

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ  
Государственное образовательное учреждение высшего профессионального образования  
«ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

---

**А.В. Воронин**

**МОДЕЛИРОВАНИЕ  
МЕХАТРОННЫХ СИСТЕМ**

Учебное пособие

Издательство  
Томского политехнического университета  
2008

УДК 621.865.8(075.8)

ББК 32.816я73

В75

**Воронин А.В.**

В75 Моделирование мехатронных систем: учебное пособие. – Томск: Изд-во Томского политехнического университета, 2008. – 127 с.

ISBN 5-98298-276-8

Изложенный в пособии материал ориентирован на формирование у студентов знаний о современном состоянии и перспективах развития средств и методов моделирования мехатронных систем; умения ставить задачу моделирования, выбирать структуру, а также алгоритмическую и программную реализацию имитационной модели сложного динамического объекта управления; умения получать математические модели объектов с элементами различной физической природы и оценивать их адекватность; умения ориентироваться в средствах и методах моделирования, выбирать и настраивать современную среду автоматизированного моделирования.

Пособие разработано в рамках реализации Инновационной образовательной программы ТПУ по направлению «Информационно-коммуникационные системы и технологии» и предназначено для студентов, обучающихся по программам подготовки бакалавров и магистров по направлению 220200 – «Автоматизация и управление» и дипломированных специалистов (инженеров) по специальностям 220301 – «Автоматизация технологических процессов и производств», 220401 – «Мехатроника», а также инженеров, аспирантов и научных сотрудников, специализирующихся в разработке и исследовании систем управления техническими объектами.

УДК 621.865.8 (075.8)

ББК 32.816я73

Рекомендовано к печати Редакционно-издательским советом  
Томского политехнического университета

*Рецензент*

Кандидат технических наук,  
доцент кафедры прикладной математики ТГУ

*Г.Н. Решетникова*

ISBN 5-98298-276-8

© Воронин А.В., 2008

© Томский политехнический университет, 2008

© Оформление. Издательство Томского  
политехнического университета, 2008

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	8
ГЛАВА 1. ОПРЕДЕЛЕНИЕ И НАЗНАЧЕНИЕ МОДЕЛИРОВАНИЯ. ....	10
1.1 Общие определения.....	10
1.2. Классификация методов моделирования по типу модели .....	13
1.3. Математическое моделирование и математические модели.....	15
1.4. Классификация методов математического моделирования применительно к этапу построения математической модели .....	17
1.5. Классификация методов математического моделирования применительно к этапу исследования математической модели.....	20
1.6. Характеристики математической модели .....	22
Глава 2. АНАЛИТИЧЕСКОЕ МОДЕЛИРОВАНИЕ МЕХАТРОННЫХ СИСТЕМ. МЕТОД ГРАФОВ СВЯЗЕЙ.....	26
2.1. Компонентное моделирование .....	26
2.2. Основные определения графов связей .....	29
2.2.1. Переменные связей .....	29
2.2.2. Интерпретация переменных связей.....	30
2.2.3. Типовые элементы графов связей.....	30
2.2.4. Физическая интерпретация основных элементов графов связей....	34
2.3. Моделирование электрических систем на графах связей .....	34
2.4. Эквивалентные преобразования графов связей.....	36
2.5. Моделирование механических систем на графах связей .....	39
2.6. Моделирование электромеханических систем .....	45
2.7. Получение математической модели графов связей в форме системы уравнений .....	47
2.8. Причинные отношения в графах связей.....	48
2.9. Построение операторно-структурных схем по графам связей .....	52
2.10. Применение правила циклов к графам связей.....	54
2.11. Общие принципы графического представления мехатронных систем в пакетах автоматизированного моделирования.....	59
Глава 3. ИССЛЕДОВАНИЕ МЕХАТРОННЫХ СИСТЕМ ВО ВРЕМЕННОЙ ОБЛАСТИ .....	67
3.1. Механизмы продвижения модельного времени .....	67
3.2. Алгоритмы численного моделирования нелинейных динамических систем .....	70
3.2.1. Свойства методов численного интегрирования .....	71
3.2.2. Методы явные и неявные.....	73
3.2.3. Выбор между явными и неявными методами в процедурах моделирования мехатронных систем .....	77
3.2.4. Многошаговые методы интегрирования .....	78
3.2.5. Порядок метода интегрирования .....	80
3.2.6. Процедуры численного моделирования с автоматическим выбором шага.....	82
3.2.7. Особенности численного интегрирования мехатронных систем.....	84

3.3. Моделирование гибридных (событийно-управляемых) мехатронных систем .....	86
Глава 4. АВТОМАТИЗИРОВАННОЕ МОДЕЛИРОВАНИЕ ТЕХНИЧЕСКИХ ОБЪЕКТОВ .....	91
4.1. Особенности современных систем автоматизированного моделирования .....	92
4.2. Иерархическое проектирование и многоуровневое моделирование мехатронных систем.....	93
4.3. Архитектура программ автоматизированного моделирования .....	95
4.3.1. Графический интерфейс программ математического моделирования динамических систем .....	96
4.3.2. Язык описания объекта, транслятор, система управления базами данных, монитор .....	96
4.3.3. Инструментальные средства моделирования (математическое ядро).....	98
4.4. Методы построения моделирующих программ.....	100
4.4.1. Структурное моделирование .....	102
4.4.2. Решатели для структурного и физического мультидоменного моделирования .....	104
Глава 5. ПАКЕТЫ ВИЗУАЛЬНОГО МОДЕЛИРОВАНИЯ МЕХАТРОННЫХ СИСТЕМ.....	106
5.1. Классификация пакетов моделирования технических систем.....	106
5.2. Пакеты структурного моделирования .....	109
5.2.1. Пакет MATLAB/Simulink .....	110
5.2.2. Пакет VisSim .....	112
5.2.3. Пакет MBTU.....	113
5.3. Пакеты физического мультидоменного моделирования .....	115
5.3.1. Пакет Modelica/Dymola .....	116
5.3.2. Пакет 20-sim .....	117
5.4. Пакеты среды MATLAB для моделирования мехатронных систем ..	118
5.4.1. Принципы моделирования механических систем в пакете SimMechanics .....	118
5.4.2. Пакет моделирования электрических систем SimPowerSystems	120
5.4.3. Пакет моделирования гибридных систем StateFlow .....	121
СПИСОК ЛИТЕРАТУРЫ .....	124

## СПИСОК СОКРАЩЕНИЙ

АВМ – аналоговые вычислительные машины;  
ДУЧП – дифференциальные уравнения в частных производных;  
ММ – математическое моделирование;  
ОДУ – обыкновенные дифференциальные уравнения;  
САМ – система автоматизированного моделирования;  
САПР – система автоматизированного проектирования;  
СЛАУ – система линейных алгебраических уравнений;  
СУБД – система управления базами данных;  
ТАУ – теория автоматического управления;  
ЦВМ – цифровые вычислительные машины;  
ЭМП – электромеханический преобразователь.

## ПРЕДИСЛОВИЕ

Данное учебное пособие предназначено для студентов, обучающихся по направлению «Автоматизация и управление», и соответствует образовательному стандарту Томского политехнического университета по специальности 220401 – Мехатроника.

Изложенный материал ориентирован на формирование у студентов знаний о современном состоянии и перспективах развития средств и методов моделирования мехатронных систем; умения ставить задачу моделирования, выбирать структуру, а также алгоритмическую и программную реализацию имитационной модели сложного динамического объекта управления; умения получать математические модели объектов с элементами различной физической природы и оценивать их адекватность; умения ориентироваться в средствах и методах моделирования, выбирать и настраивать современную среду автоматизированного моделирования.

Настоящее учебное пособие состоит из пяти глав и начинается с главы, в которой вводятся понятия модели и моделирования, перечисляются задачи, решаемые средствами моделирования, показывается роль и место моделирования в общей процедуре проектирования технических систем. Рассматриваются классификации основных методов моделирования, методов получения и исследования математических моделей с помощью ЭВМ.

Во второй главе рассматривается формализация процесса построения математических моделей мехатронных систем с использованием метода графов связей, относящегося к группе топологических методов, т. е. методов, использующих графическое представление исследуемого объекта. Данный метод основан на использовании так называемого «энергетического» подхода и позволяет на единой методологической базе строить математические модели объектов, содержащих элементы различной физической природы. Знакомство с этим методом позволит студентам лучше понять принципы построения современных систем автоматизированного моделирования.

В третьей главе изложены некоторые сведения о принципах управления модельным временем и о современных численных методах исследования динамических систем. Данные сведения необходимы студентам, в т. ч. и для того, чтобы уметь грамотно пользоваться настрой-

ками, которые допускают современные инструменты автоматизированного моделирования.

Четвертая глава посвящена современным компьютерным инструментам автоматизированного моделирования. Изложены основные сведения об архитектуре и принципах построения систем автоматизированного моделирования, описаны состав и функциональное назначение основных составляющих современных пакетов моделирования, а также принципы управления процессом моделирования.

Пятая глава содержит обзор некоторых наиболее популярных и перспективных инструментов моделирования мехатронных систем. Кратко описаны возможности пакетов структурного (MATLAB/Simulink, VisSim, MBTU) и физического мультидоменного моделирования (Modelica/Dymola, 20-sim). Более детально рассмотрены расширения среды MATLAB, в частности пакет моделирования механических систем SimMechanics, электрических систем SimPower, а также пакет событийного моделирования StateFlow. Показано, что комплексирование возможностей этих пакетов на базе пакета Simulink позволяет получить мощный инструмент исследования сложных мехатронных систем.

Курс базируется на знаниях, полученных студентами в рамках дисциплин «Информатика», «Теоретические основы электротехники», «Теоретическая механика», «Теория автоматического управления». Навыки моделирования технических систем необходимы для изучения ряда последующих курсов и в особенности при выполнении курсовых и выпускных квалификационных работ, связанных с исследованием и проектированием различных технических устройств и систем.

## ВВЕДЕНИЕ

Моделирование является важнейшим и неотъемлемым этапом процедуры проектирования современных мехатронных устройств и систем. В настоящее время сложно представить себе специалиста, не способного проверить моделированием обоснованность принятых технических решений. Соответственно, постоянно возрастает роль моделирования в учебном процессе. При подготовке курсовых и выпускных квалификационных работ по дисциплинам, связанным с разработкой и исследованием технических устройств, этап создания и использования соответствующих математических моделей является одним из основных.

Развитие микроэлектроники и микропроцессорной техники создало условия для нового качественного скачка в функциональных возможностях технических систем, связанных с движением механических устройств, что привело к возникновению новой науки – мехатроники. Существует большое число формальных определений мехатроники и мехатронных систем, по сути, мало отличающихся друг от друга. В Государственном образовательном стандарте РФ по направлению 652000 – Мехатроника и робототехника (2000 г.) оно звучит следующим образом: мехатроника – это область науки и техники, основанная на синергетическом объединении узлов точной механики с электронными, электротехническими и компьютерными компонентами, обеспечивающими проектирование и производство качественно новых модулей, систем, машин и систем с интеллектуальным управлением их функциональными движениями.

Это позволяет определить особенности математических моделей мехатронных устройств, понимая под ними объекты, для исследования и проектирования которых используются математические модели, отражающие взаимное влияние протекающих в объекте процессов различной физической природы – механических, электрических, информационных и т. п. [22].

В связи с тем, что мехатронная система – это синергетическое объединение механической, электрической и компьютерной частей, средства моделирования должны допускать совместное моделирование этих частей на единой методологической основе, давая возможность строить и исследовать многоаспектные модели [30].

Реализовать это возможно двумя способами. Во-первых, можно перейти к единой системе дифференциальных уравнений, как это обычно

делается в теории автоматического управления (ТАУ). В этом случае все физические особенности отдельных частей системы будут потеряны. Вариант такого подхода – структурное моделирование, где все переменные являются скалярными сигналами и их можно соединять (как в операторно-структурной схеме).

Недостаток подхода – большой объем предварительных преобразований (в случае системы уравнений) или получение схемы, мало напоминающей реальную систему.

Другой вариант – использование систем моделирования, которые способны на единой методологической основе моделировать механические, электрические и информационные компоненты, т. е. объединять их в единую схему, сохраняя при этом привычные для специалистов в предметных областях способы задания исходной информации.

Описываемый подход отличается от принятого в ТАУ тем, что в математических моделях используются не абстрактные сигналы, а величины, непосредственно характеризующие физическое состояние объекта (токи, потенциалы, давления, силы и т. п.) и связанные компонентными уравнениями. Именно этот подход становится в последнее время доминирующим, и именно он является основным объектом рассмотрения в данном пособии.

При этом основу моделирования мехатронных систем составляет моделирование механических конструкций. Как правило, именно моделирование механических конструкций является наиболее сложным и трудоемким. Компоненты механических конструкций описываются наиболее сложными математическими моделями. Они наиболее многомерны и предъявляют жесткие требования к инструментальным средствам моделирования.

# Глава 1

## ОПРЕДЕЛЕНИЕ И НАЗНАЧЕНИЕ МОДЕЛИРОВАНИЯ

### 1.1. Общие определения

С моделированием любой человек сталкивается постоянно, обычно не осознавая этого. Действия пешехода при переходе улицы базируются на построении некоторой модели дорожной обстановки и на прогнозе ее развития. От того, насколько верно пешеход воспринимает окружающую действительность, очень часто зависит не только его благополучие и здоровье, но и сама жизнь.

В процессе профессиональной деятельности, если она связана с проектированием и эксплуатацией современных технических объектов и систем, исследователь постоянно вынужден иметь дело с построением и исследованием моделей этих объектов. В настоящее время моделирование представляет собой основной научный инструмент, применяемый как в чисто теоретических, так и в практических целях.

Создание нового технического объекта – сложный и длительный процесс, в котором стадия проектирования имеет решающее значение в осуществлении замысла и достижении высокого технического уровня. Моделирование, в свою очередь, является одним из важнейших этапов проектирования любого технического объекта, позволяя заменить или значительно сократить этапы наладки и натурных испытаний. Роль моделирования особенно высока, когда натурные испытания слишком дороги или опасны, как, например, при работе с космическими аппаратами, химическими и ядерными реакторами и другими объектами.

Термин «моделирование» весьма многогранен и разными людьми воспринимается по-разному. Применительно к техническим (в т. ч. механическим) системам под **моделированием** будем понимать процесс, состоящий в выявлении основных свойств исследуемого объекта, построении моделей и их применении для прогнозирования поведения объекта.

Таким образом, моделирование включает в себя отображение проблемы из реального мира в мир моделей (процесс абстракции), анализ и оптимизацию модели, нахождение решения и отображение решения обратно в реальный мир.

Следует отметить, что в иностранной литературе процесс, определенный выше как моделирование, соответствует двум терминам:

- «modeling» – относится, прежде всего, к процессу построения моделей объектов и систем;
- «simulation» – обозначает проведение компьютерного эксперимента с моделью (обычно численного), с визуализацией результатов этого эксперимента.

Моделирование как процесс имеет дело с моделями. **Модель** – создаваемое человеком подобие изучаемых объектов: макеты, изображения, схемы, словесные описания, математические формулы, карты и т. д.

Более строго, модель можно определить как физическую или математическую конструкцию, определенным образом отражающую объект и служащую для его изучения.

Модель является заменителем реального объекта, обладающим, по крайней мере, двумя следующими свойствами:

- она отражает те свойства объекта, которые существенны для данного исследования;
- она всегда проще объекта.

Теория замещения одних объектов (оригиналов) другими объектами (моделями) и исследование свойств объектов на их моделях называется **теорией моделирования**.

Тот факт, что модель отражает лишь важнейшие для данного исследования свойства объекта, дает возможность соотнести одну и ту же модель с целым рядом конкретных объектов, что позволяет по установленным свойствам одного объекта судить о свойствах больших групп объектов, подобных первому объекту.

Полученные модели можно использовать для следующих целей.

1. **Познание (изучение объекта)**. Одной из особенностей хорошо построенной модели является то, что она несет в себе больше информации, чем в нее закладывалось при создании. Особенно это относится к моделям природных объектов, получаемых в результате естественно-научных исследований. Скрытая неосознанная информация проникает в модель объективно, помимо воли исследователя. Это позволяет на основе изучения модели получать новые сведения об объекте, т. е. изучать объект, изучая его модель. Свойство модели служить источником познания называют **потенциальностью**. Естественно, что разные модели в разной степени «богаты» такой дополнительной информацией.

Процесс познания в фундаментальных дисциплинах (физика, химия и др.) развивается по схеме «явление – модель – явление». Открытие и изучение нового явления приводит к построению его модели, ко-

торая, в свою очередь, позволяет предсказать новые явления. Классическим примером такой цепочки явилось открытие на основе Ньютоновского закона всемирного тяготения планеты Нептун. И в настоящее время многие объекты в космологии появляются сначала на уровне теоретических предсказаний и лишь затем подтверждаются наблюдениями.

**2. Предсказание.** Правильно построенная модель позволяет предсказывать поведение исследуемого объекта при тех или иных внешних воздействиях. Это свойство является ключевым в процессе замены объекта его моделью. Задача предсказания актуальна в тех случаях, когда эксперименты с реальным объектом не могут быть проведены по причинам повышенной опасности, чрезмерной длительности или невозможности воспроизведения внешних условий. Результаты предсказания могут использоваться для формирования управляющих воздействий на объект, а также для поиска оптимальных режимов работы этого объекта.

**3. Обучение.** Использование реального объекта для обучения часто связано с рисками как для объекта, так и для обучаемого. Заменяя реальный объект, модели могут быть использованы в качестве имитаторов при создании различных тренажеров, на которых можно не только получить первоначальные навыки управления, но и испробовать такие приемы, которые в иной ситуации отработать невозможно. Для обучения могут использоваться как физические, так и компьютерные модели, а в сложных тренажерах – сочетание тех и других. Примером могут служить тренажеры для подготовки пилотов самолетов. Кроме сложной физической системы, моделирующей кабину самолета и обладающей способностью создавать ощущение полета, имитируя движения по крену или тангажу, тренажер снабжен мощной компьютерной моделью, формирующей на экранах кабины видеокартинки и способной адекватно менять их в ответ на действия экипажа.

**4. Отработка новых конструкторских решений.** С технической точки зрения возможность использования моделей для проверки и отработки технических решений является самой важной функцией моделирования. Отсутствие реального объекта делает эту функцию безальтернативной, позволяя существенно сократить время разработки нового изделия за счет экономии на его натуральных испытаниях. Далее будет показано, что использование модели, для которой еще нет реального объекта, вносит существенные особенности в процесс ее построения и отладки.

В зависимости от типов моделей, способов их формирования и методов анализа их поведения различают различные методы моделирования [2, 28]. Подходов к классификации этих методов достаточно много. Рассмотрим важнейшие из них.

## 1.2. Классификация методов моделирования по типу модели

При **полунатурном моделировании** часть системы (обычно самая громоздкая, дорогая или опасная) заменяется моделью, которая стыкуется с реальным оборудованием (датчиками, средствами обработки информации, приводами, системой управления). Примером является исследование систем ориентации космических аппаратов на конечных этапах проектирования. На Земле невозможно создать условия невесомости, поэтому аппарат помещают на специальные имитационные стенды, обеспечивающие разгрузку несущих конструкций. Вся же остальная аппаратура реальная. Такие же полунатурные эксперименты имеют место при любых проверках ракет, самолетов и т. д. с помощью специальных диагностических устройств.

Достоинство метода заключается в высокой достоверности получаемых результатов. Недостатки – в ограничениях, накладываемых реальным оборудованием, например, невозможности «сжатия» процесса моделирования во времени. Реальный объект может быть заменен как реальной физической моделью (и тогда чаще говорят о макетировании), так и математической или компьютерной.

Широко применяемое на практике **физическое моделирование** основано на использовании моделей той же физической природы, что и моделируемый объект, но с более удобными для экспериментирования параметрами: меньшими массой, габаритами и т. п. Оно применяется тогда, когда натурные испытания очень трудно или вообще невозможно осуществить, когда слишком велики/малы размеры натурального объекта или значения других его характеристик (давления, температуры, скорости протекания процесса и т. п.).

Физическое моделирование основано на свойствах подобия. Два явления физически подобны, если по заданным физическим характеристикам одного можно получить характеристики другого простым пересчетом, который аналогичен переходу от одной системы координат к другой.

Примером физического моделирования является применение аэродинамических труб для продувки уменьшенных копий самолетов или автомобилей. Подобные методы моделирования широко используются и при моделировании гидротехнических сооружений (плотин, каналов).

Достоинство этого метода, прежде всего, в том, что физическую модель зачастую сделать гораздо проще, чем создать ее математическое описание. Современные технические средства позволяют легко получить точную уменьшенную копию самолета или автомобиля. С другой

стороны, ряд явлений гораздо легче реализовать физически, нежели расчетным путем (например, эффект трения).

Недостатки данного метода заключаются в его относительной дороговизне, сложности повторения экспериментов и сложности анализа результатов. Не всегда результаты, полученные на малой модели, легко и просто переносятся на реальный объект. Основой обработки результатов физических экспериментов является специальная наука – «**теория подобия**» [4].

Использование **моделей прямой аналогии** основано на замене реального объекта моделью иной физической природы. В природе часто физически различные процессы описываются одними и теми же дифференциальными уравнениями или другого типа математическими моделями. Например, много общего имеют течение воды по трубам и ток в электрической цепи. Или заряд конденсатора подобен накоплению кинетического момента в механической системе. Естественно, используется такая модель, которая наиболее проста для реализации и исследований. Обычно это электрические модели. Их просто реализовать, процессы в них проходят быстро, легко могут быть повторены, зафиксированы регистрирующими приборами.

Методы моделирования на ЭВМ часто называют **методами не прямой аналогии** [1]. Они делятся на методы моделирования на аналоговых вычислительных машинах (АВМ) и цифровых (ЦВМ). Во всех методах предполагается наличие исходной системы уравнений в той или иной форме. Это может быть система дифференциальных или логико-дифференциальных уравнений, описывающая весь объект, либо, например, описания отдельных компонентов и топология объекта.

Методы моделирования на АВМ являются исторически более ранними. Они выросли из методов прямой аналогии и состоят в том, что отдельный электронный компонент реализует определенную элементарную модель (интегратора, усилителя, апериодического звена, устройства умножения, нелинейного звена и т. п.). В результате электронная модель имеет ту же топологию, что и исходная система. Достоинством моделирования на АВМ является то, что процессы здесь непрерывные, такие же, как в самом объекте. Если регулятор также непрерывный, то моделирование на АВМ может быть эффективным. Недостатки моделей на АВМ заключаются в сложности настройки и перестройки модели, в необходимости специальных мер для поддержания ее стабильности, а главное, в том, что вес и габариты модели пропорциональны ее сложности. К тому же, на аналоговых моделях сложно моделировать современные логико-динамические системы.

Указанных недостатков лишены методы моделирования на ЦВМ. Модель легко перестраивается. Реализация цифровых регуляторов также не представляет проблем. Основным недостатком цифровых моделей – необходимость реализации специальных алгоритмов численного интегрирования непрерывных процессов. Если объект имеет широкий разброс постоянных времени, то возникает проблема точного численного интегрирования его динамики, которая решается путем компромисса между временем счета и точностью.

Наконец, возможен **расчетно-аналитический метод моделирования**, который состоит в получении математической модели и оперировании с ней. С точки зрения исследований систем его возможности ограничены простейшими объектами. Однако формирование математической модели является неотъемлемым элементом любого метода моделирования на ЭВМ.

### 1.3. Математическое моделирование и математические модели

Введем общее понятие **математического моделирования (ММ)**, понимая под ним все методы, основанные на построении и использовании различных форм математических моделей проектируемых объектов, независимо от того, как они реализуются. В этом случае методы непрямой аналогии и расчетно-аналитический метод являются методами ММ. При ММ описание системы производится в терминах некоторой математической теории, например, теории матриц, теории дифференциальных уравнений и т. д.

ММ основано на ограниченности числа фундаментальных законов природы и принципе подобия, означающем, что явления различной физической природы могут описываться одинаковыми математическими закономерностями.

Как и всякие модели, математические модели основаны на некотором упрощении, идеализации, отбрасывании факторов, которые для данной задачи или на данном этапе исследований представляются несущественными. Например, модели объектов, используемые на начальных этапах проектирования, могут не учитывать их стохастичность, нелинейность; модели звеньев механизма могут быть получены без учета их реальной формы и т. п.

В зависимости от формы представления математические модели можно разделить на аналитические, структурные и алгоритмические.

**Аналитические модели** представляют собой отображение взаимосвязей между переменными объекта в виде дифференциальных, алгебраических или любых других систем математических уравнений. Такие

модели обычно получают на основе физических законов. Использование аналитических моделей позволяет исследовать фундаментальные свойства объекта, часто без использования ЭВМ.

**Структурная модель** представляет систему в виде совокупности элементов, а также совокупности необходимых и достаточных отношений между этими элементами и связей между системой и окружающей средой.

В простейшем случае с помощью структурной математической модели воспроизводится структура уравнений, описывающих поведение исследуемого объекта.

Вариантами структурных моделей являются графы, структурные и функциональные схемы, диаграммы и т. д. На принципах структурного ММ работают АВМ.

**Алгоритмические модели** воспроизводят пошаговый процесс численного решения уравнений, представляющих математическую модель исследуемого объекта, и обычно реализуются в форме программы для ЭВМ. Результаты исследования на алгоритмических моделях всегда являются приближенными. Применение компьютеров делает алгоритмические модели наиболее универсальными. С их помощью могут быть воспроизведены любые другие математические модели.

Математические модели технических объектов должны быть по сложности согласованы с возможностями восприятия человеком и с возможностями ЭВМ оперировать этими моделями. Как правило, решить все задачи в рамках некоторого единого описания невозможно. Обычно требуется структурирование математических моделей на несколько иерархических уровней, отличающихся детальностью описания технического объекта.

Количество иерархических уровней при моделировании определяется сложностью проектируемых объектов и возможностью средств проектирования. Однако большинство математических моделей технических объектов можно отнести к одному из трех обобщенных уровней, называемых далее микро-, макро- и метауровнями. В зависимости от места в иерархии описания математические модели делятся на модели, относящиеся к микро-, макро- и метауровням [29].

Особенностью ММ на микроуровне является отражение физических процессов, протекающих в непрерывном пространстве и времени. Типичные ММ на микроуровне – дифференциальные уравнения в частных производных (ДУЧП). В них независимыми переменными являются пространственные координаты и время. С помощью этих уравнений рассчитываются поля механических напряжений и деформаций, электрические потенциалы и напряжения, давления и температуры и т. п. Возможности применения ММ в ДУЧП ограничены отдельными деталями. Попытки

анализировать с их помощью процессы в многокомпонентных средах, сборочных единицах, электронных схемах не могут быть успешными из-за чрезмерного роста затрат машинного времени и памяти.

На макроуровне используют укрупненную дискретизацию пространства по функциональному признаку, что приводит к представлению ММ на этом уровне в виде систем обыкновенных дифференциальных уравнений (ОДУ). В этих уравнениях независимой переменной является время  $t$ , а вектор зависимых переменных составляют фазовые переменные, характеризующие состояние укрупненных элементов дискретизированного пространства. Такими переменными являются силы и скорости в механических системах, напряжения и токи в электрических системах, давления и расходы жидкостей и газов в гидравлических и пневматических системах и т. п. Макроуровень наиболее характерен для исследования мехатронных систем.

Системы ОДУ являются универсальными моделями на макроуровне, пригодными для анализа как динамических, так и установившихся состояний объектов. Модели для установившихся режимов можно также представить в виде систем алгебраических уравнений. Порядок системы уравнений зависит от числа выделенных элементов объекта. Если порядок системы приближается к 10000, то оперирование моделью становится затруднительным, и поэтому необходимо переходить к представлениям на метауровне.

На метауровне в качестве элементов принимают достаточно сложные совокупности деталей. Метауровень характеризуется большим разнообразием типов используемых ММ. Для многих объектов ММ на метауровне по-прежнему представляются системами ОДУ. Однако т. к. в моделях не описываются внутренние фазовые переменные элементы, а фигурируют только фазовые переменные, относящиеся к взаимным связям элементов, укрупненное представление элементов на метауровне означает получение ММ приемлемой размерности для существенно более сложных объектов, чем размерность ММ на макроуровне.

#### **1.4. Классификация методов математического моделирования применительно к этапу построения математической модели**

В современной науке существуют два основных подхода к построению математических моделей систем [4, 29]. Первый из них – это широко распространенный **классический подход**, который базируется на раскрытии явлений, происходящих внутри рассматриваемой системы.

Построение модели начинается с использования основных физических законов (законов Ньютона, Максвелла или Кирхгофа, законов со-

хранения массы, энергии, кинетического момента и т. д.) для описания исследуемого объекта, являющегося, например, механическим или электрическим. Из этих законов следуют различные соотношения между рассматриваемыми переменными и, в частности, связывающие их ОДУ, ДУЧП, разностные уравнения.

Базой данного подхода к построению математической модели являются дисциплины, относящиеся к соответствующим предметным областям: теоретическая механика при построении моделей механических объектов, электротехника – при построении моделей электрических цепей и т. д.

Второй подход, характерный для методологии кибернетики и получивший развитие в трудах ее основоположников [9, 32], базируется на рассмотрении системы как некоторого объекта, у которого доступными для наблюдения являются только входные и выходные переменные. Данный подход часто называют **кибернетическим моделированием**, он сводит изучение системы к наблюдению ее реакций при известных воздействиях, поступающих на вход системы. Модель системы строится при этом как описание некоторого преобразователя вектора входных переменных в вектор выходных переменных. Такая кибернетическая модель сохраняет только подобие векторов входных и выходных переменных оригинала и модели, полностью игнорируя физический смысл и внутреннюю структуру объекта.

Следует отметить, что анализ методов моделирования с точки зрения построения модели может описываться в различных терминах. Выделение классического и кибернетического подходов лишь один из вариантов. Иначе можно говорить о теоретических и экспериментальных моделях. Наиболее же информативным представляется подход к получению модели с позиций «черного» и «белого» ящиков. Его достоинство в том, что он позволяет естественным образом ввести понятие «серого» ящика. Действительно, в реальных условиях редко бывает, что об объекте ничего не известно, кроме реакций, или что об объекте известно всё. Обычно объект представляет собой «серый» ящик той или иной степени «серости». Эта серость определяется информацией об объекте, которой владеет исследователь. Может быть известна, например, структура объекта (модели), ориентировочный порядок модели, математическая схема, которую следует применять, линейность и т. д.

Соответственно, разная степень «серости» выливается в разные методы кибернетического моделирования.

Основой кибернетического моделирования являются такие разделы математической теории систем, как **методы идентификации объектов** [32] и **методы реализации временных рядов** [5].

Цель решения задач идентификации заключается в построении по входным и выходным сигналам изучаемой системы эквивалентной ей системы из заданного класса. **Эквивалентность** обычно понимается в смысле какого-либо критерия ошибки или функции потерь, являющейся функционалом от выхода объекта  $Y(t)$  и выхода модели  $Y_m(t)$ , например:

$$E = \int_0^t (Y(\tau) - Y_m(\tau))^2 d\tau.$$

Говорят, что модели эквивалентны, если значения функций потерь для этих моделей одинаковы.

**Идентификация** предполагает как использование априорной информации, так и обработку данных измерений, полученных в результате экспериментов с системой. Такой подход соответствует рассмотрению системы как «серого» ящика.

Обычно идентификация – многоэтапная процедура, состоящая из следующих основных шагов:

- 1) структурная идентификация, которая заключается в определении структуры математической модели на основе теоретических соображений,
- 2) параметрическая идентификация, включающая в себя проведение идентифицирующего эксперимента и определение оценок параметров модели по экспериментальным данным,
- 3) проверка адекватности – проверка качества модели в смысле выбранного критерия близости выходов модели и объекта.

В большинстве технических задач априорные знания об объекте позволяют получить информацию о структуре модели. В результате задача идентификации сводится к задаче оценивания параметров и/или состояний. В связи с тем, что реальные системы всегда зашумлены, идентификация относится к задачам приближенного моделирования.

Следует иметь в виду, что кибернетические модели не учитывают всего комплекса физических свойств элементов исследуемой технической системы, а лишь устанавливают обнаруживаемую в процессе эксперимента связь между отдельными параметрами системы, которые удается варьировать и/или измерять. Такие модели дают адекватное описание исследуемых процессов лишь в ограниченной области пространства переменных, в которой осуществлялось их варьирование. Поэтому кибернетические модели носят частный характер, в то время как физические законы отражают общие закономерности явлений и процессов, протекающих в технической системе.

Важно отметить также, что два указанных способа получения математических моделей – классический метод и метод кибернетического моделирования, – конечно же, не являются взаимоисключающими.

Во-первых, они используют различную исходную информацию и, соответственно, природа ошибок и неточностей в моделях разная. В случае построения моделей на основе изучения «физической реальности» это неопределенность описания среды и неполнота физической модели объекта. В случае кибернетического моделирования основной источник неточностей – зашумленность реальных систем. Соответственно, исходная информация уже искажена помехами.

Во-вторых, при моделировании сложных систем для различных элементов этих систем могут использоваться разные методы получения математических моделей.

### **1.5. Классификация методов математического моделирования применительно к этапу исследования математической модели**

ММ процесса функционирования системы можно разделить на аналитическое и имитационное.

Для **аналитического моделирования** характерно то, что процессы функционирования элементов системы записываются в виде некоторых функциональных соотношений (интегро-дифференциальных, алгебраических, конечно-разностных и т. д.) или логических условий.

Аналитическая модель может быть исследована следующими методами:

- аналитическим, когда стремятся получить в общем виде явные зависимости для искомым характеристик;
- численным, когда, не умея решать уравнения в общем виде, стремятся получить численные результаты при конкретных начальных данных;
- качественным, когда, не имея решения в явном виде, можно найти некоторые свойства решения (например, оценить устойчивость).

Наиболее полное исследование процесса функционирования можно получить, если известны явные зависимости, связывающие искомые характеристики с начальными условиями, параметрами и переменными исследуемой системы, т. е. в результате аналитического решения задачи. Однако такие зависимости удается получить только для сравнительно простых систем.

Численный метод позволяет исследовать, по сравнению с аналитическим, более широкий класс систем, но при этом полученные решения носят частный характер.

Необходимость учета стохастических свойств системы, недетерминированность исходной информации, дискретность в отдельных элементах, наличие корреляционных связей между большим числом параметров и переменных, характеризующих процессы в системах, – всё это приводит к построению сложных математических моделей, которые не

могут быть применены в инженерной практике при исследовании таких систем аналитическими методами. Это также не позволяет расчленивать систему и использовать принцип суперпозиции в отношении влияющих факторов. Пригодные для практических расчетов аналитические соотношения удается получить лишь при упрощающих предположениях, обычно существенно искажающих фактическую картину исследуемого процесса. Указанные обстоятельства приводят к тому, что при исследовании сложных систем наиболее эффективными являются методы имитационного моделирования.

Под **имитационным моделированием** обычно понимают такое моделирование, при котором реализующий модель алгоритм воспроизводит процесс функционирования системы во времени, причем имитируются элементарные явления, составляющие процесс, с сохранением их логической структуры и последовательности протекания во времени.

Указывая, что данная модель имитационная, мы обычно подчеркиваем, что, в отличие от других типов абстрактных моделей, в этой модели сохранены и легко узнаваемы такие черты моделируемого объекта, как структура, связи между компонентами, способ передачи информации. С имитационными моделями также обычно связывают и требование иллюстрации их поведения с помощью принятых в данной прикладной области графических образов.

Основным преимуществом имитационного моделирования, по сравнению с аналитическим, является возможность решения более сложных задач. Имитационные модели позволяют достаточно легко учитывать факторы, которые создают трудности при аналитических исследованиях: наличие дискретных и непрерывных элементов, нелинейные характеристики элементов, случайные воздействия и т. д.

Кроме того, имитационная модель обладает гибкостью варьирования структуры, алгоритмов и параметров моделируемой системы, что важно с точки зрения поиска оптимального варианта построения системы. Данная модель позволяет включать в процедуру моделирования результаты натуральных испытаний реальной системы или ее частей.

В настоящее время имитационное моделирование – наиболее эффективный метод исследования больших систем, а часто и единственный практически доступный метод получения информации о поведении системы, особенно на этапе проектирования.

Главным недостатком, проявляющимся при машинной реализации метода имитационного моделирования, является то, что решение, полученное при анализе имитационной модели, всегда носит частный характер, т. к. оно соответствует фиксированным элементам структуры, алгоритмам поведения и значениям параметров системы, конкретным усло-

виям и воздействиям внешней среды. Поэтому для полного анализа характеристик процесса приходится многократно воспроизводить имитационный эксперимент, варьируя исходные данные.

Несмотря на то, что имитационное моделирование является мощным инструментом исследования систем, его применение не всегда рационально. Издержки, связанные с имитационным моделированием, всегда много выше, чем при аналитических исследованиях, и часто выше, чем при физическом моделировании. Следует хорошо подумать, прежде чем начинать решать задачу таким путем.

В качестве основных критериев целесообразности применения метода имитационного моделирования, по сравнению с аналитическим подходом, можно указать отсутствие законченной математической постановки задачи, неразработанность методов ее аналитического решения либо их чрезмерная сложность и трудоемкость, слабая подготовка персонала, не позволяющая ими воспользоваться.

Если сравнивать с физическим моделированием, то применение имитационного моделирования целесообразно в том случае, когда иных методов решения задачи просто нет либо требуется существенное «сжатие» по времени.

## **1.6. Характеристики математической модели**

Математическая модель всегда отражает только часть свойств реального объекта, определяемую целями моделирования. Например, специалиста, автоматизирующего технологический процесс, может интересовать кинематическая модель манипулятора, которая позволяет рассчитать объем зоны обслуживания и траектории перемещения рабочего органа манипулятора. Человеку, проектирующему систему управления робота, кроме кинематической, нужна динамическая модель, в которой учитывались бы приведенные к осям приводов моменты инерции звеньев манипулятора, жесткость звеньев, трение в кинематических парах и т. п. Совершенно иные модели использует конструктор, призванный обеспечить необходимые прочность, жесткость и дизайн проектируемого манипулятора.

Естественно, что при построении модели стремятся как можно более точно отразить свойства объекта, чтобы модель верно отражала свойства моделируемого объекта в смысле, определенном целью моделирования. С другой стороны, чем проще математическая модель, тем легче ее исследовать и использовать при решении задач синтеза. Искусство моделирования состоит в умении выбрать факторы, существенные с точки зрения цели моделирования, и пренебречь эффектами, которые, усложняя математическую модель, не оказывают заметного влияния на поведение системы.

### *Адекватность*

Проблема соответствия модели реальному объекту очень важна. Принято говорить, что модель адекватна оригиналу, если она верно отражает интересующие нас свойства оригинала и может быть использована для предсказания его поведения. При этом адекватность модели зависит от целей моделирования и принятых критериев. Например, модель, адекватная на этапе поискового проектирования, при детализации проекта теряет это свойство и становится слишком «грубой». Учитывая изначальную неполноту модели, можно утверждать, что идеально адекватная модель в принципе невозможна.

В рамках каждой научной дисциплины разрабатывается совокупность приемов и правил, следование которым позволяет создавать отвечающее исходным гипотезам описание и получать предварительную оценку его адекватности рассматриваемому явлению. Окончательный анализ данной оценки осуществляется на этапе проверки модели, на котором устанавливается правомерность исходных посылок в соответствии с целью исследования реального явления и определяется степень соответствия ему полученной модели.

Приближенность модели к действительному объекту можно рассматривать в следующих аспектах [2, 4, 3]:

- с точки зрения корректности связи «вход-выход»;
- с точки зрения корректности декомпозиции модельного описания применительно к целям исследования и использования моделей.

Степень соответствия моделей в первом случае принято называть собственно адекватностью, во втором – аутентичностью. В последнем случае требуется, чтобы все подмодели и их элементы были адекватны соответствующим прототипам реального объекта. Проблема аутентичности значительно сложнее адекватности и может рассматриваться лишь при получении математической модели классическим способом, т. е. «изнутри». Первая проблема допускает строгий анализ, однако также является актуальной, сложной и далекой от полного разрешения.

Можно выделить два способа оценки адекватности, один из которых используется, если есть возможность сравнить модель и объект, другой – если такой возможности нет.

Первый способ представляет собой разовую процедуру, основанную на сравнении данных, наблюдаемых на реальном объекте, с результатами вычислительного эксперимента, проведенного с моделью. Модель считается адекватной, если отражает исследуемые свойства с приемлемой точностью, где под точностью модели понимается количественный показатель, характеризующий степень различия модели и изучаемого явления. Таким образом, в первом способе мера адекватности является количественной. Ею может быть значение некой функции несогласованности между моделью и измерениями.

Мера адекватности принципиально является векторной и взвешенной. Векторность связана с тем, что реальные объекты характеризуются не одним, а несколькими выходными показателями. Причем один и тот же выходной параметр модели может оказаться важным для одних применений модели и второстепенным для других.

Обычно погрешность модели  $\varepsilon$  по всей совокупности  $m$  учитываемых выходных переменных оценивается одной из норм вектора

$$\varepsilon = (\varepsilon_1, \varepsilon_2, \varepsilon_3, \dots, \varepsilon_m):$$

$$\varepsilon = \|\bar{\varepsilon}\| = \max |\varepsilon_i|, i = 1, \dots, m, \text{ или } \varepsilon = \|\bar{\varepsilon}\| = \sqrt{\sum_{i=1}^m k_i \varepsilon_i^2},$$

где  $\varepsilon_i$  – относительная погрешность модели по  $i$ -ой выходной переменной. Возможна также вариация данного подхода, когда объект заменяется эталонной моделью, заведомо более точной, чем исследуемая. Использование количественной характеристики позволяет сравнивать различные модели по степени их адекватности.

Второй способ представляет собой перманентную процедуру, основанную на использовании верификационного подхода, нацеленного на формирование определенного уровня доверия к модели. Такая процедура всегда используется, если нет возможности проверить модель экспериментально, например, объект находится в стадии проектирования либо эксперименты с объектом невозможны.

Процесс оценки достоверности имеет две стороны:

- приобретение уверенности в том, что модель ведет себя как реальная система;
- установление того, что выводы, полученные на ее основе, справедливы и корректны.

По сути, он сводится к обычному компромиссу между стоимостью проверки и последствиями ошибочных решений.

Для проверки модели могут использоваться разные приемы:

- проверка физического смысла (соблюдение физических законов);
- проверка размерности и знаков;
- проверка пределов;
- проверка тренда, т. е. тенденции изменения выходных переменных в зависимости от внутренних и внешних переменных, и т. п.

Например, при моделировании вращательного движения твердого тела необходимо убедиться в том, что выполняется закон сохранения кинетического момента. Также необходимо быть уверенным, что модель не будет давать абсурдных результатов, если параметры выходят на пределы.

### ***Экономичность***

Экономичность математических моделей определяется двумя основными факторами:

- затратами машинного времени на прогон модели;
- затратами оперативной памяти, необходимой для размещения модели. Особенно это актуально для систем реального времени, например, при использовании модели в контуре управления космического аппарата.

### ***Универсальность***

Универсальность моделей определяет область их возможных применений. Можно строить отдельные модели для различных экспериментов (например, детерминированные и стохастические) или для разных режимов работы. Здесь нужен взвешенный подход. Обычно универсальность достигается тем, что в модель включается большое число внутренних параметров, что отрицательно влияет на экономичность.

### ***Устойчивость***

При оценке адекватности модели может быть использовано лишь ограниченное подмножество всех возможных значений входных параметров (рабочей нагрузки и внешней среды).

Устойчивость модели – это ее способность сохранять адекватность при исследовании системы на всем возможном диапазоне рабочей нагрузки, а также при внесении изменений в конфигурацию системы.

Универсальной процедуры проверки устойчивости модели не существует. Разработчик вынужден прибегать к методам «для данного случая», к частичным тестам и здравому смыслу. Часто проверка состоит в сравнении результатов моделирования и результатов измерения на системе после внесения в нее изменений. Если результаты моделирования приемлемы, уверенность в устойчивости модели возрастает.

В общем случае можно утверждать, что чем ближе структура модели структуре системы и чем выше степень детализации, тем устойчивее модель.

### ***Чувствительность***

Очевидно, что устойчивость является положительным свойством модели. Однако если изменение входных воздействий или параметров модели (в некотором заданном диапазоне) не отражается на значениях выходных переменных, то польза от такой модели невелика. В связи с этим возникает задача оценивания чувствительности модели к изменениям параметров рабочей нагрузки и внутренних параметров самой системы.

Обычно такую оценку проводят по каждому параметру отдельно. Основана она на том, что диапазон возможных изменений параметра известен. Данные, полученные при оценке чувствительности модели, могут быть использованы, в частности, при планировании экспериментов: большее внимание должно уделяться тем параметрам, по которым модель является более чувствительной.

## Глава 2

# АНАЛИТИЧЕСКОЕ МОДЕЛИРОВАНИЕ МЕХАТРОННЫХ СИСТЕМ. МЕТОД ГРАФОВ СВЯЗЕЙ

### 2.1. Компонентное моделирование

С точки зрения уровня моделирования (см. раздел 1.3.) мехатронные системы являются системами с сосредоточенными параметрами и относятся к макроуровню, на котором исследуемый объект имеет сложную неоднородную структуру, включающую элементы различной физической природы, взаимодействующие друг с другом через энергетические и информационные связи. В связи с этим серьезный интерес представляют возможные подходы к структурированию подобных объектов.

Весьма эффективен при получении моделей технических систем метод функционально законченных элементов [29]. Он основан на выделении типовых элементов технического объекта, завершенных в конструктивном отношении и предназначенных для выполнения определенных функций: двигатель, золотник, усилитель и т. д. Имея библиотеку математических моделей функционально законченных элементов и зная структуру технического объекта, можно составить его полную математическую модель.

Однако отдельный функциональный элемент может также представлять собой достаточно сложный объект. Ничто не мешает применить этот метод иерархически, т. е. строить на том же принципе модели подсистем и отдельных элементов. В результате на нижнем уровне данный метод превращается в то, что в разных источниках называется **методом сосредоточенных масс** [29], или **методом мультидоменного моделирования** [15]. Суть его в том, что в системе выделяются отдельные элементарные материальные элементы, рассматриваемые как носители определенных физических свойств с точки зрения генерации, накопления, передачи и преобразования энергии. Таких элементов совсем не много. В работе [29] они названы энергетическими доменами. Каждый из доменов получает из окружающей среды и передает в нее мощность  $N(t)$ , которая всегда может быть представлена как произведение двух фазовых переменных, одна из которых –  $f(t)$  – называется потоковой, другая –  $e(t)$  – потенциальной. Во всех случаях домены являются моделями простых физических устройств, отражают основные физические свойства технических объектов любой физической природы –

инерционные, упругие и диссипативные. С точки зрения преобразования энергии это соответствует аккумулярованию кинетической энергии, аккумулярованию потенциальной энергии и рассеиванию энергии.

Домены составляют основу любой физической модели, но их недостаточно для создания модели. Нужны еще, как минимум, модели источников энергии и преобразователей параметров потока энергии. В сумме такой набор может рассматриваться как структурный базис нижнего уровня.

Физические свойства элемента, в т. ч. и домена, описываются математической моделью

$$\varphi(e, f) = 0,$$

отражающей зависимость между фазовыми переменными их связей с окружающей средой. Эти выражения называются **компонентными уравнениями**. Из доменов нижнего уровня могут формироваться более сложные компоненты, характеризующие не одно, а несколько свойств объекта, описываемые системами компонентных уравнений, у которых потенциальные и потоковые переменные носят векторный характер.

Процедура построения математической модели технического объекта представляет собой последовательную интерпретацию свойств этого объекта в форме некоторой структуры, в которую включаются типовые компоненты – носители этих свойств. Например, свойство упругости механической системы достигается введением в нее компонента с упругими свойствами.

Достоинство такого подхода, обычно называемого **компонентным моделированием**, состоит в прозрачности процедуры, в простоте и наглядности самой модели, в легкости внесения в модель изменений, связанных с учетом или неучетом тех или иных свойств объекта.

Для получения полной аналитической математической модели технической системы необходимо объединить все компонентные уравнения в общую систему уравнений. Объединение осуществляется на основе физических законов, выражающих условия равновесия и непрерывности фазовых переменных. Уравнения этих законов называются **топологическими уравнениями**. Все топологические уравнения являются алгебраическими. Условия равновесия записываются для потенциальных переменных в виде

$$\sum_i e_i = 0,$$

а условия непрерывности – для фазовых переменных типа потока

$$\sum_k f_k = 0.$$

Если полная математическая модель строится вручную, топологические уравнения формируются исследователем. Если решается задача автоматизированного моделирования, эта функция возлагается на ЭВМ.

Форма компонентных и топологических уравнений одинакова для систем различной физической природы. Полная математическая модель технического объекта, полученная как объединение компонентных и топологических уравнений, представляет систему алгебраических уравнений и ОДУ относительно фазовых переменных компонентов – потоков и потенциалов. Полученная модель может служить для автоматизированного моделирования технического объекта либо для перехода к другим традиционным формам математических моделей.

Одним из эффективных методов реализации идеи компонентного моделирования на нижнем уровне (уровне энергетических доменов) является метод графов связей [24]. **Метод графов связей** относится к группе топологических методов, т. е. методов, использующих графическое представление исследуемого объекта. Он позволяет на единой методологической базе моделировать объекты, содержащие элементы различной физической природы – электрические, механические, электромеханические, гидравлические, пневматические и т. д. Для расширения сферы его применения нужно лишь найти соответствующую интерпретацию общих понятий метода в соответствующей предметной области.

Метод графов связей является удобным инструментом для теоретического получения моделей систем небольшой сложности. Это связано с высокой степенью формализации метода, в частности, с введением моделей узлов, что позволяет оперировать только компонентными уравнениями при формировании моделей сложных объектов. Для графов связей разработаны сравнительно простые процедуры перехода к традиционным моделям в форме систем дифференциальных и алгебраических уравнений, передаточных функций и операторно-структурных схем.

Метод графов связей перспективен и для автоматизированного моделирования, как средство для формирования моделей компонентов сложных объектов. Некоторые пакеты допускают прямое включение элементов графов связей в структурные модели систем.

Наконец, метод графов связей позволяет лучше понять особенности и взаимосвязи двух основных подходов к автоматизированному моделированию мехатронных систем – структурного и физического мультидоменного (другими словами, моделирования на уровне передачи сигналов и моделирования на уровне передачи энергии), что полезно для пользователя современных систем автоматизированного моделирования.

## 2.2. Основные определения графов связей

Метод графов связей, или связных графов [24], основан на представлении о том, что любые физические процессы состоят из элементарных актов преобразования энергии. Такими элементарными процессами являются накопление энергии, диссипация (потери) энергии и преобразование энергии без потерь. Таким образом, метод графов связей демонстрирует известное единство природы и протекающих в ней физических процессов.

**Граф связей** представляет собой совокупность элементов, соответствующих основным типам преобразования энергии и изображаемых в качестве вершин графа, соединенных связями (дугами графа).

Связь изображается в графе линией с полустрелкой, показывающей принимаемое при моделировании за положительное направление передачи энергии. Для каждой связи в графе определены шесть величин, три из которых являются интегральными.

Каждый элемент графа характеризуется уравнением или системой уравнений, включающих переменные, относящиеся к его связям.

### 2.2.1. Переменные связей

Основные переменные связей – усилие  $e(t)$  и поток  $f(t)$ . Эти величины являются функциями времени и называются **переменными мощности связи**. Остальные четыре переменные вычисляются через основные по следующим формулам:

а) мощность

$$N(t) = e(t) \cdot f(t), \quad (2.1)$$

б) энергия

$$E(t) = E(t_0) + \int_{t_0}^t N(\tau) d\tau, \quad (2.2)$$

в) перемещение

$$q(t) = q(t_0) + \int_{t_0}^t f(\tau) d\tau, \quad (2.3)$$

г) момент

$$p(t) = p(t_0) + \int_{t_0}^t e(\tau) d\tau. \quad (2.4)$$

Величина  $E(t) - E(t_0)$  – полезная энергия, передаваемая через связь в направлении, определенном полустрелкой.

### 2.2.2. Интерпретация переменных связей

Некоторые интерпретации переменных связей в системах различной физической природы приведены в табл. 2.1. Нетрудно проверить, что произведение усилия на поток в каждом случае дает мощность.

Отметим, что принятые в табл. 2.1 способы интерпретации переменных не единственно возможные. Можно назвать ток в электрических системах усилием, а напряжение – потоком. Соответственно, изменятся и интерпретации момента и перемещения. Подобное свойство называется **дуальностью графа**.

Таблица 2.1

Переменные связи	Системы			
	электрические	механические поступательные	механические вращательные	гидравлические
Усилие $e(t)$	напряжение $u(t)$	сила $F(t)$	момент силы $M(t)$	давление $P(t)$
Поток $f(t)$	ток $i(t)$	скорость $v(t)$	угловая скорость $\omega(t)$	расход $Q(t)$
Момент $p(t)$	потокосцепление $\psi(t)$	импульс силы $p(t) = \int F dt$	кинетический момент $H(t)$	импульс давления
Перемещение $q(t)$	заряд $q(t)$	перемещение $x(t)$	угол поворота $\varphi(t)$	объем $V(t)$

### 2.2.3. Типовые элементы графов связей

Элементы графа связей делятся на четыре группы: источники энергии, аккумуляторы энергии, элементы рассеивания (потери) энергии и преобразователи энергии без потерь.

В первую группу входят два идеальных источника энергии (рис. 2.1, *a, b*): источник усилия, обозначаемый как  $SE$ , и источник потока, имеющий обозначение  $SF$ . Источник усилия задает значение

$$e = e(t), \quad (2.5)$$

а источник потока

$$f = f(t). \quad (2.6)$$

В электрических системах этим элементам соответствуют идеальные источники ЭДС и тока. Легко устанавливаются аналогии и в системах иной природы. В соответствии со смыслом источников энергия выходит из них, что и отражается направлением полустрелок на связях источников.

Группа аккумуляторов тоже включает два элемента: инерционность  $I$  и емкость  $C$  (рис. 2.1, *c, d*).

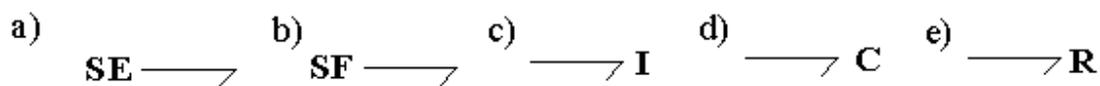


Рис. 2.1. Односвязные элементы:

$a$  – источник усилия,  $b$  – источник потока,  $c$  – инерционность,  
 $d$  – емкость,  $e$  – элемент потерь

Взаимосвязь между усилиями и потоками для аккумуляторов может быть задана уравнениями

$$p = \varphi_I(f) \quad (2.7)$$

для инерционности и

$$q = \varphi_C(e) \quad (2.8)$$

для емкости.

Если рассматривать линейные модели, то уравнения аккумуляторов можно записать в виде

$$I \frac{df}{dt} = e(t); \quad (2.9)$$

$$C \frac{de}{dt} = f(t), \quad (2.10)$$

где для обозначения параметров аккумуляторов  $I$  и  $C$  используются те же буквы, что и для обозначения элементов.

Аккумуляторы различаются тем, что инерционность имеет свойство накапливать кинетическую энергию, а емкость – потенциальную.

В третью группу входит один элемент потерь  $R$ , для которого в общем случае

$$e(t) = \varphi_R(f(t)). \quad (2.11)$$

В простейшем случае уравнению (2.11) соответствует линейное уравнение

$$e(t) = R f(t), \quad (2.12)$$

где  $R$  – параметр элемента.

Четвертая группа включает четыре преобразователя энергии: трансформатор, гиратор, узел общего усилия и узел общего потока.

Трансформатор  $TF$  (рис. 2.2) преобразует энергию в соответствии с формулами:

$$\begin{aligned} e_1(t) &= m e_2(t), \\ f_2(t) &= m f_1(t), \end{aligned} \quad (2.13)$$

где  $m$  – коэффициент передачи трансформатора.



Рис. 2.2. Трансформатор:  
*a – с постоянным коэффициентом, b – модулированный*

Нетрудно увидеть, что мощности в обеих связях трансформатора равны

$$e_1 f_1 = e_2 f_2.$$

Трансформатор может иметь переменный коэффициент передачи, зависящий как от времени, так и от некоторой другой переменной. Такой трансформатор называется модулированным и обозначается как *MTF*.

Примерами трансформаторов являются редуктор, трансформатор переменного тока, рычаг.

Гиратор *GY* можно получить из трансформатора, если в одной из его связей поменять местами усилие и поток. Уравнения гиратора имеют следующий вид:

$$\begin{aligned} e_1(t) &= r f_2(t), \\ e_2(t) &= r f_1(t), \end{aligned} \quad (2.14)$$

где *r* – коэффициент передачи гиратора.

Гиратор, как и трансформатор, сохраняет мощность, т. е.

$$e_1 f_1 = e_2 f_2.$$

Гираторы тоже могут быть модулированными и изображаются, как это показано на рис. 2.3.

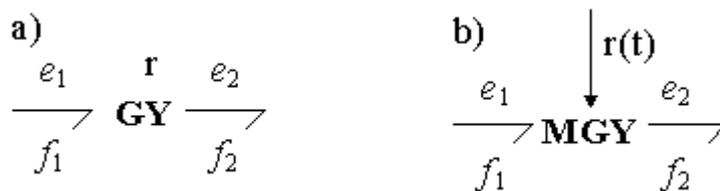


Рис. 2.3. Гиратор:  
*a – с постоянным коэффициентом, b – модулированный*

Отметим, что свойствами гиратора обладает, например, гироскоп, а в радиоэлектронике известно применение специальных устройств, называемых гираторами, с целью замены индуктивностей емкостями.

Гираторы и трансформаторы могут отображать преобразование энергии одной физической природы, а могут отображать также преобра-

зование механической энергии в электрическую, электрической – в магнитную и т. п.

**Узел общего усилия (0-узел)** может иметь любое количество связей (рис. 2.4).

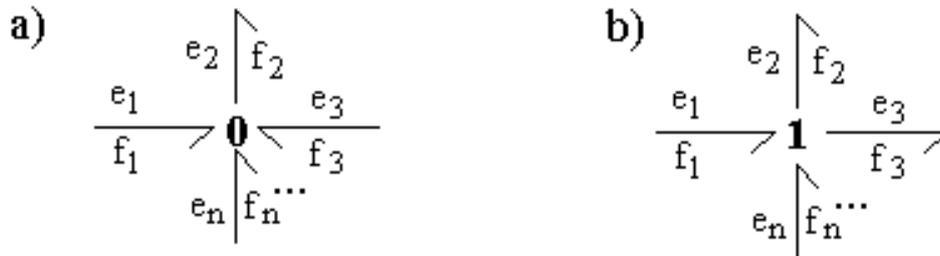


Рис. 2.4. Узлы графа связей:

*a* – узел общего усилия (0-узел), *b* – узел общего потока (1-узел)

Узел получил свое название потому, что усилия во всех его связях равны

$$e_1 = e_2 = e_3 = \dots e_n. \quad (2.15)$$

При этом алгебраическая сумма потоков в связях 0-узла равна нулю:

$$f_1 - f_2 + f_3 + \dots + f_n = 0. \quad (2.16)$$

Учитывая (2.15) и (2.16), можно получить закон сохранения энергии в 0-узле:

$$e_1 f_1 - e_2 f_2 + e_3 f_3 + \dots + e_n f_n = 0. \quad (2.17)$$

**Узел общего потока (1-узел)** во всем подобен узлу общего усилия, если поменять местами усилия и потоки. Таким образом, для 1-узла:

$$f_1 = f_2 = f_3 = \dots = f_n = 0; \quad (2.18)$$

$$e_1 - e_2 - e_3 + \dots + e_n = 0; \quad (2.19)$$

$$e_1 f_1 - e_2 f_2 - e_3 f_3 + \dots + e_n f_n = 0. \quad (2.20)$$

Знаки слагаемых в (2.16) и (2.19) определяются направлением полустрелок в связях.

Узлы общего усилия и потока отображают два возможных способа разветвления или суммирования потоков энергии в физических системах.

### 2.2.4. Физическая интерпретация основных элементов графов связей

Примеры физической интерпретации элементов графов связей для электрических и механических систем представлены в табл. 2.2 [24].

Таблица 2.2

Электрические системы		Механические системы	
Элемент графа связей	Физическое устройство, эффект	Элемент графа связей	Физическое устройство, эффект
$SE$	Источник ЭДС	$SE$	Источник силы, момента
$SF$	Источник тока	$SF$	Источник скорости
$R$	Активное сопротивление	$R$	Вязкое трение
$I$	Индуктивность	$I$	Масса, момент инерции
$C$	Емкость	$C$	Пружина
$TF$	Трансформатор	$TF$	Редуктор, рычаг
$GY$	–	$GY$	Гироскоп
0-узел	Параллельное соединение электрических цепей	0-узел	Подвижное соединение механических звеньев
1-узел	Последовательное соединение элементов	1-узел	Жесткое соединение механических звеньев

### 2.3. Моделирование электрических систем на графах связей

Аналогии между элементами графов связей и электрическими элементами очевидны. Поэтому правила построения графа связей для электрических цепей могут быть получены на основе простых рассуждений.

Первое правило касается отображения в графах связей электрических двухполюсников. Для примера на рис. 2.5 показано прохождение электрической мощности через резистор. При этом часть входной мощности  $N_1$  проходит через резистор (мощность  $N_2$ ), а часть  $N_3$  теряется в форме рассеиваемого тепла. Поскольку токи в выводах резистора равны, т. е.

$$f_1 = f_2,$$

то разветвление мощности отражается в графе связей узлом общего потока. Потери мощности отображаются элементом потерь  $R$ . Весь узел с элементом потерь описывается уравнениями

$$\begin{aligned} f_1 &= f_2 = f_3, \\ e_1 - e_2 - e_3 &= 0, \\ e_3 &= R f_3. \end{aligned}$$

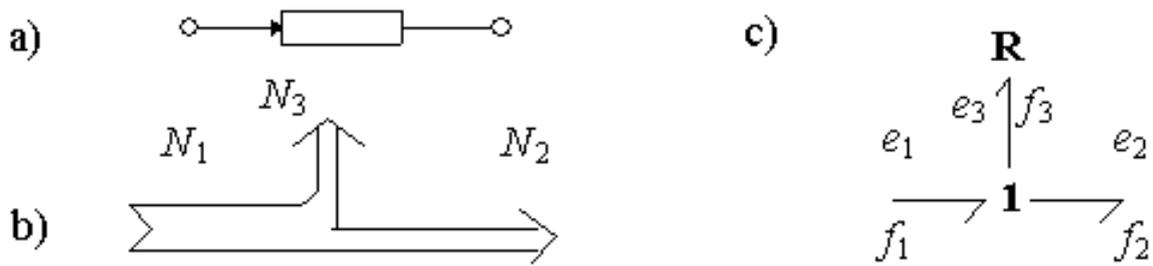


Рис. 2.5. Поток мощности в резисторе:  
*a* – резистор как двухполюсник, *b* – потоки мощности в резисторе,  
*c* – граф резистора

Подобные рассуждения могут быть проведены и для других электрических двухполюсников: источников ЭДС и тока, индуктивности и емкости. Во всех случаях двухполюсники представляются в графе связей 1-узлом и соответствующим односвязным элементом графа связей: источником усилия  $SE$  или потока  $SF$ , инерционности  $I$ , емкости  $C$ .

Второе правило, проиллюстрированное на рис. 2.6, устанавливает соответствие между узлом электрической цепи и его отображением в графе. Поскольку электрический потенциал всех входящих в узел проводников одинаков, а сумма токов в узле равна нулю, то узел электрической цепи отображается в графе связей узлом общего усилия.

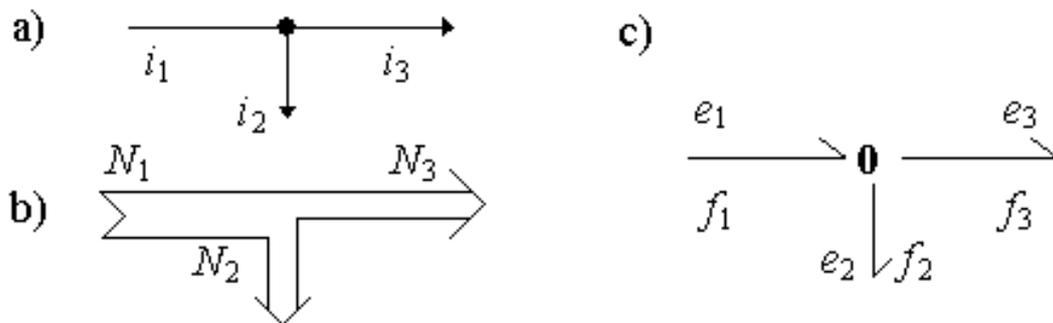


Рис. 2.6. Поток мощности в узле электрической схемы:  
*a* – узел электрической цепи, *b* – потоки мощности в узле,  
*c* – граф узла электрической цепи

Эти два простых правила позволяют строить математическую модель в форме графа связей для любой электрической цепи, состоящей из двухполюсников.

Для примера на рис. 2.7 представлены простая электрическая схема и соответствующий ей граф связей.

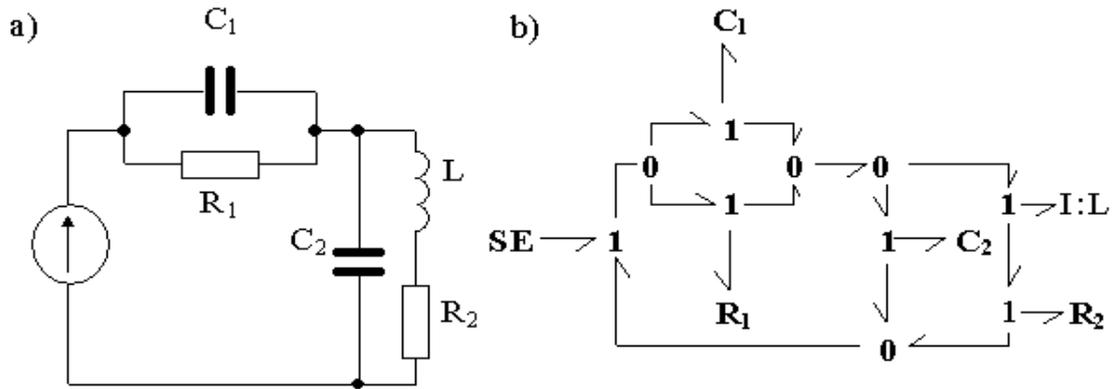


Рис. 2.7. Граф связей электрической цепи:  
*a* – электрическая цепь, *b* – граф электрической цепи

Обозначение  $I:L$  показывает, что элемент  $I$  графа соответствует индуктивности  $L$  в электрической схеме.

## 2.4. Эквивалентные преобразования графов связей

Одним из достоинств графа связей является возможность эквивалентных преобразований, позволяющих упростить граф.

Некоторые из этих преобразований приведены в табл. 2.3.

Таблица 2.3

№ п/п	Исходный граф	Результат
1	$\text{---} \nearrow \mathbf{1} \searrow$	$\text{---} \nearrow \searrow$
2	$\text{---} \rightarrow \mathbf{0} \rightarrow$	$\text{---} \rightarrow \rightarrow$
3	$  \begin{array}{c}  2 \uparrow \\  1 \rightarrow \mathbf{1} \text{---} \mathbf{1} \rightarrow 3 \\  4 \uparrow  \end{array}  $	
4	$  \begin{array}{c}  2 \downarrow \\  1 \rightarrow \mathbf{0} \text{---} \mathbf{0} \rightarrow 3 \\  4 \uparrow  \end{array}  $	$  \begin{array}{c}  2 \downarrow \\  1 \rightarrow \mathbf{0} \rightarrow 3 \\  4 \uparrow  \end{array}  $
5	$  \begin{array}{c}  2 \uparrow \\  \mathbf{0} \searrow \mathbf{1} \rightarrow 3 \\  1 \rightarrow \mathbf{1} \text{---} \mathbf{1} \\  \mathbf{0} \swarrow \\  4 \downarrow  \end{array}  $	$  \begin{array}{c}  2 \quad 4 \\  \swarrow \mathbf{1} \searrow \\  1 \quad 3 \\  \rightarrow \mathbf{0} \rightarrow  \end{array}  $

№ п/п	Исходный граф	Результат
6		
7		
8		

Первые две строки таблицы показывают, что можно исключить из графа узел с двумя связями при условии, что направление мощности (направление полустрелок) в узле не меняется. Строки третья и четвертая иллюстрируют, что два связанных узла одного типа можно заменить одним. Следствием из этого свойства является возможность переставлять местами узлы одного типа вместе с их связями.

Менее очевидные эквивалентные преобразования, показанные в пятой и шестой строках таблицы, заменяют четырехугольник из 0-узлов и 1-узлов на два узла, заметно упрощая граф. В таблице показано только два варианта направления связей четырехугольника из многих, для которых такое преобразование имеет место.

Последние две строки таблицы демонстрируют изменение направления связей, которое может производиться одновременно для всех связей узла. Это свойство можно обобщить и на другие, более сложные структуры графа. Все указанные в табл. 2.3 эквивалентные преобразования легко доказываются с использованием уравнений (2.15) – (2.19).

Следует отметить, что направление любой связи в графе, кроме односвязных элементов и трансформаторов, может быть изменено на противоположное. Такое преобразование не является эквивалентным, но допустимо, т. к. соответствующее изменение знаков некоторых потоков и усилий бывает обычно безразличным.

Одно из важных эквивалентных преобразований, не приведенное в табл. 2.3, справедливо только для графов связей электрических цепей. Оно состоит в исключении из графа одного из 0-узлов вместе со всеми его связями и объясняется линейной зависимостью уравнений суммирования потоков, записанных для всех 0-узлов. Это свойство следует из особенности электрических цепей, при расчетах которых тоже не записывается закон Кирхгофа для одного из узлов схемы.

Применим рассмотренные преобразования к графу, построенному на рис. 2.7, *b*.

Во-первых, исключим нижний 0-узел из графа. Оставшаяся его часть показана на рис. 2.8, *a*. Теперь в получившемся графе можно преобразовать «четырёхугольник», а после этого исключить лишние 1-узлы в связях элементов  $SE$ ,  $C_2$  и  $R_2$ . Результат преобразований приведен на рис. 2.8, *b*.

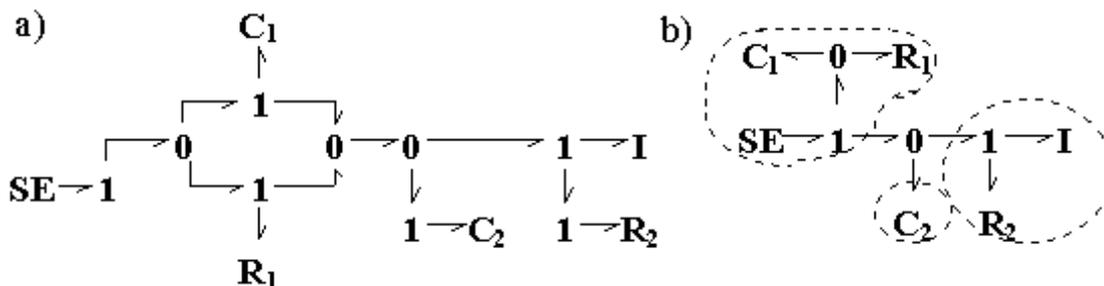


Рис. 2.8. Эквивалентные преобразования графа связей:  
*a* – исходный граф, *b* – преобразованный граф

Тот же результат может быть получен проще, если в исходном графе исключить другой 0-узел, а точнее – сразу два связанных 0-узла, разорвав, таким образом, одновременно два четырёхугольника графа. В оставшемся графе (рис. 2.9) теперь достаточно изменить направление всех связей нижнего 0-узла и исключить лишние 1-узлы.

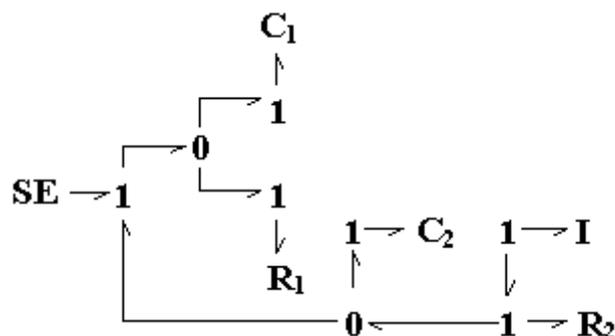


Рис. 2.9. Другой вариант эквивалентных преобразований

Несмотря на значительные «потери» в количестве связей, граф, полученный на рис. 2.8, *b*, полностью отражает все свойства исходной

электрической схемы. Весьма интересное свойство графов связей состоит в том, что каждый 0-узел соединяет графы параллельных частей схемы, а каждый 1-узел связывает модели последовательных участков. Поэтому 0-узел можно назвать узлом параллельного соединения, а 1-узел – узлом последовательного соединения. Для иллюстрации этого свойства на рис. 2.8, *b* штриховой линией обведены три части графа, связанные 0-узлом. Нетрудно убедиться в том, что этим частям в схеме действительно соответствуют параллельные цепи. Каждый 0-узел и 1-узел позволяет увидеть свой вариант топологии схемы.

Отмеченные интересные свойства узлов графа позволяют значительно сократить процедуру построения графа связей электрических цепей. Во многих случаях, когда схема может быть представлена параллельно и последовательно соединенными компонентами, граф связей в конечном виде может быть построен сразу, без промежуточных этапов.

## 2.5. Моделирование механических систем на графах связей

При моделировании механических систем естественно интерпретировать усилие  $e$  как силу  $F$  в поступательном движении или момент силы  $M$  во вращательном движении, а поток  $f$  – как скорость, линейную  $v$  или угловую  $\omega$ . Тогда уравнение инерционности представляет собой не что иное, как второй закон Ньютона:

$$m \frac{dv}{dt} = F, \quad (2.21)$$

где  $m$  – масса поступательно движущегося тела, или

$$J \frac{d\omega}{dt} = M, \quad (2.22)$$

где  $J$  – момент инерции тела.

Узел общего потока (рис. 2.10) идеально подходит для того, чтобы отобразить принцип Даламбера: равенство нулю суммы всех внешних сил и силы инерции

$$e_1 + e_2 + e_3 - e_4 - m \frac{df_5}{dt} = 0.$$

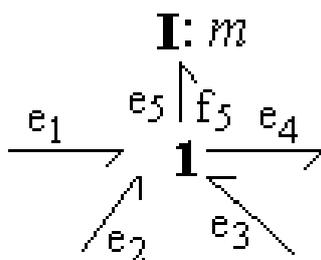


Рис. 2.10. Графическая интерпретация принципа Даламбера

Этот граф является, в сущности, моделью динамики тела с массой  $m$ , находящегося под действием суммы сил, которые могут быть как активными внешними силами, так и реакциями связей с другими телами механической системы.

Одновременно 1-узел можно использовать в качестве узла жесткого соединения твердых тел, при котором они, по существу, становятся единым телом.

Действительно, 1-узел – это узел общего потока, что в принятой терминологии соответствует общей (равной) скорости для всех связей 1-узла, а равенство скоростей означает движение двух тел как единого целого. Два эквивалентных графа на рис. 2.11 иллюстрирует такую связь твердых тел с массами  $m_1$  и  $m_2$ .

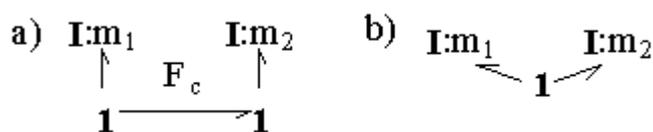


Рис. 2.11. Жесткое соединение тел

Отметим, что граф, приведенный на рис. 2.11, а, наглядно демонстрирует также третий закон Ньютона о том, что действие равно противодействию. Действительно, сила  $F_c$  действует на оба тела, но с противоположным знаком.

В противоположность 1-узлу узел общего усилия (0-узел) можно считать узлом свободного соединения твердых тел. На рис. 2.12 показано, что соединение двух тел через 0-узел позволяет каждому телу иметь свою скорость ( $v_1$  и  $v_2$  соответственно).

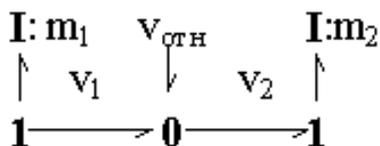


Рис. 2.12. Подвижное соединение тел

При этом третья связь 0-узла характеризует относительное движение тел:

$$v_{\text{отн}} = v_2 - v_1. \quad (2.23)$$

Поэтому 0-узел можно использовать при моделировании упругих связей и трения, которые появляются лишь при наличии относительного движения двух тел.

Рассмотрим физический смысл других элементов графов связей, которые при моделировании механических систем связываются с 0-узлом.

Уравнение емкости для поступательного движения приобретает следующий вид:

$$C \frac{dF}{dt} = v. \quad (2.24)$$

Если его проинтегрировать, то можно получить привычную форму записи закона Гука:

$$Kq = F, \quad (2.25)$$

где  $K = C^{-1}$  – жесткость пружины;  $C$  – податливость, т. е. величина, обратная жесткости;  $q = \int v dt$  – деформация пружины.

Учитывая свойства 0-узла, пружину можно представить графом, приведенным на рис. 2.13, *a*.

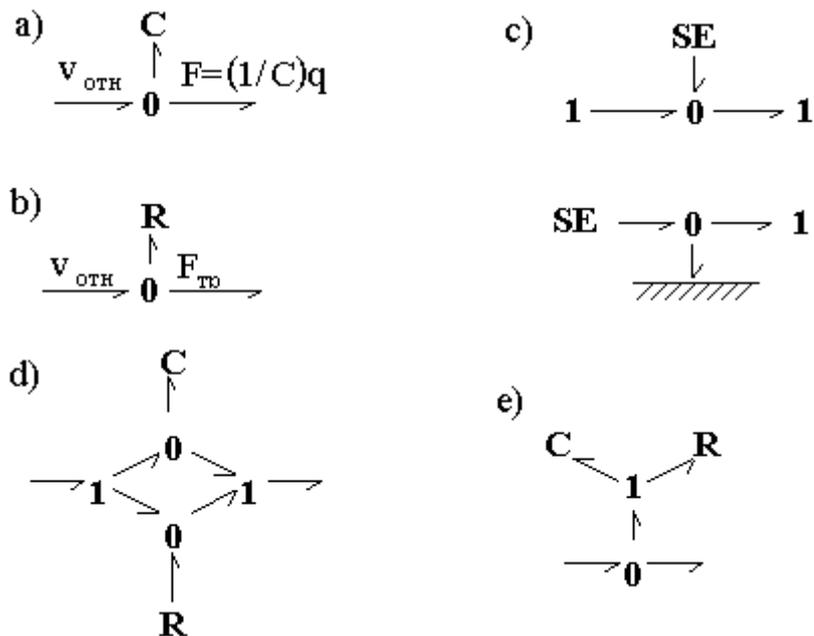


Рис. 2.13. Подвижное соединение твердых тел:  
*a* – идеальная пружина, *b* – источник механической энергии, *c* – демпфер,  
*d*, *e* – пружина с внутренним трением

Как уже отмечалось выше, 0-узел необходим и для моделирования трения между двумя движущимися относительно друг друга твердыми телами (рис. 2.13, *b*). Уравнение элемента потерь при вязком трении может иметь вид

$$R v_{отн} = F_{тр} \quad (2.26)$$

где  $R$  – коэффициент вязкого трения.

В общем случае зависимость силы трения от скорости может быть и нелинейной. Узел общего усилия с элементом потерь может отражать не только естественно существующее трение, но и специально вводимые в некоторые механизмы устройства: демпферы, амортизаторы.

Подобно упругости и трению, моделируются в механических системах источники энергии (рис. 2.13, *c*). В большинстве случаев источник механического движения, воздействуя на некоторое тело, одновременно создает равное, но противоположное по знаку усилие на свою опору.

В относительном движении могут одновременно проявляться несколько эффектов. Например, при моделировании реальных пружин иногда требуется учитывать потери энергии за счет внутреннего трения в материале пружины. Граф пружины с внутренним трением можно представить параллельно соединенными моделями идеальной пружины и демпфера (рис. 2.13, *d*) или эквивалентным графом, который приведен на рис. 2.13, *e*. Очевидно, что элементы  $C$ ,  $R$ ,  $SE$ ,  $SF$  могут соединяться подобным образом в любых сочетаниях, кроме одновременного использования  $SE$  и  $SF$ .

Рассмотрим моделирование поступательного движения трех вагонеток, из которых две, массой  $m_2$  и  $m_3$ , жестко связаны друг с другом, а между первой и второй вагонетками упруго-вязкая связь. В колесных парах присутствует трение  $R_1$ ,  $R_2$ ,  $R_3$ . Вагонетки приводятся в движение человеком, который прикладывает к первой вагонетке силу  $F$  (рис. 2.14, *a*).

Источник усилия  $SE$  в исходном графе (рис. 2.14, *b*) подключен через 0-узел. В результате человек прикладывает одно и то же усилие как к вагонетке, так и к опоре, в данном случае, например, к Земле. Величина силы трения в колесных парах пропорциональна разности между скоростью вагонетки и скоростью опоры.

Строго говоря, неподвижное основание тоже является твердым телом с очень большой массой и может быть представлено в графе 1-узлом с подключенной к нему инерционностью. Однако этот узел является узлом общего потока (скорости), принимаемого равным нулю. Поэтому связи с неподвижным основанием, а также все связи 1-узлов, соединенные с неподвижным основанием, имеют нулевую мощность и, следовательно, могут быть исключены из графа. Таким образом, граф, полученный после эквивалентных преобразований, приведен на рис. 2.14, *c*.

Свойство связей с неподвижным основанием в механических системах аналогично свойству связей с общей точкой (массой) в электрических системах. Различие состоит только в том, что в графе связей электрической системы исключается 0-узел (узел общего нулевого потенциала). Получить полную аналогию можно было бы, применяя при моделировании механических систем дуальную интерпретацию: считать силу потоком, а скорость – усилием. В этом случае инерционность  $I$  в графах заменяется емкостью  $C$ , 0-узлы – 1-узлами и наоборот. Од-

нако вряд ли достоинства такого способа интерпретации оправдывают появляющиеся терминологические неудобства.

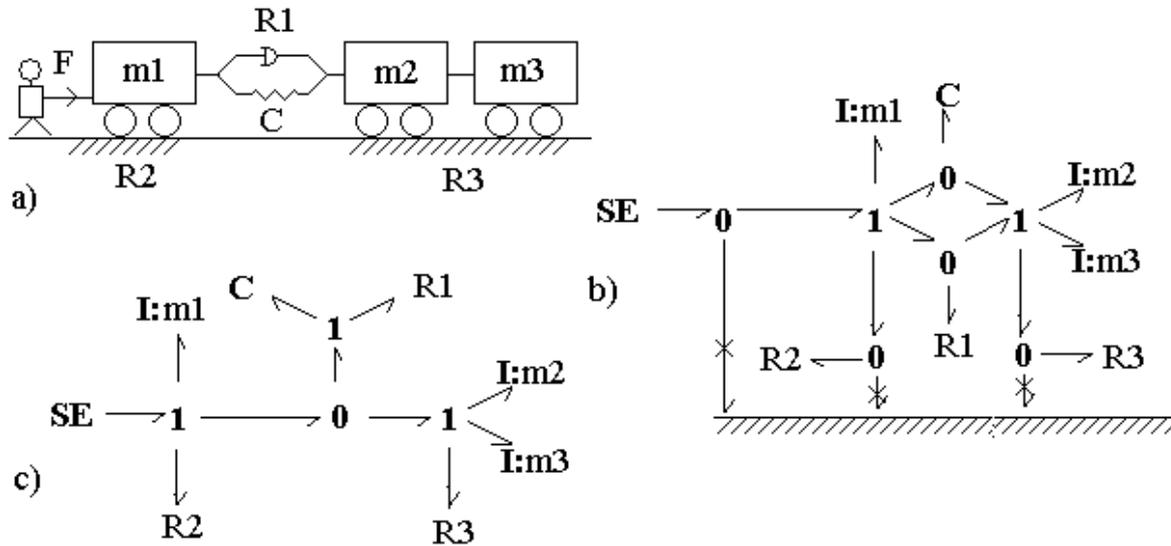


Рис. 2.14. Граф механической системы:

*a* – кинематическая схема, *b* – исходный граф, *c* – упрощенный граф

Приведенный на рис. 2.15 пример иллюстрирует моделирование вертикальных движений подвески автомобиля.

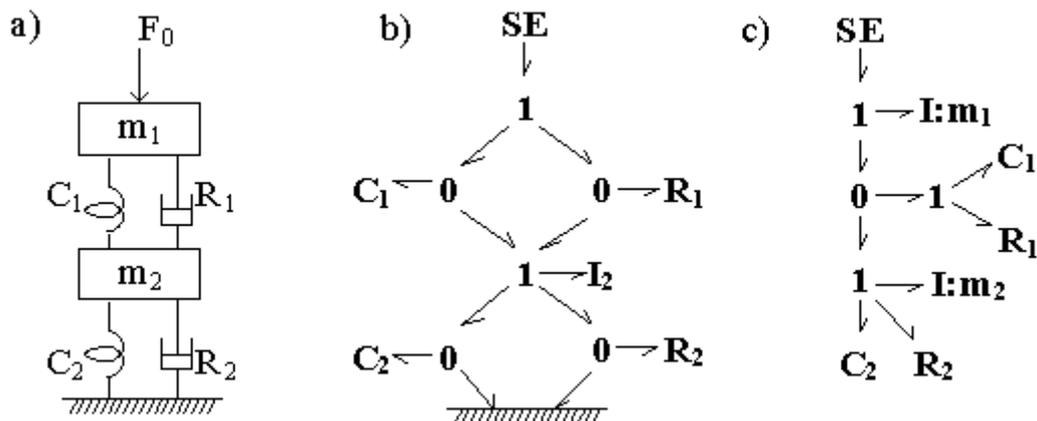


Рис. 2.15. Механическая система с поступательным перемещением:

*a* – кинематическая схема, *b* – исходный граф, *c* – упрощенный граф

Граф связей приведен на рис. 2.15, *b*. Здесь предполагается, что источник усилия движется вместе с массой  $m_1$ . Это может быть сила инерции, либо, например, реактивный двигатель. Как и в предыдущем примере, модель *b* включает неподвижное основание, а в модели *c* это основание исключено со всеми своими связями.

Еще один пример моделирования рычага представлен на рис. 2.16. Сила  $F$  действует на массу  $m_1$ , а та через рычаг приводит в движение

массу  $m_2$ . Обе массы движутся поступательно с трением. В данном случае рычаг моделируется с помощью элемента «Трансформатор».

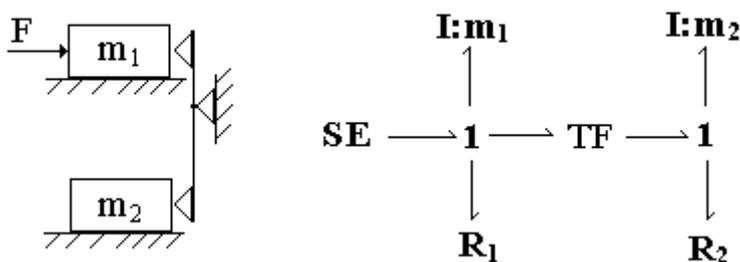


Рис. 2.16. Моделирование рычага:  
*a* – кинематическая схема, *b* – граф рычага

Рассмотренная методика моделирования одномерного поступательного движения механических систем может быть без труда распространена и на системы с вращательным движением.

В этом случае роль силы  $F$  играет момент силы  $M$ , линейной скорости  $v$  – угловая скорость  $\omega$ , массы  $m$  – момент инерции  $J$  и т. д. В уравнениях (2.21) – (2.26) изменятся только обозначения и размерности переменных и констант.

Приведенный на рис. 2.17 пример иллюстрирует построение графа связей для узла передачи вращательного движения, включающего одну ступень редуктора с зубчатыми колесами и упругие валы.

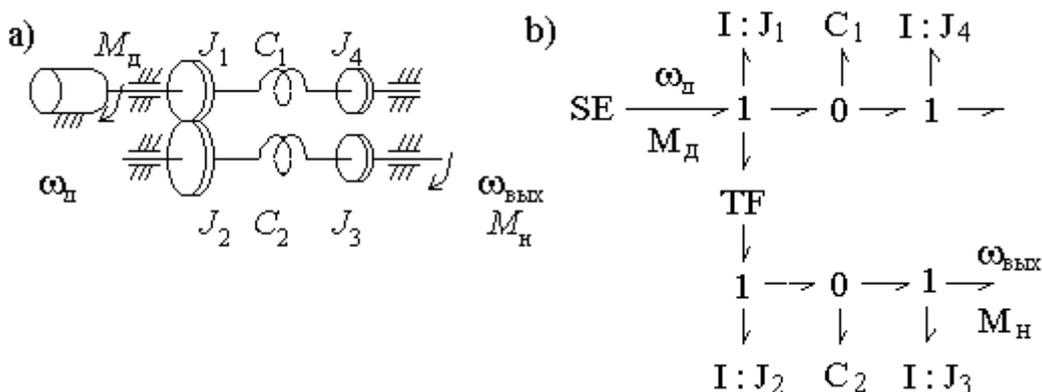


Рис. 2.17. Механическая вращающаяся система:  
*a* – схема, *b* – граф

Способ моделирования зубчатого соединения в рассмотренном примере справедлив, если основание неподвижно.

В случае, когда редуктор установлен на подвижном основании, как показано на рис. 2.18, *a*, граф связей должен учитывать угловую скорость основания  $\omega_0$ .

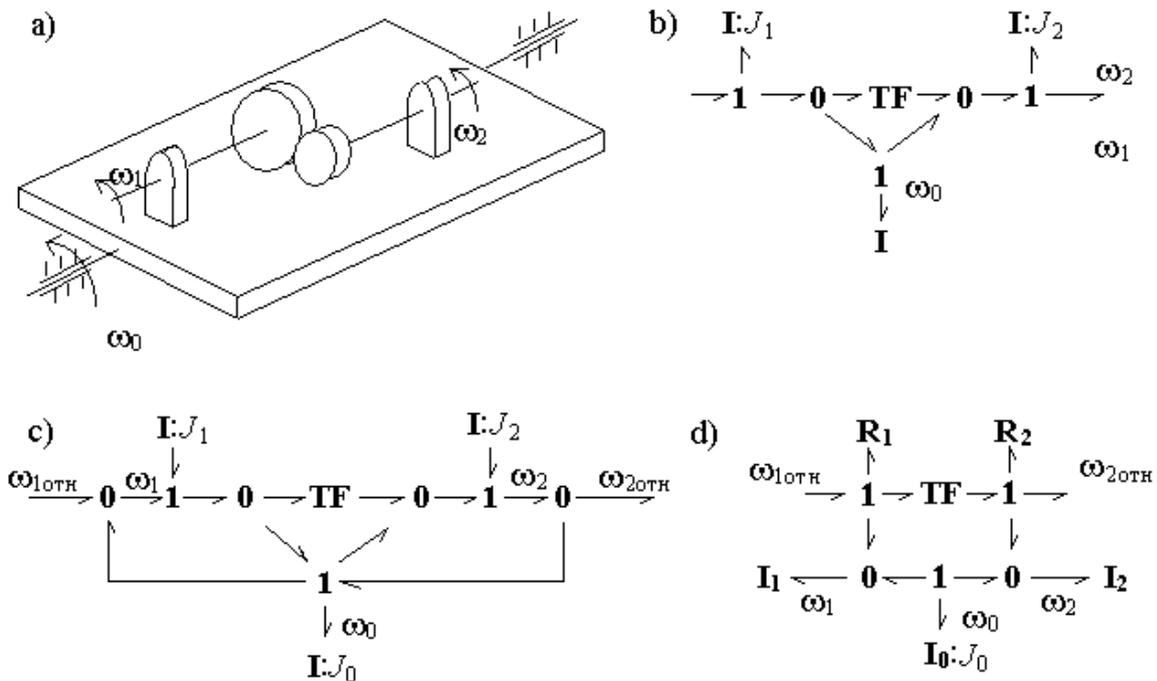


Рис. 2.18. Модель с подвижным основанием:  
*a* – схема, *b*, *c*, *d* – варианты графов

Граф на рис. 2.18, *b* показывает связь между абсолютными скоростями  $\omega_1$  и  $\omega_2$ , а граф на рис. 2.18, *c* – связь между скоростями колес зубчатого соединения  $\omega_{1отн}$ ,  $\omega_{2отн}$  относительно основания. Последний граф можно преобразовать к более простому виду, приведенному на рис. 2.18, *d*. Здесь дополнительно учтено также трение в опорах валов.

## 2.6. Моделирование электромеханических систем

Любая электромеханическая система с точки зрения преобразования энергии может быть представлена состоящей из трех частей: электрической Э, механической М и электромеханического преобразователя (ЭМП) (рис. 2.19). Построение математической модели электромеханической системы можно, таким образом, свести к детальному моделированию каждой из трех частей.



Рис. 2.19. Электромеханическая система

В качестве достаточно простого примера рассмотрим построение графа связей двигателя постоянного тока с независимым возбуждением.

В электрической части двигателя учтем индуктивное  $L_{я}$  и активное  $R_{я}$  сопротивления обмотки якоря, куда «уходит» часть входной электрической энергии. В графе связей это можно отобразить инерционностью  $I$  и элементом потерь, связанными в узле общего потока (тока).

В механической части учтем только инерционность ротора  $J$ . ЭМП будем считать идеальным, без потерь преобразующим электрическую энергию в механическую. Среди элементов графов связей роль идеального преобразователя могут выполнять только трансформатор и гиратор. Выбор из этих двух элементов определяется характером связи электрических и механических переменных. Если принять во внимание, что вращающий момент двигателя  $M_{дв}$  пропорционален току  $i_{я}$  в обмотке якоря, т. е. усилие в одной связи пропорционально потоку в другой связи, то выбор становится однозначным: ЭМП ведет себя как гиратор.

Построенный практически без формул граф связей двигателя постоянного тока приведен на рис. 2.20, *a*. Если для каждого 1-узла графа записать уравнения суммирования усилий, то получим:

$$L_{я} \frac{di_{я}}{dt} + R_{я} i_{я} = U_{я} - r \omega_{дв},$$

$$J \frac{d\omega}{dt} = r i_{я} - M_{дв}.$$
(2.27)

где  $r$  – коэффициент передачи гиратора.

Полученные уравнения для многих приложений достаточно точно описывают процессы, протекающие в двигателе постоянного тока [26].

Более точная и полная модель такого двигателя представлена на рис. 2.20, *b*. Здесь в механической части двигателя учтены неизбежные потери на трение, а в ЭМП – зависимость коэффициента  $r$  от магнитного потока  $\Phi$ , создаваемого током  $i_{в}$  в обмотке возбуждения:

$$r = k\Phi(i_{в}),$$
(2.28)

где  $k$  – конструктивный параметр, зависящий от количества пар полюсов и свойств якорной обмотки.

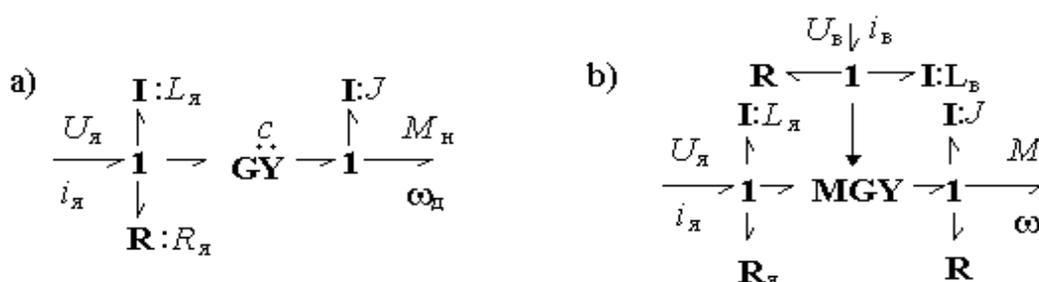


Рис. 2.20. Граф связей двигателя постоянного тока: *a* – с постоянным коэффициентом передачи гиратора, *b* – с учетом кривой намагничивания

В этой модели можно учесть также определяемую кривой намагничивания нелинейную зависимость индуктивности обмотки возбуждения  $L_B$  от тока  $i_B$ , как это показано на рис. 2.20, *b*.

Важная особенность построенных моделей состоит в том, что в них явно не определены входы и выходы, что позволяет применять их для моделирования любых электрических машин постоянного тока, работающих как в двигательном, так и в генераторном режимах.

## 2.7. Получение математической модели графов связей в форме системы уравнений

Самый простой способ построения математической модели проиллюстрируем на примере электрической схемы (рис. 2.21, *a*), граф которой приведен на рис. 2.21, *b*.

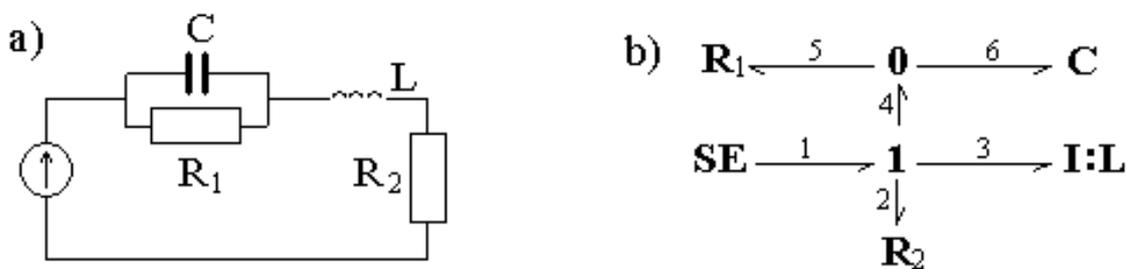


Рис. 2.21. Электрическая схема и ее граф

Для этого пронумеруем все связи в графе и, обозначая в связи с номером  $i$  поток и усилие как  $f_i$  и  $e_i$ , соответственно, запишем компонентные уравнения каждого из элементов:

$$\begin{aligned}
 SE: \quad e_1 &= U_{\text{вх}}, & 0: \quad e_4 &= e_5 = e_6, \\
 R_1: \quad R_1 f_5 - e_5 &= 0, & f_4 - f_5 - f_6 &= 0, \\
 C: \quad C p e_6 - f_6 &= 0, & 1: \quad f_1 &= f_2 = f_3 = f_4, \\
 I: \quad L p f_3 - e_3 &= 0, & e_1 - e_2 - e_3 - e_4 &= 0. \\
 R_2: \quad R_2 f_2 - e_2 &= 0,
 \end{aligned} \tag{2.29}$$

где  $p$  – оператор дифференцирования.

Примем начальные условия нулевыми. Тогда (2.29) удобнее рассматривать как систему операторных уравнений, где  $p$  – оператор Лапласа. В дальнейшем в данной главе будем придерживаться именно такой интерпретации символа  $p$ .

Полученные 12 уравнений с 12 неизвестными могут быть записаны в матричной форме:

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & R_1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Cp & -1 \\
0 & 0 & 0 & 0 & -1 & Lp & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & R_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & -1 \\
0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\
1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
e_1 \\
f_1 \\
e_2 \\
f_2 \\
e_3 \\
f_3 \\
e_4 \\
f_4 \\
e_5 \\
f_5 \\
e_6 \\
f_6
\end{bmatrix}
=
\begin{bmatrix}
U_{\text{BX}} \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix} \cdot (2.30)$$

Решение системы уравнений (2.29) или (2.30) позволяет найти аналитические выражения для изображений всех потоков и усилий в графе.

Следует отметить, что матричная форма математической модели (2.30) более удобна при численном формировании и решении систем уравнений на ЭВМ. При обычном «ручном» моделировании решение может быть получено методом подстановок в (2.29). Например, для падения напряжения на резисторе  $R_2$ , последовательно исключая в (2.29) все переменные, кроме  $e_2$ , получим

$$e_2(p) = \frac{R_2(R_1C_1p + 1)}{R_1LCp^2 + (L + R_1R_2C)p + R_1 + R_2} \cdot U_{\text{я}}(p).$$

## 2.8. Причинные отношения в графах связей

Одной из чрезвычайно интересных и полезных особенностей графа связей является возможность определения в нем вычислительной причинности. Чтобы пояснить суть этого термина, рассмотрим три формы записи одного и того же уравнения – закона Ома (элемента потерь графа связей):

$$U - Ri = 0; \quad (2.31)$$

$$U = Ri; \quad (2.32)$$

$$i = \frac{1}{R}U. \quad (2.33)$$

Формула (2.31) представляет собой неявную запись закона, говорящую о том, что между током и напряжением существует взаимно однозначное соответствие.

Уравнения (2.32) и (2.33) не только задают закон Ома, но и показывают, как вычислить одну физическую величину через другую. Тем самым, эти уравнения задают отношения причинности между переменными. В (2.32) причиной является ток  $i$ , а следствием – напряжение  $U$ . В (2.33), наоборот, напряжение  $U$  выступает как причина появления тока  $i$ . Заметим, что в реальной электронной схеме, как правило, не имеет смысла искать ответ на вопрос, что появляется раньше – напряжение или ток. Причинность может быть чаще всего только вычислительной, имеющей смысл при ММ.

Причинность, а вместе с ней и форму зависимости между усилиями и потоками можно определить в графе связей. Она задается так называемой причинной стрелкой – отрезком на одном из концов связи. На рис. 2.22 показаны два возможных варианта причинности на связи, соединяющей 1-узел и элемент потерь.

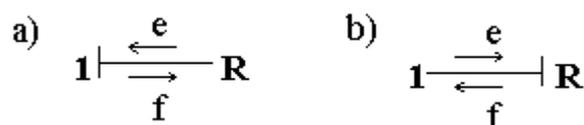


Рис. 2.22. Варианты причинности:

*a* – причинность по отношению к 1-узлу, *b* – причинность по отношению к  $R$

В первом варианте поток  $f$  является причиной, т. е. входной переменной элемента потерь, а усилие  $e$  является следствием, или выходной переменной элемента потерь. Это дополнительно иллюстрируется на рисунке стрелками, наглядно показывающими вход и выход элемента  $R$ . Первому варианту соответствует уравнение

$$e = R f .$$

Альтернативному варианту, представленному на рис. 2.22, *b*, соответствует уравнение

$$f = \frac{1}{R} e .$$

Здесь причиной для элемента  $R$  является усилие  $e$ , а следствием – поток  $f$ .

Заметим, что причинность на рис. 2.22 можно рассматривать по отношению не только к элементу потерь  $R$ , но и к 1-узлу. Тогда в первом варианте усилие  $e$  является причиной (входом) для 1-узла, а поток  $f$  – следствием (выходом) узла. Таким образом, каждая переменная является одновременно причиной (входом) для одного элемента и следствием (выходом) для другого элемента.

Будем называть связь причинной по отношению к некоторому элементу, если причинная стрелка определяет в качестве входа этого эле-

мента усилие  $e$ . Тогда связь на рис. 2.22,  $a$  можно назвать причинной по отношению к 1-узлу, а связь на рис. 2.22,  $b$  – причинной по отношению к элементу  $R$ .

Возможные варианты расстановки причинности на связях различных элементов представлены в табл. 2.4. Расстановка причинных отношений в графе подчиняется перечисленным ниже требованиям.

Таблица 2.4

№ п/п	Граф связей	Уравнения	Представление в операторно-структурной схеме
1	$SE \xrightarrow{e}$	$e = e(t)$	$\xrightarrow{e}$
2	$SF \xleftarrow{f}$	$f = f(t)$	$\xrightarrow{f}$
3	$\xrightarrow{e} R \xleftarrow{f}$	$e = R f$	$\xrightarrow{f} \boxed{R} \xrightarrow{e}$
4	$\xrightarrow{e} R \xleftarrow{f}$	$f = \frac{1}{R} e$	$\xrightarrow{e} \boxed{\frac{1}{R}} \xrightarrow{f}$
5	$\xrightarrow{e} C \xleftarrow{f}$	$e = \frac{1}{C} \int f dt$	$\xrightarrow{f} \boxed{\frac{1}{Cp}} \xrightarrow{e}$
6	$\xrightarrow{e} C \xleftarrow{f}$	$f = C \frac{de}{dt}$	$\xrightarrow{e} \boxed{Cp} \xrightarrow{f}$
7	$\xrightarrow{e} I \xleftarrow{f}$	$e = I \frac{df}{dt}$	$\xrightarrow{f} \boxed{Ip} \xrightarrow{e}$
8	$\xrightarrow{e} I \xleftarrow{f}$	$f = \frac{1}{I} \int e dt$	$\xrightarrow{e} \boxed{\frac{1}{Ip}} \xrightarrow{f}$
9	$\xrightarrow{e_1} TF \xleftarrow{e_2}$	$f_2 = m f_1$ $e_1 = m e_2$	$\xrightarrow{f_1} \boxed{m} \xrightarrow{f_2}$ $\xleftarrow{e_1} \boxed{m} \xleftarrow{e_2}$
10	$\xrightarrow{e_1} TF \xleftarrow{e_2}$	$e_2 = \frac{1}{m} e_1$ $f_1 = \frac{1}{m} f_2$	$\xleftarrow{f_1} \boxed{1/m} \xleftarrow{f_2}$ $\xrightarrow{e_1} \boxed{1/m} \xrightarrow{e_2}$
11	$\xrightarrow{e_1} GY \xleftarrow{e_2}$	$e_2 = r f_1$ $e_1 = r f_2$	$\xrightarrow{f_1} \boxed{r} \xrightarrow{f_2}$ $\xleftarrow{e_1} \boxed{r} \xleftarrow{e_2}$

№ п/п	Граф связей	Уравнения	Представление в операторно-структурной схеме
12		$f_2 = \frac{1}{r} e_1$ $f_1 = \frac{1}{r} e_2$	
13		$e_1 = e_2 = e_3$ $f_3 = f_1 - f_2$	
14		$f_1 = f_2 = f_3$ $e_3 = e_2 - e_1$	

1. Причинность в связях источников энергии определяется типом источника. Для источника потока связь всегда должна быть причинной по отношению к источнику, а для источника усилия, наоборот, – причинной по отношению к узлу, с которым связан источник усилия.

2. Аккумуляторы, подобно элементу потерь, могут иметь оба варианта причинности. Один из вариантов соответствует интегральной причинности, другой – дифференциальной, в соответствии с формой правой части уравнений аккумуляторов. Как это видно из табл. 2.4, интегральной причинности соответствует причинность по отношению к элементу для инерционности  $I$  и причинность по отношению к узлу для емкости  $C$ .

3. Гиратор  $GY$  и трансформатор  $TF$  тоже могут иметь два варианта задания причинности. При этом трансформатор сохраняет направление причинности, а гиратор меняет направление причинности на противоположное.

4. 0-узел может иметь одну и только одну причинную по отношению к 0-узлу связь. В противоположность этому все связи, кроме одной, должны быть причинными по отношению к 1-узлу.

Перечисленные правила позволяют расставить причинные отношения в любом графе связей, причем, как правило, несколькими способами. Можно рекомендовать следующую последовательность выполнения этой процедуры.

1. В первую очередь расставляются причинные отношения на связях источников энергии, поскольку они predeterminedены типами источников и не допускают свободы выбора.

2. Затем задаются причинности на связях аккумуляторов. Можно рекомендовать для всех аккумуляторов выбирать один тип причинности, например интегральный.

3. Последовательно, в соответствии с правилами, расставляются причинные отношения на остальных связях графа. Если на этом этапе появляется причинное противоречие, то можно вернуться к предыдущему пункту и изменить направление причинности у одного или нескольких аккумуляторов.

Один из вариантов расстановки причинных отношений в графе связей электрической схемы приведен на рис. 2.23. Здесь связь емкости  $C_1$  имеет дифференциальную причинность, а связи инерционности  $I$  и емкости  $C_2$  – интегральную. В этом графе невозможно одновременно обеспечить интегральную причинность для емкостей  $C_1$  и  $C_2$ .

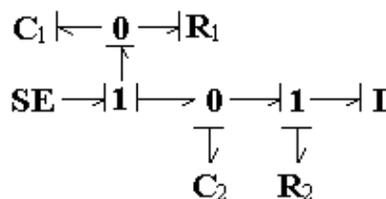


Рис. 2.23. Граф с расставленными причинными отношениями

Рис. 2.24 демонстрирует два варианта расстановки причинности для графа связей двигателя постоянного тока. В первом варианте выбраны интегральные, во втором – дифференциальные причинные отношения в связях инерционностей.

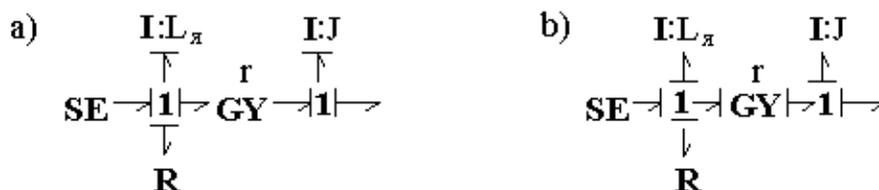


Рис. 2.24. Причинные отношения в графе связей двигателя постоянного тока:  
а – интегральные, б – дифференциальные

## 2.9. Построение операторно-структурных схем по графам связей

Построение операторно-структурных схем основано на том, что явные зависимости потоков и усилий для элементов графа связей, получаемые после расстановки отношений причинности, могут быть, в случае линейных систем, отображены в виде функциональных направленных звеньев с соответствующими передаточными функциями, как это показано в последней колонке табл. 2.3.

Трансформаторы и гираторы в операторно-структурной схеме представляются парами одинаковых звеньев, одно из которых передает сигнал в прямом направлении, другое – в обратном.

Каждый 0-узел представляется в операторно-структурной схеме точкой разветвления для усилий и сумматором для потоков. Каждый 1-узел представляется, наоборот, точкой разветвления для потоков и сумматором для усилий. Нетрудно заметить, что в каждом узле только одна связь, отличающаяся причинностью, соответствует сумме, остальные соответствуют слагаемым.

Знаки слагаемых зависят от направления полустрелок на связях. Если направление связи слагаемого совпадает с направлением связи суммы, то слагаемое входит в сумму со знаком «+», в противном случае – со знаком «-».

Процесс построения операторно-структурной схемы двигателя постоянного тока показан на рис. 2.25. Для большей наглядности на рис. 2.25, *b* сохранена форма операторно-структурной схемы, соответствующая форме графа связей. Перерисованная в более привычном виде, эта схема приведена на рис. 2.25, *c*.

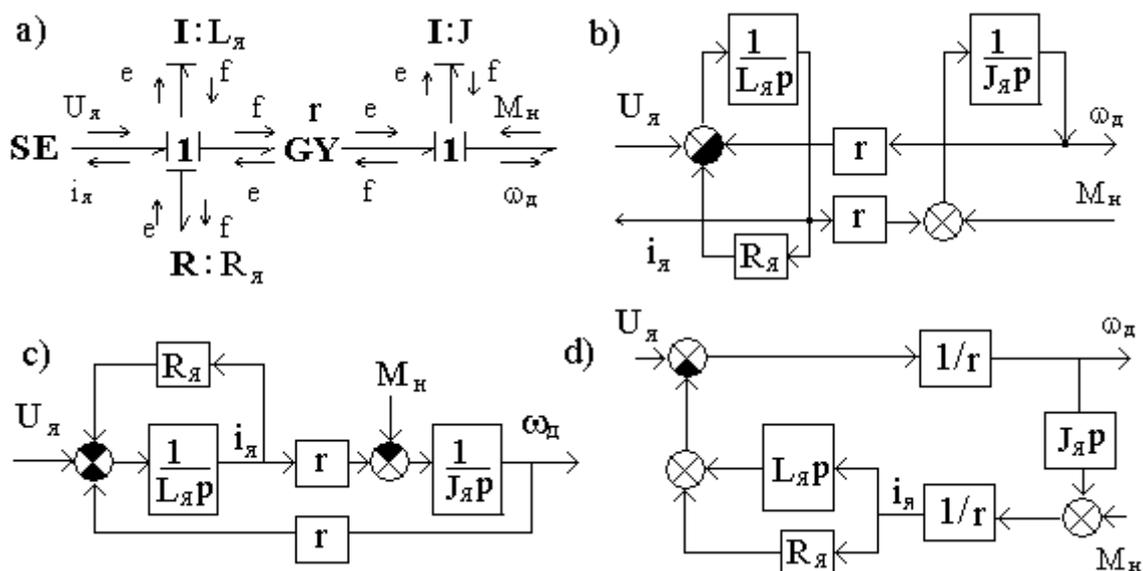


Рис. 2.25. Построение операторно-структурной схемы двигателя постоянного тока

Кроме этого, на рис. 2.25, *d* приведен и другой вариант операторно-структурной схемы, построенный по графу связей двигателя постоянного тока с дифференциальными причинностями (рис. 2.24, *b*). В соответствии с выбранной причинностью, в этой схеме, вместо интегрирующих, появились дифференцирующие звенья.

Возможность получения различных вариантов операторно-структурных схем является одним из достоинств графа связей.

Второй пример, проиллюстрированный на рис. 2.26, демонстрирует процедуру построения математической модели динамики механической многомассовой системы с упругими связями.

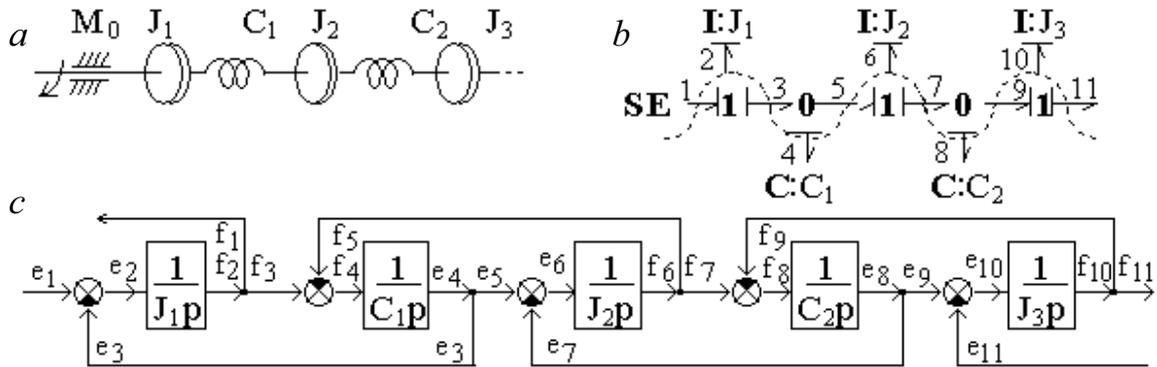


Рис. 2.26. Построение операторно-структурной схемы многомассовой механической системы с упругими связями: а – кинематическая схема, б – граф, с – структурная схема

Для того чтобы сразу получить удобную форму операторно-структурной схемы, можно предварительно проанализировать прямой путь прохождения сигнала в графе с расставленными причинными отношениями. Этот путь полностью задается направлениями причинности в узлах графа. Так, начинаясь с входного усилия в связи с номером 1, прямой путь может продолжиться только вдоль связи с номером 2, т. к. в связи 3 направление причинности противоположно. В третью связь прямой путь может прийти только из второй связи после его прохождения через инерционность. Таким образом, прямой путь прохождения сигнала в операторно-структурной схеме выглядит довольно извилистым в графе связей. После построения прямого пути в операторно-структурной схеме системы (рис. 2.26, с) остается только замкнуть обратные связи.

## 2.10. Применение правила циклов к графам связей

Правило циклов [24] позволяет для направленного графа или операторно-структурной схемы записать передаточную функцию между любыми ее входами и выходами. В соответствии с этим правилом передаточная функция графа определяется как

$$W = \frac{\sum_i P_i \Delta_i}{\Delta}, \quad (2.34)$$

где  $\Delta$  – определитель графа;  $P_i$  – передаточная функция  $i$ -го пути между заданными входом и выходом;  $\Delta_i$  – определитель сокращенного графа,

образующегося в результате исключения из исходного графа пути с передаточной функцией  $P_i$  и вершин, через которые этот путь проходит.

Определитель графа может быть записан следующим образом:

$$\Delta = 1 - \sum_j L_j^{(1)} + \sum_j L_j^{(2)} - \sum_j L_j^{(3)} + \dots, \quad (2.35)$$

где  $L_j^{(m)}$  –  $j$ -е произведение передаточных функций циклов для  $m$  циклов графа, взятых из множества независимых циклов. Сумма берется по всевозможным таким комбинациям.

Поясним некоторые из используемых терминов. **Циклом** называется замкнутый контур в графе или операторно-структурной схеме.

**Передаточная функция цикла** определяется как произведение передаточных функций всех звеньев, входящих в цикл.

**Независимыми** называются циклы, не касающиеся друг друга, т. е. не имеющие в операторно-структурной схеме общих точек.

В формуле (2.35) –  $L_j^{(1)}$  функция  $j$ -го контура,  $L_j^{(2)}$  – произведение передаточных функций двух не касающихся друг друга контуров,  $L_j^{(3)}$  – произведение передаточных функций трех взаимно не касающихся контуров и т. д.

Например, в операторно-структурной схеме двигателя постоянного тока, приведенной на рис. 2.27, есть два цикла  $L_1$  и  $L_2$  с передаточными функциями

$$L_1 = -\frac{R}{Lp}; \quad L_2 = -\frac{r^2}{LJp^2}.$$

Циклы касаются друг друга, т. к. имеют общий участок, включающий сумматор и звено с передаточной функцией  $1/Lp$ , поэтому определитель

$$\Delta = 1 - L_1 - L_2 = 1 + \frac{R}{Lp} + \frac{c^2}{LJp^2} = \frac{LJp^2 + JRp + c^2}{LJp^2}.$$

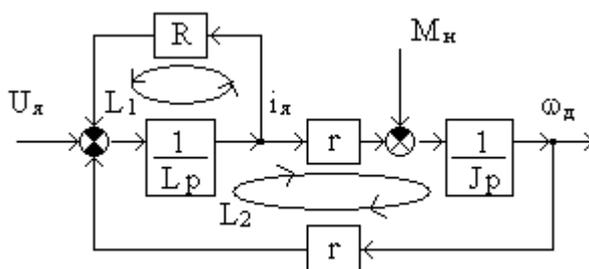


Рис. 2.27. Применение правила циклов к операторно-структурной схеме

Прямой путь от входного воздействия  $U_{я}$  к выходной величине  $\omega_{д}$  проходит через элементы с передаточными функциями

$$\frac{1}{Lp}; r; \frac{1}{Jp}.$$

Соответственно, передаточная функция этого пути равна

$$P_1 = \frac{r}{LJp^2}.$$

Этот путь касается обоих циклов, поэтому сокращенный граф циклов не имеет. Тогда  $\Delta_1 = 1$ , а передаточная функция двигателя определится как

$$W_u(p) = \frac{\omega_{д}(p)}{U_{я}(p)} = \frac{P_1 \Delta_1}{\Delta} = \frac{r}{LJp^2 + RJp + r^2}.$$

Путь от возмущающего момента нагрузки  $M_{н}$  определяется выражением

$$P_2 = -\frac{1}{Jp}.$$

Этот путь не касается цикла  $L_1$ , поэтому определитель сокращенного графа

$$\Delta_2 = 1 - L_1 = 1 + \frac{R}{Lp} = \frac{Lp + R}{Lp},$$

а передаточная функция двигателя по возмущению

$$W_m(p) = \frac{\Delta_2 P_2}{\Delta} = -\frac{Lp + R}{LJp^2 + RJp + r^2}.$$

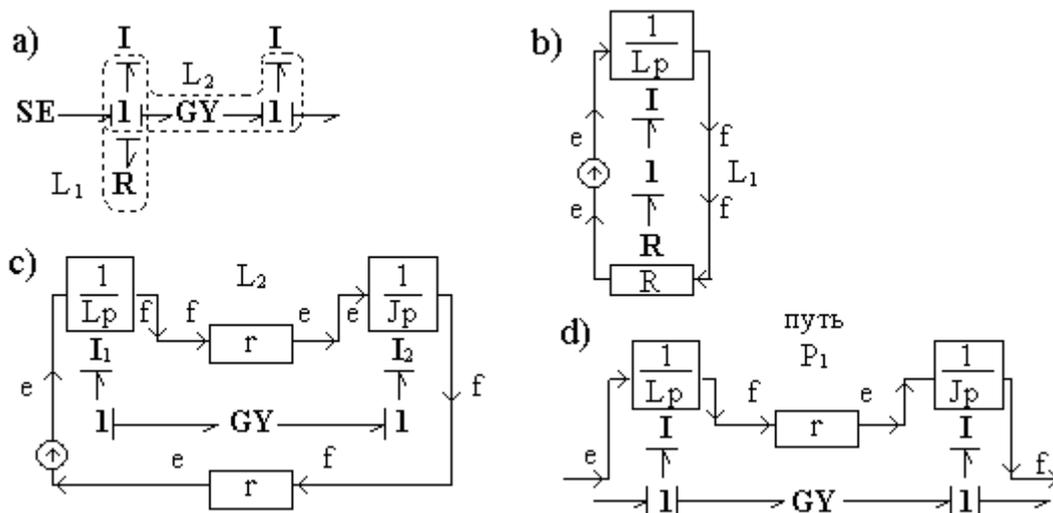


Рис. 2.28. Пути и циклы в графе связей

Вся информация, необходимая для расчета передаточной функции, есть, очевидно, и в графе связей, т. к. из него можно получить операторно-структурную схему, к тому же в различных вариантах. Рис. 2.28, *a* иллюстрирует поиск пути  $P_1$  и циклов  $L_1$  и  $L_2$  в графе связей рассмотренной выше модели двигателя.

Путь в графе связей проходит вдоль связей, не меняющих направления причинности в узлах графа.

Изменение причинности (т. е. изменение усилия на поток и обратно) может происходить только в односвязных элементах ( $R, I, C$ ) и в гираторе.

Циклы в графе связей, как это показано на рис. 2.28, *c*, образуются цепочками связей, сохраняющими направление причинности и заканчивающимися на обоих концах односвязными элементами  $R, I, C$ .

Отметим, что источники энергии в циклы входить не могут. Как это показано на рис. 2.29, цикл может включать последовательность 0-узлов и 1-узлов (рис. 2.29, *a*), трансформаторы (рис. 2.29, *b*) и гираторы (рис. 2.29, *c*).

Передаточные функции циклов на рис. 2.29, *a, b, c* имеют вид, соответственно:

$$L_1 = \frac{-1}{CLp^2}, \quad L_2 = \frac{-Im^2p}{R}, \quad L_3 = \frac{-r^2}{RIp}.$$

Коэффициенты передачи трансформаторов и гираторов входят в передаточную функцию цикла в квадрате, поскольку цикл проходит через них дважды: один раз в прямом направлении, другой раз – в обратном.

Циклы, образуемые цепочками связей, называются **плоскими циклами** (рис. 2.29).

$$a) \quad I \leftarrow 1 \mid 0 \mid 1 \mid 0 \rightarrow C$$

$$b) \quad R \leftarrow 1 \mid \rightarrow TF \mid \rightarrow 0 \mid \rightarrow 1 \mid \rightarrow I$$

$$c) \quad R \leftarrow 1 \mid \leftarrow GY \leftarrow \mid 1 \mid \rightarrow I$$

*Рис. 2.29. Примеры плоских циклов*

Рассмотрим решение задачи расчета передаточной функции механизма с редуктором, граф которого приведен на рис. 2.30.

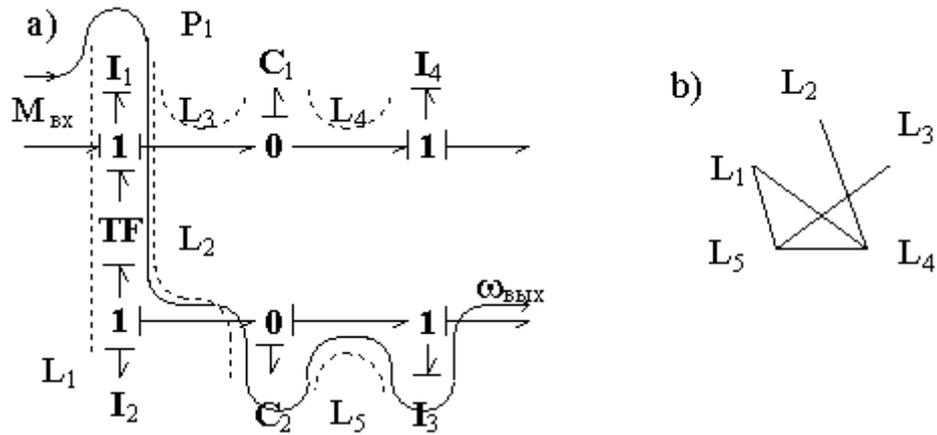


Рис. 2.30. Циклы в графе связей:  
*a* – граф связей, *b* – вспомогательный граф

Передаточная функция единственного прямого пути  $P_1$ , проходящего последовательно через инерционность  $I_1$ , трансформатор  $TF$ , емкость  $C_2$  и инерционность  $I_3$ , определяется произведением передаточных коэффициентов перечисленных элементов:

$$P_1 = \frac{T}{I_1 I_3 C_2 p^3}.$$

Граф содержит пять циклов, отмеченных штриховыми линиями в графе. Передаточные функции циклов:

$$L_1 = -\frac{T^2 I_2 p}{I_1 p}; \quad L_4 = -\frac{1}{I_4 C_1 p^2};$$

$$L_2 = -\frac{T^2}{I_1 C_2 p^2}; \quad L_5 = -\frac{1}{I_3 C_2 p^2}.$$

$$L_3 = -\frac{1}{I_1 C_1 p^2};$$

Для того чтобы найти все пары, тройки и т. д. не касающихся циклов, удобно построить вспомогательный граф (рис. 2.30, *b*), в котором каждая вершина соответствует одному из циклов, а дуга между вершинами проводится, если циклы не касаются.

Каждая дуга в этом графе соответствует паре не касающихся циклов. Таких пар пять:  $L_1, L_4$ ;  $L_1, L_5$ ;  $L_2, L_4$ ;  $L_3, L_5$ ;  $L_4, L_5$ .

Вспомогательный граф наглядно показывает также тройку независимых циклов  $L_1, L_4, L_5$ , которая образует треугольник. Четверок независимых циклов, которые образовали бы четырехугольник, здесь нет. Таким образом, определитель графа связей можно записать как

$$\Delta = 1 - L_1 - L_2 - L_3 - L_4 - L_5 + L_1L_4 + L_1L_5 + L_2L_4 + \\ + L_3L_5 + L_4L_5 - L_1L_4L_5.$$

Путь  $P_1$  не касается только цикла  $L_4$ , поэтому  $\Delta_1 = 1 - L_4$ , а передаточная функция системы имеет вид

$$W = \frac{P_1 \cdot \Delta_1}{\Delta}.$$

После необходимых подстановок получим

$$W = \frac{T(I_4C_1p^2 + 1)}{\left( (I_1 + T^2I_2)I_3I_4C_1C_2p^5 + \left[ C_1I_4(T^2I_3 + I_1 + T^2I_2) + C_2I_3(I_1 + I_4 + T^2I_2) \right] p^3 \right. \\ \left. \overline{(I_1 + I_4 + T^2I_2 + T^2I_3)} p \right)}.$$

При использовании правила циклов необходимо учитывать, что знак передаточной функции цикла в графе связей всегда отрицательный. Это следует из того, что полустрелки на концах цепочки связей в цикле, как это видно из рис. 2.29, всегда направлены в противоположные стороны. Для определения знака передаточной функции пути тоже не обязательно просматривать все изменения знака в цепочке связей, достаточно сравнить направления полустрелок в начале и в конце пути.

## 2.11. Общие принципы графического представления мехатронных систем в пакетах автоматизированного моделирования

Аппарат графов связей является хорошим инструментом для аналитического моделирования, для получения математических моделей систем и объектов. Однако использование элементов графов связей в автоматизированном моделировании имеет ряд недостатков. Во-первых, структура в виде графа связей является слишком детальной при описании сложных систем, например механических объектов в пространственном движении. Модель становится необозримой и сложной для восприятия. Во-вторых, аппарат графов связей непривычен для специалистов предметных областей. Более предпочтительными являются электрическая схема (при исследовании электрических систем) или кинематическая механическая цепь (при исследовании пространственных механизмов). Эти сложности заставляют искать в рамках компонентного моделирования другие формы задания графической информации об объекте.

Проблемы моделирования систем с элементами различной физической природы неоднократно поднимались и рассматривались рядом исследователей. В России заслуживает внимания группа авторов, опубликовавших ряд работ по теоретическим проблемам моделирования слож-

ных физически неоднородных систем и реализовавших свои идеи в виде достаточно эффективного для своего времени пакета прикладных программ [1, 6]. Эти работы тем более заслуживают внимания, что многие изложенные в них идеи явно просматриваются в современных пакетах визуального моделирования.

Практически в разработанном пакете присутствовали все главные особенности пакетов автоматизированного моделирования, которые допускал уровень развития технических средств, а именно графическое представление исходной информации о моделируемой системе, использование библиотек моделей компонентов, использование компонентов как с направленными, так и с ненаправленными связями, использование информационных и энергетических связей и т. д.

Рассмотрим вкратце основные идеи упомянутых работ и сравним их с возможностями современных пакетов визуального моделирования.

В работе [1] в качестве графической формы модели введена так называемая **формализованная схема**, являющаяся некоторым обобщением других типов схем: операторно-структурной, принципиальной, кинематической, графа связей. В рамках такой схемы каждая часть моделируемой системы представляется наиболее удобным для нее способом.

Формализованная схема может включать типовые элементы с двумя типами связей. Связи первого типа называются **информационными**. Такие связи отражают передачу сигналов или информации в системе и полностью соответствуют связям, используемым при построении функциональных и операторно-структурных схем. Направление передачи сигнала отображается на информационной связи стрелкой. В зависимости от направления стрелки информационная связь может быть входом или выходом элемента.

Связи второго типа отражают передачу энергии в системе и называются **энергетическими**. Такие связи используются при моделировании физических объектов, в частности электрических схем, исполнительных механизмов и приводов. Эти связи подобны связям в графах связей.

Формализованная схема определена как произвольная совокупность элементов, внешние связи которых соединяются в точках, называемых узлами схемы. Если пронумеровать узлы схемы числами от 0 до  $n$ , то совокупности узлов можно поставить в соответствие вектор переменных

$$e = (e_0, e_1, \dots, e_n)^T,$$

называемый по аналогии с электрическими цепями **вектором потенциалных переменных**.

Энергетические связи в формализованной схеме называются **ветвями схемы**. Совокупности ветвей, пронумерованных в схеме числами от 1 до  $m$ , ставится в соответствие вектор потоковых переменных

$$f = (f_1, f_2, \dots, f_m).$$

Для потоковых переменных выполняется правило: их алгебраическая сумма в каждом узле схемы равна нулю. Таким образом, потоковые переменные – это аналог токов в электрических цепях.

Однако, в отличие от электрических цепей, потенциальные и потоковые переменные могут быть не только скалярными, но и векторами произвольной размерности.

Частным случаем формализованной схемы является **операторно-структурная схема**. Она строится с использованием типовых звеньев, соединенных информационными направленными связями. В операторно-структурной схеме присутствуют только потенциальные переменные, связанные с узлами схемы. Потоковых переменных в операторно-структурной схеме нет. На рис. 2.31 представлена операторно-структурная схема в нотации пакета REMOS [6].

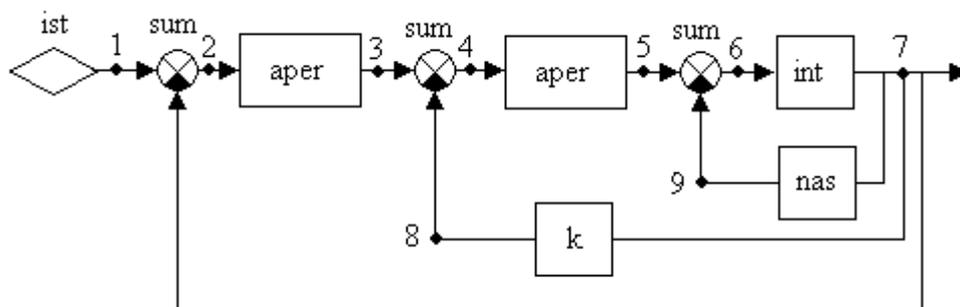


Рис. 2.31. Операторно-структурная схема в нотации пакета REMOS

Данная схема описывается вектором переменных схемы

$$e = (e_1, e_2, \dots, e_9).$$

Из них переменные  $e_1$  и  $e_7$  – переменные входа и выхода.

На рис. 2.32 аналогичная схема представлена в нотации MATLAB/Simulink. Современный графический интерфейс пакета Simulink позволил отказаться от нумерации связей в операторно-структурной схеме.

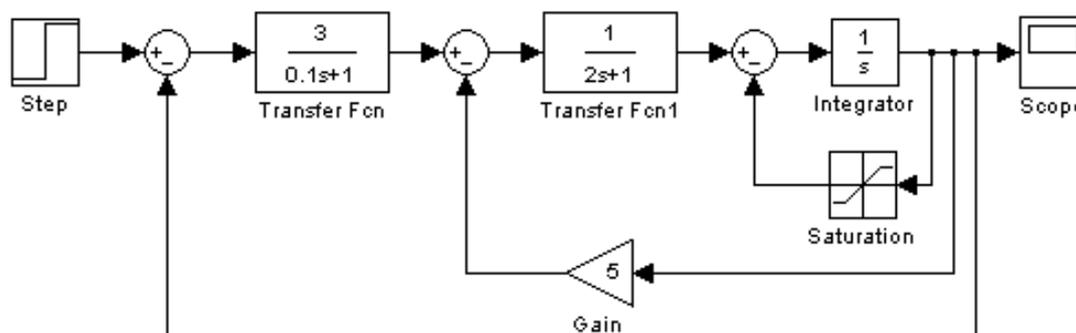


Рис. 2.32. Операторно-структурная схема в нотации MATLAB/Simulink

Приведенные определения позволяют наиболее естественным образом представлять структуру объектов различной физической природы и их систем управления. Например, электрическая цепь может быть представлена своей принципиальной схемой, построенной из типовых элементов (резисторов, емкостей, транзисторов и др.). Для примера на рис. 2.33 приведена простая электрическая схема с энергетическими связями. В ней узлам с номерами 0, 1, 2 соответствуют потенциальные переменные – электрические потенциалы  $u_0, u_1, u_2$ . Ветвям, номера которых отмечены в скобках, соответствуют потоковые переменные – электрические токи  $i_1, i_2, i_3, i_4$ .

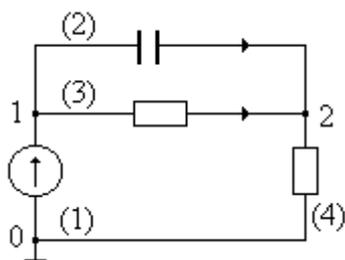


Рис. 2.33. Простая электрическая схема с энергетическими связями

При моделировании механических объектов удобно использовать подход, эквивалентный методу графов связей, с учетом некоторых изменений в принятой терминологии и в условных обозначениях элементов.

Потоковыми переменными в механических системах удобно считать силы и моменты сил, а потенциальными – линейные и угловые скорости. В этом случае 1-узел графа связей может изображаться в схеме просто точкой соединения связей, т. е. узлом схемы, и жесткое соединение твердых тел (рис. 2.11) отображается в схеме более наглядно – просто соединением связей элементов в узле схемы.

Подвижное соединение твердых тел, которое на графе связей отображается 0-узлом (рис. 2.12), в схеме отображается элементом, называемым одномерным кинематическим 1-узлом. В названии элемента отражается свойство узла предоставлять одну степень свободы в относительном движении.

С учетом введенных изменений любой граф связей может быть изображен в виде схемы с использованием соответствующих элементов. Для иллюстрации на рис. 2.34 приведена схема механической вращательной системы, соответствующая графу связей, построенному на рис. 2.17. В этой схеме пронумерованы узлы, которым соответствуют угловые скорости  $\omega_1 \dots \omega_6$ , и ветви (номера ветвей даны в скобках), которым соответствуют вращающие моменты  $M_1 \dots M_{11}$ .

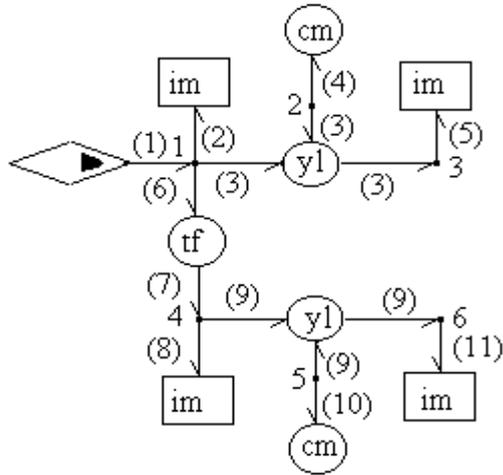


Рис. 2.34. Схема механической вращательной системы

Достоинством формализованной схемы является возможность использования моделей механических элементов с векторными энергетическими связями в качестве компонентов для моделирования пространственного движения механизмов. Такие элементы могут быть построены, в частности, с использованием аппарата графов связей. Примеры формирования сложных компонентов механических цепей рассмотрены в работах [1, 6].

На рис. 2.35 приведена модель вращательной кинематической пары, ось вращения которой параллельна осям  $O_1z_1$  и  $O_2z_2$ , в результате чего все звенья движутся в плоскости  $Oxy$ .

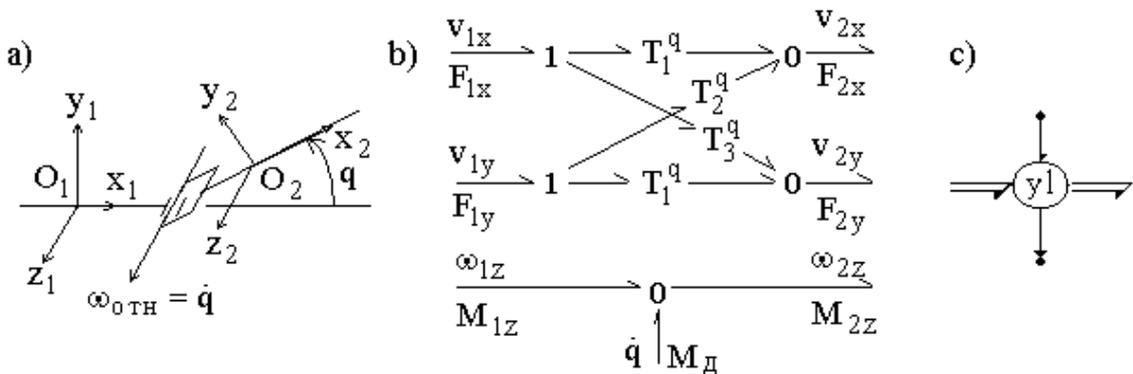


Рис. 2.35. Модель вращательной кинематической пары: а – кинематическая схема, б – граф связей, с – компонент формализованной схемы

Такому соединению соответствуют следующие уравнения:

$$\omega_2 = A\omega_1 + \omega_0; M_2 = A M_1; v_2 = A v_1; F_2 = A F_1,$$

где  $A$  – матрица поворота относительно оси  $Ox$ ,  $\omega_0 = (0, 0, \omega_{0TH})^T$  – вектор угловой скорости для вращения относительно оси  $Oz$ ,  $F_1 = (F_{1x}, F_{1y})$  и  $F_2 = (F_{2x}, F_{2y})$  – векторы силы в плоском движении,

$v_1 = (v_{1x}, v_{1y})$  и  $v_2 = (v_{2x}, v_{2y})$  – векторы линейной скорости в плоском движении.

Кроме этого, как следует из графа, момент, приложенный в шарнире:

$$M_d = M_{1z}.$$

В этом графе коэффициенты передачи трансформаторов определяются выражениями:

$$T_1^q = \cos q, T_2^q = -T_3^q = \sin q,$$

где угол поворота  $q$  определяется из уравнения

$$\dot{q} = \omega_{\text{отн}}.$$

Подобным же образом можно получить и представить математические модели твердых тел.

Моделей трех элементов – звена, шарнира и опоры – достаточно для графического представления кинематической схемы пространственного механизма. С использованием этих элементов схема механизма строится просто как последовательное соединение звеньев и шарниров (рис. 2.35). Компонент «основание» необходим в схеме для задания направления силы тяжести.

Приведенная на рис. 2.36 схема двухзвенного манипулятора в нотации пакета REMOS – векторная. В ней каждому из узлов 1–5 соответствует векторная потенциальная переменная

$$e_i = (v_{ix}, v_{iy}, v_{iz}, \omega_{ix}, \omega_{iy}, \omega_{iz}, a_{ix}, a_{iy}, a_{iz})^T, \quad i = \overline{1, 5},$$

включающая проекции векторов линейной и угловой скоростей соответствующей точки механизма на связанные оси, а также проекции единичного вектора силы тяжести.

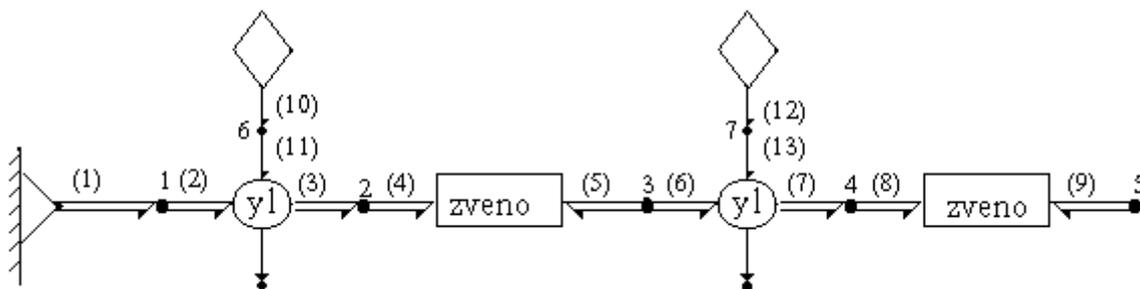


Рис. 2.36. Схема двухзвенного манипулятора

Каждой из ветвей с номерами 1–9 соответствует векторная потоковая переменная

$$f_i = (F_{ix}, F_{iy}, F_{iz}, M_{ix}, M_{iy}, M_{iz})^T, \quad i = \overline{1, 11},$$

включающая проекции векторов силы и вращающего момента реакции связи. Узлам с номерами 6, 7 соответствуют скалярные переменные

$\omega_6, \omega_7$  – относительные скорости вращения в шарнирах, а узлам 8, 9 – углы поворота в первом и втором шарнирах, соответственно. Ветвям 10–13 соответствуют скалярные потоковые переменные – вращающие моменты приводов.

Пример моделирования системы с использованием энергетических и структурных компонентов приведен на рис. 2.37. Рассмотрена схема системы управления двухзвенного манипулятора, в которой по каждой из степеней подвижности реализован простейший закон формирования управляющего момента:

$$M = K_1(q_{\text{зад}} - q) - K_2 \dot{q},$$

где  $q_{\text{зад}}$  – заданное значение угла поворота в сочленении (переменные  $x_{10}, x_{11}$  в схеме);  $q$  – угол поворота (переменные  $x_8, x_9$ ).

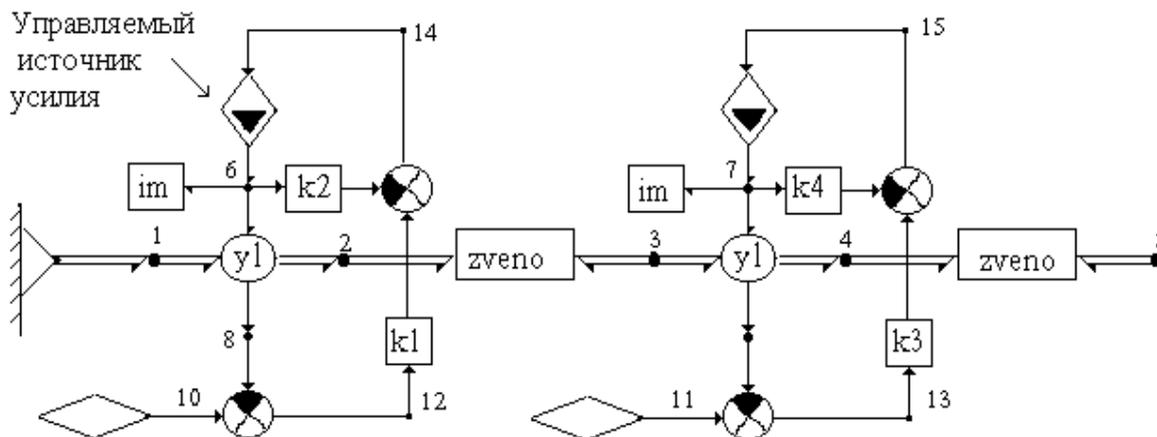


Рис. 2.37. Схема двухзвенного управляемого манипулятора

В этой схеме дополнительно компонентами  $im$  учтены моменты инерции механической части привода.

Особенностью схемы являются элементы, названные в [1, 6] управляемыми источниками. Их роль – преобразовать информационную переменную в энергетическую, в управляющий момент, приложенный в шарнире.

В современных пакетах автоматизированного моделирования механических цепей те детали, которые ранее отражались на схеме (в частности, номера узлов и ветвей) или неявно задавались в модельном соглашении (например, порядок расположения переменных в векторах связей), определяются самой формализованной схемой.

Однако общие принципы представления систем, содержащих энергетические и информационные элементы, во многом сохранились. Например, в приведенной на рис. 2.38 модели того же самого двухзвенного

манипулятора в нотации пакета SimMechanics верхняя часть схемы представляет собой кинематическую цепь, включающую основание, два вращательных кинематических узла и два твердых тела.

В схеме присутствуют порты для соединения физических элементов, помеченные символами  $B$  и  $F$  на вращательных кинематических узлах, и информационные порты, служащие для соединения энергетической и сигнальной части. Блок привода играет ту же роль, что и управляемый источник на рис. 2.36. Схема управления вторым приводом свернута в подсистему.

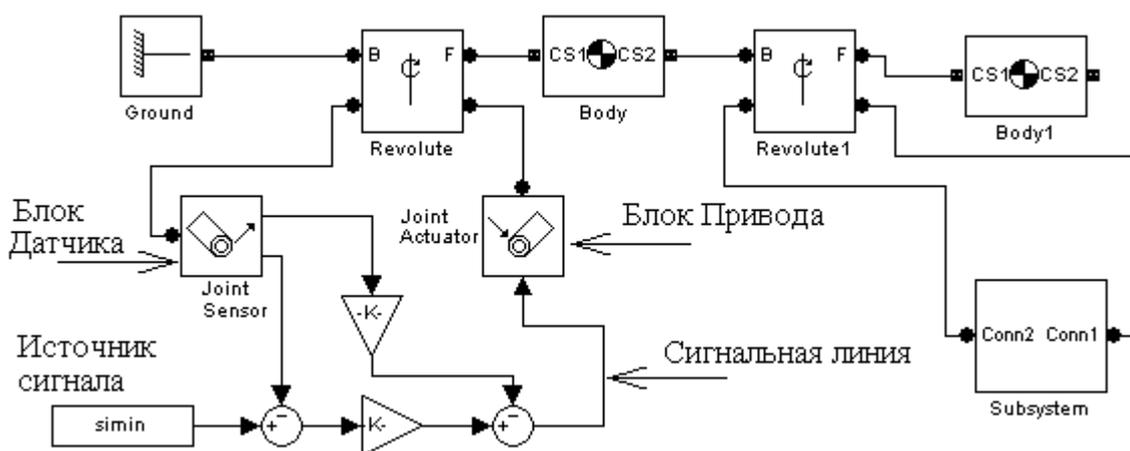


Рис. 2.38. Модель двухзвенного манипулятора в нотации пакета SimMechanics

Более подробно характеристики современных пакетов приведены в главе 5.

## Глава 3

# ИССЛЕДОВАНИЕ МЕХАТРОННЫХ СИСТЕМ ВО ВРЕМЕННОЙ ОБЛАСТИ

### 3.1. Механизмы продвижения модельного времени

Реальные мехатронные объекты являются динамическими системами. Они функционируют во времени, и ход времени необходимо моделировать так же, как и изменения всех остальных переменных.

Любой процесс моделирования на ЭВМ представляет собой взаимодействие трех видов времени:

- реального времени, к моментам которого привязаны события, происходящие в моделируемой системе;
- модельного времени, отсчитываемого программой моделирования и являющегося моделью реального времени;
- машинного времени, в котором функционирует аппаратная часть системы моделирования.

В процедурах моделирования наиболее важно управление модельным временем.

Процессы, протекающие в моделях, должны адекватно отображать поведение моделируемых объектов: если события в реальной системе совпадают, то они должны совпадать и в модели, если реальные события следуют в определенном порядке, то он не должен нарушаться и в модели.

Особенно большую роль играет правильная организация взаимодействия реального и модельного времени в процедурах имитационного моделирования, т. к. в нем очень важны причинно-следственные отношения.

В имитационных моделях возможны следующие варианты течения модельного времени:

- модельное время может течь независимо от процессов в системе, как течет реальное время;
- модельное время может изменяться скачками (такой режим является идеализацией реальных процессов, цель которой – убрать из рассмотрения «пустые» периоды, когда в модели не происходят изменения);
- модельное время может многократно проходить один и тот же интервал, если в однопроцессорной машине необходимо в режиме имитационного моделирования вести параллельные расчеты.

Любой процесс в динамической системе можно рассматривать как изменение ее состояния, которое может происходить более или менее равномерно либо в форме резких изменений, связанных с появлением событий. При компьютерном моделировании на ЦВМ модельное время может меняться только дискретно, с некоторым шагом  $h$ . Необходимо согласовать процесс выбора шага и процесс продвижения модельного времени с особенностями процессов в реальном объекте.

Существуют два основных способа продвижения модельного времени: «принцип  $\Delta t$ » и «принцип  $\Delta z$ » [31].

**Принцип  $\Delta t$**  довольно прост. Модельное время течет малыми шагами  $\Delta t$  и может принимать только дискретные значения, кратные этому временному интервалу. Величина шага связана с динамическими особенностями моделируемого объекта. Она может меняться в процессе моделирования, однако напрямую не привязана к событиям, происходящим в моделируемой системе.

Это приводит к тому, что события, обычно связанные с выполнением некоторых условий, могут попасть внутрь временного шага. В результате события могут сдвигаться во времени, а также могут нарушаться причинно-следственные связи между событиями.

Обычно события привязываются к правой границе временных интервалов  $\Delta t$ . На рис. 3.1 все события в модельном времени сдвинуты на конец такта. Кроме того, события  $E_2, E_3$ , которые в реальном времени появляются последовательно, причем  $E_2$  является причиной  $E_3$ , в модели выглядят одновременными.

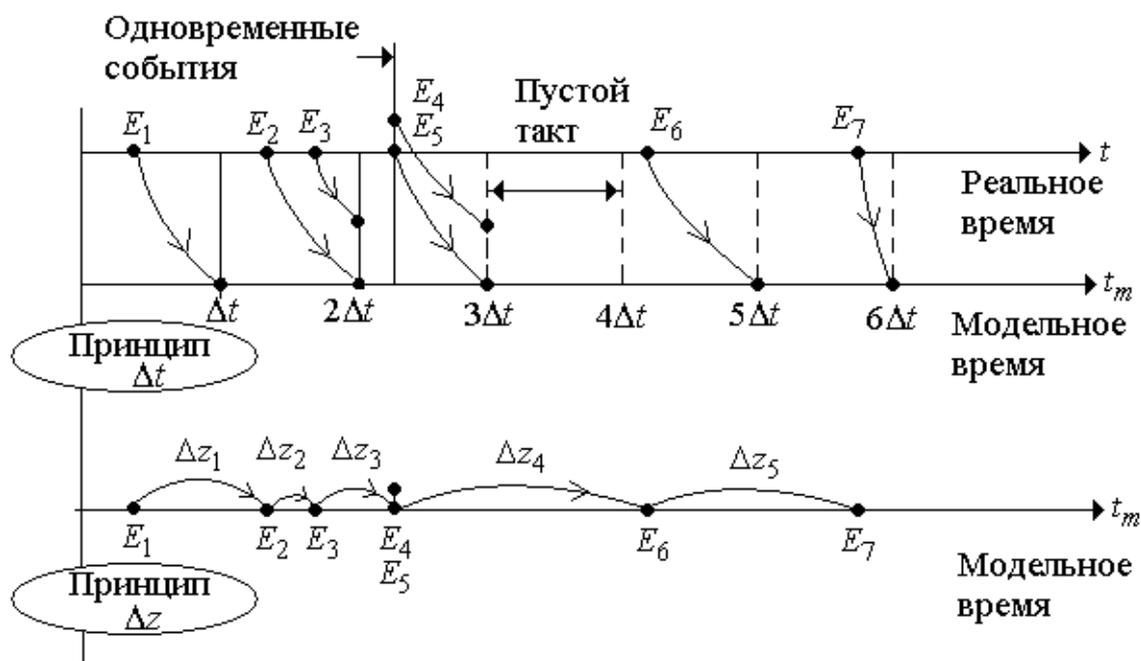


Рис. 3.1. Механизмы продвижения модельного времени

Во многих случаях, при малых значениях временного шага, это не играет существенной роли. В других – может привести к ошибкам моделирования.

Метод  $\Delta t$  целесообразно использовать в следующих случаях:

- если моделируется непрерывная система, процессы в которой представляют собой непрерывную цепь равнозначных событий;
- если в моделируемой системе моменты появления событий связаны выполнением некоторых условий, касающихся значений переменных системы, в результате чего эти моменты невозможно заранее определить.

Для мехатронных систем реализация принципа  $\Delta t$  является основным способом продвижения модельного времени, т. к. основу мехатронных систем составляют механические устройства, обладающие непрерывной динамикой.

**Принцип  $\Delta z$** , называемый также принципом особых состояний, предполагает, что продвижение модельного времени обусловлено событиями, происходящими в моделируемой системе. Как и в первом случае, модельное время меняется дискретно на величину  $\Delta z$ , однако эта величина не привязана к динамическим характеристикам объекта, а представляет собой временной интервал между последовательными событиями в системе. Величина  $\Delta z$  может иметь произвольное значение, в т. ч. быть равной нулю, если интервал между событиями пренебрежимо мал.

Необходимым условием реализации моделирования по принципу  $\Delta z$  является разработка специальной процедуры планирования событий – так называемого **календаря событий** [10].

Моделирование по особым состояниям целесообразно использовать, если моделируемая мехатронная система является принципиально дискретной, процессы в системе представляют собой цепь событий, которые распределены во времени неравномерно или интервалы между ними велики, между событиями не происходит изменений в системе.

Обычно зависимость между скоростью изменения модельного времени и скоростью изменения физического времени является переменной и определяется ресурсами компьютера. Однако эта связь может быть и постоянной, что часто весьма желательно, например, при анимации.

Для мехатронных систем достаточно характерным является режим, когда обработка модели должна быть связана с работой реального оборудования. В этом случае говорят, что имеет место моделирование в «режиме реального времени». Режим реального времени – режим обработки данных, при котором обеспечивается взаимодействие вычислительной системы с внешними по отношению к ней процессами в темпе, соизмеримом со скоростью протекания этих процессов. Этот режим об-

работки данных широко используется в информационно-поисковых системах [21]. Кроме того, моделирование в режиме реального времени актуально при полунатурном моделировании и особенно при использовании моделей в контуре управления реальными техническими системами.

Еще одна проблема в управлении модельным временем связана с тем, что многие технические системы имеют в своем составе компоненты, работающие одновременно, или, как обычно говорят, параллельно. Эти компоненты могут взаимодействовать между собой либо работать независимо друг от друга. Учитывая, что в большинстве случаев моделирование ведется на однопроцессорных ЭВМ, возникает задача не только смоделировать параллельные процессы, но и обеспечить их взаимодействие.

Обычно в таких случаях приходится организовывать квазипараллельные модельные процессы. Одновременные события обрабатываются одно за другим при остановленном модельном времени. Время остается фиксированным до тех пор, пока не будут обработаны все события, привязанные к текущему моменту. В результате два одновременных события выполняются на ЭВМ последовательно, но в один и тот же момент модельного времени, т. е. одновременно с точки зрения системы. После этого модельное время опять оживает и начинает двигаться дальше шагами фиксированной длины (принцип  $\Delta t$ ) либо прыгая от события к событию (принцип  $\Delta z$ ).

### **3.2. Алгоритмы численного моделирования нелинейных динамических систем**

Реальные мехатронные объекты и мехатронные системы описываются, как правило, системами нелинейных алгебраических и дифференциальных уравнений. Для большинства задач, представляющих практический интерес, решение их аналитическими методами невозможно. Результаты могут быть получены путем построения приближенных решений с помощью численных методов интегрирования, в частности конечно-разностных методов.

Общая идея численного интегрирования ОДУ  $\dot{x} = f(x, t)$  заключается в том, что производится дискретизация независимой переменной – времени  $t$  на интервале  $[t_0, t_n]$  – и замена ее рядом значений  $[t_0, t_1, \dots, t_n]$  (принцип  $\Delta t$ ). Расстояние между двумя соседними значениями  $h_i = t_i - t_{i-1}$  называется **шагом интегрирования**. В частном случае он может быть постоянным на всем заданном интервале изменения переменной  $t$ . В результате системе дифференциальных уравнений тем или иным способом ставится в соответствие система конечно-разностных уравнений

$$x_i = G(x_{i-1}, x_{i-2}, \dots, x_{i-p}, \dot{x}_i, \dot{x}_{i-1}, \dots, \dot{x}_{i-p}),$$

где  $G$  – некоторая вектор-функция, определяемая способом построения метода;  $p$  – количество предыдущих точек, которые используются в методе интегрирования.

Процедура интегрирования предполагает решение полученной системы конечно-разностных уравнений для фиксированных моментов времени  $[t_0, t_1, \dots, t_n]$ , начиная с момента  $t_0$ , для которого определено начальное состояние исследуемой системы  $x_0$ . Соответственно, решение получается в виде совокупности значений  $[x_0, x_1, \dots, x_n]$  для заданных моментов времени.

В теории численных методов разработано большое количество различных методов интегрирования, каждому из которых соответствует своя система конечно-разностных уравнений.

Общее представление о них можно получить, разделив их, например, на следующие группы:

- методы явные и неявные;
- методы одношаговые и многошаговые;
- методы первого, второго и т. д. порядков;
- методы с постоянным шагом и методы с автоматическим выбором шага.

### **3.2.1. Свойства методов численного интегрирования**

Основными требованиями, предъявляемыми к численным методам, являются достаточная точность и устойчивость. Дополнительными – универсальность, алгоритмическая надежность, умеренные затраты машинного времени и оперативной памяти ЭВМ [29]. При этом следует учитывать, что практически все эти характеристики имеют смысл только применительно к конкретному объекту моделирования. Поэтому выбор подходящего метода интегрирования может иметь очень большое значение с точки зрения эффективности исследования.

Анализ процесса функционирования технического объекта численными методами всегда сопровождается ошибками в определении характеристик и параметров моделируемого процесса. Эти ошибки обусловлены такими причинами, как неадекватность модели, приближенность исходных данных, свойства используемого метода интегрирования. Первые два фактора возникают на этапе получения исходной модели. Последний зависит от выбранного метода численного интегрирования.

Точность метода можно оценить, проанализировав полную ошибку на каждом шаге интегрирования, однако это достаточно сложная задача, т. к. она предполагает наличие точного решения.

Ошибка интегрирования на  $i$ -ом шаге включает следующие составляющие:

- ошибка дискретизации  $\varepsilon_k^a$ , обусловленная заменой производных конечными разностями;
- ошибка вычислений  $\varepsilon_k^b$ , связанная с округлением чисел при вычислениях;
- ошибка накопления  $\varepsilon_k^i$ , возникающая вследствие ошибок на предыдущих шагах интегрирования.

Ошибка накопления на  $j$ -ом шаге равна полной ошибке на предыдущем шаге. Ее оценка связана с исследованием устойчивости численного метода. Если метод устойчив, то  $\varepsilon_k^i$  существенно не возрастает и общую ошибку интегрирования можно связать в основном с  $\varepsilon_k^a$ .

Если даже при небольших погрешностях аппроксимации при каждом шаге интегрирования  $h$  накопленная погрешность  $\varepsilon_k^i$  растет от шага к шагу, то метод неустойчив и результаты исследований неверны.

Основным источником неустойчивости процесса интегрирования является несогласованность выбора метода интегрирования и метода управления шагом интегрирования с характером решаемой задачи, с особенностями исследуемой системы дифференциальных уравнений. Один и тот же метод может быть достаточно эффективен при решении одной задачи и неприемлем для другой.

Анализ устойчивости метода численного интегрирования для конкретного объекта строится на том, что после дискретизации и алгебраизации его модель превращается в систему разностных уравнений. Устойчивость такой системы можно проверить тем же методом, что и устойчивость обычных дискретных систем. Она зависит от расположения корней характеристического уравнения, полученного для системы разностных уравнений, которое, в свою очередь, определяется выбором формулы интегрирования, шагом интегрирования и собственными значениями матрицы Якоби исходной системы дифференциальных уравнений.

Необходимо рассмотреть влияние ошибок округления, возникающих при реализации методов численного интегрирования на ЭВМ, ограничившись следующим интуитивным рассуждением [23]. Ошибка дискретизации любого устойчивого метода зависит от величины шага интегрирования  $h$  и стремится к нулю при  $h \rightarrow 0$ . Следовательно, за счет уменьшения шага мы можем сделать ее сколь угодно малой. Однако чем меньше шаг, тем больше потребуется шагов для решения конкретной задачи, тем больше скажутся на полученном решении ошибки

округления. На практике, за счет ограниченной разрядности машинных слов, всегда существует величина шага  $h_0$ , меньше которой вклад ошибок округления начинает доминировать в суммарной ошибке. Эта ситуация схематически изображена на рис. 3.2.

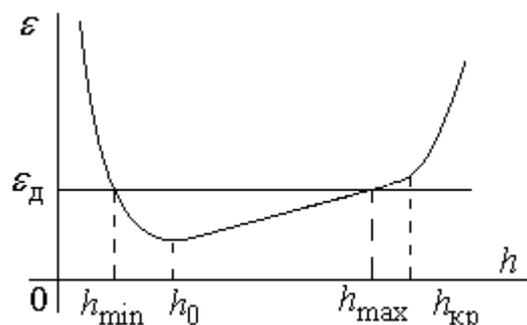


Рис. 3.2. Зависимость ошибки интегрирования от величины шага интегрирования

При малых значениях шага интегрирования велико влияние ошибок округления. В средней части диапазона ошибка растет примерно пропорционально шагу интегрирования, что соответствует методу первого порядка (см. раздел 3.2.5.). Превышение шагом значения  $h_{кр}$  приводит к неустойчивости процедуры. Значения  $h_{max}$  и  $h_{min}$  могут быть найдены исходя из заданной точности интегрирования  $\varepsilon_d$ .

Экстремальную величину шага  $h_0$  очень трудно установить заранее, но в задачах, где не требуется слишком высокая точность, необходимый шаг обычно будет значительно больше, чем этот минимум, и основной вклад в полную ошибку будет вносить ошибка дискретизации.

### 3.2.2. Методы явные и неявные

Процесс формирования математической модели для численного интегрирования обязательно включает этап алгебраизации, который состоит в преобразовании ОДУ в алгебраические уравнения. Он основан на использовании одного из методов численного интегрирования.

Если задано дифференциальное уравнение

$$\frac{dx}{dt} = f(x, t) \quad (3.1)$$

и начальные условия  $x_0 = x(t_0)$ , то очередное значение  $x_1 = x(t_1)$  может быть получено интегрированием (3.1):

$$x_i = x_{i-1} + \int_{t_{i-1}}^{t_i} f(x, t) dt. \quad (3.2)$$

Определенный интеграл в (3.2) численно равен площади под кривой  $f(x,t)$  на интервале  $(t_{i-1}, t_i)$  (рис. 3.2). Приближенно эта площадь может быть вычислена как площадь прямоугольника, высота которого равна значению функции  $f(x_{i-1}, t_{i-1})$  на левой границе интервала или значению  $f(x_i, t_i)$  на правой границе интервала. Очевидно, площади обоих прямоугольников, ограниченных сверху отрезками 1 и 2 на рис. 3.3, будут тем ближе к точному значению интеграла, чем меньше шаг интегрирования  $h_i$ .

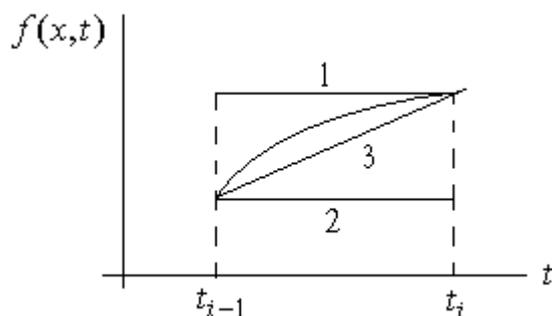


Рис. 3.3. Иллюстрация шага интегрирования

Подставив в (3.2) приближенные значения интеграла, можно получить две формулы:

$$x_i = x_{i-1} + h_i f(x_{i-1}, t_{i-1}); \quad (3.3)$$

$$x_i = x_{i-1} + h_i f(x_i, t_i). \quad (3.4)$$

Выражение (3.3) представляет собой формулу явного метода Эйлера. Называется метод явным потому, что неизвестное значение  $x_i$  может быть непосредственно вычислено по известному значению  $x_{i-1}$  в предыдущей точке.

Формула (3.4) соответствует неявному методу Эйлера. Здесь в правой части выражения используется неизвестное значение  $x_i$ , поэтому вычислить его непосредственно по этой формуле нельзя.

Более точное значение интеграла (3.2) дает метод трапеций, которому соответствует отрезок 3 на рис. 3.3. Тогда

$$x_i = x_{i-1} + h_i [f(x_{i-1}, t_{i-1}) + f(x_i, t_i)] / 2. \quad (3.5)$$

Эта формула относится, очевидно, тоже к неявным.

Для явных методов процедура формирования модели для численного интегрирования ограничивается алгебраизацией исходных дифференциальных уравнений. В частности, формула (3.3) не требует дальнейших преобразований и готова для применения.

Для неявных методов дальнейшие действия зависят от того, какой метод решения системы нелинейных уравнений реализован в данном пакете.

Одним из вариантов может быть использование итерационного метода Ньютона [23], который, как известно, обладает наибольшей скоростью сходимости среди практически применяемых методов и в котором многократно решается система линеаризованных алгебраических уравнений.

В этом случае реализуется второй этап подготовки математических моделей для неявных методов, который состоит в линеаризации нелинейных алгебраических уравнений, т. е. в разложении нелинейных функций в ряд Тэйлора и сохранении в результате только линейных членов этого ряда.

Пусть задано нелинейное алгебраическое уравнение

$$F(x) = 0, \quad (3.6)$$

где  $x = (x_1, x_2, \dots, x_n)$  – вектор переменных.

Разложение (3.6) в ряд Тэйлора с сохранением только линейных членов дает приближенную замену

$$F(x) = F(x_0) + \sum_{i=1}^n \frac{\partial F}{\partial x_i} \Big|_{x_i=x_{i0}} \Delta x_i = 0, \quad (3.7)$$

где  $x_0 = (x_{10}, x_{20}, \dots, x_{n0})$  – начальное приближение, в качестве которого берутся значения переменных на предыдущем шаге интегрирования;

$\Delta x_i = x_i - x_{i0}$ ;

$x_i$  – неизвестное значение переменной на шаге интегрирования.

Выражение (3.7) может быть записано как линейное алгебраическое уравнение

$$a_1 x_1 + a_2 x_2 + \dots + a_n x_n = b, \quad (3.8)$$

где  $a_i = \frac{\partial F}{\partial x_i} \Big|_{x=x_0}$  – вычисляется для известных значений переменных

на предыдущем шаге интегрирования;

$$b = \sum_{i=1}^n \frac{\partial F}{\partial x_i} \Big|_{x=x_0} x_{i0} - F(x_0).$$

Таким образом, процесс численного моделирования в общем случае нелинейных систем неявными методами состоит в формировании и решении на каждом шаге интегрирования системы линейных алгебраических уравнений (СЛАУ)

$$Mv = b, \quad (3.9)$$

которая включает компонентные и топологические уравнения моделируемой схемы. При этом процедурам алгебраизации и линеаризации подвергаются только компонентные уравнения, т. к. топологические уравнения всегда являются линейными алгебраическими.

Рассмотрим пример, связанный с подготовкой модели для численного решения нелинейного дифференциального уравнения второго порядка:

$$\dot{y} + \ln(\dot{y}) + y = 0.$$

Первым шагом является сведение данного уравнения к задаче Коши, т. е. к системе уравнений первого порядка за счет введения новой переменной  $z$ :

$$\begin{aligned}\dot{y} &= z, \\ \dot{z} &= -\ln(z) - y.\end{aligned}$$

Явные формулы метода Эйлера имеют вид

$$\begin{aligned}y_i &= y_{i-1} + h z_{i-1}, \\ z_i &= z_{i-1} - h(\ln(z_{i-1}) + y_{i-1}).\end{aligned}$$

Неявные формулы запишутся следующим образом:

$$\begin{aligned}y_i &= y_{i-1} + h z_i, \\ z_i &= z_{i-1} - h(\ln(z_i) + y_i).\end{aligned}$$

Для перехода к матричной записи выполним ряд преобразований:

$$\begin{aligned}F_1(x) &= y_i - y_{i-1} - h z_i, \\ F_2(x) &= z_i - z_{i-1} + h(\ln(z_i) + y_i).\end{aligned}$$

Здесь  $x^T = (y, z)$ ,

$$\begin{aligned}a_{11} &= \left. \frac{\partial F_1}{\partial y_i} \right|_{x=x_{i-1}} = 1; \quad a_{12} = \left. \frac{\partial F_1}{\partial z_i} \right|_{x=x_{i-1}} = -h; \\ b_1 &= a_{11} y_{i-1} + a_{12} z_{i-1} - F_1(x_{i-1}) = y_{i-1}; \\ a_{21} &= \left. \frac{\partial F_2}{\partial y_i} \right|_{x=x_{i-1}} = h; \quad a_{22} = \left. \frac{\partial F_2}{\partial z_i} \right|_{x=x_{i-1}} = 1 + \frac{h}{z_{i-1}}; \\ b_2 &= a_{21} y_{i-1} + a_{22} z_{i-1} - F_2(x_{i-1}) = z_{i-1} + h(1 - \ln(z_{i-1})).\end{aligned}$$

Матричная запись имеет вид

$$\begin{bmatrix} 1 & -h \\ h & 1 + \frac{h}{z_{i-1}} \end{bmatrix} \cdot \begin{bmatrix} y_i \\ z_i \end{bmatrix} = \begin{bmatrix} y_{i-1} \\ z_{i-1} + h(1 - \ln(z_{i-1})) \end{bmatrix}.$$

Формулу (3.7), вообще говоря, необходимо применять итерационно. Решение этого уравнения, найденное для заданного начального приближения  $x_0$ , следует использовать в качестве очередного приближения в (3.7) и повторять формирование и решение линейного алгебраического уравнения до тех пор, пока два последовательных приближения не станут близкими с заданной точностью.

При численном моделировании можно ограничиться только одной итерацией, выбирая достаточно малый шаг интегрирования и учитывая, что при этом значения переменных на предыдущем шаге являются достаточно хорошим приближением.

### **3.2.3. Выбор между явными и неявными методами в процедурах моделирования мехатронных систем**

Выбор между явными и неявными методами представляет серьезную проблему. Многие специалисты считают неявные методы более универсальным инструментом для решения задач моделирования технических систем [15, 23]. Следует, однако, заметить, что лишь недавно появились достаточно мощные и универсальные системы автоматизированного моделирования (САМ), такие, как, например, MATLAB или MBTU [17], допускающие выбор явного или неявного метода решения задачи. Раньше использовались либо явные, либо неявные методы, т. к. это требовало разных компонентных моделей.

В современных перспективных САМ, пригодных для моделирования мехатронных систем, применяются, как правило, неявные методы численного интегрирования. Неявные методы лучше приспособлены для решения систем дифференциальных и алгебраических уравнений, к тому же, они более устойчивы. В результате, несмотря на большие затраты машинного времени на каждом шаге интегрирования, связанные с необходимостью решения СЛАУ, общие затраты могут быть значительно меньше за счет увеличения шага интегрирования и уменьшения общего количества шагов.

Рассмотрим эту особенность неявных методов на примере явного и неявного методов Эйлера [21], определяемых формулами (3.3) и (3.4), соответственно.

Применим указанные формулы для численного интегрирования простейшего линейного дифференциального уравнения

$$\frac{dx}{dt} = -ax, \quad x(0) = x_0.$$

Характеристическое уравнение данной динамической системы имеет вид

$$s + a = 0,$$

или

$$\frac{1}{a}s + 1 = 0,$$

где  $\frac{1}{a}$  – постоянная времени системы.

Единственный полюс системы находится в левой полуплоскости, следовательно, исходная система устойчива. Соответственно, любое решение уравнения при  $t \rightarrow \infty$  стремится к нулю.

Разностное уравнение, соответствующее численному решению явным методом Эйлера, запишется как

$$x_i = x_{i-1} - hax_{i-1} = (1 - ha)x_{i-1}.$$

Известно, что условием устойчивости полученного разностного уравнения является

$$\|1 - ha\| < 1 \text{ или } h < \frac{2}{|a|}.$$

Это означает, что выбор  $h$  может качественно изменить вид решения, превратив устойчивый процесс в неустойчивый.

Таким образом, на шаг интегрирования наложено очевидное ограничение: он ограничен постоянной времени системы, иначе дискретная аппроксимация станет неустойчивой.

Если система имеет несколько постоянных времени, то подобное ограничение связывает шаг интегрирования с самой маленькой постоянной времени системы.

Переход к методам более высокого порядка мало меняет картину. Для метода Рунге–Кутты 4-го порядка требование устойчивости ограничивает шаг величиной

$$h < 2,78 \frac{1}{|a|},$$

или, в более общем виде:

$$h < 2,78 / |\lambda_{\max}|,$$

где  $\lambda_{\max}$  – максимальное собственное значение матрицы Якоби [29].

Применение неявного метода Эйлера к той же системе дает

$$x_i = x_{i-1} - hax_i = [(1 + ha)^{-1}]^i x_0,$$

где ограничение на величину шага выглядит по-другому:

$$(1 + ha)^{-1} \leq 1,$$

что позволяет выбрать шаг любой величины, ориентируясь только на требуемый уровень погрешности.

### **3.2.4. Многошаговые методы интегрирования**

До сих пор мы имели дело с методами, зависящими только от  $x_{i-1}$  и не использующими никаких иных предыдущих значений переменной. Такие методы называются одношаговыми и могут быть представлены в общем виде как

$$x_i = x_{i-1} + h\varphi(t_{i-1}, x_{i-1})$$

с соответствующей функцией  $\varphi$ . Представляется вполне вероятным, что можно добиться большей точности, если использовать информацию о нескольких предыдущих точках  $x_{i-2}, x_{i-3}, \dots$ . Именно так поступают в многошаговых методах.

Вернемся к задаче Коши

$$\frac{dx}{dt} = f(x, t), \quad t_0 \leq t \leq t_n, \quad x(t_0) = x_0,$$

при этом рассмотрим лишь один большой и важный класс многошаговых методов, который возникает на основе следующего подхода. Если подставить в приведенное дифференциальное уравнение точное решение  $x(t)$  и проинтегрировать на отрезке  $[t_{i-1}, t_i]$ , то получим

$$x_i - x_{i-1} = \int_{t_{i-1}}^{t_i} \dot{x}(t) dt = \int_{t_{i-1}}^{t_i} f(t, x(t)) dt \approx \int_{t_{i-1}}^{t_i} p(t) dt,$$

где в последнем выражении предполагается, что  $p(t)$  – полином, аппроксимирующий  $f(t, x)$ . Чтобы записать этот полином, предположим, как обычно, что  $x_{i-1}, x_{i-2}, \dots, x_{i-N-1}$  – приближения к решению в точках  $t_{i-1}, t_{i-2}, \dots, t_{i-N-1}$  и узлы расположены равномерно с шагом  $h$ . Таким образом,  $p(t)$  – полином степени  $N$ , удовлетворяющий условиям  $p(t_k) = f_k$ , ( $k = i-1, i-2, \dots, i-N-1$ ). Этот полином можно явно проинтегрировать, что ведет к методу

$$x_i = x_{i-1} + \int_{t_{i-1}}^{t_i} p(t) dt.$$

Для  $N=0$  полином  $p(t)$  есть константа, равная  $f_{i-1}$ , и мы получаем обычный метод Эйлера. Если  $N=1$ , то  $p(t)$  – линейная функция, проходящая через точки  $(t_{i-2}, x_{i-2})$  и  $(t_{i-1}, x_{i-1})$ , т. е.

$$p(t) = -\frac{(t-t_{i-1})}{h} f_{i-2} + \frac{(t-t_{i-2})}{h} f_{i-1}.$$

Интегрируя ее от  $t_{i-1}$  до  $t_i$ , получаем следующее выражение:

$$x_i = x_{i-1} + \frac{h}{2}(3f_{i-1} - f_{i-2}), \quad (3.10)$$

которое соответствует двухшаговому методу интегрирования, поскольку использует информацию в двух предыдущих точках. Аналогично, если  $N=3$ , то  $p(t)$  является квадратичным полиномом, а соответствующий трехшаговый метод имеет вид

$$x_i = x_{i-1} + \frac{h}{12}(23f_{i-1} - 16f_{i-2} + 5f_{i-3}). \quad (3.11)$$

Методы, соответствующие формулам (3.10) и (3.11), называются методами Адамса–Бишфорта.

Процесс можно продолжить, используя интерполяционный полином все более высокого порядка. При этом получаются все более громоздкие формулы, но принцип остается тот же.

Многошаговые методы порождают проблему, которая не возникала при использовании одношаговых методов. Интегрирование начинается с начального значения  $x_0$ , но при  $i = 1$  для счета, например, по формуле (3.11) необходима информация о значении функции в точках  $t_{-2}, t_{-3}$ , которая принципиально отсутствует. Обычный выход из положения состоит в использовании какого-либо одношагового метода того же порядка точности, пока не будет набрана необходимая информация.

Заметим также, что многошаговыми могут быть и неявные методы. В этом случае в формулы входят значения  $x_{i+1}, x_{i+2}, \dots$ , которые могут быть определены только неявно и найдены в результате решения системы алгебраических уравнений. Методы этой группы обычно называются методами Адамса–Моултона.

На практике часто используют совместно явную и неявную формулы, что приводит к методам, известным как методы прогноза и коррекции [23].

### **3.2.5. Порядок метода интегрирования**

Главный вопрос при использовании любого численного метода состоит в оценке точности приближенных вычислений  $x_i$ . В разделе 3.2.1. уже отмечалось, что существуют два источника погрешности при выполнении шага интегрирования:

- ошибка дискретизации, возникающая в результате замены дифференциального уравнения (3.1) разностной аппроксимацией (3.2);
- ошибка округления, накопившаяся при выполнении арифметических операций.

При этом доминирующей является, как правило, ошибка дискретизации.

Будем считать, что все вычисления проводятся точно. Интуитивно ясно, что при  $h \rightarrow 0$  ошибка дискретизации также должна стремиться к нулю, и это действительно имеет место для любого метода. Однако зависимость скорости уменьшения ошибки от скорости уменьшения  $h$  для разных методов интегрирования различна.

Введем величину

$$E(h) = \max_{1 \leq i \leq N} (x_i - x(t_i)),$$

называемую **глобальной ошибкой дискретизации**, где  $x_i$  и  $x(t_i)$  – расчетное и точное значение переменной  $x$  в момент  $t_i$ , соответственно. Отметим, что  $E(h)$  зависит от величины шага, поскольку предполагается, что значения  $x_i$  вычисляются при определенном  $h$ . Воспользуемся также стандартным обозначением  $O(h)$  для величины, стремящейся к нулю при  $h \rightarrow 0$  с той же скоростью, что и  $h$ . В общем случае будем говорить, что функция  $g(h)$  равна  $O(h^p)$ , если при  $h \rightarrow 0$  величина  $g(h)/h^p$  ограничена.

Проведя анализ, можно показать [23], что для метода Эйлера  $E(h) = O(h)$ . Это обычно выражают утверждением, что метод Эйлера имеет первый порядок. Практическим следствием данного факта является то, что при уменьшении  $h$  приближенное решение будет все более точным и при  $h \rightarrow 0$  будет сходиться к точному решению с линейной скоростью по  $h$ , т. е. при уменьшении шага вдвое ошибка дискретизации уменьшится примерно в два раза. Столь медленная сходимость служит препятствием для использования простых методов первого порядка.

Очевидно, что повышение порядка метода позволяет повысить точность интегрирования при той же величине шага интегрирования  $h$ . Способы повышения порядка могут быть различными.

Рассмотрим, например, явный одношаговый метод Хьюна (или метод Рунге–Кутты второго порядка). Он определяется формулой

$$x_i = x_{i-1} + \frac{h}{2} [f(t_{i-1}, x_{i-1}) + f(t_i, x_{i-1} + hf(t_{i-1}, x_{i-1}))].$$

Сравнивая его с методом Эйлера, легко заметить, что значение  $f(t_{i-1}, x_{i-1})$  заменено на среднее значений функции  $f$ , вычисленных в двух различных точках. Данный метод имеет ошибку дискретизации  $O(h^2)$ .

Наиболее знаменитым из методов Рунге–Кутты, а возможно, и из всех методов численного интегрирования является классический метод четвертого порядка, задаваемый формулами

$$x_i = x_{i-1} + \frac{h}{6} (F_1 + 2F_2 + 2F_3 + F_4),$$

где

$$F_1 = f(t_{i-1}, x_{i-1}); \quad F_2 = f(t_{i-1} + \frac{h}{2}, x_{i-1} + \frac{h}{2} F_1);$$

$$F_3 = f(t_{i-1} + \frac{h}{2}, x_{i-1} + \frac{h}{2} F_2); \quad F_4 = f(t_i, x_{i-1} + hF_3).$$

Суть метода в том, что вектор-функция правых частей системы ОДУ определяется не только в узлах сетки, но и в промежуточных точках интервала  $[t_i, t_{i-1}]$ , а величина  $f(x_k, y_x)$  в методе Эйлера заменена на взвешенное среднее значений  $f$ , вычисленных в четырех различных точках.

Как видно, повышение порядка метода связано с повышением затрат на вычисление значений функции  $f$ . При нелинейной связи между ростом точности метода и объемом вычислений можно ожидать, что для каждой схемы интегрирования существует некоторый оптимальный порядок метода.

Для многошаговых методов порядок напрямую связан с объемом информации, используемой на каждом шаге. В частности, двух- и трехшаговые методы Адамса–Бишфорта, задаваемые формулами (3.10) и (3.11), имеют второй и третий порядок, соответственно.

### **3.2.6. Процедуры численного моделирования с автоматическим выбором шага**

Данный раздел касается не выбора того или иного метода интегрирования, а реализации самой процедуры интегрирования на ЭВМ.

Выше уже отмечалось, что выбор шага интегрирования связан с динамическими свойствами моделируемого объекта. Для явных методов он должен быть меньше минимальной постоянной времени объекта, с тем чтобы обеспечить устойчивость и позволить моделировать самые высокочастотные составляющие процесса. Неявные методы позволяют использовать больший шаг, но общий характер зависимости остается тем же самым. В то же время на интервале моделирования характер моделируемых процессов может меняться. Например, в большинстве реальных систем высокочастотные составляющие переходного процесса затухают быстрее, чем низкочастотные, и процесс со временем приобретает более плавный характер. Это наводит на мысль организовать процедуру моделирования таким образом, чтобы шаг интегрирования менялся в процессе работы алгоритма. Там, где решение меняется плавно, можно вести интегрирование с относительно большим шагом. В областях же, где решение изменяется резко, необходимо считать с маленьким шагом. Проблема заключается в том, как определить величину шага, с которым необходимо начать следующий шаг интегрирования.

На выбор шага, как обычно, влияют два фактора – точность и устойчивость. Шаг целесообразно поддерживать таким, чтобы погрешность интегрирования не превышала допустимое значение, а величина шага при этом была максимально возможной.

Обычный подход состоит в оценке локальной ошибки дискретизации и, в зависимости от ее величины, в уменьшении или увеличении текущего значения шага.

Два простых способа состоят либо в прохождении последнего шага интегрирования с половинным шагом и сравнении двух полученных результатов, либо в использовании двух методов интегрирования, имеющих разный порядок. Оба эти способа требуют дополнительных вычислений значений  $f$ .

Первый способ реализует правило Рунге [23, 29], при котором ошибка дискретизации определяется по формуле

$$\varepsilon_i = c \frac{|x_i(h/2) - x_i(h)|}{2^p - 1},$$

где  $c$  – некоторая константа;

$x_i(h/2)$  – значение фазовой переменной  $x$ , полученной в точке  $t_i$  тем же методом, что и  $x_i(h)$ , но только за два шага интегрирования от точки  $t_{i-1}$ , каждый из которых в два раза меньше обычного шага  $h$ ;  $p$  – порядок метода интегрирования.

Для метода Эйлера формула Рунге дает

$$\varepsilon_i = c |x_i(h/2) - x_i(h)|,$$

для метода Рунге–Кутты 4-го порядка:

$$\varepsilon_i = c |x_i(h/2) - x_i(h)|/15.$$

Величина погрешности аппроксимации на каждом шаге  $\varepsilon_i$  не должна превышать допускаемой погрешности  $\varepsilon_d$ . Обычно полагают, что она составляет от 0,010 до 0,001 от текущего значения определяемой фазовой координаты  $x$ .

При автоматическом выборе шага наиболее популярным является алгоритм «трех зон» [29]:

$$h_{k+1} = \begin{cases} h_k / m, & \text{если } \varepsilon_k > \varepsilon_d \\ h_k, & \text{если } \varepsilon_d / m^2 \leq \varepsilon_k \leq \varepsilon_d, \\ h_k m, & \text{если } \varepsilon_k < \varepsilon_d / m^2 \end{cases}$$

где  $m$  – коэффициент изменения шага, обычно равный 2, позволяющий дискретно менять шаг в процессе интегрирования. Другим вариантом является алгоритм плавного изменения шага интегрирования

$$h_{i+1} = h_i^{p+1} \sqrt[p+1]{\varepsilon_d / \varepsilon_i},$$

где  $p$  – порядок метода интегрирования.

### 3.2.7. Особенности численного интегрирования мехатронных систем

Обширный набор методов численного интегрирования, которым обладают современные пакеты моделирования, позволяет эффективно решать самые различные задачи исследования мехатронных систем, но при этом возникает проблема выбора наиболее подходящего метода и правильного задания его параметров. Очень часто пользователь задает только интервал интегрирования и не обращает внимания на другие опции решателя. При решении простых задач с умеренной точностью такой подход вполне допустим, т. к. по умолчанию обычно используется достаточно универсальный метод. Однако при решении сложных задач неудачный выбор метода либо неправильное задание его параметров может привести к неоправданно большим затратам машинного времени либо вообще к невозможности получить правильное решение.

Таким образом, для профессиональной работы с любым моделирующим программным обеспечением пользователь должен обладать некоторыми знаниями о реализованных в нем численных методах и применимости их к требуемому кругу задач.

Практика показала, что наличие в мехатронных системах объектов различной физической природы приводит к тому, что процессы в них характеризуются разнотемповостью, т. е. наличием быстрых и медленных составляющих движения. Кроме того, возможно одновременное присутствие как монотонных, так и медленно затухающих гармонических составляющих. Свойство системы содержать в своем переходном процессе существенно различные по своим спектральным характеристикам составляющие принято называть **жесткостью** [27].

Примером жесткой системы может служить робототехническая система, в которой быстрые составляющие связаны с включением тормозных устройств и демпферов, с взятием или освобождением груза.

Жесткость часто является следствием избыточности модели, связанной с введением в нее малозначащих составляющих. Однако на этапе предварительных исследований такой избыточности трудно избежать. С другой стороны, часто жесткость имеет принципиальный характер, и учет быстрых движений может привести к неадекватности модели.

Жесткие системы требуют особого подхода к процедуре численного интегрирования, т. к. наличие быстрых и медленных составляющих в спектре движений предъявляет совершенно разные требования к методам интегрирования. Необходимо уметь оценивать характеристики жесткости и использовать эти оценки для выбора или настройки процедуры интегрирования.

Пусть исследуемая система  $\dot{x} = f(x)$  является линейной и может быть описана векторно-матричным уравнением состояния

$$\dot{x} = Ax + Bu.$$

Матрица  $A$  называется матрицей состояния, или матрицей Якоби. Собственные значения  $\lambda_i$  матрицы  $A$  определяют устойчивость и характер переходных процессов в исследуемой системе.

Составляющие движения (обычно называемые модами), связанные с собственными значениями  $\lambda_i$ , лежащими в левой полуплоскости далеко от мнимой оси, соответствуют быстро протекающим и (обычно) быстро затухающим процессам в системе. Собственные значения, малые по модулю, лежащие близко к мнимой оси, определяют основное движение системы.

Исходя из распределения собственных значений матрицы Якоби на комплексной плоскости, можно назвать жесткой системой ОДУ такую систему, у которой матрица Якоби имеет различающиеся на несколько порядков максимальное и минимальное по модулю собственные значения.

Оценкой жесткости системы ОДУ обычно считается число обусловленности матрицы Якоби

$$\alpha = \|A\| \cdot \|A^{-1}\|,$$

где  $\|A\|$  – норма матрицы  $A$ .

Для целей управления процессом моделирования под числом обусловленности чаще понимают отношение модулей максимального и минимального собственных значений матрицы:

$$\alpha = \max |\lambda_i| / \min |\lambda_i|, \quad i = 1, n.$$

К жестким относят системы ОДУ, у которых  $\alpha \geq 10^5$ . Их также называют плохо обусловленными, хотя чаще этот термин относится к системам алгебраических уравнений.

Для нелинейной системы матрицу Якоби можно определить после ее линеаризации в рабочей точке, как это описано в разделе 3.2.2.

Элементами матрицы Якоби являются частные производные от нелинейной вектор-функции  $f(x)$  по переменным состояния системы  $x$ . Для нелинейных систем жесткость, в общем случае, не является постоянной величиной и меняется в процессе интегрирования.

При умеренных значениях числа обусловленности  $\alpha$  (нежесткие задачи) интегрирование обычно выполняется традиционными явными методами и требует небольших вычислительных затрат. Трудности возникают при больших значениях  $\alpha$ , когда для получения правильного решения бывает необходимо выбирать очень малый шаг интегрирования.

При моделировании такой системы в начальный момент времени иницируются все или большинство мод, как быстрых, так и медленных. Однако через некоторое время быстрые моды затухают, и решение сходится к медленному движению.

Исследователя могут интересовать и быстрые, и медленные составляющие движения. В этом случае целесообразно использование явных методов в сочетании с эффективной процедурой изменения величины шага интегрирования в зависимости от состояния моделируемой системы. Это позволит достаточно точно смоделировать быстрые движения и избежать чрезмерных затрат машинного времени, т. к. явные методы требуют минимальных временных затрат на каждый шаг интегрирования.

Если исследователя мало интересуют быстро затухающие движения, но отбросить их на этапе формирования модели у него нет достаточных оснований, предпочтительными являются неявные методы, которые в сумме способны дать меньшее время интегрирования при удовлетворительном качестве. Такие методы подавляют все составляющие решения, соответствующие большим по модулю собственным значениям (если только шаг не выбран очень малым).

Изложенные рекомендации по выбору методов интегрирования жестких систем предполагают, что исследователь хорошо знаком с особенностями объекта. Например, если речь идет о проектировании или оптимизации системы управления некоторого хорошо изученного объекта, то выбор метода интегрирования может быть проведен путем сравнения нескольких вариантов. Если же объект недостаточно исследован, то эффективными могут оказаться процедуры, обладающие элементами адаптации к особенностям объекта.

Некоторые современные моделирующие программные комплексы содержат наборы методов, расчетные формулы которых настраиваются на решаемую задачу, используя для этого оценки некоторых параметров, обычно собственных значений якобиана. Особенно перспективными считаются явные адаптивные методы, не требующие при своей реализации вычисления матрицы Якоби и решения алгебраических уравнений [27]. Такие методы есть, в частности, среди решателей ОДУ программного комплекса MBTU.

### **3.3. Моделирование гибридных (событийно-управляемых) мехатронных систем**

Мехатронные системы, по определению, относятся к сложным техническим системам. Одной из особенностей этих систем является то, что поведение многих из них определяется событиями, происходящими как внутри этих систем, так и в окружающей среде. Соответственно,

они обладают непрерывной динамикой и дискретной динамикой, находящимися в сложном взаимодействии. Подобные системы часто называют **гибридными системами** [20, 21]. В отечественной литературе также используются синонимы – непрерывно-дискретные, системы с переменной структурой, реактивные, событийно-управляемые.

Примерами гибридных мехатронных систем могут служить системы управления, используемые в промышленности (автоматизированные технологические процессы), в быту (сложные бытовые приборы), в военной области (высокотехнологичные виды вооружений), в сфере космонавтики, транспорта и связи.

Гибридное поведение может быть связано со следующими факторами.

1. Совместное функционирование непрерывных и дискретных объектов. Подобное поведение является типовым для непрерывных объектов (механических, гидравлических и т. д.), управляемых цифровыми регуляторами, например, для искусственного спутника Земли. Управление может формироваться как в фиксированные моменты времени, так и асинхронно, в зависимости от изменения фазовых координат объекта.

2. Гибридное поведение, связанное с особенностями физических процессов в непрерывных объектах. Например, учет в моделях механических систем таких эффектов, как сухое трение или люфт, может быть реализован в виде некоторых логических условий, меняющих модель системы.

3. Гибридное поведение, обусловленное изменением состава системы.

Все перечисленные факторы являются типичными для мехатронных систем.

Учет дискретной динамики приводит к тому, что фазовое пространство гибридных систем разбивается на области с различным поведением, при этом фазовая траектория в зависимости от происходящих событий оказывается то в одной области фазового пространства, то в другой. Достижение фазовой траекторией границы областей является событием, приводящим к смене поведения.

События могут быть как внутренними, так и внешними. Например, при моделировании робота-манипулятора внутренние события могут быть связаны с типовыми нелинейностями механической части, а внешние являться следствием взаимодействия со средой.

Очевидной и удобной моделью систем, управляемых событиями, является конечно-автоматная модель. Например, можно использовать конечный автомат, чтобы описать автоматическую передачу автомобиля. Передача имеет ряд состояний: парковка, нейтраль, движение, ре-

верс и т. д. Система переходит из одного состояния в другое, когда водитель перемещает рычаг из одной позиции в другую, например, из позиции «парковка» в нейтральное положение.

При всей ее простоте и наглядности, конечно-автоматная модель достаточно строга и формальна. Однако классическое графическое представление конечных автоматов обладает рядом недостатков. Главным недостатком является отсутствие понятия времени, что предполагает статичность состояний. Другие недостатки: отсутствие иерархии состояний, обобщения переходов, средств выражения прерываний и продолжения нормальной работы после их обработки [11].

Для полноценного компьютерного моделирования физики процессов необходимо обеспечить сопряжение непрерывной составляющей поведения системы и логики работы управляющих ею устройств (дискретная компонента). Математический аппарат описания в данном случае – это система уравнений, но не дифференциальных, а дифференциально-алгебраическо-логических, для которых пока отсутствует стройная теория и единый подход.

В настоящее время для моделирования систем, управляемых событиями, широко используется предложенный Д. Харелом [7] визуальный формализм – Statechart (карты состояний и переходов). Карты состояния были разработаны применительно к моделированию дискретных систем, однако они могут служить хорошей основой и для моделирования гибридных систем, т. е. с их помощью можно описать поведение системы как в дискретном, так и в непрерывном времени [8]. Основные неграфические компоненты таких диаграмм – это событие и действие, основные графические компоненты – состояние и переход.

**Событие** – нечто, происходящее вне рассматриваемой системы, возможно, требующее некоторых ответных действий. События могут быть вызваны поступлением некоторых данных или некоторых задающих сигналов со стороны человека или некоторой другой части системы. События считаются мгновенными.

**Действия** – это реакции моделируемой системы на события. Подобно событиям, действия принято считать мгновенными.

**Состояние** – условия, в которых моделируемая система пребывает некоторое время, в течение которого она ведет себя одинаковым образом. В диаграмме переходов состоянию обычно соответствует графический элемент в виде прямоугольника со скругленными углами.

**Переход** – изменение состояния, обычно вызываемое некоторым событием. Как правило, состояние соответствует промежутку времени между двумя такими событиями. Переходы показываются в диаграммах переходов линиями со стрелками, указывающими направление перехо-

да. Каждому переходу могут быть сопоставлены условия, при выполнении которых переход осуществляется.

С каждым переходом и каждым состоянием могут быть соотнесены некоторые действия. Действия могут дополнительно обозначаться как действия, выполняемые однократно при входе в состояние; действия, выполняемые многократно внутри некоторого состояния; действия, выполняемые однократно при выходе из состояния.

В работе [25] в качестве примера простой дискретной системы (частного случая гибридной системы) рассмотрена диаграмма (карта) состояний и переходов цифровых часов, представленная на рис. 3.4.

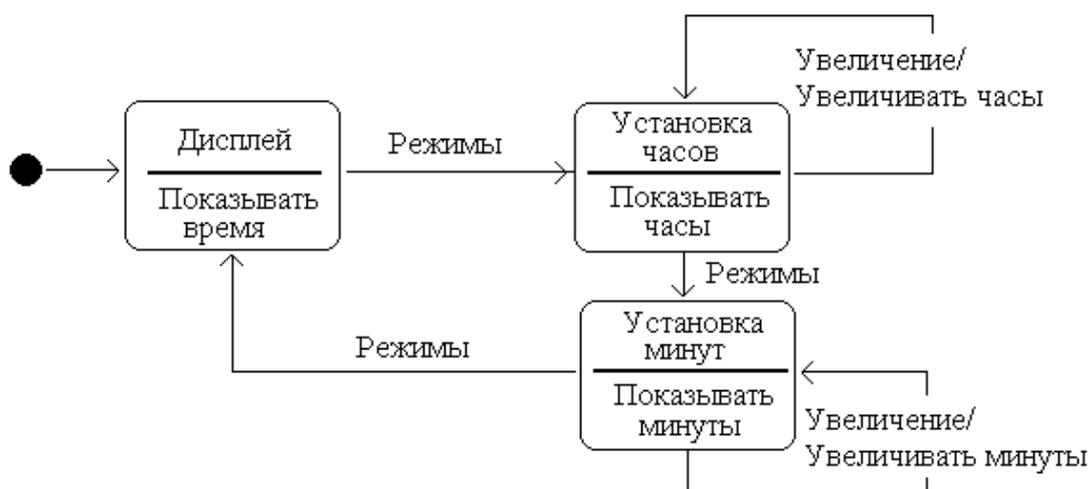


Рис. 3.4. Диаграмма состояний и переходов цифровых часов

На часах имеются две кнопки: *Кнопка режима* и *Кнопка увеличения*. Нажатие любой из них генерирует событие, которое может вызывать переход из одного состояния в другое. Имеются три состояния: *Дисплей*, *Установка часов* и *Установка минут*. Состояние *Дисплей* – начальное состояние (что обозначается стрелкой, направленной от блока перехода по умолчанию в виде черного круга). Нажатие кнопки *Режимы* в состоянии *Дисплей* вызывает появление события *Режимы* и переход в состояние *Установка часов*. В состоянии *Установка Часов* событие *Режимы* вызывает переход к состоянию *Установка минут*, тогда как событие *Увеличение* увеличивает текущее время (число часов), которое отображается на экране, причем это происходит без изменения состояния. Каждому состоянию часов соответствует действие, записанное ниже горизонтальной линии. Оно начинает выполняться после того, как переход в это состояние произошел.

В работе [21] в качестве примера динамической гибридной системы рассмотрена модель, описывающая поведение маятника в виде шарика на нити, у которого в некоторый момент времени (событие *Обрыв*)

рвется нить. У этой модели два состояния – *Колебания* и *Полет*. Оба описываются системами дифференциальных уравнений, каждая из которых является динамической системой. Карта поведений представляет собой совокупность состояний и переходов. В любой момент времени только одно состояние является текущим.

На рис. 3.5, *a* показана карта состояний обрывающегося маятника, на которой кроме двух обычных состояний помещены два особых состояния – начальное и конечное. В общем случае для состояния могут быть определены входные воздействия, выходные воздействия, а также локальная деятельность. Последовательность входных воздействий выполняется при каждом входе в состояние, последовательность выходных действий – при каждом выходе. Локальные действия выполняются все время, пока состояние является текущим.

Рис. 3.5, *a* отражает качественное поведение маятника. Для получения количественной картины необходимо связать с каждым состоянием локальное действие, в частности, интегрирование систем уравнений *Модель колебаний* и *Модель полета*, как это сделано на рис. 3.5, *b*. Карта состояний, дополненная моделями состояний, названа в работе [20] картой поведения гибридной системы.

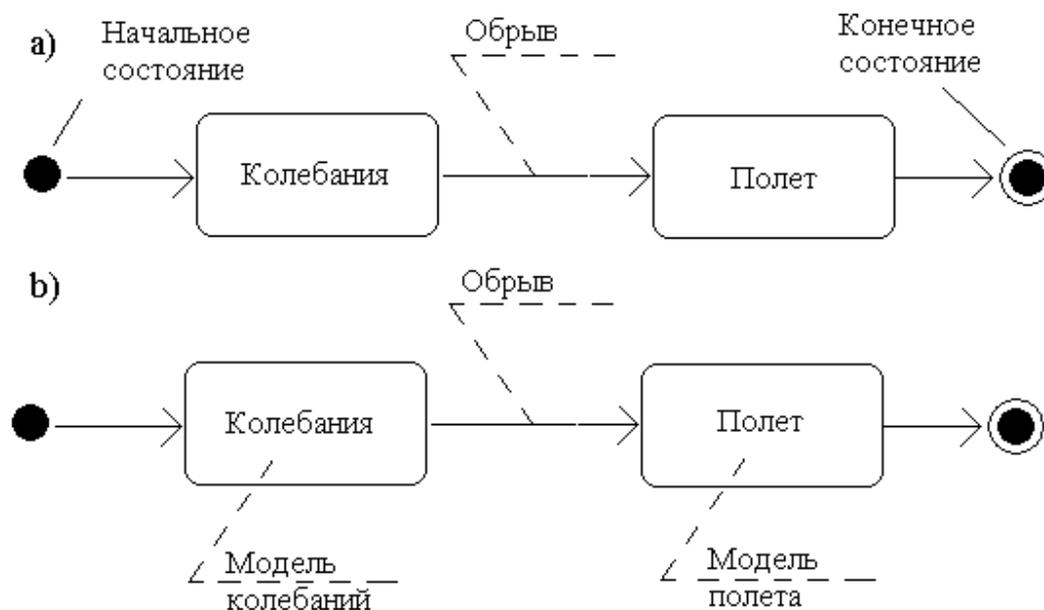


Рис. 3.5. Модель обрывающегося маятника:  
*a* – карта состояний, *b* – карта поведений

Одна из концепций гибридного моделирования реализована в пакете StateFlow среды MATLAB, который кратко описан в главе 5. Также следует выделить отечественный пакет Model Vision Studium, специально ориентированный на моделирование сложных поведений [19].

## Глава 4

# АВТОМАТИЗИРОВАННОЕ МОДЕЛИРОВАНИЕ ТЕХНИЧЕСКИХ ОБЪЕКТОВ

Необходимость автоматизации процессов моделирования технических объектов возникла практически одновременно с появлением вычислительных машин. Однако исторически разные этапы моделирования автоматизировались в разное время. Первым этапом следует считать создание библиотек численных методов исследования систем. Сами численные методы были разработаны задолго до появления ЭВМ и предназначались, прежде всего, для решения задач небесной механики. В связи с тем, что первоначально расчеты велись вручную, алгоритмы методов были хорошо отлажены и оптимизированы. К концу 70-х гг. XX в. были созданы специализированные коллекции численных методов практически для всех областей численного анализа.

Дальнейшие шаги на пути автоматизации моделирования были связаны с разработкой систем автоматизированного проектирования (САПР) и систем автоматизации вычислительного эксперимента – пакетов прикладных программ. Автоматизации подлежали стандартные расчеты и оформление результатов экспериментов. Как правило, эти системы создавались под определенную предметную область, прежде всего, в наукоемких отраслях (космос, авиация и т. д.). Для создания большинства пакетов прикладных программ для численных расчетов использовался язык Фортран, хорошо приспособленный именно для этих целей. Подобные пакеты создавались годами, их модификация и развитие требовали специальных знаний в предметной области, в численных методах и программировании.

Наиболее трудно автоматизируемым этапом явился процесс построения модели. Ручная подготовка модели сложного технического объекта связана с большим объемом преобразований, в которых легко допустить ошибку. Использование САМ позволило существенно повысить производительность труда, уменьшить число ошибок и во многих случаях исключить необходимость привлечения программистов к решению конкретных предметных задач. Опираясь на САМ, специалист в предметной (прикладной) области может самостоятельно создавать достаточно сложные модели.

При использовании САМ исследователь формулирует математическую модель исследуемой системы на формальном входном языке моделирования. На ранних этапах развития САМ, когда отсутствовали возможности прямого изображения структуры модели на экране монитора, использовались методы кодирования структурной информации [1, 6].

Программа модели представляла собой последовательность вызовов процедур, каждая из которых являлась моделью структурного компонента. После связывания с модулями исполняющей системы пакета моделирования список компонентов преобразовывался в независимую исполняемую программу.

В современных САМ, исходя из соображений удобства восприятия человеком, используются, как правило, графические средства задания исходной информации о модели. Переход к изображению структуры системы на экране монитора позволил исключить этап ручного кодирования схемы, однако потребовал разработки нового принципа организации самого процесса численных расчетов – принципа Data Flow, или принципа потока данных.

Можно отметить следующие факторы, способствующие внедрению САМ:

- трудоемкость получения математической модели сложных технических объектов, связанная с опасностью совершить ошибку в многочисленных преобразованиях модельных выражений;
- необходимость многовариантного моделирования, при котором необходимо иметь для одного объекта несколько моделей, отличающихся по сложности;
- желание иметь дружественный интерфейс с программой и возможность оперативно вносить изменения в модель, что проще всего на основе использования графических языков задания исходной информации.

#### **4.1. Особенности современных систем автоматизированного моделирования**

Современные профессиональные САМ имеют следующие отличительные черты [31]:

- поддержка иерархического проектирования как сверху вниз, так и снизу вверх, за счет реализации многоуровневого моделирования и метода локальной детализации модели;
- компонентное моделирование на основе использования библиотек, содержащих большое число графических и функциональных описаний компонентов, причем эти библиотеки открыты для добавления в них новых описаний, которые может сделать сам пользователь;

- графический пользовательский интерфейс, сочетающий графические средства формирования визуального образа исследуемого технического устройства с автоматической генерацией модели всей схемы по ее структурному описанию;
- наличие интерактивной рабочей среды проектирования (управляющей оболочки, монитора), т. е. специальной программы, из которой можно запускать все или большинство других программ пакета, не обращаясь к услугам штатной операционной системы;
- наличие постпроцессоров моделирования, что позволяет не только просматривать в удобной для пользователя форме результаты моделирования, но и обрабатывать их;
- наличие встроенных средств численного моделирования рабочего процесса в режиме реального времени или в режиме масштабирования модельного времени;
- реализация механизмов продвижения модельного времени, основанных как на принципе  $\Delta t$ , так и на принципе  $\Delta z$ ;
- интегрируемость с другими пакетами аналогичного назначения, которая обеспечивается соответствующими программами – конверторами, позволяющими импортировать и экспортировать данные из одной системы в другую;
- наличие средств, обеспечивающих формирование виртуальных аналогов измерительно-управляющей аппаратуры.

Если САМ предназначена для решения исследовательских задач, то к перечисленным качествам добавляются возможности активного вычислительного эксперимента [21], например:

- визуализация результатов во время эксперимента;
- возможность интерактивного вмешательства в ход моделирования;
- возможность использования 2D и 3D анимации.

#### **4.2. Иерархическое проектирование и многоуровневое моделирование мехатронных систем**

С точки зрения инженера основное назначение моделирования – поддержка процедур проектирования технических объектов и систем. Процедуры проектирования работают с моделями объектов реального мира и должны быть подстроены под их свойства.

Сложные системы имеют, как правило, иерархическую структуру. Естественные возможности человека позволяют оперативно обрабатывать не более  $7 \pm 2$  единиц информации одновременно. В процессе проектирования исследователю удобно сосредотачиваться сначала на поведении отдельных компонентов системы, а затем на их взаимодействии.

вии [21]. При необходимости модели компонентов могут детализироваться или, наоборот, укрупняться.

Такой подход, известный как **иерархическое проектирование** [21, 31], является типовым при разработке сложных технических объектов и заключается в разбиении исходной задачи на подзадачи.

В процессе проектирования сложной системы формируются определенные представления о системе, отражающие ее существенные свойства с той или иной степенью подробности. В этих представлениях можно выделить составные части – **уровни проектирования**.

В один уровень, как правило, включаются представления, имеющие общую физическую основу и допускающие для своего описания использование одного и того же математического аппарата. Уровни проектирования можно выделять по степени подробности, с какой отражаются свойства проектируемого объекта. Тогда их называют **горизонтальными (иерархическими) уровнями проектирования**.

В результате такого подхода объект проектирования декомпозируется на фрагменты (подсхемы), и проектирование каждого из них ведется в определенном смысле самостоятельно. На каждом уровне иерархии этот принцип применяется вновь, что позволяет заменить решение одной сложной задачи многократным решением задач меньшей размерности.

При иерархическом проектировании разработчику достаточно держать в поле зрения один фрагмент объекта. Остальные части лишь имитируют внешнюю среду, т. е. взаимодействие проектируемого фрагмента с другими частями объекта.

Использование принципа иерархического проектирования позволяет ограничить текущую сложность проекта на приемлемом уровне за счет того, что в каждый момент времени разработчик имеет дело только с двумя смежными уровнями декомпозиции объекта – структурным описанием проектируемого в данный момент фрагмента и функциональным описанием внешней среды.

Инструментальной поддержкой иерархического проектирования является **многоуровневое моделирование**. При многоуровневом моделировании различные фрагменты представляются на различных уровнях иерархии, т. е. с разной степенью детальности. Например, проектируемая в настоящий момент времени часть объекта раскрыта до уровня элементарных динамических компонентов и имитируется структурной моделью, а остальные фрагменты представлены на соседнем, более высоком, уровне в виде функциональных моделей.

Завершив проектирование одного фрагмента, разработчик может свернуть его в функциональный блок и перейти к детальной модели следующего фрагмента, с которым он собирается работать. Эта процедура

повторяется многократно, на разных уровнях иерархии проектируемого объекта. Достоинством такого подхода является то, что в поле зрения разработчика находится в каждый момент времени минимум необходимой информации, не перегруженной лишними деталями. Описанный метод проектирования называется **методом локальной детализации объекта**.

Программной поддержкой многоуровневого моделирования, реализованной в большинстве языков графического программирования, является процедура **инкапсуляции**, которая позволяет «свернуть» любой смысловой фрагмент графического представления в единичный блок. Кроме того, что инкапсуляция служит основой получения иерархически структурированных моделей, она также позволяет расширить библиотеку базовых блоков блоками пользователя, которые впоследствии можно многократно использовать (например, типовые динамические звенья).

Такой возможностью обладают, в частности, пакет LabVIEW, пакет Simulink и основанные на нем пакеты SimMechanics и SimPower. Пакеты IDEF-моделирования (ERWIN, BPWIN) принципиально основаны на многоуровневом изображении объектов.

### 4.3. Архитектура программ автоматизированного моделирования

Существующие инструментальные средства автоматизированного моделирования могут относиться к разным предметным областям и значительно различаться по своим возможностям, но их модульные структуры мало отличаются друг от друга. На рис. 4.1 представлена типовая структура современного пакета визуального компонентного моделирования. Рассмотрим более детально назначение и особенности основных элементов этой структуры.



Рис. 4.1. Модульная структура САМ

#### **4.3.1. Графический интерфейс программ математического моделирования динамических систем**

Графический интерфейс является в настоящее время стандартным компонентом современной САМ. Он создает дружелюбный интерфейс между пользователем и программой, дает возможность оперировать с графическими образами вместо аналитических выражений. Это значительно облегчает работу в САМ и снижает вероятность ошибок при вводе информации о системе.

Графический пользовательский интерфейс позволяет вводить информацию об исследуемой системе путем «рисования» на экране монитора проектируемой схемы в виде, понятном широкому кругу специалистов. Формой графического представления информации о моделируемой системе могут являться следующие схемы [16]:

- операторно-структурные схемы, принятые в ТАУ;
- функциональные и принципиальные схемы различных физических устройств;
- кинематические схемы механизмов;
- сигнальные графы;
- графы связей;
- блок-схемы алгоритмов и другие графические модели.

Однако простым «рисованием» роль графического интерфейса не ограничивается. Задачи графического интерфейса, кроме того, могут быть следующие [16]:

- контроль соблюдения некоторых правил в процессе создания графического изображения на экране монитора (обычно накладываются ограничения на способы соединения компонентов и т. п.);
- преобразование информации о схеме в команды для моделирующей программы (моделятора);
- контроль за процессом моделирования, визуализация результатов моделирования и др.

#### **4.3.2. Язык описания объекта, транслятор, система управления базами данных, монитор**

Типовой состав САМ включает графический и/или текстовый язык описания объекта, с помощью которого пользователь вводит в систему моделируемую схему.

**Языки моделирования** – это специальные языки программирования, обычно графические, т. е. имеющие графические аналоги основных своих синтаксических конструкций, позволяющие создавать строгие

(поддающиеся однозначному переводу на машинные языки) описания моделей различных объектов.

Первоначально для этих целей использовались обычные языки программирования или их макрорасширения (SLAM, Simula). Необходимость введения специальных языков моделирования была обусловлена желанием упростить процедуру общения специалиста в определенной предметной области с ЭВМ. Поэтому многие языки являются предметно-ориентированными, включают в себя семантику определенной области применения. Кроме того, многие современные системы автоматизированного моделирования содержат как бы несколько уровней языка, рассчитанных на разных пользователей и разные задачи. Примером является среда MATLAB, имеющая в своем составе пакеты Simulink, SimMechanics, SimPower и др., с которыми можно работать на уровне графических описаний. Однако использование наряду с этим текстового языка программирования среды MATLAB позволяет существенно расширить круг возможностей. Часто текстовые языки моделирования создаются как расширения какого-нибудь языка программирования (C, Fortran, Java). Например, в пакете AnyLogic в качестве языка моделирования принят язык Java.

Расширение аппаратных возможностей ЭВМ привело к появлению графических языков, наиболее удобных для исследователя-непрограммиста. В настоящее время использование графических языков стало стандартом в автоматизированном моделировании. В современных графических средах пользователь не имеет дела с системами уравнений. Основным способом моделирования объектов макроуровня, в частности мехатронных систем, является **компонентное моделирование** [21, 29], которое может рассматриваться как частный случай объектно-ориентированного моделирования.

Идея данного метода состоит в том, чтобы заранее сформировать типовые компоненты заданного класса объектов, которые затем группируются в библиотеки и хранятся в базе данных системы моделирования. Для технических систем декомпозиция на компоненты обычно не составляет проблемы.

Метод компонентного моделирования хорошо сочетается с графическим способом задания информации, однако он был разработан еще тогда, когда возможности ЭВМ не позволяли полностью реализовать графический интерфейс, т. е. задавать модель путем изображения ее на экране ЭВМ в виде графического объекта [1, 6].

Важной привлекательной стороной компонентного моделирования является то, что понятие компонента (структурного примитива) не абсолютно, каждый иерархический уровень моделирования имеет свои

примитивы. В процессе проектирования объект детализируется до заданного иерархического уровня, и для его представления используется соответствующая библиотека структурных примитивов.

При этом структура может описываться различными типами иерархических схем, включающих как структурные (направленные), так и «физические» (ненаправленные) компоненты. Набор типовых компонентов может рассматриваться как специальный предметно-ориентированный язык моделирования в данной области.

Среди других достаточно типовых характеристик современных языков моделирования можно отметить возможность моделирования параллельно протекающих процессов; возможность учета и отображения структурных изменений в моделируемой системе; мощные логические возможности, не уступающие логическим возможностям универсальных языков, а иногда и превосходящие их; наличие средств для моделирования случайных событий и процессов, а также инструментов для статистической обработки результатов.

Транслятор языка описания объекта преобразует исходное описание проекта в формат, пригодный для моделирования. В случае если язык описания объекта графический, транслятор с этого языка является частью графического интерфейса.

Система управления базами данных (СУБД) отвечает за хранение библиотек моделей компонентов графических представлений, а также составленных пользователем моделей. Базы разделяются на справочные и рабочие.

СУБД обеспечивает поиск и подключение моделей компонентов, запрашиваемых пользователем через графический интерфейс; обновление, замену и удаление моделей; расширение библиотек за счет ввода новых моделей.

Управляющая оболочка САМ (монитор) обеспечивает интерфейс между функционирующим математическим ядром и пользователем, осуществляет вызовы нужных программ и делает работу с системой удобной для пользователя.

#### **4.3.3. Инструментальные средства моделирования (математическое ядро)**

Основная вычислительная нагрузка осуществляется блоками, входящими в инструментальные средства (математическое ядро) САМ. Спектр задач, которые решают программы математического ядра, может быть весьма широк: от анализа только линейных моделей (Classic), до символьных вычислений и интервальной арифметики. Для пакетов, способных полноценно исследовать мехатронные системы, основной

частью являются коллекции программ, содержащих реализации численных методов.

Как правило, современные универсальные системы моделирования содержат как средства решения статических (аналитических) задач, так и мощные средства расчета и анализа динамических процессов. Так, большая часть задач анализа линейных систем (расчет переходных характеристик, получение и преобразование типовых математических моделей, построение частотных характеристик, анализ размещения полюсов и нулей на комплексной плоскости и т. п.), не говоря уже о задачах синтеза, решается программами, входящими в блок аналитических программ. Данная часть математического ядра в значительной мере привязана к специфике объекта проектирования, его физическим и информационным особенностям функционирования, а также к конкретным иерархическим уровням проектирования. Эти программы особенно актуальны при решении задач ТАУ.

Другую часть математического ядра составляют программы, предназначенные для анализа переходных процессов в сложных нелинейных системах, где применение аналитических методов невозможно. Эта часть является более инвариантной к особенностям объекта моделирования, включает в себя методы и алгоритмы, слабо связанные с особенностями математических моделей и используемые на многих иерархических уровнях. Наиболее важными являются разнообразные программы численного интегрирования.

Численное интегрирование (то, что в иностранной литературе понимается как «Simulation») выполняется специальной программой, называемой управляющей программой моделирования, или, коротко, **моделятором**. На модельатор обычно возлагаются следующие функции [16]:

- установка начальных условий и значений сигналов на входах моделируемой схемы;
- управление модельным временем по принципу  $\Delta t$  или  $\Delta z$ , а при моделировании гибридных схем использование одновременно обоих принципов;
- интерфейс с пользователем по ходу модельного эксперимента, а также выдача запросов и диагностических сообщений.

Для вывода, наблюдения и обработки результатов моделирования используется другая программа, называемая **постпроцессором моделирования**.

Уже отмечалось, что многие методы расчетов были хорошо отработаны задолго до появления ЭВМ и программно реализованы на самых

ранних стадиях использования ЭВМ. В настоящее время для исследователя, занимающегося моделированием технической системы, значительно более актуально уметь обоснованно выбирать те или иные программные средства расчета, нежели самостоятельно разрабатывать программное обеспечение. Характеристики математического обеспечения оказывают существенное, а иногда и определяющее влияние на возможности и показатели САМ. Важнейшими из них обычно считаются универсальность, алгоритмическая надежность, точность, затраты машинного времени [31, 29].

**Универсальность** – применимость математического обеспечения к широкому классу проектируемых объектов. Высокая степень универсальности нужна для того, чтобы имеющуюся САМ можно было применить для исследования любых или большинства проектируемых объектов. Степень универсальности характеризуется заранее оговоренными ограничениями.

**Алгоритмическая надежность** – свойство компонента математического обеспечения давать при его правильном применении правильные результаты. Алгоритмическую надежность можно оценить вероятностью получения правильных результатов при соблюдении заранее установленных и известных пользователю ограничений на применение метода. Для алгоритмически надежного метода эта вероятность близка к единице.

**Точность** – количественная характеристика применимости математического обеспечения к решению конкретных задач. Точность определяется по степени совпадения расчетных и истинных результатов, обычно с использованием одной из стандартных норм (см. раздел 1.6.). Расчет точности может быть непростой и неоднозначной задачей, если выходной сигнал векторный. Обычно в таких случаях точность рассчитывается как взвешенная сумма точностей по составляющим этого сигнала.

**Затраты машинного времени** обычно выступают как главный ограничивающий фактор при попытках повысить сложность моделируемых объектов и тщательность их исследования. Поэтому требование экономичности по затратам машинного времени традиционно является одним из основных требований к математическому обеспечению САМ.

#### **4.4. Методы построения моделирующих программ**

Современные пакеты визуального компонентного моделирования используют один из двух методов представления информации об исследуемой системе – структурное (блочное) моделирование и физическое мультидоменное моделирование [13]. При этом за каждым из двух методов могут стоять определенные особенности математического ядра.

**Структурным моделированием** называется техника моделирования, основанная на использовании моделей в виде блоков, для которых определены входы и выходы. Соответственно, блоки рассматриваются как преобразователи входных сигналов в выходные. При моделировании линейных систем связи между входными и выходными сигналами устанавливаются посредством задания передаточных функций. Для нелинейных систем эти связи задаются нелинейными алгебраическими или дифференциальными уравнениями. Поскольку структурные блоки имеют выраженные входы и выходы, модели, построенные согласно этой технике, иногда называют направленными сигнальными графами.

Структурное моделирование используется, например, в пакете Simulink (рис. 4.2).

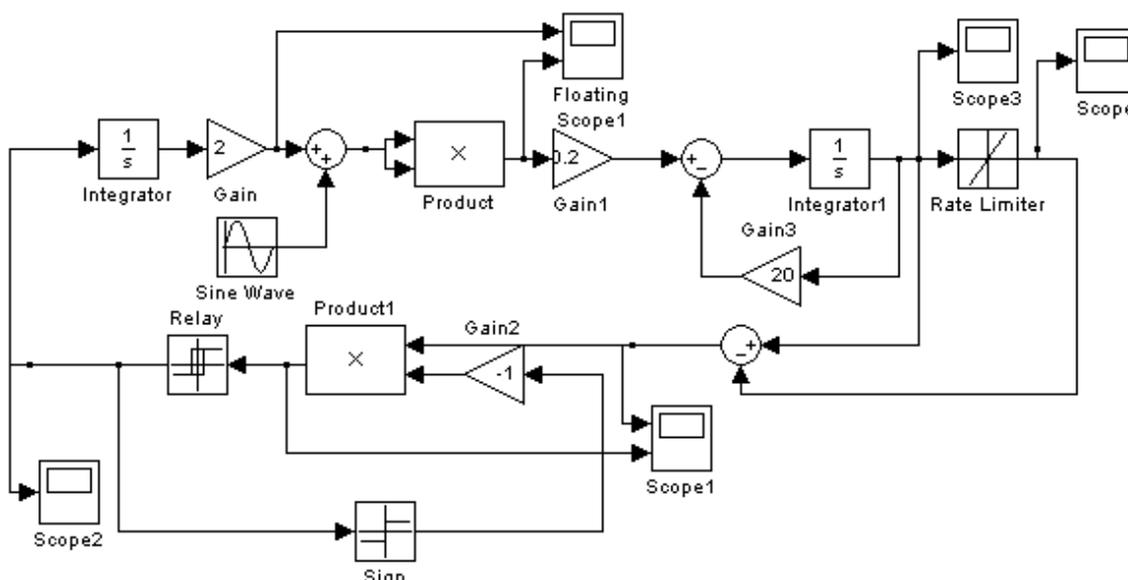


Рис. 4.2. Схема для структурного моделирования экстремальной системы в пакете Simulink

**Физическим мультидоменным моделированием** (иногда просто мультидоменным моделированием [13]) называется техника моделирования, основанная на использовании библиотеки моделей элементов физических устройств, из которых можно составлять физические принципиальные схемы. Поскольку в энергетических цепях поток энергии может менять направление, то для элементов физических схем входы и выходы не определены. Частным случаем данной техники моделирования является использование графов связей.

Идеология физического мультидоменного моделирования заключается в том, что модель любого технического устройства строится как преобразующая энергию цепь (рис. 4.3). В распоряжение пользователя предоставляется библиотека элементов физических устройств.

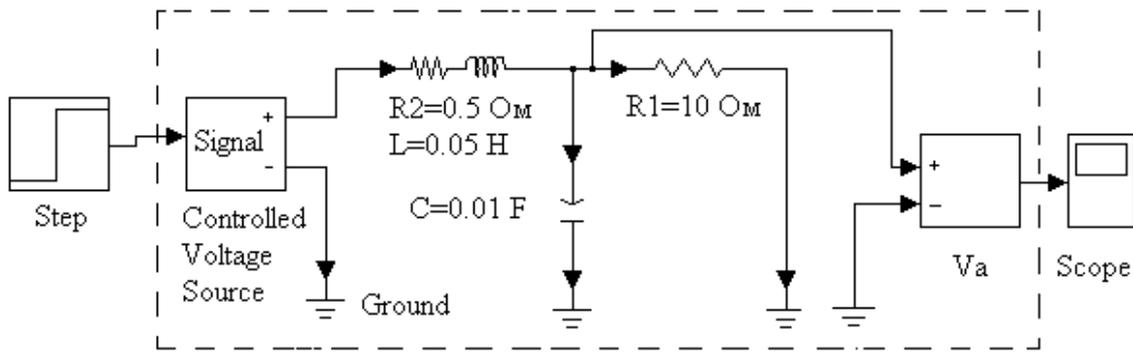


Рис. 4.3. Схема моделирования электрической цепи в пакете SimPowerSystems

В зависимости от уровня иерархии это могут быть библиотеки различных энергетических доменов (электрических, механических, гидравлических и т. д.) или более сложных функциональных узлов, например, механических звеньев и кинематических пар в механических моделях, усилителей, трансформаторов, преобразователей, двигателей – в электрических и электромеханических моделях.

#### 4.4.1. Структурное моделирование

Структурное моделирование может предусматривать два варианта управления процессом моделирования [14]:

- поток команд (Control Flow);
- поток данных (Data Flow).

**Принцип потока команд** – это обычный принцип записи программ в текстовых языках программирования, где инструкции выполняются в той последовательности, в которой они написаны. Если в процессе выполнения очередной инструкции программа обнаруживает, что какие-то данные не определены, это рассматривается как ошибка и влечет остановку программы. Упорядочивание инструкций ложится на программиста. При этом следует понимать, что сам процессор только так и может работать – выполняя команды одну за другой.

Жесткие формы языков программирования, на основе которых формируется моделирующая программа при использовании принципа Control Flow, плохо сочетаются с возрастающим уровнем абстракции графических языков. При любой форме задания графической информации ее необходимо преобразовать в выполняемые инструкции, т. е. последовательность команд для процессора, т. к. код графического языка не может выполняться «строка за строкой». Кроме того, большинство типовых программ, реализующих численные методы, рассчитаны на стандартные формы записи уравнений, которые не всегда возможно соблюдать при графическом задании информации. Это ведет к необхо-

димости автоматически преобразовывать введенные пользователем или построенные графическим интерфейсом уравнения к формам, с которыми могут работать, например, библиотеки программных реализаций численных методов.

В этом случае эффективной является технология потока данных (Data Flow). САМ, работающая на поточной технологии управления процессом моделирования, обязательно содержит блок, упорядочивающий поток данных (информационный поток).

**Технология потока данных** – система программирования, состоящая из исполняемых узлов данных. Узлы выполняются только тогда, когда все необходимые данные поступят на их входы. **Узлы данных** – это просто обобщающее название любого исполняемого элемента программы. Узлы аналогичны операторам, функциям и подпрограммам в традиционных языках программирования. Можно сказать, что поточная модель управления – это среда программирования потока данных. Как правило, эта среда графическая.

На поточной модели управления процессом моделирования основана значительная часть программ математического структурного моделирования динамических систем. Поточная модель управления – основополагающее понятие для таких программ, как VisSim, MBTU, Simulink, Easy5, LabVIEW.

Рассмотрим пример статического информационного потока, приведенный на рис. 4.4.

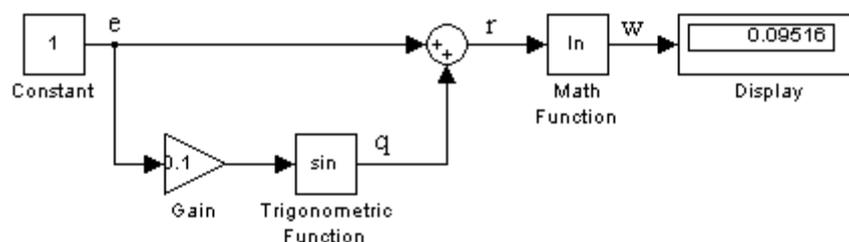


Рис. 4.4. Статический информационный поток в пакете Simulink

Блок-схема описывает набор действий, два варианта которого представлены в табл. 4.1.

Таблица 4.1

Неупорядоченный информационный поток	Упорядоченный программой информационный поток
a) $w = \ln(r)$	1) $e = 1$
b) $e = 1$	2) $q = 0,1\sin(e)$
c) $r = e + q$	3) $r = e + q$
d) $q = 0,1\sin(e)$	4) $w = \ln(r)$

Неупорядоченный информационный поток не может быть выполнен ЭВМ. Упорядоченный – легко выполняется. Обычно упорядочивание информационного потока выполняется одновременно с расчетом модели. Интерпретирующий транслятор последовательно выполняет узлы программы, для которых имеются все необходимые исходные данные. Если реализуется процедура численного интегрирования, то такой расчет повторяется на каждом шаге.

#### **4.4.2. Решатели для структурного и физического мультидоменного моделирования**

Главное различие структурного и физического мультидоменного моделирования кроется не столько в форме задания исходной информации, сколько в используемых методах численного интегрирования дифференциальных уравнений. Обычно со структурным моделированием связывают явный решатель, который представляет собой библиотеку классических подпрограмм численного интегрирования, реализующих явные методы интегрирования.

С физическим мультидоменным моделированием обычно связывают неявный решатель, иногда называемый итерационным. Неявный решатель реализует неявный метод моделирования, при котором на каждом шаге сначала формируется полная нелинейная система алгебраических уравнений, которая далее решается итерационным методом. Для этого может использоваться линеаризованная система алгебраических уравнений, а итерационный процесс – сводиться к одному шагу. Важной особенностью неявных методов (кроме всего прочего) является то, что не требуется искусственно разрывать систему, чтобы организовать поток вычислений. Кроме того, техника неявного интегрирования позволяет не использовать технологию Data Flow, т. к. вектор переменных модели рассчитывается не по частям, а весь сразу.

Вообще говоря, явные и неявные методы интегрирования не привязаны жестко к структурному и физическому мультидоменному моделированию.

В рамках структурного моделирования можно использовать как явные, так и неявные методы. В частности, в системе РЕМОС, которая использует неявные методы интегрирования [6], можно моделировать во временной области как объекты, заданные структурными схемами

(т. е. с использованием направленных звеньев), так и объекты, представленные в виде физических принципиальных схем.

И, наоборот, при задании исходной информации в виде принципиальной схемы можно включить в математическое ядро процедуру преобразования мультидоменной информации к форме направленной структуры и затем использовать те же самые методы явного численного интегрирования, что и в обычном структурном моделировании. По всей видимости, именно так сделано в пакетах SimMechanics и SimPower, которые являются подсистемами Simulink. В результате получен некоторый промежуточный вариант, основным достоинством которого является форма задания исходной информации. При этом сам переход к структуре в общем случае неоднозначен. В простых случаях машина может выполнять его самостоятельно, без обращения к пользователю. В более сложных – пользователь должен помочь программе сформировать наиболее рациональную, с точки зрения затрат на интегрирование, структуру. Например, возникает задача размыкания петель, которая отсутствует при использовании неявных методов.

## Глава 5

# ПАКЕТЫ ВИЗУАЛЬНОГО МОДЕЛИРОВАНИЯ МЕХАТРОННЫХ СИСТЕМ

На сегодняшний день на рынке существует множество инструментальных средств для автоматизированного моделирования технических, в частности мехатронных, систем. Некоторые из них хорошо известны и популярны у российских пользователей, другие появились совсем недавно.

Часть пакетов являются универсальными и могут использоваться для моделирования любых технических, и не только технических, систем. Другие имеют узкую специализацию в какой-либо предметной области.

Возможности многих пакетов в значительной степени перекрываются, и подходы к решению одних и тех же задач у них зачастую примерно одинаковы.

Поскольку освоение даже одного серьезного пакета связано с существенными затратами времени, сил и денег, правильный выбор инструмента в значительной степени определит успешность исследований.

Опираясь на такие важнейшие показатели, как назначение и возможности пакета, состав библиотек и принципы построения моделей, методы интегрирования и средства визуализации результатов, проведем достаточно приближенную классификацию инструментальных средств, которые могут в той или иной мере использоваться для моделирования мехатронных систем.

### 5.1. Классификация пакетов моделирования технических систем

Структура современных инструментальных средств (пакетов) для моделирования технических систем представлена на рис. 5.1.

Под **специализированными пакетами** понимаются программные средства, которые долгое время создавались и развивались на конкретных предприятиях и отраслях и были ориентированы на специфические понятия конкретной прикладной области (механики, теплотехники, электроники и т. д.). Написанные на языках Fortran, С и т. п., эти пакеты уходят в прошлое, заменяясь предметно-ориентированными компонентными пакетами. Связано это с тем, что специализированные пакеты требуют тесного сотрудничества программиста и специалиста в предметной области, а лучше хорошего владения специалистом искусством

программирования. Переход к компонентным пакетам позволяет разорвать эту связь. Кроме того, специализированные пакеты с трудом поддаются модернизации, в них сложно использовать современные программы визуализации и обработки результатов экспериментов и т. п.



Рис. 5.1. Классификация пакетов моделирования технических систем

**Математические пакеты**, такие как Mathcad, Maple, Mathematica, хорошо приспособлены к проведению расчетов в естественно-научных дисциплинах, когда модель задана в аналитической форме. Удобство варьирования параметров в сочетании с заранее определенной процедурой обработки и визуализации результатов существенно облегчает исследования. В таких многовариантных расчетах накладные расходы, связанные с написанием на языке пакета специальной программы, управляющей экспериментом, окупаются той легкостью, с которой возможно повторить все вычисления заново при внесении изменений в исходную модель. Программирование сводится к написанию относительно небольших по объему программ, состоящих в основном из макрооператоров.

С точки зрения моделирования мехатронных объектов основным и, пожалуй, единственным достоинством систем компьютерной математики является математическая прозрачность вычислений и легкость создания объектов, осуществляющих математические вычисления. К числу недостатков можно отнести отсутствие следующих принципиально важных возможностей:

- автоматизация построения математической модели;
- компонентное моделирование с применением достаточно большого количества типовых блоков;

- быстрая модификация модели;
- создание предметно-ориентированной среды;
- оперативное изменение метода моделирования и т. д.

В результате применение систем компьютерной математики ограничивается решением простых задач или задач, где главное – прозрачность вычислений.

**Пакеты компонентного моделирования** в основном ориентированы на численные эксперименты и являются в настоящее время доминирующими в процессах проектирования технических объектов. Они дают пользователю возможность не заботиться о программной реализации модели, как о последовательности исполняемых операторов, и, тем самым, создают на компьютере некоторую удобную среду, в которой можно создавать виртуальные системы и проводить эксперименты с ними.

Пакеты компонентного моделирования по способам их применения или технологии моделирования можно разделить на две группы.

Так называемые **универсальные пакеты** ориентированы на определенный класс математических моделей и применимы для любой прикладной области, в которой эти модели справедливы. Основу универсального пакета составляют библиотеки компонентов общего назначения. В этих пакетах используются разнообразные коллекции численных методов, способные справиться с широким спектром задач. Как правило, универсальные пакеты обладают развитыми средствами визуализации, обеспечивающими показ изучаемого явления с разных сторон, а не одним, принятым в конкретной области, способом.

**Предметно-ориентированные пакеты** предназначены для решения промышленных и научно-исследовательских задач в конкретной предметной области. Библиотеки моделей компонентов таких пакетов содержат хорошо изученные и отлаженные модели из довольно узкой предметной области, которые лишь накапливаются, модифицируются и приспособляются для решения конкретных задач. В результате накопленная база моделей со временем приобретает большую ценность. Спектр методов решения задач проектирования также ограничен известными и хорошо отработанными инструментами (возможно, ориентированными на узкий класс задач), в эффективности и надежности которых у пользователей нет сомнений. Как правило, предметно-ориентированные пакеты требуют серьезных усилий для их освоения, а также знаний в конкретной предметной области. Стоимость этих пакетов достаточно высока, что обычно заставляет пользователя работать с каким-либо одним пакетом в течение длительного времени, всячески расширяя его возможности.

Следует заметить, что между универсальными и предметно-ориентированными пакетами в настоящее время нет четкой границы. Часто разница лишь количественная. Добавление к универсальному пакету соответствующего набора специализированных модулей, прежде всего библиотек моделей компонентов, превращает этот пакет в предметно-ориентированную среду моделирования. Примером подобного подхода может служить появление предметных расширений пакета Simulink – SimPower, SimMechanics и т. п. Учитывая открытость системы, каждый пользователь может добавить к готовым моделям то, что ему нужно, создав собственную предметно-ориентированную среду.

По принципам представления исходной модели среди пакетов компонентного моделирования можно выделить две основные группы:

- 1) пакеты структурного (или блочного) моделирования;
- 2) пакеты физического мультидоменного моделирования.

Элементарные блоки пакетов структурного моделирования обладают направленным действием, последующий блок не влияет на предыдущий. К достоинствам этого подхода следует отнести, прежде всего, простоту создания не очень сложных моделей даже не слишком подготовленным пользователем. Другим достоинством является эффективность реализации элементарных блоков и простота построения эквивалентной системы. В то же время при создании сложных моделей приходится строить довольно громоздкие многоуровневые блок-схемы, не отражающие естественной структуры моделируемой системы. Наиболее известными представителями пакетов визуального структурного моделирования являются MATLAB/Simulink, EASY5, VisSim, AnyLogic.

Пакеты физического мультидоменного моделирования позволяют использовать как ориентированные, так и неориентированные компоненты и связи. Подход очень удобен и естественен для описания типовых блоков физических систем. К пакетам физического моделирования можно отнести Multisim, DYNAST, 20-sim, Dymola.

Некоторые авторы выделяют в качестве третьей группы **пакеты, предназначенные для моделирования гибридных систем** [19]. Эти пакеты позволяют очень наглядно и естественно описывать мехатронные системы со сложной логикой переключений. К этому направлению относится пакет Shift, а также отечественный пакет Model Vision Studium [21].

## 5.2. Пакеты структурного моделирования

Рассмотрим кратко возможности и особенности некоторых универсальных и достаточно распространенных пакетов визуального структурного моделирования, которые могут быть использованы для моделирования мехатронных систем.

К числу универсальных, не ориентированных на конкретные прикладные области пакетов для моделирования технических систем можно отнести пакет MATLAB/Simulink, а также построенные по его образу и подобию пакеты VisSim, MBTU.

Данные пакеты предназначены для моделирования и исследования динамических систем в широком понимании этого термина, включая и дискретные, и непрерывные, и гибридные модели. Их отличает относительная простота и интуитивная ясность входных языков в сочетании с разумными требованиями к мощности компьютеров.

### **5.2.1. Пакет MATLAB/Simulink**

Система MM MATLAB/Simulink является в настоящее время одним из наиболее популярных инструментов численных расчетов и применяется в различных областях знаний [7, 8].

Главная особенность среды MATLAB – тщательная проработанность и отлаженность всего богатейшего арсенала средств и методов. Возможности MATLAB закрывают большую часть потребностей разработчиков систем управления в различных областях техники. Обширный набор численных методов в сочетании с мощными средствами графической визуализации делает MATLAB универсальным инструментом инженерных расчетов и научных исследований.

Серьезными достоинствами среды MATLAB являются ее открытость и расширяемость. Большинство команд и функций системы реализованы в виде текстовых m-файлов (файлов с расширением .m) и файлов на языке C, причем все файлы доступны для модификации. Это дает пользователю возможность создавать не только отдельные файлы, но и целые библиотеки файлов, формируя, таким образом, собственную предметно-ориентированную среду моделирования.

Такой подход используется и самими разработчиками среды MATLAB, что нашло отражение в создании десятков пакетов прикладных программ, намного расширивших сферы применения системы.

Важнейшей составляющей среды MATLAB является пакет структурного моделирования динамических систем Simulink. Разработку этого пакета можно рассматривать как принципиальную модернизацию среды MATLAB, в результате которой она приобрела все черты современной САМ – компонентное моделирование, графическая форма задания информации об объекте и т. п. Simulink настолько органично интегрирован с системой MATLAB, что, рассматривая современные средства исследования технических систем, логично говорить о среде визуального компонентного моделирования MATLAB/Simulink.

На базе пакета Simulink разработаны дополнительные библиотеки блоков для разных областей применения (например, Power System Blockset – моделирование электротехнических устройств, Digital Signal Processing Blockset – набор блоков для разработки цифровых устройств, Control System Toolbox – пакет для разработки систем управления и т. д.). Можно ожидать, что такая политика будет продолжена и в следующих модификациях среды MATLAB появятся новые предметно-ориентированные библиотеки.

При моделировании в Simulink исследователь может воспользоваться достаточно широкой палитрой методов решения дифференциальных уравнений, а также выбрать способ изменения модельного времени (с фиксированным или переменным шагом), что позволяет проводить моделирование для широкого круга систем, включающих непрерывные, дискретные и гибридные системы любой размерности. Отдельное направление исследований связано с возможным переходом в область линеаризованных систем, где реализован весь арсенал методов линейного анализа. Средства визуализации дают возможность следить за процессами, происходящими в системе. Для этого используются специальные устройства наблюдения, входящие в состав библиотеки Simulink. Результаты моделирования могут быть представлены в виде графиков или таблиц.

Базовая библиотека Simulink содержит более 200 блоков, наиболее часто встречающихся при моделировании различных систем. Дополнительные библиотеки позволяют расширить возможности Simulink для применения в аэрокосмической области, в обработке сигналов, связи и в других приложениях.

Библиотеки блоков MATLAB/Simulink могут пополняться за счет подпрограмм, написанных как на языке среды MATLAB, так и на языках C++, Fortran и Ada. Кроме того, в данном пакете реализована процедура инкапсуляции, являющаяся еще одним удобным для пользователя средством расширения библиотек.

Недостатки MATLAB/Simulink являются общими для всех инструментов структурного моделирования. Как во всех подобных пакетах, при создании сложных моделей приходится строить довольно громоздкие многоуровневые блок-схемы, не отражающие естественной структуры моделируемой системы. Частично этот общий недостаток пакетов структурного моделирования устраняется использованием приложений, в которых реализован принцип физического мультидоменного моделирования (таких как SimMechanics и SimPowerSystems, обеспечивающих возможность построения механических и электрических систем соответственно), а также тем, что разработчики большинства современных

пакетов физического мультидоменного моделирования предусматривают возможность подключения библиотек Simulink.

### **5.2.2. Пакет VisSim**

VisSim – пакет компонентного визуального моделирования фирмы Visual Solutions, предназначенный для разработки и моделирования динамики непрерывных, дискретных и гибридных систем [12]. По своим функциональным возможностям и способу задания исходной информации VisSim близок пакету MATLAB/Simulink. Simulink получил несколько большее распространение, особенно в России, тем не менее, VisSim имеет немало почитателей и занимает заметную долю рынка инженерных пакетов.

VisSim – один из наиболее ярких представителей систем, реализующих концепцию структурного моделирования. Основным инструментом задания моделей являются типовые блоки «вход – состояние – выход». Подобно MATLAB/Simulink, каждый типовой блок VisSim реализует математическую модель или обеспечивает визуализацию того или иного явления, процесса или устройства. Базовая библиотека VisSim содержит более 100 линейных и нелинейных блоков, позволяющих моделировать весьма сложные системы и сгруппированных по функциональному признаку.

Большинство библиотечных блоков-подпрограмм написаны на языке C++, однако пакет предоставляет возможность создавать пользовательские библиотеки практически на любом процедурном языке программирования. Процедура инкапсуляции дает возможность пользователю VisSim создавать новые блоки (подсистемы), основываясь на уже существующих блоках.

Как и Simulink, VisSim имеет широкий набор инструментов оценки качества, устойчивости, синтеза, коррекции, оптимизации, линеаризации, отладки объектов в контуре модели и программирования цифровых сигнальных процессоров.

Решатель VisSim интерпретирующего типа, функционирует в динамическом режиме с возможностью online-взаимодействия с оборудованием реального времени. В составе пакета решателя VisSim реализованы как явные, так и неявные методы интегрирования.

Возможности моделирования гибридных систем в пакете VisSim примерно аналогичны Simulink, если не учитывать пакет расширения StateFlow. VisSim не работает с понятием карты состояния Харелла. Он предназначен прежде всего для моделирования непрерывных систем. Общим принципом гибридного моделирования в VisSim является использование готовых гибридных блоков или переключение заранее

заготовленных альтернативных участков блок-схем. Однако не исключено, что в ближайшее время появится соответствующее расширение, аналогичное StateFlow.

В качестве ограничений пакета VisSim можно отметить отсутствие альтернатив языку блок-схем. Однако для моделирования технических систем это не является столь уж серьезным недостатком.

Кроме того, VisSim не соответствует требованиям объектно-ориентированного моделирования, не поддерживает наследование, динамическое изменение количества объектов. Тем самым, он не дает возможности моделировать систему с меняющимся во времени количеством объектов.

Подобно Simulink, пакет VisSim включает в себя ряд дополнительных модулей, расширяющих его возможности в конкретных предметных областях, а также инструменты интеграции с пакетами MATLAB, Mathcad и транслятор проектов программы Simulink.

### **5.2.3. Пакет МВТУ**

Из отечественных программных продуктов визуального моделирования технических систем наиболее развитым в настоящее время является пакет, или, как его называют авторы, программный комплекс «Моделирование в технических устройствах» (ПК «МВТУ»), созданный в МГТУ им. Н.Э. Баумана [17, 18].

Программный комплекс МВТУ реализует структурное моделирование и предназначен для исследования динамики и проектирования разнообразных технических систем и устройств. Авторы считают его альтернативой программным продуктам Simulink, VisSim и др. Действительно, МВТУ обладает всеми основными достоинствами современных средств визуального моделирования, хотя его функциональные возможности существенно уже, чем у вышеупомянутых пакетов. Удобный редактор структурных схем, обширная библиотека типовых блоков и встроенный язык программирования позволяют реализовывать модели высокой степени сложности, обеспечивая при этом наглядность их представления. Для отечественных пользователей удобство работы с МВТУ обусловлено также русскоязычным интерфейсом и наличием обширной документации на русском языке.

Судя по литературным источникам, МВТУ успешно применяется для проектирования систем автоматического управления, следящих приводов и роботов-манипуляторов, ядерных и тепловых энергетических установок. Он может функционировать в многокомпьютерных моделирующих комплексах, в т. ч. и в режиме удаленного доступа к технологическим и информационным ресурсам.

МВТУ реализует такие режимы работы, как моделирование процессов в непрерывных, дискретных и гибридных динамических системах, в т. ч. при наличии обмена данными с внешними программами и устройствами; оптимизация показателей качества; анализ и синтез по линейным моделям; контроль и управление с использованием виртуальных аналогов пультов управления с измерительными приборами и управляющими устройствами.

Основными особенностями МВТУ, делающими его современным программным продуктом и ставящими его в один ряд с аналогичными по назначению зарубежными пакетами, являются следующие:

- принцип вложенности структур, что особо актуально при моделировании сложных динамических систем;
- наличие достаточно полной общетехнической библиотеки и ряда специализированных библиотек типовых блоков;
- обширный набор алгоритмов численного интегрирования;
- открытость, дающая пользователю возможность расширить состав личной библиотеки путем создания новых типов блоков.

Библиотека типовых блоков состоит из общетехнической и ряда обладающих значительными по количеству готовых моделей специализированных библиотек, таких как «Гидроавтоматика», «Роботы», «Электромашины» и др.

Для пользователя МВТУ существуют две возможности расширить библиотеку математических моделей – создать в виде графического представления блок-схему, которая будет потом использоваться как типовой блок (макроблок), или создать новый типовой блок, описав его поведение с помощью языка программирования, аналогичного языку системы MATLAB. Такие подходы являются типовыми и используются во многих программных комплексах, реализующих методы структурного моделирования.

На рис. 5.2 приведена модель электрической схемы в МВТУ, все элементы которой являются замаскированными макроблоками, соединенными между собой векторными линиями связи.

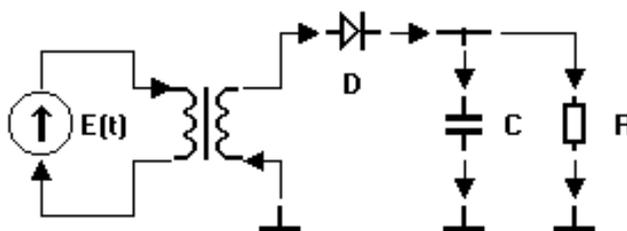


Рис. 5.2. Пример построения модели электрической схемы выпрямителя в пакете МВТУ

Векторные переменные имеют две компоненты: напряжение и ток. Модели некоторых элементов схемы показаны на рис. 5.3.

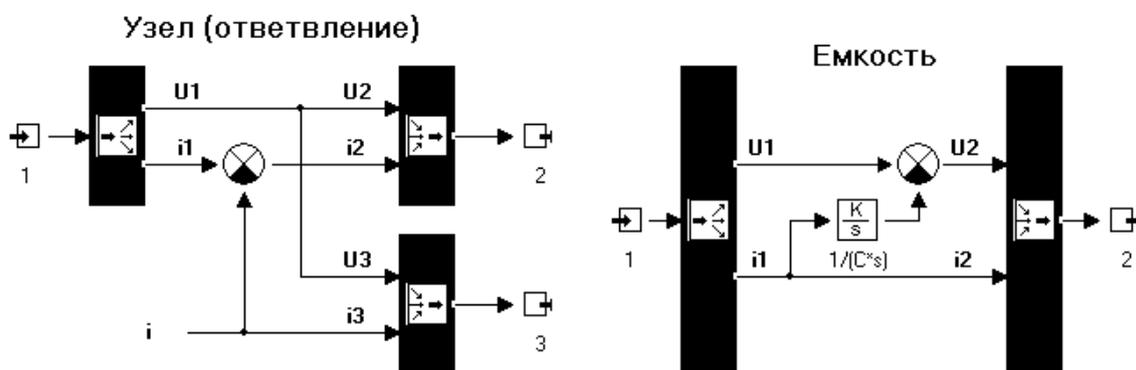


Рис. 5.3. Эквивалентные схемы узла и емкости (макроблоки)

В решателе интерпретирующего типа реализованы 10 явных и 6 неявных методов численного интегрирования, среди которых есть новые оригинальные методы, позволяющие эффективно интегрировать жесткие системы.

### 5.3. Пакеты физического мультидоменного моделирования

Пакеты физического мультидоменного моделирования многими специалистами считаются более перспективными инструментами для исследования технических систем. Во-первых, они используют форму задания исходной информации, которая гораздо ближе и понятнее специалистам в предметных областях. Во-вторых, исследователь избавлен от необходимости проводить структурные преобразования модели, с тем чтобы перейти к структуре направленных блоков. И, наконец, большинство современных пакетов физического моделирования не только допускают использование направленных блоков, но и могут сопрягаться с классическими пакетами структурного моделирования, например, с Simulink.

Первоначально пакеты физического моделирования разрабатывались для моделирования электрических и электронных схем. Структура таких схем проще, чем механических. Каждая энергетическая связь содержит лишь две переменные – ток и напряжение. Именно поэтому современные пакеты компьютерного моделирования и анализа схем электронных устройств, такие, как Multisim, являются наиболее развитыми и отработанными.

Для нас представляют интерес пакеты, способные моделировать, прежде всего, механическую часть мехатронной системы, как наиболее сложную с точки зрения формирования математической модели и с точки зрения численного интегрирования и визуализации результатов.

### **5.3.1. Пакет Modelica/Dymola**

Modelica представляет собой среду визуального моделирования, включающую универсальный объектно-ориентированный язык Modelica для моделирования сложных физических систем и собственно инструментальные средства, такие как пакеты Dymola или MathModelica. Язык Modelica, наряду с ASCEND, Smile, VHDL-AMS и др., относится к группе так называемых некаузальных языков, т. е. языков физического моделирования.

Как и во всех рассматриваемых пакетах, в Modelica используется компонентный принцип задания исходной информации об исследуемой системе, однако используемые компоненты имеют не входы и выходы, а выводы, или контакты, как правило, имеющие ясный физический смысл. Это делает пакет привлекательным для специалистов нематематического профиля и более простым для понимания и использования в целом.

Modelica имеет хорошую техническую поддержку со стороны производителя, для нее существует большое количество библиотек готовых компонентов. Постоянно происходит как дополнение уже существующих библиотек, так и разработка новых, охватывающих многие отрасли науки.

Пакет Dymola (Dynamic Modeling Laboratory), поддерживающий язык моделирования Modelica, является комплексным инструментом для моделирования и исследования сложных систем в таких областях, как мехатроника, автоматика, аэрокосмические исследования и др.

Возможность объединения в одной модели компонентов различной физической природы позволяет строить модели сложных систем, лучше соответствующие реальности, и получать более точные и прозрачные результаты.

Стандартная библиотека Dymola включает элементарные компоненты, относящиеся к электротехнике, механике, гидравлике и т. д. Специализированные библиотеки содержат модели устройств и явлений для конкретной области.

Например, MultiBody Library содержит трехмерные механические компоненты, предназначенные для моделирования роботов, искусственных спутников Земли или транспортных средств. Это модели твердых тел, шарниров, источников и средств анимации. Такие же библиотеки разработаны для электротехники, гидравлики и т. п.

Гибкость и открытость пакета Dymola дают пользователям возможность создавать свои собственные библиотеки моделей, описывая их на внутреннем языке описания блоков, или модифицировать уже существующие модели под свои конкретные нужды.

Кроме собственного языка, Dymola (Modelica) поддерживает интеграцию с такими программными средами, как Fortran, C, Simulink, и некоторыми др.

Возможность взаимодействия разработанных моделей с системой MATLAB/Simulink позволяет объединить сильные стороны структурного и физического моделирования.

### 5.3.2. Пакет 20-sim

Пакет моделирования 20-sim предназначен для моделирования динамики технических систем – механических, электрических, гидравлических, а также сложных систем, содержащих механические, электрические или гидравлические компоненты.

Пакет поддерживает визуальное компонентное моделирование: модель вводится обычным для пакетов такого типа методом – компоненты выбираются из библиотеки, переносятся на рабочий стол и соединяются связями. 20-sim позволяет работать как с ориентированными блоками, так и с неориентированными, что очень удобно при моделировании систем управления физическими объектами и установками.

Библиотеки базовых компонентов включают библиотеку типовых одномерных направленных звеньев – источников, измерителей, преобразователей, блоков математических операций и т. д.; библиотеку элементов графов связей, как простейших универсальных энергетических компонентов; библиотеки механических, электрических, гидравлических, тепловых элементов. В качестве примера на рис. 5.4 приведена модель системы, где энергетическая часть представлена графом связей, а управляющая – блок-схемой, включающей направленные звенья.

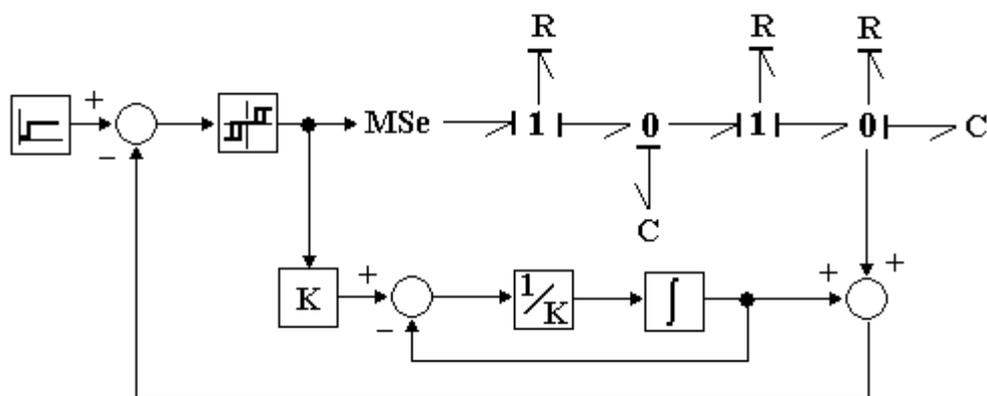


Рис. 5.4. Схема системы в пакете 20-sim с использованием графов связей

Пакет 20-sim имеет ряд расширений, одним из которых является Mechatronics Toolbox, включающий в основном модели различных электромеханических устройств. Важно отметить, что большинство моделей,

представленных в данном расширении, относятся к конкретным мехатронным устройствам, поддерживаются и сопровождаются производителями технических средств. Они точны, подробны, закрывают все аспекты функционирования устройств и с этой точки зрения очень полезны проектировщикам, занимающимся разработкой мехатронных систем.

Пакет допускает создание пользователем собственных библиотек. В качестве внутреннего языка описания моделей используется язык, подобный языку среды Maple, достаточно понятный и простой в изучении.

20-sim имеет мощный решатель, позволяющий решать как ОДУ, так и дифференциально-алгебраические уравнения, используя одношаговые и многошаговые методы различных порядков.

Как и положено современному пакету визуального моделирования, 20-sim имеет развитые средства анимации, в т. ч. 3D анимацию.

#### **5.4. Пакеты среды MATLAB для моделирования мехатронных систем**

Наиболее распространенными, универсальными и доступными в учебном процессе являются пакеты моделирования технических систем, работающие в среде MATLAB. Приложения среды MATLAB позволяют моделировать мехатронные системы, включающие механические, электрические и информационные элементы. Среди этих приложений имеет смысл выделить пакеты визуального моделирования SimMechanics, SimPowerSystems, StateFlow.

##### ***5.4.1. Принципы моделирования механических систем в пакете SimMechanics***

Пакет SimMechanics предназначен для технического проектирования и моделирования пространственных механизмов и способен моделировать поступательное и вращательное движение тел в трехмерном пространстве. Для этого SimMechanics снабжен набором инструментов для описания массовых и геометрических свойств твердых тел, их возможных движений, кинематических ограничений, систем координат, источников внешних воздействий и средств измерения движений.

В отличие от Simulink, формирование моделей в SimMechanics соответствует концепции физического мультидоменного моделирования. Модель представляет не математическое описание движения механизма, т. е. алгебраические и дифференциальные уравнения, хотя и выраженные в графической форме, а структуру механизма, геометрические и кинематические отношения между составляющими его телами.

Связи между основными компонентами SimMechanics – твердыми телами и кинематическими парами – имеют энергетический векторный характер. Они могут включать линейные и угловые скорости, силы, мо-

менты и перемещения. К телам и кинематическим парам могут подключаться компоненты, играющие роль приводов и датчиков. Через эти компоненты на механическую систему могут быть поданы внешние воздействия. Через них же механическая часть может быть подключена к информационно-управляющей системе, для моделирования которой может быть использован пакет Simulink.

Кроме основных элементов, библиотека блоков SimMechanics содержит блоки ограничений, которые ограничивают движения тел относительно друг друга; блоки начальных условий, которые определяют начальное состояние механизма; блоки приводов, которые определяют силы или движения, прикладываемые к соединениям и телам; блоки датчиков.

На рис. 5.5 приведена блок-схема простейшей механической системы (маятника), включающая такие физические компоненты, как блок тела (body), блок одномерного вращательного движения (revolute), блок основания (ground).

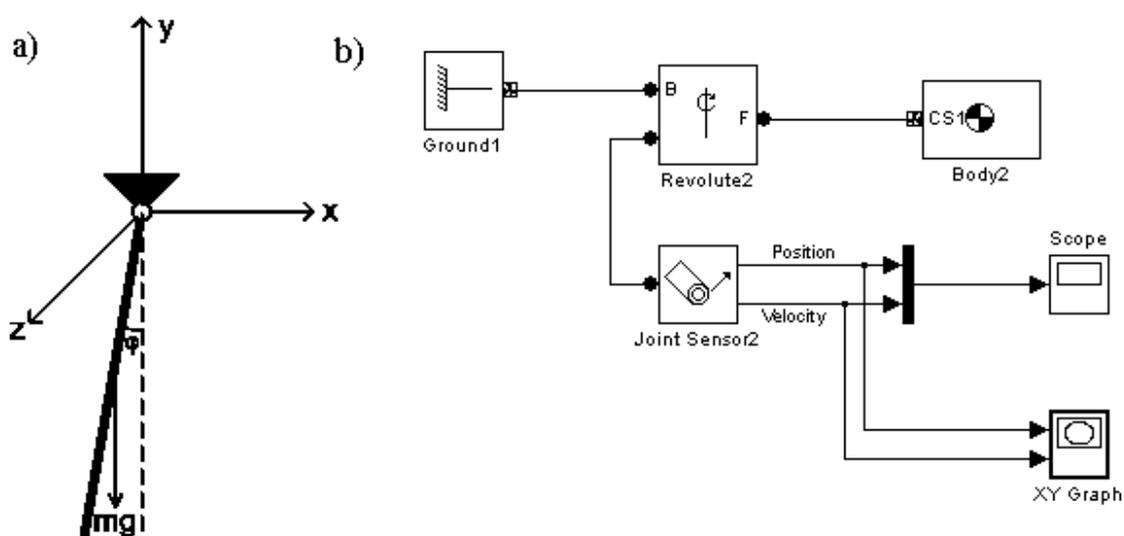


Рис. 5.5. Модель маятника в пакете SimMechanics:  
*a* – кинематическая схема, *b* – схема моделирования

Энергетические связи между телами и кинематическими парами определяют кинематику механизма. Они могут быть векторными и скалярными. Например, на рис. 5.5, *b* связи между основанием, вращательной кинематической парой и телом являются векторными, т. к. определяют передачу энергии во вращательном движении по трем осям системы координат.

Важнейшим достоинством пакета SimMechanics является возможность объединения блоков SimMechanics и блоков Simulink в единой схеме. В частности, блоки привода SimMechanics (Actuator blocks) могут

подключаться к стандартным выходным портам Simulink, как это показана на рис. 5.5, *b*. Таким образом, есть возможность смоделировать в Simulink источник управляющего сигнала и осуществлять управление телом через блок привода. Точно так же блоки датчиков SimMechanics содержат выходные порты, через которые можно соединить модель SimMechanics с блоками Simulink.

Возможности настройки режимов моделирования обеспечивают следующие способы анализа, каждый из которых соответствует специфическому типу модели.

1. **Прямой режим исследования динамики механизма.** В данном режиме вычисляют положения и скорости тел системы на каждом временном шаге, с учетом начальных условий по положениям и скоростям тел и действия любых сил, приложенных к системе.

2. **Инверсный режим исследования динамики механизма.** В данном режиме вычисляются силы, необходимые для того, чтобы получить заданную скорость для каждого тела разомкнутой системы.

3. **Кинематический режим.** Вычисляются силы, которые требуются для получения заданной скорости для каждого тела замкнутой системы.

4. **Режим балансировки.** Этот режим – вариант прямого режима исследования динамики, основанный на использовании для исследования модели команды *trim* Simulink, которая позволяет находить установившиеся решения для исследуемой модели.

#### **5.4.2. Пакет моделирования электрических систем SimPowerSystems**

Пакет SimPowerSystems является, как и SimMechanics, расширением среды MATLAB. Он работает под управлением Simulink, используя тот же способ визуального компонентного программирования с применением «энергетических» компонентов.

SimPowerSystems является современным инструментом проектирования, позволяющим ученым и инженерам быстро и легко строить и исследовать модели энергетических систем. Используемый графический ввод информации дает возможность задать топологию электрической цепи. Анализ может включать взаимодействие электрической цепи с механическими, тепловыми, управляющими и другими элементами. Это возможно потому, что все электрические части модели взаимодействуют с обширной Simulink библиотекой моделирования.

Библиотеки SimPowerSystems содержат модели стандартного энергетического оборудования типа источников энергии, трансформаторов,

линий передачи энергии и силовой электроники. Все модели хорошо проработаны, и их достоверность весьма высока.

С точки зрения моделирования мехатронных систем следует особенно выделить библиотеку, включающую модели различных электрических машин – постоянного тока, синхронных, асинхронных. Объединение в единой схеме блоков SimPowerSystems и блоков Simulink открывает возможности исследования как самих машин, так и сложных систем электропривода.

В качестве примера на рис. 5.6 представлена схема набора модели для получения переходной функции простой электрической цепи в пакете SimPowerSystems. Часть рисунка, обведенная пунктиром, относится к энергетической части схемы и включает элементы, передающие мощность. Эти элементы входят в библиотеки SimPowerSystems. Модель ступенчатого входного воздействия (Step) и модель индикатора (Scope) взяты из библиотек Simulink. В результате источник постоянного напряжения управляется входным сигналом. Выходное напряжение также преобразуется в сигнал и подается на индикатор. Блоки управляемого источника и измерителя играют роль интерфейсов между информационной и энергетической частями модели.

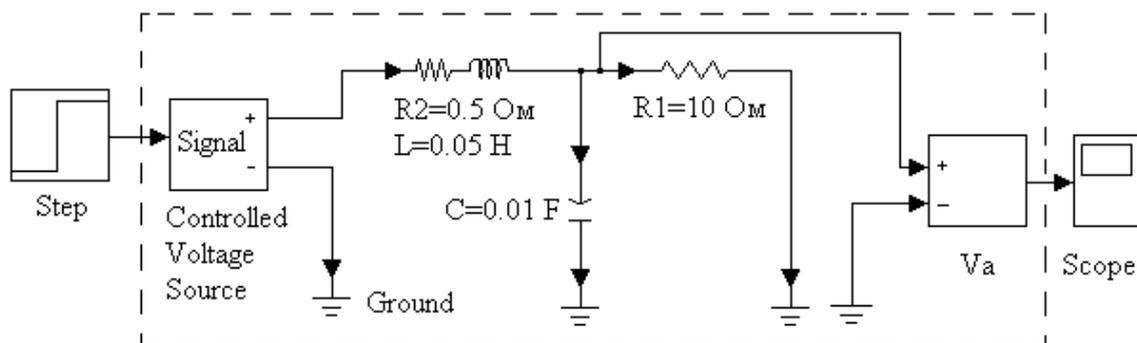


Рис. 5.6. Схема моделирования электрической цепи в пакете SimPowerSystems

#### 5.4.3. Пакет моделирования гибридных систем StateFlow

В разделе 3.3. отмечалось, что особенность исследования динамики гибридных систем состоит в том, что фазовое пространство этих систем разбивается на области с различным поведением, при этом фазовая траектория в зависимости от происходящих событий переходит из одной области фазового пространства в другую. Еще недавно единственным способом изучения гибридных систем было исследование их отдельных фаз или режимов и «склеивание» общего поведения вручную. В настоящее время появились пакеты, позволяющие моделировать глобальное поведение таких объектов [7, 18]. Одним из эффективных и наибо-

лее доступных инструментов численного моделирования систем, управляемых событиями, является пакет StateFlow, входящий в состав среды MATLAB.

StateFlow используется вместе с пакетом Simulink и позволяет моделировать сложные событийно управляемые системы, основываясь на теории конечного автомата. При этом MATLAB обеспечивает доступ к данным, программированию высокого уровня и инструментальным средствам визуализации; Simulink поддерживает моделирование непрерывных и дискретных динамических систем в среде графических блок-схем; диаграммы StateFlow расширяют возможности Simulink по моделированию сложных, управляемых событиями систем.

Традиционной формой представления конечного автомата является таблица истинности, описывающая логику поведения системы [11]. Однако с точки зрения визуального моделирования более удобным подходом к проектированию управляемых событиями систем является описание в терминах переходов между состояниями. На этом подходе основано построение графических представлений – диаграмм переходов.

Диаграмма StateFlow (SF-диаграмма) в нотации обозначений конечного автомата, предложенной Дэвидом Харелом [7, 19], – графическое представление конечного автомата, где состояния и переходы формируют базовые конструктивные блоки системы. Графический интерфейс пакета StateFlow позволяет создавать SF-диаграммы динамического типа. Это означает, что в ходе моделирования на SF-диаграмме отражаются все изменения модели: например, показывается, какие переходы реализуются и по каким условиям изменяются цвета блоков в зависимости от их активности и т. п. Это дает возможность визуально отслеживать поведение системы в ходе моделирования, существенно повышая степень наглядности. Комбинация MATLAB-Simulink-StateFlow является мощным универсальным инструментом моделирования систем, управляемых событиями.

Обычно при моделировании мехатронных систем SF-диаграмма встроена в Simulink-модель и работает совместно с ней, обмениваясь сигналами (данные, события) через интерфейс. На рис. 5.7 показана схема, в которой источник переменного напряжения подключен к нагрузке – осциллографу – через управляемый коммутатор. Переключатель должен работать следующим образом: при положительном напряжении на выходе генератора он должен подключать это напряжение на вход индикатора; если напряжение становится меньше нуля, на индикатор должен передаваться сигнал, равный 2.

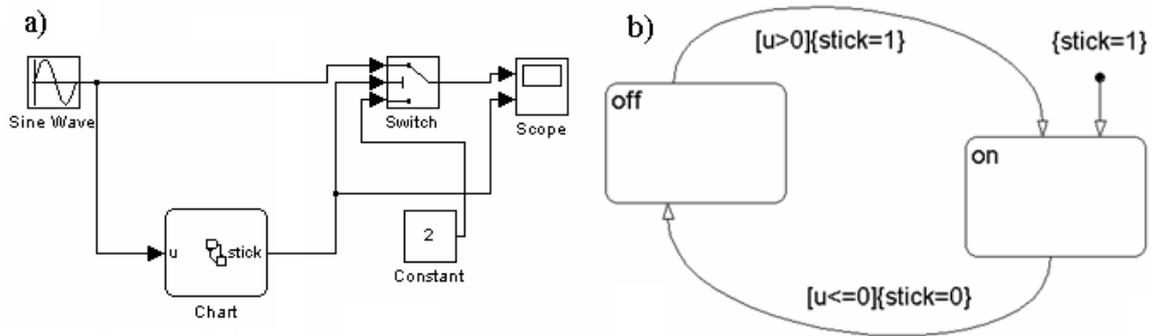


Рис. 5.7. Система управления коммутатором:  
*a* – блок-схема модели, *b* – SF-диаграмма

Управление осуществляется SF-диаграммой в функции состояния сигнала источника. На вход диаграммы поступает сигнал от генератора  $u$ . Выходом диаграммы является сигнал управления переключателем  $Stick$ . С точки зрения нотаций StateFlow сигналы  $u$  и  $Stick$  являются данными. При этом  $u$  – входные данные,  $Stick$  – выходные. SF-диаграмма включает два состояния: *On* – включено и *Off* – выключено. По умолчанию диаграмма переходит в состояние *On*, переменной  $Stick$  присваивается значение 1, и переключатель подключает источник к индикатору. При выполнении условия  $u \leq 0$  переменной  $Stick$  присваивается 0, и переключатель подает на вход индикатора сигнал уровня 2.

## СПИСОК ЛИТЕРАТУРЫ

1. Арайс Е.А., Дмитриев В.М. Автоматизация моделирования многосвязных механических систем. – М.: Машиностроение, 1987. – 240 с.
2. Бусленко Н.П. Моделирование сложных систем. – М.: Главная редакция физико-математической литературы, 1978. – 399 с.
3. Введение в математическое моделирование: учеб. пособ. / В.Н. Ашихмин и др.; под ред. П.В. Трусова. – М.: «Интернет Инжиниринг», 2000. – 336 с.
4. Веников В.А., Веников Т.В. Теория подобия и моделирования. – М.: Высшая школа, 1998. – 453 с.
5. Виллемс Я.К. От временного ряда к линейной системе // Теория систем. Математические методы и моделирование. – М.: Мир, 1989. – 235 с.
6. Гончаров В.И., Петерс Д.П., Вадутова Ф.А. Проектирование исполнительных систем роботов. – Томск: Изд-во ТПУ. – 96 с.
7. Дьяконов В.П. SIMULINK 4. Специальный справочник. – Санкт-Петербург, 2002. – 518 с.
8. Дьяконов В.П. MATLAB 6/6.1/6.5 + SIMULINK 4/5 в математике и моделировании. Полное руководство пользователя. – М.: СОЛОН-Пресс. – 2003. – 576 с.
9. Заде Л.А., Дезоер Ч.А. Теория линейных систем. Метод пространства состояний: пер. с англ. / под ред. Г.С. Поспелова. – М.: Наука, 1970. – 703 с.
10. Имаев Д.Ч. и др. Анализ и синтез систем управления. – Сургут: Информ. издат. центр Сургутского гос. ун-та, 1998. – 169 с.
11. Карпов Ю.Г. Теория автоматов. – СПб.: Питер, 2002. – 224 с.
12. Клиначёв Н.В. «Моделирование систем в программе VisSim: Справочная система» [Электронный ресурс]. – Режим доступа: <http://model.exponenta.ru/k2/doc/knv.htm>.
13. Клиначёв Н.В. Введение в дисциплину «Основы моделирования систем» [Электронный ресурс]. – Режим доступа: [http://model.exponenta.ru/lectures/sml\\_01.htm](http://model.exponenta.ru/lectures/sml_01.htm).
14. Клиначёв Н.В. Введение в технологию моделирования на основе направленных графов [Электронный ресурс]. – Челябинск, 2003. – Режим доступа: [http://www.vissim.nm.ru/lectures/sml\\_02.htm](http://www.vissim.nm.ru/lectures/sml_02.htm).
15. Клиначёв Н.В. Введение в технологию мультидоменного физического моделирования с применением ненаправленных графов [Электронный ресурс]. – Челябинск, 2003. – Режим доступа: [http://www.vissim.nm.ru/lectures/sml\\_03.htm](http://www.vissim.nm.ru/lectures/sml_03.htm).
16. Клиначёв Н.В. Обзор архитектурного построения программ математического моделирования динамических систем [Электронный ресурс]. – Режим доступа: <http://www.exponenta.ru/educat/news/klinachev>.
17. Козлов О.С., Кондаков Д.Е., Скворцов Л.М. и др. Программный комплекс для исследования динамики и проектирования технических систем // Информационные технологии. – 2005. – № 9. – С. 12–19.

18. Козлов О.С., Кондаков Д.Е., Скворцов Л.М. и др. Программный комплекс «Моделирование в технических устройствах» [Электронный ресурс]. – Режим доступа: [model.exponenta.ru](http://model.exponenta.ru).
19. Колесов Ю.Б., Сениченков Ю.Б. Имитационное моделирование сложных динамических систем [Электронный ресурс]. – Режим доступа: [http://www.exponenta.ru/soft/others/mvs/ds\\_sim.asp](http://www.exponenta.ru/soft/others/mvs/ds_sim.asp).
20. Колесов Ю.Б., Сениченков Ю.Б. Компьютерное моделирование в научных исследованиях и образовании // Exponenta Pro. Математика в приложениях. – 2003. – № 1. – С. 4–11.
21. Колесов Ю.Б., Сениченков Ю.Б. Моделирование систем. Динамические и гибридные системы. /Учебное пособие. – СПб.: БХВ-Петербург, 2006. – 224 с.
22. Норенков И.П., Трудоношин В.А., Федорук В.Г. Математическое моделирование объектов мехатроники // Электронный журнал «Наука и образование». – Март 2005. – № 3. – Режим доступа: <http://www.techno.edu.ru:16001/db/msg/25272.html>.
23. Ортега Д., Пул Н. Введение в численные методы решения дифференциальных уравнений: пер. с англ. – М.: Наука, 1986. – 288 с.
24. Применение метода графов связей в технике / под ред. Д. Кэрнопа и Р. Розенберга. – М.: Мир, 1973. – 94 с.
25. Рогачев Г.Н. StateFlow 5. Руководство пользователя [Электронный ресурс]. – Режим доступа: <http://matlab.exponenta.ru/stateflow/book1>.
26. Сабинин Ю.А. Электромашинные устройства автоматики. – М.: Наука, 1988. – 408 с.
27. Скворцов Л.М. Явные адаптивные методы численного решения жестких систем // Математическое моделирование. – 2000. – № 12. – С. 97–107.
28. Советов Б.Я., Яковлев С.А. Моделирование систем. – М.: Высшая школа, 1985. – 269 с.
29. Тарасик В.П. Математическое моделирование технических систем: учебник для вузов. – Мн.: ДизайнПРО, 1997. – 640 с.
30. Трудоношин В.А., Пивоваров Н.В. САПР. Математические модели технических объектов. – М.: Высшая школа, 1986. – 160 с.
31. Шалагинов А.В. Моделирование [Электронный ресурс] // Курс лекций. – Режим доступа: [http://ermak.cs.nstu.ru/~shalag/tems/lect\\_6.html](http://ermak.cs.nstu.ru/~shalag/tems/lect_6.html).
32. Эйкхофф П. Основы идентификации систем управления. – М.: Мир, 1975. – 683 с.

Учебное издание

Воронин Александр Васильевич

## МОДЕЛИРОВАНИЕ МЕХАТРОННЫХ СИСТЕМ

Учебное пособие

Научный редактор  
профессор,  
доктор технических наук *А.М. Малышенко*

Редактор *О.А. Казакова*

Верстка *В.П. Аршиноаа*

Дизайн обложки *О.Ю. Аршинова*  
*О.А. Дмитриев*

Подписано к печати 20.08.2008. Формат 60x84/16. Бумага «Снегурочка».  
Печать XEROX. Усл. печ. л. 7,33. Уч.-изд. л. 6,63.  
Заказ 735. Тираж 500 экз.



Томский политехнический университет  
Система менеджмента качества  
Томского политехнического университета сертифицирована  
NATIONAL QUALITY ASSURANCE по стандарту ISO 9001:2000



ИЗДАТЕЛЬСТВО  ТПУ. 634050, г. Томск, пр. Ленина, 30.