Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»

Кафедра «Информационные технологии»

Т. А. Трохова

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ ТЕХНИЧЕСКИХ СИСТЕМ

ПОСОБИЕ
по одноименному курсу для студентов специальности 1-40 01 02 «Информационные системы и технологии (по направлениям)» дневной формы обучения

Электронный аналог печатного издания

Гомель 2011

УДК 681.3.06(075.8)
ББК 32.965в631я73
Т76

Рекомендовано к изданию научно-методическим советом факультета автоматизированных и информационных систем ГГТУ им. П. О. Сухого (*протокол № 10 от 28.06.2010 г.*)

Рецензент: нач. сектора программных средств АСУ центра информационных технологий ГГТУ им. П. О. Сухого *Н. С. Шестакова*

Трохова, Т. А.

Т76 Математическое моделирование и автоматизированное проектирование технических систем : пособие по одноим. курсу для студентов специальности 1-40 01 02 «Информационные системы и технологии (по направлениям)» днев. формы обучения / Т. А. Трохова. – Гомель : ГГТУ им. П. О. Сухого, 2011. – 45 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: http://lib.gstu.local. – Загл. с титул. экрана.

ISBN 978-985-420-979-1.

Содержит теоретические сведения по математическому и компьютерному моделированию технических систем, практические примеры различных видов моделей и их реализации в системах компьютерной математики.

Для студентов специальности 1-40 01 02 «Информационные системы и технологии (по направлениям)» дневной формы обучения.

УДК 681.3.06(075.8)
ББК 32.965в631я73

ISBN 978-985-420-979-1

1. ВВЕДЕНИЕ В МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

1.1. Основные понятия теории моделирования

Моделирование является неотъемлемой частью инженерного проектирования. Разработка систем автоматизации проектирования (САПР) включает математическое и компьютерное моделирование как один из видов обеспечения – математическое обеспечение.

Применение компьютерного моделирования обосновывается несколькими причинами. При создании технических устройств и систем обычно рассматривают несколько вариантов различных решений, которые называются альтернативами. Выбор нужного технического решения из набора альтернатив выполнялся на стадии экспериментальной отработки технических устройств. По мере усложнения и удорожания устройств росли и затраты на эксперимент. Достоверность экспериментальных данных была не совсем надежной. Возникла необходимость повышения достоверности данных при отборе альтернатив на начальной стадии проектирования. Вычислительный эксперимент с моделью еще не существующей системы позволил проводить более тщательный анализ предлагаемых проектных решений, чем при традиционном проектировании, когда большинство ошибок и несоответствий требованиям обнаруживалось на этапе эксперимента с опытным образцом.

Прорыв в совершенствовании вычислительной техники являлся материальной базой для быстрого развития математического моделирования и вычислительного эксперимента. Вычислительный эксперимент позволил снизить затраты на проведение натурных аэродинамических испытаний созданного в США аэробуса. Математическое моделирование и вычислительный эксперимент – ведущие методологии изучения глобальных моделей процессов и явлений на Земле, например климата Земли. Проведение работ по глобальному моделированию стимулировалось деятельностью Римского клуба, неправительственной организации. Первую из таких моделей опубликовал в 1971 г. американский специалист по теории управления Д. Форрестер. Компьютерные игры, проведенные Д. Форрестером с глобальной моделью, показали, что в сер. XXI в. человечество ждет кризис, связанный прежде всего с истощением природных ресурсов, падением численности населения и производства продуктов, ростом загрязнения окружающей среды. Известны результаты глобального моделирования явления «ядерной зимы», выполненные в ВЦ АН СССР В. В. Александровым и Г. Л. Стенчиковым под руководством академика Н. Н. Моисеева. Эти результаты дали человечеству, в том числе политикам, неопровержимые аргументы против ядерной войны, даже так называемой «ограниченной ядерной войны».

В настоящее время моделирование и компьютерный эксперимент позволили объединить формальное и неформальное мышление и естественным образом сочетать способности компьютера и человеческого интеллекта.

Ниже приведены основные определения теории моделирования.

Модель – это физическая или абстрактная система, адекватно (правдоподобно) представляющая объект исследования или проектирования.

Объект, с целью изучения которого проводятся исследования, называется оригиналом, а объект, исследуемый вместо оригинала для изучения определенных свойств, называется моделью.

В качестве моделей могут выбираться естественные объекты, обладающие свойствами, подобными соответствующим свойствам оригинала, или же создаваться специальные искусственные объекты с нужными свойствами.

Моделирование – исследование каких-либо явлений, процессов или систем объектов путем построения и изучения их моделей; использование моделей для определения или уточнения характеристик вновь конструируемых объектов.

Математическая модель – это совокупность математических объектов и связей между ними, отражающих основные свойства проектируемого технического объекта.

Абстрагируясь от физической природы объекта, математическая модель описывает идеальные объекты. Например, с помощью теории дифференциальных уравнений можно изучить электрические и механические колебания в наиболее общем виде, а затем полученные знания применить для исследования объектов конкретной физической природы.

Компьютерная модель – это программная реализация математической модели. Компьютерная модель имеет две составляющие – программную и аппаратную. Компьютерная модель проявляет свойства физической модели, когда ее абстрактная составляющая – программа – интерпретируется физическим устройством – компьютером. Компьютерная модель как физическое устройство может входить в состав испытательных стендов, тренажеров, виртуальных лабораторий.

1.2. Описание процесса моделирования

На рис. 1.1 приведена схема процесса моделирования технических систем, в которой указаны такие основные этапы моделирования, как составление формальной модели, исследование свойств модели, проведение экспериментов по модели и т. д. [3].

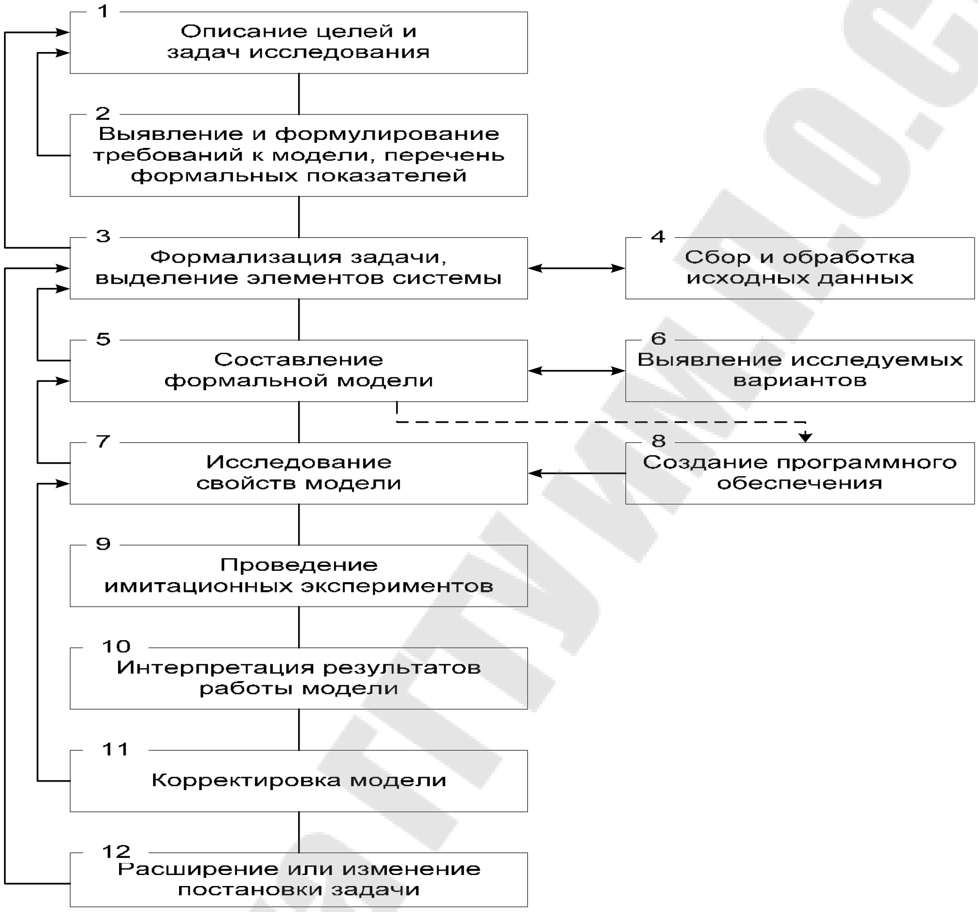


Рис. 1.1. Процесс моделирования технических систем

1.3. Понятие технической системы, параметры системы

Технический объект может иметь несколько видов описаний в зависимости от точки зрения на него. Если описание объекта составляется с точки зрения пользователя, то объект может быть представлен в виде «черного ящика», т. е. структура (внутреннее содержание) объекта пользователю неизвестно. Если технический объект рассматривается с точки зрения разработчика, то необходимо знать его составные элементы и связи между ними, т. е. внутреннюю структуру объекта. Такое описание объекта называется внешним и позволяет рассматривать объект как систему.

Любой технический объект различной физической природы (электрический, механический, гидравлический и т. д.) можно рассматривать как систему.

Система – это совокупность связанных элементов, объединенных в одно целое для достижения определенной цели.

Чтобы разобраться в системе, изучить, исследовать ее (задача анализа), надо описать систему, зафиксировать ее свойства, поведение, структуру и параметры, т. е. построить одну или несколько моделей.

Для этого надо ответить на три основных вопроса:

1. Что она делает (узнать поведение, функцию системы)?

2. Как она устроена (выяснить структуру системы)?

3. Каково ее качество (насколько хорошо она выполняет свои функции)?

На рис. 1.2 приведена схема видов описаний системы: функционального, структурного и параметрического [4].

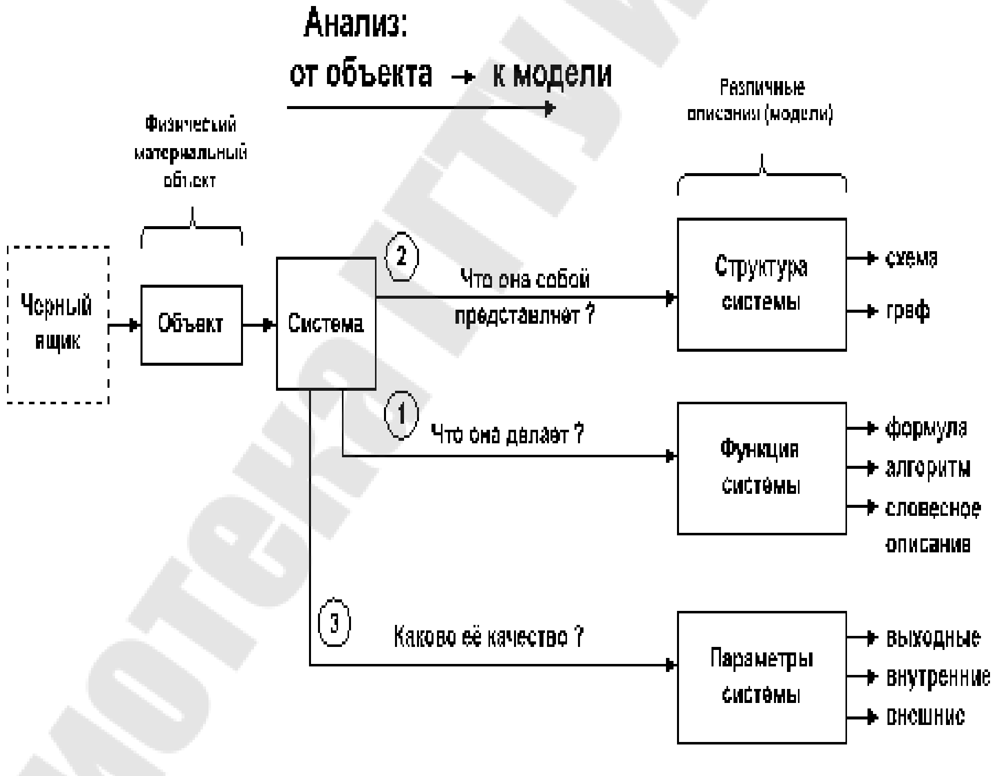


Рис. 1.2. Схема видов описаний системы

Параметры системы – это величины, характеризующие качество, свойства или режимы работы объекта.

Различают выходные, внутренние, внешние параметры.

Выходные параметры – это показатели качества системы. По ним можно судить о правильности функционирования системы и ее качестве. Они позволяют сравнивать однотипные по назначению системы, сделать выбор подходящего варианта.

Выходных параметров обычно много, и их принято представлять вектором

$$Y = (y1,\ y2,\ ...,\ yn).$$

Внутренние параметры – это параметры структурных (внутренних) элементов системы:

$$X = (x1,\ x2,\ ...,\ xn).$$

Внешние параметры – это параметры внешней среды, оказывающие влияние (обычно отрицательное) на функционирование системы:

$$Q = (q1,\ q2,\ ...,\ qn).$$

Параметры входных сигналов иногда выделяют в отдельную группу и называют *входными параметрами*:

$$I = (i1,\ i2,\ ...,\ in).$$

Выходные параметры объекта (а значит, и его качество) зависят от входных воздействий, параметров внешней среды и, конечно же, от качества составляющих объект элементов (X-параметров).

Такая зависимость представляется в аналитической форме и называется глобальной функцией объекта (оператором) W.

В случае черного ящика мы ничего не знаем о составляющих объект элементах, то есть нам неизвестны его X-параметры. Поэтому глобальная функция объекта записывается в упрощенном виде, как его реакция на внешние воздействия I и Q:

$$Y = Wc\,(I,\ Q) - \textit{черный ящик}.$$

Если объект рассматривается как система, то его X-параметры известны, следовательно:

$$Y = Wc\,(I,\ Q,\ X) - \textit{система}.$$

Для динамических объектов в глобальную функцию добавляется еще одна координата – время t.

$$Y = Wc\,(I,\ Q,\ t)\ -\ \textit{черный ящик}.$$

$$Y = Wc(I,\ Q,\ X,\ t) - \textit{система}.$$

Существование глобальной функции еще не означает, что она известна исследователю или проектировщику объекта. Задача как раз и заключается в том, чтобы отыскать эту функцию.

1.4. Задачи моделирования

С помощью моделирования решаются две глобальные задачи:

1) исследование (изучение, анализ) естественных материальных объектов и процессов;

2) проектирование (разработка, синтез) искусственных материальных объектов и процессов.

Задачи синтеза делятся на две группы:

– синтез структуры технических систем (структурный синтез);

– синтез параметров технических систем (параметрический синтез).

В процессе исследования или проектирования могут ставиться более конкретные цели и задачи, такие, как:

– выбор оптимального варианта решения;

– усовершенствование базового варианта решения;

– прогноз;

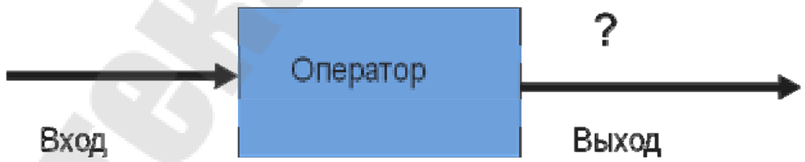
– планирование;

– управление;

– обучение.

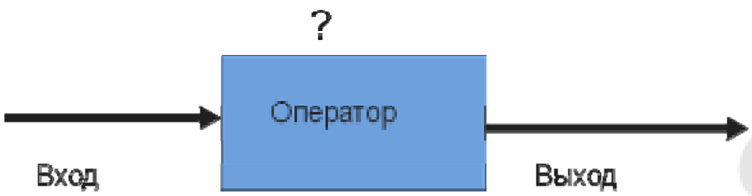
Ниже приведено описание различных задач моделирования.

Задачи анализа: по заданному входному воздействию и оператору системы исследовать закон изменения выходного сигнала.

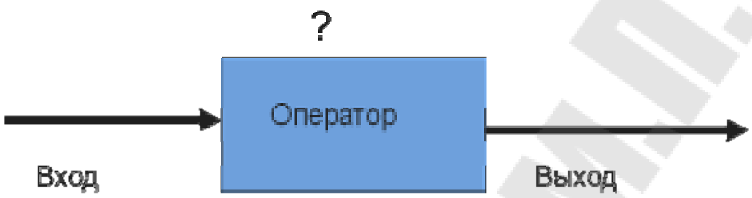


Задачи синтеза: по желаемому выходу найти входной сигнал и оператор системы (неопределенные параметры оператора).

Задачи идентификации: по заданному входному воздействию и выходному сигналу определить оператор системы.



Обратная задача получает входной сигнал при заданном выходном сигнале и операторе системы.



1.5. Классификация моделей. Принципы построения, требования к моделям

Модели можно классифицировать многими способами, однако ни один их них не является полностью удовлетворительным.

Прежде всего следует различать физические и абстрактные модели.

Физическая модель – это материально реализованная система, например: макет, тренажер или экспериментальный образец.

Абстрактная модель – это описание объекта проектирования или исследования на каком-либо языке, например: график, чертеж, схема, граф, таблица, формула, граф- или блок-схема алгоритма, программа для ЭВМ или словесное описание.

Разновидностью абстрактных моделей является *математическая модель*. Такая модель представляет собой описание объекта на языке математических отношений. Математические модели в свою очередь делятся на аналитические и имитационные.

Аналитические модели представляют собой уравнения или системы уравнений. Они записаны и решены в буквенном виде. Отсюда и происходит их название.

Имитационная модель представляет собой алгоритм (процедуру), которым описывается поведение соответствующего ей объекта или способ вычисления его выходных сигналов.

Статические модели описывают статические состояния, в них не присутствует время в качестве независимой переменной.

Динамические модели отражают поведение системы, т. е. в них обязательно используется время.

Стохастические и детерминированные модели различаются в зависимости от учета или неучета случайных факторов.

Каждая модель строится на следующих принципах.

1. Принцип полноты. Модель должна быть полной, чтобы предоставлять экспериментатору различные наборы характеристик.

2. Принцип изменяемости. Модель должна быть достаточно гибкой, чтобы имелась возможность воспроизводить различные ситуации.

3. Принцип модульности. Модель должна быть блочной, т. е. допускается возможность замены, добавления и исключения некоторых частей без переделки модели.

4. Принцип адекватности. Модель должна допускать возможность выбора необходимой точности задания ее параметров. Адекватность – это воспроизведение моделью с необходимой полнотой и точностью всех свойств объекта, существенных для целей данного исследования.

5. Принцип эффективности. Модель должна обеспечивать эффективное (по быстродействию и памяти) функционирование программы модели, удобство ее использования.

Модель должна соответствовать следующим требованиям:

– должна быть достаточно простой, для оптимизации времени ее создания (иначе при длительном процессе ее создания она может стать практически бесполезной);

– должна обеспечиваться возможность работы модели с банком данных системы;

– должна обеспечиваться возможность проведения целенаправленных экспериментов с использованием опытных данных для оценки адекватности модели;

– допускать изменения системы, если результаты моделирования существенно отличаются от результатов физического эксперимента.

Наиболее важными являются требования точности, экономичности и универсальности. Они противоречивы, например, повышение точности модели делает ее сложнее, а значит, и менее экономичной. Поэтому на практике приходится довольствоваться компромиссными решениями.

Точность модели – это количественная оценка степени совпадения модельных результатов с натурными.

Точность модели тесно связана с понятием «адекватность». Но это не синонимы: понятие «адекватность» носит качественный характер, тогда как за понятием «точность» стоит число, количественная оценка модели.

Для предварительных оценок на этапе поискового проектирования простые модели можно считать адекватными объекту, но при детализации проекта они теряют это свойство и становятся слишком «грубыми». А ведь это те же самые модели с одной и той же точностью. Но теперь она (точность) становится недостаточной.

Количественная оценка точности модели – задача непростая даже для опытного исследователя. Дело в том, что реальные объекты характеризуются не одним, а несколькими выходными параметрами.

В модели выходные параметры могут представляться с различной погрешностью – одни упрощенно, другие точно. Отсюда вытекает первоначальный векторный характер оценки и необходимость сведения ее к скалярной величине. В противном случае трудно говорить о качестве моделей вообще и сравнивать их между собой.

Прежде чем получить точную модель, надо сначала добиться, чтобы она была «правильной», т. е. не давала абсурдных результатов.

Для оценки правильности модели используются простые приемы, такие, как:

– проверка физического смысла (соблюдение физических законов);

– проверка размерности и знаков;

– проверка пределов;

– проверка тренда, т. е. тенденции изменения выходных параметров в зависимости от внутренних и внешних.

Убедившись, что модель работает правильно, можно попытаться довести ее до требуемой точности. Эта работа называется калибровкой (подгонкой) модели и состоит в том, что в базовый (грубый) вариант модели добавляются детали и используются уже известные нам методы улучшения, пока модель не достигнет желаемого качества (необходимой точности).

Универсальность моделей определяет область их возможных применений.

Можно построить модель транзистора, пригодную для анализа цифровых схем, но совершенно неприемлемую для аналоговых схем.

Понятно, что такие модели будут просты и удобны для частных задач, но универсальностью они не обладают.

Экономичность моделей оценивается затратами на вычислительные ресурсы (машинное время и память), необходимые для реализации модели на компьютере. Косвенным показателем экономичности математической модели служит также количество внутренних параметров, используемых в ней. Чем их больше, тем выше требования к оперативной и дисковой памяти, тем длительнее будет их обработка. Наконец, чем больше параметров, тем больше времени потребуется для отыскания сведений об их численных значениях.

2. ИНСТРУМЕНТАРИЙ МОДЕЛИРОВАНИЯ. СРАВНИТЕЛЬНЫЙ АНАЛИЗ СИСТЕМ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ

2.1. Системы визуального моделирования

Все современные универсальные прикладные системы, которые могут служить инструментом компьютерного моделирования, можно условно разделить на:

– системы компьютерной математики (СКМ);

– системы визуального моделирования.

На сегодня существует огромное число пакетов визуального моделирования. В них пользователю предоставляется возможность описывать моделируемую систему преимущественно в визуальной форме, графически представляя как структуру системы, так и ее поведение. Такой подход позволяет пользователю не заботиться о реальной программной реализации модели, что значительно упрощает процесс моделирования.

Результаты эксперимента в пакетах визуального моделирования представляются в более наглядной для человека форме: в виде графиков, гистограмм, схем, с применением анимации и т. д. Также в той или иной мере поддерживается технология объектно-ориентированного моделирования, что позволяет повторно использовать экземпляры моделей с возможностью внесения в них тех или иных корректив.

Из множества существующих на сегодняшний день пакетов визуального моделирования особый интерес вызывают универсальные пакеты, не ориентированные на определенную узкоспециальную область (физика, химия, электроника и т. д.) или определенные типы моделей (чисто дискретные или чисто непрерывные).

Такие пакеты позволяют моделировать принадлежащие различным прикладным областям структурно сложные гибридные системы.

Современные универсальные пакеты визуального моделирования можно разделить на три основные группы:

– пакеты, использующие язык блочного моделирования;

– пакеты, использующие язык физического моделирования;

– пакеты, ориентированные на использование схемы гибридного автомата.

Пакеты, принадлежащие к первой группе, используют графический язык иерархических блок-схем.

Концепция работы пакетов строится на следующих принципах.

Блок высшего уровня иерархии собирается из некоторого набора стандартных блоков (созданных ранее разработчиками пакета, либо написанных самим пользователем), соединяемых однонаправленными функциональными связями. Собранную функциональную схему можно использовать как блок на следующем уровне иерархии и можно запомнить в библиотеке блоков.

В число стандартных блоков входят блоки с чисто непрерывным, чисто дискретным и гибридным поведением. Каждый блок автоматически формирует программу на внутреннем языке системы, которая запускается на выполнение при запуске имитационной модели.

К достоинствам этого подхода следует отнести, прежде всего, чрезвычайную простоту создания не очень сложных моделей даже не слишком подготовленным пользователем. Недостаток: при создании сложных моделей приходится строить довольно громоздкие многоуровневые блок-схемы, не отражающие естественной структуры моделируемой системы, что осложняет процесс моделирования.

Наиболее известными представителями первой группы являются:

– пакет *Simulink* системы *Matlab*;

– пакет *EASY*5;

– подсистема *SystemBuild* пакета *MATRIXx*;

– *VisSim*.

VisSim (*фирма Visual Solutions*, *www.vissim.com*) – это диалоговая визуальная оболочка для разработки непрерывных, дискретных, мультичастотных и гибридных моделей систем и моделирования динамики этих систем. Набор команд, предоставляемый *VisSim*, позволяет автоматизировать решения многих задач.

В *VisSim* модель системы строится в виде структурной схемы в привычном виде. Основными инструментами являются функциональные блоки и связи между ними. Каждый блок выполняет определенную функцию. *VisSim* содержит более 100 линейных и нелинейных блоков, позволяющих моделировать сколь угодно сложные системы. Моделирование системы в *VisSim* означает пошаговое решение уравнений, описывающих данную систему, и вычисление выходов модели. Если изменить параметры системы во время процесса моделирования, *VisSim* немедленно пересчитает параметры системы и учтет их при моделировании. Для решения дифференциальных и разностных уравнений *VisSim* использует семь различных методов интегрирования, а также вычислители для «жестких» систем.

Выбор методов интегрирования позволяет найти компромисс между скоростью и точностью вычислений.

Пример модели в *VisSim* приведен на рис. 2.1.

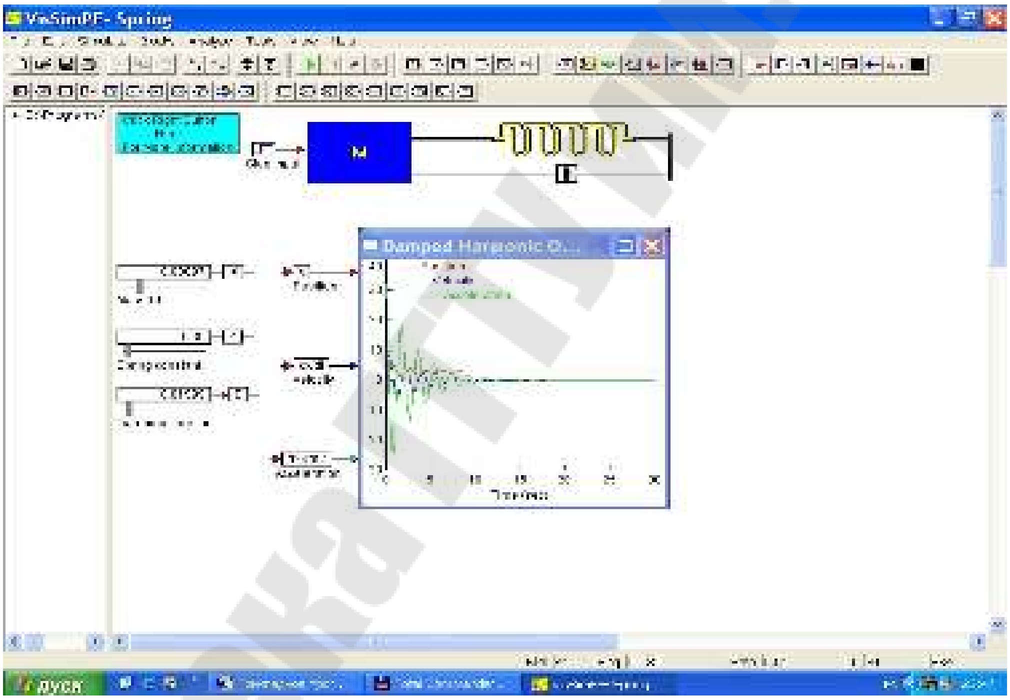


Рис. 2.1. Пример модели в *VisSim*

Пакет *Simulink* является приложением к системе *Matlab*. При моделировании с использованием *Simulink* реализуется принцип визуального программирования, в соответствии с которым пользователь из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты.

Simulink является автономным пакетом *Matlab*. Модели, созданные в *Simulink*, построены на основе внутреннего языка *Matlab*, которые могут быть откорректированы в *Matlab* и данные из *Matlab* могут быть переданы для обработки в *Simulink*. Часть входящих в состав пакетов имеет инструменты, встраиваемые в *Simulink* (например, *LTI-Viewer* приложения *Control System Toolbox* – пакета для разработки систем управления). Имеются также дополнительные библиотеки блоков для разных областей применения (например, *Power System Blockset* – моделирование электротехнических устройств, *Digital Signal Processing Blockset* – набор блоков для разработки цифровых устройств и т. д.).

При работе с *Simulink* пользователь имеет возможность модернизировать библиотечные блоки, создавать свои собственные, а также составлять новые библиотеки блоков.

При моделировании пользователь может выбирать метод решения дифференциальных уравнений, а также способ изменения модельного времени (с фиксированным или переменным шагом). В ходе моделирования имеется возможность следить за процессами, происходящими в системе. Для этого используются специальные устройства наблюдения, входящие в состав библиотеки *Simulink*. Результаты моделирования могут быть представлены в виде графиков или таблиц.

На рис. 2.2 приведена модель в пакете *Simulink*.

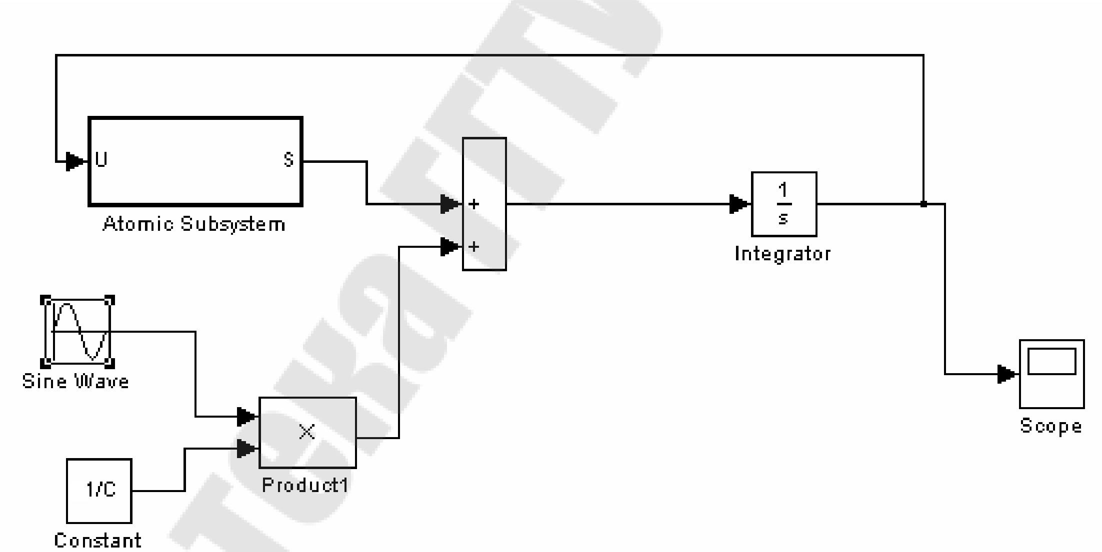


Рис. 2.2. Модель в пакете *Simulink*

Пакеты, принадлежащие ко второй группе, предназначены для моделирования сложных физических систем.

Среди пакетов, принадлежащих ко второй группе, можно отметить:

– *Dymola*;

– *Omola и OmSim*;

– *Smile*;

– *Modelica*.

Modelica – свободно распространяемый объектно-ориентированный язык для моделирования сложных физических систем.

Язык имеет хорошую техническую поддержку со стороны производителя, для него существует большое количество библиотек готовых компонентов, которые можно использовать для моделирования, происходит как дополнение уже существующих библиотек, так и разработка новых.

Modelica обеспечивает возможность создания широкого диапазона моделей различных типов: механических, электрических, гидравлических, химических и др.

Язык *Modelica* основан на концепции блоков с контактами, при соединении которых необходимые уравнения генерируются автоматически, что делает его привлекательным для специалистов нематематического профиля и более простым для понимания и использования в целом.

Modelica не ограничивает количество компонентов моделируемой системы компонентами, поставляемыми разработчиками, – пользователь может создавать свои собственные компоненты, описывая их на внутреннем языке описания блоков. Для описания блоков используется понятие класса.

Непрерывная составляющая поведения элементарного блока задается системой алгебро-дифференциальных уравнений или формулами.

Дискретная составляющая задается описанием дискретных событий (события задаются логическим условием или являются периодическими), при возникновении которых могут выполняться мгновенные присваивания переменным новых значений.

Язык *Modelica* поддерживает интеграцию с такими пакетами моделирования как *Matlab* и *SimuLink*, обеспечивает поддержку таких стандартов, как *ACSL*, *M-file*, *Simnon*, также поддерживается возможность использования функций и процедур, написанных на языке *C*.

Благодаря объектно-ориентированному подходу, созданные на языке Modelica модели легко модернизировать и создавать на их основе более сложные экземпляры.

Для работы на языке *Modelica* необходим компилятор, например, *Dymola*, *Omola*, который, как правило, в отличие от самого языка *Modelica*, является коммерческим, а не свободно распространяемым.

Третья группа включает в себя пакеты, основанные на использовании схемы гибридного автомата.

Использование карты состояний при описании переключений состояний, а также непосредственное описание непрерывных поведений системы системами алгебро-дифференциальных уравнений предоставляет большие возможности в описании гибридного поведения со сложной логикой переключений.

К этой группе относятся:

– пакет *Shift*;

– пакет *Model Vision Studium*.

Model Vision Studium (*MVS*) – это интегрированная графическая оболочка для быстрого создания интерактивных визуальных моделей сложных динамических систем и проведения вычислительных экспериментов с ними.

MVS – разработка исследовательской группы факультета технической кибернетики Санкт-Петербургского технического университета. Свободно распространяется бесплатная версия пакета – *www.xjtek.com*. На рис. 2.3 приведена модель в пакете *MVS*.

К системам визуального моделирования относятся и системы компьютерные лаборатории, позволяющие моделировать работу лабораторных стендов на компьютере.

Система *LabVIEW*, разрабатываемая компанией *National Instruments*, – это среда для моделирования электронных приборов и устройств. Она содержит большой набор инструментов для сбора данных (через подключаемые к персональному компьютеру измерительные платы, платы захвата видеоизображений и управления движением, а также через стандартные интерфейсы *GPIB*, *PXI*, *VXI* и т. д.) их анализа и отображения.

Моделирование в среде *LabVIEW* осуществляется путем создания блок-схем из так называемых «виртуальных приборов» (*virtual instruments*) – компьютерных моделей, полностью воспроизводящих все свойства реальных электронных приборов.

Система обладает большим набором функций по программированию, обработке сигналов, коммуникации, управлению приборами и обмена данными по стандартным интерфейсам.

Система включает элементы объектно-ориентированного программирования и язык *MathScript*. На рис. 2.4 приведена модель в пакете *LabVIEW*.

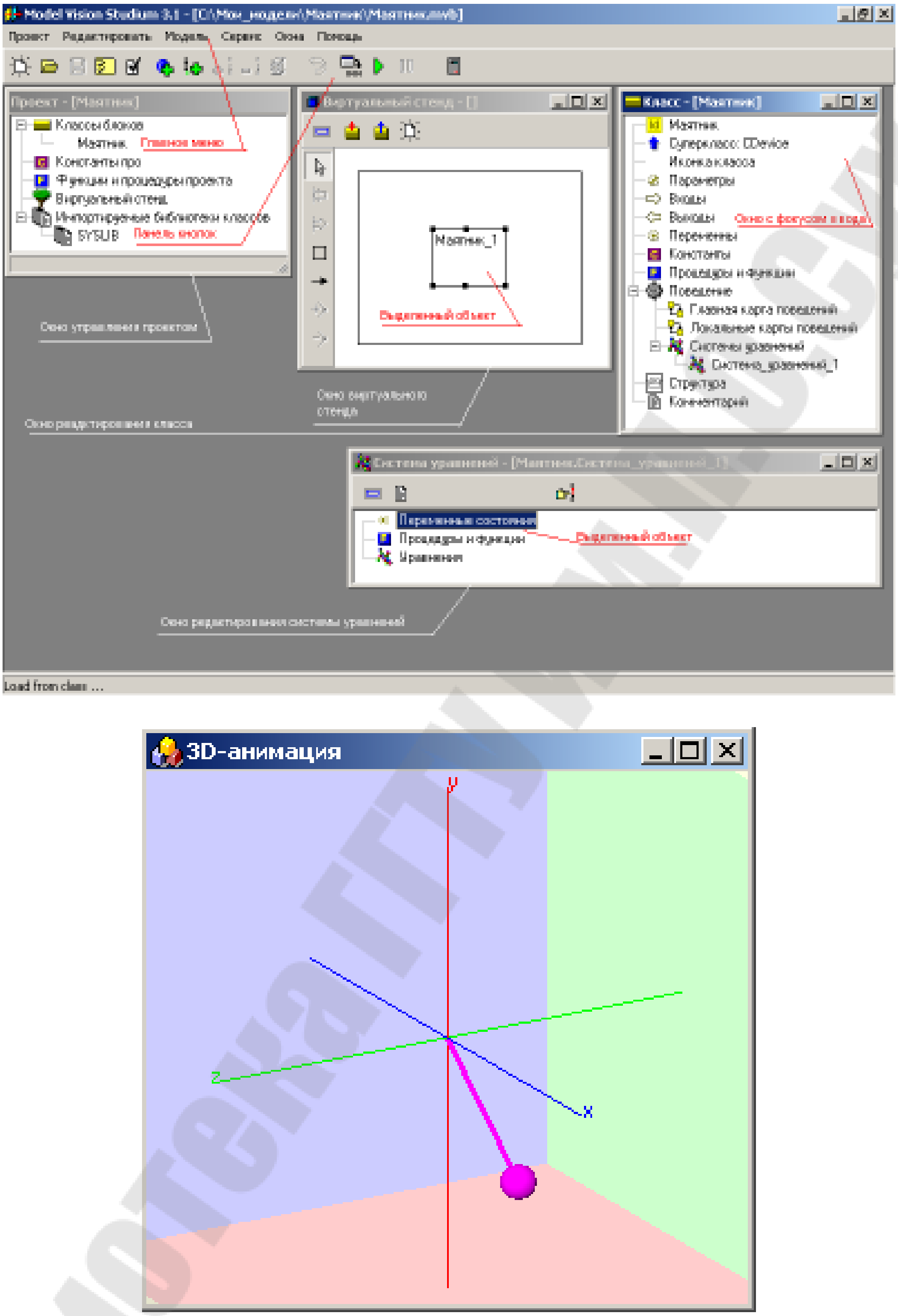


Рис. 2.3. Модель в пакете *MVS*

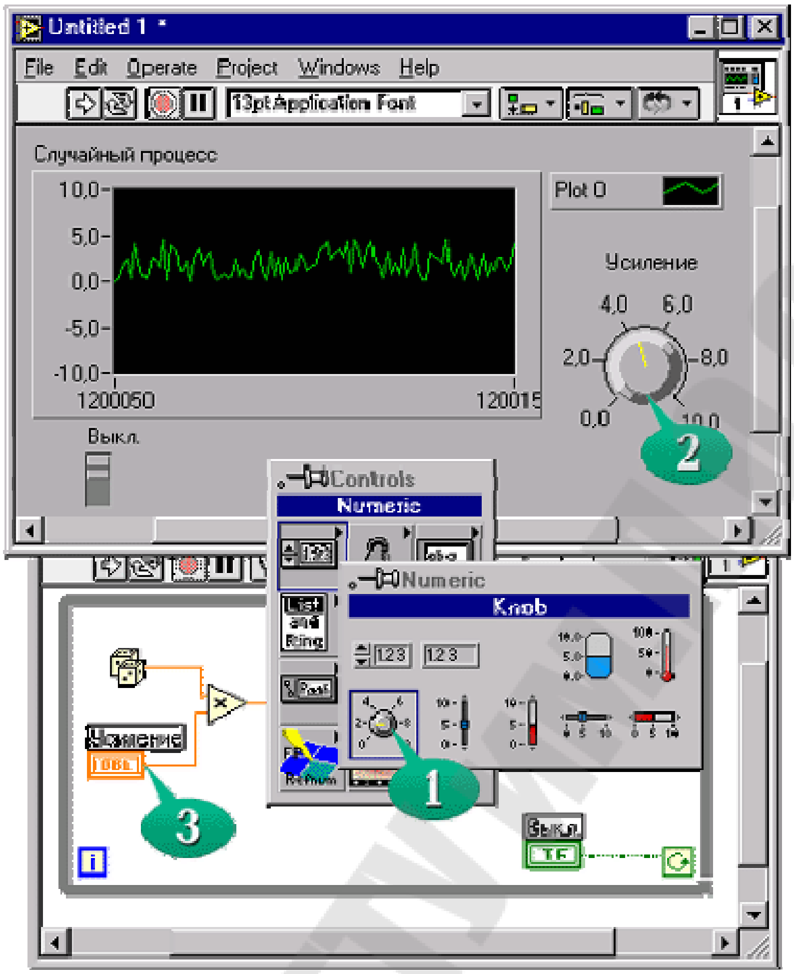


Рис. 2.4. Модель в пакете *LabVIEW*

2.2. Сравнительный анализ систем компьютерной математики

Системы компьютерной математики – компьютерные системы, предназначенные:

– для решения инженерных задач;
– для проведения научных расчетов;
– для моделирования технических объектов или процессов и др.

Система компьютерной математики *Matlab* (матричная лаборатория) изначально задумывалась разработчиком – компанией *The MathWorks* (*www.matlab.ru*) – как система, ориентированная сугубо на численные расчеты, визуализацию и многочисленные технические приложения. Аналитические вычисления поддерживаются в *Matlab*, но только за счет использования весьма ограниченной части функций ядра *Maple*, при этом сразу следует отметить, что аналитические преобразования – далеко не самая сильная сторона СКМ. Но в своем сегменте СКМ практически не имеет равных.

Matlab состоит из базового приложения – ядра системы, обеспечивающего основные функции, и набора дополнительных пакетов (*Toolbox*'ов). Ядро содержит большое количество общематематических и графических процедур, достаточных для проведения базовых инженерных расчетов. Но главное достоинство *Matlab*, в силу которого СКМ завоевывает лидирующие позиции среди численных систем, – это все-таки дополнительные пакеты, число которых насчитывает уже несколько десятков. Эти пакеты позволяют решать проблемы, пожалуй, большинства инженерных задач – от моделирования и оптимизации систем управления до создания сложнейших интерактивных программ визуализации численных экспериментов.

Язык *Matlab* – высокоуровневый язык программирования, вобравший в себя конструкции *Pascal*, *Basic*, *Fortran* и *C*, ориентированный на структурный подход, но поддерживающий и объектно-ориентированный подход. Компилятор *Matlab* позволяет получать исполняемые модули, а также объектные модули *C* и *C++*.

Система *Mathcad* популярна как в инженерной, так и в научной среде. Характерной особенностью системы является использование привычных стандартных математических обозначений, то есть документ на экране выглядит точно так же, как обычный математический расчет. Система ориентирована в первую очередь на проведение численных расчетов, но имеет встроенный символьный процессор *Maple*, что позволяет выполнять аналитические преобразования. *Mathcad* является средой визуального программирования, то есть не требует знания специфического набора команд, имеет чрезвычайно удобный математико-ориентированный интерфейс и прекрасные средства научной графики. В состав системы входят удобный текстовый редактор, позволяющий вводить, редактировать и форматировать как текст, так и математические выражения, вычислительный процессор, выполняющий расчеты по заданным формулам с использованием встроенных функций, реализующих численные методы, символьный процессор и обширная справочная система, представленная электронными учебниками. Подробную информацию о новых версиях системы можно найти на сайте производителя *Mathcad* *http://www.mathcad.com* и дистрибьютора *Mathcad* в России *http://www.mathcad.ru*.

Система *Maple* – система аналитической математики, обладающая такими возможностями, как развитые графические средства, эффективные средства решения систем дифференциальных уравнений, средства создания графических интерфейсов пользователя, мощная библиотека математических функций, большой набор сопутствующих пакетов для различных приложений, современный встроенный язык программирования интерпретирующего типа, интерфейс с рядом других *Windows*-приложений, перспективная концептуальная поддержка и др. (*www.maplesoft.com*).

Система используется для формулировки, решения и исследования различных математических моделей. Ее алгебраические средства существенно расширяют диапазон проблем, которые могут быть решены на качественном уровне. Серьезным доводом в пользу *Maple* служит возможность его интеграции с интерактивным математическим справочником (*Standard Math Interactive*) *CRC Standard Mathematical Tables and Formulae* издательства *CRC Press Inc*. и интерактивным математическим словарем (*Interactive Math Dictionary*) канадской фирмы *MathResources Inc*.

Пакет *Maple* воплощает новейшую технологию символьных вычислений, числовых вычислений с любой точностью, наличие инновационных *Web*-компонентов и весьма развитых математических алгоритмов для решения сложных математических задач. В последние годы разработчики уделяют много внимания расширяемой технологии пользовательского интерфейса (*Maplets*), что должно позволить устранить существенный недостаток (впрочем, присущий всем СКМ) – сложность организации развитого пользовательского интерфейса.

Пакет *Scilab* свободно распространяется центром *Scilab Consortium*, с *Web*-узла *www.scilab.org*, с которого можно загрузить последнюю версию программы и комплект документации. *Scilab* не требует больших системных ресурсов: инсталляционный модуль имеет размер около 20 MB, дискового пространства требуется около 41 MB. Пакет имеет много общего с СКМ *Matlab*, его можно рассматривать как облегченный вариант своего более мощного «собрата», что в некоторой степени идеально подходит для изучения в средней школе.

Scilab является командным интерпретатором и состоит из интерпретирующей системы, принимающей команды пользователя и возвращающей результаты, и двух библиотек: собственных функций и дополнительных на языках *C* и *Fortran*, имеет более тысячи функций. Как и *Matlab*, он предназначен почти исключительно для реализации численных методов, а поддерживаемые символьные операции носят скорее демонстрационную нагрузку.

Scilab позволяет также обмениваться данными с другими приложениями: поддерживаются форматы документов *Matlab* и *Maple*, структурированный текст и *TeX*. Аналогия пакета с *Matlab* и в том, что функции системы, относящиеся к некоторым прикладным областям математики и техники, собраны в дополнительные пакеты расширений (так называемые *toolboxes*). Для целей преподавания очень важно, что для создания пользовательских алгоритмов и программ *Scilab* располагает встроенным языком, обладающим широким набором конструкций для организаций циклов, условных переходов, операций ввода/вывода, с помощью которого можно получить доступ ко всем внутренним возможностям приложения. В *Scilab* реализованы концепция процедурного программирования и взаимодействие с кодом на языках *C* и *Fortran*. Как и большинство СКМ, *Scilab* имеет развитые графические средства и добротную справочную систему со множеством примеров.

СКМ *Maxima* (первое название – *Macsyma*) была создана в конце 1960-х гг. в Массачусетском технологическом институте. Инсталляционный модуль системы занимает 10 MB. Его вместе с документацией и учебником по системе можно получить на *Web*-узле системы *http://www.maxima.sourceforge.net*. Если *Scilab* система численной математики, то *Maxima* способна решать сложные аналитические задачи.

Умение выполнять сложные аналитические операции и преобразования – главное достоинство *Maxima*. Среди основных операций – операции анализа (дифференцирование, интегрирование, вычисление пределов), совершенный механизм векторно-матричных операций, представление выражений в развернутой форме, разложение функций в ряды, упрощения, преобразования, подстановки и т. п. *Maxima* способна решать уравнения и системы различных типов – алгебраические, трансцендентные и дифференциальные. Графические возможности *Maxima*, пожалуй, скромнее, чем у других СКМ, но в целом позволяют получить качественные графики для практических приложений.

Язык программирования поддерживает сложные конструкции, допуская операторы ветвления, циклов, ввода/вывода и т. д., имеет гибкие средства для работы с массивами. *Maxima* написана на языке *Lisp* и непосредственно поддерживает многие его команды. По существу, *Lisp* является ядром системы, и к нему допускается обращаться при «низкоуровневом» программировании. Таким образом, *Maxima* может быть использована как в учебных целях, так и в качестве СКМ для инженерных разработок.

MuPAD – совместный проект фирмы *SciFace Software* и исследовательской группы университета города Падерборн (Германия). Стоимость *MuPAD* позволяет рассматривать его как возможность легального использования СКМ в учебных и исследовательских целях. (Студенческая лицензия на версию 3.0 стоит 65 *EUR*, а индивидуальная или преподавательская – 90 *EUR*). При этом для большинства решаемых задач *MuPAD* практически не уступает наиболее известным разработкам данного класса.

СКМ имеет сильное аналитическое ядро, которое поддерживает все стандартные операции анализа и выдает результат в общепринятой математической нотации, а для приближенного решения сложных задач в системе имеется специальный пакет *numeric*, содержащий численные алгоритмы для дифференциальных уравнений, линейной алгебры, нахождения корней функциональных уравнений и систем, а также численного интегрирования. В *MuPAD* имеются пакет *linalg*, выполняющий линейно-алгебраические операции, и пакеты, поддерживающие функции комбинаторики, линейного программирования, интегральных преобразований, статистики, теории чисел, теории графов и др.

Последняя версия *MuPAD* позволяет строить практически все основные виды двух-, трехмерных и специальных графиков с тонкой настройкой их свойств, создавать анимационные эффекты, включать в документы внешние графические файлы. Построенные графики сохраняются в распространенных растровых и векторных форматах, а также в формате цифрового видео *AVI*.

Система обладает развитым языком для написания программ, вобравшим в себя наиболее общие конструкции *C/C++*, *Pascal* и *Fortran*. Язык не только располагает стандартными операторами циклов, условных переходов, ввода/вывода, но и обеспечивает доступ к обширным библиотекам алгоритмов, входящим в ядро и пакеты расширений системы.

MuPAD позволяет создавать программы от простейших, состоящих в определении собственных процедур и функций, до сложных модульных программ. Но для написания хороших программ в *MuPAD* придется довольно основательно овладеть языком *C*.

С пакетом можно ознакомиться на *Web*-узле *www.mupad.de*, с него же можно загрузить и использовать в течение 30 дней последнюю его версию.

3. СТАТИЧЕСКИЕ МОДЕЛИ

3.1. Обзор методов построения математических моделей

При построении и исследовании математических и компьютерных моделей можно выделить несколько подходов и методов. Устойчивой их классификации нет, так как выбор метода зависит и от типа модели, и инструментария ее реализации.

Математическая модель может быть получена:

– аналитически (закономерности протекающих в объекте процессов полностью известны);

– по результатам экспериментального исследования входных и выходных переменных объекта без изучения его физической сущности.

Последний подход особенно широко используется на практике, так как позволяет обойтись минимумом априорных сведений об объекте при построении его модели.

Наиболее достоверную математическую модель объекта можно найти аналитическим путем. Для этого необходимо располагать всесторонними сведениями об объекте (о конструкции, о законах, описывающих протекающие в нем процессы, об условиях функционирования и взаимодействия со средой). Однако часто из-за отсутствия достаточных данных получить решение задачи таким путем не удается. Трудности применения аналитических методов возникают и при описании реальных объектов, процессы в которых имеют сложный характер. Поэтому в подобных случаях эти методы дополняются экспериментальными исследованиями.

Преимуществом моделей, полученных теоретическим путем, как правило, является их достаточно общий вид, позволяющий рассматривать поведение объектов в различных возможных режимах. С практической точки зрения, более привлекательны экспериментальные методы, позволяющие находить модели объектов по результатам измерения их входных и выходных переменных.

Хотя эти методы также предполагают наличие априорных сведений об изучаемом объекте, но их характер может быть не столь обстоятельным.

Как правило, уровень априорных сведений должен быть достаточным лишь для выбора структуры модели и условий проведения эксперимента.

С другой стороны, построение математических моделей базируется на физическом или формальном подходах.

Физический подход основан на непосредственном применении физических законов (закон Гука, закон Фурье, закон Кирхгофа и т. д.).

Формальный подход использует общие математические принципы при описании физических свойств объектов.

Классификация методов построения моделей при физическом подходе следующая:

– узловой метод;

– контурный метод;

– метод переменных состояния;

– табличный метод.

Контурный и узловой методы основаны на широко применяемых в технике методах:

– метод контурных токов и узловых потенциалов в электрических системах;

– метод перемещения и метод сил в строительной механике и т. д.

Формальный подход предполагает применение различных численных методов при построении моделей. Непосредственная разработка численных методов относится к одному из разделов высшей математики – вычислительной математике.

Можно привести следующую классификацию численных методов, использующихся в математическом моделировании:

– методы решения уравнений;

– методы решения систем уравнений;

– методы вычисления интегралов;

– методы аппроксимации и интерполяции;

– методы решения дифференциальных уравнений и систем;

– методы оптимизации и т. д.

Для численного решения алгебраических и трансцендентных уравнений пригодны одни и те же методы уточнения приближенных значений действительных корней:

– метод половинного деления (метод дихотомии);

– метод простых итераций;

– метод Ньютона (метод касательных);

– модифицированный метод Ньютона (метод секущих);

– метод хорд и др.

Численное решение систем алгебраических уравнений реализуется следующими численными методами.

Для систем линейных уравнений применяются:
– метод определителей Крамера;
– матричный метод;
– метод Гаусса.
Для систем нелинейных уравнений:
– метод Ньютона;
– метод простых итераций и др.
К методам вычисления определенного интеграла относятся:
– метод прямоугольников;
– метод трапеций;
– метод Симпсона.

К численным методам аппроксимации и интерполяции относятся следующие.

Интерполяция:
– метод линейной интерполяции;
– метод сплайновой интерполяции;
– метод тригонометрической интерполяции.

Основным численным методом аппроксимации является метод наименьших квадратов.

Численные методы решения дифференциальных уравнений используются при моделировании динамических технических объектов.

Этот класс численных методов представлен следующими:
– метод последовательных приближений;
– конечно-разностные методы;
– методы Рунге–Кутта;
– методы прогноза и коррекции и т. д.

К численным методам оптимизации относятся следующие:
– метод «золотого сечения»;
– методы градиентного спуска;
– метод сопряженных градиентов;
– метод Ньютона и др.

3.2. Статические модели

Статической называется такая модель технического объекта, в которой не учитываются изменения параметров объекта или внешних воздействий во времени.

Статические модели применяются при проектировании конструкции объекта, при расчете его конструктивных параметров.

Статическая модель описывается алгебраическими уравнениями или системами линейных или нелинейных уравнений, в зависимости от свойств самого объекта.

Такая модель осуществляет преобразование числовых значений входных переменных X в числовое значение выходной переменной Y:

$$Y = F(X).$$

Математическая модель системы также называется статической, если значение выхода $Y(t)$ зависит от значения входа $X(t)$ только в тот же момент времени t. Математическая запись имеет вид:

$$Y(t) = F(X(t)),$$

где F – некоторый оператор.

Кроме того, статические модели могут задаваться неявно в виде, например, $\Phi(Y(t),\ X(t))$.

Так обычно записываются уравнения статических режимов многих механических и электрических систем.

Статические модели строятся, когда технический объект соответствует ряду условий:

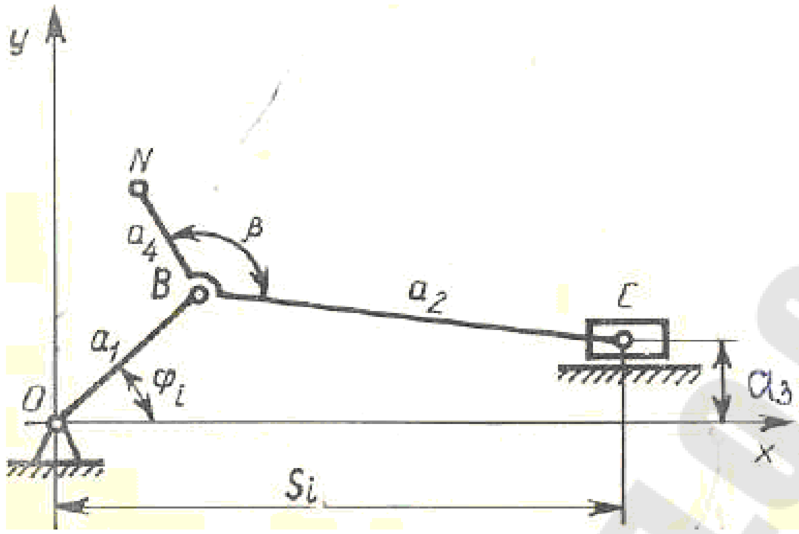
– система устойчива, т. е. переходные процессы после скачкообразного изменения входов затухают;

– входы системы меняются медленно;

– выходы системы изменяются редко.

Пример 1. Дан кривошипно-ползунный механизм, исходными данными для проектирования которого служит функциональная зависимость перемещения ползуна S от угла поворота кривошипа φ. Необходимо определить длины звеньев $a1$, $a2$ и значение параметра $a3$.

Таблица значений φ_i и S_i может содержать по три значения, т. е. задаются три положения механизма (i = 1, 2, 3). На рис. 3.1 приведен вид механизма и его математическая модель.



При i=3 механизм описывается системой уравнений вида:

$$K_1 S_1 \cos\varphi_1 + K_2 \sin\varphi_1 - K_3 = S_1^2$$

$$K_1 S_2 \cos\varphi_2 + K_2 \sin\varphi_2 - K_3 = S_2^2$$

$$K_1 S_3 \cos\varphi_3 + K_2 \sin\varphi_3 - K_3 = S_3^2$$

Длины звеньев вычисляются по формулам:

$$a_2 = \sqrt{a_1^2 + a_3^2 - K_3} \qquad a_1 = \frac{K_1}{2} \qquad a_3 = \frac{K_2}{2a_1}$$

Рис. 3.1. Вид механизма и его математическая модель

Общая характеристика модели:
– статическая (не зависит от времени);
– линейная, так как описывается системой линейных уравнений.
Входные параметры модели – вектора *S* и φ.
Выходные параметры – длины звеньев *a*1, *a*2 и расстояние *a*3.
Внутренние параметры – значения вектора *К*.
Ниже приведена реализация модели в *Matlab*.

М-файл	Результаты расчета
S=[0.7; 1.3; 1.5]; deg=pi/180;	K =
fi=[30*deg; 20*deg;15*deg];	2.1466
for i=1:3	-1.2785
A(i,1)=S(i)*cos(fi(i));	0.3638
A(i,2)=S(i)*sin(fi(i));	a1 =
A(i,3)=-1;	1.0733
B(i)=S(i).^2; B1=B.';	a3 =
end	-0.5956

K=inv(A)*B1 a1=K(1)/2 a3=K(2)/(2*a1) a2=sqrt(a1^2+a3^2-K(3))	a2 = 1.0690

3.3. Построение моделей по результатам эксперимента

Любому специалисту в своей практической деятельности приходится изучать зависимости между различными параметрами исследуемых объектов, процессов и систем. Например: зависимость числа оборотов двигателя от нагрузки, т. е. $n = f(\text{Мкр.})$; зависимость силы резания при обработке детали на металлорежущем станке от глубины резания, т. е. $P = f(t)$, и т. д.

Из всех способов задания зависимостей наиболее удобным является аналитический способ задания зависимости в виде функции $n = f(\text{Мкр.})$, $P = f(t)$, $y = f(t)$.

Однако на практике специалист чаще всего получает зависимости между исследуемыми параметрами экспериментально. В этом случае ставится натурный эксперимент, изменяются значения параметров на входе системы, измеряются значения параметров на выходе системы. Результаты измерений заносятся в таблицу.

Таким образом, в результате проведения натурного эксперимента получаем зависимости между исследуемыми параметрами в виде таблицы, т. е. получаем так называемую табличную функцию.

Далее с этой табличной функцией необходимо вести научно-исследовательские расчеты. Например, необходимо проинтегрировать или продифференцировать табличную функцию и т. д.

При такой постановке задачи моделирования нужно заменить табличную функцию аналитической. Для этой цели используются методы аппроксимации и интерполяции.

Аппроксимация – это замена исходной функции $f(x)$ функцией $\varphi(x)$ так, чтобы отклонение $f(x)$ от $\varphi(x)$ в заданной области было наименьшим. Функция $\varphi(x)$ называется аппроксимирующей.

Если исходная функция $f(x)$ задана таблично (дискретным набором точек), то аппроксимация называется дискретной. Если исходная функция $f(x)$ задана аналитически (на отрезке), то аппроксимация называется непрерывной или интегральной.

Интерполяция – это замена исходной функции $f(x)$ функцией $\varphi(x)$ так, чтобы $\varphi(x)$ точно проходила через точки исходной функции $f(x)$.

Интерполяция еще называется точечной аппроксимацией.

Точки исходной функции $f(x)$ называются узлами интерполяции.

Для интерполирующей функции справедливо

$$\varphi(x_i) = f_i, \; i = 0, 1, 2, ..., n.$$

Экстраполяцией называется аппроксимация вне заданной области определения исходной функции, т. е.

$$x < x_0 \quad \text{и} \quad x > x_n.$$

Найдя интерполяционную функцию, мы можем вычислить ее значения между узлами интерполяции, а также определить значение функции за пределами заданного интервала (провести экстраполяцию).

Основной мерой отклонения функции $y(x)$ от функции $f(x)$ при аппроксимации является величина, равная сумме квадратов разностей между значениями аппроксимирующей и исходной функций:

$$\sum_{i=1}^{n} (y_i - f(x_i))^2 \to \min.$$

Простейшими видами интерполяции является линейная и квадратичная.

При линейной интерполяции точки заданной функции соединяются линейными отрезками и функция $f(x)$ приближается ломаной с вершинами в данных точках. В качестве уравнения интерполяционного многочлена используются уравнения прямой, проходящей через две точки.

При квадратичной интерполяции в качестве приближающей функции, соединяющей соседние точки, принимается квадратный трехчлен. Такая интерполяция называется параболической.

Распространенным видом интерполяции является интерполяция с использованием кубических сплайн-функций.

Сплайн представляет собой модель гибкого тонкого стержня из упругого материала, закрепленного в двух соседних узлах интерполяции с заданными углами наклона α и β так, чтобы потенциальная энергия стержня была минимальна (рис. 3.2).

Рис. 3.2. Вид сплайн-функции

Интерполяция может выполняться с помощью многочленов Ньютона, Эрмита, Лагранжа и т. д.

Наиболее известными методами аппроксимации являются метод наименьших квадратов, метод многочленов Чебышева, рядов Тейлора и т. д.

При решении задач аппроксимации часто используются функции регрессии.

Регрессия – представление совокупности данных некоторой функцией $f(x)$.

Задачей регрессии является вычисление параметров функции $f(x)$ таким образом, чтобы функция приближала последовательность исходных точек с наименьшей погрешностью. При этом функция $f(x)$ называется *уравнением регрессии*.

При регрессии не требуется, чтобы функция проходила через все заданные точки, что особенно важно при аппроксимации данных, заведомо содержащих ошибки.

Применение методов аппроксимации при моделировании технических объектов имеет несколько направлений.

Основными из них являются следующие:

– обработка результатов эксперимента по компьютерной модели;

– обработка результатов натурного эксперимента, синтез модели;

– создание робастных моделей элементов при моделировании систем и др.

Применение аппроксимации при обработке результатов компьютерного моделирования заключается в следующем.

Модель, построенная в численном виде, в качестве выходных параметров имеет дискретные функции, для получения новых значений которых нужно проводить новый эксперимент по модели.

Для того чтобы при каждом новом значении варьируемого параметра не обрабатывать модель, нужно заменить ее аналитической функцией, полученной путем аппроксимации результатов исследований по модели.

Если компьютерная модель получена в аналитическом виде, например, разработана в системе *Maple*, то в аппроксимации нет необходимости, так как исследование модели сводится к простому табулированию полученной функции.

Второе направление позволяет применить методы аппроксимации к дискретным данным, полученным, как правило, в виде файлов после проведения натурных экспериментов с применением специальных датчиков, регистраторов и т. д.

Третье направление применения аппроксимации – замена точной модели элемента в системе робастной моделью при синтезе моделей сложных технических систем.

Термин «робастность» (англ. *robustness*) образован от *robust* – крепкий, грубый. Сравните с названием одного из сортов кофе – *robusta*.

Имеется в виду, что робастные модели должны «выдерживать» ошибки, которые теми или иными способами могут попадать в исходные данные.

Суть этого направления заключается в следующем: при создании и предварительной обработке модели сложной технической системы (электронной схемы, механической системы) невозможно разработать точные модели всех ее элементов сразу. Поэтому один или несколько элементов, для которых известны входные и выходные параметры, задается аппроксимирующей эти параметры функцией. Эта функция используется в общей модели технической системы при получении предварительных результатов моделирования.

Для получения уточненной модели системы все робастные модели элементов заменяются на точные, и расчет модели системы производится заново.

Пример обработки результатов компьютерного эксперимента приведен ниже.

Пример 2. После реализации модели и проведения экспериментов с моделью получены вектора $U1$, $U2$, ..., $U9$, *в которых содержались значения напряжений в разные моменты времени.*

Вектор варьируемого параметра содержит значения емкостей, при которых проводились опыты по модели (рис. 3.3).

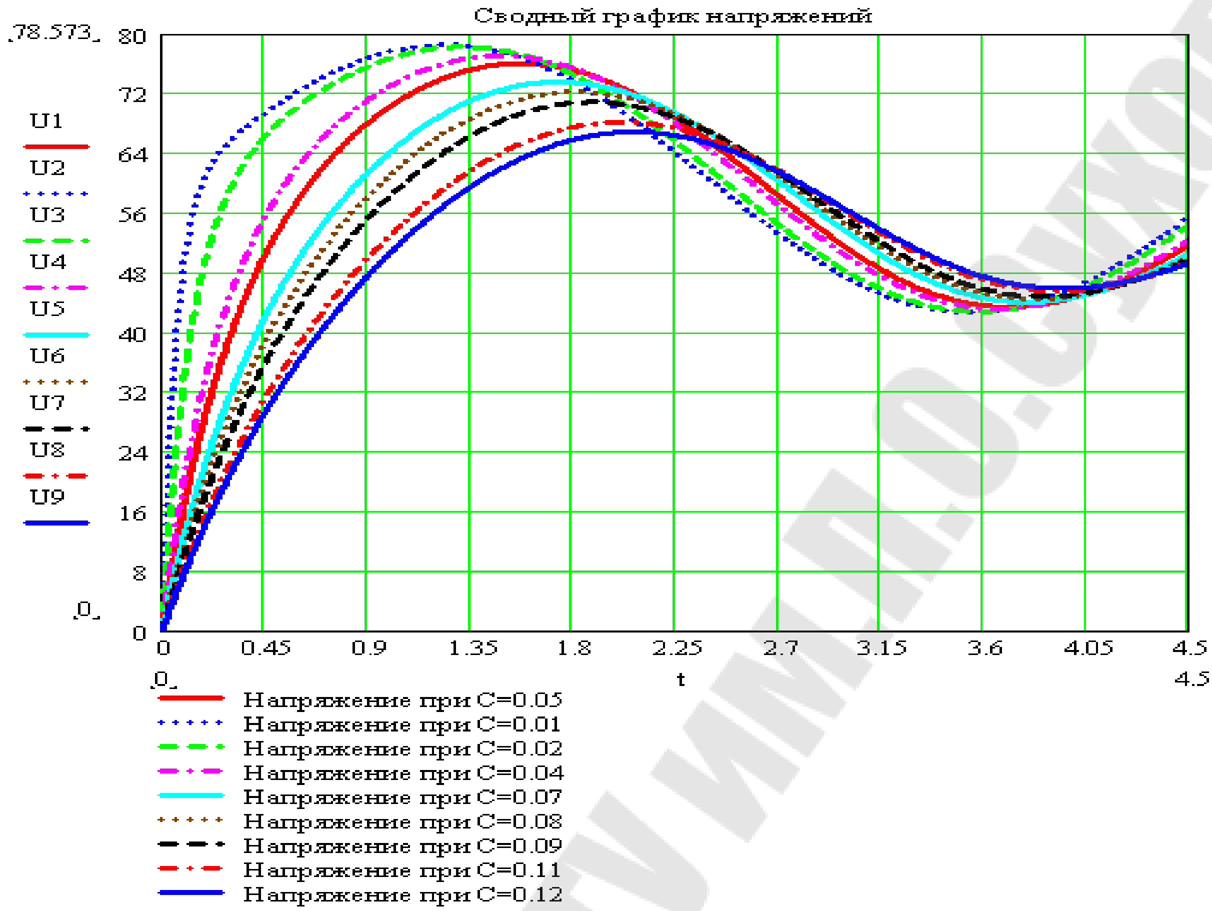


Рис. 3.3. Сводный график результатов компьютерного эксперимента

Алгоритм расчета приведен ниже.

• Сформировать вектор, содержащий максимальные значения элементов исходных векторов – $A_{\max}$.

• Сформировать вектор значений емкостей, при которых проводились эксперименты по модели – *CR*.

• Для аппроксимации применить функцию полиномиальной регрессии *polyfit*. В качестве аппроксимирующей функции – полином третьей степени.

• Получить коэффициенты полинома *p* и вектора аппроксимирующей функции *f*.

Ниже приведена реализация модели в *Matlab*.

М-файл	Результаты расчета

CR=[0.01 0.02 0.04 0.05 0.07 0.08 0.09 0.11 0.12];
Amax=[78.573 78.295 77.035 76.03 73.572 72.235 70.877 68.18 66.867];
%Полиномиальная регрессия
p=polyfit(CR,Amax,3);
xx=0.009:0.001:0.13;
f=polyval(p,xx)
figure(1)
plot(CR, Amax, ‘o’, xx,f)
	p=
	1.0e+003*
	6.9379 -1.8558 0.0262 0.0785

Результаты аппроксимации приведены на рис. 3.4.

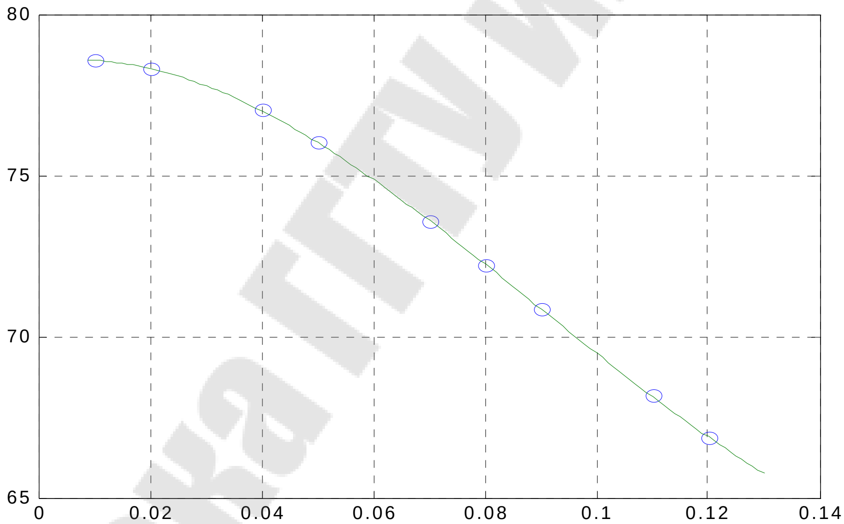


Рис. 3.4. Результаты аппроксимации данных компьютерного эксперимента

4. ДИНАМИЧЕСКИЕ МОДЕЛИ

4.1. Определение динамической модели, понятия времени, пространства, движения

Термин «динамическая система» первоначально отождествлялся с системой ОДУ, правая часть которой удовлетворяет условиям, гарантирующим единственность решения.

Сейчас этот термин применяется ко всем системам, чье поведение зависит от времени.

С моделированием динамических систем тесно связано понятие времени. В представлении Ньютона, время является самостоятельной сущностью реального мира, что позволяет говорить о его математическом моделировании.

Ньютон различал:

– *физическое* (*астрономическое*) время: «Относительное, кажущееся или обыденное время есть постигаемая чувствами, внешняя мера продолжительности, употребляемая в обыденной жизни вместо истинного математического времени, как то час, день, месяц, год»;

– *абсолютное* (*математическое*) *время*: «Абсолютное математическое время само по себе без всякого отношения к чему-либо внешнему протекает равномерно и иначе называется длительностью».

Ньютоновское время характеризуется непрерывным направленным течением с постоянной скоростью.

Математической моделью ньютоновского времени является вещественная ось, по которой с постоянной скоростью пробегает переменная t (время), двигаясь из прошлого в будущее в пределах от $-\infty$ до $+\infty$.

С непрерывным временем связаны обыкновенные дифференциальные уравнения, применяющиеся для описания непрерывных во времени процессов. В этих уравнениях время – это единственная независимая переменная, от которой зависит положение точки $x(t)$.

При рассмотрении многих реальных физических процессов часто отказываются от свойства непрерывности времени и вводят дискретное время.

Дискретное время – это любая упорядоченная, неограниченно возрастающая последовательность вещественных или рациональных чисел.

Дискретное время естественным образом появляется там, где наблюдается разделение поведения технического объекта на «быстрые» (дискретные или мгновенные) и «медленные» (непрерывные или длительные) действия.

Примером может служить «быстрый» цифровой регулятор, управляющий движением «медленного» непрерывного объекта.

В ньютоновских законах движения наряду с независимой переменной «время» присутствуют пространственные координаты движущейся точки (x, y, z). Четверка (t, x, y, z) определяет положение тела в пространстве и времени.

Можно следить за положением точки ($x(t)$, $y(t)$, $z(t)$) как функцией от времени. Графики функций $x(t)$, у(t), $z(t)$ называются *временными диаграммами*.

Движение точки в пространстве может быть свободным и вынужденным. Свободное движение зависит только от начального состояния. Вынужденное движение зависит от возмущающих и управляющих воздействий.

4.2. Виды и параметры динамических моделей

Модель динамической системы можно рассматривать как пару «оператор системы и модель внешних воздействий».

Оператором системы называется закон, в соответствии с которым система преобразует внешнее (входное) воздействие в выходной сигнал.

По виду оператора динамические системы и их модели делятся на:

– линейные и нелинейные;
– непрерывные и дискретные;
– нестационарные и стационарные;
– одномерные и многомерные;
– с сосредоточенными и распределенными параметрами.

Линейные системы описываются линейными дифференциальными уравнениями.

Нелинейные системы описываются нелинейными дифференциальными уравнениями.

Непрерывные системы описываются дифференциальными уравнениями.

Дискретные системы описываются разностными уравнениями.

Нестационарные системы описываются уравнениями с переменными коэффициентами.

Стационарные системы описываются уравнениями с постоянными коэффициентами.

Одномерные системы имеют один вход и один выход.

Многомерные системы имеют суммарное число входов и выходов, большее двух.

Системы с сосредоточенными параметрами описываются обыкновенными дифференциальными уравнениями.

Системы с распределенными параметрами описываются уравнениями в частных производных.

Внешние воздействия делятся на:

– непрерывные (функции непрерывного аргумента) и дискретные (функции дискретного аргумента);

– детерминированные и случайные;

– одномерные и многомерные.

Для того чтобы классифицировать конкретную динамическую модель, нужно указать каждый из пяти классификационных признаков оператора системы и каждый из трех признаков, к которым принадлежат внешние воздействия.

Например, динамическая система описывается дифференциальным уравнением вида:

$$a_n(t)\frac{d^n x(t)}{dt^n}+...+a_0(t)x(t)=b_m(t)\frac{d^m g(t)}{dt^m}+...+b_0(t)g(t).$$

Здесь $g(t)$ – входной сигнал; $x(t)$ – выходной сигнал; $a(t)$, $b(t)$ – внутренние параметры системы, коэффициенты левой и правой частей уравнения.

Эта система:

– линейная;

– непрерывная;

– нестационарная;

– детерминированная;

– одномерная.

Внешнее воздействие:

– непрерывное;

– детерминированное;

– одномерное.

Если коэффициенты уравнения постоянны, система называется линейной стационарной:

$$a_n\frac{d^n x(t)}{dt^n}+...+a_0x(t)=b_m\frac{d^m g(t)}{dt^m}+...+b_0g(t).$$

Динамические модели могут быть представлены в следующих формах:

– явная функция, зависящая от времени $z = z(t)$;

– нелинейное алгебраическое уравнение относительно переменных, явно зависящих от времени – $G(z(t), t) = 0$;

– дифференциальное уравнение или система дифференциальных уравнений;

– алгебро-дифференциальное уравнение;

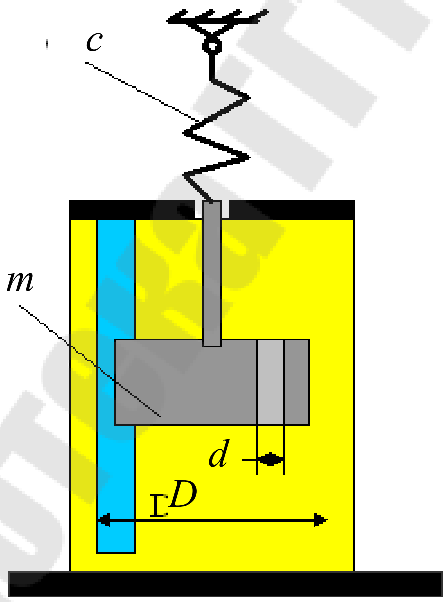
– интегро-дифференциальное уравнение или система;

– передаточная функция.

4.3. Методы реализации динамических моделей

Основными методами реализации динамических моделей являются аналитические и численные методы. В основе аналитических методов лежат математические приемы решения ОДУ и систем ОДУ с получением результирующих функций в аналитическом виде.

Например, результат получения аналитического выходного сигнала (перемещения демпфера) для модели гидравлического демпфера (рис. 4.1) в СКМ *Maple* приведен ниже.



$$\ddot{y} + 2n\dot{y} + py = 0$$

Рис. 4.1. Гидравлический демпфер

Вид дифференциального уравнения движения демпфера

> restart;

*> Sys := diff (y(t) , t$2) + 2*n * diff (y(t) , t$1)+ p^2*y(t) = 0;*

$$Sys := \left(\frac{d^2}{dt^2}\,\mathrm{y}(t)\right) + 2\,n\left(\frac{d}{dt}\,\mathrm{y}(t)\right) + p^2\,\mathrm{y}(t) = 0$$

Задание начальных условий

> Nu := y(0) = 0.05 , D(y)(0) = 0;

$$\mathrm{N} := \mathrm{y}(0) = 0.05,\ \mathrm{D}(y)(0) = 0$$

Решение дифференциального уравнения

> Rs := dsolve ({Sys , Nu} , y(t)) ;

$$Rs := \mathrm{y}(t) = \frac{1}{40}\,\frac{\left(n+\sqrt{n^2-p^2}\right)\mathrm{e}^{\left(\left(-n+\sqrt{n^2-p^2}\right)t\right)}}{\sqrt{n^2-p^2}} + \frac{1}{40}\,\frac{\left(-n+\sqrt{n^2-p^2}\right)\mathrm{e}^{\left(\left(-n-\sqrt{n^2-p^2}\right)t\right)}}{\sqrt{n^2-p^2}}$$

Если невозможно получить аналитическую функцию выходного параметра модели, то применяются численные методы решения ОДУ и систем ОДУ.

Пример применения метода Рунге-Кутта к моделированию гидравлического демпфера и реализация его в *Matlab* приведены ниже.

```
% Задание начальных условий
y0=[0.05 0];
% Решение дифференциального уравнения
[T,Y]=ode45(@vid2,[0 1],y0);
% Графики функций перемещения и скорости демпфера
figure(1)
plot(T,Y(:,1))
figure (2)
plot(T,Y(:,2))

function ur2=vid2(t,y);
ur2=zeros(2,1);
% Исходные данные
m=2.73; H=0.05; c=3e3; D1=0.1; d=0.01; z=25; mu=6e-2;
% Расчет приведенного коэффициента вязкого сопротивления
% и частоты собственных колебаний демпфера
p=sqrt(c/m); n=4*pi*mu*H/(m*z)*(D1/d)^4;
% Описание системы дифференциальных уранений
ur2(1)=y(2);
ur2(2)=-2*n*y(2)-p^2*y(1);
```

4.4. Анализ моделей динамических систем

Для линейных динамических систем ставятся следующие задачи анализа:

– анализ выходных процессов;
– анализ устойчивости;
– анализ чувствительности;
– анализ управляемости.

Для разработанной динамической модели можно проводить следующие виды анализа выходного сигнала:

– влияние изменения начальных условий на выходной сигнал;
– влияние изменения внутренних параметров на выходной сигнал;
– влияние изменения параметров возмущающих воздействий на выходной сигнал.

Для многих сложных и малоизученных процессов невозможно получить математическую модель только исходя из априорных знаний.

Недостаток знаний в этом случае компенсируется наблюдениями за функционированием процесса.

Эта информация может быть использована:

– для определения неизвестных параметров системы при известной структуре модели (модель в виде «серого» ящика);
– для построения модели без знания ее структуры («черный ящик»).

В случае «серого ящика» известна структура объекта, например, дифференциальное уравнение, но неизвестны конкретные значения параметров.

В этом случае задачу определения параметров можно свести к задаче параметрической оптимизации – минимизации ошибки моделирования.

При проектировании технической системы в качестве критерия ее оптимальности принимаются показатели эффективности и качества процессов функционирования.

Функционирование технических систем происходит в условиях внешней среды.

Любое изменение внешних управляющих или возмущающих воздействий приводит к возникновению переходного процесса.

В переходном процессе могут возникать большие амплитуды отклонений внутренних параметров, сопровождающиеся значительным повышением деформаций и напряжений в конструктивных элементах технических систем.

Периодические внешние воздействия возбуждают вибрации, оказывающие вредные воздействия на организм человека и сокращающие срок службы механизмов технической системы.

При изменении внешнего воздействия $u(t)$ любой выходной сигнал технической системы $v(t)$ может быть представлен состоящим из двух составляющих:

$$v(t) = v_{\text{в}}(t) + v_{\text{п}}(t),$$

где $v_{\text{в}}(t)$ – вынужденная установившаяся составляющая, определяемая частным решением неоднородного дифференциального уравнения, описывающего функционирование динамической технической системы; $v_{\text{п}}(t)$ – переходная составляющая, характеризующая свободный переходный процесс и определяемая общим решением однородного дифференциального уравнения без правой части.

Если техническая система устойчива, переходная составляющая с течением времени затухает и остается лишь вынужденная составляющая.

О качестве переходных процессов есть смысл говорить лишь для устойчивых систем.

При анализе переходных процессов применяются следующие виды воздействий:

– ступенчатое;

– импульсное;

– кусочно-линейное;

– гармоническое.

Переходной характеристикой называется реакция технической системы на ступенчатое воздействие.

Переходную характеристику рекомендуют получать при нулевых начальных условиях и единичном ступенчатом воздействии.

$$u(t) = \begin{cases} 0, & t < 0; \\ 1, & t \geq 0. \end{cases}$$

Для оценки качества переходного процесса используются следующие показатели [4]:

1. Время переходного процесса – $t_{\text{п}}$, характеризуется длительностью пребывания технической системы в неустановившемся состоянии.

2. Коэффициент динамичности – $k_{\text{д}}$, характеризуется максимальным отклонением выходного сигнала от его значения в установившемся конечном состоянии.

3. Колебательность K, определяет число колебаний за время $t_п$.

4. Декремент колебаний показывает отношение амплитуды двух смежных отклонений фазовой координаты от установившегося значения процесса.

Длительность переходного режима даже при быстром затухании динамического процесса теоретически бесконечна, поэтому на практике считают переходный процесс завершенным, если значение выходного сигнала перестает отличаться от установившегося конечного значения не более чем на определенную величину (рис. 4.2) [4].

Эта величина называется коридором стабилизации установившегося состояния Δ. Обычно $\Delta = 5\,\%|v(t) - v(\infty)|$.

Время переходного процесса равно интервалу времени, измеряемому от момента начала ступени сигнала воздействия до момента последнего пересечения переходной характеристикой линии коридора стабилизации.

Время переходного процесса характеризует быстродействие технической системы.

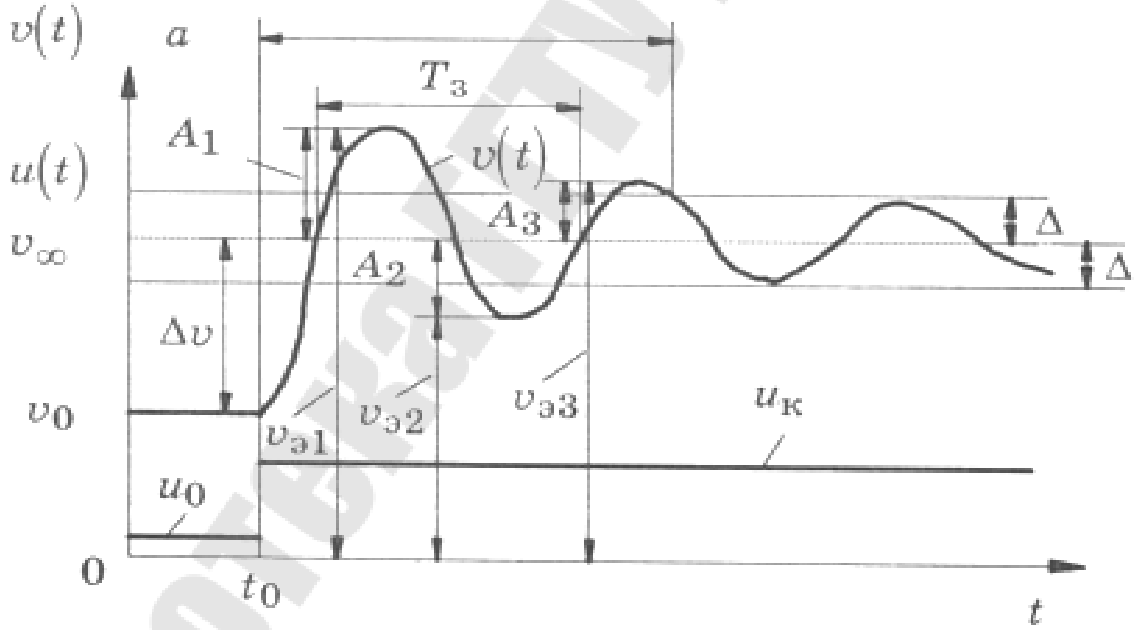


Рис. 4.2. График переходного процесса

Коэффициент динамичности определяется по формуле

$$k_d = 1 + A_{\max} / \Delta v,$$

где $A_{\max}$ – максимальная амплитуда отклонения выходного сигнала V от конечного установившегося ее значения:

$$\Delta v = \left| v(t) - v(\infty) \right|,$$

$$A_{\max} = \max \left| V_{эi} - V(\infty) \right|,$$

$V_{эi}$ – i-е экстремальное значение выходного сигнала, определяемое из условия $d_v/d_t = 0$.

При высоких требованиях к характеристикам функционирования технических систем значение $k_д$ не должно превышать 1,25–1,45.

Декремент колебаний определяется по формуле

$$D = A1 / A2,$$

где $A1$, $A2$ – амплитуды двух смежных отклонений фазовой координаты от значения $V\infty$.

Чем выше значение D, тем быстрее затухают колебания.

Колебательность K определяется числом амплитудных значений $V_{эi}$ за время $t_п$ или числом полупериодов колебаний.

При высоких требованиях к переходным процессам K не должно превышать 2, при умеренных – 5.

Литература

1. Охорзин, В. А. Компьютерное моделирование в системе *Mathcad*. Учебный курс / В. А. Охорзин. – Москва : Финансы и статистика, 2006.

2. Лазарев, Ю. Моделирование процессов и систем в *Matlab* / Ю. Лазарев. – Санкт-Петербург : БХВ–Петербург, 2005.

3. Зарубин, В. С. Математическое моделирование в технике / В. С. Зарубин. – Москва : Изд-во МГТУ им. Н. Э. Баумана, 2003.

4. Тарасик, В. П. Математическое моделирование технических систем / В. П. Тарасик. – Минск : ДизайнПро, 1997.

Содержание

Учебное электронное издание комбинированного распространения

Учебное издание

Трохова Татьяна Анатольевна

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ ТЕХНИЧЕСКИХ СИСТЕМ

Пособие по одноименному курсу для студентов специальности 1-40 01 02 «Информационные системы и технологии (по направлениям)» дневной формы обучения

Электронный аналог печатного издания

Редактор *А. В. Власов*
Компьютерная верстка *Н. Б. Козловская*

Подписано в печать 23.02.11.
Формат 60x84/$_{16}$. Бумага офсетная. Гарнитура «Таймс».
Ризография. Усл. печ. л. 2,79. Уч.-изд. л. 2,5.
Изд. № 63.
E-mail: ic@gstu.by
http://www.gstu.by

Издатель и полиграфическое исполнение:
Издательский центр учреждения образования
«Гомельский государственный технический университет
имени П. О. Сухого».
ЛИ № 02330/0549424 от 08.04.2009 г.
246746, г. Гомель, пр. Октября, 48.