

Моделирование информационных систем

Санкт-Петербург

СПб ГУТ)))

Санкт-Петербургский государственный университет
телекоммуникаций им. проф. М.А. Бонч-Бруевича



2013

Анализ компьютерных средств

На этапе проектирования СУ возможные решения обычно оцениваются на основании интуиции и предыдущего опыта. Однако в настоящее время, как правило, требуется инструментарий, позволяющий автоматизировать процесс принятия решений.

Для моделирования СУ используются различные программные продукты: специализированные пакеты; библиотеки программ; математические системы программирования. Хорошо разработаны методики и алгоритмы, позволяющие исследовать режимы работы сложных СУ; анализировать их качество, рассчитывать частотные характеристики и импульсные переходные функции; исследовать динамику сложных систем, содержащих элементы с нелинейными характеристиками; рассчитывать оптимальные процессы при наличии ограничений; исследовать динамику стохастических систем, и т. д.

При выборе того или иного программного продукта необходимо

учитывать следующее:

- вид математического описания исследуемой

электромеханической системы;

- особенности представления данных модели;

- каким порядком дифференциальных уравнений, или каким

порядком и видом матрицы (симметричная, избыточная, вырожденная и т.

д.), или каким количеством структурных элементов графа описывается

электромеханическая система;

- вид представления результатов расчета;

- количество и вид нелинейных характеристик, описывающих

управляющие и возмущающие воздействия;

- возможность гибкого изменения математической модели;

- технологию функционирования математического ядра

моделирующей программы.

В настоящее время в основе существующих функционально ориентированных методов, направленных на совершенствование методологии программирования, лежит процесс целенаправленного расчленения реализуемых моделей на составные, более простые части – декомпозиция модели. Декомпозиция может быть как функциональной, так и по структуре данных.

К категории функционально ориентированных методов программирования относятся следующие: усовершенствованная методика программирования (IPT) фирмы «IBM»; метод структурного анализа и проектирования (SADT) фирмы «Софтек» (США); метод анализа и структурного программирования (MAPS) фирмы «Ефрософт» (США); методология метадетализации (MSR); составное проектирование Майерса и структурное проектирование Констайна и Йордана и отечественные: Р-технологии И. В. Вельбицкого и Прометей-технологии В. В. Липаева.

Усовершенствованная методика программирования (ИРТ) является наиболее перспективным подходом к усовершенствованию методологии и технологии программирования с точки зрения практического внедрения. В основе этой методики лежат теоретические и методологические концепции структурного программирования, проектирования методом «сверху – вниз», метода HIPO и библиотеки обеспечения разработки.

Для моделирования СУ применяют следующие программные средства: MATLAB («MathWorks, Inc»), GPSS (компания «Minuteman Software» (США)), SCILAB, MATHCAD («Mathsoft, Inc»), DERIVE («Soft Warehouse»), SystemBuild (MATRIXx), Anylogic (Model Vision Studium), VisSim, MBTY, Easy5S, Dynast, 20-sim, Dymola, Simplorer, ITI-sim, Pspice, Multisim, Micro-Cap и др., системы Modelica, CADA (Supervisory Control And Data Acquisition – диспетчерское управление и сбор данных) или DCS (Distributed Control Systems – распределённая система управления), реализующие в том числе и функции схематического отображения технологических процессов, а также Case-средства (Computer-Aided Software/System Engineering) и др.

Math Works MATLAB Simulink, VisSim, MBTY, Easy5S относятся к

программам с поточной моделью управления (модель программирования, в которой инструкции, процедуры или функции выполняются только тогда, когда все входные данные (т. е. параметры и аргументы) готовы). Эти программы легко интегрируются с системами сбора данных, благодаря чему возможно создание (компьютерных) моделей с физическими объектами в контуре. В большинстве из них возможно программирование цифровых сигнальных процессоров. Структура их моделей может меняться в процессе симуляции без затрат времени, согласно событийному управлению.

Dynast, 20-sim, Dymola, Simplorer, ITI-sim, Pspice, Multisim, Micro-

Cap относятся к программам – интерпретаторам систем уравнений. Эти программы, в скрытом от пользователя режиме, легко преобразуют текстовую запись систем уравнений к требуемому решателем виду. Эти особенности технологии позволили не ограничиваться моделями в виде передаточных функций, и, временно, эти программы заняли лидирующие позиции в области мультидоменного моделирования.

Недостатки первой группы моделирующих программ связаны с достоинствами второй, и наоборот. Существует технология построения моделей элементов физических устройств с помощью бинаправленных (ненаправленных) графов связи, которая дает шанс программам первой группы однозначно занять лидирующую позицию.

Modelica – свободно распространяемый объектно-ориентированный язык для моделирования сложных физических систем. Modelica обеспечивает возможность создания широкого диапазона моделей различных типов: механических, электрических, гидравлических, химических и др. Язык основан на концепции блоков с контактами, при соединении которых необходимые уравнения генерируются автоматически.

Он 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 «класс». 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 Math Works MATLAB Simulink, обеспечивает поддержку таких стандартов, как ACSL, M-file, Simnon, также поддерживается возможность использования функций и процедур, написанных на языке C. 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. В настоящее время для задач моделирования шире всего применяются MATLAB и GPSS.

Math Works MATLAB предназначен для проектирования систем управления; анализа данных; обработки изображений; цифровой обработки сигналов; визуализации полученных результатов и разработки собственных приложений. MATLAB помимо обычных языковых конструкций, позволяющих выполнять процедурное, объектно-ориентированное и визуальное программирование, содержит большое количество встроенных алгоритмов для математических расчетов и графической визуализации, для анализа и визуализации данных, численных и символьных вычислений, создания инженерной и научной графики, имитационного моделирования, программирования, разработки приложений и графического интерфейса пользователя (GUI).

Программы MATLAB являются платформонезависимыми, поэтому пользовательские программы могут при необходимости быть перенесены на любую платформу без изменения.

Открытая архитектура облегчает применение MATLAB для изучения языковых конструкций, принципов программирования и создания пользовательских приложений.

MATLAB содержит специализированные графики, помогающие понять работу сложных систем, а также представить результаты их исследования. С помощью MATLAB можно создавать высококачественную графику для визуализации динамических процессов в исследуемых ЭК и презентации.

Набор специализированных приложений (тулбоксов) позволяет обрабатывать сигналы и изображения, разрабатывать системы управления, проводить исследования с использованием нейронных сетей и др. Имеются тулбоксы, расширяющие функциональные возможности программы в сборе данных, создании отчетов и написании программ, включающих процедуры на языках C/C++ или Fortran, функционирующих в среде MATLAB или в виде независимых приложений.

В MATLAB входят специализированные Toolbox (пакеты прикладных программ) для решения математических задач (табл. 1).

Пакет прикладных программ	Назначение
NAG Foundation	Библиотека математических функций The Numerical Algorithms Group Ltd
<u>Spline</u>	Сплайн-аппроксимация
Statistics	Статистический анализ данных и моделирование методом Монте-Карло
Пакет прикладных программ	Назначение
Optimization	Нахождение экстремумов линейных и нелинейных функционалов при наличии связей и ограничений
Partial Differential Equations	Решение уравнений в частных производных
Symbolic Math	Символьная математика
Extended Symbolic Math	Расширенная символьная математика (включает систему Maple)

MATLAB также содержит интерфейс для вызова Java-процедур; коммуникационный интерфейс (последовательный порт) для связи с внешним оборудованием; современные инструменты проектирования графического пользовательского интерфейса. Он поддерживает популярные форматы файлов – CDF (Common Data Format) и FITS (Flexible Image Transport System), а также файлов данных HDF и HDF-EOS.

MATLAB Simulink и его приложения

Для моделирования, имитации и анализа динамических систем применяется интерактивный инструмент Math Works MATLAB Simulink. Он позволяет при помощи графических блок-диаграмм моделировать динамические системы, исследовать их работоспособность. Simulink полностью интегрирован с MATLAB, обеспечивая немедленный доступ к широкому спектру инструментов анализа и проектирования. Simulink также интегрируется с приложением MATLAB Stateflow для моделирования поведения, управляемого событиями. Эти преимущества делают Simulink наиболее популярным инструментом для проектирования систем управления и коммуникации, цифровой обработки и других приложений.

Simulink Performance Tools содержит четыре инструмента для пользователей Simulink, проектирующих крупномасштабные, сложные модели: Simulink Accelerator ускоряет выполнение кода при имитации моделей от двух до десяти раз; Model Differencing графически подчеркивает отличия между двумя моделями; Model Profiler собирает данные функционирования и генерирует HTML-отчет; Model Coverage выдает отчет о возможных путях имитации модели.

Для мониторинга сигналов и параметров в Simulink-моделях применяется приложение Dials & Gauges Blockset.

При создании и настраивании отчетов из MATLAB, Simulink и Stateflow моделей и данных в различных выходных форматах, включая HTML, RTF, XML и SGML, применяются приложения MATLAB Report Generator и Simulink Report Generator. Это дает возможность автоматически документировать создаваемые системы, включая в них любые данные из рабочей среды MATLAB.

MATLAB Compiler включает MATLAB C/C++ Math and Graphics Libraries. Он позволяет компилировать m-файлы в коды на C и C++. Это дает возможность создавать независимые программные продукты. Компилятор содержит новые алгоритмы оптимизации, существенно увеличивающие скорость выполнения. Модуль для Visual Studio позволяет использовать MATLAB-компилятор из Microsoft Visual Studio.

В MATLAB включены специализированные Toolbox, предназначенные для автоматизации проектирования систем управления (табл. 2).

Пакет прикладных программ	Назначение
Control System	Моделирование, анализ и проектирование как непрерывных, так и дискретных во времени линейных автоматических систем управления
<u>Nonlinear Control Design (NCD) Blockset</u>	Проектирование нелинейных систем
Fuzzy Logic	Проектирование, моделирование и анализ систем с нечетким откликом. Он имеет средства для преобразования входных данных в выходные данные системы правил и связей произвольной сложности, выраженных обычным языком. Системы могут быть имитированы в MATLAB или включены в блочные диаграммы <u>Simulink</u> с возможностью генерации кода для независимого выполнения
LMI Control	Синтез систем управления на основе линейных матричных неравенств. Содержит функции для проектирования и анализа таких характеристик систем управления, как помехоустойчивость, производительность и др.
Model Predictive Control	Управление с эталонной моделью. <u>Полезен</u> для управления системами с большим количеством входных и выходных переменных, имеющих много связей
M-Analysis and Synthesis	<u>μ</u> -анализ и синтез. Содержит набор функций для использования в анализе и проектировании устойчивых линейных систем со многими переменными

Robust Control	Робастное управление. Содержит специализированный набор инструментов для анализа и синтеза систем управления, устойчивых по отношению к случайным возмущениям
Quantitative Feedback Theory (QFT) Control Design	Проектирование робастных систем с обратной связью
Neural Network	Нейронные сети. Применяется в областях, в которых формальный анализ чрезвычайно труден или невозможен, как, например, распознавание образов, идентификация и управление нелинейными системами
Пакет прикладных программ	Назначение
System Identification	Идентификация параметров в системах управления
Frequency Domain System Identification	Идентификация в частотной области
<u>xPC Target</u>	Моделирование и тестирование взаимосвязанного оборудования, управляющих систем и систем цифровой обработки сигналов
Communications	Разработка современных коммуникационных систем, включая моделирование в реальном масштабе времени

В MATLAB также включены специализированные приложения, предназначенные для автоматизации проектирования в реальном времени (табл. 3).

Таблица 3

Приложение	Назначение
Real-Time Workshop	Дополняет <u>Simulink</u> , позволяя автоматически генерировать С-код прямо из блочных диаграмм пакета <u>Simulink</u>
Real-Time Workshop Embedded Coder	Расширяет возможности приложения Real-Time Workshop, позволяя осуществлять генерацию высококачественного встраиваемого кода для <u>Simulink</u> -моделей, и осуществляет генерацию кода из <u>Stateflow</u>
Real-Time Windows Target	Обеспечивает простую в использовании среду макетирования в реальном времени, полностью контролируемую через пользовательский интерфейс <u>Simulink</u>
<u>DSP Blockset</u>	Моделирование и симуляция, позволяющие проектировать системы обработки сигналов в режиме реального времени и осуществлять расчет этих систем. <u>DSP Blockset</u> позволяет использовать инструменты <u>Simulink</u> для обработки потоковых данных и выполнения многоступенчатых операций

Для моделирования и исследования электромеханических систем полезными будут следующие библиотеки MATLAB (табл. 4).

Таблица 4

Библиотека	Назначение
<u>Sim Power Systems</u>	Моделирование энергетических систем, включая устройства электротехники и промышленной электроники
<u>Sim Mechanics</u>	Моделирование механических систем, составленных из отдельных элементов, таких как механические блоки, соединения, узлы крепления, исполнительные элементы и датчики. С помощью этих элементов формируется описание сложной механической системы с присоединенными к ней другими динамическими моделями и контроллером
Aerospace	Моделирование, интегрирование и имитация авиационных, космических, реактивных и турбореактивных систем и подсистем в приложении <u>Simulink</u>
Filter Design	Проектирование, моделирование и анализ цифровых фильтров
Image Processing	Набор средств для анализа и обработки цифровых изображений

Специализированный Toolbox Sim Power Systems

Исходя из состава СУ для моделирования наиболее подходит Toolbox Sim Power Systems.

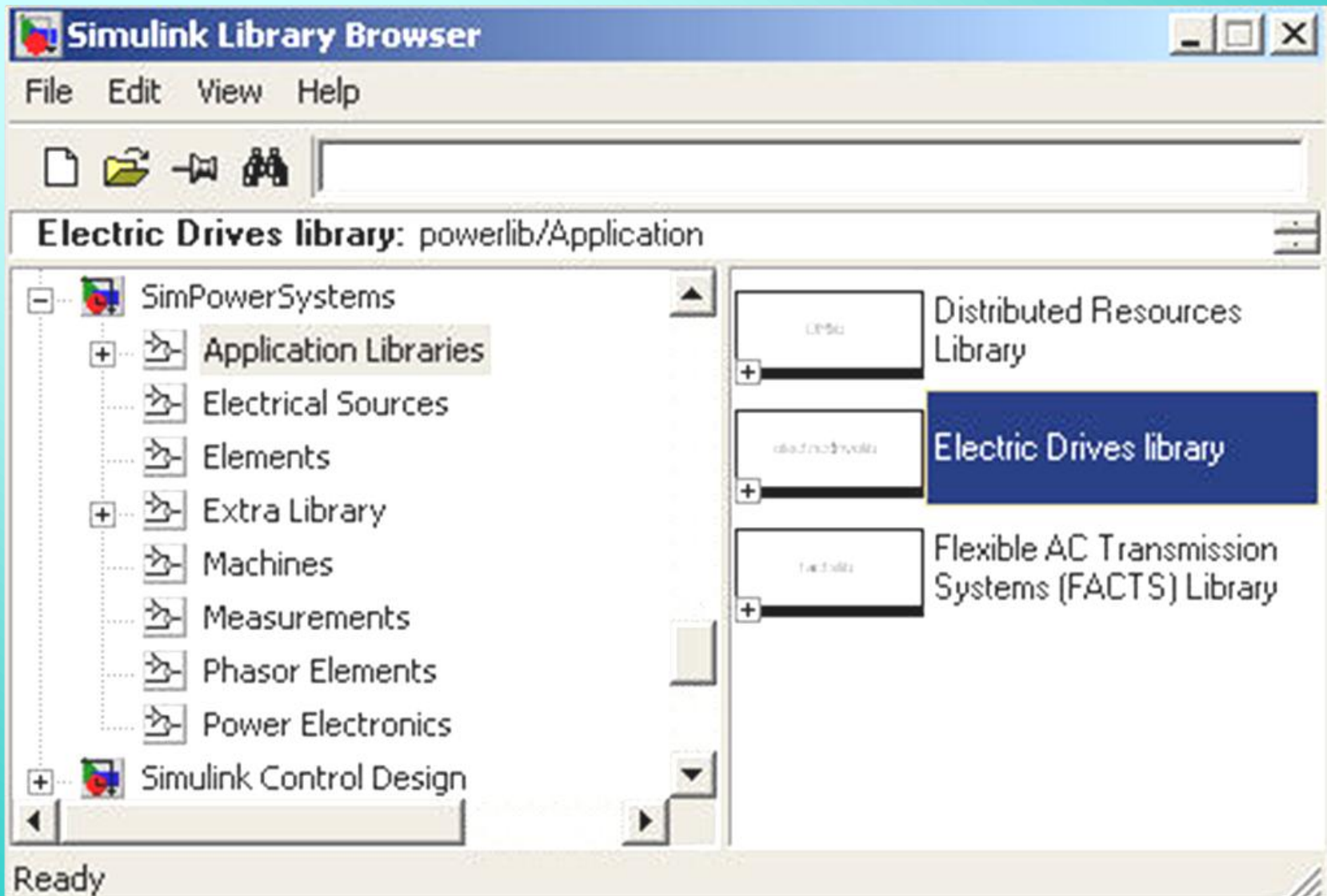


Рис. 1. Состав Toolbox Sim Power Systems

Toolbox Sim Power Systems имеет восемь основных разделов (рис.

1):

- 1) Application Library – библиотека инженерных приложений;
- 2) Electrical Sources – источники электрической энергии;
- 3) Elements – электротехнические элементы;
- 4) Extra Library – дополнительные библиотеки;
- 5) Machines – электрические машины;
- 6) Measurements – измерительные и контрольные устройства;
- 7) Phasor Elements – статический компенсатор;
- 8) Power Electronics – устройства силовой электроники.

В состав библиотеки Sim Power Systems входят модели пассивных и активных электротехнических элементов, источников энергии, электродвигателей, трансформаторов, линий электропередачи и другое оборудование. Имеется также возможность моделирования полупроводниковых устройств, включая системы управления ими.

В случае если в библиотеке нет нужной модели, пользователь имеет возможность создать свою модель как с помощью уже имеющихся в библиотеке моделей, так и на основе моделей основной библиотеки Simulink и управляемых источников тока или напряжения. Используя Toolbox Sim Power Systems, пользователь в состоянии за короткое время создать компьютерную модель сложной электромеханической системы.

Toolbox Sim Power Systems обеспечивает широкие возможности для расчета электротехнических устройств и систем и в настоящее время может считаться одним из лучших пакетов для моделирования устройств силовой электроники.

Рассмотрим отдельные библиотеки Toolbox Sim Power Systems.

Источники электрической энергии

В состав библиотеки Electrical Sources входят неуправляемые и управляемые источники постоянного и переменного напряжения и тока (рис. 2).

В полях окна настройки параметров каждого блока устанавливаются значения амплитуды напряжения, начальной фазы и частоты.

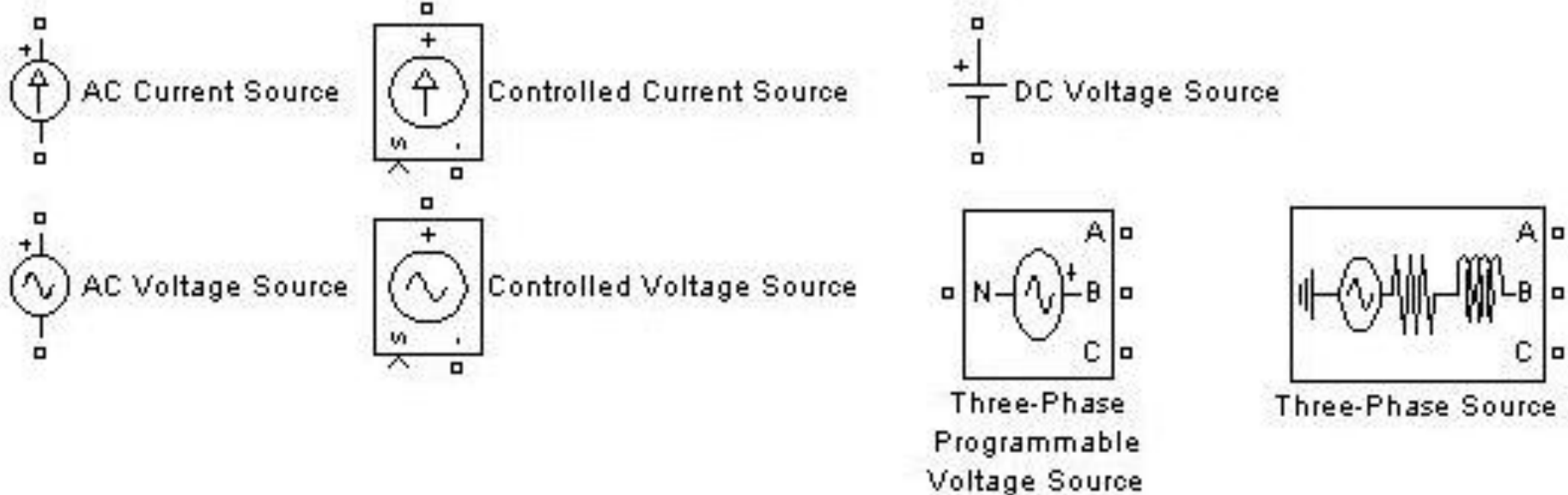


Рис. 2. Состав библиотеки Electrical Sources

Идеальный источник постоянного напряжения (DC Voltage Source). Модуль вырабатывает постоянное по уровню напряжение. В полях окна настройки параметров задаются: Amplitude (V) – амплитуда (В) выходного напряжения источника.

Идеальный источник переменного (синусоидального) напряжения (AC Voltage Source). В полях окна задания параметров задаются: Peak Amplitude (V) – амплитуда выходного напряжения источника; Phase (deg) – начальная фаза в градусах; Frequency (Hz) – частота источника; Sample time – параметр задает шаг дискретизации по времени выходного напряжения источника при создании дискретных моделей.

Модуль является идеальным источником напряжения, т. е. его собственное сопротивление равно нулю.

Идеальный источник переменного (синусоидального) тока (AC Current Source). В полях окна задания параметров задаются: Peak Amplitude (V) – амплитуда выходного тока источника; Phase (deg) – начальная фаза в градусах; Frequency (Hz) – частота источника; Sample time – параметр задает шаг дискретизации по времени выходного тока источника при создании дискретных моделей. Блок является идеальным источником тока, т. е. его собственное сопротивление равно бесконечности.

Управляемый источник напряжения (Controlled Voltage Source).

Модуль вырабатывает напряжение в соответствии с сигналом управления.

Управляемый источник тока (Controlled Current Source). Модуль

вырабатывает ток в соответствии с сигналом управления.

В полях задания этих элементов задаются начальные параметры – амплитуда, фаза и частота; выбирается тип источника (постоянного или переменного напряжения (тока)).

Трехфазный источник синусоидального напряжения (3-Phase Source). В полях окна задаются: *Phase-to-phase rms voltage (V)* – действующее значение линейного напряжения, В; *Phase angle of phase A (deg)* – начальная фаза напряжения в фазе А, ...°; *Frequency (Hz)* – частота источника, Гц; *Internal connection* – соединение фаз источника (Y – звезда; Yn – звезда с нулевым проводом; Yg – звезда с заземленной нейтралью); *X/R ratio* – отношение индуктивного и активного сопротивлений и др.

Электротехнические элементы

На рис. 3 представлен состав библиотеки Elements, которая содержит:

1. Последовательные и параллельные, одно- и трехфазные пассивные элементы R , L , C .
2. Одно- и трехфазные магнитосвязанные цепи (цепи с взаимной индукцией).
3. Линейные одно- и трехфазные трансформаторы (Linear Transformer, Three-Phase Transformer) и трансформатор с реальным магнитным сердечником, учитывающим его насыщение (Saturable Transformer).
4. Одно- и трехфазный ключ (Breaker), параметры (сопротивление, индуктивность) которого в открытом состоянии задаются в полях настройки. Там же задается состояние ключа (открыт, закрыт).

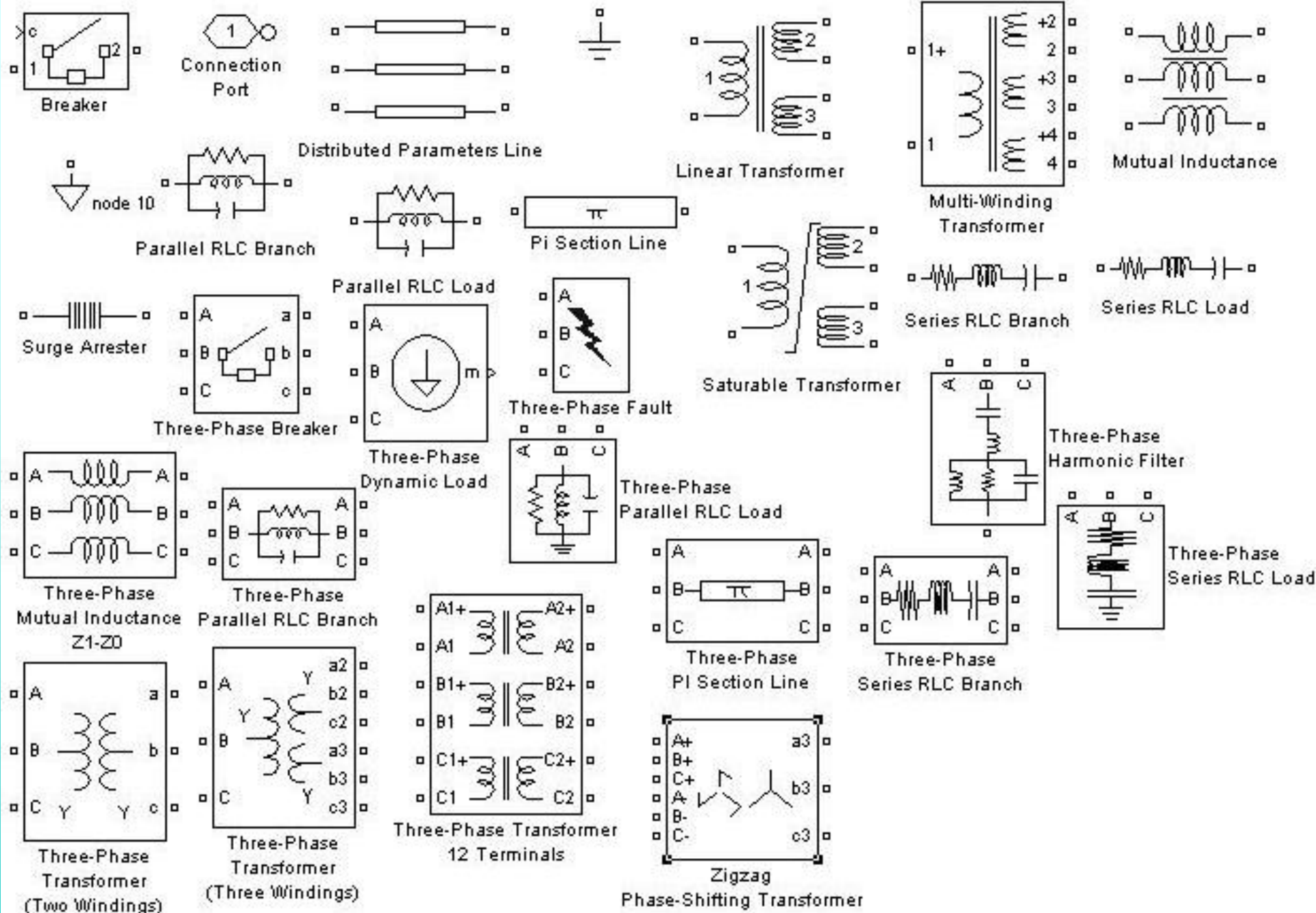


Рис. 3. Состав библиотеки Elements

5. Трёхфазный двух- и трехобмоточный трансформаторы (Three-Phase Transformer, Two windings, Three windings).
6. Модули, реализующие параметры однофазной и трехфазной линии передачи (PI Section Line, Distributed Parameters Line).
7. Нелинейный элемент (Surge Arrester), позволяющий сформировать требуемую нелинейную зависимость между входным и выходным сигналами.

Устройства силовой электроники

В библиотеке устройств Power Electronics (рис. 4) находятся модели полупроводниковых приборов (диод, тиристор, запираемый тиристор, транзисторы IGBT и MOSFET), а также универсальный мост, позволяющий моделировать одно-, двух- и трехфазные выпрямительные или инверторные схемы. Кроме того, в библиотеке присутствует модель трехуровневого моста и модель идеального ключа.

В дополнительных библиотеках Extras помещены дополнительные дискретные и непрерывные модули управления.

Измерительные и контрольные устройства

Состав библиотеки Measurements показан на рис. 5.

В библиотеке измерительных устройств находятся модули измерения мгновенных значений тока и напряжения, модуль измерения токов и напряжений в трехфазной системе, измеритель полного сопротивления участка электрической схемы, а также модуль Multimeter, позволяющий измерять переменные модулей без непосредственного подключения измерительных устройств.

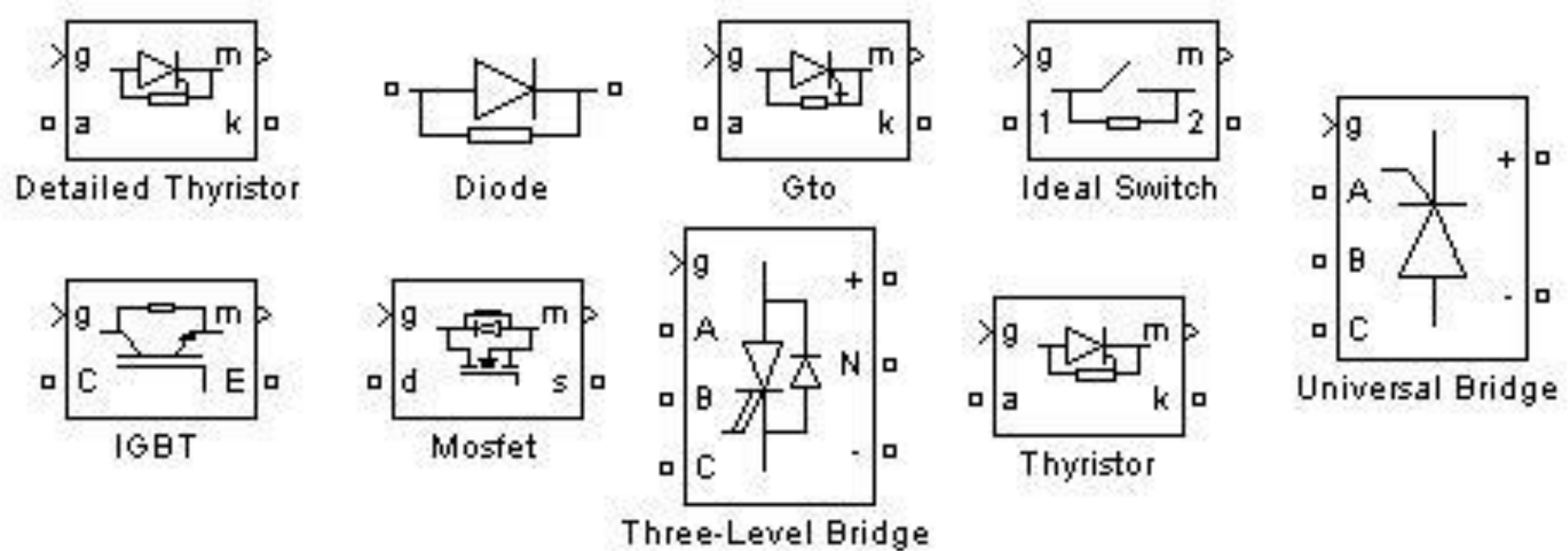


Рис. 4. Состав библиотеки Power Electronics

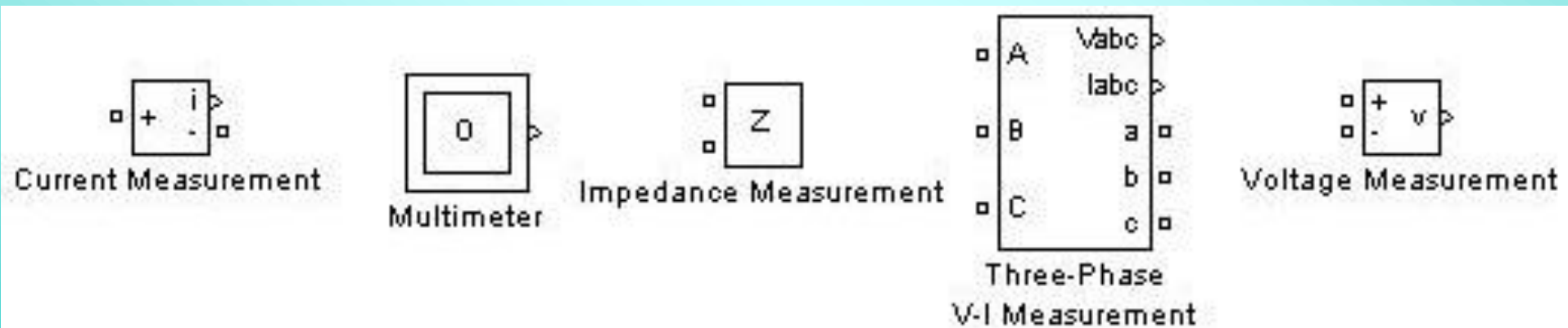


Рис. 5. Состав библиотеки Measurements

Выходными сигналами измерительных модулей являются сигналы Si-mulink, которые можно подать на модули отображения (Scope, Display) или на вход любого модуля.

Расширенные библиотеки

Среди дополнительных библиотек Extra Library (рис. 6) находится большое количество разнообразных модулей.

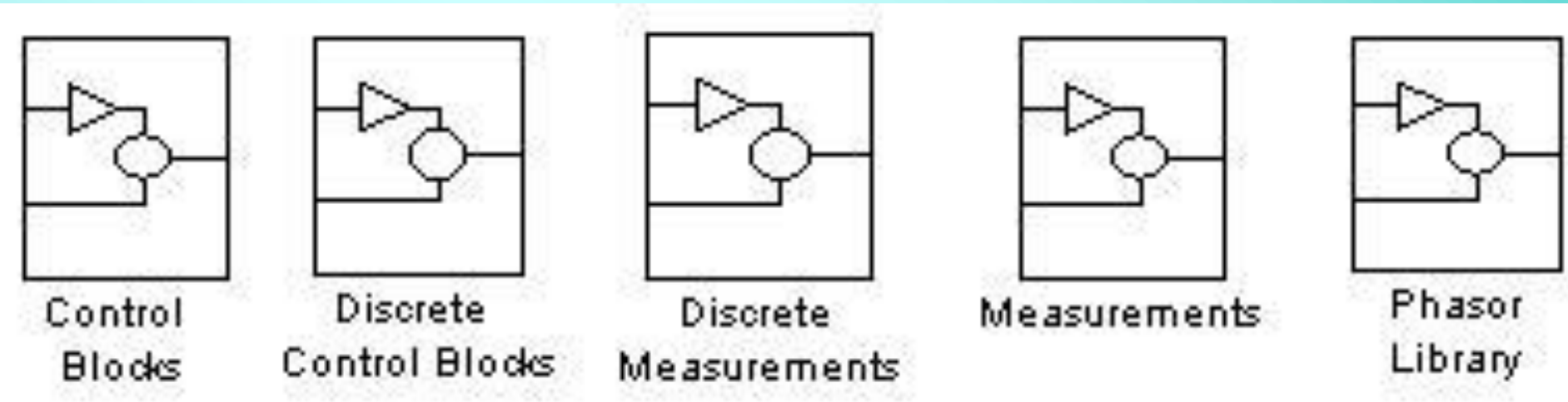


Рис. 6. Состав дополнительных библиотек Extra Library

В частности, библиотека дополнительных измерительных приборов, библиотека устройств управления преобразователями и т. д.

Трансформаторы

В состав библиотеки Sim Power System входят следующие модели трансформаторов (рис. 7):

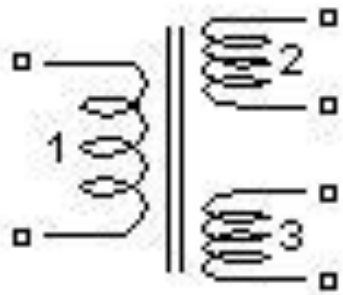
– линейный однофазный двух- или трехобмоточный трансформатор (Li-near Transformer);

– многообмоточный однофазный трансформатор (Multi-Winding Transformer);

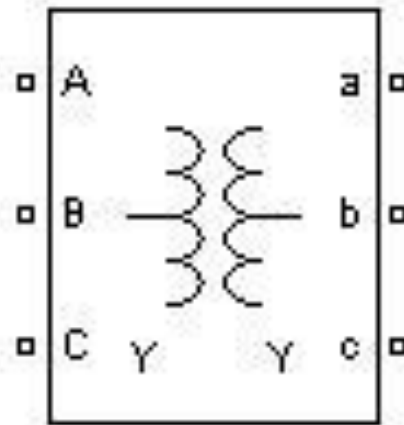
– двухобмоточный трехфазный трансформатор (Three-Phase Transformer (Two Windings));

- трехобмоточный трехфазный трансформатор (Three-Phase Transformer (Three Windings));
- двухобмоточный трехфазный трансформатор с выведенными наружу зажимами всех обмоток (Three-Phase Transformer 12 Terminals);
- трехобмоточный трехфазный трансформатор (Zigzag Phase-Shifting Transformer). У него первичная и одна из вторичных обмоток соединены в зигзаг, соединение другой вторичной обмотки выбирается из выпадающего списка.

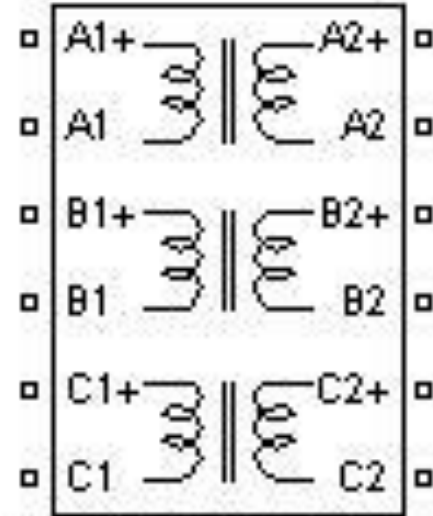
Для всех трансформаторов параметры схемы замещения задаются в относительных величинах. Эти параметры рассчитываются исходя из паспортных данных на трансформатор.



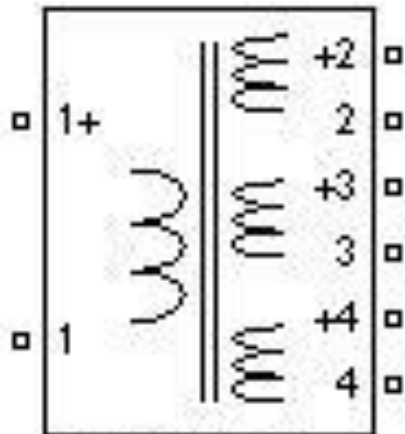
Linear Transformer



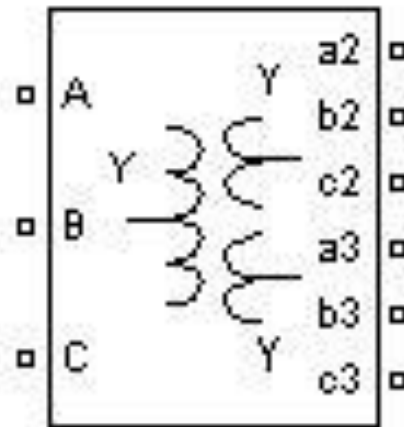
Three-Phase Transformer (Two Windings)



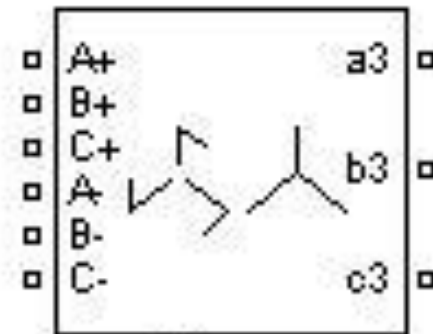
Three-Phase Transformer 12 Terminals



Multi-Winding Transformer



Three-Phase Transformer (Three Windings)



Zigzag Phase-Shifting Transformer

Рис. 7. Состав библиотеки Sim Power System

Моделирование полупроводниковых преобразователей

Кроме силовых полупроводниковых элементов в состав полупроводникового преобразователя, как правило, входят и другие элементы, к ним относятся: 1) неуправляемые и управляемые источники напряжения и тока; 2) дроссели, катушки индуктивности, конденсаторы; 3) силовые трансформаторы, измерительные трансформаторы; 4) коммутационно-защитная аппаратура (выключатели, переключатели, автоматы, предохранители и т. д.); 5) система управления (реализуется либо на элементах интегральной микросхемотехники, либо на контроллере); 6) сигнализация режимов работы.

При расчетах электромагнитных процессов в силовых преобразователях реальную электрическую цепь представляют в виде эквивалентной схемы, состоящей из отдельных идеальных участков, каждый из которых обладает только сопротивлением, только индуктивностью или только емкостью. Например, катушку можно заменить схемой двух последовательно соединенных участков, из которых первый обладает только сопротивлением, второй – индуктивностью, емкостью же катушки часто можно пренебречь или учесть ее параллельно подключенным конденсатором.

Соотношение между напряжениями, токами и мощностями цепи подчинены законом Ома, Фарадея, Кулона, Кирхгофа и Джоуля–Ленца. Все эти законы справедливы и в цепи переменного тока для мгновенных значений напряжения, токов и мощностей, если эти значения берутся для одного момента времени. Но при переменных напряжениях и токах электрические и магнитные поля тоже переменны и будут создавать в самой цепи переменные ЭДС; эти внутренние ЭДС, так же как и напряжение, приложенное к цепи, необходимо учитывать при применении основных законов.

Таким образом, расчеты цепей сводятся по существу к моделированию эквивалентных схем с сосредоточенными параметрами, что и рассматривается в дальнейшем. В цепях переменного тока напряжения и токи непрерывно изменяют свое направление. При математическом описании электрической схемы составление уравнений осуществляется по законам Ома и Кирхгофа. При этом необходимо задаться произвольным условным направлением напряжений и токов в элементах.

Математические модели электродвигателей постоянного тока

Модуль моделирования электродвигателя постоянного тока в пакете Sim Power System показан на рис. 8 (где TL – момент нагрузки; m – выходной измерительный порт; +A, –A, +F, –F – порты подключения внешних цепей), а его математическое описание имеет следующий вид:

$$u_a = L_a \frac{di_a}{dt} + R_a i_a + e_a;$$
$$u_f = L_f \frac{di_f}{dt} + R_f i_f;$$
$$e_a = k_E \omega \quad k_E = L_{af} i_f$$

– электромагнитная часть модели двигателя;

$$J \frac{d\omega}{dt} + B \omega + M_c + \text{sign } \omega M_H = k_T i_a$$

– механическая часть модели двигателя. Здесь u_a, i_a, e_a – напряжение, ток и противоЭДС якоря; u_f, i_f – напряжение и ток возбуждения; $L_a, R_a, L_f, R_f, L_{af}$ – параметры двигателя; ω, M_H, i_a – угловая скорость, момент нагрузки, ток якоря; J – момент инерции ротора с присоединенной нагрузкой; $B_{тр}$ – коэффициент вязкого трения; M_c – момент сухого трения. Параметры двигателя задаются в окне параметров при моделировании.

Как видно из описания, модель двигателя состоит из двух частей: электромагнитной и механической. Электромагнитная часть модели включает:

1. Якорную цепь (Armature circuit), которая присоединяется к внешней цепи при помощи портов +A, -A и содержит последовательно соединенные сопротивление, индуктивность (R_a, L_a), источник противоЭДС (FCEM) и измеритель тока якоря.

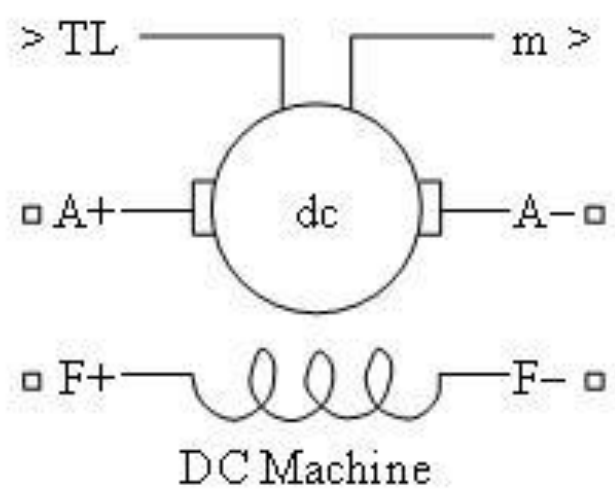


Рис. 5.8. Модель электродвигателя постоянного тока

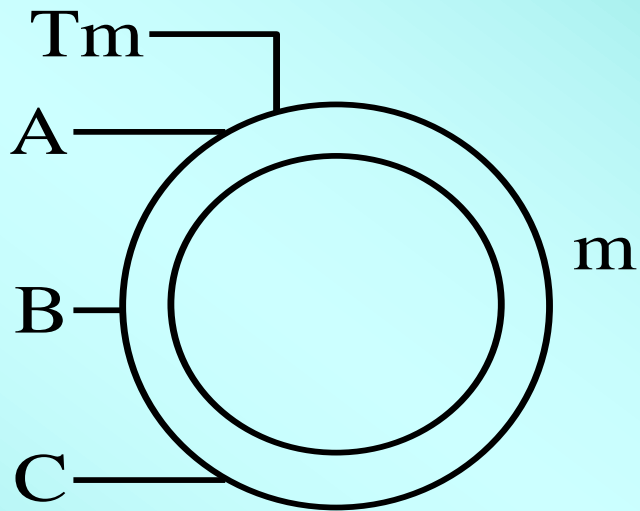
2. Цепь возбуждения (Field circuit), которая присоединяется к внешней цепи при помощи портов $+F$, $-F$, содержит последовательно соединенные сопротивление, индуктивность R_f , L_f и измеритель тока возбуждения.

На вход механической части подаются момент нагрузки (M_H), ток якоря (i_a), ток возбуждения (i_f). С выхода механической части снимаются противоЭДС двигателя (этот сигнал подается на электромагнитную часть модели).

Используя различные подключения внешних цепей к портам $+A$, $-A$, $+F$, $-F$, можно моделировать электродвигатели постоянного тока с независимым, параллельным и последовательным возбуждением.

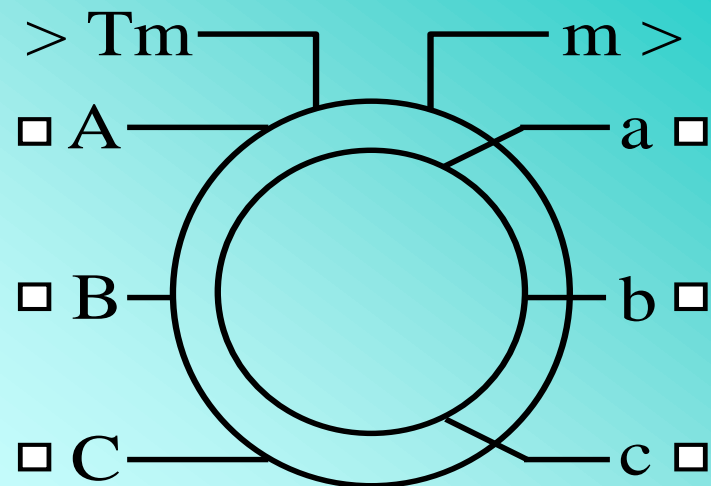
Математические модели асинхронных электродвигателей

Модули моделирования асинхронного электродвигателя (АД) в пакете Sim Power System показаны на рис. 9 (с короткозамкнутым – рис. 9, а и с фазным ротором – рис. 9, б).



Asynchronous Machine
SI Units

a



Asynchronous Machine
SI Units

b

Рис. 9. Модели асинхронных электродвигателей
(T_m – момент; m – выходной измерительный порт)

В основе модели лежит математическое описание АД во вращающейся системе координат $d, q, 0$, жестко связанных с ротором.

Математическая модель АД состоит из двух частей – электромагнитной и электромеханической. Электромагнитная часть модели описывается системой дифференциальных уравнений во вращающейся системе координат, которые обозначены через d и q :

$$U_{qs} = R_s i_{qs} + \frac{d\Phi_{qs}}{dt} + \omega \Phi_{ds};$$

$$\Phi_{qs} = L_s i_{qs} + L_m i'_{qr}, \quad L_s = L_{ls} + L_m;$$

$$U_{ds} = R_s i_{ds} + \frac{d\Phi_{ds}}{dt} + \omega \Phi_{qs};$$

$$\Phi_{ds} = L_s i_{ds} + L_m i'_{dr};$$

$$U'_{qr} = R'_r i'_{qr} + \frac{d\Phi'_{qr}}{dt} + (\omega - \omega_r) \Phi'_{dr};$$

$$\Phi'_{qr} = L'_r i'_{qr} + L_m i_{qs}; \quad L'_r = L'_{kr} + L_m;$$

$$U'_{dr} = R'_r i'_{dr} + \frac{d\Phi'_{dr}}{dt} + (\omega - \omega_r) \Phi'_{qr};$$

$$\Phi'_{dr} = L'_r i'_{dr} + L_m i_{ds}.$$

Электрохимическая часть АД описывается уравнениями

$$M_e = \frac{3}{2} p (\Phi_{ds} i_{qs} - \Phi_{qs} i_{ds});$$
$$\frac{d\omega_m}{dt} = \frac{1}{J} (M_e - F\omega_m - M_m); \quad \frac{d\theta_m}{dt} = \omega_m,$$

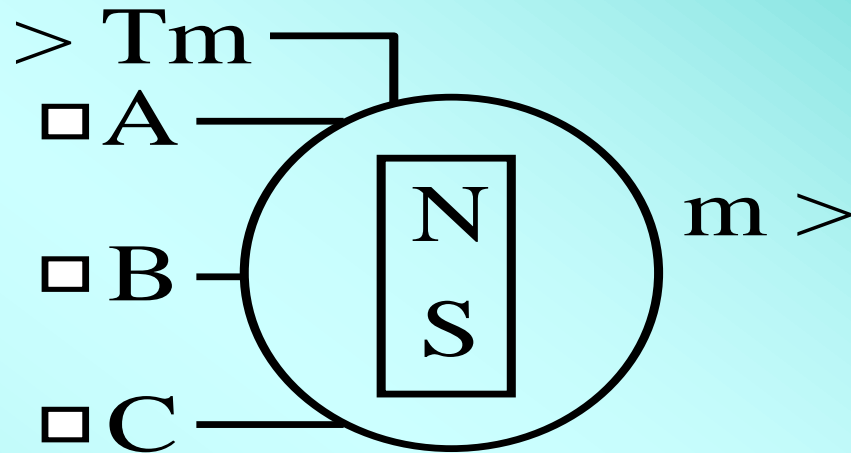
где M_e – электромагнитный момент электродвигателя, Н·м; U_{qs} – проекция пространственного вектора напряжения статора на ось q , В; R_s – сопротивление рассеяния статора, Ом; i_{qs} – проекция пространственного вектора тока статора на ось q , А; Φ_{qs} , Φ_{ds} – проекции пространственного вектора потокосцепления статора на ось q и d , В·с; L_s – полная индуктивность статора, Гн; L_m – взаимная индуктивность между статором и ротором, Гн; L_{ds} – индуктивность рассеяния статора, Гн; L_r' – приведенная индуктивность ротора; U'_{qr} – проекция пространственного вектора напряжения ротора на ось q , В; U_{ds} – проекция пространственного вектора напряжения статора на ось d , В; i'_{qr} – проекция пространственного вектора тока ротора на ось q , А; i_{ds} – проекция пространственного вектора тока статора на ось d , А; i'_{dr} – проекция пространственного вектора тока ротора на ось d , А; R_r' – приведенное сопротивление ротора, А; U'_{dr} – проекция про-

пространственного вектора напряжения ротора на ось d , В; L'_{lr} – индуктивность рассеяния ротора, Гн; Φ'_{qr} , Φ'_{dr} – проекции пространственного вектора потокосцепления ротора на ось q и d , В · с; ω_m – угловая скорость вращения ротора, рад/с; θ_m – угол поворота ротора, рад; p – число пар полюсов.

Математические модели синхронных электродвигателей

Модуль моделирования синхронного электродвигателя (СД) в пакете Sim Power System показан на рис. 10.

Математическая модель магнитоэлектрического синхронного электродвигателя (МСД) состоит из двух частей – электромагнитной и электромеханической. Математическое описание электромагнитной части модели зависит от характера распределения поля в зазоре. При синусоидальном распределении поля электромагнитная часть СД описывается системой дифференциальных уравнений во вращающейся системе координат, которые обозначены через d и q :



Permanent Magnet Synchronous Machine

Рис. 10. Модель МСД

$$\frac{di_d}{dt} = \frac{1}{L_d} u_d - \frac{R}{L_d} i_d + \frac{L_q}{L_d} p \omega_r i_q;$$

$$\frac{di_q}{dt} = \frac{1}{L_q} u_q - \frac{R}{L_q} i_q + \frac{L_d}{L_q} p \omega_r i_d - \frac{\lambda p \omega_r}{L_q};$$

$$M_e = \frac{3}{2} p \left[\lambda i_q + (L_d - L_q) i_d i_q \right],$$

где i_d, i_q, u_d, u_q – амплитуды тока [А] и напряжения статора [В] по осям d, q ; λ – амплитуда потока постоянного магнита ротора, сцепленного с обмоткой статора; M_e – электромагнитный момент, Н·м; ω_r – угловая скорость вращения ротора, рад/с; L_d, L_q – индуктивности по осям d, q , Гн; R – сопротивление статора, Ом; p – число пар полюсов.

При трапецеидальном распределении магнитного поля в зазоре электромагнитная часть СД описывается системой дифференциальных уравнений в фазовых координатах неподвижной системы координат:

$$\frac{di_a}{dt} = \frac{1}{3L_s} \left[2u_{ab} + u_{bc} - 3R_s i_a + \lambda p \omega_r (-2\Phi'_a + \Phi'_b + \Phi'_c) \right];$$

$$\frac{di_b}{dt} = \frac{1}{3L_s} \left[-u_{ab} + u_{bc} - 3R_s i_a + \lambda p \omega_r (\Phi'_a + \Phi'_b + \Phi'_c) \right];$$

$$\frac{di_c}{dt} = - \left(\frac{di_a}{dt} + \frac{di_b}{dt} \right);$$

$$M_e = p\lambda (\Phi'_a i_a + \Phi'_b i_b + \Phi'_c i_c),$$

где u_{ab}, u_{bc} – линейные напряжения, В; i_a, i_b, i_c – фазные токи, А; $\Phi'_a, \Phi'_b, \Phi'_c$ – единичные трапецидальные функции.

Электромеханическая часть МСД описывается системой уравнений одинаково для обеих моделей:

$$\frac{d\omega_r}{dt} = \frac{1}{J} (M_e - F\omega_r - M_m);$$

$$\frac{d\theta_r}{dt} = \omega_r,$$

где J – суммарный момент инерции на валу, $\text{кг} \cdot \text{м}^2$; F – коэффициент вязкого трения; θ_r – угол поворота ротора, рад; M_m – внешний момент, приложенный к валу МСД, $\text{Н} \cdot \text{м}$.

Пример моделирования ленточного конвейера с асинхронным электродвигателем

Наиболее распространенными по конструкции являются ленточные конвейеры. Модель электропривода ленточного

конвейера представлена на рис. 11. Модель содержит следующие блоки:

- модуль асинхронного электродвигателя (Asynchronous Machine SI Units);
- трехфазный автономный инвертор на IGBT-транзисторах (Universal Bridge);
- модуль вычисления потокосцепления ротора (Flux Calculator);
- модули ABC-dq и dq-ABC, осуществляющие преобразование неподвижной трехфазной системы координат во вращающуюся двухфазную и вращающуюся двухфазную в неподвижную трехфазную;
- модуль задания потокосцепления (Phir);
- регулятор тока по оси d (Phir controller);
- регулятор скорости (Speed Controller);
- модуль задания скорости (Speed).

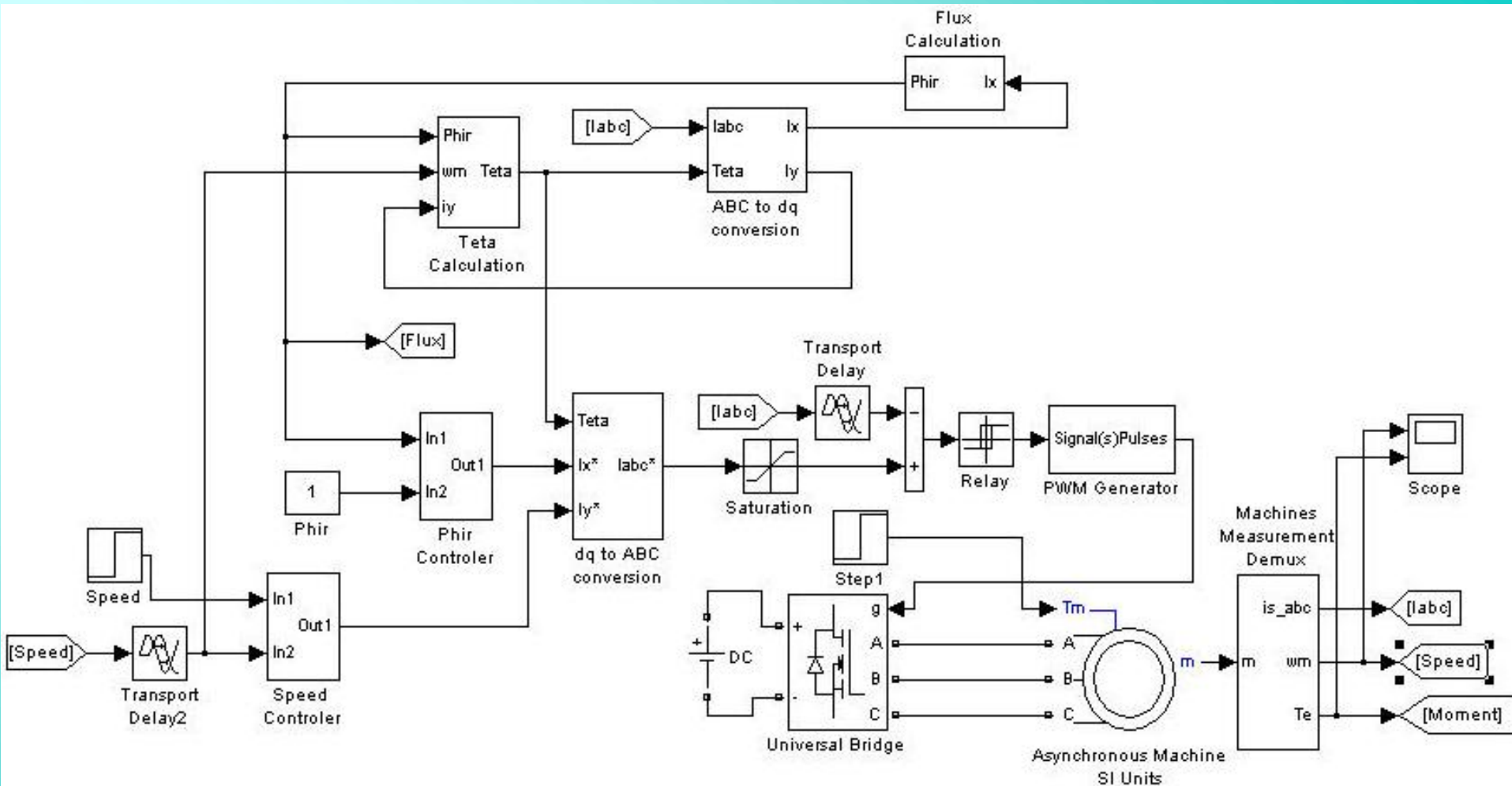


Рис. 11. Модель электропривода ленточного конвейера

Мощность приводного электродвигателя ленточного конвейера определяется как

$$N = \frac{k_3 N_0}{\eta_{\Pi} \eta_0} = \frac{k_3 N_{\text{В}}}{\eta_0},$$

где N_0 и $N_{\text{В}}$ – расчетная мощность, соответственно, на приводном органе (барабан, звездочка, шкив) и на валу приводного органа; η_{Π} и η_0 – КПД, соответственно, приводного органа и передаточного механизма.

Расчетная мощность [кВт] на валу приводного органа вычисляется по следующей приближенной формуле:

$$N_{\text{В}} = \frac{Q(\omega_0 L \pm H)}{367},$$

где L – длина трассы; H – высота подъема; $\omega_0 = 0.1 \dots 4.2$ – обобщенный коэффициент сопротивления, причем меньшие значения относятся к несущим конвейерам (ленточным, пластинчатым), а большие – к волочащим (скребковым, винтовым). Этот коэффициент зависит от сопротивления опорных элементов, отклоняющих, приводных, загрузочных, разгрузочных и очистительных устройств конвейера.

Программные средства для визуализации движений механизмов и процессов обработки вещества (материала)

Эффективность проектирования повышается, если при исследовании СУ применяется визуализация динамических процессов в ней. С этой целью, во-первых, применяются различные программные средства (OpenGL, Direct3D, язык VRML (Virtual Reality Modeling Language)), во-вторых, разрабатываются программные комплексы визуализации с помощью языков высокого уровня, например, C++.

Вначале остановимся на втором подходе. Для работы на этапе макропроектирования программная система должна иметь библиотеку модулей, отражающих конфигурацию и свойства типовых элементов промышленных механизмов или технологических комплексов и эмулятор программируемого контроллера (ЭПК), функциональным назначением которого является непосредственно реализация алгоритма управления.

Развивая блочно-модульную идеологию построения средств и систем управления до уровня механизмов, агрегатов и комплексов, можно выделить группы оборудования, для которых характерны общие функциональные задачи управления в технологическом процессе (функциональные модули). В соответствии с этим можно подготовить программные модели модулей и программные блоки, решающие задачу управления модулями. Из таких модулей можно формировать блоки-комплексы и в соответствии с этой идеологией обеспечить экономичную технологию проектирования компьютерных систем управления любыми СУ.

В результате вышеизложенного подхода может быть определена объектно-ориентированная модель программного обеспечения и определены все основные объекты, включая данные и основные операции, требуемые для выполнения системных функций, а также следующие ее компоненты:

- диаграммы классов, содержащие ключевые классы предметной области;
- спецификации классов, описывающие семантику классов, их отношения, атрибуты и ключевые операции;

– диаграммы взаимодействий, которые изображают, как

взаимодействуют объекты при выполнении системных функций;

– схемы организации функционирования электромеханических

комплексов, классификации/детализации и схемы потоков данных,

используя связи родитель/потомок с различными основаниями

классификации;

– словарь данных, в котором перечислены все основные сущности,

относящиеся к предметной области, включая классы, отношения и

атрибуты.

Организационная структура отображает собой сложившуюся модель функционирования комплекса. При декомпозиции выделяются функциональные подсистемы, определяемые типовыми решаемыми задачами.

В основе предлагаемой методики визуализации динамических процессов происходящих в СУ, лежит одновременное решение дифференциальных уравнений и в зависимости от этого динамическое изменение в пространстве или плоскости положения графических примитивов (точек, линий, окружностей, прямоугольников и др.), отображающих движения механизмов оборудования, входящего в состав ЭК (табл. 5).

Визуализация более сложных технологических объектов

производится с использованием совокупности простейших графических примитивов.

Основная проблема, которая возникает при этом – время решения и объем информации о перемещении. Время решения зависит от порядка дифференциальных уравнений, которыми описывается СУ технологическим объектом. Но если учесть, что СУ на первой стадии проектирования отлажена, т. е. настроены все ее контуры (рассчитаны регуляторы), то все описание можно свести в этом случае к звену первого порядка, т. е. к передаточной функции замкнутого внешнего контура, а также математического описания свойств обрабатываемого материала.

Это позволит повысить быстродействие в получении выходной координаты, характеризующей исполнительный орган, и, как следствие, уменьшить объем информации о перемещении. Но упрощение математического описания потеряет всякий смысл, так как частоты процессора ПК приближаются к той частоте, когда все расчеты будут выполняться в реальном времени. Поэтому основной задачей является формирование графических примитивов и их динамическое отображение в зависимости от результатов компьютерного моделирования.

Таблица 5

Графические примитивы	Вид визуализации	Характеристика исполнительных механизмов и приводов	Рабочий орган или обрабатываемое вещество	Группы оборудования [10]
Точка на плоскости		Один механизм и привод	Пресс, штамп, молот, кабина лифта, нож	3, 8
Точка в пространстве		<u>Три и более механизмов и приводов</u>	<u>Схват</u> , ковш, крюк, нож	7, 8, 10
Линия на плоскости		Один механизм и привод	Транспортерная и конвейерная ленты, рольганг	6, 8
Окружность на плоскости с постоянным радиусом			Валок, пила, вал, барабанные ножницы	4, 6
Окружность на плоскости с изменяющимся радиусом			Валок, вал	6
Прямоугольник на плоскости или утолщенная линия		—	Заготовка металлическая, картон, пластмасса	3, 4, 6, 8

Для решения вышепоставленных задач применяют математический пакет MATLAB, дописав недостающие функции (программные единицы).

Существует несколько способов взаимодействия системы MATLAB с внешними программами:

1. Можно написать расширение MATLAB на обычных языках программирования, таких как C/C++, Fortran и т. п.
2. Можно вызвать вычислительное ядро MATLAB из своей программы и получить результат вычислений среды MATLAB в свою программу. В этом случае MATLAB выступает как своего рода вычислительный сервер для внешней программы.
3. Средой MATLAB можно управлять при помощи команд DDE (Dynamic Data Exchange) или ActiveX (OLE) Automation.
4. Можно просто обмениваться данными со средой MATLAB через MAT-файлы, структура которых описана в документации к системе.

Для того чтобы написать модуль, расширяющий набор функций MATLAB, нужно создать обычную динамическую библиотеку (DLL) для Microsoft Windows со специальным набором функций (интерфейсом). Для этого выбрана система программирования Borland C++ Builder версия 6.0, которая является одной из систем программирования для разработок, ориентированных на концепцию визуально-событийного программирования. Borland C++ Builder версия 6.0 позволяет разрабатывать 32-разрядные приложения для операционных систем Windows; корректировать производные классы, функцию WinMain и пр., что при профессиональном подходе уменьшает вероятность ошибок в программе и сама программа имеет меньший код по сравнению с другими системами программирования в связи со спецификой языка С.

В документации по системе MATLAB для подобных расширений употребляется термин MEX-файл (Matlab Extension). Специально назначать MEX в качестве расширения имени файла необязательно, MATLAB может прекрасно работать и со стандартным расширением подобных динамических библиотек – DLL.

Папка `\matlab\extern` на диске, на котором установлена система, содержит все необходимые файлы-заголовки для программ на C (каталог `\matlab\extern\include`), а также несколько примеров модулей (`\matlab\extern\examples`), реализующих некоторые расширения системы, которые приведены целиком в виде исходных текстов программ на C.

Кроме того, имеется папка `\matlab\extern\src` в которой приведен исходный текст некоторых вспомогательных функций, весьма облегчающий отладку модулей расширения MATLAB.

В модулях-расширениях MATLAB (MEX-файлах) для обмена параметрами всех типов с вычислительной средой MATLAB используется одна структура – mxArray (Matlab Extension Array).

Программный интерфейс создаваемой динамической библиотеки достаточно прост, в нем должна быть экспортирована единственная функция с двумя параметрами – входным массивом структур mxArray и выходным массивом структур mxArray.

Прототип интерфейсной функции объявлен в заголовочном файле `\matlab\extern\include\mex.h`.

Порядок создания dll-файла в Borland C++ Builder версия 6.0, вызываемого из MATLAB:

1. File->New

Откроется диалог «New» вкладка «Projects»:

а) выбираем тип создаваемого проекта: Win32 Dynamic-Link Library

б) заполняем Project Name: {prog1 }

Location: {указываем путь к нужной директории}

2. File->New

Откроется диалог «New» вкладка «Files»:

а) выбираем тип создаваемого файла: C++ source file

б) заполняем File Name: {prog1 }

3. Скопировать в директорию проекта файл matlab.lib (прилагается)

4. Добавить скопированный файл к проекту:

Project->Add ToProject->Files-> matlab.lib

5. Работа с C++-м файлом, входящим в проект:

обязательно включить следующие хедера:

```
#include <windows>
```

```
#include <mex.h>
```

```
#include <matrix.h>
```

.....

(Далее идет текст нужной программы (функции))

6. Добавить к проекту файл {prog1.def}:

а) File->New: откроется диалог «New» вкладка «Files»:

б) выбираем тип создаваемого файла: text files

в) заполняем File Name: {prog1.def}

7. Открыть {prog1.def} и заполнить его:

```
LIBRARY «{prog1}»
```

```
EXPORTS
```

```
mexFunction
```

8. Tool->Options->Вкладка «Directories» включить путь:

```
C:\MATLAB\extern\include
```

9. Project->Settings: Output file name: {prog1.dll}

10. F7 (Build) -> создастся файл {prog1.dll}

11. Открываем MATLAB

12. File->Set Path: указываем путь к созданному dll-файлу

13. С командной строки MATLAB: >>{prog1} а выполнится нужный

C++-й код.

Если для исследования КЭС применяется подсистема моделирования динамических процессов Math Works MATLAB Simulink, то для создания нужного блока целесообразно применять технологию S-функций (Simu-link-функции). С помощью языков программирования (С, С++, Ada или Fortran) пользователь может создавать описание сколь угодно сложных блоков (блоков, обеспечивающих взаимодействие системы Simulink с аппаратными средствами; блоков на основе математических уравнений; блоков, реализующих анимационные возможности) и включать их в Simulink-модель, при этом они ничем не отличаются от стандартных библиотечных блоков системы Simulink.

Создаваемые блоки могут реализовывать непрерывные, дискретные или гибридные (дискретно-непрерывные) модели. S-функции, созданные на языках C, C++, Ada или Fortran, компилируются в исполняемые файлы с расширением *.dll.

Для решения поставленной задачи можно также применить следующие программные продукты.

OpenGL – это графический стандарт, который обеспечивает широкие возможности несмотря на то, что поддерживает простейшую модель программирования. Её процедурный интерфейс позволяет программистам легко и эффективно описывать как простые, так и комплексные задачи воспроизведения.

OpenGL является программным интерфейсом для графических устройств и включает в себя несколько сотен функций и процедур, которые позволяют программисту определять объекты и сложные операции для создания высококачественных образов. OpenGL является низкоуровневым, процедурным API-языком, что вынуждает программиста создавать точную последовательность шагов, чтобы построить результирующую растровую графику (императивный подход). Это является основным отличием от дескрипторных подходов, когда вся сцена передается в виде структуры данных (чаще всего дерева), которая обрабатывается и строится на экране.

С одной стороны, императивный подход требует от программиста глубокого знания законов трёхмерной графики и математических моделей, с другой стороны, даёт свободу внедрения различных инноваций.

OpenGL поддерживает только графический вывод и не работает ни с какими другими периферийными устройствами (мышь, джойстик, клавиатура, звуковая карта). Для доступа к этим устройствам под ОС Windows необходимо использовать Win32 API или DirectX.

Система Direct3D (как и всё семейство DirectX) построена на основе COM – составной объектной модели (Component Object Model), технологии, которая используется подсистемой связывания и внедрения объектов (OLE/ActiveX). COM-объекты при программировании на C++ играют ту же роль, что и библиотеки динамической компоновки (DLL) при программировании для Windows на языке C++. Это не означает, что COM-объекты могут использоваться только в приложениях, написанных на C++, однако в программах, написанных на C++, при обращении к интерфейсу COM-объекта необходимо явно указывать, что вызов функции производится посредством таблицы указателей (аналог vtable для виртуальных функций классов C++).

Интерфейс Direct3D в отличие от других API для

программирования 3D-графики состоит из нескольких уровней:

- верхнего уровня или «абстрактного режима» (Retained Mode);
- нижнего уровня или «непосредственного режима» (Immediate Mode);
- драйверов устройств.

Кроме того, Direct3D включает в себя два абсолютно независимых набора функций, предоставляющих одни и те же функции для работы с 3D-графикой на низком уровне:

- буфер выполнения (Execute Buffers);
- рисование примитивов (Draw Primitives).

Основные возможности Direct3D:

- переключаемый буфер глубины;
- освещение Ламберта и Гуро;

- множество источников разных типов;
- полная поддержка работы с материалами и текстурами,

мипмэппинг;

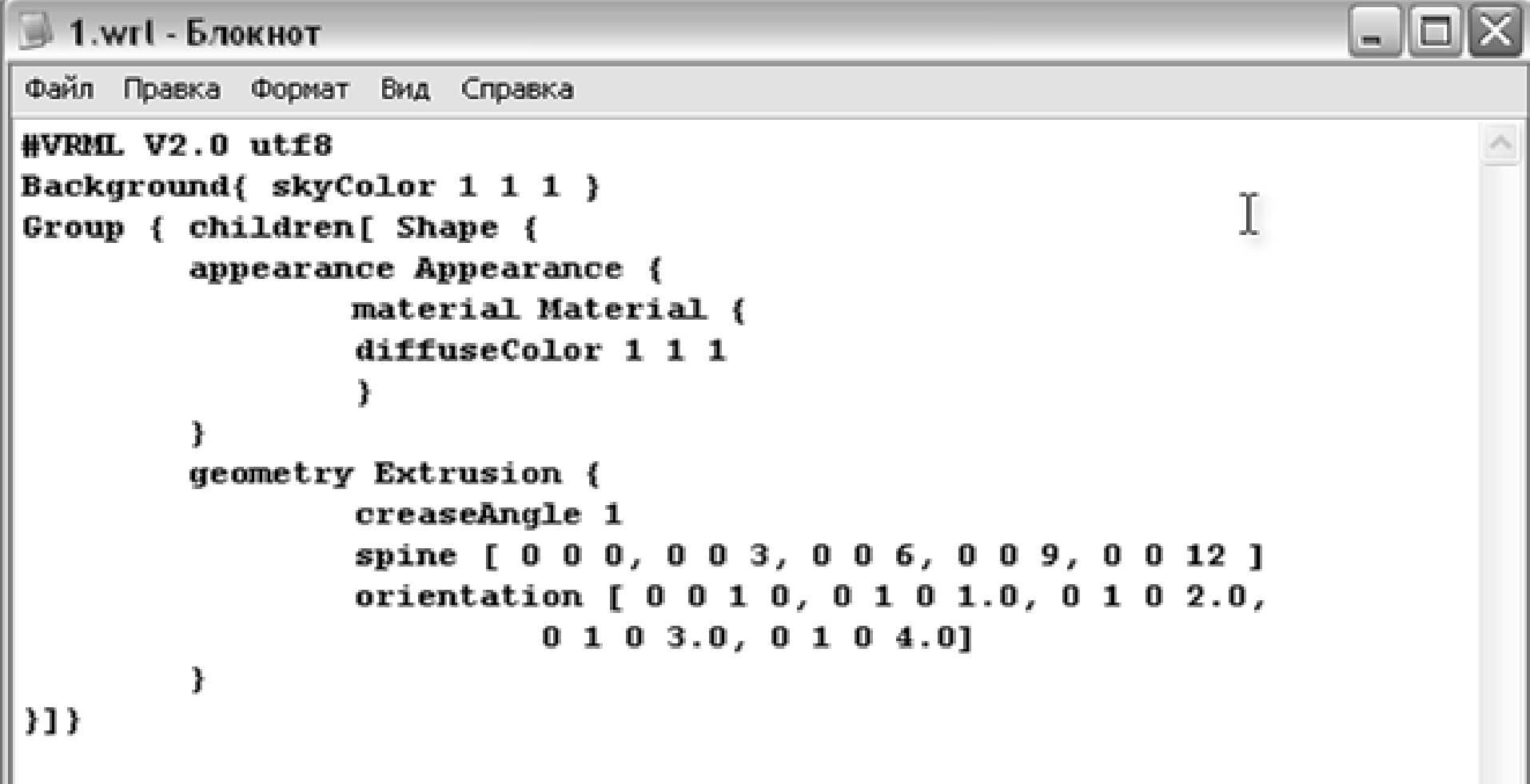
- быстрые эффективные драйверы программной эмуляции;
- трансформации и отсечение;
- полная поддержка на NT.

Библиотека Direct3D постоянно изменяется и расширяется с целью поддержки всех возможных ускорителей новых наборов инструкций.

Новые возможности библиотеки часто заимствуются из расширений библиотеки OpenGL.

Язык VRML (Virtual Reality Modeling Language – язык

моделирования виртуальной реальности) – предназначен для описания трехмерных изображений и оперирует объектами, описывающими геометрические фигуры и их расположение в пространстве. VRML-файл представляет собой обычный текстовый файл, интерпретируемый браузером (рис. 12). Поскольку большинство браузеров не имеет встроенных средств поддержки VRML, для просмотра VRML-документов необходимо подключить вспомогательную программу – VRML-браузер (например Live3D, CosmoPlayer или Cortona). Существует немало VRML-редакторов, делающих процесс создания VRML-документов удобнее и быстрее, однако несложные модели можно создать и при помощи простого текстового редактора.

A screenshot of a Notepad window titled "1.wrl - Блокнот". The window contains VRML code. The code defines a background with a sky color of (1, 1, 1) and a group of children. The first child is a Shape with an Appearance containing a Material with a diffuse color of (1, 1, 1). The second child is an Extrusion with a crease angle of 1, a spine of [0 0 0, 0 0 3, 0 0 6, 0 0 9, 0 0 12], and an orientation of [0 0 1 0, 0 1 0 1.0, 0 1 0 2.0, 0 1 0 3.0, 0 1 0 4.0]. The code ends with three closing braces. The window has a menu bar with "Файл", "Правка", "Формат", "Вид", and "Справка". The text is in a monospaced font, and there is a vertical scrollbar on the right side.

```
#VRML V2.0 utf8
Background{ skyColor 1 1 1 }
Group { children[ Shape {
    appearance Appearance {
        material Material {
            diffuseColor 1 1 1
        }
    }
    geometry Extrusion {
        creaseAngle 1
        spine [ 0 0 0, 0 0 3, 0 0 6, 0 0 9, 0 0 12 ]
        orientation [ 0 0 1 0, 0 1 0 1.0, 0 1 0 2.0,
                    0 1 0 3.0, 0 1 0 4.0]
    }
}
]]}
```

Рис. 12. Пример текста на VRML

VRML обладает следующими возможностями:

- геометрические примитивы (куб, сфера, многоуровневые поверхности);
- работа с цветом в RGBA и индексном режимах;
- удаление невидимых линий и поверхностей;
- прозрачность;
- наложение текстуры;
- применение освещения;
- использование эффекта «тумана»;
- перевод трехмерных файлов форматов DXF, 3D Studio и IGES в

VRML;

- интегрирован в систему MATLAB Simulink.

VRML позволяет создавать трехмерные модели типовых механизмов в виде совокупности простых (параллелепипед, конус, цилиндр и эллипсоид) и сложных геометрических фигур (экструзия и др), которые имеют иерархические связи. В среду MATLAB интегрирован V-Realm Builder. V-Realm Builder – это инструментальное средство языка трехмерного моделирования VRML, включающее возможность работы с библиотеками файлов VRML. При создании сложных трехмерных объектов V-Realm Builder позволяет импортировать их из таких пакетов, как 3Dmax и Autodesk MAP 3D, которые специально для этого предназначены. Также в VRML существует возможность задания дерева наследования, поэтому изменение параметров (например, перемещение или масштабирование) одной из фигур может повлечь за собой изменение подчиненных фигур, что удобно использовать при управлении моделями. В библиотеку Simulink выходят блоки, связывающие среду с VRML-объектами. Эти возможности определили выбор средства визуализации.

Программные средства для оптимизации

Для оптимизации требуется применение таких программных средств, которые обеспечивают достижение требуемого результата за минимальное время.

В настоящее время существует множество работ, в которых подробно изложены различные методы оптимизации, поэтому не имеет смысла излагать подробно их суть. Приведем только наиболее важные методы и их иерархию.

Существуют следующие методы оптимизации:

- *Одномерная оптимизация*, включающая: а) прямые методы (перебора, поразрядного поиска, исключения отрезков, парабол); б) методы, использующие производные функции (средней точки, хорд, Ньютона, кубической аппроксимации); в) минимизации многомодальных функций (перебора, ломаных).

- *Безусловная минимизация функций многих переменных:* а)

прямые методы (минимизация по правильному симплексу, по деформированному симплексу, циклического покоординатного спуска, Хука–Дживса, случайного поиска, сопряженных направлений); б) использующие производные функции (градиентного спуска, наискорейшего спуска, сопряженных градиентов, Ньютона, квазиньютоновские).

- *Многомерная минимизация при наличии ограничений:* а)

линейное программирование (графический, симплекс-таблица, симплекс-метод, искусственного базиса, симплекс-метод при отсутствии условий неотрицательности переменных); б) нелинейное программирование (возможных направлений (случай линейных ограничений, случай нелинейных ограничений), градиентные (проекции градиента, условного градиента), последовательной безусловной минимизации (штрафных функций, барьерных функций)).

- *Решение дискретных задач оптимизации:* отсечений, ветвей и

границ, динамического программирования.

- *Оптимизация в бесконечномерных пространствах:* а)

приближенные методы безусловной минимизации функционалов в гильбертовом пространстве (градиентного спуска, наискорейшего спуска, сопряженных градиентов); б) приближенные методы минимизации функционалов при наличии ограничений (проекция градиента, штрафных функционалов, условного градиента).

Выбор алгоритма оптимизации зависит от таких факторов, как точность поиска экстремума, быстродействие, условия функционирования системы, и определяется решением конкретной задачи. К этой группе методов можно отнести методы безусловной минимизации функций многих переменных и многомерной минимизации при наличии ограничений (нелинейное программирование).

Для решения поставленной задачи возможно применение двух toolbox (приложений) SIMULINK и OPTIMIZATION пакета MATLAB.

В Optimization Toolbox используется три метода для решения нелинейных задач без наличия ограничений: *квази-Ньютоновский метод; метод Нелдера Мида; метод доверительных областей.*

Задачи нелинейной оптимизации при наличии ограничений представляют собой комбинацию неких нелинейных целевых функций и могут иметь линейные или нелинейные ограничения. В Optimization Toolbox используется два основных метода для решения подобных задач: *доверительных областей; последовательное применение активного набора задач квадратичного программирования.*

Многокритериальная оптимизация предназначена в основном для минимизации многоцелевых функций с учетом некоего набора ограничений. В Optimization Toolbox реализованы два типа постановки задачи многокритериальной оптимизации: задача достижения цели и задача минимакса.

Optimization Toolbox может решать нелинейные задачи методом наименьших квадратов. Для этого в Optimization Toolbox используется три метода: доверительных областей; Левенберга–Марквардта; Ньютона–Гаусса.

В задачах линейного программирования целевая функция представляет собой некое линейное выражение, на которое могут быть наложены ограничения в виде линейных равенств или неравенств. Для решения данного типа задач используются следующие два метода: симплексный; внутренней точки. Кроме известных и традиционных методов оптимизации могут применяться и современные методы, основанные на нейронных сетях и нечеткой логики.

MATLAB Optimization Toolbox

Основные типы задач оптимизации, решаемых системой MATLAB Optimization Toolbox, представлены в табл. 5.

В табл. 5 приняты следующие обозначения: a – скалярный аргумент; x, y – в общем случае векторные аргументы; $f(a), f(x)$ – скалярные функции; $F(x), c(x), seq(x), K(x, w)$ – векторные функции; A, Aeq, C, H – матрицы; $b, beq, d, f, w, goal$ – векторы; xL, xU – соответственно, нижняя и верхняя границы области изменения аргумента.

Но этапу поиска экстремума функционала качества, как правило, предшествует подготовительная работа, связанная с созданием удобного пользовательского интерфейса. Это необходимо делать, так как напрямую методы оптимизации, приведенные в табл. 5, использовать сложно.

В качестве примера рассмотрим оптимизацию коэффициентов ПИ регуляторов скорости секции бумагоделательной машины.

Оптимизация производилась по функционалу качества, имеющему вид

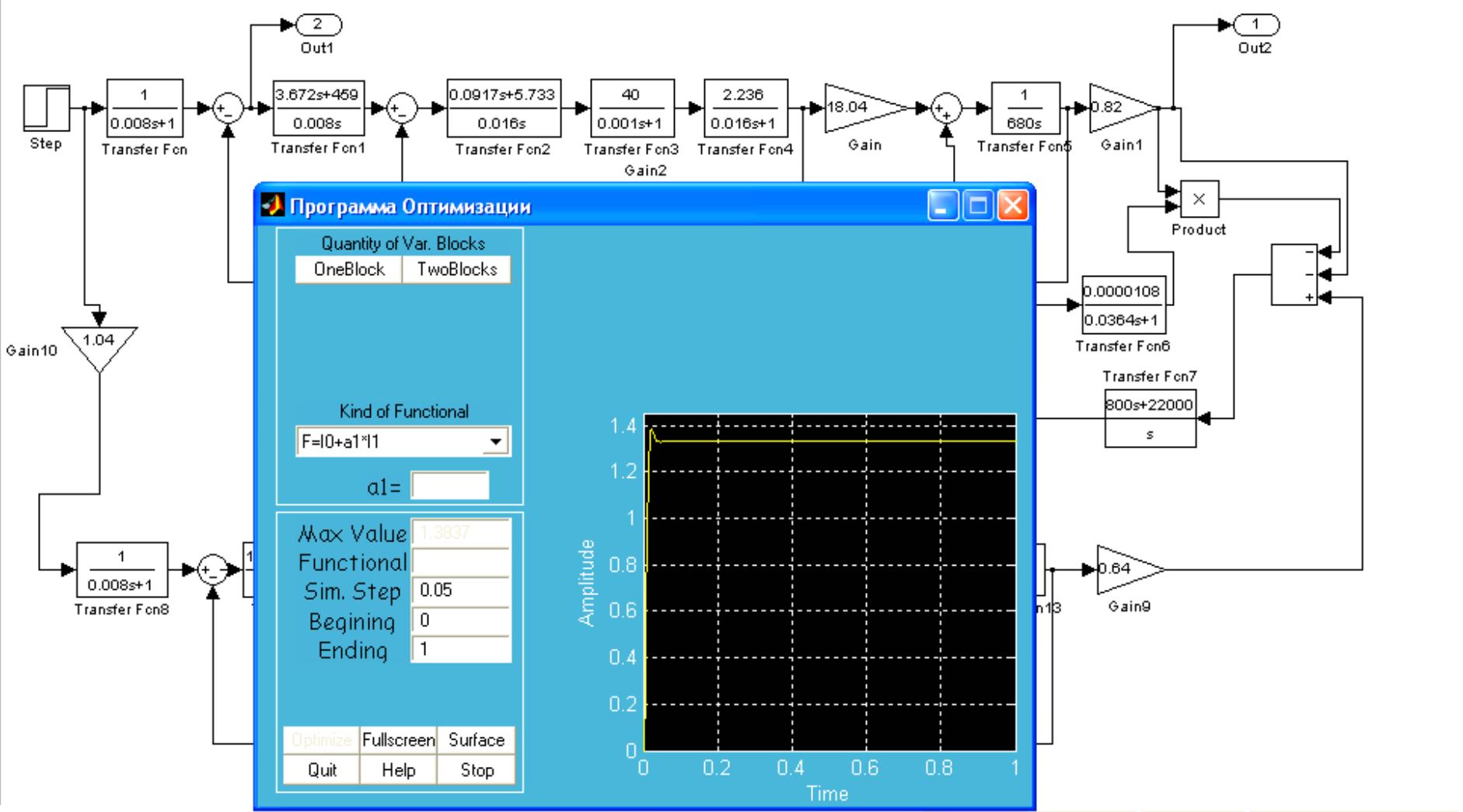
$$I = a_0 I_0 + a_1 I_1 + a_2 I_2, \text{ где } a_0, a_1, a_2 \text{ – весовые коэффициенты, } I_0 = \int e^2(t) dt, \\ I_1 = \int (de(t)/dt)^2 dt, I_2 = \int (d^2e(t)/dt^2)^2 dt \text{ (где } e(t) \text{ – ошибка по скорости).}$$

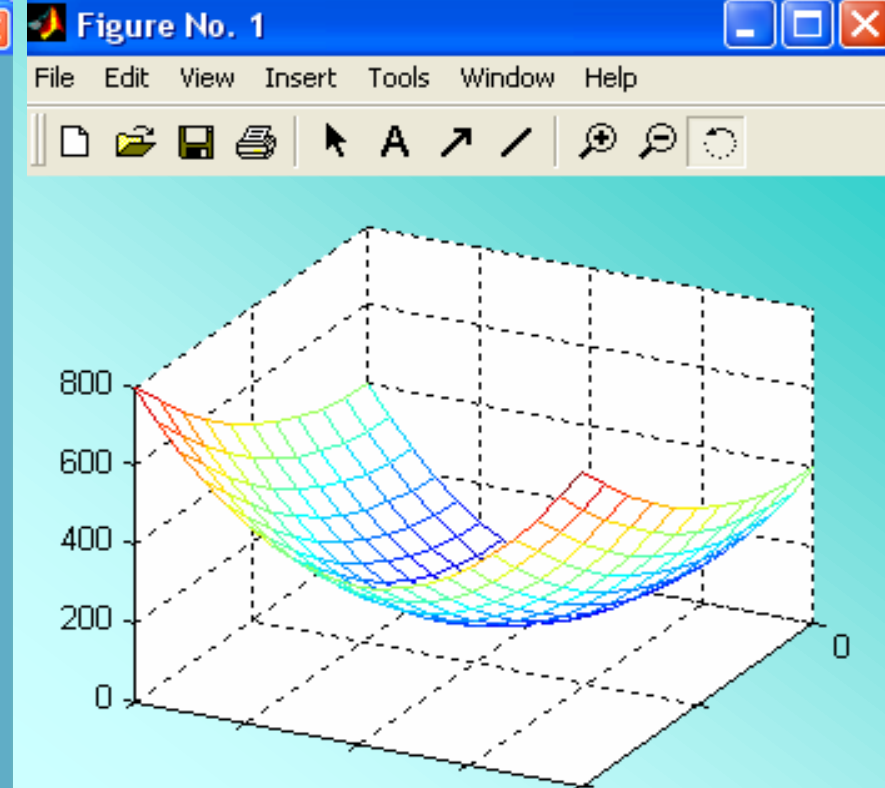
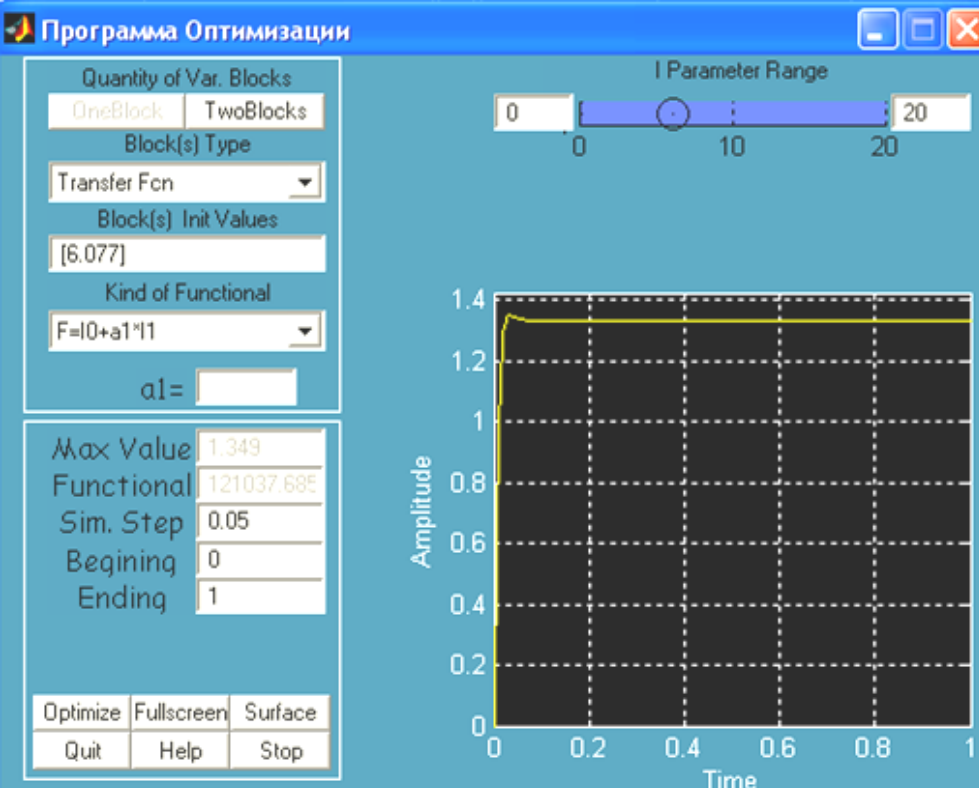
Тип задачи	Математическая запись	Используемая функция MATLAB
Скалярная (одномерная) минимизация	$\min f(a), a_1 < a < a_2$ a	<u>fminbnd</u>
Безусловная минимизация (без ограничений)	$\min f(x)$ x	<u>fminunc</u> , <u>fminsearch</u>
Линейное программирование	$\min f^T x$ при условиях x $A \cdot x \leq b, A_{eq} \cdot x = b_{eq}, x_L \leq x \leq x_U$	<u>linprog</u>
Квадратичное программирование	$\min 0.5x^T H x + f^T x$ при условиях x $A \cdot x \leq b, A_{eq} \cdot x = b_{eq}, x_L \leq x \leq x_U$	<u>quadprog</u>
Минимизация при наличии ограничений	$\min f(x)$ при условиях $c(x) \leq 0, c_{eq}(x) = 0, A \cdot x \leq b, A_{eq} \cdot x = b_{eq}, x_L \leq x \leq x_U$	<u>fmincon</u>
Достижение цели	$\min \gamma$ при условиях x, e $F(x) - w\gamma \leq goal, c(x) \leq 0,$ $c_{eq}(x) = 0, A \cdot x \leq b,$ $A_{eq} \cdot x = b_{eq}, x_L \leq x \leq x_U$	<u>fgoalattain</u>
Минимакс	$\min \max \{F_i(x)\}$ при условиях $x \{F_i\}$ $c(x) \leq 0, c_{eq}(x) = 0,$ $A \cdot x \leq b, A_{eq} \cdot x = b_{eq}, x_L \leq x \leq x_U$	<u>fminimax</u>

Исходные данные для предварительных расчетов, исследования и оптимизации:

– параметры секций и бумажного полотна: $L_{12} = 4.2$ м; $C_{12} = 2.2 \cdot 10^4$ Н/м; $r_1 = 0.82$ м; $r_2 = 0.64$ м; $i_{p1} = 4.4$ м; $i_{p2} = 5.4$ м; $J_1 = 680$ кг · м²; $J_2 = 540$ кг · м²; $B_{12} = 800$ Н/(м/с);

– параметры электроприводов и датчиков систем управления секциями бумагоделательной машины: $K_{п1} = 40$; $K_{п2} = 40$; $R_{яц1} = 0.43$ Ом; $R_{яц2} = 0.47$ Ом; $T_{яц1} = 0.016$ с; $T_{яц2} = 0.02$ с; $c_{д1} = 4.1$ Н · м/А; $c_{д2} = 4.3$ Н · м/А; $K_{дт1} = 0.03$ В/А; $K_{дт2} = 0.032$ В/А; $K_{дс1} = 0.14$ В/рад/с; $K_{дс2} = 0.17$ В/рад/с; $K_{сс2} = 0.03$; $T_{п1} = T_{п2} = 1 \cdot 10^{-3}$ с.





б

в

Рис. 13. Экранные копии интерфейса программы оптимизации: *а* – переходный процесс по скорости до оптимизации; *б* – переходный процесс по скорости после оптимизации; *в* – процесс поиска оптимального параметра регулятора

Контуры тока настроены на оптимум по модулю ($\tau_{рТ} = T_{яц}, T_{\mu i} = T_{п}$):

$$W_{рТ}(p) = \beta_{рТ} \frac{\tau_{рТ} p + 1}{\tau_{рТ} p}; \quad \beta_{рТ} = \frac{T_{яц}}{2T_{\mu i} \frac{K_{п} K_{дТ}}{R_{яц}}}$$

В результате расчетов получены следующие значения $\beta_{рТ1} = 2.87; \beta_{рТ2} = 3.672$.

Контуры скорости настроены на симметричный оптимум: $T_{\mu i} = T_{\mu} = T_{\pi} = 1 \cdot 10^{-3}$ с; $\tau_{рс} = 4 \cdot T_{\mu i} = 8 \cdot 10^{-3}$ с.

Общий вид ПИ-регулятора и расчет β -регулятора описываются выражениями

$$W_{рс}(p) = \beta_{рс} \frac{\tau_{рс} p + 1}{\tau_{рс} p} ; \beta_{рс} = \frac{K_{дт} J}{2 T_{\mu \omega} c_{дi}^2 K_{дс}}$$

В результате расчетов получены следующие значения $\beta_{рс1} = 458.94$, $\beta_{рс2} = 202.67$.

Результаты оптимизации представлены на рис. 13, а-в.

Перечислим некоторые другие программные продукты, используемые для моделирования систем. Полные сведения об их возможностях можно получить в Internet.

AnyLogic – графическая среда для моделирования сложных дискрет-но/непрерывных (гибридных) систем. Удобно, когда много движущихся объектов, они исчезают, появляются, видят друг друга, взаимодействуют и т.д.

DyMoLa – лаборатория (пакет) для динамического моделирования во временном домене механических, электронных, гидравлических, энергетических систем, в блочном представлении, с открытыми интерфейсами для интеграции и поддержкой языка Modelica.

DYNAST – ПО для расчета переходных процессов, символического и частотного анализа линеаризованных систем, описываемых системами дифференциальных и алгебраических уравнений, физическими процессами, и блок-схемами. Обслуживается в пакетном режиме удаленным реша-телем.

EASY5 (MSC. Software.) – специализированный продукт для моделирования широкого круга сложных технических систем и устройств на схемном уровне (рис. 2.21). Средствами этого программного продукта моделируются цифровые и аналоговые системы управления, гидроприводы, трансмиссии, двигатели внутреннего сгорания, пневматические, механические и электрические устройства, системы кондиционирования, управления аэродинамическими поверхностями, системы впрыска топлива и т.д.

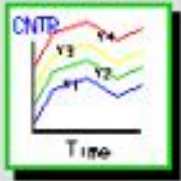
Смоделированные в EASY5 системы включаются в модели, созданные в Patran и/или Adams с целью проведения всесторонних исследований функционирования изделий на различных режимах работы и в различных условиях эксплуатации. При этом моделируется конструкция машин, механизмов, приборов и других технических объектов в совместной работе с системами управления, гидравлическими, пневматическими, электрическими системами и т.д. Для углубленного анализа и моделирования используется тесная интеграция с другими системами MSC. Software., а также с системами других компаний-разработчиков: MSC Nastran, Adams, Marc, Dytran, LS-Dyna, Flight Loads & Dynamics, Fatigue, Mvision, Simdesigner, Sofy, Explore, matlab/simulink, dads, ricardo wave, ANSYS, Abaqus и др.

Система EASY5 является одним из важнейших компонентов современных vpd-технологий. Программный пакет EASY5 включает множество готовых математических моделей, реализованных в виде отдельных функциональных блоков (сумматоров, делителей, фильтров, интеграторов, клапанов, двигателей, теплообменников, редукторов, муфт сцепления и др.). С помощью языков C или FORTRAN можно создавать собственные функциональные блоки.

Пользователь EASY5 – моделирование изделия происходит на уровне функциональных блоков, и оптимизирует параметры разрабатываемых систем изделия. EASY5 поддерживает возможность экспорта модели (например, модели системы управления) в модель Adams или, наоборот, импорта модели механизма из Adams в EASY5.

Кроме того, модели EASY5 можно встраивать в модели системы matlab/simulink. В состав EASY5 включен модуль matrix algebra tool, предназначенный для выполнения операций над матрицами (может использоваться для подготовки данных, анализа результатов и т.п.).

Interactive Sim



Supply Boundary Conditions

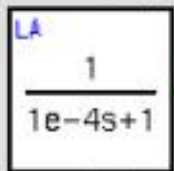
T = 50

LP P = 2500



Supply Tubing Losses

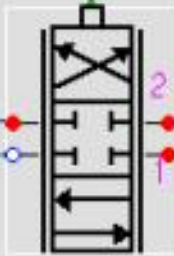
Solenoid Response



PI Controller

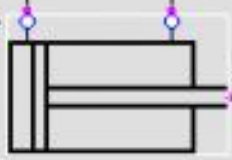


Servo Valve



VI

AP



Actuator

Piston_force

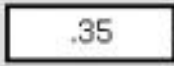


ADAMS Mechanical

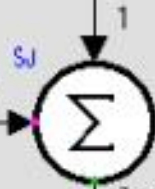
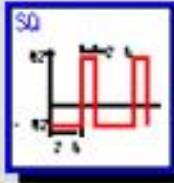
MSC Software



Aileron Null Position



Command



Sj

1

ail_act_len_command

1

1

ail_act_len_command

ail_act_len_command

ail_act_len_command

ail_act_len_command

ail_act_len_command

ail_act_len_command

ail_act_len_command

ail_act_len_command

Global Fluid Properties



Hydraulic Fluid

4

TN

Return reservoir

Пользователям EASY5 предложены библиотеки компонентов для многодисциплинарного анализа, включая специализированные наборы для аэрокосмической, автомобильной и других отраслей промышленности. Модуль library developer toolkit позволяет создавать собственные библиотеки компонентов.

На базе перечисленных возможностей системы EASY5 и в интеграции с другими системами MSC. Software. создаются полные функциональные модели самолетов, автомобилей, танков, экскаваторов, приборов и т.д., исследуется их работа при выполнении различных задач и в различных условиях эксплуатации.

Adams – программное средство для виртуального моделирования сложных машин и механизмов. С помощью Adams быстро создается полностью параметризованная модель изделия: она строится непосредственно в препроцессоре или импортируется из наиболее популярных САД-систем. Задав связи компонентов модели, приложив нагрузки, определив параметры кинематического воздействия и запустив расчет, можно получить данные, полностью идентичные результатам натуральных испытаний системы.

Пользователю доступны: выявление параметров изделия, определяющих его работоспособность и точность; проверка компонентов машины на столкновения, определение габаритных размеров пространства, необходимого для ее движущихся частей; определение уровня действующих нагрузок, необходимой мощности приводов; оптимизация параметров изделия.

Adams позволяет исследовать десятки, сотни и даже тысячи вариантов конструкции, выбирать лучший, совершенствовать и совершенствовать будущее изделие, затрачивая на это во много раз меньше времени и средств, чем при традиционном подходе.

Adams может использоваться для улучшения самых разных конструкций: от простых механических и электромеханических устройств до автомобилей и самолетов, железнодорожной техники и космических аппаратов.

Adams от других систем моделирования и проектирования отличаются:

- широкий набор видов кинематических связей, упругих и диссипативных звеньев, нагрузок, кинематических воздействий;

- совместимость с системами моделирования систем автоматического регулирования и управления Easy5 и MATLAB/Simulink, а также пользовательскими программами, что обеспечивает моделирование и исследование сложных гетерогенных динамических систем;

- легкость в изучении и использовании – моделирование соответствует основным шагам построения физического макета (построение виртуальных прототипов, выполнение набора тестов, совершенствование конструкции);

- полная параметризация виртуальных моделей – любые параметры прототипа могут быть связаны функциональной зависимостью, модификация какого-либо размера модели автоматически приводит к изменению ее конфигурации и т. п.;

- эффективные средства визуализации результатов моделирования, включая анимацию и построение графиков.

Наряду с развитием универсальных возможностей пакета, разработчиками Adams созданы проблемно-ориентированные модули, обеспечивающие точное и быстрое моделирование самых сложных объектов, например, устройств автоматического управления (ADAMS / Control) и др.

В программном пакете Adams предусмотрена возможность учета податливости компонентов исследуемой машины (механизма). Для этого упругие характеристики частей машины определяются в конечно-элементной системе (например, MD Nastran или MSC Nastran), а затем в специальном формате с помощью модуля Adams/Flex передаются в Adams и включаются в виртуальную модель. Использование такой модели позволяет не только учесть влияние податливости звеньев на динамику механизма, но также вычислить (с помощью модуля Adams/Durability) и отобразить на экране компьютера непосредственно в среде Adams общий уровень напряжений в упругом компоненте.

Результаты расчёта деформаций упругого компонента в Adams могут быть также «переданы» в конечно-элементную систему (MD Nastran или MSC Nastran) для точного определения уровня напряжений с учётом всех особенностей её конфигурации. В дальнейшем эти данные могут быть использованы и для анализа долговечности детали с помощью системы Fatigue. Такая технология обеспечивает существенное повышение точности моделирования современных, оптимизированных по массе высокочастотных механизмов. В программном пакете Adams обеспечивается возможность создания моделей, включающих контактное взаимодействие упругого тела с твёрдым телом, а также двух упругих тел.

Основой Adams являются системы дифференциальных уравнений, описывающих динамику исследуемого объекта.

МВТУ – программа «Моделирование в технических устройствах».

Является отечественной разработкой с классическим интерфейсом блочного моделирования. Исследование моделей возможно как во временном, так и в частотном доменах.

Model Vision Studium – новационный инструмент для визуального интерактивного моделирования во временном домене структурно-сложных гибридных систем с применением карт поведения и объектно-ориентированного языка описаний.

Modelica – универсальный объектно-ориентированный язык для моделирования физических систем. Поддерживается пакетами: Dymola, MathModelica (Mathematica), ALLAN, 20-sim, SMILE/M.

SamSim – моделирование линейных и нелинейных цепей САУ, построение временных, частотных характеристик, фазовых портретов и годографов. ПО начального уровня, поможет студентам визуализировать процессы в любой точке модели.

SIMPLORER Simulation Center – интегрированные инструменты для моделирования и анализа систем под контролем графа конечного автомата, описанных элементами электрических и блок-схем + собственный SML-язык для компилятора.

VTB Virtual Test Bed – моделирующий комплекс для визуальной (3D) симуляции технических систем различной физической природы – энергетика, электромеханика и др. Модели описываются на языках: ACSL, Modelica, Siber, SPICE, Simulink. + аппаратура ввода вывода.

КОПРАС – комплекс программ для анализа и синтеза автоматических систем (для ТАУ). Имеет большой набор решающих, функциональных элементов и сервисных функций. Динамические модели описываются структурными схемами.

Пакет **PDELab** – решение систем нелинейных дифференциальных уравнений в частных производных. Компьютерное моделирование в физике, химии, биологии и медицине.

APM WinMachine (НТЦ «АПМ») – CAD/CAE-система автоматизированного расчета и проектирования механического оборудования и конструкций в области машиностроения. Эта система в полном объеме учитывает требования государственных стандартов и правил, относящихся как к оформлению конструкторской документации, так и к расчетным алгоритмам.

Комплекс APM WinMachine позволяет решать следующие задачи:

- проектирование и анализ механического оборудования и его

элементов с использованием инженерных методик;

- анализ напряженно-деформированного состояния трехмерных

объектов любой сложности при произвольном закреплении, статическом

или динамическом нагружении;

- создание конструкторской документации;

- проектирование и расчет технологических процессов.

В состав APM WinMachine входят следующие APM:

1. *APM Graph* – двумерный графический редактор, который используется для оформления графической части конструкторской документации в различных областях техники, науки, в архитектуре и строительстве. Модуль имеет стандартный набор инструментов для оформления чертежей, а также инструменты параметризации объектов и расчетов размерных цепей.

2. *APM Studio* – модуль моделирования и импорта (поддерживается импорт из формата STEP) трехмерных поверхностных и твердотельных моделей с инструментами указания опор и приложения различных нагрузок и встроенным генератором разбиения на конечно-элементную сетку.

3. *Конечно-элементный анализ*. APM Structure3D. Модуль APM Structure3D предназначен для расчета напряженно-деформированного состояния стержневых, пластинчатых, оболочечных и твердотельных конструкций, а также их произвольных комбинаций.

Модуль APM Structure3D позволяет решать следующие задачи: 1) сопределение полей эквивалентных напряжений и их составляющих; 2) расчет линейных, угловых и результирующих перемещений; 3) определение внутренних усилий; 4) расчет устойчивости и формы потери устойчивости; 5) определение частот собственных колебаний и собственных форм; 6) расчет вынужденных колебаний и анимация колебательного процесса по заданной вынуждающей нагрузке, расчет на вибрацию оснований; 7) расчет температурных полей и термонапряжений; 8) расчет усталостной прочности; и др.

4. *APM Dynamics*. Модуль APM Dynamics предназначен для кинематического анализа рычажных механизмов. Результатами расчета модели являются траектории перемещения, графики скоростей, ускорений, силовых факторов в узловых точках и анимация работы механизма.

5. *APM Beam*. С помощью модуля APM Beam можно выполнить расчет балки и подбор ее наиболее подходящего поперечного сечения.

6. *Инженерный анализ. APM Bear*. APM Bear – модуль расчета и проектирования неидеальных подшипников качения. С помощью APM Bear можно рассчитать: перемещения (жесткость); долговечность; момент трения; наибольшие контактные напряжения; потери мощности; тепловыделение; силы, действующие на тела качения.

7. *APM Shaft*. APM Shaft – модуль проектирования валов и осей. Модуль APM Shaft позволяет выполнить весь цикл проектирования валов и осей, начиная от разработки конструкции и заканчивая статическим и динамическим анализом.

8. *APM Trans*. APM Trans – модуль проектирования и проверки механических передач вращения. С помощью APM Trans можно проектировать и проверять следующие типы передач: 1) цилиндрические с прямым зубом как внешнего, так и внутреннего зацепления; 2) цилиндрические с косым зубом внешнего зацепления; 3) шевронные; 4) конические с прямым и круговым зубьями; 5) червячные, с архимедовым, эвольвентным и конволютным червяком и глобоидные; и др.

9. *APM Drive*. APM Drive – модуль автоматизированного проектирования привода вращательного движения произвольной структуры.

10. *APM Spring*. APM Spring – модуль проектирования упругих элементов машин.

11. *APM Joint*. APM Joint – модуль проектирования соединений элементов машин. Модуль позволяет выполнить комплексный расчет и анализ соединений, которые наиболее часто используются в машиностроении.

12. *APM Plain*. APM Plain – модуль проектирования и расчета подшипников скольжения.

13. *APM Cam*. APM Cam – модуль проектирования кулачковых механизмов.

14. *APM Screw*. APM Screw – модуль проектирования неидеальных винтовых передач. С помощью APM Screw могут быть рассчитаны наиболее широко распространенные типы винтовых передач: винтовая передача скольжения, шариковая винтовая передача и планетарная винтовая (роликовая винтовая) передача. Модуль предназначен для комплексного анализа винтовых передач и расчета их основных параметров.

15. *Технологическая подготовка производства. APM Technology*. APM Technology – модуль проектирования технологических процессов. С помощью APM Technology можно: 1) разрабатывать технологические процессы, используя справочные данные, базы данных оборудования и технологической оснастки; 2) выполнять расчеты режимов обработки и норм времени; 3) оформлять технологическую документацию (маршрутные и операционные карты, карты технологических процессов, карты эскизов).

16. *APM Calculation Instrument.* APM Calculation Instrument – модуль, предназначенный для выполнения инженерных (конструкторских и технологических) расчетов. Вместе с модулем поставляется APM Technology Calculation – база стандартных технологических расчетов, выполняемых при проектировании технологических процессов. В модуле также предусмотрена возможность добавления собственных расчетных программ (алгоритмов расчетов).

17. *Базы данных.* В состав комплекса APM WinMachine входят следующие базы данных: а) APM Mechanical Data – база данных стандартных деталей и узлов, справочных данных по общему машиностроению; б) APM Technology Data – база стандартных информационных данных для проектирования технологических процессов; в) APM Material Data – модуль хранения и редактирования параметров материалов.

Работа с существующими базами данных проходит в модуле создания и редактирования баз данных – *APM Base*. APM Base содержит необходимый функционал для поиска, редактирования и расширения информационного массива вышеперечисленных баз данных или составления собственных.

MD Nastran (от MultiDiscipline Nastran) – система, объединяющая лучшие в своем классе системы компьютерного инженерного анализа, включая MSC Nastran, Marc, Dytran и LS-DYNA в одну полностью интегрированную суперсистему для проведения «многодисциплинарного» моделирования в масштабах предприятия.

MD Nastran дает возможность пользователям перейти от узкоспециализированных инструментов моделирования и анализа к интегрированному решению в масштабах предприятия. Использование единой расчетной модели для «многодисциплинарного» моделирования позволяет минимизировать время, обычно затрачиваемое на подготовку множества расчетных моделей, преобразование и передачу данных от одной расчетной модели в одной системе к другой модели в другой системе (решается проблема исключения вносимых при таких преобразованиях ошибок).

MD Nastran построен на базе MSC Nastran –системы конечно-элементного анализа. В дополнение к возможностям MSC Nastran, MD Nastran включает в себя решатель LS-DYNA в виде последовательности решения SOL 700, расширенный нелинейный статический и динамический анализ с помощью новой последовательности решения SOL 400, алгоритмы решения контактных задач в рамках линейного и нелинейного анализа и т.д.

Решение SOL 700 предназначено, преимущественно, для моделирования высокоскоростного нагружения различных изделий, включая их большие деформации и разрушение (например, соударения различных объектов, технологические процессы типа штамповки,ковки и т.п.) и т.д. В SOL 700 для решения нелинейных задач применяется явная схема интегрирования. Производительность SOL 700 особенно высока при решении задач на многопроцессорных вычислительных комплексах.

Вычислительная производительность MD Nastran оптимизирована для решения больших и сверхбольших задач и выполнения комплексных, связанных видов «многодисциплинарного» анализа.

MSC Nastran. MSC Nastran – продукт MSC. MSC Nastran обеспечивает полный набор расчетов, включая расчет напряженно-деформированного состояния, запасов прочности, собственных частот и форм колебаний, анализ устойчивости, исследование установившихся и неустановившихся динамических процессов, решение задач теплопередачи, акустических явлений, нелинейных статических и нелинейных переходных процессов, анализ сложного контактного взаимодействия, расчет критических частот и вибраций роторных машин, анализ частотных характеристик при воздействии случайных нагрузок и импульсного широкополосного воздействия, исследование аэроупругости на дозвуковых и сверхзвуковых скоростях.

В MSC Nastran моделируются практически все типы материалов, включая композитные и гиперупругие. Расширенные функции включают технологию суперэлементов (подконструкций), модальный синтез, анализ динамики сложных структур на основе метода Крейга-Бемптона, макроязык DMAP для создания пользовательских приложений.

Предусмотрена возможность передачи моделей упругих тел в Adams, которые будут использоваться в этом программном комплексе при построении расчётных моделей машин и механизмов.

MSC Nastran располагает аппаратом автоматической оптимизации параметров, формы и топологии конструкций. Возможности оптимизации позволяют использовать MSC Nastran для автоматической идентификации компьютерной расчетной модели и планирования экспериментов.

В составе MSC Nastran имеются специальные возможности моделирования динамики роторных машин, что обуславливает незаменимость этого программного продукта в отраслях, связанных с разработкой турбомашин.

MSC Nastran предоставляет возможности расчёта характеристик работы конструкций из композиционных материалов.

MSC Nastran используется для планирования экспериментов (определение мест расположения датчиков) и оценки полноты полученных экспериментальных данных. С помощью MSC Nastran решаются задачи моделирования систем управления, систем терморегулирования с учетом воздействия этих систем на конструкцию.

Тесная связь MSC Nastran с системой Patran обеспечивает полностью интегрированную среду для моделирования и анализа результатов. Все ведущие производители пре- и постпроцессоров, а также систем автоматизированного проектирования, предусматривают прямые интерфейсы с этой системой.

MSC Nastran работает на ПК, рабочих станциях, кластерах, суперкомпьютерах и предусматривает возможности векторной и параллельной обработки данных на вычислительных комплексах, которые поддерживают эти функции.

Patran – это интегрирующая среда систем проектирования, моделирования и анализа на основе современного графического пользовательского интерфейса.

Patran является интегрирующей средой и графической оболочкой для MD Nastran, MSC Nastran, Marc, Dytran, Sinda и других конечно-элементных систем анализа. Patran обеспечивает импорт геометрических моделей из CAD-систем, создание расчетных моделей, запуска их на расчет, графическое отображение и обработку полученных результатов.

Помимо доступа к стандартным геометрическим форматам, таким как Parasolid, ACIS, STEP, IGES и др., предоставляет прямой доступ к программным пакетам автоматизированного проектирования: CATIA v4 и v5, NX, Pro/ENGINEER.

Возможность использования данных о материалах обеспечивается через прямой интерфейс с системой Mvision.

При использовании Patran, как правило, именно CAD-геометрия становится основой конечно-элементной модели. Поэтому Patran включает в себя специальные методы и функции для контроля и автоматизированного исправления дефектов, импортированных из CAD-систем геометрических моделей, а также преобразования их в расчетные модели, что при современном уровне сложности и подробности компьютерных моделей существенно облегчает работу конструкторов и расчетчиков.

Patran включает в себя обширные функции создания и модификации геометрических моделей, в том числе твердотельное моделирование (включая Булевы операции), создание срединных поверхностей, автоматическое распознавание и параметризацию отверстий, скруглений и фасок.

Наличие инструментария для создания, редактирования и контроля качества конечно-элементных (КЭ) сеток позволяет в короткое время создавать оптимальные расчетные модели. Наряду со стандартными генераторами КЭ сеток в Patran присутствуют специализированные генераторы: SheetBodyMesher, AdvancedSurfaceMesher и др., позволяющие создавать высококачественные КЭ сетки на наборе поверхностей «плохого» (с точки зрения генерации КЭ сеток) качества.

В Patran предоставляется обширный набор возможностей для задания нагрузок, граничных условий, свойств материалов и элементов, параметров расчета, а также для визуализации, обработки и преобразования результатов счета. Кроме того, широкие возможности языка Patran Command Language (PCL) позволяют адаптировать все упомянутые функции к конкретным требованиям пользователя.

Функции «Группы», «Списки», «Суперэлементы», «Области» и др.

дают возможность создавать и обрабатывать полноразмерные подробные модели сложных изделий, таких как самолет, автомобиль и т.д., включающие миллионы конечных элементов.

Наряду с интерфейсами к CAD- и CAE-системам Patran имеет ряд специальных встроенных приложений:

THERMAL – анализ тепловых процессов и гидравлических сетей;

LAMINATE MODELER – подготовка моделей и обработка результатов расчетов для конструкций из композитных материалов;

ANALYSIS MANAGER – многофункциональная система, позволяющая управлять процессами выполнения заданий.

Autodesk Algor Simulation (Autodesk). Autodesk Algor Simulation обеспечивает решение широкого спектра задач механики деформируемого твердого тела, гидродинамики, электростатики, а также связанных задач.

Autodesk Algor Simulation – инструмент расчета деталей и сборок конструкций, который теперь тесно интегрирован с Moldflow 2011, что позволяет инженерам пользоваться результатами анализа и обширной библиотекой материалов последнего, чтобы принимать важные решения на более ранней стадии проекта.

Разнообразие типов конечных элементов, богатые возможности для построения конечно-элементных сеток и встроенная в Autodesk Algor Simulation библиотека моделей материалов позволяют создавать цифровые прототипы, эквивалентные реальным изделиям. С большинством CAD-программ поддерживается прямой ассоциативный обмен данными, что обеспечивает возможность интерактивно вносить изменения в проект (в CAD системе) без необходимости снова определять данные моделирования (в Autodesk Algor Simulation).

Autodesk Algor Simulation поддерживает прямой импорт геометрии из Autodesk Inventor, Autodesk Mechanical Desktop, IronCAD, KeyCreator, Pro/ENGINEER, Rhinoceros, Solid Edge, SolidWorks и импорт моделей в универсальных форматах: ACIS, STEP, IGES, STL.

Marc 2008 (MSC Software). Marc представляет собой универсальную конечно-элементную программу для проведения углубленного анализа высоконелинейного поведения конструкций и решения задач теплопередачи. Marc также используется для компьютерного моделирования технологических процессов прокатки, прессования, листового и объемного формования, производства шин, суперпластического формования и т.д. В Marc доступны также неструктурные типы анализа: 1) электростатический; 2) магнитостатический; 3) электромагнитный; 4) акустический; 5) гидродинамический анализ подшипников; 6) теплопередача и ряд других.

Дополняя возможности MSC Nastran и Dytran, программа позволяет проводить комплексный анализ ситуаций, когда элементы конструкции испытывают большие перемещения и повороты, а свойства материалов существенно нелинейны. Также возможен эффективный анализ сложного контактного взаимодействия конструкций.

Marc обладает широкими возможностями решения сложных нелинейных задач. Так, для изучения поведения резиновых уплотнений, строительных конструкций и некоторых других специальных объектов требуются особые модели материала. Контакт зубьев шестерен может быть корректно представлен лишь в том случае, если модель включает в себя соответствующие алгоритмы анализа контакта.

Множество специальных типов анализа в среде Marc поддерживается полным набором конечно-элементных формулировок. Программа предусматривает применение пользовательских подпрограмм, с помощью которых упрощается моделирование поведения изделия в особых ситуациях.

AFEA – интегрированное персональное рабочее место на основе комплекса «Patran – Marc». В единой среде графического интерфейса пре- и постпроцессора Patran пользователь может формировать задачи, осуществлять их расчет и обрабатывать результаты.

AFEA – комплекс на базе решателя Marc, ориентированный на решение физически и геометрически нелинейных задач механики, а также задач теплопередачи, включая связанные задачи теплопрочности. Функциональные возможности решателя Marc (полный набор моделей материала, автоматический трехмерный контакт, расчет больших пластических и упругих деформаций, циклическое нагружение, глобальное перестроение сетки, перенос решения из двумерного случая в трехмерный и т.д.) делают этот продукт высокоэффективным средством исследования сложного поведения различных конструкций.

SimDesigner – семейство встроенных решений для CAD-системы

CATIA V5, обеспечивающее моделирование динамики механизмов и быстропротекающих по времени существенно нелинейных процессов, расчет напряженно-деформированного и теплового состояний элементов конструкции.

SimDesigner создан на основе мощных систем, таких как MD Nastran, Marc, Adams, LS-DYNA, интегрированных с CATIA V5.

БАЗЫ ДАННЫХ МОДЕЛИРОВАНИЯ

Расширение возможностей моделирования различных классов систем **S** неразрывно связано с совершенствованием средств вычислительной техники и техники связи. Перспективным направлением является использование для целей моделирования иерархических многомашинных информационно-вычислительных систем и связанных с ними телекоммуникационными сетями удаленных персональных ЭВМ, работающих в режиме телеобработки.

При создании больших систем **S** их компоненты разрабатываются различными коллективами, которые используют средства моделирования при анализе и синтезе отдельных подсистем. При этом разработчикам необходим доступ как к коллективным, так и индивидуальным средствам моделирования, а также оперативный обмен результатами моделирования отдельных взаимодействующих подсистем. Таким образом, появляется необходимость в создании диалоговых систем моделирования коллективного пользования, для которых характерны следующие особенности: возможность одновременной работы многих пользователей, занятых разработкой одной системы **S**; доступ пользователей к программно-техническим ресурсам системы моделирования, включая распределенные банки данных и пакеты прикладных программ моделирования; обеспечение диалогового режима работы с различными вычислительными машинами и устройствами, включая цифровые и аналоговые вычислительные машины, установки физического моделирования, элементы реальных систем и т. п.; диспетчирование работ в автоматизированных системах моделирования (АСМ) и оказание различных услуг пользователям, включая обучение работе с диалоговой системой моделирования; использование сетевых технологий.

Рассмотрим основные моменты связанные с разработкой распределенной базы данных моделирования (РБДМ).

Ключевые аспекты разработки баз данных. Технология баз данных (БД) относится к числу основных компьютерных технологий и представляет собой совокупность методов и средств определения и манипулирования интегрированными в базу данными. Важной целью применения технологии БД является создание разделяемого между функционально связанными приложениями *информационного ресурса* с обеспечением независимости внешнего, логического представления БД от способов ее внутренней, физической организации в памяти компьютера. Для достижения поставленной цели технология БД использует соответствующий набор технологических инструментов.

Современное представление технологии БД определяется тем, что в основу этой технологии положено применение *реляционной модели данных (РМД)*, базирующейся на строгом аппарате реляционной алгебры и математической логики. Технологические операции определения и манипулирования БД выполняются с использованием систем реляционного исчисления. Реляционный подход в целом рассматривается в качестве идеологии создания баз данных и *баз знаний*. Такой подход является наиболее эффективным при решении многих задач моделирования сложных систем **S**.

С одной стороны, широкое применение РМД позволило разрешить одну из серьезнейших проблем достижения модельной однородности баз данных, создаваемых в средах различных *систем управления базами данных (СУБД)*, поскольку практически все современные СУБД используют модели, приводимые к реляционной. С другой стороны, опора на реляционную модель существенно ограничивает возможности определения данных в БД и, тем самым, предопределяет соответствующие границы применения всей технологии БД.

Такой подход, безусловно, оправдан при проектировании БД в тех случаях, когда администратор БД владеет схемой соответствия множества данных в реляционной модели с множеством данных о реальном мире. В тоже время, интеграционные тенденции, характерные для современного этапа развития компьютеризированных технологий (в том числе и в моделировании систем), ставят на повестку дня проблему построения *интегрированных распределенных баз данных* (ИРБД), для которых обеспечение схемной однородности на основе РМД в силу целого ряда причин оказывается недостаточно. Это не означает требования революционных изменений принципов реляционного подхода при проектировании БД в условиях построения ИРБД. Это означает только то, что при определении и построении ИРБД реляционный подход должен применяться с учетом классической схемы проектирования баз данных, согласно которой необходимо знать, каким образом был выполнен полный цикл этапов моделирования заданной предметной области в виде реляционных схем интегрируемых БД. Очевидно, что расширение границ применения реляционного подхода, при этом, позволит проектировать новые БД уже с учетом возможности их будущей интеграции и ИРБД. Характерным примером реализации расширения реляционного подхода для разработки распределенных приложений на основе интегрированных реляционных баз данных стало создание методов и средств CASE-технологий.

С учетом сказанного, все основные понятия и определения технологии баз данных будут формулироваться именно с ориентацией на реализацию расширенного реляционного подхода для достижения цели методологического определения ИРБД. Полная технологическая схема определения и манипулирования интегрированными в базу данными представлена на рис. 4.

База данных. База данных составляет ключевое понятие технологии БД и стержневой объект управления в системах баз данных. Определение базы данных в качестве разделяемого информационного ресурса компьютеризированных технологий требует уточнения самих понятий *данные* и *информация*. Иногда база данных трактуется в качестве "подобия электронной картотеки", "хранилища для некоторого набора занесенных в компьютер файлов данных", подразумевая под термином файл "абстрактный набор данных, не обязательно совпадающий с физическим дисковым файлом". Очевидно, что при таком взгляде данные и информация рассматриваются в качестве синонимов. Как следствие, истинным становится утверждение о том, что в этом случае любые данные, извлеченные любым способом из БД, являются информацией.



Рис. 4. Полная технологическая схема реализации БДМ

Классическое определение "база данных это - данные и связи между ними" представляется более точным и уместным с учетом высказанных выше соображений. Тогда данные, извлеченные из БД на основе установленных связей, являются информацией. В противном случае извлеченные из БД данные требуют *интерпретации*. Безусловно, хранящиеся в БД фрагменты связанных данных также соответствуют понятию информации. Вне связей данные являются информацией только в том случае, если они типизированы, или классифицированы, и известна примененная классификационная схема. С учетом применения реляционного подхода связи между данными можно разделить на связи совместимости (совместность атрибутивных значений табличного определения прикладного объекта) и связи соответствия (совместность атрибутивных значений межтабличного определения прикладного объекта).

Таким образом, в дальнейшем под термином *база данных* будем понимать совокупность *связанных данных*, с одной стороны, являющихся информацией, и с другой стороны, составляющих основу для получения информации, как произвольных комбинаций хранящихся связанных данных. Тогда *данные БД* и *информация* по определению оказываются синонимами.

Разновидности систем баз данных. В зависимости от способов определения и манипулирования связанными данными системы БД можно разделить на следующие основные разновидности.

Системы с файловыми базами данных в качестве БД используют простые структурированные файлы в форматах *dbf*, *bd* и др., а все информативные связи определяются и обрабатываются в приложениях, использующих такие БД. Эффективность организации структурированных файлов обычно повышается путем построения индексов и других систем указателей, что, вообще говоря, характерно при создании картотек. Индексируются, как правило, ключевые поля структур с целью ускорения доступа (за счет сортировки индексов), обеспечения уникальности значений полей, запрета на существование неопределенных значений и т. п. К числу наиболее существенных недостатков систем файловых БД (только в смысле их использования) можно отнести полную зависимость от приложений. Доступ к информации файловых БД возможен только посредством содержащего программные связи приложения. Очевидно, что как разделяемый информационный ресурс файловые БД могут существовать только в симбиозе с обеспечивающими связывание данных приложениями. Программная реализация связей на *SQL*-серверах или в виде *DLL* -библиотек естественно придает файловым БД совершенно новое качество реально разделяемого информационного ресурса.

К противоположной разновидности относятся такие системы БД, в которых все связи между данными определены как данные и хранятся в БД. Такие системы можно назвать системами с предметными базами данных.

Суть названия предметная БД заключается в достижении полной независимости предметных баз данных от приложений. Предметные БД являются полноценными, самостоятельными ресурсами компьютеризированных технологий, что составляет главное преимущество их применения. В тоже время, полное определение всех связей между данными порождает существенную сложность проектирования таких БД.

Промежуточные варианты организации баз данных, при которых связи распределяются между приложениями и БД, определяют разновидность систем с прикладными базами данных. Суть названия отражает слабо или сильно выраженную ориентацию организации прикладной БД на потребности использующих ее приложений. Очевидно, что как компромиссный вариант, прикладные БД могут выступать в качестве оптимизируемого информационного ресурса компьютеризированных технологий. Прикладные базы данных находят широкое применение при моделировании на ЭВМ конкретных систем **S**.

Предметная область. Первичным источником связанных данных при проектировании баз данных любых разновидностей является соответствующая *предметная область*, которая будет рассматриваться как совокупность знаний и данных об *объектах и процессах*, подлежащих проектированию и хранению в БД. В данном случае в качестве предметной области рассматривается проблема имитационного моделирования сложных систем на ЭВМ.

Знания о предметной области, посвященном методике разработки и машинной реализации моделей систем **S**, могут быть получены из разнообразных источников, таких, как:

- фундаментальные законы и устойчивые закономерности, теоретические знания прикладных наук о процессах в моделируемой системе **S**;
- эвристические знания в виде принятых правил, соглашений и обозначений, характерных для систем **S**;
- экспертные знания и экспертные оценки специалистов в области моделирования конкретных систем **S**;
- существующие базы данных, компьютеризированные системы, технологии и проекты.

Перечисленные источники знаний о предметной области "моделирование систем" не являются источниками собственно данных, а определяют методологические принципы выделения прикладных объектов и процессов, а также методы приобретения и связывания данных о выделенных объектах и процессах. Источники знаний о предметной области определяют выбор *классификационной схемы* формализованного описания объектов и процессов, которая в процессе проектирования БД отображается в результирующей реляционной схеме.

Таким образом, процесс проектирования базы данных по полной технологической схеме есть процесс пошагового отображения исходной классификационной схемы предметной области в реляционную схему реализации базы данных. Выбор той или иной методологической основы процесса проектирования БД определяет вид реализованных в БД связей между данными и характеризует *интеграционные возможности* построенной БД.

Важным методологическим признаком проектируемой БД и ее интеграционных возможностей является естественная или искусственная природа выделения прикладных объектов и процессов, связанные данные о которых будут храниться в проектируемой БД. Особое значение это обстоятельство приобретает при применении *объектно-ориентированного подхода* к проектированию баз данных и построению ИРБД.

В практике моделирования сложных систем **S** при проектировании локальных баз данных (ЛБД) применение классификационных схем зачастую осуществляется интуитивно, с твердым убеждением в том, что выбранный способ определения данных БД естественен для решения данной задачи. На самом деле, проектировщик ЛБД просто принимает одну из существующих классификационных схем (например, традиционную схему построения базы данных "система **S** - концептуальная модель M_k - машинная модель M_M " из литературы). В тоже время, отсутствие знаний о правилах формирования и связывания атрибутивных значений даже в такой БД может привести к искажению и ошибкам в трактовке извлекаемых из БД данных как информации.