Разработка алгоритма, написание, отладка, тестирование программы, получение решения задачи.

6. Анализ полученной информации. Сопоставляется полученное и предполагаемое решение, проводится контроль погрешности моделирования.

7. Проверка адекватности реальному объекту. Результаты, полученные по модели сопоставляются с имеющейся о системе информацией.

Процесс моделирования является итеративным. В случае неудовлетворительных результатов этапов 6 или 7 осуществляется возврат к одному из ранних этапов до тех пор, пока не будут получены приемлемые результаты.

Уровни моделирования:

1. **Метауровень** характеризуется наименьшей детализацией объекта. На этом уровне моделируются, например, развитие вселенной, глобальные и локальные вычислительные системы, энергосистемы, транспортные системы, экономические системы и объекты. В качестве математического аппарата используется теория вероятностей, математическая статистика и теория массового обслуживания.

2. **Макроуровень** используется для исследования физических процессов в непрерывном времени на дискретном пространстве. При этом считается, что параметры элементов моделируемых систем сосредоточены в одной точке.

3. **Микроуровень** – объекты рассматриваются в непрерывном времени и непрерывном пространстве. В качестве математического аппарата часто используются дифференциальные уравнения в частных производных с заданными краевыми условиями. Результатами моделирования на микроуровне является поля напряжений, деформация в деталях механических конструкций, электромагнитные поля в электропроводных средах, поля температур нагретых деталей и т.п.

2 МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

2.1 Математические модели

Поведение системы S в рамках рассматриваемой модели исследуется на заданном временном промежутке, обозначенном $T_U = [t_0, T]$, – временном интервале моделирования системы S (интервале модельного времени), где t_0 – время начала моделирования (обычно $t_0 = 0$); T – время окончания моделирования; $t \in T_U$ – текущее значение модельного времени;

Для описания системы S вводится **вектор параметров системы** $\vec{\theta} = (\theta_1, \theta_2, \dots, \theta_m)$, т.е. характеристик системы, остающихся постоянными на всем интервале моделирования T_U . Пример параметров системы массового обслуживания (СМО) – число каналов (если оно остается неизменным в течение T_U).

Характеристики системы, величины которых зависят от времени, называют **переменными величинами**. Множество переменных разбивают на два подмножества – зависимых и независимых переменных.

К **независимым переменным** относят следующие характеристики:

– вектор входных воздействий на систему $\vec{v} = (\vec{u}, \vec{w})$, включающий вектор управляющих воздействий

$\vec{u} = \vec{u}(t) = (u_1(t), \dots, u_{n_u}(t))$ и вектор остальных (неуправляющих) воздействий $\vec{w} = \vec{w}(t) = (w_1(t), \dots, w_{n_w}(t))$, где $t \in T_U$;

– вектор воздействий внешней среды (возмущений)

$\vec{\xi} = \vec{\xi}(t) = (\xi_1(t), \dots, \xi_{n_{\xi}}(t))$, $t \in T_U$, среди которых могут быть

наблюдаемые и ненаблюдаемые, детерминированные и случайные воздействия. Примеры: наличие отклонений времени прилета ВС в аэропорт от расписания – наблюдаемые; износ техни-

ки, который может привести к ее выходу из строя – ненаблюдаемые воздействия.

– переменные $\vec{x} = \vec{x}(t) = (x_1(t), \dots, x_{n_x}(t)) \in X$, характеризующие состояние системы, где X – пространство состояний или фазовое пространство системы (множество возможных значений вектора \vec{x}). Если $t_1 < t_2 < \dots$ – моменты изменения состояния системы, то последовательность $\vec{x}_1(t), \vec{x}_2(t), \dots$ называется фазовой траекторией системы.

К **зависимым переменным** относят характеристики:

– вектор выходных характеристик системы $\vec{y} = \vec{y}(t) = (y_1(t), \dots, y_{n_y}(t))$, $t \in T_U$, включающий переменные, которые могут измеряться в процессе работы реальной системы;

– вектор показателей эффективности и/или качества системы $\vec{z} = (z_1, \dots, z_k)$. Показатель эффективности характеризует процесс и эффект от функционирования системы. Показатели качества – пригодность системы для использования её по назначению.

Связи между зависимыми и независимыми переменными выражаются схемой, представленной на рисунке 1.

Если в состав одного или нескольких векторов из множества $\{\vec{\theta}, \vec{u}, \vec{w}, \vec{\xi}\}$ входят случайные величины, то значения z_1, \dots, z_k также будут случайными и поэтому не могут служить показателями. В этом случае в качестве показателей могут использоваться например их средние значения:

$$\bar{z}_i = M[z_i], \quad i = 1, \dots, k.$$

Обозначим $a^{[t_0, T]}$ – отрезок функции $a(t)$ на интервале T_U , т.е. множество пар $\langle t, a(t) \rangle$, заданных для любого $t \in T_U$.

Процесс функционирования системы во времени описывается **операторными соотношениями** (заданными аналитически или алгоритмически) для состояний, выходных характеристик и показателей:

$$\begin{aligned} \vec{x}(t) &= \vec{F}_x(\vec{x}^0, \vec{\theta}, \vec{u}^{[t_0, T]}, \vec{w}^{[t_0, T]}, \vec{\xi}^{[t_0, T]}, t), \\ \vec{y}(t) &= \vec{F}_y(\vec{x}^0, \vec{\theta}, \vec{x}^{[t_0, T]}, \vec{u}^{[t_0, T]}, \vec{w}^{[t_0, T]}, \vec{\xi}^{[t_0, T]}, t), \end{aligned} \quad (1)$$

$$\vec{z} = \vec{F}_z(\vec{x}^0, \vec{\theta}, \vec{x}^{[t_0, T]}, \vec{y}^{[t_0, T]}, \vec{u}^{[t_0, T]}, \vec{w}^{[t_0, T]}, \vec{\xi}^{[t_0, T]}),$$

где $\vec{F}_x(\cdot), \vec{F}_y(\cdot), \vec{F}_z(\cdot)$ – некоторые операторы, $\vec{x}^0 = \vec{x}(t_0)$ – начальное состояние системы. Зависимости (1) – законы функционирования системы S . Зависимость $\vec{y} = \vec{y}(t)$, $t \in T_U$ – выходная траектория системы. Зависимость $\vec{x} = \vec{x}(t)$, $t \in T_U$ – фазовая траектория системы.

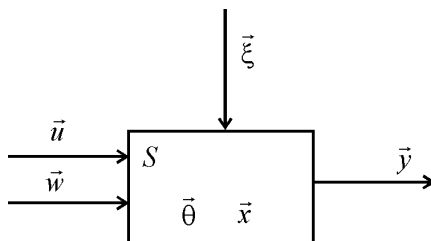


Рис. 1. Связи между зависимыми и независимыми переменными

Переменные $x_1(t), \dots, x_{n_x}(t)$ как правило выбираются так, чтобы можно было упростить закон (1) и привести два последних выражения (1) к виду

$$\begin{aligned} \vec{y}(t) &= \vec{G}_y(\vec{x}^{[t_0, T]}, t), \\ \vec{z} &= \vec{G}_z(\vec{y}^{[t_0, T]}). \end{aligned} \quad (2)$$

Математическая модель системы включает множество $\{\vec{\theta}, \vec{u}, \vec{w}, \vec{\xi}, \vec{x}, \vec{y}\}$ вместе с законом функционирования (1) или (2).

2.2 Критерии адекватности моделей

Адекватность модели – соответствие модели моделируемому объекту или процессу. Адекватность показывает, насколько удачной получилась математическая модель. Адекватность – в какой-то мере условное понятие, так как полного соответствия модели реальному объекту быть не может, иначе это была бы не

модель, а сам объект. При моделировании имеется в виду адекватность не вообще, а по тем свойствам модели, которые для исследования считаются существенными.

Модель считается адекватной, если отражает заданные свойства с приемлемой точностью. Точность определяется как степень совпадения значений выходных характеристик модели и объекта. Основной причиной рассогласования указанных характеристик является неизбежное искусственное ухудшение адекватности математической модели в процессе её формирования путём исключения тех или иных элементов связей, пренебрежением малозначимыми факторами.

Точность модели различна в разных условиях функционирования объекта. Эти условия характеризуются входными воздействиями и возмущениями. В пространстве входных воздействий и возмущений необходимо выделить область адекватности модели, где погрешность будет меньше заданной предельно допустимой погрешности. Определение области адекватности моделей – сложная процедура, требующая больших вычислительных затрат.

Точность математической модели оценивается степенью совпадения значений выходных характеристик реального объекта и значений тех же характеристик, рассчитанных с помощью модели.

Адекватность математической модели – это ее способность отражать заданные свойства объекта с погрешностью, не выше заданной.

Простейшей мерой адекватности может служить отклонение расчётной характеристики y от эталонной y^* :

$$\Delta y = |y^* - y| \quad \text{или} \quad \Delta y = |y^* - y| / y^*$$

Тогда можно считать, что модель адекватна системе, если вероятность того, что отклонение Δy не превышает предельной величины Δ , больше допустимой вероятности P_Δ :

$$P(\Delta y < \Delta) \geq P_\Delta.$$

Другие примеры критериев адекватности, суть которых иллюстрирует рисунок 2, перечислены ниже:

– максимальное абсолютное отклонение (рис.2а):

$$\max \Delta y_i = \max |y_i^* - y_i|, \quad i = 1, 2, \dots, k, \dots, n.$$

– сумма абсолютных отклонений (рис.2б):

$$\sum_{i=1}^n \Delta y_i = \sum_{i=1}^n |y_i^* - y_i|.$$

– сумма квадратов разности (рис.2в):

$$\sum_{i=1}^n \Delta y_i^2 = \sum_{i=1}^n (y_i^* - y_i)^2.$$

– интегральная оценка (рис.2г):

$$\int_{t_0}^T \Delta y(t) dt = \int_{t_0}^T |y^*(t) - y(t)| dt.$$

В различных ситуациях предпочтение отдаётся разным критериям.

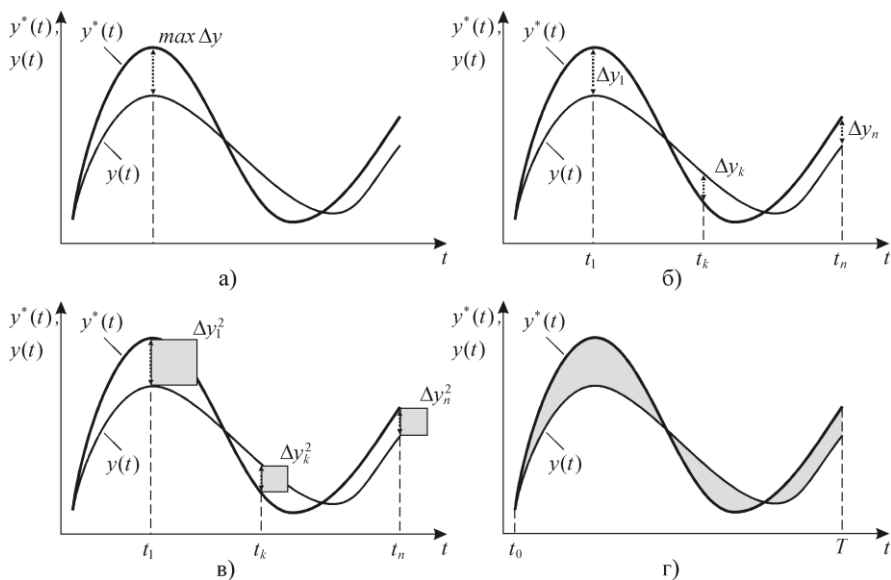


Рис. 2. Критерии адекватности моделей

2.3 Классификация математических моделей

Классификация математических моделей по следующим признакам.

А. Методы компьютерной реализации модели (методы задания закона (1) или (2)):

1. Аналитические – решения получаются в абстрактном виде, т.е. подстановка чисел вместо символов производится уже после получения решения.

1.1. Алгебраические – модели, в которых используется конечное число арифметических операций, операций возведения в целочисленную степень и извлечения корня.

1.2. Приближенные – модели, содержащие формулы с показательными, логарифмическими, тригонометрическими и т.п. функциями, предполагающими приближен-

ные вычисления на ЭВМ путем разложения в бесконечный ряд.

2. Алгоритмические – используются для изучения сложных систем.

2.1. Численные – модель упрощается, например, переходом от системы дифференциальных уравнений к системе линейных уравнений.

2.2. Имитационные – математические соотношения для исследуемой системы вообще не записываются, а заменяются некоторым алгоритмом, моделирующим ее поведение.

Б. Наличие возможности управлять исследуемыми моделями (и системами):

1. Управляемые – предоставляют исследователю возможность выбора не только варианта функционирования, но и поставленных целей (например, максимизация прибыли, минимизация затрат и т.п.).

1.1. Оптимизационные – поиск управления, обеспечивающего наилучшие (оптимальные) значения выбранного показателя (критерия) при заданных ограничениях.

1.2. Неоптимизационные – поиск управления, переводящего систему из неустойчивого положения в устойчивое.

2. Наблюдаемые – позволяют исследователю только фиксировать и анализировать состояние системы, но не влиять на него.

В. Наличие возможности учета случайных факторов:

1. Детерминированные – модели, в которых установлено взаимно-однозначное соответствие между переменными, описывающими систему.

2. Стохастические – модели, в которых связь между переменными носит случайный характер.

Г. Характер режимов:

1. Статические – модели, включающие описание связей между основными переменными исследуемой системы в установившемся режиме без учета изменения параметров во времени.

2. Динамические – модели, описывающие связи между основными переменными исследуемой системы при переходе от одного режима к другому.

Д. Способ задания модельного времени t :

1. Дискретные – модели, в которых время t рассматривается как дискретная величина:

$$t = i \cdot \Delta t, \quad i = 0, 1, \dots, M, \quad M = \lceil \Delta T / \Delta t \rceil,$$

где ΔT – продолжительность интервала моделирования T_U , Δt – шаг дискретизации, M – число шагов.

2. Непрерывные – модели, в которых время t рассматривается как непрерывная величина: $t \in T_U$.

2.4 Сравнительный анализ аналитических и имитационных моделей

Аналитическая модель (АМ) системы – совокупность функциональных соотношений или логических условий, описывающих связи между параметрами, переменными и показателями эффективности и качества системы.

Условия применения АМ:

- сравнительно простые системы;
- системы, получаемые в результате упрощения реальных систем с целью изучения их некоторых свойств.

Достоинства АМ: универсальность, высокая степень общности и значимости результатов. Недостатки АМ: чувствительность к степени сложности системы и неадекватность реальной системе.

Имитационная модель (ИМ) системы – машинная программа (или алгоритм), позволяющая имитировать на ЭВМ поведение отдельных элементов системы и связей между ними в течение заданного времени моделирования. ИМ к алгоритмическим моделям.

Эксперименты на ЭВМ с имитационной моделью называются имитационными (вычислительными) экспериментами. **Имитационные эксперименты** состоят из *наблюдений*, а наблюде-

ния – из **прогонов** (реализаций) модели. Под наблюдением понимается серия прогонов модели при одинаковых исходных данных.

Отличительные особенности ИМ:

- при создании ИМ сложной системы законы функционирования всей системы в целом могут быть неизвестными (известны только алгоритмы, описывающие поведение отдельных элементов системы и связей между ними);

- в ИМ связи между параметрами и характеристиками системы выявляются, а значения исследуемых характеристик определяются в ходе имитационного эксперимента на ЭВМ.

Условия применения ИМ:

- широкий класс систем практически любой сложности, аналитические модели которых частично или полностью не определены;

- в случаях, когда в силу сложности АМ ее практическое использование невозможно.

Достоинства ИМ:

- единственно возможный метод исследования сложной системы;

- возможность исследования системы на различных уровнях ее детализации;

- возможность исследования динамики взаимодействия элементов системы во времени и пространстве;

- возможность оценивания характеристик системы в определенные моменты времени.

Недостатки ИМ:

- дороговизна, часто разработка хорошей ИМ обходится дороже создания АМ и требует больших временных затрат;

- результаты имитационного моделирования обладают меньшей степенью общности по сравнению с АМ и не позволяют выявить общие закономерности функционирования классов систем;

- не существует надежных методов оценки адекватности ИМ.

2.5 Модельное время

При разработке компьютерных имитационных моделей сложных систем необходимо учитывать следующие особенности функционирования ЭВМ:

1) сложная система состоит из многих элементов, которые функционируют одновременно, но в большинстве ЭВМ параллельное выполнение нескольких программ, имитирующих поведение отдельных элементов системы, невозможно;

2) поскольку имитационные модели – это программы для ЭВМ, то они должны оперировать с конечным множеством данных и имитировать поведение системы не во все моменты времени $t \in [t_0, T]$, а лишь в некоторые, составляющие конечное множество $T_H \subset [t_0, T]$.

Чтобы обеспечить имитацию параллельных или одновременных событий системы на конечном множестве моментов времени T_H , в имитационной модели используется специальная переменная t – модельное время (МВ).

Кроме МВ, при имитационном моделировании используется t_P – реальное (физическое) время системы, функционирование которой имитируется, и $t_Э$ – машинное (процессорное) время имитации, отражающее затраты ресурсов времени ЭВМ на организацию имитационного моделирования.

Соотношение физического и модельного времени определяется спецификой моделируемой системы и задается коэффициентом K . K – это диапазон физического времени, принимаемого за единицу модельного времени. Таким образом значению физического времени t_P соответствует значение модельного времени t , вычисляемое как

$$t = \frac{t_P - t_0}{K},$$

где $t_0 \leq t_P \leq T$, t_0 , t_T – начальное и конечное физическое время моделирования. При выполнении имитационной модели выполняется продвижение модельного времени от 0 до $\frac{T - t_0}{K}$ единиц.

Существует 2 способа формирования конечного множества моментов времени T_H :

– **принцип** « Δt » заключается в изменении МВ с фиксированным шагом Δt . Событиям, которые попадают в один и тот же временной промежуток с некоторым номером r , присваивается один и тот же момент наступления $t = r \cdot \Delta t$. Фазовая траектория системы $\vec{x} = \vec{x}(t)$ на множестве $T_H \subset [t_0, T]$ искажена. Выбор величины Δt влияет на быстродействие и точность модели. Недостатки метода: медленная реализация, несовпадение модельных и реальных моментов наступления событий;

– **принцип** « Δx » состоит в изменении МВ при скачкообразном изменении вектора состояния x системы на некоторую величину $\Delta x (\Delta x \neq 0)$. Изменение МВ происходит в моменты наступления событий. Фазовая траектория $\vec{x} = \vec{x}(t)$ на множестве $T_H \subset [t_0, T]$, полученная с помощью имитационной модели, совпадает с траекторией системы. Модельное время t зависит от числа событий, происходящих в модели.

На практике предпочтение отдается принципу « Δx ».

2.6 Способы получения случайных чисел

Проведение имитационного моделирования требует наличия простых и экономных способов формирования случайных чисел (СЧ).

СЧ являются реализациями последовательности взаимно независимых и одинаково распределенных СВ.

Способы получения СВ:

1. Специальные таблицы СЧ.
2. Физические генераторы СВ (источники радиоактивного излучения, шумы радиоэлектронных приборов).
3. Программные генераторы (или датчики) псевдослучайных чисел.

Наиболее удобна последовательность СЧ с равномерным распределением в интервале $[0, 1]$, позволяющая получать СВ с

любым законом распределения с помощью метода обратной функции.

Метод обратной функции. Пусть $F(z)$ – заданная функция распределения СВ Z , U – СВ, равномерно распределенная на интервале $[0, 1]$. Разыграем на оси ординат точку U_1 . Используя функцию $F(z)$ получим искомое значение величины z , такое, что $F(z) = U_1$, т.е. $z = F^{-1}(U_1)$.

Трудность использования метода обратной функции связана с необходимостью обращения функции распределения.

Пример. Моделирование показательного распределения СВ с параметром λ , для которого

$$F(x) = 1 - e^{-\lambda x}, \quad \lambda > 0, \quad 0 \leq x < \infty.$$

Откуда $U = 1 - e^{-\lambda x}$ и, согласно методу обратной функции:

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

Так как U – равномерно распределенная СВ, то очевидно СВ $(1 - U)$ так же распределена равномерно и, значит, можно сделать замену $(1 - U)$ на U . Таким образом расчетное выражение принимает вид:

$$x = -\frac{1}{\lambda} \ln U.$$

Итак, если имеется ряд реализаций равномерно распределенной на интервале $[0, 1]$ СВ U , то получаемые с использованием последней формулы значения x будут подчинены показательному распределению с параметром λ .

Принцип метода обратной функции поясняет рисунок 3.

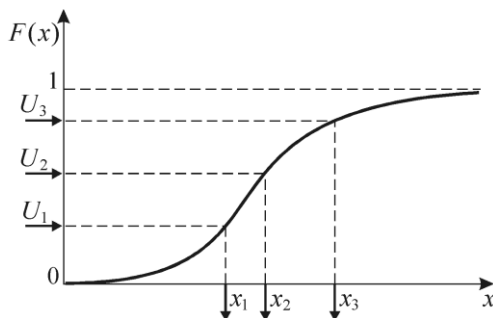


Рис. 3. Схема метода обратной функции

2.7 Сущность и цели планирования эксперимента

Инструментом анализа и синтеза сложных систем является модель (натурная, математическая, компьютерная и др.). Модель создается для проведения с ней экспериментов (физических, вычислительных, компьютерных и др.). Эксперимент должен давать нужную информацию при приемлемых временных, материальных, финансовых и пр. затратах. Для успешного исследования системы при приемлемых затратах необходимо решить задачу оптимального планирования эксперимента.

План компьютерного эксперимента определяет:

- объем вычислений на компьютере;
- порядок проведения вычислений на компьютере;
- способы накопления и статистической обработки результатов моделирования.

Цели планирования экспериментов:

- сокращение времени моделирования при соблюдении требований к точности и достоверности результатов;
- увеличение информативности каждого наблюдения;
- создание структурной основы процесса исследования.

Действия по планированию эксперимента разделяют на две функциональные части:

1) стратегическое планирование, включающее разработку условий проведения эксперимента; определение режимов, обеспечивающих наибольшую информативность эксперимента;

2) тактическое планирование – комплекс действий, обеспечению заданных точности и достоверности результатов.

2.8 Стратегическое планирование эксперимента

Предположим, что исследователь может наблюдать входы и выходы имитационной модели как «черного ящика» и по результатам наблюдений определять зависимость между входами и выходами.

Входные переменные $x_i, i = 1, \dots, n$, называются **факторами**. Выходные переменные $y_j, j = 1, \dots, m$, называются **наблюдаемыми переменными (реакциями, откликами)**. Ниже рассмотрены модели с одним откликом y .

Факторное пространство – это множество факторов модели, значения которых исследователь может контролировать в ходе подготовки и проведения эксперимента.

Каждый фактор имеет уровни. **Уровни** – это значения, которые устанавливаются для каждого фактора при определении условий прогона модели в наблюдении (см. п. 2.4).

Целью эксперимента является нахождение функции y , при этом предполагается, что значение отклика складывается из двух составляющих:

$$y = f(x_1, x_2, \dots, x_n) + \xi(x_1, x_2, \dots, x_n),$$

где $f(x_1, x_2, \dots, x_n)$ – функция отклика (неслучайная функция факторов); $\xi(x_1, x_2, \dots, x_n)$ – ошибка эксперимента (случайная величина); x_1, x_2, \dots, x_n – определенное сочетание уровней факторов из факторного пространства.

В результате стратегического планирования эксперимента решаются следующие проблемы:

Проблема 1. Выбор отклика (реакции), то есть определение, какие величины нужно измерять во время эксперимента, чтобы получить искомые ответы.

Проблема 2. Выбор (определение) существенных факторов и их сочетаний, влияющих на работу моделируемого объекта. Факторами могут быть интенсивность потока заявок (ВС, пассажиры, багажа и пр.), надежность работы аппаратуры, интенсивность обслуживания заявок, ритмичность поставок комплектующих и многое другое.

Виды факторов:

- количественные, значениями которых являются числа (интенсивности входных потоков и потоков обслуживания, емкость буфера, число каналов в СМО, доля брака при изготовлении деталей и др.);

- качественные факторы (дисциплины обслуживания (LIFO, FIFO и др.) в СМО, квалификация персонала и т. п.).

В стратегических планах факторы обозначают x_i , где i – номер (тип) фактора.

Проблема 3. Выбор значений каждого фактора – уровней фактора. Желательно выбирать минимальное достаточное число уровней. Например, если в качестве одного из факторов выступает время обслуживания, то уровнями могут быть: 1 мин., 2 мин, 3 мин.

Обычно факторы рассматривают в стандартизованном или натуральном масштабе изменения переменных. Переход к стандартизованному масштабу:

$$z_i = \frac{x_i - \frac{1}{2}(x_i^{max} + x_i^{min})}{\frac{1}{2}(x_i^{max} - x_i^{min})}, \quad (3)$$

где x_i – значение i -й переменной в натуральном масштабе. Согласно (3) в стандартизованном масштабе факторы принимают значения $-1, 0, +1$.

Центр эксперимента – точка факторного пространства с координатами $\frac{1}{2}(x_i^{max} + x_i^{min})$ (середина интервала), $i = 1, 2, \dots, n$.

Симметричный план – план с одинаковым числом уровней для всех факторов.

Ортогональный план – план с равноотстоящими друг от друга уровнями факторов.

Полный факторный экспериментом (ПФЭ) – эксперимент, в котором реализуются все сочетания уровней всех факторов. План ПФЭ предельно информативен, но может требовать неприемлемых затрат ресурсов. ПФЭ применяют, когда число факторов невелико.

Число измерений откликов (реакций) модели при ПФЭ:

$$N_C = q_1 \cdot q_2 \cdot \dots \cdot q_n,$$

где q_i – число уровней i -го фактора, $x_i, i = 1, \dots, n$; n – число факторов эксперимента.

Пусть число прогонов имитационной модели в каждом наблюдении p . Тогда для симметричного ПФЭ общее число необходимых прогонов модели равно:

$$N = pq^n.$$

Пример. Планируется провести компьютерный эксперимент, в котором на отклик модели влияют три фактора. Для каждого фактора установлены три уровня. Требования по точности требуют 6000 прогонов модели на каждом уровне (для каждого наблюдения). Время одного прогона модели равно 2 с. В этом случае затраты времени на проведение компьютерного эксперимента составят 810 часов (!).

В ряде случаев на основе имеющейся имитационной модели процесса требуется получить т.н. «вторичную модель» в виде аналитической зависимости. В таких случаях «вторичная модель» формируется по результатам эксперимента методом регрессионного анализа.

Примеры «вторичных моделей»:

1) модель, учитывающая два фактора:

$$y = b_0 + b_1x_1 + b_2x_2 + b_{12}x_1x_2; \quad (4)$$

2) модель, учитывающая три фактора:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_{12}x_1x_2 + b_{13}x_1x_3 + b_{23}x_2x_3 + b_{123}x_1x_2x_3. \quad (5)$$

Здесь b_0 – свободный член; b_1, b_2, b_3 – главные коэффициенты, соответствующие т.н. линейным эффектам; $b_{12}, b_{13}, b_{23}, b_{123}$ – коэффициенты, соответствующие эффектам взаимодействия факторов. Эффекты взаимодействия двух и более факторов проявляются, если влияние каждого из них на отклик зависит от уровней, на которых установлены другие факторы.

ПФЭ позволяет определять все перечисленные коэффициенты. Пусть, например, все факторы двухуровневые, тогда необходимое число измерений в ПФЭ для модели (4) составляет $N_C = 2^2 = 4$, для модели (5) – $N_C = 2^3 = 8$.

Рассмотрим формирование плана ПФЭ для построения модели (4).

Перейдем к факторам в стандартизованном масштабе – z_1, z_2 . Взаимосвязь факторов оценим, введя новый фактор $z_3 = z_1 \cdot z_2$ и соответствующий ему коэффициент b_3 . Таким образом «вторичная модель» преобразуется к виду:

$$y = b_0 + b_1 z_1 + b_2 z_2 + b_3 z_3. \quad (6)$$

План ПФЭ представим в форме таблицы 1.

Таблица 1. План ПФЭ

Номер опыта	z_1	z_2	z_3	Y
1	–1	–1	+1	Y_1
2	+1	–1	–1	Y_2
3	–1	+1	–1	Y_3
4	+1	+1	+1	Y_4

Обработка результатов данного ПФЭ позволяет получить коэффициенты зависимости (6).

Проведение ПФЭ при учете большого числа факторов может потребовать неприемлемых временных затрат. Сокращение временных затрат обеспечивается сокращением плана эксперимента, которое достигается пренебрежением эффектами парных, тройных и более взаимодействий факторов. Это снижает точ-

ность «вторичной модели», но во многих случаях допустимо. Например, модели (4) и (5) приводятся к линеаризованному виду: $y = b_0 + b_1x_1 + b_2x_2$ и $y = b_0 + b_1x_1 + b_2x_2 + b_3x_3$, соответственно.

Для определения коэффициентов «вторичной модели» по результатам эксперимента может быть использован матричный метод, аналогичный использованному при построении уравнения множественной линейной регрессии.

Пусть по результатам K экспериментов необходимо определить коэффициенты линейного уравнения «вторичной модели» с факторами в стандартизованном масштабе

$$y = b_0 + b_1z_1 + \dots + b_nz_n,$$

где n – число факторов.

Уравнение представим в матричной форме:

$$\vec{y} = Z\vec{b}, \quad (7)$$

где Z – матрица условий эксперимента, \vec{b} – вектор неизвестных коэффициентов регрессии, \vec{y} – вектор результатов наблюдений,

$$Z = \begin{bmatrix} z_{1,0} & z_{1,1} & \dots & z_{1n} \\ z_{2,0} & z_{2,2} & \dots & z_{2n} \\ \dots & \dots & \dots & \dots \\ z_{K,0} & z_{K,2} & \dots & z_{Kn} \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} b_0 \\ b_1 \\ \dots \\ b_n \end{bmatrix}, \quad \vec{y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \dots \\ Y_K \end{bmatrix}.$$

Решение матричного уравнения (7), как известно, имеет вид:

$$\vec{b} = (Z^T Z)^{-1} (Z^T \vec{y}). \quad (8)$$

Пример. Найти коэффициенты регрессии для линейной «вторичной модели» эксперимента с двумя факторами в стандартизованном масштабе.

Решение.

Уравнение регрессии в рассматриваемом случае имеет вид

$$y = b_0 + b_1z_1 + b_2z_2.$$

План факторного эксперимента представлен в таблице 2.

Таблица 2. План ПФЭ

Номер опыта	z_1	z_2	Y
1	+1	+1	Y_1
2	-1	+1	Y_2
3	+1	-1	Y_3
4	-1	-1	Y_4

Таким образом, матрица Z и векторы \vec{b} и \vec{y} запишутся как

$$Z = \begin{bmatrix} +1 & +1 & +1 \\ +1 & -1 & +1 \\ +1 & +1 & -1 \\ +1 & -1 & -1 \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix}, \quad \vec{y} = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix}.$$

Далее без комментариев приведены выкладки поэтапного вычисления в соответствии с (8).

$$Z^T = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \end{bmatrix},$$

$$Z^T Z = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \end{bmatrix} \begin{bmatrix} +1 & +1 & +1 \\ +1 & -1 & +1 \\ +1 & +1 & -1 \\ +1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix},$$

$$(Z^T Z)^{-1} = \begin{bmatrix} 1/4 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/4 \end{bmatrix},$$

$$Z^T \vec{y} = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} = \begin{bmatrix} Y_1 + Y_2 + Y_3 + Y_4 \\ Y_1 - Y_2 + Y_3 - Y_4 \\ Y_1 + Y_2 - Y_3 - Y_4 \end{bmatrix},$$

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 1/4 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/4 \end{bmatrix} \begin{bmatrix} Y_1 + Y_2 + Y_3 + Y_4 \\ Y_1 - Y_2 + Y_3 - Y_4 \\ Y_1 + Y_2 - Y_3 - Y_4 \end{bmatrix},$$

Результатом будут следующие выражения для коэффициентов регрессии:

$$b_0 = \frac{Y_1 + Y_2 + Y_3 + Y_4}{4}, \quad b_1 = \frac{Y_1 - Y_2 + Y_3 - Y_4}{4}, \quad b_2 = \frac{Y_1 + Y_2 - Y_3 - Y_4}{4}.$$

2.9 *Тактическое планирование имитационного эксперимента*

Целью тактического планирования имитационного эксперимента является определение такого количества прогонов имитационной модели (объема выборки), которое обеспечивает получение достоверного результата с заданной точностью.

Рассмотренные ниже методы достижения сформулированной цели базируются на использовании **центральной предельной теоремы**, гласящей в одной из формулировок, что, если $X_1, X_2, \dots, X_n, \dots$ – независимые, одинаково распределенные СВ, то при неограниченном увеличении n закон распределения суммы

$$Y_n = \sum_{k=1}^n X_k$$

неограниченно приближается к нормальному.

Следует помнить, что плотность $f(x)$ и функция $F(x)$ непрерывной нормально распределенной СВ X определяются по формулам

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

$$F(x) = P(X < x) = \int_{-\infty}^x f(x) dx = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(s-\mu)^2}{2\sigma^2}} ds, \quad (9)$$

где μ – параметр положения – математическое ожидание; σ – параметр масштаба – среднее квадратическое отклонение. Внутри интеграла вместо x использована другая буква, чтобы не возникало неясности с обозначениями, так как здесь каждому значению x ставится в соответствие несобственный инте-

грал $\frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(s-\mu)^2}{2\sigma^2}} ds$, который равен некоторому числу из интервала $(0;1)$.

Интеграл из выражения для $F(x)$ нельзя представить в виде элементарных функций, однако его можно свести к табулированной функции стандартного нормального распределения. Для этого используется метод замены переменных, согласно которому для вычисления некоторого интеграла $\int g(s)ds$ необходимо выполнить подстановку

$$s = \varphi(y),$$

где $\varphi(y)$ – функция, имеющая непрерывную производную. Тогда, поскольку $ds = \varphi'(y)dy$, интеграл принимает вид

$$\int g(s)ds = \int f[\varphi(y)]\varphi'(y)dy.$$

В случае определенного интервала замена переменных приводит к аналогичной формуле:

$$\int_{x_1}^{x_2} g(s)ds = \int_{y_1}^{y_2} g[\varphi(y)]\varphi'(y)dy,$$

где $x_1 = \varphi(y_1)$, $x_2 = \varphi(y_2)$ и, значит, $y_1 = \varphi^{-1}(x_1)$, $y_2 = \varphi^{-1}(x_2)$.

Применяя в отношении выражения для функции нормального распределения $F(x)$ замену

$$y = \frac{s - \mu}{\sigma}, \tag{10}$$

имеем $\frac{dy}{ds} = \frac{1}{\sigma}$, и, следовательно, $ds = \sigma dy$. При использован-

ной, согласно (10), замене нижняя граница интегрирования остается неизменной, а верхняя граница $t = x$ переходит в

$y = \frac{x - \mu}{\sigma}$. Таким образом окончательно получаем

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{x-\mu}{\sigma}} e^{-\frac{y^2}{2}} dy. \quad (11)$$

Интеграл (11) можно вычислить через специальную функцию – функцию (или интеграл вероятностей) Лапласа, один из вариантов записи которой имеет вид

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-\frac{s^2}{2}} ds. \quad (12)$$

Сравнивая (12) с выражением (9) для функции $F(x)$ легко установить, что $\Phi(z)$ представляет собой функцию нормального распределения с параметрами $\mu = 0$, $\sigma = 1$, то есть стандартного нормального распределения, характеризующегося следующими свойствами:

- 1) $\Phi(-\infty) = 0$,
- 2) $\Phi(+\infty) = 1$,
- 3) $\Phi(-z) = 1 - \Phi(z)$.

График стандартного нормального распределения приведен на рисунке 4.

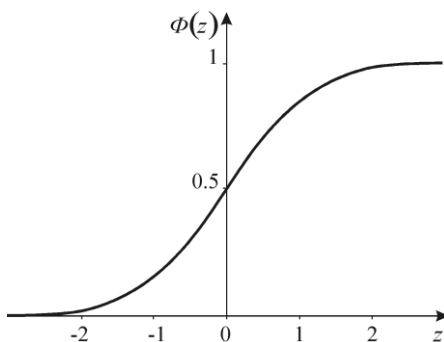


Рис. 4. График функции стандартного нормального распределения

Сравнение выражений (11) и (12) позволяет выразить функцию распределения $F(x)$ с параметрами μ и σ через функцию стандартного нормального распределения:

$$F(x) = \Phi\left(\frac{x - \mu}{\sigma}\right).$$

Определение $\Phi(z)$ не вызывает затруднений благодаря тому, что ее значения при различных вещественных z вычислены и сведены в таблицы, которые содержатся в литературе по теории вероятностей. Значения вероятности $p = \Phi(z)$ и p -квантили $z = \Phi^{-1}(p)$ могут быть определены также с помощью статистических функций табличного процессора Microsoft Excel НОРМСТРАСП(z) и НОРМСТОБР(p) соответственно.

Функция $\Phi(z)$ используется при определении необходимого для достижения заданной точности числа прогонов имитационной модели. Подход к решению указанной задачи выбирается в зависимости от цели проведения имитационного эксперимента. Ниже рассмотрено два варианта цели имитационного моделирования:

- 1) определение среднего значения некоторой СВ;
- 2) нахождение вероятности исхода какого либо события.

Вариант 1. Задача состоит в определении объема выборки N_{II} , позволяющего найти такую оценку \bar{X} истинного значения математического ожидания $\mu = M[X]$ СВ X , которая с вероятностью p отличалась бы от μ не более чем на величину ε :

$$P\{\mu - \varepsilon \leq \bar{X} \leq \mu + \varepsilon\} = p, \quad (13)$$

где \bar{X} – выборочное среднее, ε – заданная погрешность оценки, p – вероятность того, что интервал $\mu \pm \varepsilon$ содержит \bar{X} .

Пусть произведено N_{II} прогонов имитационной модели, давших в результате N_{II} реализаций СВ $X - X_1, X_2, \dots, X_{N_{II}}$, с помощью которых оценка \bar{X} определяется как

$$\bar{X} = \frac{\sum_{i=1}^{N_{II}} X_i}{N_{II}}.$$

Если $X_1, X_2, \dots, X_{N_{II}}$ независимы и одинаково распределены, то, согласно центральной предельной теореме, при достаточно большом N_{II} (на практике больше 10-20) закон распределения

СВ \bar{X} можно приближенно считать нормальным. В этом случае можно показать, что необходимый для выполнения (13) объем выборки N_{II} определяется как

$$N_{II} = \frac{\sigma_X^2}{\varepsilon^2} t_p^2,$$

где σ_X – среднеквадратическое отклонение, t_p – аргумент функции Лапласа, такой, что

$$t_p = \Phi^{-1}\left(\frac{1+\rho}{2}\right).$$

Неизвестная до начала эксперимента величина σ_X может быть приближенно определена в результате проведения нескольких предварительных прогонов.

Вариант 2. Задача состоит в определении объема выборки N_{II} , позволяющего найти такую частоту p^* некоторого случайного события A , которая с вероятностью ρ отличалась бы от неизвестной вероятности p указанного события не более чем на величину ε :

$$P\{|p^* - p| \leq \varepsilon\} = \rho, \quad (14)$$

где ε – заданная погрешность оценки, ρ – вероятность того, что p^* отклоняется от p не более, чем на ε .

Число прогонов N_{II} , обеспечивающее выполнение условия (14):

$$N_{II} = \frac{p \cdot (1-p)}{\varepsilon^2} t_p^2.$$

Неизвестная до начала эксперимента величина p может быть определена в результате проведения нескольких предварительных прогонов. Может быть также принято ориентированное на «худший случай» значение $p = 0.5$, для которого

$$N_{II} = \frac{t_p^2}{4\varepsilon^2}.$$

В качестве примера оценим число прогонов модели, необходимое чтобы с вероятностью 0.95 наблюдаемая частота успеш-

ных исходов некоторого события, отличалась от истинной p не более чем на 0.05. Имеем $\varepsilon=0.05$, $t_p=1.96$ и, следовательно,

$$N_{II} = \frac{(1.96)^2}{4(0.05)^2} \approx 385.$$

2.10 Программные средства имитационного моделирования

К настоящему времени разработано большое число программ, ориентированных на решение задач имитационного обеспечения. Ниже перечислены наиболее известные из них.

Система моделирования AnyLogic. Система AnyLogic поддерживает три технологии создания имитационных моделей: процессно-ориентированный (дискретно-событийный), системно динамический и агентный, а также любую их комбинацию. Графический интерфейс AnyLogic, инструменты и библиотеки позволяют быстро создавать модели для широко спектра задач от моделирования производства, логистики, бизнес-процессов до стратегических моделей развития компании и рынков. AnyLogic стал корпоративным стандартом на бизнес-моделирование во многих транснациональных компаниях, широко используется в образовании. Сайт разработчика: <http://www.xjtek.ru/>

Система моделирования Arena. Arena – система дискретного моделирования. Сфера основных приложений системы – имитационное моделирование производственных технологических процессов и операций, складской учет, банковская деятельность, оптимизация обслуживания клиентов в сфере услуг, транспортные задачи. Разработчик: Rockwell Automation Inc., Wexford, PA, США. Сайт разработчика : <http://www.arenasimulation.com/>

Среда моделирования AweSim. AweSim – это универсальная система имитационного моделирования для сети с дискретной или непрерывной интерпретацией. Возможные области приме-

нения: бизнес, промышленность, здравоохранение, военное дело. Будучи совместимым с языком имитационного моделирования Visual SLAM, продукт включает построение интерактивной модели, одновременную и последующую анимацию, статистическую информацию в текстовом и графическом видах, интерактивное представление и выбор сценариев. Сетевые модели строятся графически и могут быть иерархическими. Они могут быть расширены по заданным пользователем правилам, написанным на языке C или Visual Basic. Одновременно могут отображаться несколько анимированных изображений. Сценарии сравниваются статистически, после чего из них выбирают набор альтернатив с лучшими показателями. Разработчик: Symix Systems Inc., США.

Среда моделирования Enterprise Dynamics. Enterprise Dynamics – это ведущая программная платформа для бизнес моделирования. С Enterprise Dynamics можно анализировать и оптимизировать текущее и будущее поведение системы или инфраструктуры. Каждая отрасль или индустрия имеют дело с уникальными материалами, оборудованием и другими ресурсами. В сотрудничестве со специалистами на этих рынках компания INCONTROL создала определенные объекты с соответствующими функциональными возможностями и объединила их в библиотеку. Следующие продукты доступны: ED Logistics (Логистика), ED Plato, ED Airport (Аэропорт), ED Transport (Транспорт), ED Warehouse (Склад), ED Educational, ShowFlow.

ED Airport. ED Airport – интегрированный программный инструмент имитационного моделирования в целом для аэропортов. Программное обеспечение ED Airport создано на фактических и реальных данных. С ED Airport можно анализировать и проверять долгосрочные события, обнаруживать узкие места, оценивать возможности и распределение персонала, прогнозировать развитие событий.

Разработчик: INCONTROL Simulation Solutions, Нидерланды. Сайт разработчика: <http://www.incontrolsim.com>

Система GPSS/H. Система GPSS/H – это среда моделирования общего назначения, охватывает области как дискретного, так и непрерывного моделирования. Разработчик: компания

Wolverine Software Corp., США. Сайт разработчика: <http://www.wolverinesoftware.com>

Система GPSS World. Система GPSS World – это среда моделирования общего назначения, охватывает области как дискретного, так и непрерывного моделирования. GPSS World включает PLUS-язык программирования нижнего уровня моделирования. Моделирование с использованием PLUS выражений может быть включено почти везде в GPSS программы, в любом блоке или процедуре вызова. Язык PLUS позволяет программно управлять размещением результатов. Система GPSS World решает многозадачность, позволяя нескольким имитационным процессам выполняться одновременно. Разработчик: компания Minuteman Software Corp., США. Сайт: <http://www.minutemansoftware.com/>

Среда моделирования «МВТУ». Программный комплекс «МВТУ» (SimInTech) предназначен для исследования динамики и проектирования разнообразных систем и устройств. По своим возможностям является альтернативой зарубежным программным продуктам Simulink, VisSim. Программный комплекс «МВТУ» применяется для проектирования систем автоматического управления, следящих приводов и роботоманипуляторов, ядерных и тепловых энергетических установок, а также для решения нестационарных краевых задач (теплопроводность, гидродинамика и др.). Используется в учебном процессе, позволяя моделировать различные явления в физике, электротехнике, в динамике машин и механизмов, в астрономии. Может функционировать в многокомпьютерных моделирующих комплексах, в том числе и в режиме удаленного доступа к технологическим и информационным ресурсам. Сайт разработчика: <http://3v-services.com>

3 СИСТЕМА ИМИТАЦИОННОГО КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ GPSS WORLD

3.1 Общее описание системы имитационного моделирования GPSS World

Программное обеспечение GPSS World предназначено для имитационного моделирования систем с дискретными и непрерывными процессами.

Взаимодействие пользователя с GPSS World осуществляется с помощью оконного интерфейса в соответствии со стандартами ОС Windows.

Основные компоненты системы моделирования GPSS World:

- «Модель» создается при помощи встроенного текстового редактора и включает текст модели, созданный пользователем;

- «Процесс моделирования» – результат трансляции модели, который может быть запущен на выполнение с помощью команд GPSS.

- «Отчет» – создается автоматически по завершении моделирования и содержит результаты моделирования.

- «Текст» (текстовый файл GPSS World) предназначен для упрощения разработки больших моделей и создания библиотеки исходных текстов.

- программа-планировщик, выполняющая следующие функции:

- 1) обеспечение продвижения по заданным разработчиком маршрутам динамических объектов, называемых транзактами;

- 2) планирование событий, происходящих в модели, путем регистрации времени наступления каждого

события и выполнения их в нарастающей временной последовательности;

3) регистрация статистической информации о функционировании модели;

4) продвижение модельного времени в процессе моделирования системы.

Средства программирования системы GPSS World:

– язык моделирования GPSS, предназначенный для описания объектов моделирования;

– язык программирования низкого уровня PLUS, ориентированный на вычисления и управление имитационным экспериментом, предназначен для придания гибкости системе моделирования.

Модель на языке GPSS состоит из объектов, которые подразделяются на 7 категорий и 15 типов, представленных в таблице 3.

Таблица 3. Объекты GPSS

Категория	Типы
Динамическая	Транзакты
Операционная	Блоки
Аппаратная	Одноканальные устройства (ОКУ), многоканальные устройства (МКУ), переключатели
Вычислительная	Переменные, функции, генераторы случайных чисел
Статистическая	Очереди, таблицы
Запоминающая	Ячейки, матрицы ячеек
Группирующая	Числовые группы, группы транзактов, списки

Ниже представлено описание основных типов объектов GPSS.

Транзакты – динамические объекты, которые создаются в определенных точках модели, продвигаются планировщиком через блоки, а затем уничтожаются. Транзакты являются аналогами заявок (требований, запросов) в СМО. С каждым транзактом связаны параметры, которые используются для конкретных данных, выражающих свойства транзакта. Каждый транзакт может иметь любое число параметров. Параметры нумеруются или им даются имена. Номера параметров и имена используют-

ся для ссылок на значения, присвоенные параметрам. Транзактам может присваиваться приоритет. Приоритет определяет предпочтение, которое получает транзакт, когда он и другие транзакты претендуют на один и тот же ресурс.

Объекты аппаратной категории – это абстрактные элементы, на которые может быть декомпозирована реальная система. Воздействуя на эти объекты, транзакты могут изменять их состояние и влиять на движение других транзактов. **ОКУ/МКУ** представляют собой оборудование, которое в любой момент времени может быть занято одним/несколькими транзактами. **Переключатели** – устройства, которые могут иметь только два состояния (вкл./выкл., исправно/неисправно). В модели GPSS работа объектов аппаратной категории моделируется объектами операционной категории.

Блоки – операционные объекты, т. е. объекты, задающие логику функционирования модели системы и определяющие пути движения транзактов между объектами аппаратной категории. В блоках могут происходить события четырех основных типов:

- 1) создание или уничтожение транзактов;
- 2) изменение характеристики (числового атрибута) объекта;
- 3) задержка транзакта на определенный период времени;
- 4) изменение маршрута движения транзакта в модели.

В зависимости от назначения блоки подразделяются на 9 групп. В перечне полужирным шрифтом выделены блоки, описанные ниже подробно.

1. Блоки, осуществляющие модификацию атрибутов транзактов:
 - 1.1) генерирование и уничтожение транзактов **GENERATE, SPLIT, TERMINATE, ASSEMBLE**;
 - 1.2) временная задержка **ADVANCE**;
 - 1.3) синхронизация движения двух **MATCH** и нескольких **GATHER** транзактов;
 - 1.4) изменение приоритета транзакта **PRIORITY**;
 - 1.5) изменение параметров транзактов **ASSIGN, INDEX, MARK, PLUS**.
2. Блоки, изменяющие последовательность движения транзактов (блоки передачи управления): **DISPLACE, TRANSFER, LOOP, TEST, GATE**.

3. Блоки, связанные с группирующей категорией: ADOPT, ALTER, EXAMINE, JOIN, REMOVE, SCAN.
4. Блоки, описывающие объекты аппаратной категории:
 - 4.1) одноканальные устройства **SEIZE, RELEASE, PREEMPT, RETURN, FUNAVAIL, FAVAIL;**
 - 4.2) многоканальные устройства **ENTER, LEAVE, SAVAIL, SUNAVAIL;**
 - 4.3) логические переключатели LOGIC.
5. Блоки, сохраняющие необходимые значения для дальнейшего использования: **SAVEVALUE, MSAVEVALUE.**
6. Блоки для получения статистических результатов:
 - 6.1) очереди **QUEUE, DEPART;**
 - 6.2) таблицы **TABULATE.**
7. Блоки для организации списка пользователя: LINK, UNLINK.
8. Блоки для организации ввода-вывода:
 - 8.1) открытие/закрытие файла: OPEN/CLOSE;
 - 8.2) считывание/запись в файл: READ/WRITE;
 - 8.3) установка позиции текущей строки: SEEK.
9. Специальные блоки: BUFFER, COUNT, EXECUTE, INTEGRATION, SELECT, TRACE, UNTRACE.

Объекты вычислительной категория служит для выражения в виде математических соотношений связей между компонентами моделируемой системы. ***Переменные*** представляют собой сложные выражения, которые включают константы, системные числовые атрибуты, библиотечные арифметические функции, арифметические и логические операции.

Системные числовые атрибуты – атрибуты, описывающие состояние объекта в данный момент времени. Соответствуют любому объекту. Доступны для использования в течение всего процесса моделирования. Всего в GPSS World имеется свыше 50 СЧА.

Примеры СЧА.

СЧА общесистемные:

С1 – относительное модельное время;

СЧА транзактов:

P_i – значение i -го параметра активного (обрабатываемого в данный момент) транзакта;

PR – приоритет активного транзакта;

M_1 – время пребывания в модели (или в ее участке) активного транзакта.

СЧА генератора случайных чисел:

RN_n – номер датчика случайных чисел, $n = 1, \dots, 99$.

(Датчик генерирует последовательность равномерно распределенных случайных целых чисел от 0 до 999);

СЧА ОКУ:

$F\$name$ – занятость устройства с именем $name$, 1 – занято, 0 – не занято.

$Q\$name$ – текущая длина очереди с именем $name$.

СЧА ячеек (сохраняемых ячеек):

$X\$name$ – значение ячейки с именем $name$ или

X_j – значение ячейки под номером j .

СЧА функций:

$FN\$name$ – значение функции с именем $name$.

Функция – вычислительный объект, позволяющий пользователю производить вычисления непрерывных или дискретных функциональных зависимостей между аргументом и зависимым значением функции.

Генераторы случайных чисел – кроме датчика равномерно распределенных случайных целых чисел (RN) GPSS World имеет 24 встроенных генератора случайных чисел, подчиняющихся заданному распределению (показательному, нормальному, Пуассона и т.д.).

Пример.

Для генерации случайных величин, распределенных по экспоненциальному закону, можно использовать встроенное вероятностное распределение, которое описывается выражением

$EXPONENTIAL(A,B,C)$

где A – номер датчика случайных чисел, B – сдвиг среднего, C – сжатие функции (или среднее при $B=0$). В результате значение среднего определяется суммой полей B и C , а значение дисперсии равно квадрату поля C . Выражение $\text{EXPONENTIAL}(A,B,C)$ при обращении к нему заключается в скобки.

Примеры операторов, используемых для моделирования распределений.

$\text{UNIFORM}(\text{Stream}, \text{Min}, \text{Max})$ – равномерное;

$\text{LOGNORMAL}(\text{Stream}, \text{Locate}, \text{Scale}, \text{Shape})$ – логарифмически-нормальное;

$\text{NORMAL}(\text{Stream}, \text{Mean}, \text{StdDev})$ – нормальное (Гаусса);

$\text{GAMMA}(\text{Stream}, \text{Locate}, \text{Scale}, \text{Shape})$ – гамма-распределение;

$\text{WEIBULL}(\text{Stream}, \text{Locate}, \text{Scale}, \text{Shape})$ – распределение Вейбулла.

Объекты статистической категории используются для сбора статистики. Планировщик автоматически накапливает определенную статистику относительно устройств и очередей. Объекты *«очередь»* позволяют собирать дополнительную статистику по очередям. Объекты *«таблица»* используются для облегчения табулирования статистической информации в GPSS.

Объекты запоминающей категории используются для сохранения некоторой числовой информации. Любой активный транзакт может произвести запись информации в эти объекты. Впоследствии записанную в эти объекты информацию может считать любой транзакт. *Ячейки* – ячейки памяти, начальные значения которых могут быть заданы перед моделированием и к которым можно обратиться из любого места модели во время моделирования.

Объекты группирующей категории используются для расширения возможностей системы моделирования, например они позволяют организовать очереди с дисциплинами, отличными от дисциплины «первым пришел – первым обслужен».

3.2 Структура модели на языке GPSS

Текст модели на языке GPSS записывается в виде последовательности операторов. Операторы делятся на два вида: операторы GPSS и PLUS-операторы. Операторы GPSS в свою очередь подразделяются на операторы блоков, т.е. те операторы, которые создают блоки, и – команды, которые не создают блоков.

Правила записи операторов GPSS:

1) операторы записываются одной текстовой строкой длиной не более 250 символов;

2) операторы состоят из частей, называемых полями. Поле – набор символов, отделенный пробелами;

3) оператор состоит из следующих полей: номер строки (не обязательно), метка (не обязательно), глагол (обязательно), операнды (зависит от глагола), комментарий (не обязательно);

4) номера строк игнорируются программой-отладчиком.

Для различения объектов GPSS, например, ОКУ, МКУ, очередей и т.д., которых в модели может быть несколько, им дают имена. Имена должны начинаться с буквы, могут содержать до 200 букв и цифр, а также символы подчеркивания. Имя не должно совпадать с ключевыми словами GPSS.

Для обеспечения возможности перехода от одного объекта модели к другому объекты могут быть идентифицированы при помощи меток. Метки образуются также, как и имена блоков.

В поле глагола необходимо записать ключевое слово GPSS, т.е. блок или команду.

Команды используются для описания МКУ, переменных, функций, таблиц, управления процессом моделирования. Как правило, в тексте модели набор команд предшествует набору операторов.

Программа модели может содержать несколько сегментов, каждый из которых соответствует процессам, протекающим в «реальной» системе параллельно.

3.3 Объекты вычислительной и запоминающей категории

Рассматриваются только используемые при выполнении лабораторных работ объекты. *Арифметические, условные и логические операторы*, а также *библиотечные математические функции* приведены в таблицах 4 и 5.

Таблица 4. Арифметические, условные и логические операторы

Оператор	Операция	Аргумент, значение
\wedge	Возведение в степень	$A \wedge B$; A, возведенное в степень B
#	Умножение	$A \# B$; произведение A на B
/	Деление	A / B ; частное от деления A на B
\	Целочисленное значение	$A \setminus B$; частное от целочисленного деления A на B
@	Целый остаток	$A @ B$; целый остаток от деления A на B
–	Вычитание	
+	Сложение	
\geq 'GE'	Больше или равно	1, если $A \geq B$, иначе — 0
\leq 'LE'	Меньше или равно	1, если $A \leq B$, иначе — 0
$>$ 'G'	Больше	1, если $A > B$, иначе — 0
$<$ 'L'	Меньше	1, если $A < B$, иначе — 0
= 'E'	Равно	1, если $A = B$, иначе — 0
\neq 'NE'	Не равно	1, если $A \neq B$, иначе — 0
& 'AND'	Логическое «И»	1, если A и B не 0, иначе — 0
'OR'	Логическое «ИЛИ»	1, если A или B или оба не 0, иначе — 0

Таблица 5. Библиотечные математические функции

Оператор	Функции
ABS(A)	Абсолютное значение
ATN(A)	Арктангенс
COS(A)	Косинус
EXP(A)	Экспонента
INT(A)	Выделение целой части
LOG(A)	Натуральный алгоритм
SIN(A)	Синус
SQR(A)	Квадратный корень
TAN(A)	Тангенс

Переменные пользователя. Используются для хранения нужных при моделировании числовых и строковых данных. Создаются с помощью команды EQU или присваивающих PLUS-процедур. Например:

```
N_      EQU    25.3
VS      EQU    10
Imya    EQU    "Vasya"
```

Сохраняемые ячейки. Представляют собой ячейки памяти, начальные значения которых могут быть заданы перед моделированием и к которым можно обратиться из любого места модели во время моделирования. Значения сохраняемых ячеек могут быть изменены в ходе выполнения модели. Перед началом моделирования по умолчанию сохраняемым ячейкам устанавливаются значения «0». Ненулевые значения могут быть заданы пользователем с помощью команды INITIAL, имеющей следующий формат:

```
INITIAL  A,[B]
```

Операнд А – имя или номер инициализируемой сохраняемой ячейки. Может принимать значения Xположительное целое, X\$имя.

Операнд В – присваиваемое первоначальное значение. Может быть именем, числом, строкой или UNSPECIFIED (не определено). Например:

```
INITIAL X32,123.45
INITIAL X$Cost,Cena
```

В первом примере в ячейку номер 32 заносится число 123.45. Во втором – в ячейку с именем Cost – значение переменной пользователя Cena. Предварительно переменная Cena должна быть определена с помощью команды EQU.

Для изменения значения сохраняемой ячейки в процессе моделирования используется блок SAVEVALUE. Формат записи блока:

```
SAVEVALUE A,B
```

При входе транзакта в блок SAVEVALUE величина, стоящая в операнде В, становится значением сохраняемой ячейки, номер или имя которой записан в операнде А. Операнд А может быть именем, положительным целым числом, выражением в скобках, СЧА, а операнд В, кроме этого, может быть числом или строкой.

Блок SAVEVALUE может использоваться в режимах замещения, накопления или уменьшения. В режиме накопления (уменьшения) предыдущее значение сохраняемой ячейки увеличивается (уменьшается) на значение, стоящее в операнде В. Примеры:

```
SAVEVALUE 5,234
SAVEVALUE 5+,234
SAVEVALUE 5-,234
SAVEVALUE ATZ+,P2
SAVEVALUE 5,(P2+ EXPONENTIAL(2,0,1))
```

В первом примере значение сохраняемой ячейки номер 5 при входе транзакта в блок SAVEVALUE становится равным числу 234. Во втором (третьем) примерах значение той же ячейки увеличивается (уменьшается) на 234. В четвертом примере

значение сохраняемой ячейки с именем ATZ увеличивается на величину, записанную во втором параметре вошедшего транзакта. В пятом примере значение сохраняемой ячейки номер 5 при входе транзакта в блок SAVEVALUE становится равным результату вычисления выражения в скобках $(P2 + EXPONENTIAL(2,0,1))$.

Сохраняемые ячейки имеют единственный СЧА с названием X, значением которого является текущее значение соответствующей сохраняемой ячейки. Примеры:

```
SAVEVALUE 5,X$Cost  
SAVEVALUE 5,X32
```

В первом примере значение сохраняемой ячейки номер 5 при входе транзакта в блок SAVEVALUE становится равным значению сохраняемой ячейки под именем Cost. Во втором примере значение той же ячейки становится равным значению 32-й сохраняемой ячейки.

Косвенная адресация.

Косвенная адресация используется в сочетании с любым СЧА, кроме текущего времени C1, случайного числа RN*n* и времени пребывания транзакта в системе M1. Структура записи косвенной адресации

СЧА**<целое число/имя>*,
где *<целое число/имя>* – это номер/имя параметра транзакта, в котором указан номер или имя соответствующего СЧА.

Примеры:

```
SEIZE *1  
Q*5
```

В первом примере при поступлении транзакт занимает ОКУ, имя которого указано в первом параметре транзакте.

Во втором примере определяется длина очереди, имя которой указано в пятом параметре транзакта.

3.4 Организация поступления транзактов в модель и удаления транзактов из нее

Поступление транзактов в модель реализуется с помощью блока GENERATE. Формат записи блока GENERATE:

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

где [] – символы, означающие, что данный операнд является необязательным. Число различных блоков GENERATE в одной модели неограничено. Значения операндов:

A – средний интервал времени между последовательными поступлениями транзактов в модель;

B – модификатор, изменяющий значение интервала генерации транзактов по сравнению с интервалом, указанным операндом A. Есть два типа модификаторов: модификатор-интервал и модификатор-функция. С помощью модификатора-интервала задается равномерный закон распределения времени между генерацией транзактов.

Операнды A и B могут быть именем, положительным числом, выражением в скобках или СЧА.

При вычислении разности (A–B), заданных операндами A и B, получается нижняя граница интервала; при вычислении суммы (A+B) – верхняя граница.

Когда операнды A и B задают в виде констант (B – модификатор-интервал), они должны быть неотрицательными числами.

Когда один или оба операнда (A и/или B) не указаны, по умолчанию предполагается их нулевое значение. Например, детерминированное значение интервалов времени может быть задано как A=16.4, B=0.

Более сложные интервалы времени поступления транзактов (не по равномерному закону) могут быть заданы с использованием модификатора-функции или встроенных генераторов случайных чисел. Под действием модификатора-функции значение операнда A умножается на значение функции, заданной операндом B.

При любом способе вычисления интервала времени значение операнда В не должно превышать значения операнда А, в противном случае в блоке GENERATE может быть получен отрицательный интервал времени, который вызовет останов по ошибке.

С – смещение интервалов (первоначальная задержка) – это момент времени, в который в блоке GENERATE должен появиться первый транзакт. После этого первого прихода все остальные приходы транзактов возникают в соответствии с распределением интервалов времени, задаваемых операндами А и В.

Д – граничное значение общего числа транзактов, которые могут войти в модель через данный блок GENERATE в течение времени моделирования. Когда это число достигнуто, данный блок GENERATE перестает быть активным. Если операнд D не используется, блок GENERATE остается активным в течение всего времени моделирования, т. е. по умолчанию ограничения на количество создаваемых транзактов нет.

Е – класс приоритета транзактов, входящих в модель через данный блок GENERATE. Для задания приоритетов рекомендуется использовать последовательность целых чисел 0, 1, 2, 3, Чем выше число, тем выше приоритет. Если операнд Е не используется, по умолчанию приоритет генерируемых блоком GENERATE транзактов равен нулю.

В любом блоке GENERATE должен быть обязательно задан либо операнд А, либо операнд D. Нельзя использовать в качестве операнда параметры транзактов. Транзакт не должен входить в блок GENERATE.

Примеры записи блоков GENERATE:

с операндом А

GENERATE 38.6

GENERATE (Exponential(11,0,1))

с операндами А и В

GENERATE 73.25,10

с операндами А и С

GENERATE 7.3,,4.1

GENERATE (Normal(3,10,2)),,5.55

с операндами A, B, E
GENERATE 13.3,2.8,,,1

Все описанные в блоке GENERATE переменные и сохраняемые ячейки, должны быть предварительно определены. Поэтому в модели блоку GENERATE должны предшествовать команды определения EQU, INITIAL, FUNCTION и некоторые другие.

Например:

SrZnach	EQU	47.2
Razbr	EQU	28.6
	INITIAL	X\$KolTranz,43
...		
GENERATE		SrZnach,Razbr,,X\$KolTranz

В примере блоку GENERATE предшествуют присвоения командой EQU именам переменных числовых значений и командой INITIAL начального значения сохраняемой ячейке с именем Kol-Tranz.

Для запуска и завершения процесса моделирования используется команда START. Команда START указывается в конце (или в начале) программы модели и имеет следующий формат записи:

START A,[B],C,[D]

Операнд A задает значение счетчика завершения, определяющего момент окончания прогона модели. Может быть только целым положительным числом. Счетчик завершения – ячейка памяти с именем TG1, которая хранит положительное целое число, записываемое в ячейку командой START в начале процесса моделирования. В ходе моделирования при выходе транзактов из модели через определенные блоки из счетчика завершения вычитаются определенные целые положительные числа. При достижении содержимым счетчика нуля моделирование завершается.

Операнд В – операнд вывода статистики. Он может быть NP («нет вывода данных») либо опущен. При задании NP стандартный отчет не выводится. По умолчанию выводится стандартный отчет. Операнд С не используется и сохранен для совместимости с описаниями ранних версий GPSS. Операнд D определяет необходимость вывода содержимого списков событий. Если он опущен, то по умолчанию списки в стандартном отчете не выводятся.

Транзакты удаляются из модели, попадая в блок TERMINATE. В модели может быть любое число блоков TERMINATE. Блок имеет следующий формат записи:

TERMINATE [A]

где А – число единиц, на которое блок TERMINATE уменьшает содержимое счетчика завершения. По умолчанию значение операнда А равно нулю, т.е. при уничтожении транзакта значение счетчика завершения не меняется. В модели может быть много блоков TERMINATE, но счетчик завершения только один.

Сочетание блока TERMINATE и команды START может использоваться для задания временного интервала моделирования.

Пример.

Пусть в качестве единицы модельного времени выбрана 1 мин. Необходимо промоделировать поведение системы в течение 10 часов (600 минут). В модели должен присутствовать сегмент вида

```
GENERATE    „600
TERMINATE   1
START       600
```

Значение операнда А для других блоков TERMINATE, расположенных во всех других сегментах модели, должно быть равно нулю (или быть незадаанным).

В этом случае в процессе моделирования транзакты, которые двигаются в других сегментах модели, время от времени выводятся из модели в других блоках TERMINATE, но они не оказы-

вают воздействия на счетчик завершения. В момент модельного времени 600 транзакт в приведенном выше сегменте выходит из блока GENERATE и сразу же переходит в следующий блок TERMINATE.

3.5 Изменение значений параметров транзактов

Каждый транзакт может иметь любое число параметров. Интерпретация смысла параметров произвольная. В момент генерации транзактов все его параметры нулевые. Блок ASSIGN является основным средством для задания значений параметров транзактов.

Формат записи:

ASSIGN A,B,[C]

Операнд А – задание номера параметра, которому присваивается значение. Операнд А может быть именем, положительным целым числом, выражением в скобках, СЧА, СЧА*параметр и следующими за ними знаками + (если нужно увеличить), – (если нужно уменьшить).

Операнд В – значение, которое следует добавить, вычесть или которым следует заменить значение в параметре, заданном операндом А. Если такой параметр не существует, то он создается со значением, равным 0. Операнд В может быть таким же как и операнд А, кроме того, числом и строкой.

Операнд С задает номер модификатора-функции. При использовании операнда С значение операнда В умножается на значение модификатора-функции. Полученное произведение становится значением, которое изменяет значение параметра, заданного операндом А.

Примеры:

ASSIGN 1,754.3


```
ASSIGN 4+,$MyOchered  
ASSIGN Name,"Plan"
```

В первом примере параметру 1 присваивается значение 754.3. Во втором примере к значению параметра 4 прибавляется значение длины текущей очереди с именем MyOchered. В третьем примере параметру с именем Name присваивается строка Plan.

Для занесения в активный транзакт или его параметр значения абсолютного модельного времени используется блок MARK.

Формат записи:

```
MARK [A]
```

Операнд А – номер параметра транзакта, в который записывается значение абсолютного модельного времени. Может быть именем, положительным целым числом, выражением в скобках, СЧА или СЧА*параметр.

Примеры:

```
MARK  
MARK 2
```

В первом примере вошедшему в блок транзакту устанавливается время входа, равное абсолютному модельному времени. С помощью блока MARK может быть определено время пребывания в модели или в ее сегменте активного транзакта, хранящегося в СЧА М1 транзакта. Значение СЧА М1 транзакта вычисляется как разность абсолютного модельного времени и отметки времени активного транзакта, в которой хранится либо время поступления транзакта в модель, либо время последнего прохождения транзактом блока MARK без операнда А.

Во втором примере абсолютное модельное время входа в блок записывается во 2-й параметр транзакта.

3.6 Занятие и освобождение одноканального устройства

Если в какой-то момент транзакт должен быть обслужен некоторым ОКУ, то для этого он входит (или пытается войти) в соответствующий блок SEIZE, описывающий это ОКУ. Блок SEIZE обладает следующими свойствами:

- если ОКУ уже используется, транзакт не может войти в блок и должен ждать в очереди;
- если ОКУ не используется, транзакт может войти в блок, статус ОКУ изменяется на «занято».

Вход транзакта в блок SEIZE моделирует занятие ОКУ. После обслуживания (блок, имитирующий обслуживание, рассмотрен ниже) вход того же транзакта в блок RELEASE моделирует освобождение ОКУ. Назначение блока RELEASE – изменение состояния ранее занятого ОКУ с «занято» в «незанято».

Форматы блоков:

SEIZE	A
RELEASE	A

Операнд A – имя занимаемого (освобождаемого) ОКУ. Может быть именем, положительным целым числом, выражением в скобках, СЧА, СЧА*параметр.

3.7 Имитация обслуживания посредством задержки во времени

После моделируемого блоком SEIZE занятия транзактом ОКУ, начинается его обслуживание, которое длится некоторый промежуток модельного времени. В течение этого времени

транзакт должен прекратить двигаться по модели. Только по истечении времени обслуживания он должен попасть в блок RELEASE для освобождения ОКУ.

Для задержки транзакта в течение некоторого интервала модельного времени используется блок ADVANCE. Чаще всего этот интервал задается случайной переменной. Как и при использовании блока GENERATE информация, необходимая для описания соответствующего времени обслуживания и его распределения, задается операндами А и В. Формат записи блока ADVANCE:

ADVANCE А,[В]

Значения операндов следующие: А – среднее время обслуживания, В – способ модификации операнда А. Каждый из операндов А и В может быть именем, числом, выражением в скобках, СЧА или СЧА*параметр.

Как и в блоке GENERATE, модификаторы могут быть двух типов: модификатор-интервал и модификатор-функция, т. е. блок ADVANCE вычисляет время задержки такими же способами, как и блок GENERATE.

Блок ADVANCE никогда не препятствует входу транзакта. Любое число транзактов может находиться в этом блоке одновременно.

Блок ADVANCE можно располагать в любых местах модели, а не только между блоками SEIZE и RELEASE.

Пример использования блоков SEIZE, ADVANCE и RELEASE:

SEIZE	Trap
ADVANCE	21,7
RELEASE	Trap

Транзакт, войдя в блок SEIZE, займет ОКУ с именем Trap, задержится там благодаря блоку ADVANCE на 21 ± 7 единиц времени и затем покинет его. После того, как транзакт войдет в блок RELEASE, планировщик попытается продвинуть транзакт

в следующий блок модели, и следующий транзакт попытается использовать ОКУ, называемое Tgap.

3.8 Регистратор очереди

Регистратор очереди – средство GPSS, обеспечивающее возможность сбора следующей статистики, характеризующей процессы ожидания транзактами в очереди:

- 1) число транзактов, вошедших в очередь;
- 2) число транзактов, фактически присоединившихся к очереди (задержавшихся в ней), и – сразу, без задержки, занявших ОКУ;
- 3) максимальное значение длины очереди;
- 4) среднее число ожидающих транзактов в очереди;
- 5) среднее время ожидания тех транзактов, которым пришлось ждать.

Регистраторы различных очередей различаются именами. Условия задания имен те же, что и у устройств. Разработчик вносит регистратор очереди в модель с помощью пары взаимодополняющих блоков QUEUE (стать в очередь) и DEPART (покинуть очередь).

Формат блоков:

QUEUE A,[B]
DEPART A,[B]

Блок QUEUE (DEPART) увеличивает (уменьшает) длину очереди. Операнд A задает номер или имя очереди, к длине которой добавляются (из длины которой вычитаются) единицы.

Операнд B определяет число единиц, на которое увеличивается (уменьшается) текущая длина очереди. Если операнд B не используется, то прибавляется (вычитается) единица.

Операнды A и B в блоках QUEUE и DEPART могут быть именем, положительным целым числом, выражением в скобках, СЧА или СЧА*параметр.

Пример:

QUEUE MyQ

...

DEPART MyQ

Сначала, при входе каждого транзакта в блок QUEUE, длина очереди MyQ увеличивается на единицу, а затем, при входе в блок DEPART, длина указанной очереди на единицу уменьшается.

Пример:

QUEUE P14,P1

...

DEPART P14,P1

Сначала, при входе каждого транзакта в блок QUEUE, длина очереди, номер или имя которой задан в параметре P14 транзакта, увеличивается на число единиц, заданных в параметре P1, а затем, при входе в блок DEPART, длина указанной очереди уменьшается на то же число единиц.

Пример использования блоков SEIZE, ADVANCE, QUEUE, DEPART и RELEASE:

QUEUE MyQ

SEIZE Trap

DEPART MyQ

ADVANCE 21,7

RELEASE Trap

Если транзакт вошел в сегмент в момент, когда ОКУ Trap не занято, транзакт входит в блок QUEUE. Далее транзакт пытается войти в блок SEIZE и, поскольку ОКУ Trap свободно, эта попытка оказывается успешной. Состояние Trap меняется на «занято», и далее транзакт сразу попадает в блок DEPART и, затем, в блок ADVANCE, где задерживается на некоторое время, вычисленное в соответствии с равномерным распределением 21 ± 7 .

Предположим теперь, что ОКУ Trap находится в занятом состоянии и следующий транзакт входит в сегмент модели. Он

проходит в блок QUEUE и получает далее отказ, поскольку Tgap находится в занятом состоянии. Транзакт перестает двигаться, оставаясь в блоке QUEUE.

Позднее, когда транзакт, находящийся на обслуживании в устройстве, покидает его, ожидающий транзакт опять попытается войти в блок SEIZE. На этот раз попытка окажется успешной. Двигаясь дальше, транзакт войдет в блок DEPART, уменьшая значение счетчика содержимого очереди на 1 (по умолчанию, так как операнд B не используется).

3.9 Статистические таблицы

Для получения плотности распределения, ее интегральных (накопленных) относительных частот, среднего значения и стандартного отклонения некоторых аргументов, которыми могут быть СЧА (например, времени нахождения транзакта в модели или задержки в ее отдельных частях, длин очередей, содержимого МКУ, коэффициентов использования устройств и т. д.), используются статистические таблицы TABLE и QTABLE.

Формат команды описания таблицы TABLE:

Name TABLE A,B,C,D

где Name – метка, определяющая имя таблицы;

A – аргумент таблицы, т.е. элемент данных, чье частотное распределение будет табулироваться. Операнд может быть именем, выражением в скобках или СЧА;

B – верхний предел первого частотного интервала;

C – ширина частотного интервала, т.е. разница между верхней и нижней границей каждого частотного класса;

D – число частотных интервалов (положительное целое число).

Для сбора данных транзакт должен войти в блок TABULATE с тем же именем таблицы, которое определено в

блоке TABLE. Блок помещается в ту точку модели, которая соответствует исследуемому объекту.

Формат блока TABULATE:

TABULATE A,[B]

Операнд А – имя таблицы, в которую табулируется значение аргумента.

Операнд В – (весовой коэффициент) число единиц, которые должны быть занесены в тот частотный интервал, в который попало значение аргумента. Если операнд В отсутствует, то по умолчанию эта величина равна 1.

Операнды А и В могут быть именем, выражением в скобках, СЧА или СЧА*параметр. Операнд А может быть только положительным целым числом, а операнд В – положительным числом.

Пример:

```
VrObsl    TABLE        P10,8,2,20
          ...
          TABULATE      VrObsl
```

Оператором TABLE описывается таблица с именем VrObsl. Аргументом таблицы является СЧА значение десятого параметра активного транзакта с верхним пределом первого интервала 8, шириной 2 и числом интервалов 20.

Кроме таблиц TABLE могут использоваться таблицы QTABLE, являющиеся средством получения распределения только времени пребывания транзакта в очереди. Для создания в модели такой таблицы ее нужно предварительно определить с помощью команды QTABLE формата:

Name QTABLE A,B,C,D

Операнд А – имя очереди. В, С, D – операнды такого же назначения, что и в команде описания TABLE. Операнд В может быть нулем или положительным числом.

Пример:

TimeMyQ QTABLE MyQ,1,1,20

где MyQ – имя очереди к устройству.

При использовании QTABLE информация в таблицу заносится автоматически при входе транзакта в блоки QUEUE и DEPART и никаких специальных мер, т. е. блока TABULATE, при этом не требуется.

Собранная в таблицах TABLE и QTABLE статистика выводится в стандартный отчет.

3.10 Построение моделей систем с многоканальными устройствами

МКУ используется для моделирования нескольких однородных (с одинаковыми характеристиками) параллельных устройств. МКУ может быть использовано несколькими транзактами одновременно. Ограничений на число МКУ в модели нет. Для различия им дают имена.

МКУ определяется до его использования командой STORAGE. Формат команды:

Name STORAGE A

Name – имя МКУ; A – операнд, задающий количество каналов (емкость) МКУ, может быть только целым положительным числом.

Занятие и освобождение МКУ имитируется блоками ENTER и LEAVE. Форматы блоков:

ENTER A,[B]

LEAVE A,[B]

Значение операндов: А – имя МКУ; В – число устройств (каналов), которое должно быть занято в блоке ENTER или освобождено в блоке LEAVE. По умолчанию операнд В = 1.

Имитация обслуживания транзакта в МКУ осуществляется блоком ADVANCE.

Пример:

```

Nak      STORAGE  20
...
ENTER    Trap,2
ADVANCE  120,40
LEAVE    Trap,2
...

```

Командой STORAGE определяется МКУ с именем Trap, располагающее двумя устройствами. При входе транзакта в блок ENTER занимает 2 единицы и столько же освобождается в блоке LEAVE при выходе из МКУ.

3.11 Управление продвижением транзактов

Блок TEST описывает условие, которое проверяется при входе в него транзакта, и определяет номер следующего блока для вошедшего транзакта в зависимости от того, выполняется требуемое условие или нет. Формат блока TEST:

TEST X A,B,[C]

где X – операция отношения (GE, LE, G, L, E, NE; см. выше); Операнды А и В – левая и правая части отношения соответственно; операнд С – метка или номер блока, в который будет передан транзакт в случае невыполнения условия, заданного блоком TEST. А, В, С могут быть именем, числом, СЧА, СЧА*параметр.

Блок TEST функционирует в двух режимах:

1. Если операнд *C* не используется, блок TEST функционирует в режиме отказа. Когда транзакт пытается войти в блок TEST, работающий в режиме отказа, и заданное условие не выполняется, транзакт блокируется, и ему не позволяется войти в блок TEST, проверка повторяется до тех пор, пока условие не будет выполнено. После выполнения заданного условия активный переходит к следующему по порядку блоку.

2. Если операнд *C* используется, блок TEST функционирует в режиме передачи транзакта. Когда транзакт пытается войти в такой блок TEST и проверяемое условие не выполняется, транзакт переходит к блоку, указанному в операнде *C*. Если проверяемое условие выполняется, активный транзакт входит в блок TEST и затем переходит к следующему по порядку блоку.

Примеры:

```
TEST L   Q$MyQ,100
TEST LE  Q$MyQ,100,MET1
```

В первом примере после входа в блок TEST транзакт при выполнении условия $Q\$MyQ < 100$ («длина очереди с именем *MyQ* меньше 100 единиц») перейдет к следующему блоку, иначе транзакт будет задержан вплоть до выполнения указанного условия.

Во втором примере после входа в блок TEST транзакт при выполнении условия $Q\$MyQ \leq 100$ («длина очереди с именем *MyQ* не превышает 100 единиц») перейдет к следующему блоку, иначе транзакт будет передан на блок, помеченный меткой MET1.

Для передачи транзакта в любой указанный блок модели используется блок TRANSFER. Характеризуется широким набором режимов работы. Здесь рассматриваются наиболее простые режимы – режимы статистической и безусловной передачи, в которых блок TRANSFER имеет следующий формат:

```
TRANSFER [A],B
```

В режиме безусловной передачи операнд *A* не используется. Операнд *B* указывает имя блока, в который транзакт должен по-

пытаться войти. Блок TRANSFER не может отказать транзакту во входе. Операнд В может быть именем, положительным целым числом, выражением в скобках, СЧА, СЧА*параметр.

Пример:

TRANSFER ,Met

После входа транзакт сразу же пытается войти в блок с меткой Met. Если этот блок отказывает во входе, транзакт остается в блоке TRANSFER.

В режиме статистической передачи операнд А используется и блок TRANSFER передает транзакт в один из двух блоков случайным образом. Значение операнда А рассматривается как вероятность Р перехода транзакта в следующий блок. С вероятностью $(1-P)$ транзакт переходит в блок с меткой Met. Операнды А и В могут быть именем, положительным целым числом, выражением в скобках, СЧА, СЧА*параметр.

Пример:

TRANSFER 0.3, Met

Транзакт с вероятностью 0.3 продолжает движение к следующему блоку, с вероятностью 0.7 – поступает в блок, помеченный меткой Met.

3.12 Работа с копиями транзактов

Зависимость между процессами, протекающими в разных частях системы, может выражаться в форме взаимного согласования этих процессов по времени, то есть в форме синхронизации. Для организации синхронизации используется создание копий транзактов (транзактов-потомков), образующих вместе с транзактом-оригиналом (исходным, порождающим транзактом, транзактом-родителем) семейство транзактов.

Для создания копий транзакта используется блок SPLIT. Формат блока SPLIT:

SPLIT A,[B],[C]

Значения операндов: А – количество генерируемых копий одного семейства; В – имя или номер блока, куда переходят копии порождающего транзакта; С – номер параметра транзакта, используемого для присвоения копиям и порождающему транзакту последовательных номеров.

Операнды А, В, С могут быть именем, положительным целым числом, выражением в скобках, СЧА, СЧА*параметр.

Примеры:

SPLIT 2

SPLIT 3,Met,1

В первом примере создаются две копии, которые вместе с порождающим транзактом переходят к следующему блоку.

Во втором примере создаются три копии, которые переходят к блоку с меткой Met. Параметр 1 порождающего транзакта и копий будет модифицирован. Если параметр 1 порождающего транзакта не был определен, то он будет создан и после выполнения блока SPLIT его значение порождающего транзакта станет равным 1, у первой копии – 2, у второй – 3, у третьей – 4.

Для объединения заданного числа транзактов одного семейства в один транзакт используется блок ASSEMBLE. Формат блока ASSEMBLE:

ASSEMBLE A

Операнд А – число транзактов одного семейства, подлежащих объединению. Может быть именем, положительным целым числом, выражением в скобках, СЧА, СЧА*параметр.

Примеры:

ASSEMBLE 7

ASSEMBLE *7

В первом примере блок собирает 7 транзактов одного семейства, 6 уничтожаются, один переходит в следующий блок.

Во втором примере использована косвенная адресация. Блок собирает число транзактов одного семейства, равное значению параметра 7 первого пришедшего в блок транзакта семейства. Способ удобен, когда количество собираемых транзактов одного семейства заранее не известно, а определяется в ходе моделирования.

Для синхронизации движения двух транзактов одного семейства, продвигающихся по различным путям в модели, предназначен блок MATCH. Необходимы два сопряженных блока MATCH, находящихся в разных местах модели. Формат записи:

Name MATCH A

где Name – имя рассматриваемого блока MATCH; операнд A – метка или номер сопряженного ему блока. Может быть именем, положительным целым числом, выражением в скобках, СЧА, СЧА*параметр.

Пример:

PosadkaPassazhirov	MATCH	PogruzkaBagazha
...		
PogruzkaBagazha	MATCH	PosadkaPassazhirov

В примере два транзакта одного семейства могут пройти через сопряженные блоки MATCH только одновременно.

3.13 Функции

В GPSS могут быть использованы функции, заданные таблицей. Таблично функции задаются с помощью команды FUNCTION, записываемой по формату:

Name FUNCTION A, BN

где Name – числовое или символическое имя функции; A – аргумент функции. Допустимые значения – имя, положительное

целое число, строка, выражение в скобках, СЧА, СЧА*параметр; В – тип функции (одна буква) и N – количество пар данных в списке данных функции.

Типы функций, применяемые в GPSS:

С – непрерывная числовая;

D – дискретная числовая;

E – дискретная атрибутивная;

L – списковая числовая;

M – списковая атрибутивная;

За строкой FUNCTION A,BN сразу же должна следовать строка, содержащая список пар данных, разделенных символом «/». Каждая пара данных определяет значения аргумента X и значения функции Y, разделенные запятой. Список данных используются для вычисления значения функции по заданным значениям аргумента.

Команды задания функции располагаются в начале программы модели, до первого блока GENERATE.

Обращение к функции выглядит, как FN\$I, где I – имя функции.

Ниже рассмотрено 2 наиболее часто используемых типа функций.

1. **Функции типа С** – непрерывные числовые функции.

В списке данных функций типа С значения X и Y должны быть целочисленными или вещественными.

Вычисление функции начинается с определения аргумента. Далее определяется интервал $(X_i; X_{i+1})$, на котором находится найденное значение аргумента и на этом интервале выполняется линейная интерполяция с использованием соответствующих значений Y_i и Y_{i+1} . Результатом является значение функции. Если аргумент попадает за предельные значения области определения функции, возвращается значение функции в ближайшей предельной точке.

Пример:

```
ART      FUNCTION  X1,C3
1.1,10.1/20.5,98.7/33.3,889.2
```

Команда определяет кусочно-линейную функцию с двумя линейными участками, график которой представлен на рисунке 5. Если мы обращаемся к функции FN\$ART, то по значению сохраняемой ячейки X1 вычисляется функция в соответствии с заданной совокупностью точек.

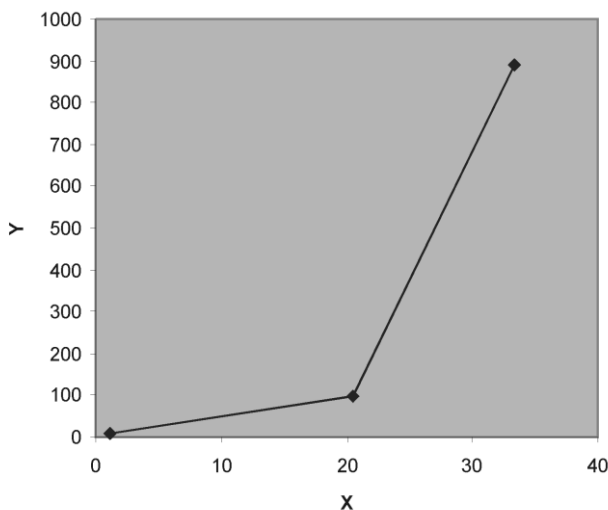


Рис. 5. График кусочно-линейной функции

2. *Функции типа D* – дискретные числовые функции.

В списках данных функций типа D значения X должны быть целочисленными или вещественными, а значения Y – целочисленными или вещественными.

Функция типа D задает одно и то же значение функции Y_i для всех значений аргумента $X_{i-1} < X \leq X_i$. Значения X в списке данных функции должны быть неубывающими. Когда вычисляется функция, значения X в списке данных функции просматриваются от наименьшего к наибольшему. Когда найдено значение X , которое больше или равно текущему значению аргумента, возвращается соответствующее ему значение Y . Если такое значение X отсутствует, возвращается значение Y или именованная величина, соответствующая самому большому значению X .

Пример:

RAF FUNCTION RN1, D5

.2,1.0/4,7.2/.6,6.667/.8,8/1.0,10

Командой определена кусочно-постоянная функция с графиком, изображенным на рисунке 6.

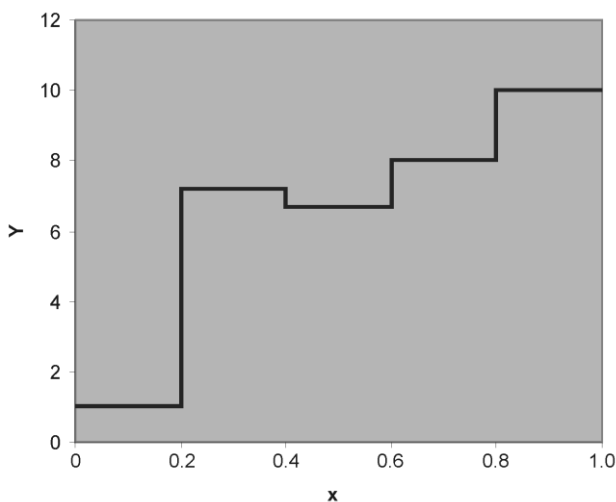


Рис. 6. График кусочно-постоянной функции

3.14 Язык PLUS

Алфавит включает алфавитно-цифровые и специальные символы. Алфавитно-цифровые символы содержат прописные буквы A...Z, строчные a...z, цифры 0...9 и знак подчеркивания «_». Алфавитно-цифровые символы используются для задания имен.

Специальные символы: «#», «*», «—», «/», «\», «,», «;», «^». Специальные символы используются для обозначения операторов и пунктуации.

Символ «#» используется для обозначения операции умножения, а символ «*» — в качестве оператора косвенной адресации СЧА GPSS.

Имена — это созданные пользователем последовательности алфавитно-цифровых символов для идентификации объектов переменных и программ.

Имя должно начинаться с буквы и может содержать от одного до 250 алфавитно-цифровых символов. Имена не должны совпадать с ключевыми словами и с СЧА GPSS.

GPSS World не различает регистр алфавита.

Поименованные величины — это имена, использующиеся для обозначения объектов или для хранения величин.

Метки объектов — поименованные величины, которые используются для обозначения операторов GPSS, определяющих объекты.

Переменные пользователя — поименованные величины, применяющиеся для присвоения значений.

Переменные пользователя могут быть локальными и глобальными. Локальные переменные пользователя объявляются внутри Plus-процедуры с помощью оператора TEMPORARY. Один оператор TEMPORARY создает одну или несколько временных именованных величин. Их можно использовать только внутри той процедуры, в которой они были определены.

Остальные переменные пользователя являются глобальными. Они могут использоваться во всей модели.

Выражения строятся с использованием операторов и вызовов процедур, соединяющих факторы. Выражения применяются в Plus-процедурах и в операндах операторов GPSS.

В выражениях переменные пользователя, элементы матриц, ячейки, параметры транзактов могут иметь значения различных типов данных (целочисленный (integer), вещественный (real) и строковый (string)).

Факторы — это основные элементы выражений. Факторами операторов GPSS являются:

- строковые константы;

- вещественные константы;
- целочисленные константы;
- имена;
- элементы Plus-матрицы;
- обращения к процедурам;
- стандартные числовые атрибуты.

Для объединения факторов в выражения используются операторы и вызовы процедур.

Plus-операторы. В отличие от операторов GPSS, которые записываются одной текстовой строкой, Plus-операторы могут записываться в несколько строк. Plus-операторы могут быть использованы только внутри оператора PROCEDURE. Plus-процедуры могут записываться в любом месте модели, кроме тела другой процедуры.

Plus-операторы могут быть простыми и составными. Каждый Plus-оператор должен заканчиваться разделителем «;» (точка с запятой). Основные Plus-операторы приведены в таблице 6.

Таблица 6. Основные Plus-операторы

Оператор	Назначение	Синтаксис
1	2	3
Присваивания	Устанавливает значение именованной величины или элемента матрицы	LValue = Expression LValue – именованная величина или элемент матрицы; Expression – Plus-выражение.
BEGIN/END	Создает блок Plus-операторов	BEGIN StatementList END; StatementList – последовательность Plus-операторов, каждый из которых должен заканчиваться «;».
PROCEDURE	Определяет Plus-процедуру	PROCEDURE Name(ArgumentList) Statement Name – имя процедуры, заданное пользователем; ArgumentList – список имен (формальных параметров), разделенных запятыми; Statement – блок Plus-операторов.

1	2	3
GOTO	Передаёт управление внутри процедуры	GOTO Label; Label – метка Plus-оператора.
IF/THEN	Проверяет выражение, если результат TRUE, выполняет блок Plus-операторов	IF (Expression) THEN Statement Expression – проверяемое Plus-выражение; Statement – блок Plus-операторов.
IF/THEN/ELSE	Проверяет выражение и в зависимости от результата выполняет один из двух блоков Plus-операторов	IF (Expression) THEN Statement1 ELSE Statement2 Expression – проверяемое Plus-выражение; Statement1 – блок Plus-операторов, выполняемый, если результат проверки TRUE; Statement2 – блок Plus-операторов, выполняемый, если результат проверки FALSE.
С меткой	Указывает место перехода для оператора GOTO	Label: Statement Label – метка Plus-оператора, задаваемая пользователем; Statement – Plus-оператор.
Вызов процедуры	Вызывает библиотечную процедуру	ProcedureName(ExpressionList) ProcedureName – имя процедуры; ExpressionList – список аргументов процедуры, разделённых запятыми (Plus-выражения).
RETURN	Останавливает выполнение процедуры и может возвращать результат её выполнения	RETURN Expression; Expression – Plus-выражение (необязательный операнд).
TEMPORARY	Определяет и ограничивает область действия локальных	TEMPORARY NameList; NameList – список имен, определённых пользователем и разделённых запятыми;

	переменных пользователя и локальных матриц	<p>TEMPORARY MATRIX</p> <p>Name[IntegerList];</p> <p>Name – имя локальной матрицы, определенной пользователем;</p> <p>IntegerList – последовательность положительных чисел от 0 до 1, разделенных запятыми – количество элементов в каждом измерении матрицы.</p>
WHILE/DO	Выполняет действие несколько раз	<p>WHILE (Expression) DO Statement</p> <p>Expression – проверяемое Plus-выражение;</p> <p>Statement – блок Plus-операторов.</p>

4 ЗАДАЧИ МОДЕЛИРОВАНИЯ СИСТЕМ ВОЗДУШНОГО ТРАНСПОРТА. СБОРНИК ЗАДАНИЙ НА ЛАБОРАТОРНЫЕ РАБОТЫ.

4.1 Задание на лабораторную работу и пример её выполнения

Лабораторная работа

Моделирование одноканальной системы массового обслуживания

Задание

Для одноканальной системы массового обслуживания с неограниченным временем ожидания определить:

- коэффициент загрузки канала обслуживания;
- среднюю длину очереди;
- среднее число требований, находящихся в системе.

Рассмотреть два случая:

А. потоки поступления требований и обслуживания простейшие (распределение интервалов времени между последовательными прибытиями требований и продолжительностей обслуживания подчинены экспоненциальному закону);

Б. потоки поступления требований и обслуживания равномерные (распределение интервалов времени между последовательными прибытиями требований и продолжительностей обслуживания подчинены равномерному закону).

Этапы выполнения лабораторной работы:

- 1) для случая А рассчитать искомые величины с помощью аналитических формул;
- 2) для случая А, с целью определения искомых величин, провести имитационное моделирование в среде GPSSW. Опре-

делить число входящих требований, моделирование которых обеспечивает ошибку по искомым величинам не более 0.01%.

3) для случая Б провести имитационное моделирование с учетом числа требований, определенных на втором этапе работы.

Для равномерного распределения считать, что минимальное и максимальное значения интервалов времени между прибытиями требований и времени обслуживания отличаются от средних значений на 20%.

Таблица 7. Заданные характеристики системы

Вариант задания	1	2	3	4	5	6	7	8
Интенсивность потока, треб/ч	6	7	8	9	10	11	14	15
Среднее время обслуживания, мин	6.00	5.50	5.00	4.00	3.75	3.00	2.50	2.00

Используемые объекты GPSS World

Блоки: *GENERATE, QUEUE, DEPART, ADVANCE, TERMINATE, SEIZE, RELEASE;*

Команды: *START;*

Операторы вероятностных распределений:

Exponential() .

Построение аналитической и имитационной моделей системы массового обслуживания

Ниже рассмотрен пример одноканальной (с одним аппаратом обслуживания) системы массового обслуживания (СМО) с ожиданием и простейшим потоком требований.

Дано: система имеет один канал обслуживания, на который поступает простейший поток заявок с интенсивностью λ . Поток обслуживаний имеет интенсивность μ . Заявка, заставшая систему занятой, «становится» в очередь и ожидает обслуживания.

Определить:

- среднюю длину очереди (среднее число требований, находящихся в очереди);
- среднее число требований, находящихся в системе (обслуживаемых и ожидающих обслуживания);

– вероятность простоя канала.

Расчетные формулы:

Интенсивность поступления требований на обслуживание:

$$\lambda = \frac{1}{t_p},$$

где t_p – среднее время между поступлениями двух смежных требований. Как известно (см. выше) для простейшего потока интервал t_p между соседними событиями имеет показательное (экспоненциальное) распределение.

Интенсивность обслуживания требований:

$$\mu = \frac{1}{t_o},$$

где t_o – среднее время обслуживания одного требования. Во многих случаях время обслуживания можно считать случайной величиной, подчиняющейся показательному распределению. На практике это означает, что большая часть требований не требует значительных временных затрат на обслуживание, и только некоторые из требований обслуживаются довольно долго.

Вероятность простоя канала:

$$P_0 = 1 - \frac{\lambda}{\mu}.$$

Среднее число требований, находящихся в системе:

$$N = \frac{\lambda/\mu}{1 - \lambda/\mu}.$$

Средняя длина очереди:

$$N_{оч} = N \frac{\lambda}{\mu}.$$

Следует отметить, что, если интервалы времени между прибытиями требований и продолжительности обслуживания не подчиняются показательному распределению, то получение простой аналитической модели, подобной приведенной выше, невозможно.

В этом случае требуется создание нового или использование имеющегося программного обеспечения для имитационного моделирования поведения такой системы.

Рассмотрим создание имитационной модели на примере одноканальной системы массового обслуживания с ожиданием с простейшими потоками поступления требований и обслуживания для варианта 1 задания.

Построение имитационной модели следует начать с создания заголовка, который может быть в виде, представленном на рисунке 7.

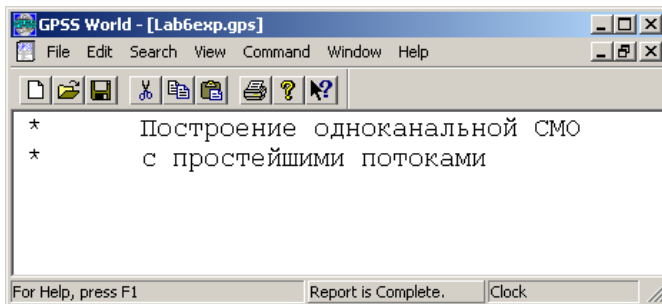


Рис. 7. Копия экрана GPSS World с заголовком модели

Поступление требования к обслуживающему каналу моделируется оператором **GENERATE**, который в рассматриваемой задаче может быть записан так:

```
GENERATE (Exponential (1, 0, 10))
```

В поле первого операнда определяется интервал времени между прибытием двух идущих одно за другим требований. Здесь используется встроенная функция экспоненциального распределения времени между прибытием двух поступающих одно за другим требований. Среднее время в данном примере составляет 10 мин – это третий аргумент функции *Exponential*.

Поступившее требование встает в очередь перед каналом. Это моделируется оператором **QUEUE**, который только в совокупности с оператором **DEPART** собирает статистическую информацию о работе моделируемой очереди.

В нашем примере оператор **QUEUE** будет выглядеть так:

```
QUEUE OCHER
```

В поле первого операнда дается символьное или числовое имя очереди. В нашем примере очереди дано имя OCHER. Желательно, чтобы присваиваемое имя отражало суть описываемого элемента системы.

Требование может выйти из очереди только тогда, когда освободится канал. Для этого вводится оператор **SEIZE**, который определяет занятость канала обслуживания, и при его освобождении находящееся впереди требование выходит из очереди и идет в канал на обслуживание. Это может выглядеть так:

```
SEIZE KANAL
```

В поле первого операнда дается символьное или числовое имя канала обслуживания. В рассматриваемом случае каналу дано имя KANAL.

Выход требования из очереди фиксируется оператором **DEPART** с соответствующим названием очереди. В примере это выглядит так:

```
DEPART OCHER
```

Далее должно быть промоделировано время обслуживания требования каналом. Для этого используется оператор **ADVANCE**, который в описываемой задаче будет выглядеть так:

```
ADVANCE (Exponential (1, 0, 6))
```

В поле первого операнда выполняется обращение к встроенному экспоненциальному распределению с указанием среднего времени обслуживания – 6 мин. После обслуживания требования каналом должно быть послано сообщение об освобождении обслуживающего канала. Это делается с помощью оператора **RELEASE**, который в рассматриваемой задаче записывается так:

RELEASE KANAL

Следует особо подчеркнуть, что парные операторы **QUEUE** и **DEPART** для каждой очереди должны иметь одно и то же, но свое уникальное имя. Это же относится и к операторам **SEIZE** и **RELEASE**.

После обслуживания требование покидает систему. Это действие моделируется оператором **TERMINATE**, который записывается в программе в таком виде:

TERMINATE 1

Число требований, прошедших обслуживание в системе, принимается равным 10000.

Копия экрана GPSS World с текстом готовой программы приведена на рисунке 8.

В случае равномерных потоков поступления требований и обслуживания описанная выше модель должна быть модифицирована следующим образом.

1) Оператор **GENERATE**, моделирующий поступление требований в систему, может быть записан так:

GENERATE 10,2

Здесь в поле первого операнда указан средний интервал времени между прибытием двух смежных требований. Для первого варианта задания он составляет 10 мин.

В поле второго операнда дано отклонение времени поступления требования от среднего, которое соответствует равномерному распределению поступления требований в систему в замкнутом интервале [8-12] с. Величина интервала определена в соответствии с заданием, требующим считать, что для равномерного распределения минимальное и максимальное значения интервалов времени между прибытиями требований отличаются от средних значений на 20%.

2) Оператор **ADVANCE**, используемый для моделирования обслуживания, может быть модифицирован следующим образом:

ADVANCE 6,1.2

Здесь назначение операндов такое же, как и в случае оператора . Время обслуживания, в соответствии с заданием, меняется в пределах 6 ± 1.2 мин.

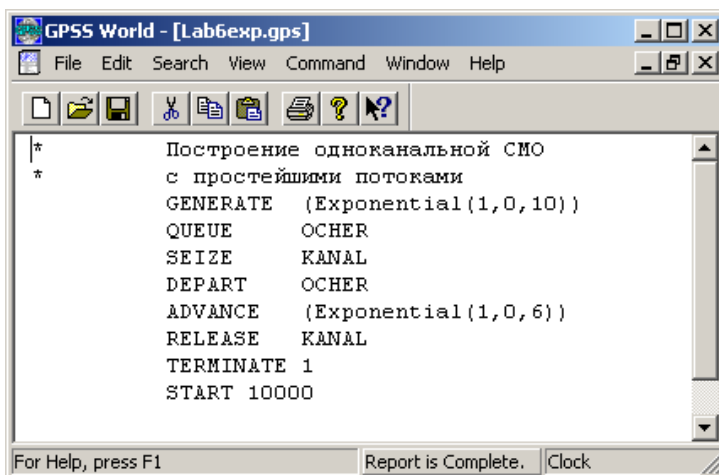


Рис. 8. Копия экрана с текстом готовой программы

Подготовка к моделированию и моделирование

Для ввода текста программы:

1) В главном меню программы GPSS World выполнить команды File – New.

2) В диалоговом окне New выбрать пункт Model. Щелкнуть по кнопке ОК.

3) В появившемся окне модели ввести текст программы.

Для задания вывода на экран необходимых параметров моделирования:

1) В главном меню программы GPSS World выполнить команды Edit – Settings.

2) Перейти во вкладку Reports/

3) Установить выходные данные, как это показано на рисунке 9.

Включенные флажки говорят о том, что эта информация будет в окне результатов моделирования. В рассматриваемом примере будет выведена информация по следующим объектам:

Facilities – Каналы обслуживания;

Queues – Очереди.

После создания имитационную модель необходимо оттранслировать и запустить на выполнение. Для этого нужно выполнить команды главного меню: Command – Create Simulation.

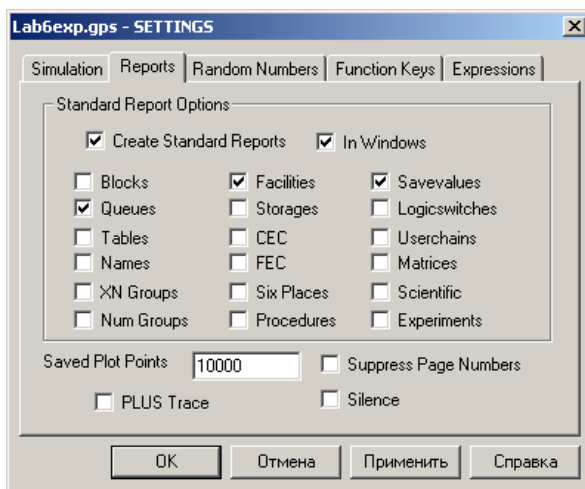


Рис. 9. Меню настройки параметров отчета с результатами моделирования

Результаты моделирования отображаются в окне REPORT. Ниже приводятся результаты только для системы с простейшими потоками (см. рисунок 10).

Расшифровка основных результатов:

START TIME – Начальное время;

END TIME – Время окончания

Результаты для канала обслуживания (FACILITY)

ENTRIES – Число входов;

UTIL. – Коэффициент использования;

AVE. TIME – Среднее время обслуживания

Результаты для очереди (QUEUE)

MAX CONT. – Максимальная длина очереди;

AVE. CONT. – Средняя длина очереди;

AVE. TIME – Среднее время ожидания в очереди;

AVE. (-0) – Среднее время ожидания в очереди тех требований, для которых это ожидание имело место.

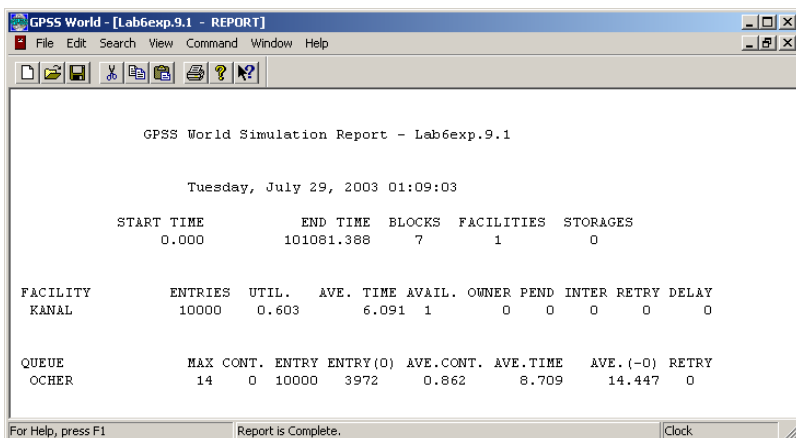


Рис. 10. Копия экрана с отчетом о результатах моделирования

4.2 Задания на лабораторные работы

Лабораторная работа.

Моделирование распределений случайных величин

Построить частотные гистограммы и определить (при необходимости) путем моделирования в GPSS World оценки математического ожидания и среднего квадратического отклонения (СКО) для следующих распределений случайной величины.

- 1) Равномерное в промежутке $5 \div 15$.
- 2) Нормальное (Гаусса) с математическим ожиданием 50 и СКО 5.
- 3) Логарифмически-нормального со значениями параметров положения (сдвига) – 0, формы – 1.0, масштаба – 2.5.
- 4) Гамма-распределения со значениями параметров положения (сдвига) – 0, формы – 4.0, масштаба – 0.5.
- 5) Вейбулла со значениями параметров положения (сдвига) – 1, формы – 1.6, масштаба – 3.
- 6) Вейбулла со значениями параметров положения (сдвига) – 5, формы – 1.7, масштаба – 12.
- 7) Вейбулла со значениями параметров положения (сдвига) – 10, формы – 1.6, масштаба – 40.

Для равномерного, логарифмически-нормального и гамма-распределений выполнить аналитический расчет математического ожидания и СКО.

Используемые объекты GPSS World

Блоки: *GENERATE*, *TERMINATE*, *TABULATE*, *ASSIGN*;

Команды: *START*, *TABLE*;

Операторы вероятностных распределений: *Gamma()*, *Lognormal()*, *Normal()*, *Weibull()*.

Лабораторная работа.
Тактическое планирование имитационного эксперимента

1. Рассчитать объем выборки N_{II} , потребный для определения:

а) оценки \bar{X} математического ожидания $\mu = M[X]$ случайной величины X , которая с заданной вероятностью ρ отличалась бы от μ не более чем на величину ε ;

б) эмпирической функции распределения $F^*(x)$ случайной величины X , значения которой $F^*(x_i)$ для заданных x_i ($i = 1, 2, \dots$) отличались бы от истинных $F(x_i)$ не более, чем на величину ε_1 с вероятностью ρ . Принять во внимание, что $F(x_i) = P\{X < x_i\}$.

Расчеты провести для случайной величины X , распределенной по логарифмически-нормальному и гамма- законам с заданными параметрами Locate, Scale, Shape.

2. Определить аналитически значения математического ожидания $\mu = M[X]$.

3. Построить в системе GPSS имитационные модели и выполнить с их использованием проверку результатов расчетов п.1. Сравнить аналитические и модельные значения $\mu = M[X]$.

Таблица 8. Варианты задания на лабораторную работу

Вар.	1	2	3	4	5	6	7	8	9	10
ρ	0,99	0,975	0,99	0,95	0,99	0,99	0,99	0,99	0,975	0,95
ε	0,05	0,025	0,025	0,02	0,02	0,05	0,025	0,025	0,02	0,02
ε_1	0,01	0,01	0,01	0,005	0,001	0,005	0,01	0,005	0,001	0,01
Логарифмически-нормальное распределение										
Locate	0	2	4	6	8	10	12	14	16	18
Scale	2,5	2,4	2,6	3	2,9	2,5	2,4	2,6	3	2,9
Shape	1	1,1	0,9	0,3	0,4	1	1,1	0,9	0,3	0,4
Гамма-распределение										
Locate	0	1	2	3	4	5	6	7	8	9
Scale	5	5.1	5.2	5.3	5.4	5.5	5.6	5.7	5.8	5.9
Shape	5	4.9	4.8	4.7	4.6	4.5	4.4	4.3	4.2	4.1

Используемые объекты GPSS World

Блоки: *GENERATE, TERMINATE, TABULATE, ASSIGN;*

Команды: *START, TABLE;*

Операторы вероятностных распределений: *Gamma () ,
Lognormal () .*

Лабораторная работа.

Имитационное моделирование работы аэродрома как системы массового обслуживания с приоритетами

Рассматривается модель работы взлетно-посадочной полосы (ВПП) аэродрома. Имеется одна ВПП, обеспечивающая взлет и посадку самолетов типов А и В. Интенсивность простейших потоков ВС на взлет составляет: ВС типа А – 3 ВС/ч, ВС типа В – 7 ВС/ч. Интенсивности простейших потоков ВС на посадку принимаются равными интенсивностям потоков ВС на взлет.

Длительности занятости ВПП при соответствующих операциях (в порядке уменьшения их приоритета) принимаются постоянными и равными:

- посадка ВС типа А – 2.5 мин;
- посадка ВС типа В – 2 мин,
- взлет ВС типа А – 3 мин,
- взлет ВС типа В – 2.5 мин.

Определить среднее время ожидания самолетами различных типов посадки и взлета.

Используемые объекты GPSS World

Блоки: *GENERATE, QUEUE, DEPART, ADVANCE, TERMINATE, SEIZE, RELEASE;*

Команды: *START;*

Операторы вероятностных распределений:
Exponential () .

Лабораторная работа.

Имитационное моделирование системы технического обслуживания самолетов как многоканальной СМО с ожиданием при произвольном распределении длительности обслуживания

Рассматривается процесс технического обслуживания (ТО) самолетов в аэропорту, располагающем 7 бригадами ТО. Поток самолетов, поступающих на ТО, простейший с интенсивностью 2 ВС/ч. Найти среднее время ожидания (простоя самолета в ожидании ТО) для трех законов распределения времени ТО самолета одной бригадой:

- 1) показательного со средним $\bar{t}_{обсл} = 3ч$;
- 2) равномерного с параметрами $\bar{t}_{обсл} = 3ч, \sigma = 1ч$;
- 3) треугольного равнобедренного с параметрами $\bar{t}_{обсл} = 3ч, \sigma = 1ч$.

Построить гистограммы распределений:

- времени ТО;
- времени ожидания ТО;
- времени пребывания самолета в подсистеме ТО.

Сделать выводы о влиянии характера распределения времени обслуживания на время ожидания. Сделать выводы о влиянии числа прогонов модели на точность результатов.

Используемые объекты GPSS World

Блоки: GENERATE, ASSIGN, QUEUE, DEPART, ENTER, LEAVE, ADVANCE, TERMINATE, TABULATE;

Команды: START, TABLE, QTABLE, STORAGE;

Операторы вероятностных распределений:

Exponential(), Triangular(), Uniform().

Для справки: среднее время ожидания, полученное по формулам ТМО, составляет:

- для показательного распределения $\bar{\tau} \approx 1.84ч$;
- для других распределений $\bar{\tau} \approx 1.02ч$.

Лабораторная работа.

Определение временных характеристик стоянки в аэропорту самолета, выполняющего транзитный рейс

На время вылета из аэропорта транзита ВС, выполняющего транзитный рейс, влияют величины продолжительности технологических операций, которые могут быть объединены в четыре группы:

Группа 1. Операции, выполняемые по прилету (постановка на МС; открытие дверей, люков; высадка пассажиров; выгрузка багажа, грузов и др.);

Группа 2. Заправка ВС авиатопливом.

Группа 3. Подготовка ВС к посадке пассажиров (заправка спецжидкостями, газами; уборка салона, кабины экипажа; транзитное ТО и др.);

Группа 4. Операции, выполняемые перед вылетом (погрузка багажа, грузов; посадка пассажиров; закрытие дверей, люков и др.).

В соответствии с требованиями технологического графика наземного обслуживания ВС, выполняющего транзитный рейс, операции групп 2 и 3 выполняются параллельно и только по завершении операций группы 1. Операции группы 4 выполняются по окончании выполнения операций всех прочих групп.

Продолжительности каждой группы операций заданы как случайные величины с треугольной функцией распределения:

$\Delta t_1 = (18; 20; 23)$ мин; $\Delta t_2 = (20; 23; 30)$ мин; $\Delta t_3 = (18; 22; 32)$ мин; $\Delta t_4 = (20; 22; 28)$ мин.

Время прилета согласно плану полетов $t_{II} = 10 : 00$.

Определить плановое время готовности ВС к вылету как 95%-квантиль времени готовности ВС к вылету.

Используемые объекты GPSS World

Блоки: *GENERATE, QUEUE, DEPART, ENTER, LEAVE, ADVANCE, TERMINATE, SPLIT, ASSEMBLE, MARK, TABULATE;*

Команды: *START, TABLE, STORAGE;*

Операторы вероятностных распределений: *Triangular()*.

Лабораторная работа.

Определение временных характеристик стоянки в аэропорту самолета, выполняющего транзитный рейс, с учетом аэровокзальных операций обслуживания пассажиров

На время вылета из аэропорта транзита ВС, выполняющего транзитный рейс, влияют величины продолжительностей технологических операций, которые могут быть объединены в пять групп:

Группа 1. Операции, выполняемые по прилету.

Группа 2. Заправка ВС авиатопливом.

Группа 3. Подготовка ВС к посадке пассажиров.

Группа 4. Операции, выполняемые перед вылетом.

Группа 5. Операции по аэровокзальному обслуживанию вылетающих пассажиров.

В соответствии с требованиями технологического графика наземного обслуживания ВС, выполняющего транзитный рейс, операции групп 2 и 3 выполняются параллельно и только по завершении операций группы 1. Операции группы 5 могут выполняться параллельно с операциями первых трех групп. Операции группы 4 выполняются по окончании выполнения операций всех прочих групп.

Продолжительности каждой группы операций заданы как случайные величины с треугольной функцией распределения:

$\Delta t_1 = (18; 20; 23)$ мин; $\Delta t_2 = (20; 23; 30)$ мин; $\Delta t_3 = (18; 22; 32)$ мин; $\Delta t_4 = (20; 22; 28)$ мин; $\Delta t_5 = (24; 35; 51)$ мин.

Время прилета согласно плану полетов $t_{\Pi} = 10:00$.

Определить плановое время готовности ВС к вылету к вылету как 95%-квантиль времени готовности ВС к вылету и плановое время начала регистрации, минимизирующее плановое время вылета.

Используемые объекты GPSS World

Блоки: *GENERATE, QUEUE, DEPART, ENTER, LEAVE, ADVANCE, TERMINATE, SPLIT, ASSEMBLE, MARK, TABULATE, MATCH;*

Команды: *START, TABLE, STORAGE;*

Операторы вероятностных распределений: *Triangular()* .

Лабораторная работа.

Применение PLUS-операторов для сокращения текста программы имитационной модели

Рассматривается одноканальная СМО, в которую поступают потоки заявок двух типов, имеющие одинаковый приоритет. Как поток заявок первого типа, так и поток заявок второго типа являются простейшими. Средний интервал времени между поступлениями в СМО заявок первого типа составляет 180 единиц модельного времени (ЕМВ), второго типа – 360 ЕМВ. Время обслуживания заявок первого и второго типов фиксировано и составляет соответственно 60 и 90 ЕМВ. Построить гистограмму времени ожидания заявками обслуживания вне зависимости от их типа.

Разработать два варианта модели: 1) без использования операторов языка PLUS и 2) с использованием PLUS-операторов с целью сокращения программы.

Используемые объекты GPSS World

Блоки: *GENERATE, QUEUE, DEPART, SEIZE, RELEASE, ADVANCE, TERMINATE, ASSIGN;*

Команды: *START, QTABLE;*

Операторы вероятностных распределений:

Exponential(), RN1;

Операторы языка PLUS: *Procedure, BEGIN END, TEMPORARY, RETURN, IF THEN, RN1;*

Для справки: поток, полученный суперпозицией пуассоновских потоков, является пуассоновским.

Лабораторная работа.
Применение функций

Задание 1

Рассматривается одноканальная СМО, в которую поступает простейший поток заявок интенсивностью 0.1 1/мин. В результате наблюдений получено следующее распределение времени обслуживания:

Промежуток времени	Частота
менее 1 мин.	0
1 – 2 мин.	0.1
2 – 3 мин.	0.3
3 – 4 мин.	0.4
4 – 5 мин.	0.1
5 – 6 мин.	0.08
6 – 10 мин.	0.02
более 10 мин.	0

Выполнить моделирование обслуживания 100 тыс. заявок. Построить гистограмму распределения времени обслуживания. Определить основные характеристики ожидания заявок в очереди.

Используемые объекты GPSS World

Блоки: *GENERATE, QUEUE, DEPART, SEIZE, RELEASE, ADVANCE, TERMINATE, TABULATE, ASSIGN;*

Команды: *START, TABLE, FUNCTION;*

Операторы вероятностных распределений:

Exponential(), RN1;

Задание 2

Рассматривается региональный узловой аэропорт, прибытие самолетов в который носит характер периодически повторяющихся волн. Интервал между волнами составляет 6 ч. Волна начинается с промежутка с линейным ростом интенсивности

потока самолетов с 2 ВС/ч до 10 ВС/ч, продолжающимся 1 ч., который сменяется промежутком с таким же линейным уменьшением интенсивности с 10 ВС/ч до 2 ВС/ч, продолжающимся также 1 ч. В оставшееся время интенсивность потока самолетов остается на уровне 2 ВС/ч. В аэропорту 2 ВПП, одна из которых работает на прием, вторая – на выпуск самолетов. Построить распределение времени ожидания посадки, если ее продолжительность принимается равной 2 мин. вне зависимости от типа ВС. Провести моделирование работы ВПП в течении 100 суток. Поток прибывающих ВС считать пуассоновским.

Используемые объекты GPSS World

Блоки: *GENERATE, QUEUE, DEPART, SEIZE, RELEASE, ADVANCE, TERMINATE;*

Команды: *START, QTABLE, FUNCTION;*

Операторы вероятностных распределений:

Exponential ();

Задание 3

Рассматривается процесс регистрации вылетающих пассажиров одного рейса. Считать, что число пассажиров рассматриваемого рейса составляет 100 человек. Наблюдениями установлено, что пассажиры прибывают в аэропорт для регистрации в течение временного промежутка, начинающегося за 3.5 ч до ее окончания. Вероятностное распределение времени прибытия пассажиров представлено в таблице, где τ – время от момента прибытия пассажира до момента окончания регистрации.

Построить распределение времени ожидания пассажирами обслуживания при использовании одной, двух и трех стоек регистрации. Провести моделирование обслуживания 100 рейсов. Считать, что стойки начинают работу за 3.5 ч до планового времени окончания регистрации. Время обслуживания пассажира считать распределенным показательным со средним 1 мин. Определить продолжительность занятости стоек регистрацией пассажиров каждого рейса.

Используемые объекты GPSS World

Блоки: *GENERATE, MARK, SPLIT, QUEUE, DEPART, ENTER, LEAVE, ADVANCE, TERMINATE, ASSEMBLE, SAVEVALUE;*

Команды: *START, QTABLE, FUNCTION, STORAGE, EQU, INITIAL;*

Операторы вероятностных распределений:

Exponential(), RN1;

Таблица 9. Распределение времени прибытия пассажира в аэропорт

i	τ_{i-1} , мин	τ_i , мин	P_i
1	0	15	0.001
2	15	30	0.004
3	30	45	0.015
4	45	60	0.160
5	60	75	0.232
6	75	90	0.201
7	90	105	0.155
8	105	120	0.113
9	120	135	0.062
10	135	150	0.026
11	150	165	0.015
12	165	180	0.010
13	180	195	0.005
14	195	210	0.001

Лабораторная работа.
Детализованная модель наземного обслуживания ВС,
выполняющего транзитный рейс

На время вылета из аэропорта транзита ВС, выполняющего транзитный рейс, влияют величины продолжительности технологических операций, которые могут быть объединены в пять групп:

Группа 1. Операции, выполняемые по прилету (постановка на МС; открытие дверей, люков; высадка пассажиров; выгрузка багажа, грузов и др.);

Группа 2. Заправка ВС авиатопливом.

Группа 3. Подготовка ВС к посадке пассажиров (заправка спецжидкостями, газами; уборка салона, кабины экипажа; транзитное ТО и др.);

Группа 4. Операции, выполняемые перед вылетом (погрузка багажа, грузов; посадка пассажиров; закрытие дверей, люков и др.).

Группа 5. Операции по аэровокзальному обслуживанию вылетающих пассажиров (регистрация, предполетный досмотр и др.).

В соответствии с требованиями технологического графика наземного обслуживания ВС, выполняющего транзитный рейс, операции групп 2 и 3 выполняются параллельно и только по завершении операций группы 1. Операции группы 5 могут выполняться параллельно с операциями первых трех групп. Операции группы 4 выполняются по окончании выполнения операций всех прочих групп.

Продолжительности групп операций 1, 3, 4 заданы как случайные величины с треугольной функцией распределения:

$\Delta t_1 = (18; 20; 23)$ мин; $\Delta t_3 = (18; 22; 32)$ мин; $\Delta t_4 = (20; 22; 28)$ мин.

Время заправки ВС (Группа 2) в случае использования одного средства заправки (АТЗ) определяется как случайная величина с треугольной функцией распределения $\Delta t_2 = (20; 23; 30)$ мин;. Если фактически ожидаемое время заправки может превысить модальное значение (23 мин.), то для обслуживания ВС

выделяются 2 АТЗ, благодаря чему время заправки сокращается в 2 раза.

Для определения времени окончания регистрации пассажиров и багажа (Группа 5) рассмотреть обслуживание каждого отдельного пассажира. Исходные данные по данной операции принять из условия Задания 3 предыдущей лабораторной работы.

Время прилета согласно плану полетов $t_{II} = 10:00$.

Определить плановое время готовности ВС к вылету к вылету как 95%-квантиль времени готовности ВС к вылету и плановое время начала интервала поступления пассажиров на регистрацию, минимизирующее плановое время вылета.

Используемые объекты GPSS World

Блоки: *GENERATE, QUEUE, DEPART, ENTER, LEAVE, ADVANCE, TERMINATE, SPLIT, ASSEMBLE, MARK, TABULATE, MATCH, TRANSFER, TEST;*

Команды: *START, TABLE, QTABLE, STORAGE;*

Операторы вероятностных распределений: *Triangular().*

СПИСОК ЛИТЕРАТУРЫ

1. Андронов А.М., Хижняк А.Н. Математические методы планирования и управления производственно-хозяйственной деятельностью предприятий гражданской авиации. – М.: «Транспорт», 1977. – 215 с.
2. Боев В.Д. Моделирование систем. Инструментальные средства GPSS World: Учеб. пособие. – СПб.: БХВ-Петербург, 2004. – 386 с.
3. Вентцель Е.С. Теория вероятностей: Учебник для студ. вузов. – 10-е изд. – М.: Издательский центр «Академия», 2005. – 576 с.
4. Кудрявцев Е.М. GPSS World. Основы имитационного моделирования различных систем.. – М.: ДМК Пресс, 2004. – 320 с.
5. Лоу А.М., Кельтон В.Д. Имитационное моделирование. – 3-е изд. – СПб.: Питер, 2004. – 847 с.
6. Шеннон Р. Имитационное моделирование систем – искусство и наука. – М.: Мир, 1978. – 420 с.