Глава 14
МОДЕЛИРОВАНИЕ ТЕХНИЧЕСКИХ СИСТЕМ

При изучении технических устройств (механических, гидравлических, тепловых, электрических и магнитных систем) применяется **агрегатный подход**, позволяющий моделировать всевозможные процессы и системы: дискретные и непрерывные, детерминированные и стохастические [5, 10]. При этом используется **метод функционально законченных элементов**, предполагающий выделение типовых элементов технического объекта, завершенных в конструктивном отношении и предназначенных для выполнения определенных функций [6, 12]. Для расчета механических напряжений, тепловых, электрических и магнитных полей применяются методы сеток [11, 15].

14.1. Использование пакетов программ для моделирования технических систем

Компьютерное моделирование используется с целью анализа существующих и создания новых технических систем. Любой технический объект можно представить в виде совокупности взаимосвязанных компонентов, элементарных блоков или звеньев, взаимодействующих определенным образом, между которыми происходит обмен сигналами, веществом или энергией. Параметры блоков, характер взаимодействия, способы обмена информацией определяются из эксперимента. Создается компьютерная модель, и методом имитационного моделирования исследуются свойства и особенности функционирования изучаемого устройства.

Задачи анализа и синтеза технических объектов могут быть решены в различных математических пакетах, которые предоставляют широкие возможности в плане структурного и математического моделирования системы. Для расчета различных скалярных и векторных полей записывается система дифференциальных уравнений и применяется метод ко-

нечных разностей или конечных элементов. На рис. 14.1 приведены результаты использования метода конечных элементов для расчета токов утечки и распределения магнитного поля в трансформаторе [16], что помогает улучшить его конструкцию.

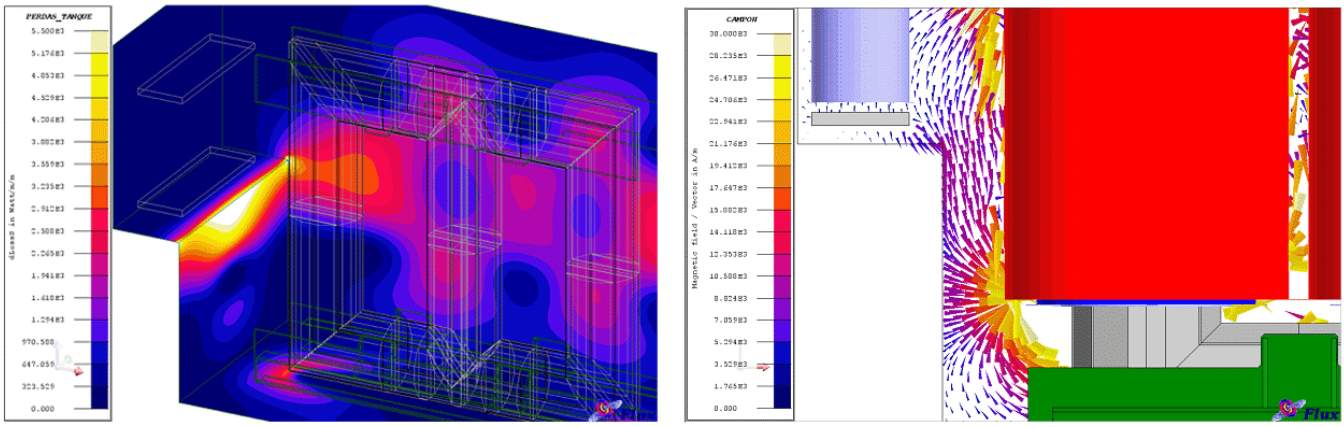

Рис. 14.1. Результаты расчета тока утечки и магнитного поля в трансформаторе (рисунок из [16]).

Все существующие пакеты программ для моделирования технических систем можно разделить на три класса [6, с. 91–123]:

1. **Математические пакеты** типа MathCAD, SciLab, Maple, позволяющие решать системы алгебраических и дифференциальных уравнений. Они оперируют числами в формате с плавающей запятой и используются для решения разнообразных задач с помощью численных методов.

2. **Пакеты компонентного моделирования** (MatLab, Workbench, Simulink), в которых создается визуальная модель системы из отдельных блоков. Пакеты компонентного моделирования подразделяются на универсальные (MatLab, ANSYS) и предметно-ориентированные (Electronics Workbench) программные пакеты. При этом различают структурное моделирование, физическое мультидоменное моделирование и моделирование гибридных систем.

3. **Специализированные пакеты**, к которым относятся программы на языках Fortran, Си и т.д., созданные на конкретных предприятиях с целью моделирования технических объектов, относящихся к конкретной прикладной области (механике, теплотехнике, электронике). Они постепенно заменяются предметно-ориентированными компонентными пакетами.

В качестве примера универсального пакета компонентного моделирования рассмотрим систему MatLab, используемую для научных и инженерных расчетов [8]. Этот матрично-ориентированный пакет программ, как и большинство других современных прикладных пакетов, имеет интерактивный интерфейс пользователя, состоящий из окон для задания входных параметров задачи, окон визуализации результатов вычислений, окна помощи, управляющих кнопок для запуска и остановки вычислений. Для работы с ним пользователю нет необходимости вникать в детали программирования, он может легко менять исходные данные и метод вычислений. В книге [8] рассмотрены различные модели схем электропривода в пакетах Power Sysem Blockset и MatLab Simulink.

Для компьютерного моделирования неоднородных технических систем часто используется пакет ANSYS. В нем применяется метод конечных элементов для решения таких задач, как расчет формы тел при деформации, расчет прочности конструкций, подвергающихся статическим и динамическим нагрузкам, моделирование течения жидкости и газа, распространения тепла, моделирование электротехнических систем, расчет электрического и магнитного полей и другие. Это универсальная программа по **компьютерному инжинирингу** позволяет анализировать устойчивость систем, изучать переходные процессы, решать оптимизационные задачи. Она оперирует с: 1) переменными проекта, которые изменяются с целью минимизации выходных параметров; 2) переменными состояния системы; 3) набором параметров проекта, определяющих конфигурацию модели; 4) целевой функцией, которая должна быть минимизирована. Используются следующие средства оптимизации: однократный анализ, случайное варьирование, сканирование области варьирования параметров, факторный и градиентный анализ.

К предметно-ориентированным программным средствам автоматизации моделирования, предназначенным для моделирования систем, принадлежащих определенному физическому классу, относятся MultiSim, Simulink, AutoCAD, SolidWork и другие. Они не позволяют создавать и исследовать компьютерные модели других физических классов, а также физически неоднородных объектов и систем. Так, для моделирования

конструкций при разнообразных воздействиях методом конечных элементов применяются программные продукты AutoCAD Mechanical Power Pack, Autodesk Mechanical Desktop Power Pack и другие [3]. Пакет SolidWorks создан для трехмерного моделирования и автоматизированного 3-D проектирования. Он используется в машиностроении, гидрогазодинамике, приборостроении, двигателестроении, конструировании технологической оснастки и т. д.

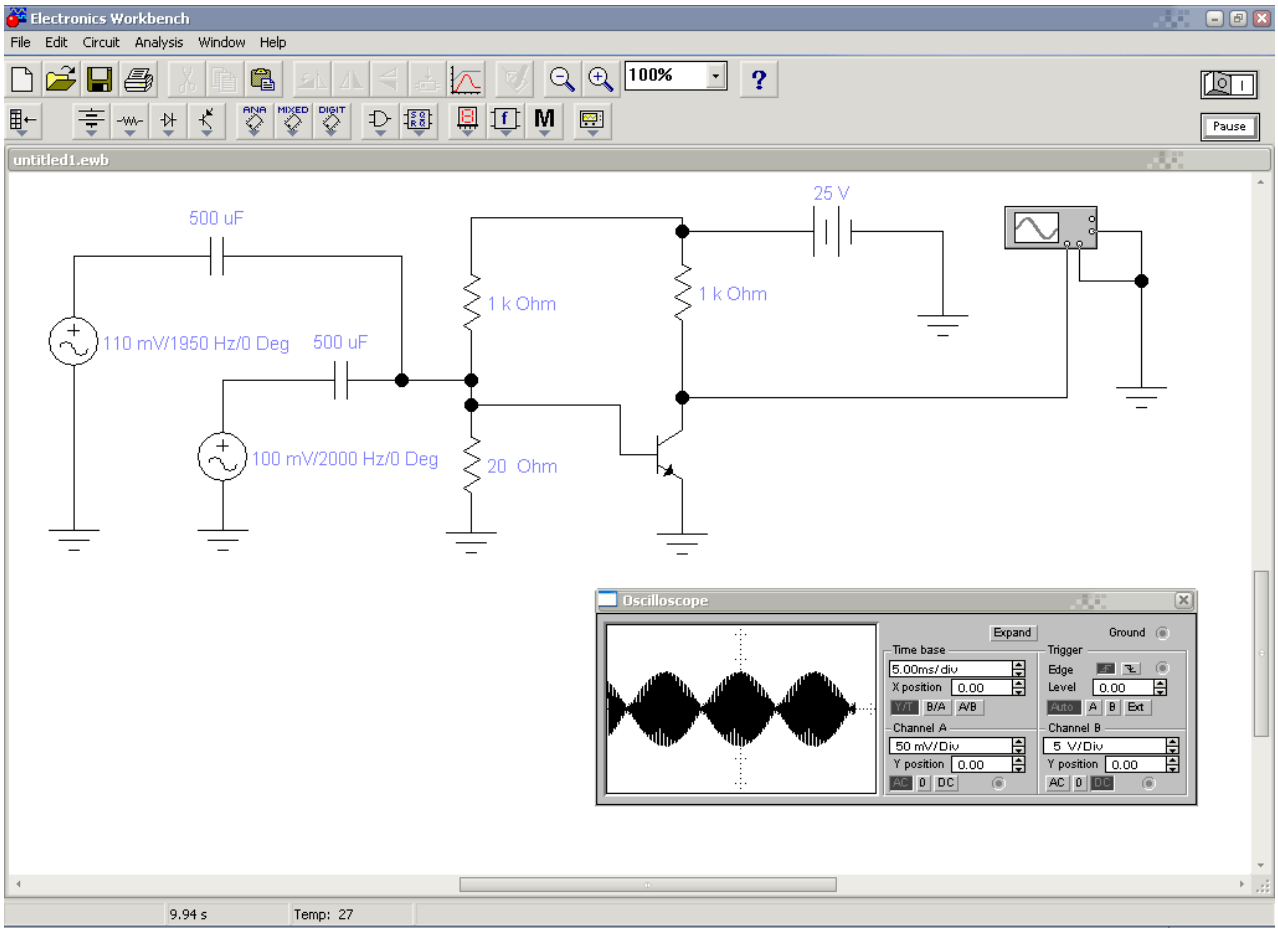

Рис 14.2. Моделирование электронной схемы в Workbench.

Для исследования электронных схем применяются прикладные пакеты OrCAD, Realise, DesignLab, MultiSim, Workbench, Circuit Marker. Рассмотрим пакет Workbench, который фактически является виртуальной лабораторией, позволяющей исследовать электрические цепи на транзисторах и микросхемах. На рис 14.1 представлено окно программы Electronics Workbench; в нем моделируется работа транзисторного усилителя, на вход которого подаются сигналы от двух генераторов. Пакет OrCAD также позволяет моделировать электронные схемы и имеет библиотеку, содержащую более 200 тыс. компонентов.

14.2. Передача информации по каналу связи

Рассмотрим передачу сообщения "101100...10" по двоичному каналу связи. Пусть с целью выявления ошибки кодер передающего устройства разбивает сообщение на блоки длиной $D-1$ и добавляет 1 бит четности так, что получаются кадры длиной D, причем сумма бит в кадре четна. Они поступают в канал связи, в котором с вероятностью p вносятся ошибки (инвертируются биты), и, пройдя через него, попадают в декодер приемника. Для простоты будем считать, что на передачу 1 бита затрачивается 1 с. Декодер находит сумму всех бит в кадре и проверяет их на четность. Реализуется **система с переспросом**: в случае ошибки (сумма бит нечетна) по каналу переспроса принимающее устройство посылает сигнал о повторе передачи последнего кадра. На его повторную передачу снова затрачивается D тактов. Допустим, что сигнал по каналу переспроса не вносит задержки. Определим скорость передачи информации методом статистических испытаний (глава 5).

Для решения задачи используется программа ПР-1 [9]. Длина кадра D равна величине константы Dlina_k, при этом число информационных бит составляет $D-1$, число передаваемых кадров Chislo_k$=10000$. В программе организован цикл, в котором моделируется побитовая передача первого кадра, второго кадра и т.д. При каждой итерации время t увеличивается на 1 и моделируется ошибка с вероятностью p. Генерируется случайное число x из интервала $[0;1]$, и если оно меньше p, то считается, что бит передан с ошибкой, в этом случае переменной $error$ присваивается 1. Источник продолжает передачу кадра до конца. Если кадр прошел по каналу связи без ошибок ($error=0$), то счетчик N правильно переданных кадров увеличивается на 1; в случае хотя бы одной ошибки ($error=1$) счетчик кадров N не изменяет своего значения. После этого моделируется передача следующего кадра и т.д. На экран выводится номер переданного кадра, число безошибочно переданных кадров и время t. Скорость передачи равна отношению количества правильно переданных

информационных бит к затраченному времени: $\upsilon = N(D-1)/t$, где $(D-1)$ -- число информационных бит в кадре.

В случае, когда длина кадра $D=8$, и сообщение передается без помех ($p=0$), скорость передачи полезной информации получается равной $\upsilon = 7/8 = 0{,}875$ бит/с (из 8 бит 7 информационных). Из результатов моделирования следует, что при увеличении вероятности ошибки p скорость передачи информации υ уменьшается: 1) при $p=0{,}001$, $\upsilon=0{,}867$ бит/с; 2) при $p=0{,}01$, $\upsilon=0{,}808$ бит/с; 3) при $p=0{,}02$, $\upsilon=0{,}747$ бит/с; 4) при $p=0{,}05$, $\upsilon=0{,}574$ бит/с. Это объясняется тем, что при увеличении вероятности ошибки растет количество ошибочно переданных кадров. Поэтому графиком зависимости $\upsilon=\upsilon(p)$ является убывающая кривая, которая с увеличением p приближается к 0 (рис. 14.3).

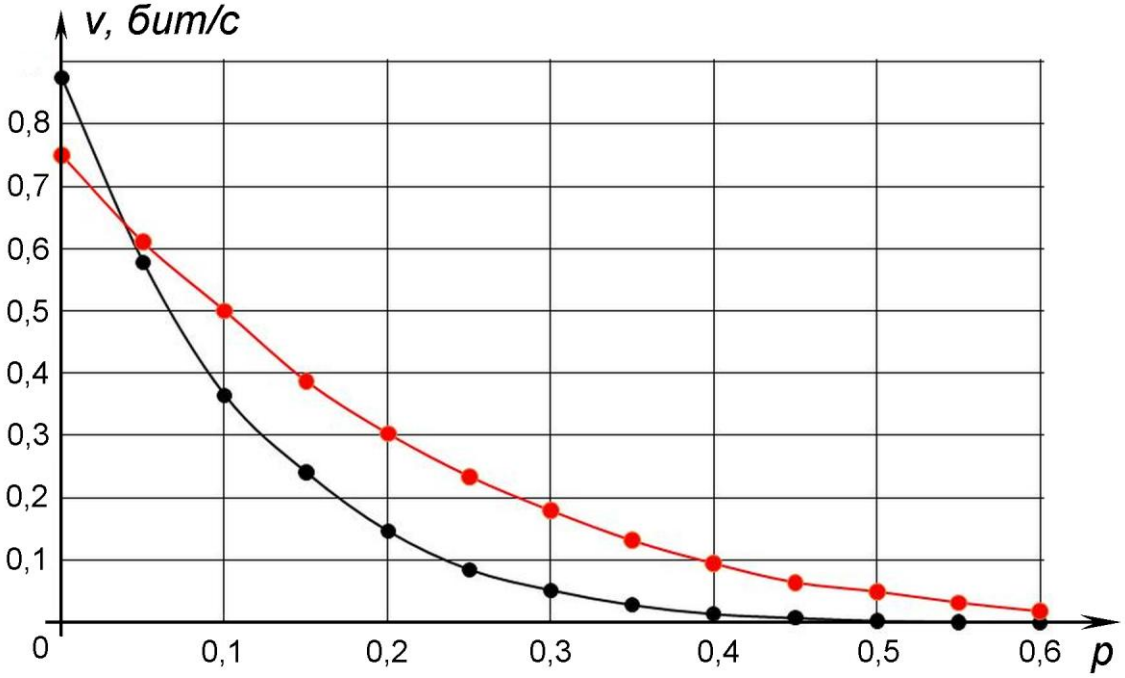

Рис. 14.3. Зависимость скорости передачи от вероятности ошибки: 1) длина кадра $D=4$ (красный); 2) длина кадра $D=8$ (черный).

Изучим зависимость скорости передачи информации от длины кадра D при различных p. Если $p=0{,}1$, то при длине кадра 2 или 3 бита скорость передачи невелика за счет большой доли проверочных битов четности. При больших D она уменьшается из-за увеличения вероятности ошибки в кадре и затрат времени на повторную его передачу. Результаты

моделирования для $p = 0{,}02$ и $0{,}05$ приведены на рис. 14.4. Используется программа ПР-1, графики построены в Excel.

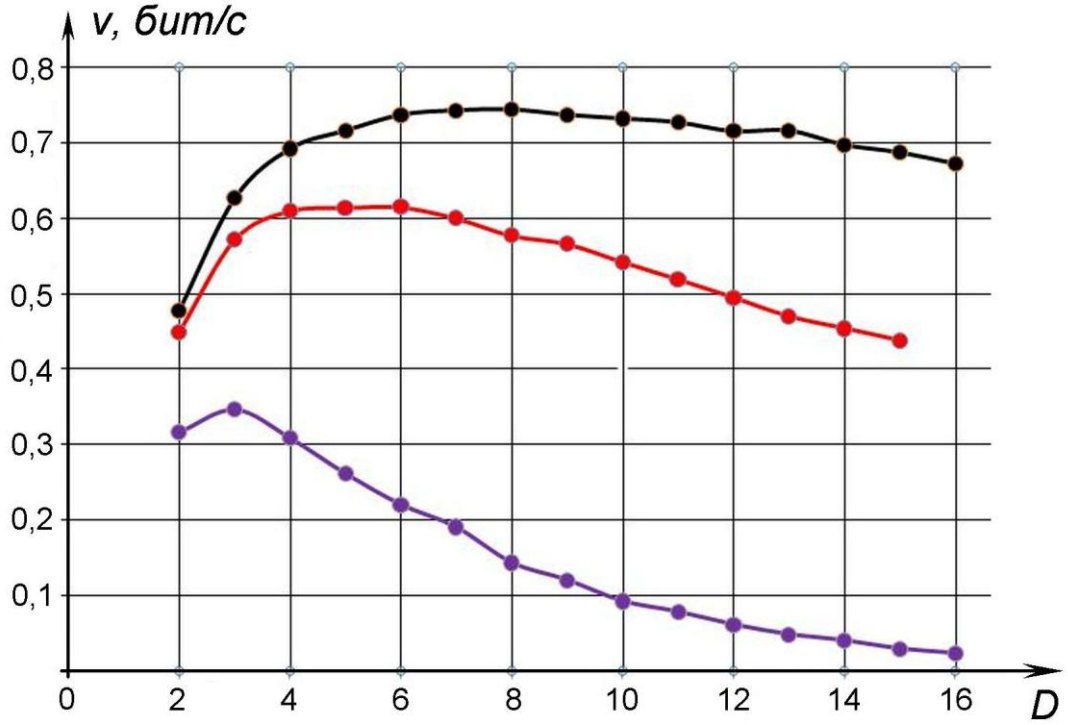

Рис. 14.4. Зависимость скорости передачи от длины кадра D при вероятностях ошибки: 1) $p = 0{,}02$ (черный); 2) $p = 0{,}05$ (красный); 3) $p = 0{,}2$ (фиолетовый).

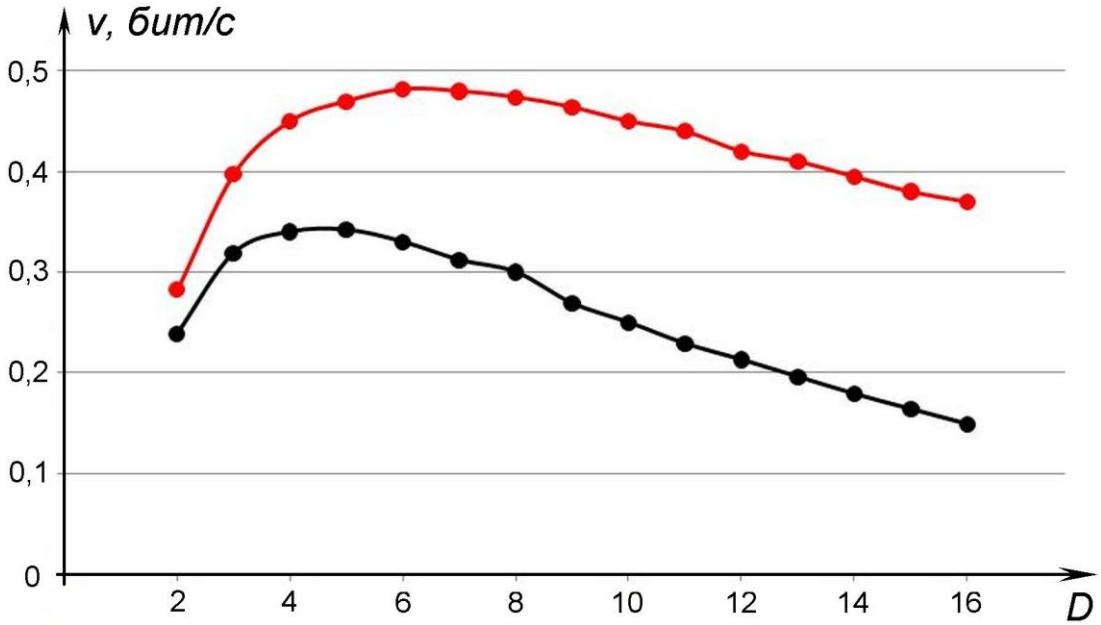

Рис. 14.5. Зависимость скорости передачи от длины кадра: 1) $p = 0{,}05$ (красный); 2) $p = 0{,}1$ (черный).

Используя программу ПР-1, можно изучить зависимость скорости передачи информации от вероятности ошибки p в случае, когда на про-

верку одного кадра затрачивается время τ_1, а на передачу сигнала по каналу переспроса -- время τ_2. Для этого достаточно присвоить переменным t_1 и t_2 значения 1 и 2. На рис. 14.5 представлены получающиеся графики при $\tau_1 = 1$ с и $\tau_1 = 2$ с.

14.3. Моделирование терморегулятора

Рассмотрим функционирование простейшей **системы замкнутого управления**, -- терморегулятора, поддерживающего температуру жидкости в заданном интервале. Система состоит из наполненного водой сосуда, нагревателя (спирали), термодатчика, подключенного к электронному устройству, выход которого соединен с реле, и блока питания. Когда температура датчика ниже T_1, срабатывает электронное устройство и реле замыкает цепь. Включается нагреватель, температура жидкости T повышается. Когда она оказывается выше T_2, электронное устройство отключает реле, цепь размыкается. За счет теплообмена с внешней средой T уменьшается, при $T < T_1$ нагреватель снова включается и т.д.

При протекании тока через нагреватель сопротивлением R за время Δt выделяется теплота $\Delta q_1 = (U^2 / R)\Delta t$. За счет теплообмена с внешней средой вода теряет энергию $\Delta q_2 = \alpha(T - T_S)\Delta t$, где T_S -- температура среды. Энергия воды увеличивается на $\Delta Q = (U^2 / R - \alpha(T - T_S))\Delta t$. Жидкость нагревается, ее температура повышается на $\Delta T = \Delta Q/(cm)$. Чтобы промоделировать функционирование термодатчика, электронного устройства и реле, зададим изменение напряжения на нагревателе так:

$$U = \begin{cases} 220, & если \quad T < T_1, \\ 0, & если \quad T > T_2. \end{cases}$$

Для моделирования работы терморегулятора используется программа Пр-2. В цикле по времени τ вычисляется количество теплоты, получаемое и теряемое водой за время $\Delta\tau$, и температура воды T. Результаты вычислений представлены на рис. 14.6. Видно, что моделируемый терморегулятор работает правильно и температура жидкости под-

держивается в заданном интервале от $T_1 = 70^0 C$ до $T_2 = 85^0 C$ при изменении температуры окружающей среды T_S. Если $T_S = 5^0 C$, то терморегулятор переключается чаще, так как после его выключения вода охлаждается быстрее. При $T_S = 50^0 C$ вода охлаждается медленнее, поэтому включения и выключения нагревателя происходят реже. Так эта замкнутая САУ адаптируется к изменениям температуры окружающей среды.

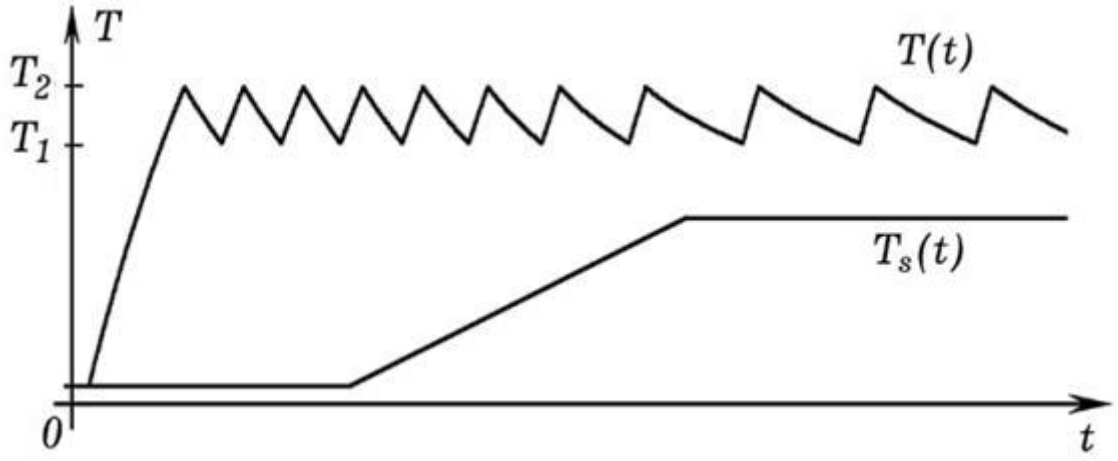

Рис. 14.6. Работа терморегулятора при различных температурах среды.

Проанализированная выше модель не учитывает размеров сосуда и неравномерное нагревание жидкости. Усложним задачу и промоделируем нагревание и конвекцию жидкости в двумерном прямоугольном сосуде. Методы расчета конвективного движения жидкости рассмотрены в главе 11. Используется программа ПР-3, результаты моделирования -- на рис. 14.7. В точке Н расположен нагреватель, а в точке Д -- датчик, соединенный с управляющим устройством. Периодическое включение и выключение нагревателя (график $i(t)$), происходящие в моменты, соответствующие точкам 1, 2, 3, ..., 12, приводят к колебаниям средней температуры T_{cp} воды в сосуде и температуры T_D датчика (рис 14.8). Из-за инерционности движущихся слоев жидкости значения T_{cp} и T_D выходят за пределы заданного интервала $[T_1;\ T_2]$. В программе следует так задать гранич-

ные условия, чтобы модель учитывала теплообмен с внешней средой. Это можно сделать с помощью условного оператора:

```
If ((i=1)or(i=N)or(j=1)or(j=M))and(T[i,j]>Tsr)then
                              q:=-0.003*(T[i,j]- Tsr) else q:=0;.
```

Когда температура стенок сосуда превышает температуру среды, на стенках появляются источники холода $q' = -\alpha'(T - T_S)$, они охлаждаются.

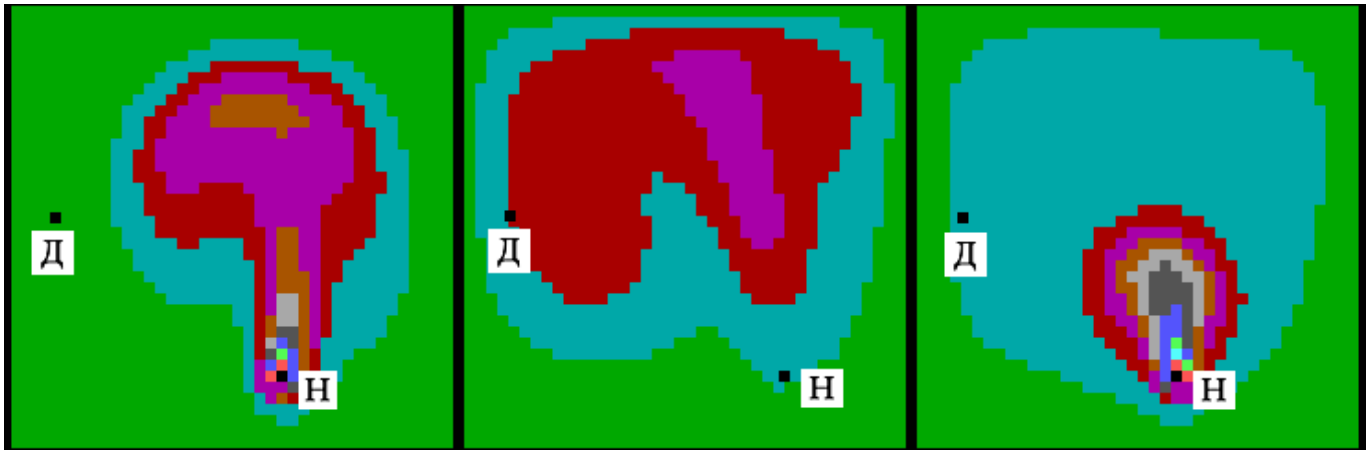

Рис 14.7. Периодическое включение и выключение нагревателя.

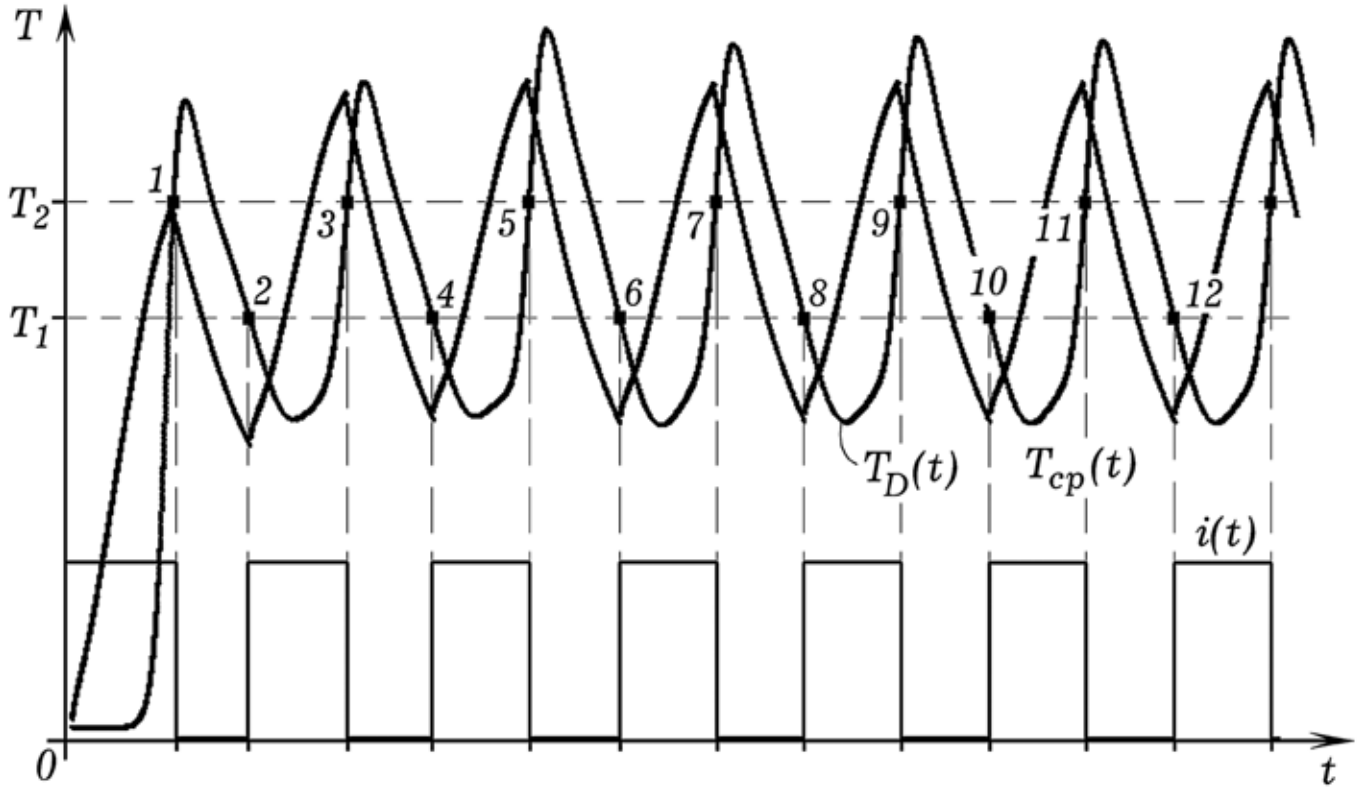

Рис. 14.8. Колебания температуры датчика и средней температуры жидкости.

14.4. Моделирование работы электроизмерительного прибора

Рассмотрим измерительный прибор электромагнитной системы, состоящий из двух постоянных магнитов, цилиндра из магнитомягкого материала, подвижной обмотки, к которой прикреплена стрелка и один конец пружины. Второй конец пружины прикреплен к корпусу прибора. При подаче на концы обмотки напряжения по ней течет ток, и со стороны магнитного поля неподвижных магнитов действует вращающий момент, поворачивающий обмотку со стрелкой и деформирующий пружину. Чтобы колебания стрелки быстро затухали, используется воздушный демпфер, состоящий из изогнутого сосуда и поршня. Методом компьютерного моделирования изучим движение стрелки прибора при подаче постоянного напряжения, прямоугольных импульсов и т.д. [9].

Для подвижной обмотки со стрелкой запишем основной закон динамики вращательного движения: $I\varepsilon = M_M - M_Y - M_{TP} = k_1 i - k_2\varphi - k_3\omega$, где I -- момент инерции подвижной обмотки со стрелкой, $\varepsilon = \dot{\omega} = \ddot{\varphi}$ -- ее угловое ускорение, $M_M = k_1 i$ -- вращающий момент магнитных сил, $M_Y = k_2\varphi$ -- момент силы упругости, $M_{TP} = k_3\omega$ -- момент сил вязкого трения. Если учесть ЭДС индукции $-k_4 d\varphi/dt$ и самоиндукции $-Ldi/dt$, возникающие во вращающейся в магнитном поле обмотке, по которой течет изменяющийся ток, то из второго закона Кирхгофа следует:

$$Ri + L\frac{di}{dt} + k_4\frac{d\varphi}{dt} = u(t).$$

Угол поворота стрелки изменяется от 0 до $\pi/2$; на концах шкалы установлены ограничители движения. При частично упругом ударе стрелки об ограничитель она теряет часть своей скорости и отскакивает в противоположную сторону. Учтем отскакивание стрелки от ограничителей так: если $\varphi^t < 0$ или $\varphi^t > \pi/2$, то $\omega^{t+1} = -b\omega^t$, $b = 0{,}8$ -- коэффициент восстановления при ударе. В конечных разностях получаем:

$$\varepsilon^{t+1} = (k_1 i^t - k_2\varphi^t - k_3\omega^t)/I,\ \ \omega^{t+1} = \omega^t + \varepsilon^{t+1}\Delta t,\ \ \varphi^{t+1} = \varphi^t + \omega^{t+1}\Delta t,$$

$$i^{t+1} = i^t + \frac{u^t - Ri^t - k_4(\varphi^t - \varphi^{t-1})/\Delta t}{L}\Delta t\,.$$

Используется программа ПР-4. Она содержит цикл по времени, в котором пересчитываются сила тока в обмотке, действующие моменты сил, угловые ускорение, скорость и координата подвижной обмотки со стрелкой. Результаты выводятся на экран в виде графиков (рис. 14.9).

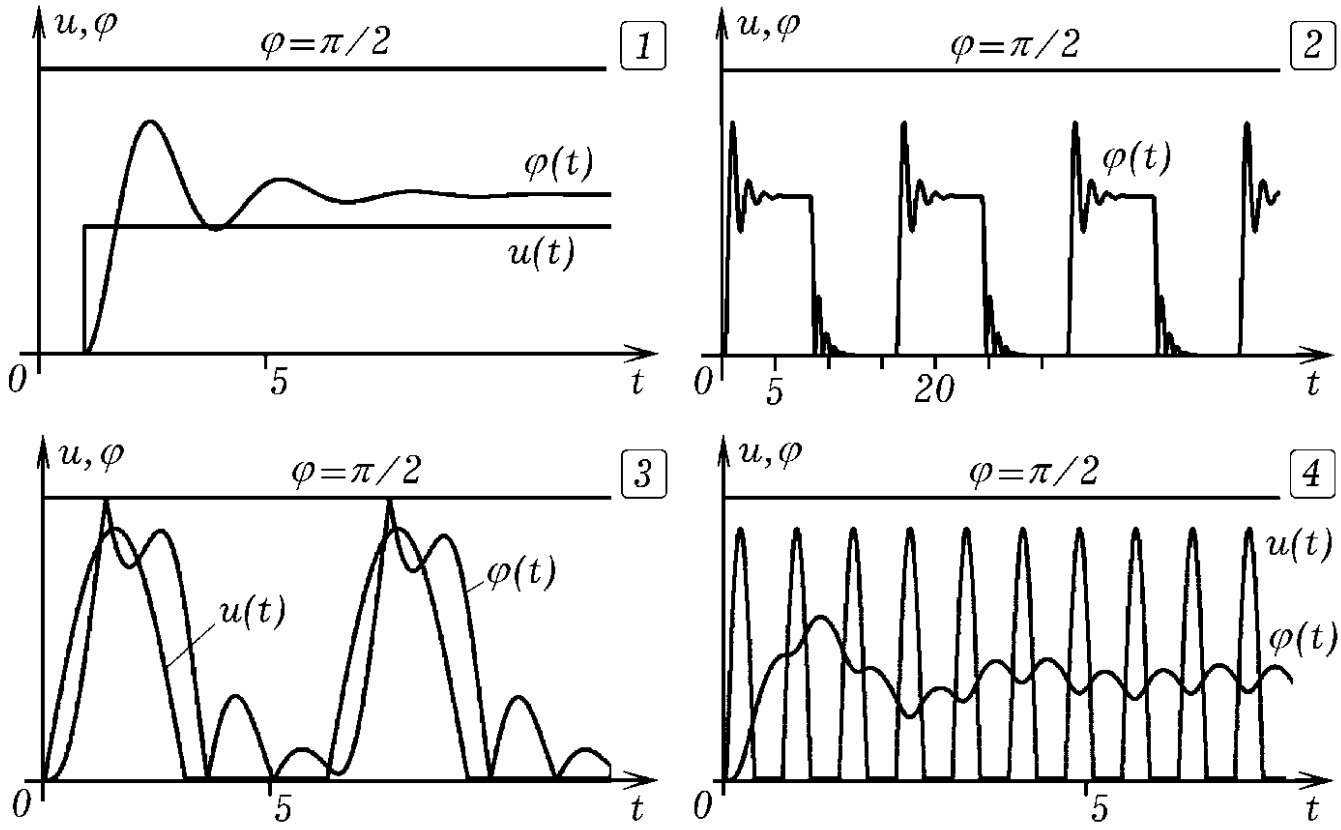

Рис. 14.9. Колебания стрелки прибора при подаче импульсов напряжения.

На рис. 14.9.1 представлен график затухающих колебаний стрелки при подаче на подвижную обмотку постоянного напряжения. Если на вход прибора подать прямоугольные импульсы напряжения низкой частоты, то стрелка будет совершать вынужденные колебания, изображенные на рис. 14.9.2. При подаче на вход импульсов напряжения, получающихся в результате однополупериодного выпрямления, стрелка прибора совершает колебания, изображенные на рис. 14.9.3 и 14.9.4. Можно убедиться, что амплитуда вынужденных колебаний сильно зависит от соотношения собственной частоты подвижной обмотки со стрелкой и частоты колебаний $u(t)$. Компьютерная модель позволяет исследовать поведение измерительного прибора при различных его параметрах и входных сигналах.

14.5. Моделирование работы асинхронного двигателя

Для преобразования электрической энергии в механическую часто используют асинхронный двигатель (АД). Трехфазный АД (рис. 14.10) состоит из статора, содержащего три обмотки, повернутые на 120^0, и короткозамкнутого ротора. При подаче на обмотки трехфазного напряжения возникает вращающееся магнитное поле, которое создает в роторе индукционные токи и, взаимодействуя с ними, вызывает его вращение.

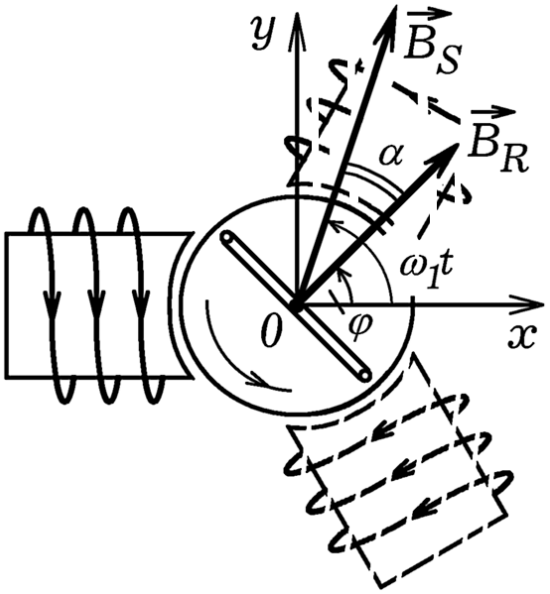

Рис. 14.10. Устройство трехфазного асинхронного двигателя.

Рассмотрим упрощенную модель, в которой не учитывается влияние магнитного поля ротора на токи в обмотках статора [9]. Внутри статора возникает магнитное поле с индукцией B_S, вращающееся со скоростью ω_1. Так как угол поворота ротора равен φ, то угол между вектором магнитного поля статора B_S и нормалью к виткам обмотки ротора составляет $\alpha = \omega_1 t - \varphi$. Тогда магнитный поток через виток ротора площадью S_R равен $\Phi_R = B_S S_R \cos\alpha$. В обмотке ротора возникает ЭДС индукции $e_{iR} = -N_R d\Phi_R / dt$. Для обмотки ротора можно записать:

$$e_{iR}(t) = L_R \frac{di_R}{dt} + Ri_R,$$

где L_R -- индуктивность обмотки ротора. Со стороны магнитного поля статора на ротор действует вращающий момент $M = N_R B_S i_R S_R \sin\alpha$. Угловое ускорение ротора равно $\varepsilon = (M - M_{TP} - M_H)/I_R$, где $M_{TP} = k\omega$ -- тормозящий момент, действующий на ротор со стороны подшипников, M_H -- механическая нагрузка на валу, I_R -- момент инерции ротора. Угловые скорость и ускорение ротора равны: $\omega = d\varphi/dt$, $\varepsilon = d\omega/dt$. В конечных разностях получаем: $\Phi_R^t = B_S S_R \cos(\omega_1 t - \varphi^t)$,

$$e_{iR}^t = -N_R \frac{\Phi_R^t - \Phi_R^{t-1}}{\Delta t}, \qquad i_R^{t+1} = \frac{e_{iR}^t \Delta t + L_R i_R^t}{R\Delta t + L_R},$$

$$\varepsilon^{t+1} = (M^{t+1} - M_{TP}^{t+1} - M_H^{t+1})/I_R,\ \omega_2^{t+1} = \omega_2^t + \varepsilon^{t+1}\Delta t,\ \varphi^{t+1} = \varphi^t + \omega_2^{t+1}\Delta t.$$

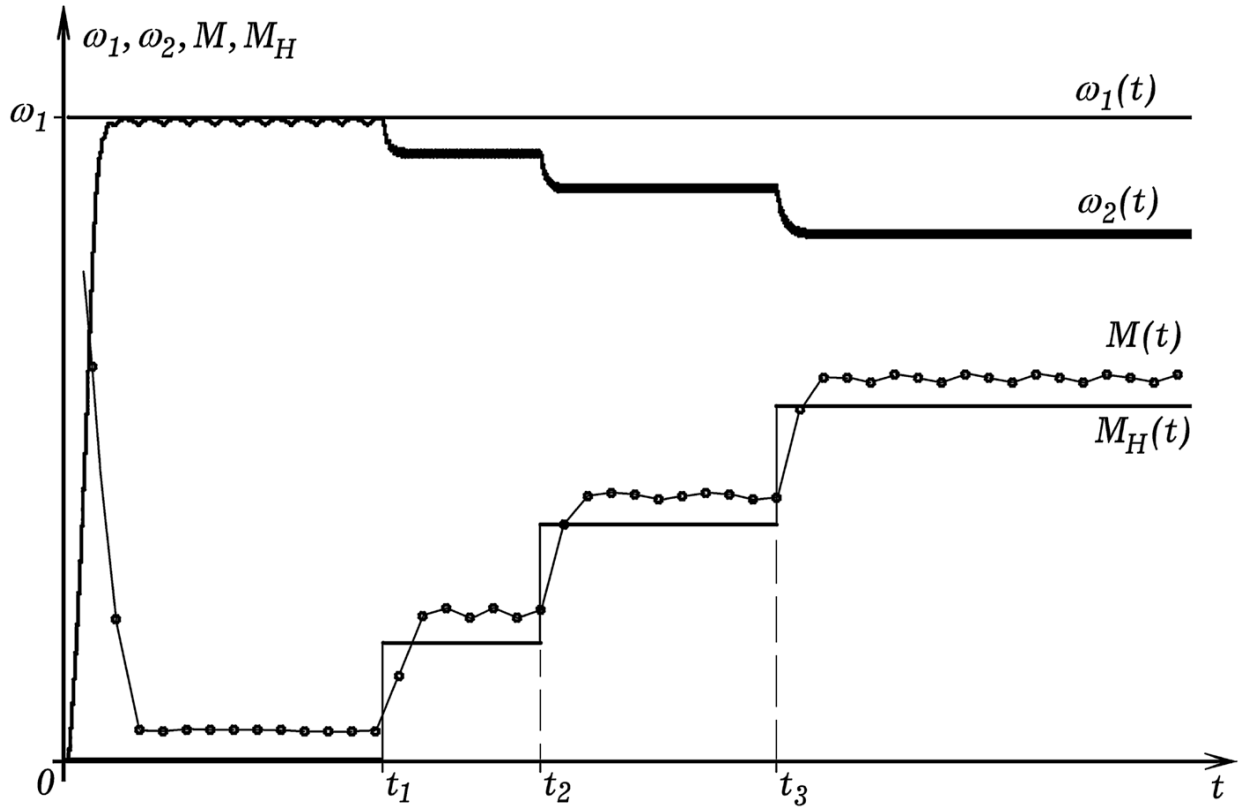

Рис. 14.11. Работа двигателя при изменении нагрузки на валу.

Программа ПР-5, моделирующая работу асинхронного двигателя, содержит цикл по времени, в котором пересчитываются ток в обмотке ротора, мгновенное и среднее значения вращающего момента, угловые ускорение, скорость и перемещение ротора в последовательные моменты времени. Магнитное поле статора движется относительно ротора АД со

скоростью $\omega_1 - \omega_2$, поэтому мгновенный вращающий момент периодически изменяется с этой частотой.

На рис. 14.11 показано, как изменяется скорость вращения ротора асинхронного двигателя и действующий на него средний вращающий момент M_{cp} при ступенчатом изменении механической нагрузки на валу. При увеличении нагрузки скорость ротора ω_2 уменьшается, средний вращающий момент M_{cp} растет. Программа позволяет изучить влияние параметров двигателя (I_R, N_R, S_R, L_R и т.д.) на его работу.

14.6. Моделирование системы “двигатель–генератор”

Теперь рассмотрим асинхронный двигатель, вал которого соединен с валом синхронного генератора [9]. Генератор состоит из статора с неподвижной обмоткой и ротора, вращающегося под действием механического момента, создаваемого двигателем. Через обмотку вращающегося ротора генератора течет постоянный ток, возникает вращающееся магнитное поле с индукцией B. Магнитный поток через обмотку статора равен $\Phi = BS\cos\varphi$, в ней возникает ЭДС индукции $e_G = -N_S d\Phi / dt$, где N_S -- число витков. Если к обмотке статора подключить нагрузку сопротивлением R_H, то потечет ток i_G; при этом $e_G = i_G R_H + L di_G / dt$. Этот ток создает магнитное поле статора с индукцией $B_S = k_1 N_S\, i_G$, которое тормозит ротор генератора (и двигателя). Тормозящий момент равен

$$M_H = k_2 B_S S N_R \sin\varphi = k N_S i_G S N_R \sin\varphi.$$

В конечных разностях получаем следующее. Для двигателя:

$$M^{t+1} = N_R B_S i_R^t S_R \sin(\omega_1 t - \varphi^t),\ M_H^t = k N_S i_G^t S N_R \sin\varphi^t,\ M_H = k i_G \sin\varphi^t,$$

$$\varepsilon^{t+1} = (M^{t+1} - M_{TP}^{t+1} - M_H^{t+1}) / I_R,\ \omega_2^{t+1} = \omega_2^t + \varepsilon^{t+1}\Delta t,\ \varphi^{t+1} = \varphi^t + \omega_2^{t+1}\Delta t.$$

Для генератора:

$$e_G^{t+1} = -N_S \frac{\Phi^{t+1} - \Phi^t}{\Delta t},\quad e_G^{t+1} = i_G^{t+1} R_H + L\frac{i_G^{t+1} - i_G^t}{\Delta t},\quad i_G^{t+1} = \frac{e_G^{t+1}\Delta t - L i_G^t}{R_H \Delta t + L}.$$

При запуске программы ПР-6, моделирующей работу системы "двигатель-генератор", получаются графики, изображенные на рис. 14.12. В моменты t_1 и t_2 нагрузка генератора ступенчато повышается (R_H уменьшается), что вызывает увеличение механической нагрузки на валу двигателя. Среднее значение вращающего момента АД M_{cp} растет, скорость ротора ω_2 уменьшается. Сила тока генератора $i_G(t)$ совершает колебания с частотой вращения его ротора; в моменты t_1 и t_2 ее амплитуда резко увеличивается, так как после резкого уменьшения R_H ротор по инерции продолжает двигаться с большой скоростью. Через некоторое время скорость ротора ω снижается, амплитуда тока $i_G(t)$ уменьшается.

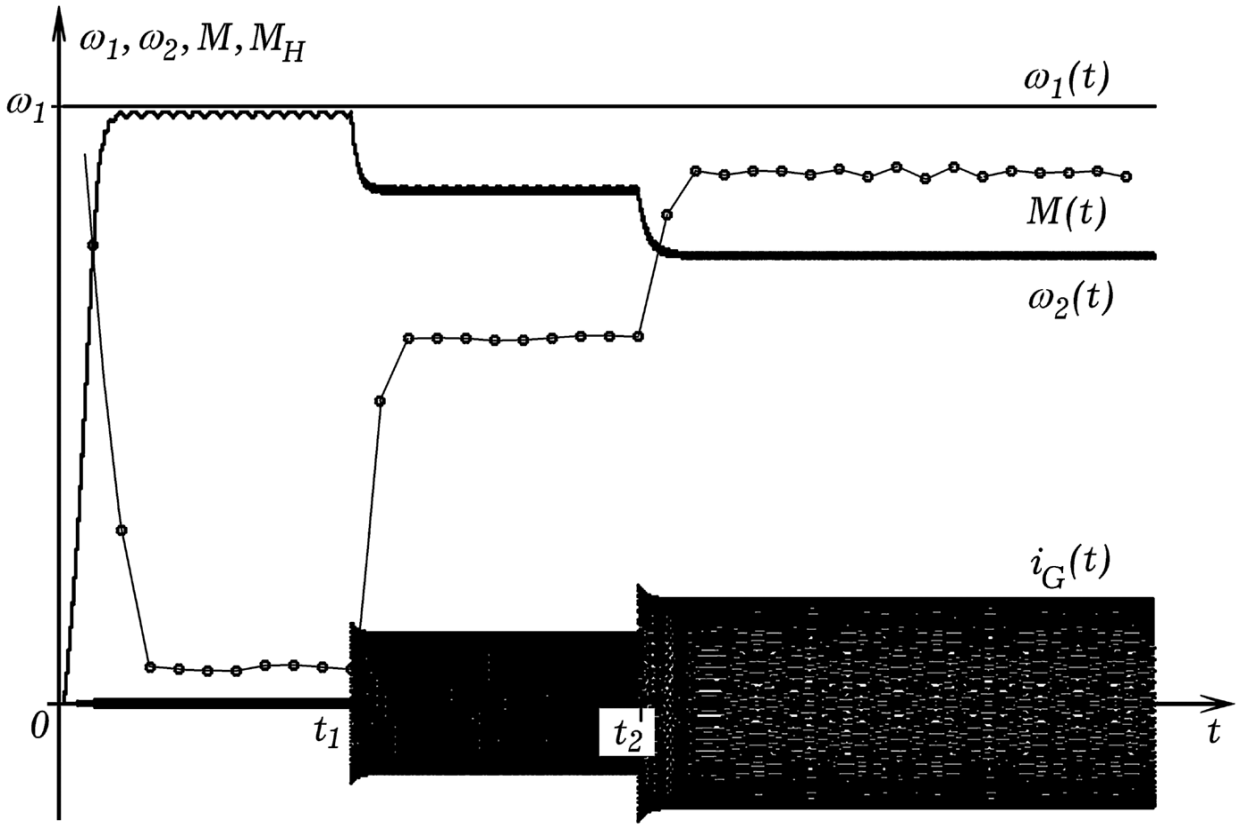

Рис. 14.12. Моделирование системы "двигатель-генератор".

14.7. Модель системы автоматического регулирования

В качестве непрерывной замкнутой системы автоматического управления (САУ) рассмотрим систему автоматического регулирования скорости (рис. 14.13), состоящую из двигателя постоянного тока 2, вал которого

соединен с управляемым объектом 4 и тахометром 3, подключенным к электронному устройству управления 1. Ток возбуждения, текущий через обмотку статора, остается неизменным; он создает магнитный поток $\Phi = const$. Малоинерционный тахометр 3 выдает напряжение, пропорциональное скорости вращения ротора ω, которое поступает на вход устройства управления 1. Оно работает так: при отклонении скорости вращения вала ω от заданной величины A происходит изменение напряжения питания на якоре двигателя $u(t)$ на величину Δu в соответствии с определенным **законом регулирования**. В результате скорость вращения снова возвращается к заданному значению. Изучим работу этой **самоадаптирующейся системы** при изменении механической нагрузки на валу [9].

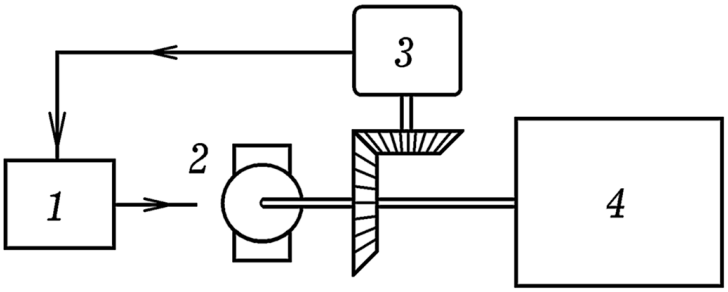

Рис. 14.13. Система автоматического регулирования скорости вращения.

По второму закону Кирхгофа для цепи якоря двигателя:

$$L\frac{di}{dt} + Ri + E_{Я} = u(t),$$

где $E_{Я} = k_1 \omega \Phi = K_1 \omega$ -- противо-ЭДС, возникающая в обмотке якоря вследствие его вращения, Φ -- поток возбуждения, L и R -- индуктивность и сопротивление обмотки якоря, $u(t)$ -- напряжение питания. По законам динамики $I\varepsilon = M - M_{TP} - M_H$, где I -- момент инерции ротора двигателя, $M = k_2 i \Phi = K_2 i$ -- вращающий момент, действующий на ротор со стороны магнитного поля статора, $M_{TP} = K_3 \omega$ -- тормозящий момент, действующий на ротор со стороны подшипников, $M_H(\omega)$ -- механическая нагрузка на валу, $\varepsilon = d\omega / dt$ и $\omega = d\varphi / dt$ -- угловые ускорение и скорость вала. Получаем систему уравнений:

$$\frac{di}{dt}=\frac{u(t)-Ri-K_1\omega}{L},\qquad \frac{d\omega}{dt}=\frac{K_2 i-K_3\omega-M_H}{I}.$$

Учесть влияние **цепи обратной связи**, состоящей из тахометра и устройства управления, можно, задав закон регулирования. Он выражается зависимостью вида $du/dt=\alpha(A-\omega)$, где A -- заданное значение угловой скорости. Если $\omega<A$, то $du/dt>0$, управляющее напряжение u и скорость ротора ω растут, а если $\omega>A$, то $du/dt<0$, u и ω уменьшаются.

Заменим производные их конечно-разностными аппроксимациями:

$$i^{t+1}=i^t+\frac{u^t-Ri^t-K_1\omega^t}{L}\Delta t,\qquad \omega^{t+1}=\omega^t+\frac{K_2 i^t-K_3\omega^t-M_H^t}{I}\Delta t.$$

Закон регулирования $du/dt=\alpha(A-\omega)$ дает $u^{t+1}=u^t+\alpha(A-\omega)\Delta t$.

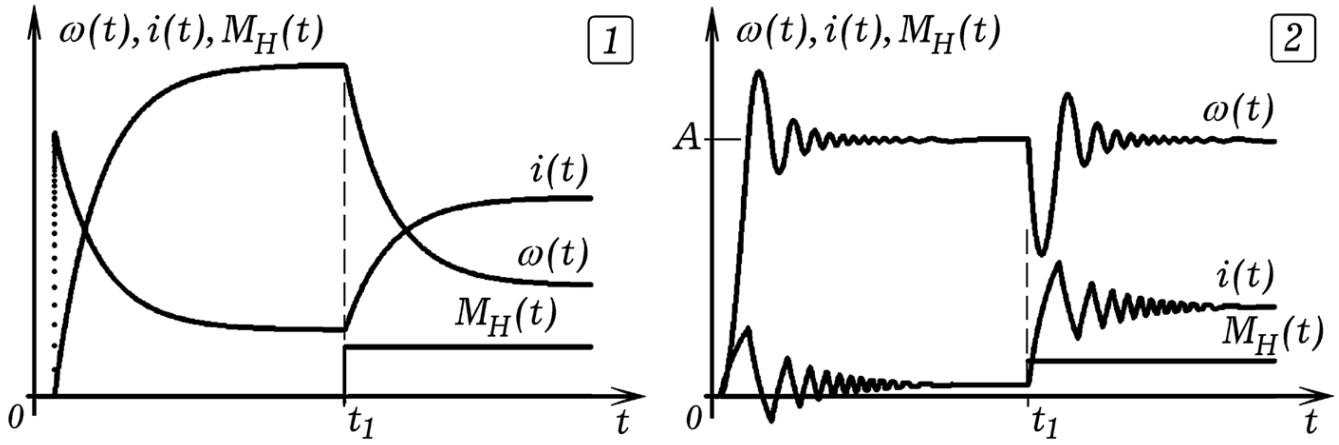

Рис. 14.14. Результаты компьютерного моделирования работы двигателя.

Сначала промоделируем работу машины постоянного тока в режиме двигателя и изучим, как изменяется угловая скорость ротора и ток якоря при включении и резком увеличении нагрузки. Используемая программа ПР-7, включает в себя цикл по времени t, в котором вычисляются перечисленные выше величины, и строятся графики $\omega(t)$ и $i(t)$. Результаты представлены на рис. 14.14.1. Видно, что при включении ток якоря резко возрастает, а затем плавно убывает, скорость $\omega(t)$ начинает расти. Когда наступает равенство $M=M_T+M_H$, система входит в установившейся режим, скорость ротора ω и ток якоря i достигают своих предельных значений. В момент t_1 резко возрастает момент нагрузки M_H, скорость

вращения ротора ω уменьшается, потребляемый ток i растет; ω и i стремятся к новым предельным значениям.

Теперь рассмотрим замкнутую САУ, состоящую из двигателя (МПТ), тахометра, цепи управления. Пусть закон регулирования напряжения питания двигателя задан так:

$$u^{t+1} = \begin{cases} u^t - b_1, & \omega > A + \Delta A, \\ u^t + b_2, & \omega < A - \Delta A. \end{cases}$$

Для моделирования этой системы используется программа ПР-8; результаты представлены на рис. 14.14.2. Так как при включении системы $\omega = 0 < A - \Delta A$, то сначала напряжение питания u растет, ток якоря i и его скорость вращения ω увеличиваются. Когда $\omega > A + \Delta A$, управляющее напряжение начинает уменьшаться, уходя в область отрицательных значений. Скорость ротора падает, снова проскакивая заданное значение A. После двух-трех затухающих колебаний скорость системы стабилизируется в интервале $A \pm \Delta A$. В момент t_1 скачком увеличивается момент нагрузки на валу, угловая скорость ω резко уменьшается. Напряжение питания u и ток якоря i растут, затем уменьшаются, совершая затухающие колебания относительно новых "положений равновесия". Угловая скорость ротора ω после нескольких колебаний опять стабилизируется в интервале $A \pm \Delta A$.

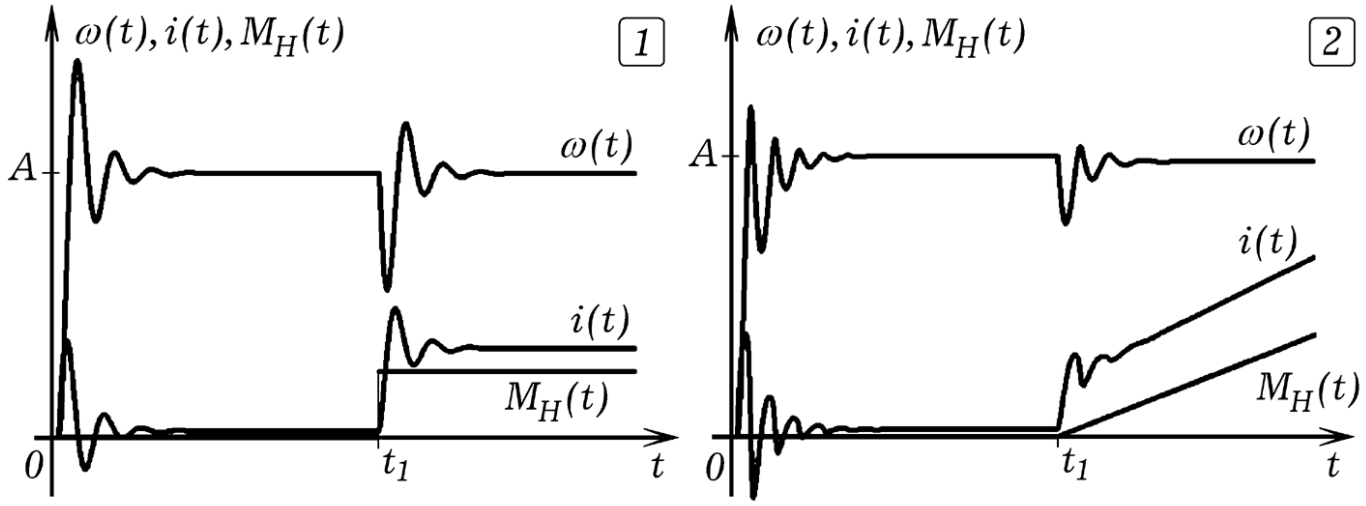

Рис. 14.15. Работа системы автоматического регулирования скорости.

Теперь исследуем функционирование анализируемой САУ в случае, когда закон регулирования имеет вид: $du/dt = b(A-\omega)$, то есть напряжения питания двигателя изменяется пропорционально отклонению ω от заданного значения A. Для этого в программе ПР-8 необходимо раскомментировать оператор `dU:=0.002*(a-w);`. Получающиеся графики представлены на рис. 14.15.1. Видно, что при резком изменении момента нагрузки скорость ротора, совершив несколько колебаний, возвращается к заданному значению A.

Для того, чтобы изучить поведение системы при равномерном увеличении механической нагрузки на валу M_H, следует активизировать оператор: `If t>70 then Mn:=1.1*(t-70);`. Если при этом мы хотим учесть, что коэффициент b в законе регулирования для случаев $\omega < A$ и $\omega > A$ имеет различные значения, то необходимо раскомментировать операторы: `If w<a then dU:=0.002*(a-w); If w>a then dU:=0.02*(a-w);`. Когда коэффициенты неодинаковы, кривые колебаний $i(t)$ и $\omega(t)$ относительно установившихся значений несимметричные (рис. 14.15.2).

14.8. Моделирование движения мотоциклиста

Рассмотрим модель движения мотоцикла с мотоциклистом по неровной дороге. Подобная математическая модель движения автомобиля проанализирована в [12, с. 71-73]. Будем считать, что мотоциклист и мотоцикл образуют единое тело, -- мотоцикл, упрощенная модель которого состоит из 6 материальных точек, связанных между собой вязко-упругими связями. Массы частиц 5 и 6 равны массам колес, а массы частиц 1, 2, 3 и 4 подбираются так, чтобы их общая масса была бы равна массе мотоцикла, а центр масс C располагался где-то вблизи точки O (рис. 14.16.1). Частицы соединены между собой связями, состоящими из упругого элемента жесткостью k и диссипативного элемента с коэффициентом вязкости r.

В начале программы задаются координаты частиц и рассчитываются расстояния s_{ij} между ними, когда мотоцикл находится в недеформированном состоянии. Она содержит цикл по времени, в котором пересчитываются ускорения, скорости и координаты всех 6 частиц системы. При этом $\vec{a} = (\vec{F} + \vec{N} + m\vec{g} + \vec{F}_m + \vec{F}_{mp})/m$, где $\vec{F}$ -- вязкоупругая сила между частицами системы, $\vec{N}$ -- сила реакции дороги, $m\vec{g}$ -- сила тяжести, $\vec{F}_m$ -- сила тяги, $\vec{F}_{mp}$ -- сила трения. Сила тяги приложена к заднему колесу мотоцикла и направлена параллельно поверхности дороги.

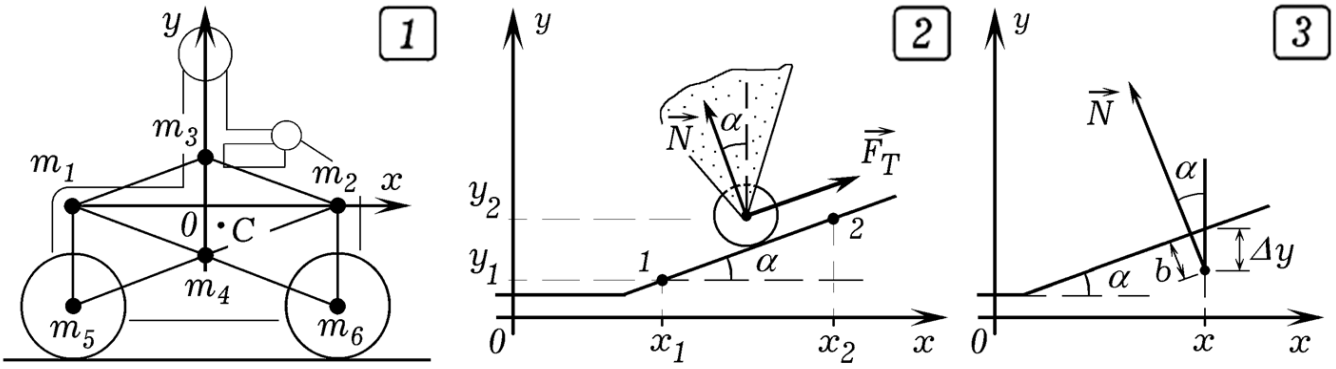

Рис. 14.16. К задаче о моделировании движения мотоцикла.

Проекции силы взаимодействия между частицами равны:

$$F_{yn,j} = k(s_{ij} - l_{ij}), \quad l_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2},$$

$$F_{ix} = \sum_{j=1}^{6}\left(F_{yn,j} - \frac{r(l_{ij} - d_{ij})}{\Delta t}\right)\cdot\frac{x_i - x_j}{l_{ij}},$$

$$F_{iy} = \sum_{j=1}^{6}\left(F_{yn,j} - \frac{r(l_{ij} - d_{ij})}{\Delta t}\right)\cdot\frac{y_i - y_j}{l_{ij}} - m_i g,$$

где d_{ij}-- расстояние между i-ой и j-ой частицами на предыдущем шаге (рис. 14.16.2). Следует учесть, что сила реакции возникает во время касания колесами дороги, и ее модуль N зависит от $b = \Delta y\cos\alpha$, где $\Delta y = y_i - y(x)$ ($i = 5$ или 6), $y(x)$ -- функция, определяющая профиль дороги. Проекции $\vec{N}$: $N_x = -N\sin\alpha$, $N_y = N\cos\alpha$ (рис. 14.16.3). Сила тяги приложена к точке m_5 и действует во время касания заднего колеса и поверхности; ее проекции $F_x = F\cos\alpha$, $F_y = F\sin\alpha$. Модуль силы тяги F зависит от $b = \Delta y\cos\alpha$.

Компьютерная программа ПР-9 позволяет промоделировать движение мотоцикла по неровной поверхности в случае, когда сила тяги изменяется заданным образом. Если начальная скорость мотоцикла мала, а мотоциклист дает газ, то при определенном соотношении силы тяги, времени ее действия и распределения массы мотоцикл может встать "на дыбы" (рис. 14.17.1), а затем вернуться в нормальное положение. Если увеличить силу тяги или время ее действия, то мотоцикл перевернется назад. При небольшой силе тяги мотоцикл ускоряется вперед, его корпус совершает крутильные колебания вокруг оси проходящей через центр масс C перпендикулярно рисунку. На рис. 14.17.2 показан результат моделирования при разгоне мотоцикла и прыжке с небольшого трамплина; на рис. 14.17.3 -- при движении по волнистой поверхности. Компьютерная модель позволяет изучить вопрос об оптимальном расположении центра масс мотоцикла, при котором он после отрыва от трамплина движется практически поступательно, не заваливаясь вперед или назад. Программа после изменений позволяет получить график крутильных колебаний корпуса мотоцикла вокруг горизонтальной оси, проходящей через центр масс.

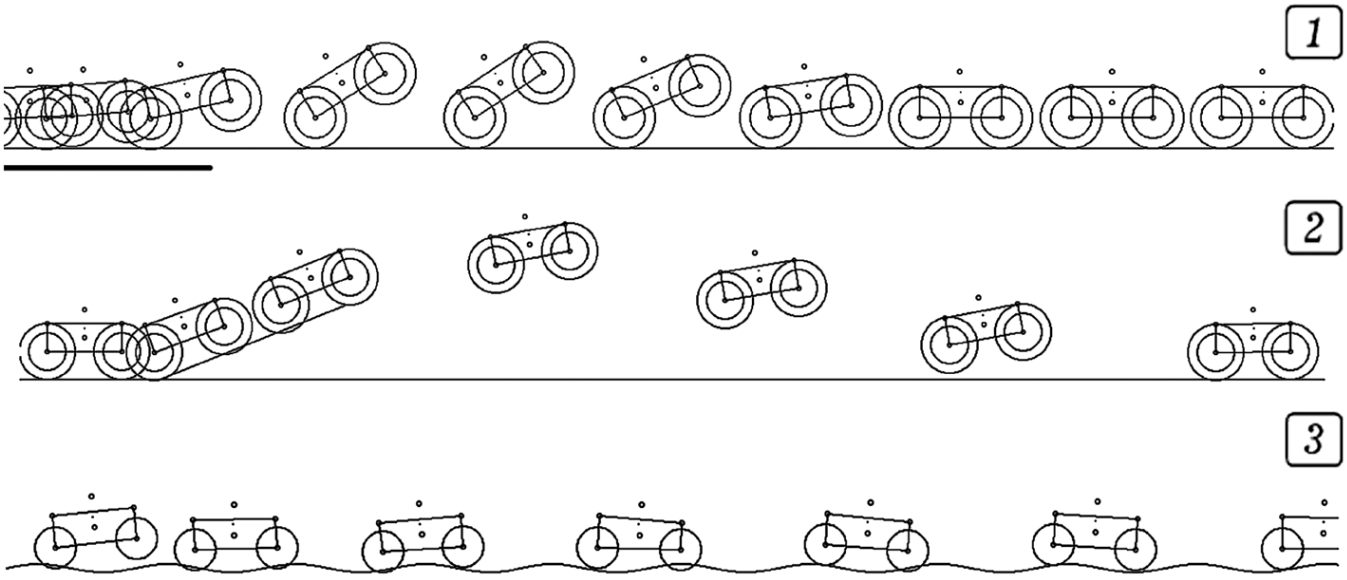

Рис. 14.17. Результаты моделирования движения мотоциклиста.

14.9. Моделирование ядерного реактора

Рассмотрим ядерный реактор -- устройство, в котором протекает **управляемая цепная ядерная реакция**, сопровождающаяся выделением

большого количества энергии (рис. 14.18.1). Он состоит из камеры 1, окруженной защитой, в которую помещают ядерное топливо и замедлитель нейтронов. Для управления работой реактора используются стержни 3, вдвигаемые в активную зону с помощью двигателя 4. Через активную зону прокачивается водяной или жидкостный теплоноситель, который поступает в теплообменник, где и передает тепло водяному пару, вращающему паровую турбину. Для контроля интенсивности ядерной реакции используют датчики радиоактивности и температуры 6, подключенные к устройству управления 5, которое включает двигатель, перемещающий управляющие стержни.

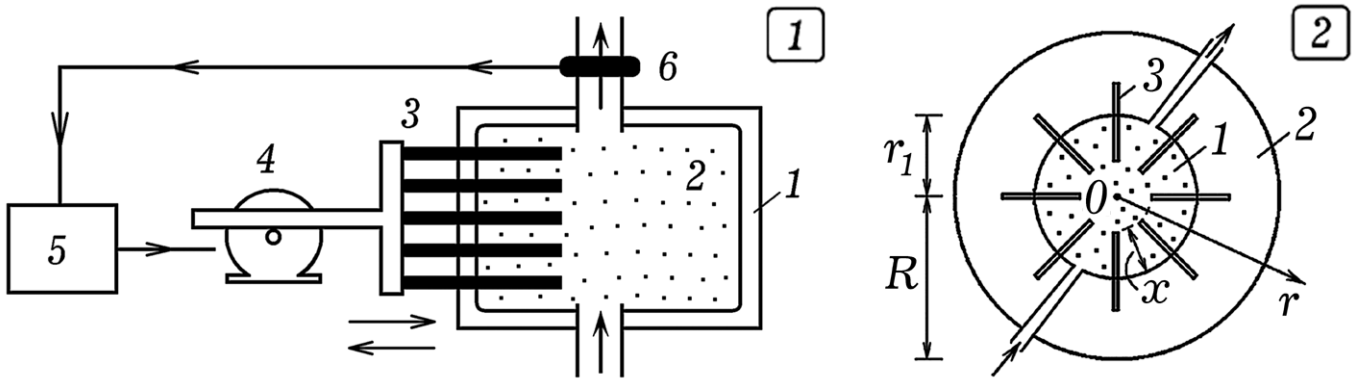

Рис. 14.18. Устройство ядерного реактора.

В реактор загружают ядерное топливо так, чтобы его масса превысила критическое значение. Начинается **цепная ядерная реакция**: в результате поглощения нейтрона ядром и его деления в среднем образуется $K \approx 2{,}5$ нейтронов. После деления ядра 99 % нейтронов появляются сразу же -- это мгновенные нейтроны. Остальные испускаются через некоторое время (10-50 с) после одного или двух актов бета-распадов получившихся осколков деления. Часть нейтронов захватываются другими ядрами, это также вызывает их деление. Чтобы уменьшить интенсивность ядерной реакции, в активную зону вводят управляющие стержни, поглощающие нейтроны [1, с. 336–340].

Рассмотрим сферический ядерный реактор (рис. 14.18.2), состоящий из ядерного топлива 1, защиты 2 и управляющих стержней 3. Модель должна учитывать образование нейтронов в результате ядерного распада,

их диффузию и теплопроводность среды [2, 7, 13]. Запишем соответствующие уравнения в сферических координатах:

$$\frac{\partial n}{\partial t} = k_1\left(\frac{\partial^2 n}{\partial r^2} + \frac{2}{r}\frac{\partial n}{\partial r}\right) + k_2(1-\beta)n + \lambda C - S\,,\quad \frac{\partial C}{\partial t} = k_3\beta n - \lambda C\,,$$

$$\frac{\partial T}{\partial t} = k_4\left(\frac{\partial^2 T}{\partial r^2} + \frac{2}{r}\frac{\partial T}{\partial r}\right) + \frac{q}{c\rho}\,,\quad q = k_5 n\,,$$

здесь n -- концентрация нейтронов на расстоянии r от центра O, β -- доля запаздывающих нейтронов, возникающих при распаде ядер-предшественников, $(1-\beta)$ -- доля мгновенных нейтронов, C и λ -- концентрация и постоянная распада ядер-предшественников, S -- быстрота изменения концентрации нейтронов из-за поглощения их управляющими стержнями, T -- температура, q -- теплота, выделяющаяся в результате ядерных распадов, c и ρ -- удельная теплоемкость и плотность среды (ядерное топливо, замедлитель нейтронов и теплоноситель).

Датчик температуры отслеживает температуру теплоносителя, прокачиваемого через активную зону, на его выходе получается напряжение U, пропорциональное средней температуре в центральной части активной зоны. В устройство управления заложена программа запуска, разгона и остановки реактора, а также регулирующая функция, связывающая напряжение U на выходе датчика с усилием F, создаваемым двигателем. Каждый управляющий стержень имеет массу m и соединен с демпфером, который необходим для затухания возникающих колебаний. Уравнение движения управляющих стержней имеет вид: $ma_x = F_x - r\upsilon_x$, где r -- коэффициент сопротивления. Увеличение x приводит к росту доли нейтронов, поглощаемых управляющими стержнями. Регулирующая функция, связывающая усилие двигателя F с напряжением U на выходе датчика, имеет вид: $F = \alpha(U - U_0) + \beta(U(t+dt) - U(t))/dt$. Здесь U_0 -- заданное значение напряжения, разность $U(t+dt) - U(t)$ пропорциональна быстроте изменения напряжения U. Первое слагаемое учитывает отклонение наблюдаемого U от заданного уровня U_0, а второе слагаемое -- быстроту

изменения U. Когда напряжение U превышает U_0 и/или быстро растет ($(U(t+dt)-U(t))/dt$ велико), двигатель действует на систему стержней с силой F, вдвигая их в активную зону на расстояние x. Для шарового реактора поглощающая способность материала с ростом расстояния r до центра O убывает обратно пропорционально r^2, поэтому $S = kn(r)/r^2$.

При заданных параметрах реактора подберем управляющую функцию $F(U)$ и зависимость $U_0(t)$, при которых он плавно входит в рабочий режим, увеличивает свою мощность до некоторого значения, а затем останавливается. Это **нестационарная задача реакторной физики**. Используемая программа ПР-10 содержит цикл по времени, в котором рассчитываются концентрация нейтронов n, число ядер предшественников C и температура T (процедура Raschet), определяется напряжение на выходе датчика и глубина x погружения стержня (процедура Upravlenie), результаты выводятся на экран (процедура Draw). При этом управляющая функция имеет вид: $F^t = 0.01(U^t/5 - U_0) + 10(U^t - U^{t-1})$, а зависимость напряжения U_0 от времени может быть задана так: 1) если $t < 50$, то $U_0 = 50$; 2) если $50 \le t < 150$, $U_0 = 50 + 5(t-50)$; 3) если $150 \le t < 350$, то $U_0 = 550$; 4) если $t > 350$, то $U_0 = 25$.

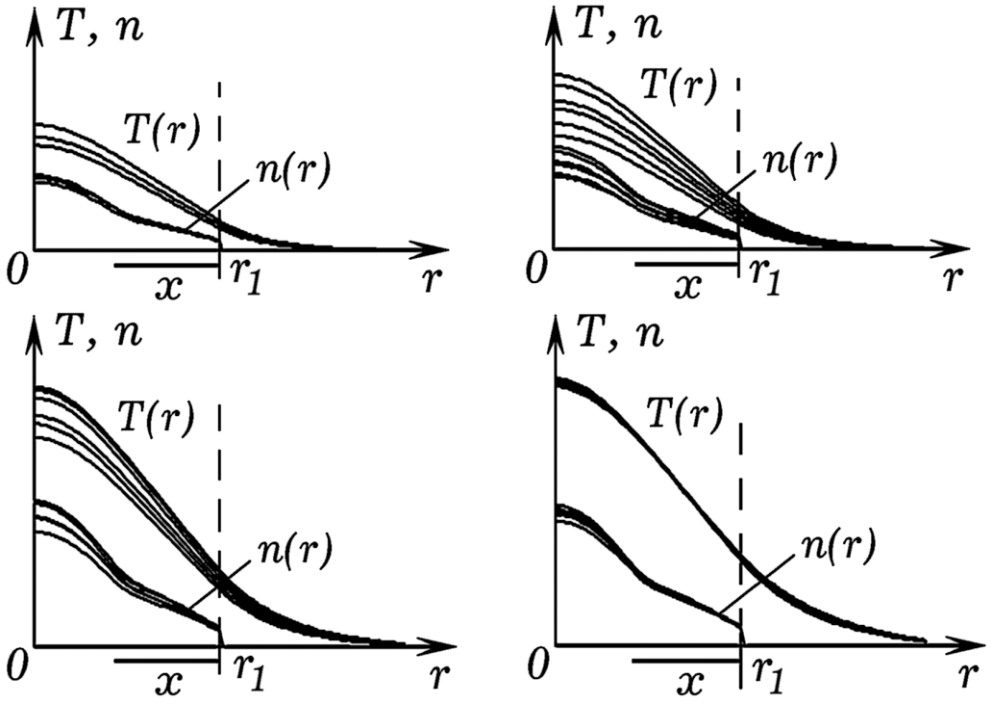

Рис. 14.19. Графики $n(r)$ и $T(r)$ в различные моменты времени.

Можно убедиться, что при таком задании управляющей функции $F(U)$ и зависимости $U_0(t)$ интенсивность "ядерной реакции" изменяется плавно, "реактор" работает без резких всплесков уровня "радиоактивности". На рис. 14.19 представлены графики распределения нейтронов $n(r)$ и температуры $T(r)$ в процессе работы реактора в различные моменты времени. Графики зависимостей $n(t)$, $T(t)$ и $x(t)$ при пуске, работе и остановке реактора показаны на рис. 14.20. Видно, что в установившемся режиме система находится в **динамическом равновесии**, управляющие стержни, а с ними концентрация нейтронов n и температура T, колеблются с небольшой амплитудой. В момент t_2 заданный уровень U_0 падает до нуля, стержни погружаются на максимальную глубину, ядерная реакция быстро затухает, температура понижается до температуры среды. Если управляющая функция $F(U)$ не будет зависеть от скорости изменения U (например, $F^t = 0.01(U^t/5 - U_0)$), то при запуске "реактора" происходит увеличение уровня "радиоактивности".

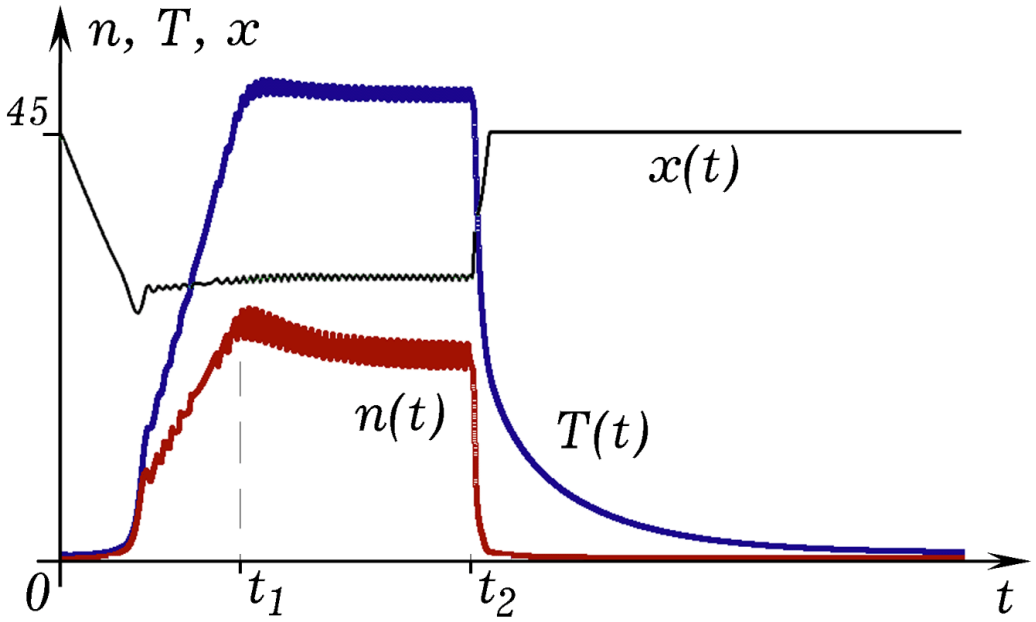

Рис. 14.20. Моделирование пуска и остановки ядерного реактора.

14.10. Точечная модель ядерного реактора

Если пренебречь размерами активной зоны, то получим точечную модель ядерного реактора [2, 7, 13]:

$$\frac{dn}{dt}=\frac{\rho(t)-\beta}{l}n+\lambda C+S,\quad \frac{dC}{dt}=\frac{\beta}{l}n-\lambda C,\quad \frac{\partial T}{\partial t}=\alpha_1 n-\alpha_2(T-T_0),$$

где n -- осредненная по объему концентрация нейтронов в реакторе, l -- среднее время генерации нейтронов, ρ -- реактивность, C и λ -- концентрация и постоянная распада ядер-предшественников, S -- мощность источника (поглотителя) нейтронов, α_1 и α_2 -- коэффициенты, характеризующие скорость изменения средней температуры в активной зоне в результате ядерного распада и теплообмена с окружающей средой, имеющей температуру $T_0=0$. Мощность реактора пропорциональна n; его реактивность $\rho=(K-1)/K$, где K -- коэффициент размножения нейтронов.

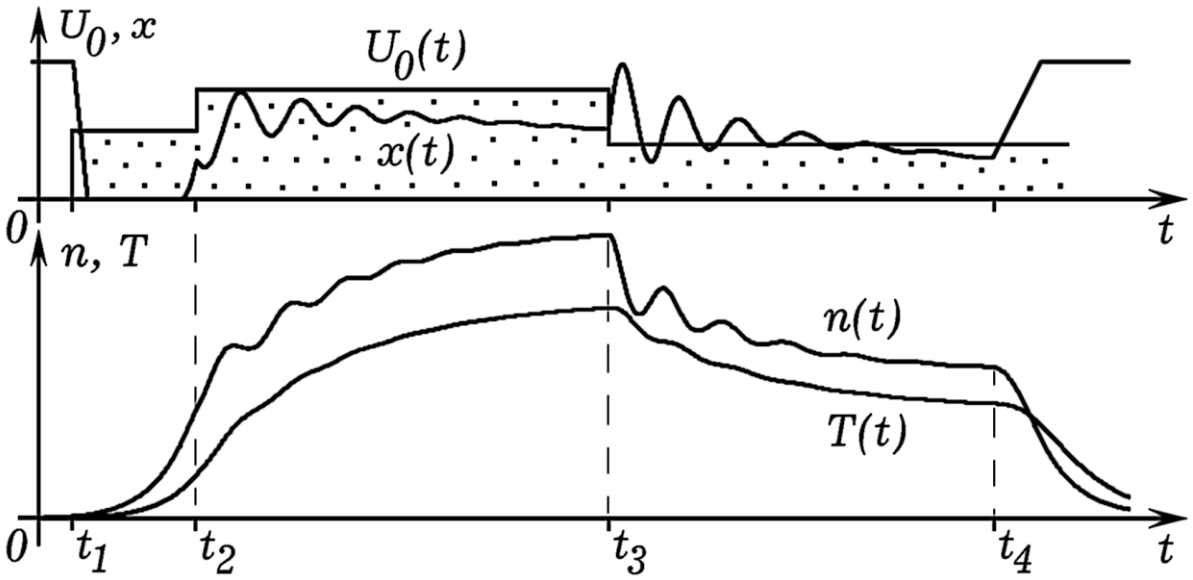

Рис. 14.20. Работа реактора при ступенчатом изменении $U_0(t)$.

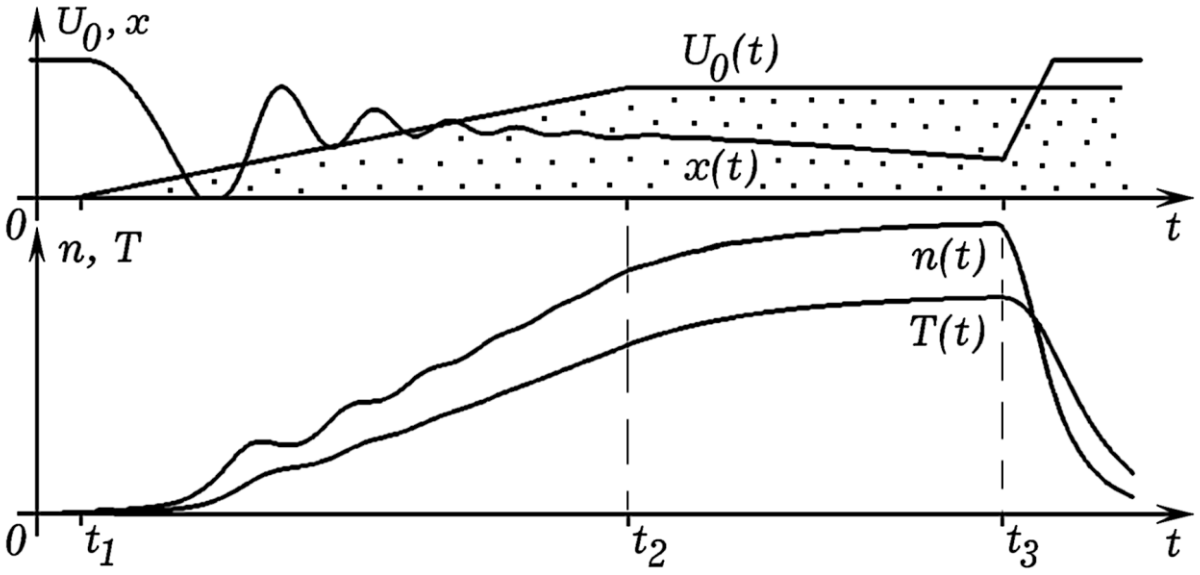

Рис. 14.21. Плавный запуск реактора. Управление по температуре.

Промоделируем работу ядерного реактора, управление которым осуществляется по температуре теплоносителя при ступенчатом изменении $U_0(t)$. Используется программа ПР-11, результаты представлены на рис. 14.20. Видно, что глубина погружения стержней x при резком изменении U_0 совершает затухающие колебания. В момент t_4 глубина x начинает увеличиваться, ядерная реакция прекращается. На рис. 14.21 представлены результаты моделирования работы реактора при плавном повышении U_0. Программа позволяет промоделировать аварийную ситуацию, когда интенсивность ядерной реакции, пропорциональная n, резко возрастает. Система автоматически реагирует и снова входит в нормальный режим.

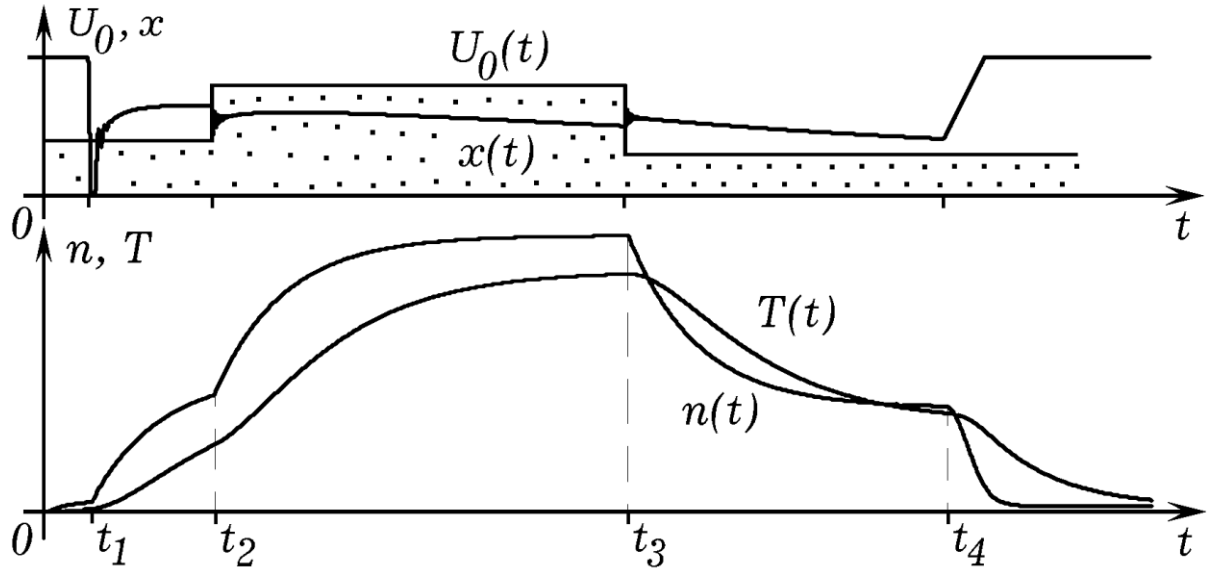

Рис. 14.22. Результаты моделирования пуска и остановки реактора.

На рис. 14.22 приведены результаты моделирования работы реактора, управляемого по уровню радиоактивности (программа ПР-12). Модель также позволяет проанализировать различные способы пуска ядерного реактора, и убедиться в том, что при резком изменении интенсивности ядерной реакции в небольших пределах система ведет себя устойчиво. Понятно, что рассмотренные выше модели являются приблизительными и представляют интерес только для обучения.

Приложение к главе 14

В приложении представлены тексты программ, которые позволяют промоделировать технические системы, которые рассмотрены выше. Они написаны в средах Borland Pascal 7.0 и Free Pascal 1.0.10.

Программа ПР-1.

```
Program Kanal_svyazi;
{$N+}Uses crt, dos;
Const Chislo_k=10000; Dlina_k=8; p=0.1; t1=1; t2=2;
Var j,N,i,t: longint; x,skorost: single; err: integer;
BEGIN
For i:=1 to Chislo_k do begin
j:=0; err:=0;
Repeat inc(t); inc(j);
   x:=random(1000)/1000;
   If x<p then begin
   Writeln('OSHIBKA V KADRE ',i,' BIT ',j, ' VREMYA', t);
   err:=1;  end;
until j=Dlina_k; t:=t+t1;
If err=0 then inc(N) else t:=t+t2;
writeln('KADR ',i,' ',N,' ',t); end;
skorost:=N*(Dlina_k-1)/t;
Writeln(t,' ',skorost); Readkey;
END.
```

Программа ПР-2.

```
Program Termoregulyator;
Uses crt, graph;
Const dtime=0.05; R=50; c=4800; m=0.5; alfa=5; MT=0.3;
Var k,i,j,DV,MV,x,x1,l: integer; T,T_sr,dQ,time,U:real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
T_sr:=5; T:=T_sr;
Repeat time:=time+dtime;
  If T>85 then U:=0; If T<70 then U:=220;
  If time>700 then T_sr:=T_sr+0.05*dtime;
  If T_sr>50 then T_sr:=50;
  dQ:=U*U/R*dtime-alfa*(T-T_sr)*dtime; T:=T+dQ/(c*m);
  Circle(10+round(time*MT),490-round(T*3),1);
  Circle(10+round(time*MT),490-round(T_sr*3),1);
until KeyPressed; CloseGraph;
END.
```

Программа ПР-3.

```
Program Termoreg_konvekciya;
{$N+}Uses crt, graph; {termoreg}
Const N=40; M=40; dt=0.0015; h=1; Tsr=0;
Var i,j,k,vkl,DV,MV,EC:integer; q,tt,ts,nu: single;
psi,w,T: array[0..N+1,0..M+1] of single;
Procedure Gr_usl;
begin For i:=0 to N+1 do For j:=0 to M+1 do begin
If (T[5,20]>2.5)and(vkl=1) then vkl:=0;
If (T[5,20]<2.0)and(vkl=0) then vkl:=1;
T[25,35]:=30*vkl; T[24,35]:=30*vkl; T[0,j]:=T[1,j];
T[N+1,j]:=T[N,j]; T[N+1,j]:=T[N,j]; T[M+1,j]:=T[M,j];
psi[0,j]:=0; psi[N+1,j]:=0; psi[i,0]:=0; psi[i,M+1]:=0;
w[i,M]:=2*psi[i,M]/h/h; w[1,j]:=2*psi[1,j]/h/h;
w[N,j]:=2*psi[N,j]/h/h; w[i,1]:=2*psi[i,1]/h/h;
w[0,j]:=0; w[N+1,j]:=0; w[i,M+1]:=0; w[i,0]:=0; end; end;
Procedure Ur1;
begin psi[i,j]:=(psi[i-1,j]+psi[i+1,j]+
psi[i,j-1]+psi[i,j+1]-w[i,j]*h*h)/4; end;
Procedure Ur2;
begin nu:=4;
w[i,j]:=w[i,j]-(psi[i,j+1]-psi[i,j-1])*(w[i+1,j]-
w[i-1,j])*dt/4/h+(psi[i+1,j]-psi[i-1,j])*(w[i,j+1]-
w[i,j-1])*dt/4/h+nu*(w[i-1,j]-4*w[i,j]+w[i+1,j]+
w[i,j-1]+w[i,j+1])/h/h*dt+2E-4*(T[i+1,j]-T[i-1,j]); end;
Procedure Ur3;
begin
If ((i=1)or(i=N)or(j=1)or(j=M))and(T[i,j]>Tsr)then
q:=0.003*(Tsr-T[i,j]) else q:=0;
T[i,j]:=T[i,j]+3*(T[i-1,j]-4*T[i,j]+T[i+1,j]+
T[i,j-1]+T[i,j+1])/h/h*dt+q-
(psi[i,j+1]-psi[i,j-1])*(T[i+1,j]-T[i-1,j])*dt/4/h+
(psi[i+1,j]-psi[i-1,j])*(T[i,j+1]-T[i,j-1])*dt/4/h; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat  Gr_usl; For i:=1 to N do For j:=1 to M do Ur1;
  Gr_usl; For i:=N downto 1 do For j:=M downto 1 do Ur1;
  Gr_usl; For i:=1 to N do For j:=1 to M do Ur2;
  Gr_usl; For i:=N downto 1 do For j:=M downto 1 do Ur2;
  Gr_usl; For i:=1 to N do For j:=1 to M do Ur3;
  Gr_usl; For i:=N downto 1 do For j:=M downto 1 do Ur3;
  inc(k); tt:=tt+0.001; If k mod 50=0 then begin k:=0;
{cleardevice;}{For i:=1 to N do For j:=1 to M do begin
  setcolor(abs(round(psi[i,j]/10))mod 12+2);
  rectangle(i*6,j*6,i*6+5,j*6+5);
  rectangle(i*6+1,j*6+1,i*6+4,j*6+4);
  rectangle(i*6+2,j*6+2,i*6+3,j*6+3);
  setcolor(abs(round(T[i,j]/2))mod 12+2);
```

```
  rectangle(i*6,j*6,i*6+5,j*6+5);
  rectangle(i*6+1,j*6+1,i*6+4,j*6+4);
  rectangle(i*6+2,j*6+2,i*6+3,j*6+3);
  setcolor(white); circle(5*6,20*6,2); end;} Ts:=0;
  For i:=1 to N do For j:=1 to M do Ts:=Ts+T[i,j]/N/M;
  circle(10+round(tt*5),520-round(120*T[5,20]),1);
  circle(11+round(tt*5),521-round(120*T[5,20]),1);
  circle(10+round(tt*5),520-round(120*Ts),1);
  circle(11+round(tt*5),521-round(120*Ts),1); end;
until KeyPressed; CloseGraph;
END.
```

Программа ПР-4.

```
Program Izmerit_pribor;
uses crt, graph;
const dt=0.0001; MIner=0.001; R=3; L=0.005;
k1=0.01; k2=0.01; k3=0.002; k4=0.1; Mt=100;
var DV, MV: integer;
    ii,i,t,U,fi,fi1,eps,w,Myp,Mel,Mtr: real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat t:=t+dt;
   If sin(t*2)>0 then U:=3 else U:=0;
 { U:=2*abs(sin(t*4));} {If sin(t)>0 then U:=7*sin(t);}
 { If sin(t*10)>0 then U:=7*sin(t*10) else U:=0.04;}
   ii:=(U-R*i-k4*(fi-fi1)/dt)/L; i:=i+ii*dt;
   Myp:=-k1*fi; Mel:=k2*i; Mtr:=-k3*w;
   eps:=(Mel+Myp+Mtr)/MIner; fi1:=fi;
   w:=w+eps*dt; fi:=fi+w*dt;
   If fi>3.14/2 then begin fi:=3.14/2; w:=-0.8*w; end;
   If fi<0 then begin fi:=0; w:=-0.8*w; end;
   circle(10+round(t*Mt),450-round(U*40),1);
   circle(10+round(t*Mt),450-round(fi*200),1);
until KeyPressed; CloseGraph;
END.
```

Программа ПР-5.

```
Program Asinhron_dvig;
uses crt, graph; const dt=0.0005; I=0.1; w1=314; Rr=1;
Sr=0.0002; Lr=0.007; Nr=200; Mt=10; Bs=20;
var DV,MV,j,k: integer; ir,t,t1,M,Mn,Mtr,SM,fi,ffi,w2,wsr,
eps,Eir,alfa,Fr1,Fr2 :real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); k:=1;
Repeat t:=t+dt;
  alfa:=w1*t-fi; M:=Nr*ir*Sr*Bs*sin(alfa); Mtr:=0.002*w2;
  eps:=(M-Mtr-Mn)/I; w2:=w2+eps*dt; fi:=fi+w2*dt;
  Fr1:=Fr2; Fr2:=Bs*Sr*cos(alfa); Eir:=-Nr*(Fr2-Fr1)/dt;
  ir:=(Eir*dt+Lr*ir)/(Rr*dt+Lr);
```

```
  If t>20 then Mn:=5; If t>30 then Mn:=10;
  If t>45 then Mn:=15; inc(k); t1:=t1+dt;
  If w2<0 then w2:=0; ffi:=ffi+w2*dt; SM:=SM+M;
  If t1>1.5 then begin wsr:=ffi/t1; ffi:=0; t1:=0;
  circle(round(Mt*t),450-round(15*SM/k),2); SM:=0; k:=1;
end; circle(round(Mt*t),460,1);
 {circle(round(Mt*t),460-round(1*M),1);}
  circle(round(Mt*t),460-round(15*Mn),1);
  circle(round(Mt*t),460-round(1.3*w1),1);
  circle(round(Mt*t),460-round(1.3*w2),1);
until KeyPressed; CloseGraph;
END.
```

Программа ПР-6.

```
Program Dvigat_generator;
uses crt, graph;
const dt=0.0005; I=0.1; w1=314; Rr=1; Lg=0.03;
      Sr=0.0002; Lr=0.007; Nr=200; Mt=10; k2=2;
var DV,MV,j,k,Rn: integer; ir,t,t1,M,Mn,Mtr,SM,fi,ffi,w2,
wsr,eps,Eir,alfa,Fr1,Fr2,Bs,Fig1,Fig2,eg,ig :real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
line(0,400,640,400); k:=1; Rn:=5000;
Repeat t:=t+dt; Bs:=20; {dvigatel:}
alfa:=w1*t-fi; M:=Nr*ir*Sr*Bs*sin(alfa); Mtr:=0.002*w2;
eps:=(M-Mtr-Mn)/I; w2:=w2+eps*dt; fi:=fi+w2*dt;
Fr1:=Fr2; Fr2:=Bs*Sr*cos(alfa);
Eir:=-Nr*(Fr2-Fr1)/dt; ir:=(Eir*dt+Lr*ir)/(Rr*dt+Lr);
If t>15 then Rn:=100; If t>30 then Rn:=10; {generator:}
Fig2:=Fig1; Fig1:=10*cos(fi); eg:=(Fig2-Fig1)/dt;
ig:=(eg*dt-Lg*ig)/(Rn*dt+Lg); Mn:=k2*ig*sin(fi); inc(k);
t1:=t1+dt; If w2<0 then w2:=0; ffi:=ffi+w2*dt; SM:=SM+M;
If t1>1.5 then begin wsr:=ffi/t1; ffi:=0; t1:=0;
circle(round(Mt*t),400-round(15*SM/k),2); SM:=0; k:=1; end;
circle(round(Mt*t),400-round(3*ig),1);
circle(round(Mt*t),400-round(1*w1),1);
circle(round(Mt*t),400-round(1*w2),1);
until KeyPressed; CloseGraph;
END.
```

Программа ПР-7.

```
Program Dvigatel_nagruzka;
uses crt, graph; { ПР - 4 }
const dt=0.001; MIner=0.4; R=5; L=0.05; K1=0.6; K2=0.5;
      K3=0.02; Mt=10;
var DV, MV : integer; i,t,U,eps,w,M,Mtr,Mn : real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Line(0,450,800,450);
Repeat t:=t+dt; Mtr:=K3*w;
```

```
If t>2 then U:=100;  If t>35 then Mn:=8;
eps:=(K1*i-Mtr-Mn)/MIner; w:=w+eps*dt;
i:=i+(U-R*i-K2*w)/L*dt;
circle(10+round(t*Mt),450-round(i*15),1);
circle(10+round(t*Mt),450-round(w*2.5),1);
circle(10+round(t*Mt),450-round(Mn*7),1);
until KeyPressed; CloseGraph;
END.
```

Программа ПР-8.

```
program Regul_scorosti;
uses crt, graph;
const dt=0.001; MIner=0.5; R=5; L=0.05;
K1=0.8; K2=1.2; K3=0.05; Mt=5; da=0.001;
var DV, MV : integer;
a,dU,i,t,U,eps,w,M,Mtr,Mn : real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Line(0,400,800,400); a:=150;
Repeat t:=t+dt; Mtr:=K3*w;
  If t>70 then {Mn:=1.1*(t-70);}Mn:=20;
  eps:=(K1*i-Mtr-Mn)/MIner; w:=w+eps*dt;
  i:=i+(U-R*i-K2*w)*dt/L;
  If w<a-da then dU:={0.002*(a-w);}+0.03;
  If w>a+da then dU:={0.02*(a-w);} -0.05; U:=U+dU;
  circle(10+round(t*Mt),400-round(i*3),1);
  circle(10+round(t*Mt),400-round(w*1.8),1);
  circle(10+round(t*Mt),400-round(Mn*2),1);
until KeyPressed; CloseGraph;
END.
```

Программа ПР-9.

```
Program Motocikl;
{$N+}uses crt, graph;
const dt=0.001; N=6; x0=10{-900}; y0=30; r3=1;
dd=150{-900}; Lg=200;
var kk,nn,i,i1,i2,j,DV,MV: integer;
xc,yc,mm,Ft,yy,r6,k,r,t,l,F,ax,ay,a,FN: single;
m,x,y,vx,vy,Fx,Fy: array[1..N] of single;
s: array[1..N,1..N] of single;
Procedure Nach;
begin x[1]:=-40+x0; y[1]:=y0; x[2]:=40+x0; y[2]:=y0;
x[3]:=x0; y[3]:=15+y0; x[4]:=x0; y[4]:=y0-15;
x[5]:=-40+x0; y[5]:=y0-30; x[6]:=40+x0; y[6]:=y0-30;
For i:=1 to 4 do m[i]:=180; m[3]:=50; m[2]:=180;
m[5]:=15; m[6]:=15;
For i:=1 to 6 do begin vx[i]:=10; mm:=mm+m[i]; end; end;
Procedure Sila; label Metka;
begin For i:=1 to N do begin Fx[i]:=0; Fy[i]:=0; end;
```

```
For i:=1 to N do For j:=1 to N do begin
  If (j=i) then goto Metka; k:=2E+2; r:=1000;
  If (i>4)or(j>4) then begin k:=0.005; r:=200000; end;
  l:=sqrt(sqr(x[i]-x[j])+sqr(y[i]-y[j]));
  F:=k*(s[i,j]-l); {If l<s[i,j]*0.9 then F:=10000;}
  Fx[i]:=Fx[i]+(F-r*(l-s[i,j]))*(x[i]-x[j])/(l+1E-3);
  Fy[i]:=Fy[i]+(F-r*(l-s[i,j]))*(y[i]-y[j])/(l+1E-3)
  -m[i]*3.8; Metka: end; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); Nach;
For i:=1 to 6 do For j:=1 to 6 do
s[i,j]:=sqrt(sqr(x[i]-x[j])+sqr(y[i]-y[j]));
Repeat t:=t+dt; inc(kk); Sila;
For i:=1 to 6 do begin {If x[i]<150 then yy:=35 else}
yy:=0; a:=0;
If (x[i]>dd)and(x[i]<Lg+dd) then begin yy:=0.4*(x[i]-dd);
a:=arctan(0.4); end; FN:=0; r6:=100;
If y[i]<yy then begin FN:=abs(20000*(y[i]-yy))*cos(a);
If y[i]<yy-4 then FN:=3E+5; r6:=1000; end;
If {(x[5]>20)and}(y[5]<yy)and(i=5){and(t<1.5)} then begin
FT:=18000*(yy-y[5])*cos(a); end else FT:=0;
ax:=(Fx[i]+FT*cos(a)-FN*sin(a)-r3*vx[i])/m[i];
ay:=(Fy[i]+FT*sin(a)+FN*cos(a)-r3*vy[i]-r6*vy[i])/m[i];
vx[i]:=vx[i]+ax*dt; vy[i]:=vy[i]+ay*dt;
x[i]:=x[i]+vx[i]*dt; y[i]:=y[i]+vy[i]*dt; end;
If kk mod 2500=0 then begin kk:=0; {cleardevice;}
line(0,430,940,430); {line(0,400,150,400);}
line(dd,430,dd+Lg,430-round(0.4*Lg));
line(round(x[1]),400-round(y[1]), round(x[2]),400-
round(y[2])); line(round(x[1]),400-round(y[1]),round(x[5]),
400-round(y[5])); line(round(x[2]),400-round(y[2]),
round(x[6]),400-round(y[6])); line(round(x[5]),400-
round(y[5]),round(x[6]),400-round(y[6]));
For i:=1 to 6 do begin xc:=xc+m[i]*x[i];
yc:=yc+m[i]*y[i]; end; xc:=xc/mm; yc:=yc/mm;
circle(round(xc),round(400-yc),1);
For i:=1 to 6 do begin
circle(round(x[i]),round(400-y[i]),2); end;
circle(round(x[5]),round(400-y[5]),20);
circle(round(x[5]),round(400-y[5]),30);
circle(round(x[6]),round(400-y[6]),20);
circle(round(x[6]),round(400-y[6]),30); end;
until (KeyPressed)or(t>50); ReadKey; CloseGraph;
END.
```

Программа ПР-10.

```
program Yadernii_reaktor1;
{$N+}uses crt, graph; const h=1; dt=0.002; r1=50; r2=100;
```

```
b=0.1; L1=0.05; k1=250; k2=1.6; rs=1.5;
var i,j,time,DV,MV: integer;
v,tt,F,kk,U0,NN,x,p,sl,U,U1: single;
T,T1,N,N1,C1: array[0..100] of single;
Procedure Upravlenie;
begin U:=0; For i:=0 to 40 do U:=U+T[i];
If tt<50 then U0:=50 else U0:=50+5*(tt-50);
If tt>150 then U0:=550; If tt>350 then U0:=25;
{If U/4>U0 then x:=x+0.8*dt;
If U/4<U0 then x:=x-0.8*dt;}
{If SUM>AA*1.2 then F:=0.2; If SUM<AA then F:=-0.2;}
F:=0.01*(U/5-U0)+10*(U-U1); U1:=U; v:=v+(F-rs*v)*dt;
x:=x+v*dt; If x>45 then x:=45; If x<1 then x:=1; end;
Procedure Raschet;
begin N[0]:=N[1]; N[50]:=N[49]*0.9; NN:=0;
T[100]:=T[99]*0.9; T[0]:=T[1];
For i:=48 to 50 do NN:=NN+N[i];
For i:=1 to r1-1 do begin sl:=random(20)/100;
If i>round(r1-x) then p:=-1200/i/i else p:=0;
N1[i]:=N[i]+(k1*((N[i+1]-2*N[i]+N[i-1])/(h*h)+2/i*(N[i]-
N[i-1])/h)+sl+(1-b)*k2*N[i]+L1*C1[i]+p*N[i])*dt; end;
For i:=1 to r2-1 do begin If i>r1 then kk:=10 else kk:=140;
T1[i]:=T[i]+(kk*((T[i+1]-2*T[i]+T[i-1])/(h*h)+
2/i*(T[i]-T[i-1])/h)+0.8*N[i])*dt; end;
For i:=1 to r1-1 do N[i]:=N1[i];
For i:=1 to 99 do T[i]:=T1[i];
For i:=0 to r1 do C1[i]:=C1[i]+(b*k2*N[i]-L1*C1[i])*dt; end;
Procedure Draw;
begin  time:=1; {cleardevice;
For i:=1 to 100 do begin If i>r1-x then
circle(20+i*5,430,1); line(i*5+20,420-round(2*T[i]),(i-1)*5
+20,420-round(2*T[i-1])); line(i*5+20,420-round(2*N[i]),
(i-1)*5+20,420-round(2*N[i-1])); end; line(20,420,500,420);
line(20,0,20,420);} circle(round(tt),450-round(T[10]*3),1);
circle(round(tt),450-round(N[10]*3),1);
circle(round(tt),450-round(x*6),1); end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); Randomize;
For i:=1 to r1 do N[i]:=random(100)/100; x:=45;
Repeat tt:=tt+dt; inc(time); Upravlenie; Raschet;
If time mod 20=1 then Draw;
until KeyPressed; CloseGraph;
END.
```

Программа ПР-11.

```
program Yadernii_reaktor2;
{$N+}uses crt, graph; const dt=0.005; l=25; lam=0.3;
b=0.1; m=0.02; a1=30; a2=0.04; k=0.0005;
```

```
var DV,MV: integer;
F,A,x,v,rho,n,n1,C,q,t,Tem,Tem1: real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
rho:=1.05; x:=100; A:=0; q:=1;
Repeat t:=t+dt; n1:=n; Tem1:=Tem; rho:=rho-1E-3*dt;
q:=0.3; n:=n+((rho-b)/l*n+lam*C+q-k*x*n)*dt;
C:=C+(b/l*n-lam*C)*dt; Tem:=Tem+a1*n*dt-a2*Tem*dt;
If n<0 then n:=0; If C<0 then C:=0;
{If t>30 then A:=250; If t>120 then A:=400;
If t>420 then A:=200;} If t>30 then A:=(t-30);
If t>430 then A:=400; {If t>420 then A:=400-(t-420);}
F:=0.1*(Tem/2000-A)+0.005*(Tem-Tem1)/dt;
{F:=0.001*(n-A);} v:=v+(F-2.5*v)*dt/m;
If t<700 then x:=x+v*dt else x:=x+0.01;
If x<0 then x:=0; If x>100 then x:=100;
circle(round(t),520-round(n/5),1); circle(round(t),
520-round(Tem/5000),1); circle(round(t),520,1);
circle(round(t),250-round(A/5),1); circle(round(t),250,1);
circle(round(t),110-round(x),1); circle(round(t),110,1);
until KeyPressed; ReadKey; CloseGraph;
END.
```

Программа ПР-12.

```
program Yadernii_reaktor3;
{$N+}uses crt, graph;
const dt=0.005; l=10; lam=0.3; b=0.1; m=0.02;
a1=20; a2=0.02;
var DV,MV: integer;
F,A,x,v,rho,n,n1,C,q,t,Tem: real;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
rho:=1.4; x:=100; A:=0; q:=1;
Repeat t:=t+dt; q:=1; n1:=n; rho:=rho-1E-3*dt;
n:=n+((rho-b)/l*n+lam*C+q-0.002*x*n)*dt;
C:=C+(b/l*n-lam*C)*dt; Tem:=Tem+a1*n*dt-a2*Tem*dt;
If n<0 then n:=0; If C<0 then C:=0; If t>30 then A:=200;
If t>120 then A:=400; If t>420 then A:=150;
F:=2*(n-A)*dt+2E+4*(n-n1)*dt; v:=v+(F-0.01*v)*dt/m;
If t<650 then x:=x+v*dt else x:=x+0.01;
If x<0 then x:=0; If x>100 then x:=100;
circle(round(t),520-round(n/2),1); circle(round(t),
520-round(Tem/2300),1); circle(round(t),520,1);
circle(round(t),250-round(A/5),1);
circle(round(t),250,1); circle(round(t),110-round(x),1);
circle(round(t),110,1);
until KeyPressed; ReadKey; CloseGraph;
END.
```

Литература

1. Акоста В. и др. Основы современной физики / В. Акоста, К. Кован, Б. Грэм; Пер. с англ. -- М.: Просвещение, 1981. -- 495 с.
2. Бартоломей Г.Г. Основы теории и методы расчета ядерных энергетических реакторов: Учеб пособие для вузов / Г.Г.Бартоломей, Г.А.Бать, В.Д.Байбаков, М.С.Алхутов -- М.: Энергоиздат, 1982. -- 512 с.
3. Басов К.А. ANSYS в примерах и задачах. -- М.: КомпьютерПресс, 2002. -- 224 с.
4. Бесекерский В.А., Попов Е.П. Теория систем автоматического регулирования. -- М.: Наука, 1975. -- 768 с.
5. Бусленко Н.П. Моделирование сложных систем. -- М.: Наука, 1968. -- 356 с.
6. Воронин А.В. Моделирование мехатронных систем: учебное пособие -- Томск: Изд-во Томского политехнического университета, 2008.--127 с.
7. Ганев И.Х. Физика и расчет реактора: Учеб. пособие для вузов. -- М.: Энергоатомиздат, 1992. -- 496 с.
8. Герман-Галкин С.Г. Компьютерное моделирование полупроводниковых систем в MATLAB 6.0: Учебное пособие -- СПб.: КОРОНА принт, 2001 -- 320 с.
9. Майер Р.В. Задачи, алгоритмы, программы: Электронное учебное пособие. – Глазов: Глазовск.гос.пед.ин-т, 2012. http://maier-rv.glazov.net
10. Неймарк Ю.И. Математические модели в естествознании и технике: Учебник. -- Н. Новгород: Издательство Нижегородского госуниверситета, 2004. -- 401 с.
11. Самарский А.А., Михайлов А.П. Математическое моделирование: Идеи. Методы. Примеры. -- М: Физматлит, 2001. -- 320 с.
12. Тарасик В.П. Математическое моделирование технических систем: Учебник для вузов. -- Мн.: ДизайнПРО, 2004 -- 640 с.
13. Хетрик Д. Динамика ядерных реакторов -- М.:Атомиздат, 1975. – 400с.
14. Чикуров Н.Г. Моделирование технических систем. Учебное пособие. -- Уфимск. гос. авиац. техн. ун-т. -- Уфа: УГАТУ 2009. -- 357 с.
15. Giordano N.J. Computational Physics. -- New Jersey, Prentice Hall, 1997. -- 419 p.
16. Milagre A.M., Ferreira da Luz M.V., Cangane G.M., Komar A., Avelino P.A. 3D Calculation and Modeling of Eddy Current Losses in a Large Power Transformer // XXth International Conference on Electrical Machines – ICEM 2012, vol. 1, p. 2280-2284, 2 a 5 de setembro de 2012, Marseille/Franca.