

MATLAB 8.0 (R2012b): математическое и физическое моделирование, Simulink 8.0

Новая (август 2012 г.) матричная система компьютерной математики MATLAB 8.0 (R2012b) приспособлена для решения фундаментальных и прикладных задач математического и физического моделирования. В статье впервые в отечественной литературе описаны возможности моделирования как в среде самой системы MATLAB 8.0 с применением современных аналитических и численных методов, так и с помощью обновленного пакета расширения этой системы Simulink 8.0, реализующего блочное визуально-ориентированное имитационное моделирование. Приведенные примеры моделирования ориентированы на решение широкого круга задач и знакомство с основными методами их решения. Автор благодарит корпорацию The MathWorks за предоставленную систему MATLAB 8.0 + Simulink 8.0, использованную для подготовки этой серии статей.

Владимир ДЬЯКОНОВ,
д. т. н., профессор
vpdyak@yandex.ru

Состав систем MATLAB 8.0 + Simulink 8.0

Система MATLAB + Simulink (рис. 1) состоит из нескольких частей [1]. Сама система MATLAB 8.0 — это один из лучших языков программирования высокого уровня, ориентированный на матричные операции и простой синтаксис, минимизирующий обращение к циклам [2, 3]. Он содержит около 1000 встроенных операторов и функций, большинство из которых написано на нем самом в виде файлов, имеющих открытый код и допускающих знакомство с ними и модификацию пользователем. Есть средства работы и с другими языками. Четкая графика, удобные средства графического интерфейса

пользователя, встроенный редактор программного кода и компилятор добавляют достоинства системе MATLAB. Рабочий стол новой версии MATLAB 8.0 сразу дает пользователю эти и многие другие возможности.

Порою нескольких строк программного кода MATLAB достаточно для составления программы математического моделирования фундаментальной задачи в области физики, механики, гидравлики, электро- и радиотехники. Но подлинную уникальность системе придают почти сотня поставляемых с ней пакетов расширения, относящихся к разным областям науки и техники. Они объединены в два обширных инструментальных ящика — Toolbox для MATLAB и Blockset для Simulink (в MATLAB 8.0 пакеты расширения ящика

Blockset имеют название, начинающееся с сокращения Sim).

Основной, построенный на основе системы MATLAB, пакет расширения Simulink автоматизирует технику имитационного блочного моделирования систем и устройств, требуя лишь подготовки диаграмм (схем) их моделей [3–5]. Он выполняет сложнейший матрично-топологический анализ диаграмм, автоматическое составление систем описывающих их алгебраических и дифференциальных уравнений, их решение указанным пользователем численным методом и наглядную графическую иллюстрацию решения с генерацией его отчета. Система Stateflow обеспечивает проведение моделирования систем с переменной во времени структурой.

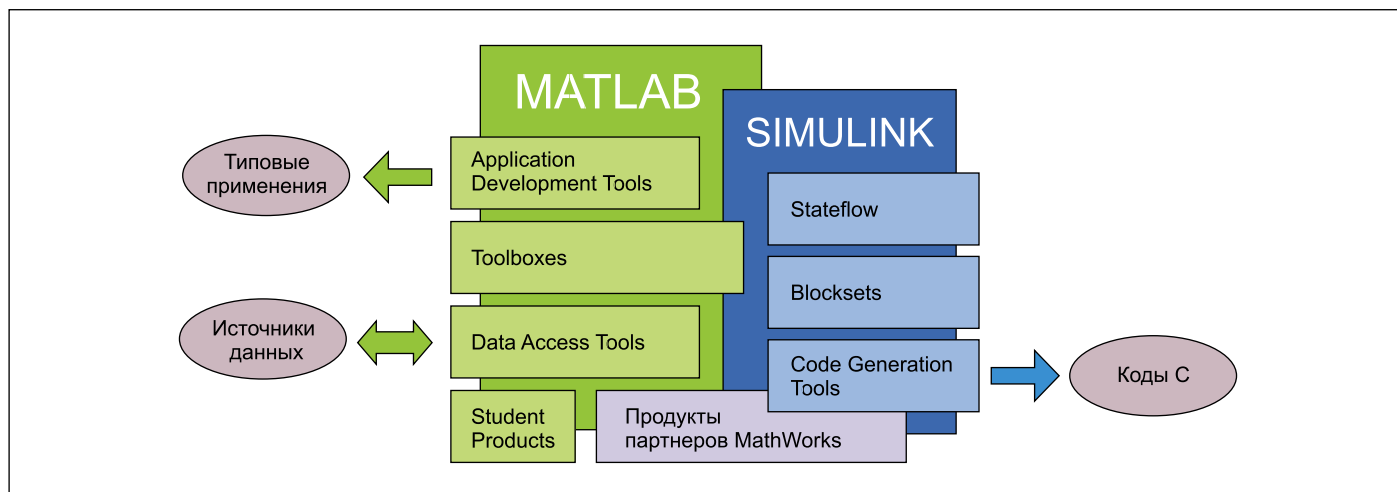


Рис. 1. Структура системы MATLAB+Simulink

Простейшие примеры математического моделирования цепей постоянного тока

Появление в MATLAB пакета расширения Symbolic Math Toolbox позволило перейти к широкому применению в решении электро-, радиотехнических и других задач символьных методов вычислений, имеющих высокую общность и точность и не дающих неустойчивости, часто присущей численным методам. И это видно уже при решении самых простых «калькуляторных» задач. Например, при параллельном включении нескольких резисторов их проводимости складываются:

$$\frac{1}{R_0} = \frac{1}{R_1} + \frac{1}{R_2} \dots$$

Если известны все сопротивления, кроме одного, то его нетрудно найти, используя функцию решения уравнений *solve* относительно неизвестного сопротивления. Например:

```
>> simplify(solve('1/R0=1/R1+1/R2','R0'))
ans =
(R1*R2)/(R1 + R2)
>> simplify(solve('1/R0=1/R1+1/R2','R2'))
ans =
-(R0*R1)/(R0 - R1)
```

Далее рассмотрим классический пример на передачу энергии от источников постоянного напряжения на нагрузку. Пусть к зажимам двух последовательно соединенных источников энергии $E_1 = 12$ В и $E_2 = 48$ В с внутренними сопротивлениями $r_1 = 0,4$ Ом и $r_2 = 0,6$ Ом подключен резистор нагрузки с изменяющимся сопротивлением R (рис. 2). Найдём значение R , при котором получаемая резистором R мощность максимальна:

```
>> P='R*(E1+E2)^2/(r1+r2+R)^2'; dP=diff(P)
dP =
(E1+E2)^2/(r1+r2+R)^2-2*r*(E1+E2)^2/(r1+r2+R)^3
>> rpmax=solve(dP,'R')
rpmax =
r1+r2
```

Итак, максимальная мощность в нагрузке выделяется при сопротивлении нагрузки, равном сумме внутренних сопротивлений включенных последовательно источников напряжения. При этом КПД системы равен 1/2. При расчете полагалось, что максимум $P(R)$ соответствует нулю производной этой зависимости.

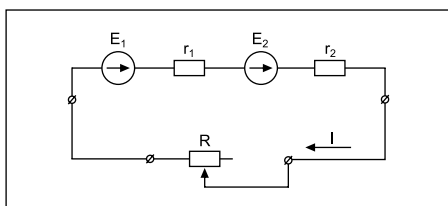


Рис. 2. Передача энергии от двух источников постоянного напряжения

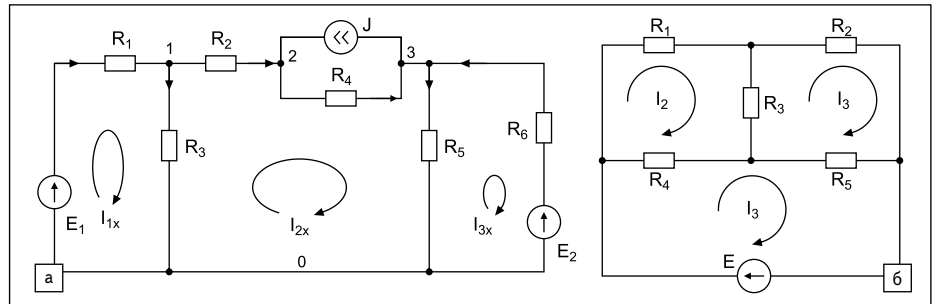


Рис. 3. Примеры мостовых схем

Методы узловых потенциалов и контурных токов

При расчете более сложных цепей широко используются методы узловых потенциалов и контурных токов. Например, для схемы на рис. 3а можно в MATLAB записать следующую, вполне очевидную программу:

```
>> R=[4 40 16 80 20 8]; E=[100 20]; J=2;
R11=R(1)+R(2); R22=R(2)+R(3)+R(4)+R(5); R33=R(5)+R(6);
R12=-R(2); R23=-R(5); R13=0; EE1=E(1); EE2=-J*R(4); EE3=-E(2);
RR=[R11 R12 R13; R12 R22 R23; R13 R23 R33]
EE=[EE1 EE2 EE3]; Ik=EE/RR
Ik=[Ik(1) (Ik(1)-Ik(2)) Ik(2) (Ik(2)+) (Ik(2)-Ik(3)) -Ik(3)]
```

После ее исполнения (в редакторе или прямо в командной строке) получим:

```
RR =
44 -40 0
-40 156 -20
0 -20 28
EE =
100 -160 -20
Ik =
1.5533 -0.7914 -1.2796
Ik =
1.5533 2.3447 -0.7914 1.2086 0.4882 1.2796
```

Теперь рассмотрим пример расчета широко распространенной мостовой схемы (рис. 3а) методом узловых потенциалов:

```
>> Q1='R4*I1-R4*I2+R5*I1-R5*I3-Es=0';
Q2='R3*I3-R3*I2+R2*I3+R5*I3-R5*I1=0';
Q3='R4*I2-R4*I1+R1*I2+R3*I2-R3*I3=0';
S=solve(Q1,Q2,Q3,'I1','I2','I3')
```

Аналитическое решение получается как символьные значения трех переменных:

```
S =
I1: [1x1 sym]
I2: [1x1 sym]
I3: [1x1 sym]
```

Выведем значение одной из них (ток в первом контуре):

```
>> S.I1
Ans =
(Es*R1*R2 + Es*R1*R3 + Es*R2*R3 + Es*R1*R5 + Es*R2*R4 +
Es*R3*R4 +
Es*R3*R5 + Es*R4*R5)/(R1*R2*R4 + R1*R2*R5 + R1*R3*R4 +
R1*R3*R5 +
R2*R3*R4 + R1*R4*R5 + R2*R3*R5 + R2*R4*R5)
```

Выражение получается длинным, но, при необходимости, легко вычислить численное

значение тока при заданных параметрах этой схемы:

```
>> R1=8;R2=1;R3=10;R4=4;R5=15;Es=40;
(Es*R1*R2 + Es*R1*R3 + Es*R2*R3 + Es*R1*R5 + Es*R2*R4 +
Es*R3*R4 + Es*R3*R5 + Es*R4*R5)/(R1*R2*R4 + R1*R2*R5 +
R1*R3*R4 +
R1*R3*R5 + R2*R3*R4 + R1*R4*R5 + R2*R3*R5 + R2*R4*R5)
ans =
7.8601
```

Нетрудно заметить, что при расчете цепей постоянного тока используются решения алгебраических уравнений для таких цепей.

Переходные процессы в RC-цепи

Рассмотрим вначале переходные процессы в простейшей линейной RC-цепи, подключенной к источнику напряжения E . Они описываются линейным дифференциальным уравнением первого порядка:

$$E = (RC) \frac{du}{dt} + u,$$

которое в MATLAB аналитически решается следующим образом:

```
>> dsolve('tau*D u+u=E')
ans = E+exp(-1/tau*t)*C1
```

Если задано начальное условие $u(0) = 0$, то:

```
>> dsolve('tau*D u+u=E','u(0)=0')
ans = E-exp(-1/tau*t)*E
simplify(dsolve('tau*D u+u=E','u(0)=0'))
ans = -E*(-1+exp(-1/tau*t))
>> simplify(dsolve('tau*D u+u=E','u(0)=0'))
-exp(1)*(exp(-t/tau) - 1)
```

Здесь получено уравнение заряда конденсатора в классическом виде. Чаще всего такой вид решения позволяет получить функция упрощения *simplify*. А теперь вычислим реакцию RC-цепи (рис. 4) на экспоненциаль-

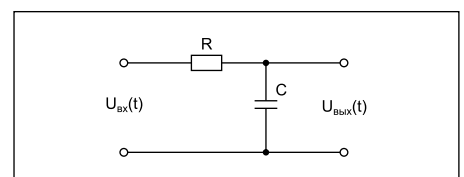


Рис. 4. Интегрирующая RC-цепь

ный перепад и на синусоидальный сигнал с экспоненциально спадающей амплитудой:

```
>> simplify(dsolve('tau*Du+u=1-exp(-t/tau)', 'u(0)=0'))
>> simplify(dsolve('tau*Du+u=sin(w*t)', 'u(0)=0'))
>> dsolve('tau*Du+u=exp(-t/tau)*sin(w*t)', 'u(0)=0')
>> simplify(dsolve('Du+u=exp(-t)*sin(t)', 'u(0)=0'))
ans =
-exp(-t)*(cos(t)-1)
```

Опять-таки, решение удалось получить в аналитическом виде. Несложно построить и графическую зависимость решения (рис. 5):

```
>> t=0:1:6; plot(t,exp(-t).*sin(3*t),t,-1/3.*exp(-t).*(cos(3.*t)-1))
```

Интеграл Дюамеля и свертка

Реакция линейной цепи на единичный скачок называется передаточной характеристикой цепи $a(t)$. Если она известна, то реакция цепи на входное воздействие $U_{\text{вх}}(t)$ определяется интегралом Дюамеля или сверткой:

$$U_{\text{вых}}(t) = U_{\text{вх}}(t) \times a(t) + \int_0^t \frac{dU_{\text{вх}}(x)}{dx} a(t-x) dx,$$

где x — вспомогательная переменная.

MATLAB имеет функции определения $a(t)$ для множества цепей, систем и устройств.

Для интегрирующей RC-цепи в MATLAB этот интеграл находится следующим образом (входное воздействие — перепад напряжения):

```
>> syms x R C E t tau
>> simplify(int(diff(1-exp(-x/(R*C)),x)*E,x,0,t))
ans =
-E*(exp(-t/R/C)-1)
```

При экспоненциальном перепаде:

```
>> int(diff(1-exp(-x/(R*C)),x)*E*(1-exp(-(t-x)/tau)),x,0,t)
ans =
-E*(exp(-t/(C*R)) - 1) - (E*tau*(exp(t/tau) - exp(-t/(C*R))))/(tau + C*R)
```

Дифференциальное уравнение с заданными граничными условиями

Часто решение осложняется особыми условиями: например, требуется соблюсти оба граничных условия — в начале интервала решения и в конце. Для этого приходится применять специальные численные методы решения дифференциальных уравнений. Рассмотрим функцию *twoode* для решения двух дифференциальных уравнений первого порядка, которая есть в MATLAB:

```
function dydx = twoode(x,y)
dydx = [ y(2); -abs(y(1)) ];
```

Решение дается следующей программой:

```
solinit = bvpinit([0 1 2 3 4],[1; 0]);
sol = bvp4c(@twoode,@twobc,solinit);
xint = linspace(0,4,50); yint = deval(sol,xint);
plot(xint,yint(1,:),'b'); xlabel('x')
ylabel('solution y'); hold on
```

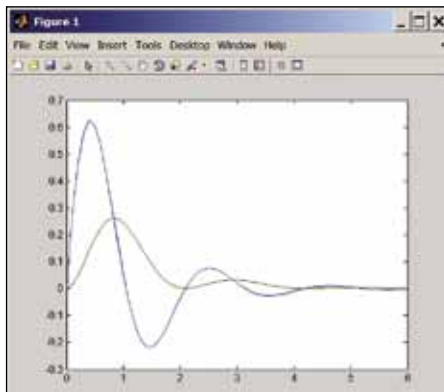


Рис. 5. Реакция интегрирующей RC-цепи на синусоидальный сигнал с экспоненциально спадающей амплитудой

При конечных условиях $y(x) = -1$, $y'(x) = 0$ другое решение задается программой:

```
solinit = bvpinit([0 1 2 3 4],[-1; 0]);
sol = bvp4c(@twoode,@twobc,solinit);
xint = linspace(0,4,50); yint = deval(sol,xint);
plot(xint,yint(1,:), 'r'); hold off
```

Итак, второе решение имеет те же граничные условия в начале и в конце, что и первое (рис. 6). Это обеспечивает специальный решатель *bvp4c*.

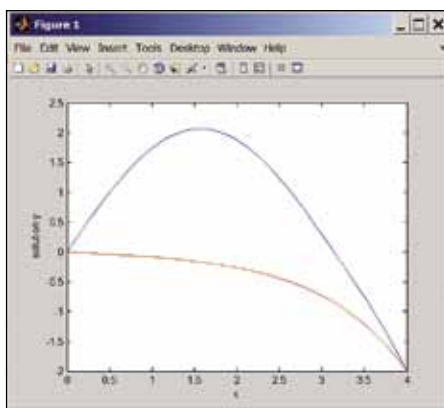


Рис. 6. Два решения дифференциальных уравнений с одинаковыми граничными условиями в начале и в конце

Нелинейное дифференциальное уравнение Ван дер Поля второго порядка

Классическим примером анализа поведения колебательных систем второго порядка является дифференциальное уравнение Ван дер Поля второго порядка. Его можно записать в виде системы из двух уравнений первого порядка:

$$dy_1/dt = y_2,$$

$$dy_2/dt = \mu(1-y_1^2)y_2 - y_1,$$

где y_1 — искомая функция; y_2 — ее производная.

Эта система описывает колебания в LC-контуре, подключенном к отрицательной проводимости или к активной части генератора (усилителя с положительной обратной связью). Степень нелинейности задается параметром μ . В MATLAB задана функция *vanderpoldemo* для решения системы Ван дер Поля:

```
dydt = [y(2); Mu*(1-y(1)^2)*y(2)-y(1)];
```

Вы можете убедиться в ее наличии, применив команду:

```
>> type vanderpoldemo
```

Следующая программа решает уравнение Ван дер Поля для параметра нелинейности $\mu = 0,25$ и начальных условий $y_0 = 0,25$ и 0 :

```
tspan = [0, 50]; y0 = [0.25; 0]; Mu = 0.25;
ode = @(t,y) vanderpoldemo(t,y,Mu);
[t,y] = ode45(ode,tspan,y0);
figure(1): plot(y(:,1),y(:,2)); title('vdp фазовый портрет')
figure(2): plot(t,y(:,1),t,y(:,2));
title('vdp временные зависимости')
```

Тут уместно отметить, что нелинейные дифференциальные уравнения не имеют в общем случае аналитического решения, и последнее осуществляется одним из численных методов — в данном случае это метод Дорманда-Принца 4-го и 5-го порядка, реализованный в MATLAB решателем *ode45*. Он возвращает вектор решения y из трех столбцов — значений времени t (нулевой столбец) и значений y_1 и y_2 (первый и второй столбцы).

При параметрической форме вывода получается фазовый портрет колебаний (рис. 7, сверху). Видно, что вначале колебания нарастают, а затем рост амплитуды замедляется и формируется предельный цикл фазового портрета, похожий на несколько искаженный эллипс. Графики временной зависимости также показаны на рис. 7 (снизу).

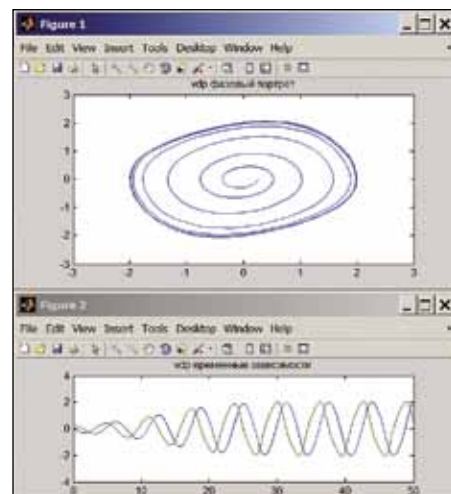
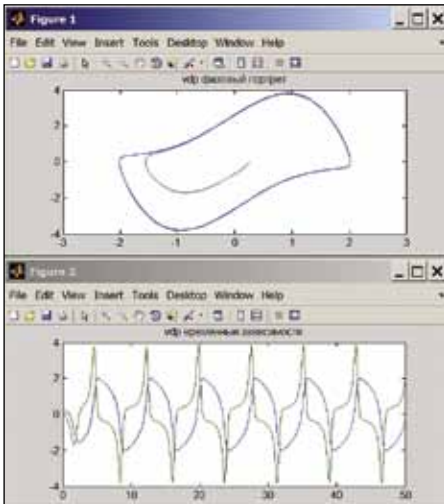


Рис. 7. Решение уравнения Ван дер Поля при $\mu = 0,25$

Рис. 8. Оценка линейности параметра Mu

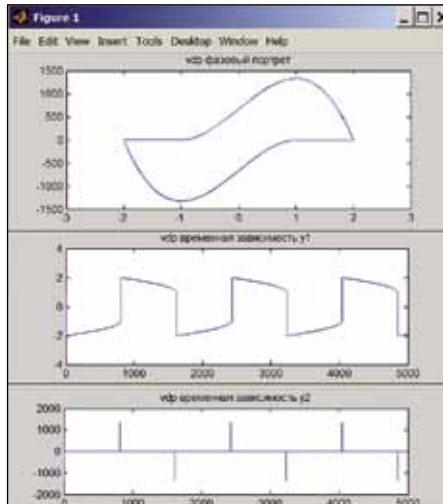
Оценим роль параметра нелинейности Mu . Для этого заменим Mu на значение 2 и снова запустим эту программу. Полученный результат представлен на рис. 8. Видно, что теперь предельный цикл далек от эллиптического, а форма колебаний содержит участки с резким изменением крутизны.

При больших Mu уравнение Ван дер Поля становится жестким, и графики временной зависимости колебаний приобретают очень быстрые перепады (почти разрывы) рядом с пологими участками. Для его решения нужно применять специальные решатели, например *ode15s*, *ode23s*, *ode23t* и *ode23tb*.

Пример программы для *ode23s*:

```
tspan = [0, 5000]; y0 = [1; 0]; Mu = 1000;
ode = @(t,y) vanderpoldemo(t,y,Mu);
[t,y] = ode23s(ode, tspan, y0);
figure(1); plot(y(:,1), y(:,2)); title('vdr фазовый портрет')
figure(2); plot(t, y(:,1)); title('vdr временная зависимость y1')
figure(3); plot(t, y(:,2)); title('vdr временная зависимость y2')
```

На рис. 9 показаны результаты расчета по этой программе, подтверждающие эти закономерности. Они характерны для релаксационных генераторов.

Рис. 9. Решение жесткого дифференциального уравнения Ван дер Поля при $Mu = 1000$

Заметим, что программа наглядна как в математическом, так и в физическом отношении и выгодно отличается от программ на универсальных языках программирования. Она имеет резко упрощенные конструкции ввода и вывода, не требует применения многочисленных циклов, использует проблемно-ориентированные функции и т. д. И это справедливо для всех MATLAB-программ.

Модель экологической среды Лотки-Вольтерры

К решению системы из двух дифференциальных уравнений первого порядка сводится и модель изменения популяций хищников и их жертв в экологической среде:

$$\begin{aligned} \dot{y}_1 &= (1 - \alpha y_2) y_1 \\ \dot{y}_2 &= (-1 + \beta y_1) y_2 \end{aligned}$$

Эта, одна из классических моделей — лотка — есть в MATLAB и используется в следующей программе (рис. 10):

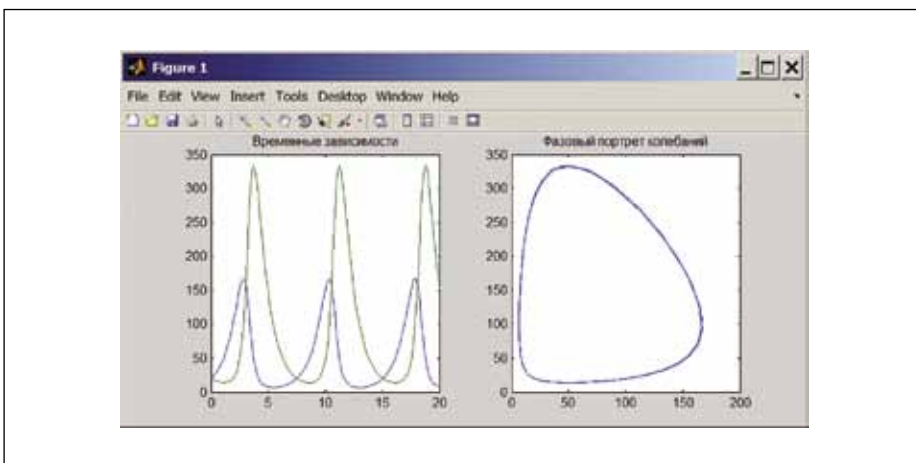


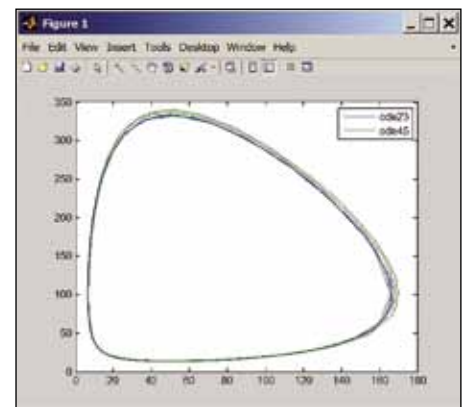
Рис. 10. Результаты моделирования модели «хищники — жертвы» Лотки-Вольтерры

```
t0 = 0; tfinal = 20; y0 = [20 20];
[t,y] = ode23('lotka',[t0 tfinal],y0);
subplot(1,2,1); plot(t,y); title('Временные зависимости')
subplot(1,2,2); plot(y(:,1),y(:,2)); title('Фазовый портрет колебаний')
```

Ранее для численного интегрирования подобных уравнений использовался метод Рунге-Кутты 4-го порядка. Но он дает медленный выход на предельный цикл фазового портрета. Новые решатели *ode23* и *ode45* имеют меньшее время выхода на предельный цикл, а метод *ode23* приспособлен и для решения жестких систем дифференциальных уравнений.

Следующая программа обеспечивает сравнение этих двух решателей (рис. 11):

```
[T,Y] = ode45('lotka',[t0 tfinal],y0);
subplot(1,1,1); title('Фазовый портрет колебаний')
plot(y(:,1),Y(:,2),'-';Y(:,1),Y(:,2),'-');
legend('ode23','ode45')
```

Рис. 11. Сравнение двух методов решения модели Лотки-Вольтерры *ode23* и *ode45*

Хаос в колебаниях — моделирование «странного» аттрактора Лоренца

Изучая конвективное движение подогретой жидкости, Э. Н. Лоренц сумел свести эту задачу к решению системы из трех нелинейных дифференциальных уравнений первого порядка:

$$\begin{aligned} \frac{dx}{dt} &= -\sigma x + \sigma y, \\ \frac{dy}{dt} &= rx - y - xz, \\ \frac{dz}{dt} &= xy - bz. \end{aligned}$$

Оказалось, что движение жидкости носит колебательный характер в двух взаимно перпендикулярных плоскостях расположения фазовых траекторий колебаний, а их взаимодействие и переброс при определенных параметрах (например, $\sigma = 10$, $b = 2,7$ и $r = 27$) из одной плоскости в другую приводят к хаотическим. Подобные системы получили название «странных» аттракторов. Они встречаются в механических, электронных и других системах и обычно ответственны за странности в работе таких детерминированных систем. В MATLAB есть пример ви-

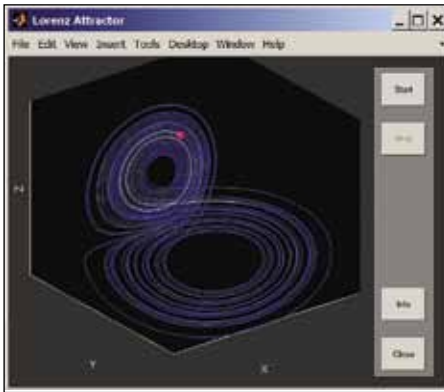


Рис. 12. Окно анимации фазового портрета «странного» аттрактора Лоренца

зуализации аттрактора Лоренца со своим окном GUI (рис. 12).

Этот пример слишком обширный, чтобы разместить его в статье. Поэтому на рис. 13 приведено окно справки по нему и начало листинга его программы. Оно демонстрирует полную открытость кода программ в MATLAB, включая коды демонстрационных примеров, которые можно изучать, модернизировать и применять в своих примерах.

Решение дифференциальных уравнений с частными производными

Широкий круг задач (например, деформация и колебания струн, мембран и поверхностей) сводится к решению систем дифференциальных уравнений с частными производными. В MATLAB 8.0 есть пара функций:

```
function [c,f,s] = pdex1pde(x,t,u,DuDx)
function u0 = pdex1ic(x)
```

Они иллюстрируют решение системы дифференциальных уравнений с частными производными. Следующая программа решает эти уравнения (рис. 14) при:

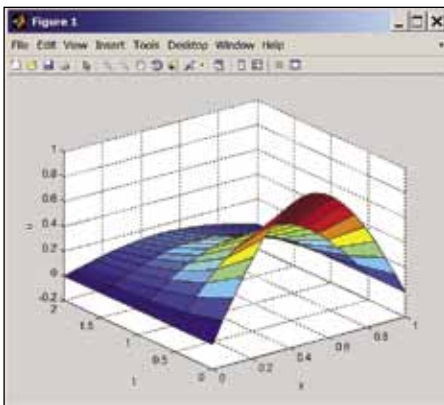


Рис. 14. Пример решения системы дифференциальных уравнений с частными производными

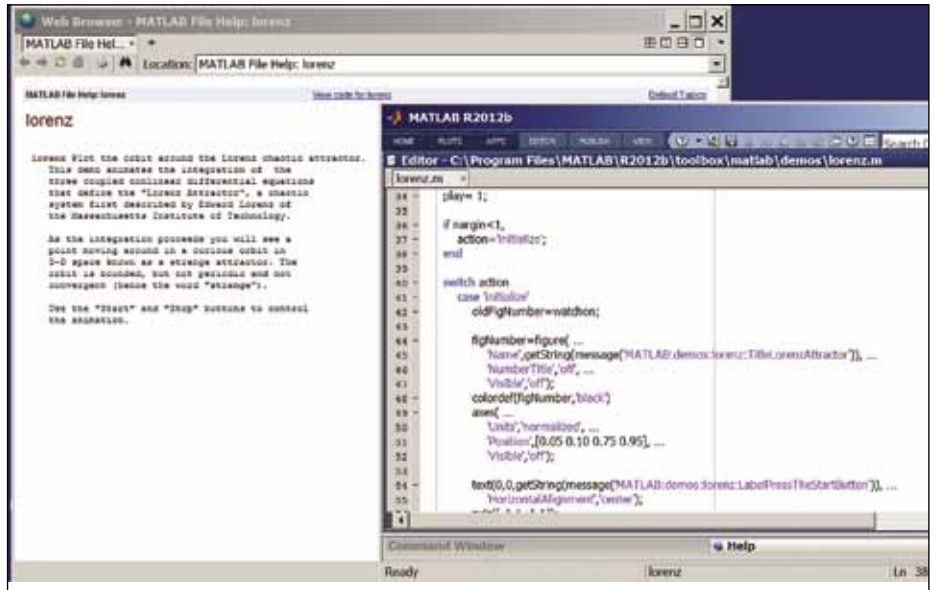


Рис. 13. Окно справки по аттрактору Лоренца и листинг программы его моделирования в окне редактора программ MATLAB

```
c = pi^2; f = DuDx; s = 0;
x = linspace(0,1,20); t = [0 0.5 1 1.5 2];
sol = pdexpe(0,@pdex1pde,@pdex1ic,@pdex1bc,x,t);
u1 = sol(:,:,1); surf(x,t,u1); xlabel('x');ylabel('t');zlabel('u');
hold on;
u1 = sol(:,:,1); surf(x,t,u1);
xlabel('x'); ylabel('t'); zlabel('u');
```

Моделирование в частотной области

Многие задачи требуют моделирования в частотной области — в частности, задачи анализа цепей на переменном токе. Обычно при этом используется аппарат комплексных чисел и операторный метод расчета. MATLAB имеет такой аппарат и позволяет полноценно проводить моделирование на основе всех известных методов анализа таких цепей — символического, операторного и т. д. Многие его функции сосредоточены в пакете расширения Signal Processing Toolbox с доступом через каталог Home.

Большинство функций MATLAB работает с комплексными числами. Например, функция *solve* находит частоту нулевого сопротивления последовательного LC-контура и выводит формулу резонансной частоты:

```
>> solve('i*2*pi*f*L+1/(i*2*pi*f*C)=0','f')
ans =
1/2/pi/(L*C)^(1/2)
-1/2/pi/(L*C)^(1/2)
```

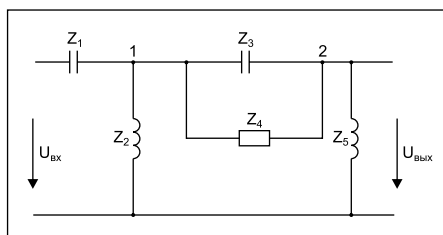


Рис. 15. Цепь переменного тока

Рассмотрим более сложный пример — расчет электрической цепи переменного тока (рис. 15). Представим программу для аналитического расчета этой цепи (с комментариями после знаков %):

```
syms s real %Задание символической переменной s
Z=[10/s 10*s 10/s 10 10*s]; % Обобщенные сопротивления
Y=1./Z % Вычисление массива проводимостей
YY=[Y(1)+Y(2)+Y(3)+Y(4) -(Y(3)+Y(4)) % Матрица узл. проводимостей
-(Y(3)+Y(4)) (Y(3)+Y(4)+Y(5))];
iu=[Y(1) 0]; % Задание матрицы узловых токов при Uвх=1
KK=YY\iu' % Решение системы узловых уравнений
[n d]=numden(KK(2)); % Нахождение передаточной функции
b=sym2poly(n); % Нахождение числителя передаточной функции
a=sym2poly(d); % Нахождение знаменателя передаточной функции
freqs(b,a) % Вычисление ЛАЧХ и ЛФЧХ и построение их графиков
```

Функция *freqs* вычисляет амплитудно-частотную и фазо-частотную характеристики (рис. 16) линейной системы по числителю и знаменателю передаточной характеристики четырехполюсника в операторной форме. АЧХ строится в логарифмическом масштабе, то есть ЛФЧХ при изменении

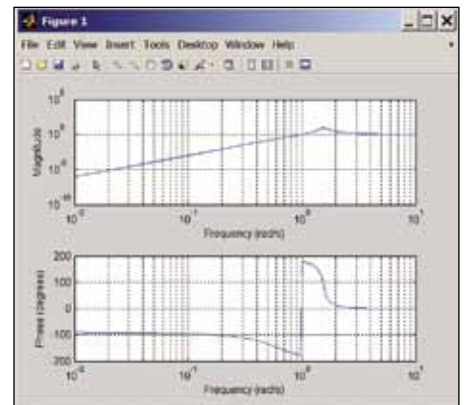


Рис. 16. ЛАЧХ и ЛФЧХ четырехполюсника, показанного на рис. 15

угловой частоты от $0,1\omega_0$ до $10\omega_0$, где ω_0 — базовая частота.

Непрерывное изменение фазы обеспечивается только в диапазоне углов 360° . Если угол достигает значений -180° или $+180^\circ$, то на графике формируется характерный разрыв фазы. Замена последней строки в программе на следующие строки:

```
[h,w]=freqs(b,a);
subplot(211); loglog(w,abs(h)); grid on
subplot(212); semilogx(w,360+180*unwrap(angle(h))/pi); grid on
```

позволяет убрать этот разрыв и построить непрерывный график (рис. 17). Он выглядит более естественным.

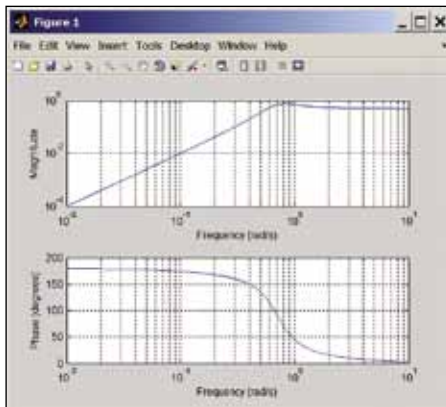


Рис. 17. Другой вариант графиков ЛАЧХ и ЛФЧХ четырехполюсника, показанного на рис. 15

Пакет блочного имитационного моделирования Simulink 8.0

Чтобы начать работу с пакетом Simulink 8.0, достаточно в каталоге Home активизировать кнопку Library (библиотека Simulink). Появится окно обозревателя (браузера) библиотеки (рис. 18, слева). Оно имеет две панели: слева панель дерева разделов и луков библиотеки, справа — панель обзора текущего раздела. Этот раздел можно открыть и наблюдать все входящие в него блоки.

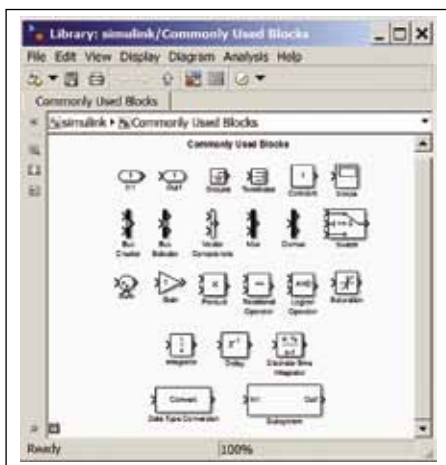


Рис. 19. Блоки общего применения



Рис. 18. Окна библиотеки блоков пакета Simulink 8.0 и редактора диаграмм

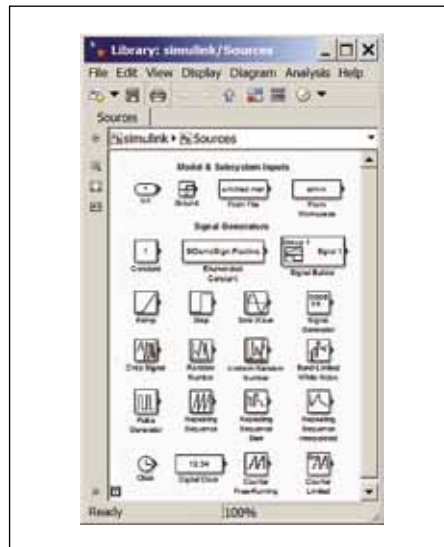


Рис. 20. Блоки источников сигналов Sources

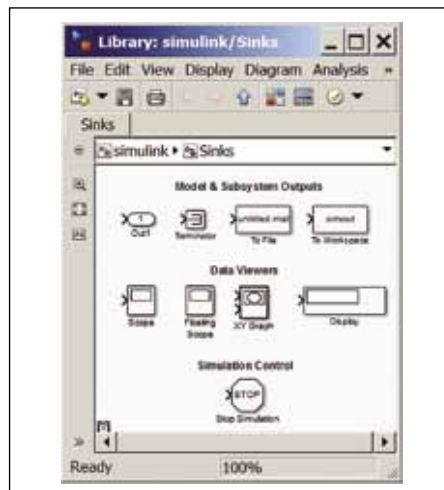


Рис. 21. Блоки приемников/регистраторов сигналов Sinks

Библиотека содержит три десятка разделов, в каждом по несколько блоков и подразделов (до нескольких десятков). Каждый блок имеет наглядное изображение (пиктограмму) и имя. Блоки имеют входы и выходы, тип которых задается их видом. На рис. 19 показано окно с блоками общего применения.

Почти каждая диаграмма моделируемого устройства или системы имеет источник (источники) сигналов и устройства, принимающие входные сигналы и регистрирующие их. На рис. 20 показано окно с блоками источников сигналов: перепада, линейно нарастающего сигнала, синусоидального сигнала, прямоугольного сигнала и т. д. Есть и блоки для создания более сложных систем генерации сигналов, в том числе подсистем.

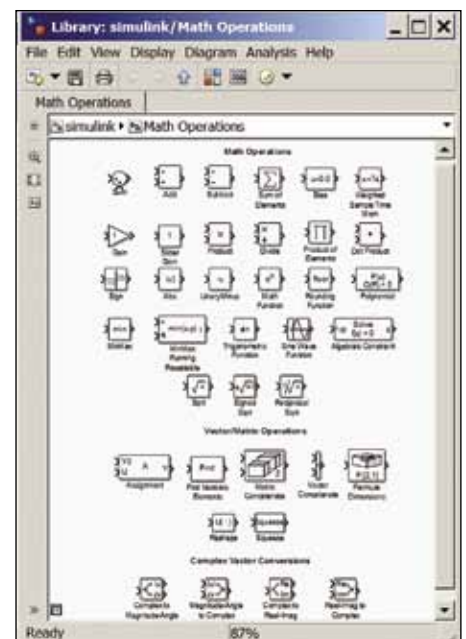


Рис. 22. Блоки математических операций

На рис. 21 показано окно с блоками приемников сигналов. Это блок виртуального осциллографа с проводным и беспроводным подключением, графопостроитель XY Graph, цифровой дисплей и др. Набор блоков ограничен, но функционально достаточен. Есть блоки приема сигналов в файл и в рабочем пространстве MATLAB, а также блок остановки моделирования.

Многие математические модели должны содержать блоки математических операций. Окно с ними представлено на рис. 22. Здесь показаны блоки арифметических операций, дифференцирования и интегрирования сигналов, работы с векторами и матрицами, решения уравнений, оптимизации и т. д. Практически все, что может выполнять система MATLAB в командном режиме, реализуется математическими блоками Simulink.

Подготовка диаграммы модели, ее модернизация и запуск

Модели систем и устройств в Simulink представлены диаграммами в редакторе диаграмм, составленными соответственно соединенными друг с другом блоками. Например, перенесем мышью блоки синусоидального сигнала, ограничителя амплитуды и визуального осциллографа в окно редактора диаграмм и соединим их (также применяя мышью) друг с другом. Получим простейшую диаграмму модели (рис. 23).

Перед пуском диаграммы следует убедиться в правильности задания параметров основных блоков. Двойной щелчок на пиктограмме блока выводит список его параметров. Для примера на рис. 24 приведен блок параметров ограничителя амплитуды, в котором пороги в 0,5 В изменены на пороги в 0,45 В. Как правило, вполне приемлемые параметры блоков заданы по умолчанию, но не всегда.

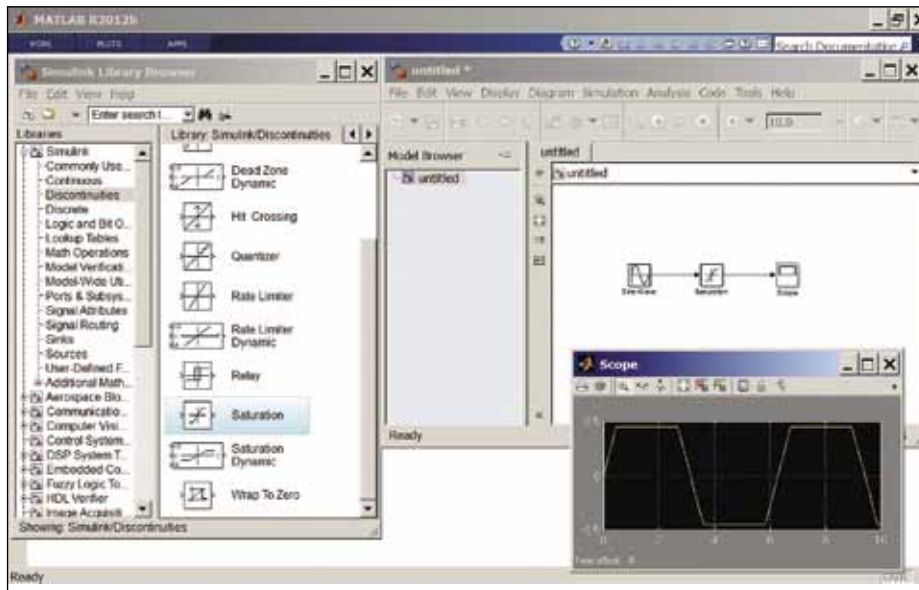


Рис. 23. Подготовка и пуск простой диаграммы из трех блоков

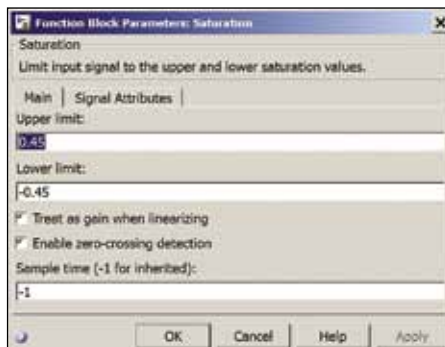


Рис. 24. Окно установки параметров блока ограничителя

Пуск диаграммы выполняется активизацией кнопки с треугольником в кружке, расположенной в инструментальной строке редактора

диаграмм. После выполнения программы может понадобиться открыть окно виртуального осциллографа, указав на пиктограмму осциллографа и дважды щелкнув по ней левой клавишей мыши. Осциллограмма работы диаграммы показана на рис. 23.

Диаграммы модели можно легко редактировать и модернизировать удалением и добавлением блоков, их перемещением, изменением размеров и ориентации, вырезкой, копированием и т. д. Например, на рис. 25 в диаграмму добавлен новый блок дифференцирования сигнала. Запуск диаграммы позволяет наблюдать новую осциллограмму продифференцированного сигнала.

Блочное моделирование в Simulink 8.0 системы Ван дер Поля

А теперь рассмотрим диаграмму моделирования колебательной системы Ван дер Поля, но не программными средствами MATLAB, а с помощью блоков Simulink 8.0. Диаграмма модели и результат моделирования представлены на рис. 26. Она более наглядна, чем программная MATLAB-модель, и не требует от пользователя знания основ программирования.

Для примера на рис. 27 показано окно установки параметров моделирования. Важнейшими из них являются начальное и конечное время моделирования, тип моделирования (с автоматическим и постоянным шагом), начальный шаг изменения времени, выбор типа решателя и др.

При выборе решателя дифференциальных уравнений следует использовать их список, показанный на рис. 28 в открытом состоянии. Название решателя указывает на его порядок и жесткость (решатели жестких уравнений отмечены буквой «S»).

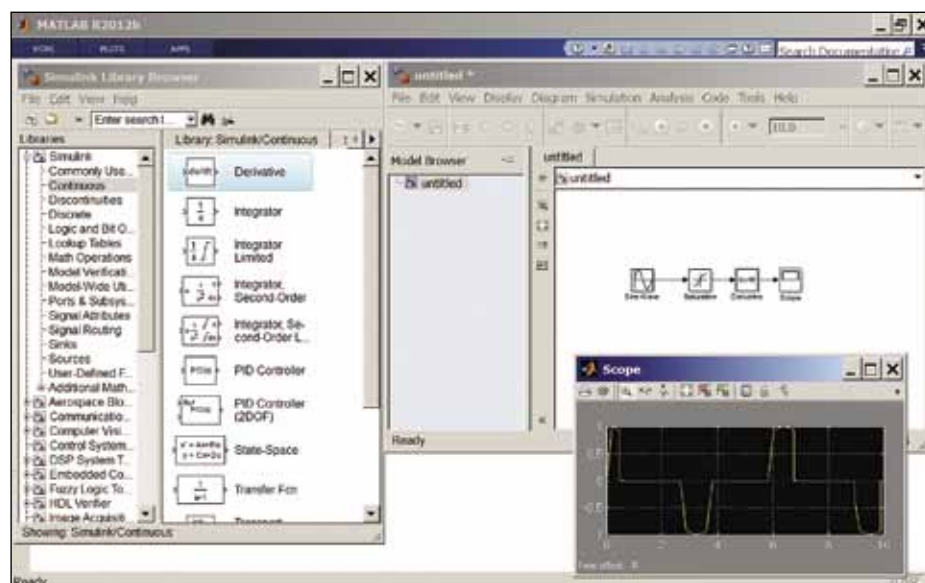


Рис. 25. Добавление в диаграмму модели нового блока и запуск новой диаграммы

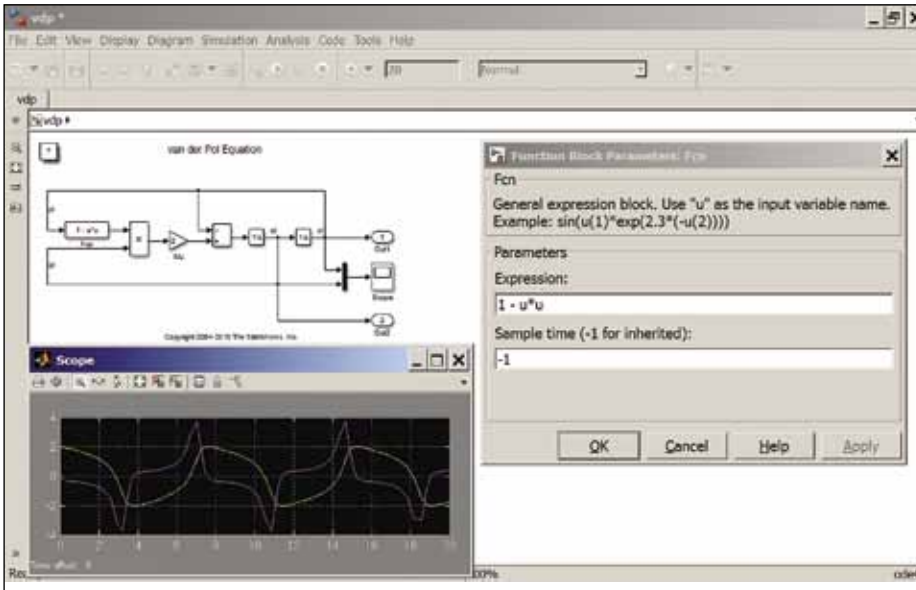


Рис. 26. Диаграмма системы Ван дер Поля в Simulink 8.0

димость применения преобразователей сигналов и вычислительного блока $F(x) = 0$, подключаемого к диаграмме, чаще всего к ее общему проводу. В целом диаграмма модели устройства напоминает обычную принципиальную схему. Более того, в Simulink 8.0 можно использовать PSPICE-модели полупроводниковых приборов с учетом температурных изменений их параметров (разумеется, там, где это нужно).

На рис. 29 показана одна из типовых схем высокочастотного LC-генератора на биполярном транзисторе с положительной обратной связью. Это уже не расчетная модель, а диаграмма модели вполне реального устройства.

Обратите внимание на сложный и длительный переходный процесс установления колебаний в схеме генератора. Он связан с применением высокочастотного резонансного контура. Он же обеспечивает близкую к синусоидальной форму генерируемого сигнала и его высокую стабильность по частоте.

При желании можно наблюдать более ограниченный участок этого процесса. Для этого при нажатой левой клавише мыши нужно выделить желаемый участок на осциллограмме виртуального осциллографа, а затем отпустить клавишу: в результате на весь экран будет развернута осциллограмма выделенного участка колебательного процесса. В пределах малого участка она близка к синусоидальной, хотя некоторые искажения можно заметить.

Моделирование мультивибратора на биполярных транзисторах

К широко распространенным типам генераторов почти прямоугольных импульсов относятся мультивибраторы, обычно выполняемые на биполярных транзисторах малой

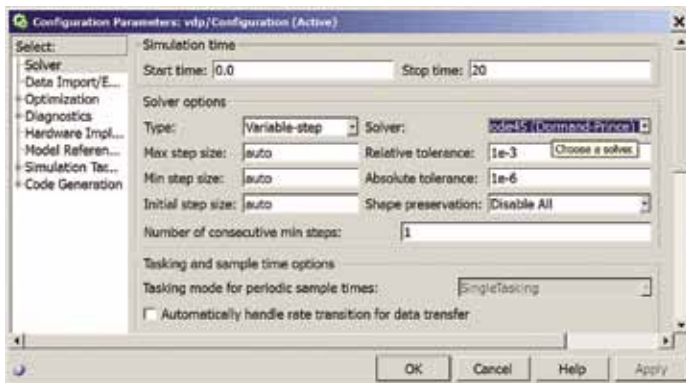


Рис. 27. Установка параметров имитационного моделирования

ode45 (Dormand-Prince)
discrete (no continuous sta
ode45 (Dormand-Prince)
ode23 (Bogacki-Shampine)
ode113 (Adams)
ode15s (stiff/NDF)
ode23s (stiff/Mod. Rosenbr
ode23t (mod. stiff/Trapezoi
ode23tb (stiff/TR-BDF2)

Рис. 28. Список решателей дифференциальных уравнений

Моделирование LC-автогенератора на биполярном транзисторе

Далее мы рассмотрим моделирование нескольких типовых радиоэлектронных устройств на физическом уровне с помощью блоков пакета расширения SimElectronic пакета Simulink 8.0. С этим связана необхо-

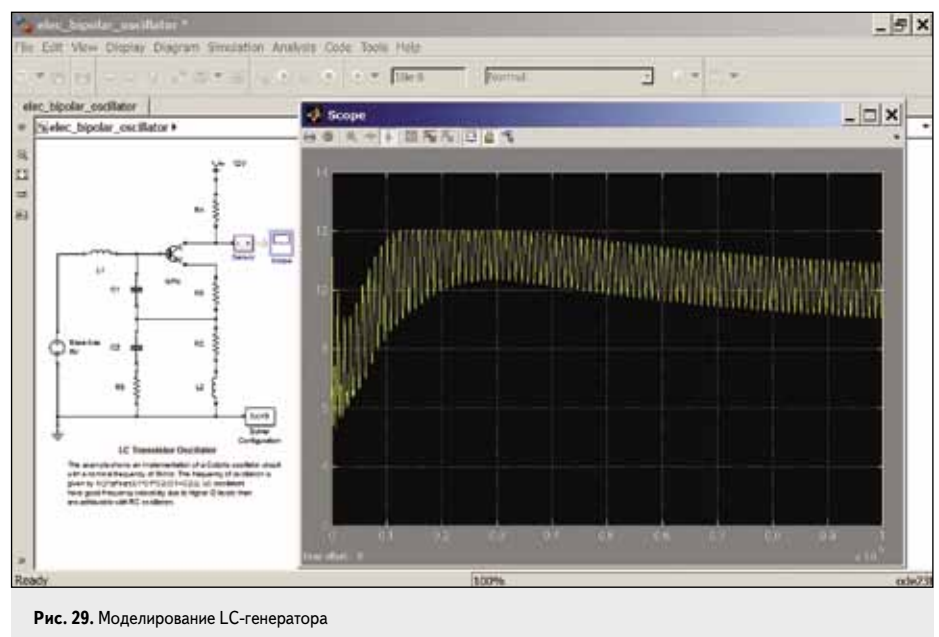


Рис. 29. Моделирование LC-генератора

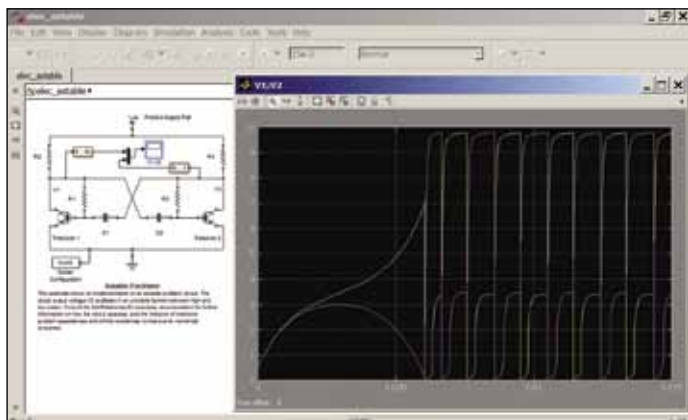


Рис. 30. Моделирование мультивибратора

и средней мощности. На рис. 30 показана диаграмма модели одной из схем таких мультивибраторов. Представленные там же осциллограммы импульсов мультивибратора наглядно указывают на сложность переходных процессов в этом устройстве и длительный характер установления стационарного режима его работы.

Моделирование генератора треугольных импульсов на интегральных операционных усилителях

На рис. 31 показана диаграмма модели амплитудно-модулированного генератора треугольных импульсов на основе двух интегральных операционных усилителей с ограниченной полосой пропускания. Один играет роль емкостного интегратора, а другой — управляемого ограничителя амплитуды. Эта диаграмма — типичный пример моделирования релаксационного генератора.

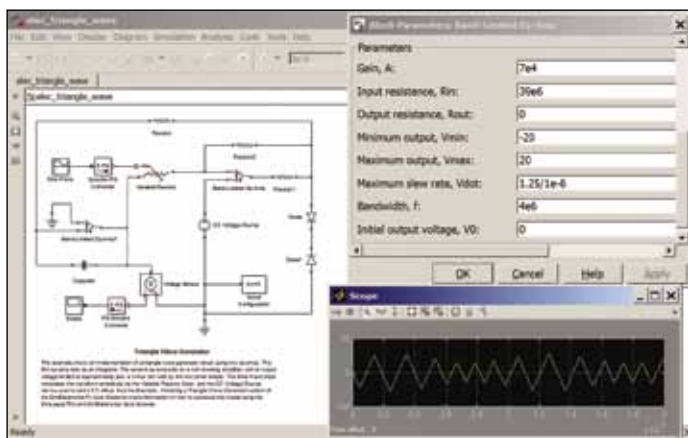


Рис. 31. Моделирование генератора треугольных импульсов на двух интегральных операционных усилителях

Моделирование инверторов на мощных полевых транзисторах

Simulink позволяет выполнять наглядное и точное блочное имитационное моделирование сложных систем и устройств. Для этого предназначен обширный набор типовых и специальных блоков, есть также возможность создавать субблоки пользователя и новые блоки.

На рис. 32 показана диаграмма модели обратного однофазного инвертора (преобразователя) постоянного напряжения одного уровня в постоянное напряжение другого уровня с импульсным преобразователем на мощном полевом транзисторе, работающим с высоким КПД и управляемым от блока широтно-импульсного модулятора. Использован и блок контроля температуры транзистора при его са-

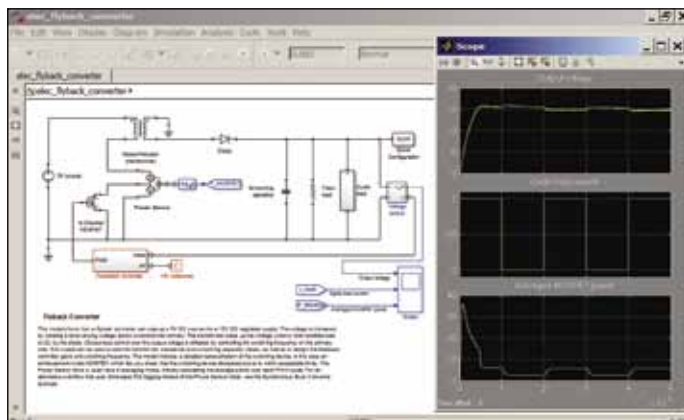


Рис. 32. Моделирование обратного однофазного инвертора на мощном полевом транзисторе

морозогреве (кстати, введенный в последние версии MATLAB, как и совершенные PSPICE-модели приборов).

Диаграмма более мощного и совершенного инвертора полумостового типа представлена на рис. 33. Транзисторы инвертора также управляются от блока широтно-импульсного модулятора.

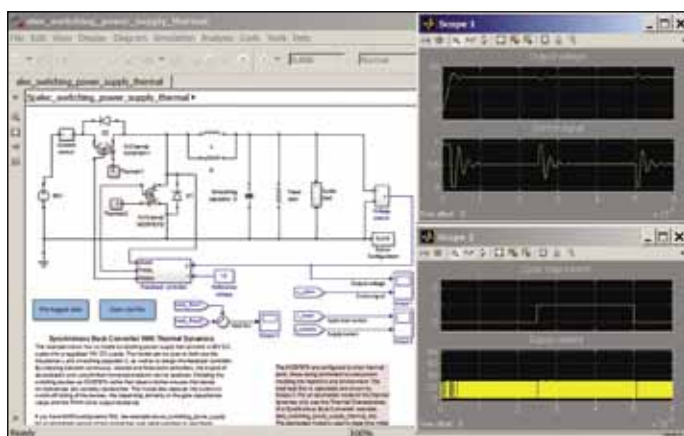


Рис. 33. Моделирование полумостового инвертора на мощных полевых транзисторах

Помимо осциллограмм, иллюстрирующих его работу, предусмотрен вывод подробных графиков в форме временных диаграмм напряжений, токов, составляющих рассеиваемой мощности и даже температуры кристалла и корпуса мощных полевых транзисторов. Лист с результатами расчета одного из устройств представлен на рис. 34.

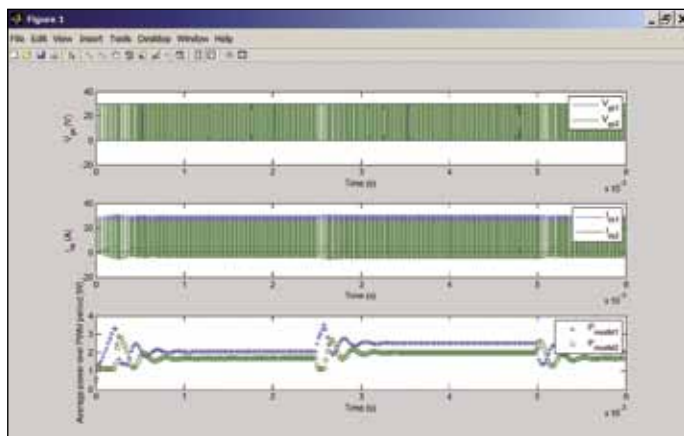


Рис. 34. Лист с результатами расчета устройства, изображенного на рис. 33

Моделирование линейных цепей и фильтров на переменном токе

Simulink позволяет легко провести анализ линейных цепей на переменном токе. Пример этого дан на рис. 35. Результатом является АЧХ и ФЧХ цепи на дискретных компонентах. В окне диаграммы модели показано также окно установки параметров блока преобразования частот. Рекомендуется просмотреть параметры компонентов всей диаграммы.

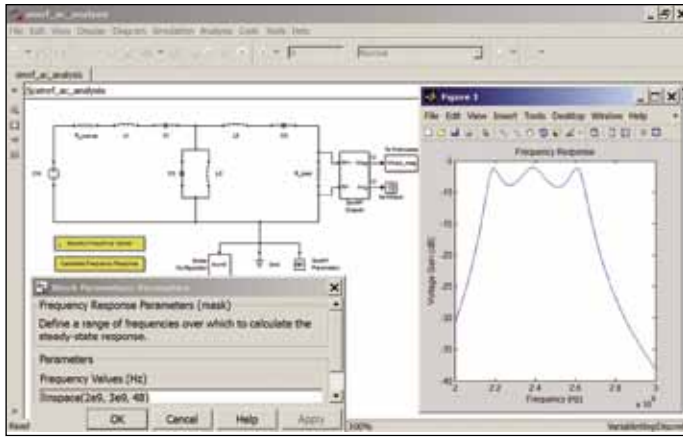


Рис. 35. Моделирование линейной цепи на дискретных компонентах

Огромным достоинством Simulink является возможность моделирования СВЧ линейных устройств с полосой частот выше 1 ГГц. При этом учитываются все весьма специфические особенности электромагнитного взаимодействия и работы устройств СВЧ. Для этого служит пакет моделирования радиочастот SimRF (есть и RF Toolbox), который интенсивно развивается в последних реализациях MATLAB+Simulink.

На рис. 36 приведен пример моделирования реального полосового СВЧ-фильтра со средней частотой 1 ГГц и полосой частот 0,2 ГГц на уровне спада ФЧХ на 3 дБ. АЧХ такого фильтра снимается при подаче на его вход шумового сигнала с равномерным вплоть до СВЧ-частот спектром. Смоделированные при этом АЧХ, полученные виртуальными анализатором спектра и графопостроителем, показаны на рис. 36. Они очень похожи на характеристики реальных приборов.

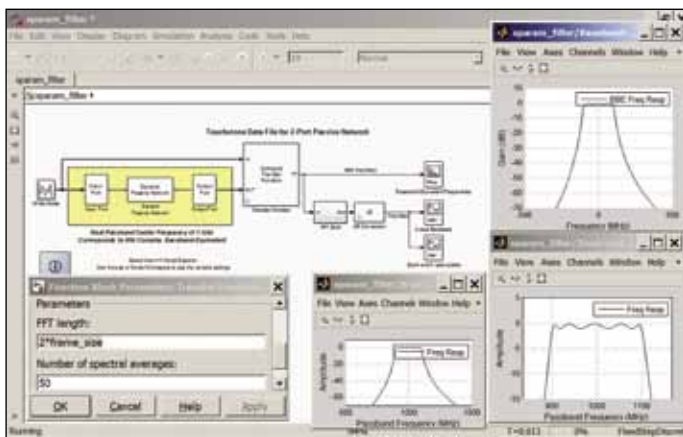


Рис. 36. Моделирование полосового фильтра со средней полосой 1 ГГц

Фильтры на более высокие частоты часто строятся на основе отрезков линий передачи СВЧ-диапазона частот. На рис. 37 проведено сравнение двух вариантов таких фильтров со средней частотой 4 ГГц.

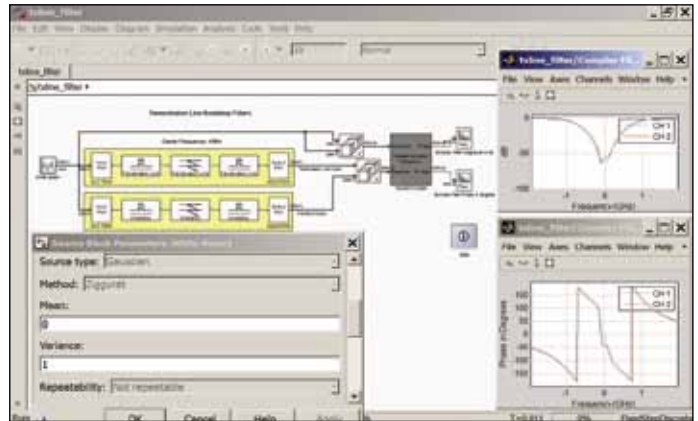


Рис. 37. Моделирование двух вариантов СВЧ-фильтров на основе отрезков линий передачи со средней частотой 4 ГГц

Моделирование нелинейных СВЧ-усилителей

RF Blockset позволяет также моделировать реальные СВЧ-усилители, имеющие заметную нелинейность. Часто нелинейность оценивается двухтоновым сигналом на входе, порождающим интермодуляционные искажения спектра выходного сигнала (рис. 38). Он строится анализатором спектра высокого разрешения, представляющим частотные составляющие спектра как отрезки вертикальных прямых.

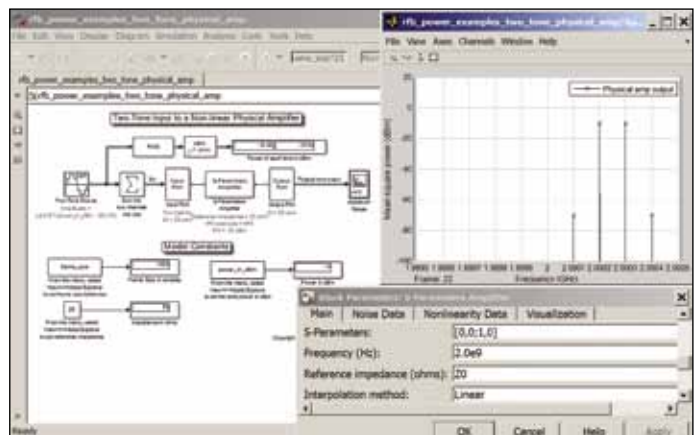


Рис. 38. Моделирование усиления двухтонового сигнала физического СВЧ-усилителя и построение спектра его выходного сигнала

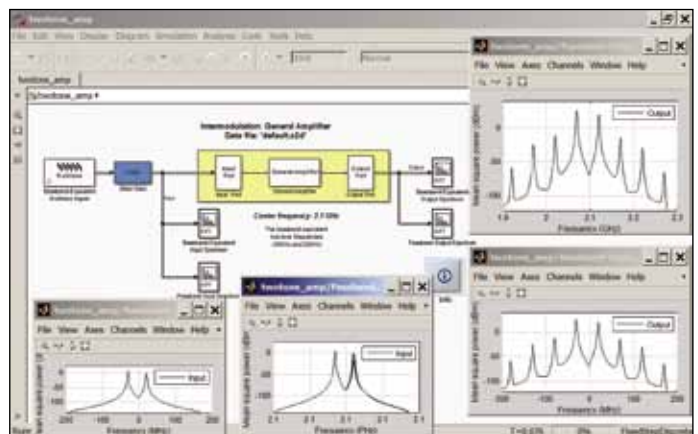


Рис. 39. Моделирование интермодуляционных искажений в СВЧ-усилителе с центральной частотой 2,1 ГГц

Возникновение интермодуляционных искажений демонстрирует диаграмма модели реального усилителя с центральной частотой 2,1 ГГц и частотами двухтонового сигнала 30 и 20 МГц (рис. 39). Искажения отчетливо видны и легко оцениваются количественно.

Моделирование системы управления двигателем

Примером моделирования электромеханических и тепловых систем является система термисторного контроля двигателя постоянного тока (рис. 40). Вообще следует отметить, что системы управления разнообразными электрическими машинами — любимая тема разработчиков MATLAB. Вы найдете в MATLAB более десятка примеров на эту тему.

Заключение

Разумеется, большинство приведенных примеров относится к последним версиям систем MATLAB+Simulink, а не только к самой последней реализации 8.0 (R2012b). Отличия последней проявляются в ее интерфейсе, сделавшем многие функции MATLAB и Simulink более наглядными и доступными. Обеспечена если не безукоризненная, то очень хорошая совместимость с предшествующими версиями системы MATLAB+Simulink. И конечно, в новейшей версии сохранены и приумножены все возможности матричных методов анализа и расчета, а также возможности визуально ориентированного блочного моделирования различных систем и устройств — от фундаментальных физических систем до реальных устройств, охватывающих ВЧ- и СВЧ-диапазоны. ■

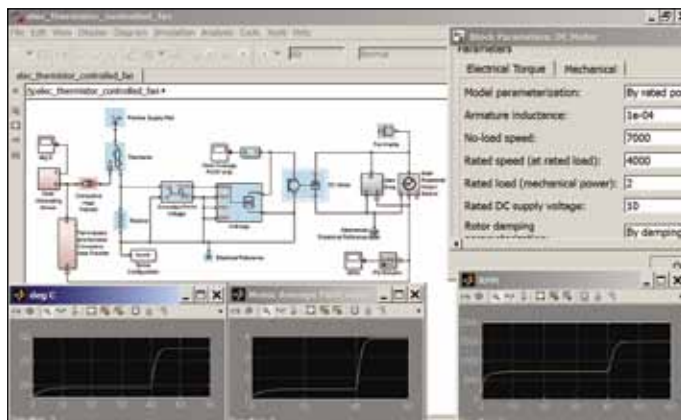


Рис. 40. Моделирование системы термисторного контроля двигателя постоянного тока

Литература

1. www.mathworks.com
2. Дьяконов В. П. MATLAB R2006/2007/2008 + Simulink 5/6/7. Основы применения. М.: СОЛОН-Пресс, 2008.
3. Дьяконов В. П. MATLAB. Полный самоучитель. М.: ДМК-Пресс, 2012.
4. Дьяконов В. П. MATLAB и Simulink для радионинженеров. М.: ДМК-Пресс, 2011.
5. Дьяконов В. П., Пеньков А. А. MATLAB и Simulink в электроэнергетике. М.: Горячая линия – Телеком, 2009.