Министерство РФ по связи и информатизации

Поволжская Государственная Академия
Телекоммуникаций и Информатики

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторным работам

"Программирование в системе MatLab"

Одобрено Методическим Советом ПГАТИ
__ апреля 2001 года

Автор-составитель: АКЧУРИН Э.А. к.т.н., доцент

Редактор: КОРАБЛИН М.А. д.т.н., профессор

Рецензент: ТЯЖЕВ А.И. д.т.н., профессор

Самара

2001

Введение

Лабораторный цикл содержит 7 работ по изучению программирования с использованием математической системы MatLab и входящей в него программы моделирования Simulink:

1. Основы MatLab.
2. Простые вычисления в MatLab.
3. Многомерные вычисления в MatLab.
4. Решение уравнений в MatLab.
5. Символьные вычисления в MatLab.
6. Моделирование устройства с помощью Simulink.
7. Моделирование системы с помощью Simulink.

Цикл может использоваться в дисциплинах: «Программирование на языках высокого уровня» и "Цифровые сигнальные процессоры" специальности 220400, «Высокоуровневые методы информатики и программирования» специальности 351400 и «Информатики» специальностей 200700, 200900, 201000, 20110, 201200.

Рекомендуемая литература:

1. Дьяконов В.П., Абраменкова И.В. MatLab 5.0/5.3. М.: Нолидж, 1999, 640 с.
2. Гультяев А.К. MatLab 5.2. Имитационное моделирование в среде Windows. СПб: Корона, 1999, 288 с.

Содержание отчета по каждой работе:

1. Название работы, задание в соответствии с вариантом.
2. Программа.
3. Результаты выполнения программы на ПК.
4. Выводы.

Содержание

1. Основы MatLab

Подготовка к работе

По указанной литературе изучить:

- основы системы MatLab,
- системное меню MatLab,
- основные системные команды,
- правила ввода команд и данных,
- ранжированные переменные,
- правила вывода результатов.

Контрольные вопросы

1. Структура окна системы MatLab.
2. Команды пункта "File" системного меню.
3. Команды пункта "Edit" системного меню.
4. Команды пункта "View" системного меню.
5. Команды пункта "Web" системного меню.
6. Команды пункта "Window" системного меню.
7. Команды пункта "Help" системного меню.
8. Правила ввода команд.
9. Правила ввода функций и операндов.
10. Правила ввода выражений.
11. Организация циклов.
12. Правила ввода комментариев.
13. Правила просмотра результатов операций.

Задание к работе

Задача 1. Изучить интерфейс MatLab.

Задача 2. Ознакомиться с демонстрационными примерами MatLab.

Задача 3 Выполнить в режиме калькулятора следующие действия:

- Ввод исходных операндов.
- Выполнить над операндами 1 и 2 операцию 1.
- Выполнить над результатом и операндом 1 операцию 2.
- Выполнить над результатом и операндом 2 операцию 3.
- Возвести почленно операнд 1 в степень 3.

Варианты заданий

№	Операнд 1	Операнд 2	Операторы		
			1	2	3
1	V=[12 34 61 45 11]	v = 34	*	./	+
2	V=[80 67 34 11 45]	v = 43	/	.*	-
3	V=[19 77 45 11 67]	v = -5	+	.\	/
4	V=[11 98 67 45 22]	v = 7	-	.*	/
5	V=[67 34 67 45 56]	v = -12	+	.\	*
6	V=[18 36 45 45 4]	v = 10	/	./	-
7	V=[55 43 8 45 23]	v = 44	/	.*	/
8	V=[32 28 55 45 34]	v = 87	*	-	/
9	V=[14 34 33 45 15]	v = 78	*	+	+
10	V=[15 23 17 45 9]	v = -22	/	-	*
11	V=[10 34 10 45 7]	v = -14	*	-	*
12	V=[95 56 5 45 54]	v = 99	+	./	+
13	V=[18 90 35 45 46]	v = 32	*	.*	-
14	V=[24 34 87 45 88]	v = -43	/	.*	/
15	V=[14 41 90 45 77]	v = 55	/	+	+

Методические указания

1. В MatLab все данные рассматриваются, как матрицы. Тип результата определяется автоматически по виду выражения.
2. В идентификаторах высота буквы имеет значение. Рекомендуется для имен простых переменных выбирать строчные буквы, а для структурированных (векторы и массивы) прописные.
3. Векторы вводятся в квадратных скобках, компоненты вектора разделяются пробелами. Например, V=[1 2 3].
4. Матрицы вводятся в квадратных скобках, внутри которых размещаются векторы строк, разделенные знаком точка с запятой (;).Например, V=[1 2 3 ; 4 5 6; 7 8 9].
5. Если данные не умещаются в строке, строку можно отобразить в нескольких строках, используя разделитель в виде многоточия (не менее трех точек).
6. Значение π задается системной константой с именем pi.
7. В MatLab возможны два режима работы:
   - В командном окне, как с калькулятором. В этом случае каждое действие сразу же исполняется.
   - В редакторе программ. В этом случае программа вводится, как обычно, а исполняется по команде встроенного отладчика.
8. При работе в режиме калькулятора выражения могут вводиться:
   - В прямой форме, тогда после завершения ввода ответ будет выведен под встроенным системным именем ans. Переменная с этим именем всегда хранит результат последнего вычисления.
   - В форме оператора присвоения, когда переменной с выбранным именем присваивается значение выражения. Ответ в этом случае выводиться под именем этой переменной.

- Любое уже определенное значение можно вызвать из рабочей области по имени переменной.

9. Если вычисляется значение переменной с выбранным именем по заданному выражению, результат выводится под именем этой переменной в следующей строке. Векторы выводятся в строке с пробелами, матрицы - построчно, каждая содержит вектор строки.
10. При работе с программой неграфические результаты выводятся в окно командной строки. При необходимости их можно выводить, как текст, в специально создаваемое окно.
11. Вывод результата можно заблокировать, если в конце строки ввода ввести знак точка с запятой (;). Значение переменной, которой результат присваивается, храниться в рабочей области.
12. При работе с массивами определены операторы почленного выполнения. В них перед символом операции вводится точка (.).
13. Символ присвоения - знак равенства (=). Равенство, как оператор отношения в условиях, вводится, как двойное равенство (==).

2. Простые вычисления в MatLab

Подготовка к работе

1. По указанной литературе изучить:
   - системное меню редактора MatLab,
   - основные системные команды,
   - правила ввода команд и данных,
   - ранжированные переменные,
   - правила вывода результатов,
   - правила вывода результатов в виде двумерных графиков,
   - правила отладки программ.
2. Разработать алгоритмы решения задач из варианта задания.
3. Составить программы решения задач.

Контрольные вопросы

1. Структура окна редактора MatLab.
2. Правила ввода команд.
3. Правила ввода функций и операндов.
4. Правила ввода выражений.
5. Организация циклов.
6. Правила ввода комментариев.
7. Правила просмотра результатов операций.
8. Правила создания двумерных графиков.
9. Запуск и отладка программ.

Задание к работе

Задача 1.

- Ввести текст в виде комментария, как заглавие программы.
- Ввести исходные данные.
- Задать изменение аргумента.
- Вычислить значения функций 1 и 2 для аргумента в заданном интервале.
- Вывести графики функций одновременно на одном графике в декартовых координатах. Для разных графиков использовать разный тип линий.

Задача 2 .

- Пункты 1...4 задачи 1.
- Вывести графики функций в двух подокнах на одном графике. Графики сделать в столбиковом формате.

Варианты заданий

№	Функция 1	Функция 2	a	b	h
1	y = sin(x)	z= exp(x+3)/5000 - 1	-2π	2π	π/20
2	y = cos(x)	z = 0.00025e^3-x - 0.6	-2π	2π	π/20
3	y = |tg(x)| + 0.1	z = (1+x)^6	-2π	2π	π/20
4	y = (x^2-1)/15	z = 1+sin(x)	-2π	2π	π/20
5	y = (x^3-2)/15	z = 5cos(x)	-2π	2π	π/20
6	y = x^2 - 10	z = 0.025exp(-1.2x)	-5	5	1
7	y = 3sin(x)	z=0.015x^3	-5	5	1
8	y = 4sin(x)	z = 0.05x^2	1	10	1
9	y = 6sin(x)	z = 0.01x^3	-10	10	1
10	y = 2+cos(x)	z = - 0.05(x^2 + 10cos(x))	-8	8	1
11	y = sin^2(x/3)	z = 0.01(x^2 - 40sin(x))	-8	8	1
12	y = cos^3(x)	z = sin(x) + sin(2x)	-π	□	π/8
13	y = 0.5x + cos^2(x)	z = sin^2(x) + cos(x)	-π	□	π/8
14	y = sin(x) + cos^2(2x)	z = x(0.5 + x)exp(0.1x)	-π	□	π/8
15	y = |sin(x)|exp(x/2)	z = 5x - x$^{1.5}$+sin(x)	0	5	0.5

Методические указания

1. Текстовые пояснения в программу вводятся, как комментарий. Он начинается с символа %, который располагается в первой позиции строки. Комментарий - это текст! В него не надо включать символы операций.
2. Для формирования XY графика необходимо:
   - Задать аргумент в формате x=<нач. значение>:<шаг>:<нач. значение>.
   - Вычислить функцию, например, y=f(x).
   - Вывести график процедурой plot(x,y,s). Процедура рисует график прямыми линиями между вычисленными точками.. Здесь s - строковая константа, задающая параметры линии, ее можно пропускать. Определены следующие значения s:

Цвет линии		Тип точки		Тип линии	
y	желтый	.	точка	-	сплошная
m	фиолетовый	o	кружок	:	двойной пунктир
c	голубой	x	крест	-.	штрих пунктир
r	красный	+	плюс	--	штрих
g	зеленый	*	звездочка		
b	синий	s	квадрат		
w	белый	d	ромб		
k	черный	v	треугольник вверх		
		<	треугольник влево		
		>	треугольник вправо		
		p	пятиугольник		
		h	шестиугольник		

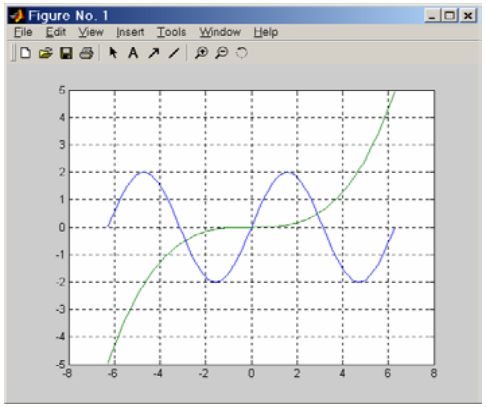
   - Если на одном графике нужно отобразить несколько функций, например, y1=f(x) и y2=f(x)., то они вначале вычисляются, а затем выводятся процедурой plot(x,y1,'s1',x,y2,'s2...), в которой в качестве параметров для каждой функции следуют группы <аргумент, функция, строка типа линии>.
   - Для создания в графическом окне нескольких подокон для вывода графиков используется процедура subplot(m,n,p), где m - число подокон в окне по горизонтали, n - по вертикали, p - номер используемого подокна (нумерация с 1).
   - Для формирования графика в столбиковой форме нужно использовать процедуру bar(x,y). При выводе такого графика в подокно строка программы имеет вид subplot(m,n,p), bar(x,y).

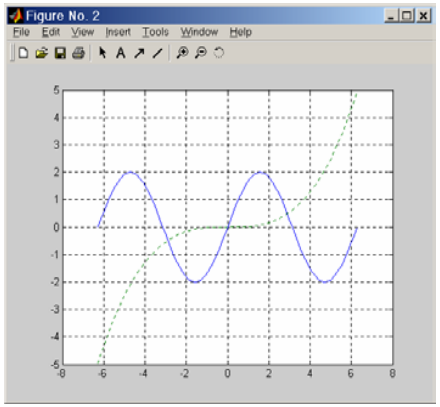
Пример выполнения

Задание	Функция 1	y = 2sin(x)
	Функция 2	z = 0.02x^3
	Начальное значение аргумента	a = -2π
	Конечное значение аргумента	b = 2π
	Шаг изменения аргумента	h = π/20

Задача 1

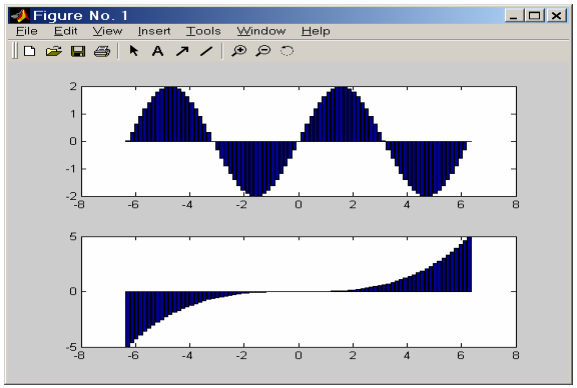
```
% Задача 1
% Диапазон и шаг
a=-2*pi;
b=2*pi;
h=pi/20;
% Задание аргумента
X=a:h:b;
%Расчет функций
Y=2*sin(X);
Z=0.02*X.^3;
% Вывод графиков с одинаковым типом линии в окно 1
figure(1);
plot(X,Y,X,Z);
% Включим координатную сетку
grid on
% Вывод графиков с разными типами линии в окно 2
figure(2);
plot(X,Y,'-',X,Z,':');
% Включим координатную сетку
grid on
```





Задача 2

```
% Задача 2
% Диапазон и шаг
a=-2*pi;
b=2*pi;
h=pi/20;
% Задание аргумента
X=a:h:b;
%Расчет функций
Y=2*sin(X);
Z=0.02*X.^3;
% Вывод графика 1 в виде столбиков в подокно 1
subplot(2,1,1),bar(X,Y);
% Вывод графика 2 в виде столбиков в подокно 2
subplot(2,1,2),bar(X,Z);
```



3. Многомерные вычисления в MatLab

Подготовка к работе.

1. По указанной литературе изучить:
   - правила организации вложенных циклов,
   - правила получения многомерных результатов,
   - вывод многомерных данных в табличной форме,
   - объемная графика,
   - контурная графика.
2. Разработать алгоритмы решения задач из варианта задания.
3. Составить программы решения задач.

Контрольные вопросы

1. Организация вложенных циклов.
2. Правила задания многомерных функций.
3. Связь двумерной функции с матрицей для вывода графиков.
4. Вывод многомерных результатов в форме таблицы.
5. Трехмерная графика в аксонометрии.
6. Трехмерная графика с функциональной окраской раскраской.
7. Трехмерная графика с функциональной раскраской и проекцией.
8. Контурная графика.
9. Объемная контурная графика.
10. Объемная графика с освещением.

Задание к работе

Задача 1. Двумерная функция и объемные графики в своих окнах.

- Ввести исходные данные.
- Вычислить двумерную функцию.
- Вывести функцию в виде 5 трехмерных графиков разного типа.
- Вывести функцию в виде 2 контурных графиков разного типа.

Задача 2. Двумерная функция и объемные графики в подокнах общего окна.

Варианты заданий

№	Функция	Пределы изменения	
		x	y
1	z=sin(x)cos(y)	от -2π до 2π	от -2π до 2π
2	z=sin(x/2)cos(y)	от -2π до 2π	от -2π до 2π
3	z=sin(2x)cos(y)	от -2π до 2π	от -2π до 2π
4	z = sin(x)cos(y/2)	от -2π до 2π	от -2π до 2π
5	z = sin(x/2)cos(2y)	от -2π до 2π	от -2π до 2π
6	z = sin(2x)cos(2y)	от -2π до 2π	от -2π до 2π
7	z = (1+sin(x)/x)(sin(y)/y)	от -2π до 2π	от -2π до 2π
8	z = (sin(x)/x)cos(y)	от -2π до 2π	от -2π до 2π
9	z = (sin(x)/x)|cos(y)|	от -2π до 2π	от -2π до 2π
10	z = (sin(x)/x)y	от -2π до 2π	от -2π до 2π
11	z = (sin(x)/x)|y|	от -2π до 2π	от -2π до 2π
12	z = (sin(x)/x)sin(y)	от -2π до 2π	от -2π до 2π
13	z = (sin(x)/x)|sin(y)|	от -2π до 2π	от -2π до 2π
14	z = (sin(x)/x)(1-y)	от -2π до 2π	от -2π до 2π
15	z = (sin(x)/x)|y+0.5|	от -2π до 2π	от -2π до 2π

Методические указания

1. Формирование задач. В работе предусмотрены 2 задачи, в каждой из которых вычисляется двумерная функция, описывающая объемную фигуру, и строятся поверхностные и контурные графики с использованием различных графических функций. В первой задаче каждый график выводится в свое окно, во второй в подокна общего окна.
2. Представление матриц. Значения матрицы выводятся в текстовой форме построчно. Если столбцы в экране не умещаются, происходит разбиение на группы столбцов, которые выводятся последовательно. Табличный вывод в MatLab, как в MathCAD, не предусмотрен.
3. Поверхностный и контурный графики. Для формирования поверхностного или контурного графика необходимо:
   - задать число точек по координатам X иY,
   - создать вложенные циклы по X иY, вычислить функцию Z=f(X,Y),
   - ввести номер графического окна, вывести туда график выбранного типа.
4. Следует использовать графики:
   - трехмерный с аксонометрией, функция plot3(X,Y,Z),
   - трехмерный с функциональной окраской, функция mesh(X,Y,Z),
   - трехмерный с функциональной окраской и проекцией, функция meshc(X,Y,Z),
   - трехмерный с функциональной окраской и проекцией, функция surf(X,Y,Z),
   - контурный, функция contour(X,Y,Z),
   - объемный контурный, функция contour3(X,Y,Z),
   - трехмерный с освещением, функция surfl(X,Y,Z).

5. В каждом окне можно рисовать несколько графиков с наложением друг на друга. В списке параметров для каждого графика параметры перечисляются группами последовательно (в работе график для окна один). В каждую группу входят:
   - X - первая координата площадки основания,
   - Y - вторая координата площадки основания,
   - Z - значение функции.

Пример выполнения

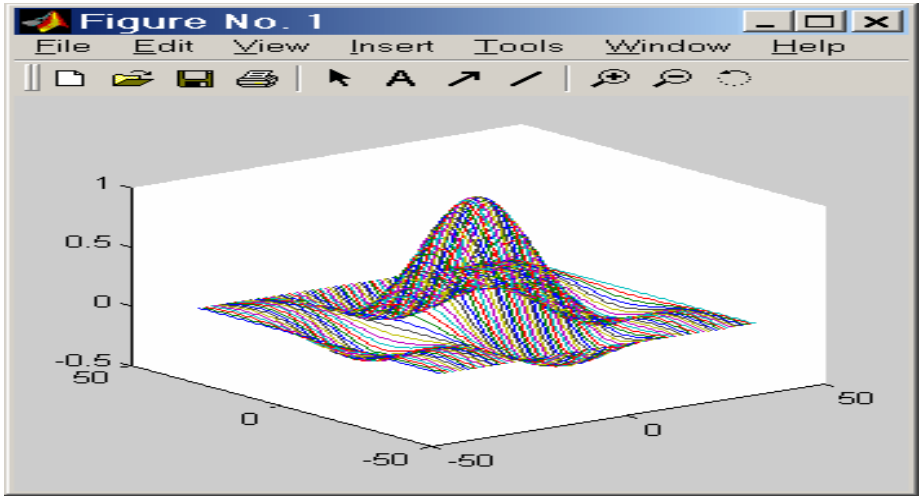
Задание

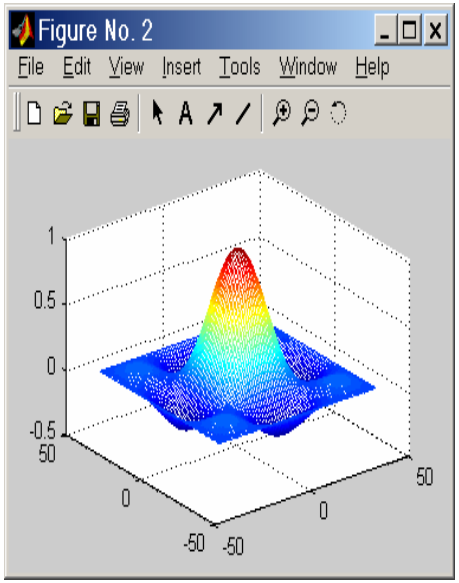
Функция $z = \frac{\sin(x)}{x} \cdot \frac{\sin(y)}{y}$. Пределы изменения аргументов -2π...2π

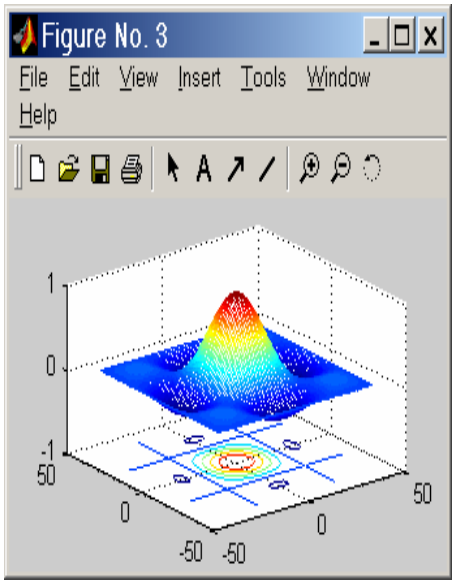
Задача 1

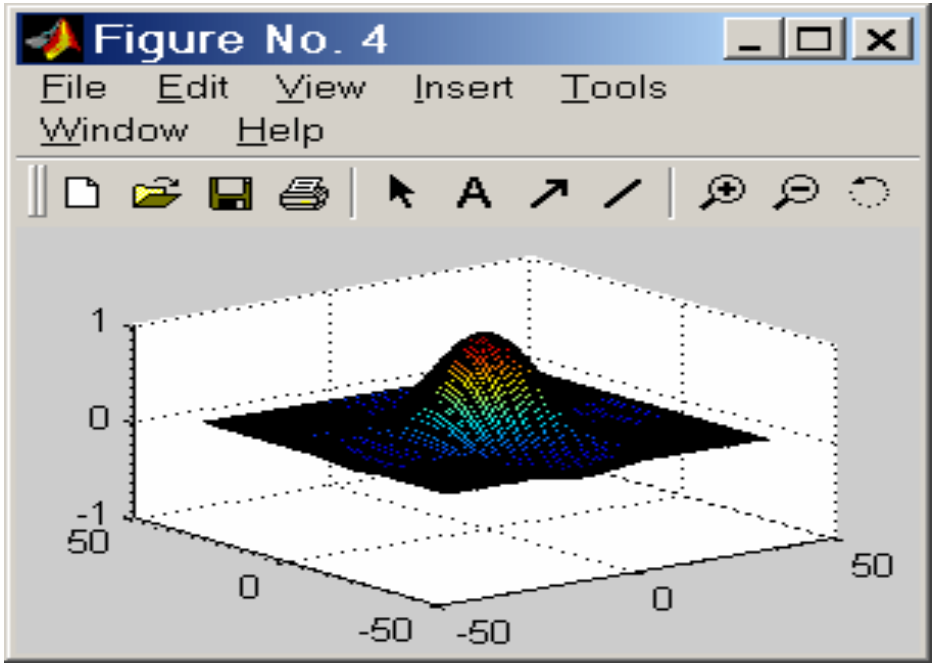
```
% Задача 1
% Число точек и шаг
N=40;
h=pi/20;
% Расчет матрицы
for n=1:2*N+1
 if n==N+1 A(n)=1; else A(n)=sin(h*(n-N-1))/(h*(n-N-1)); end;
end;
for n=1:2*N+1
   for m=1:2*N+1
      Z(n,m)=A(n)*A(m);
   end;
end;
% Задание площадки
[X,Y]=meshgrid([-N:1:N]);
% Вывод графика в аксонометрии в окно 1
figure(1);
plot3(X,Y,Z);
% вывод трехмерного графика с функциональной окраской в окно 2
figure(2);
mesh(X,Y,Z);
% вывод трехмерного графика с функциональной окраской и проекцией в окно 3
figure(3);
meshc(X,Y,Z);
% вывод трехмерного графика с проекцией в окно 4
figure(4);
surf(X,Y,Z);
% Вывод контурного графика в окно 5
figure(5);
contour(X,Y,Z)
```

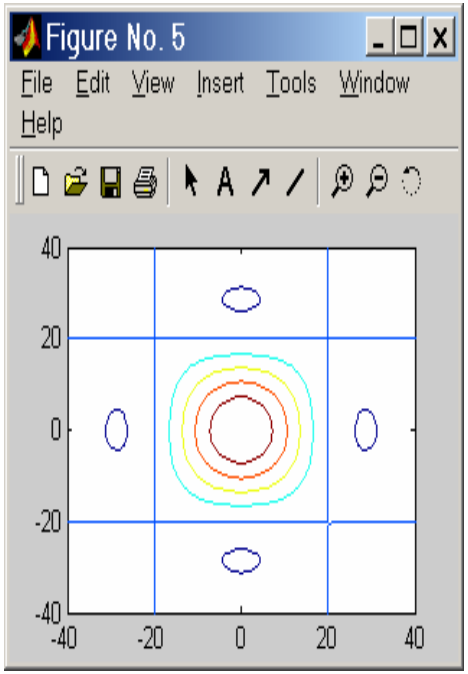
```
% Вывод объемного контурного графика в окно 6
figure(6);
contour3(X,Y,Z)
% Вывод объемного графика с освещением в окно 7
figure(7);surfl(X,Y,Z)
```

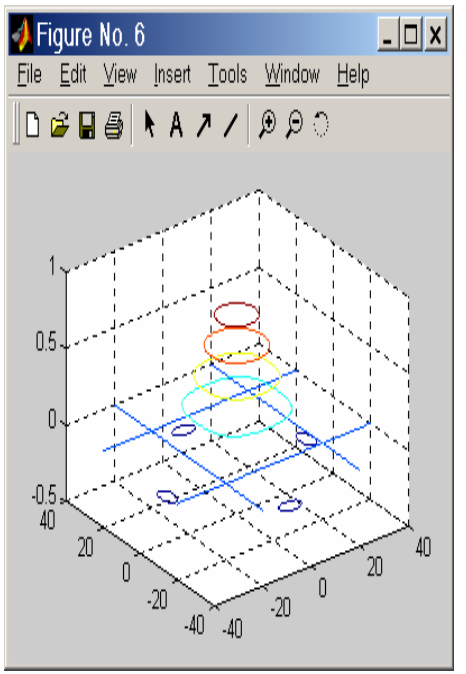


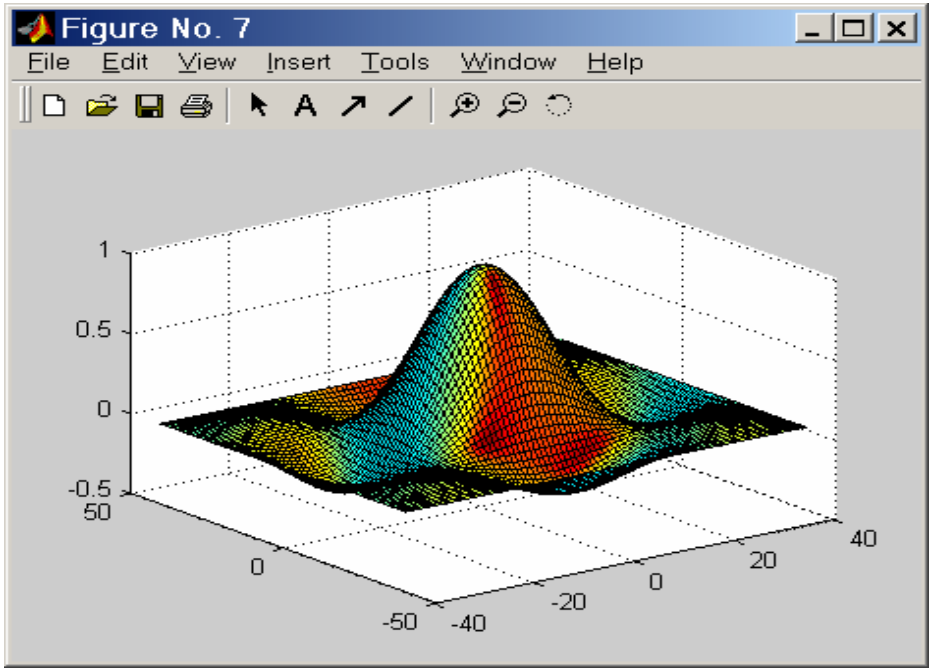










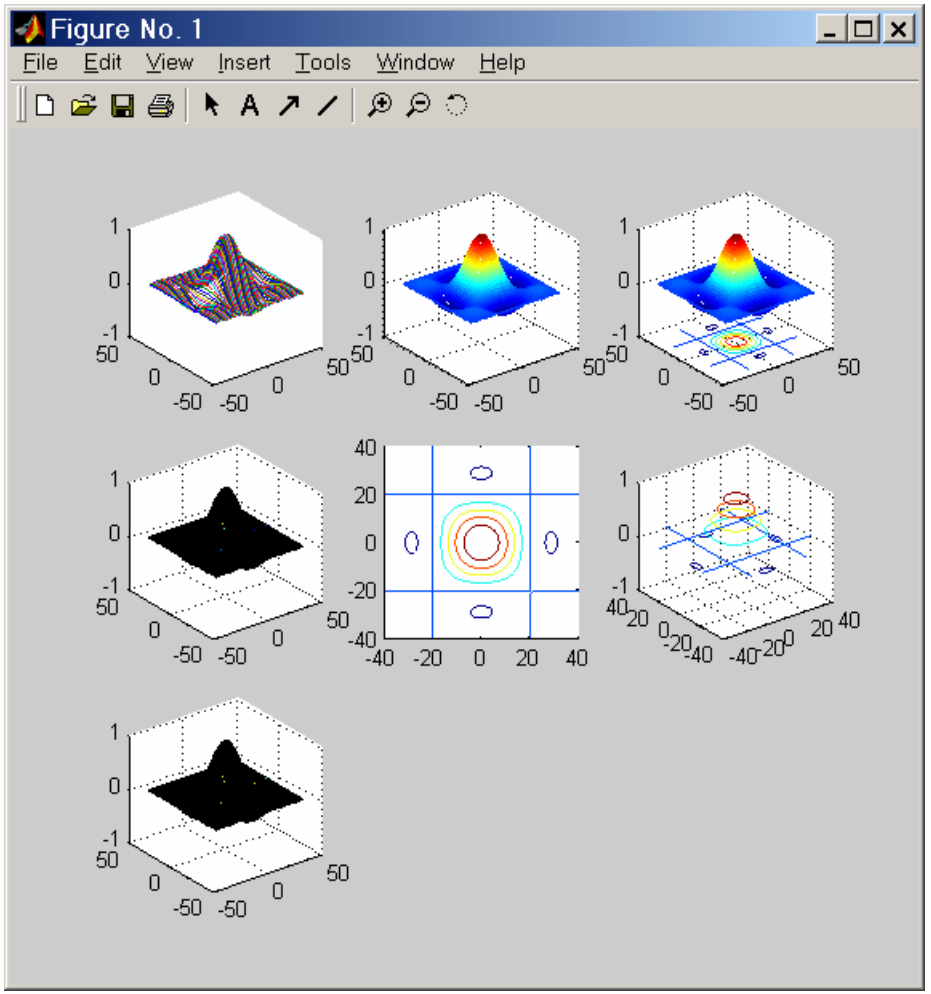


Задача 2

```
% Задача 2
% Число точек и шаг
N=40;
h=pi/20;
% Расчет матрицы
for n=1:2*N+1
 if n==N+1 A(n)=1; else A(n)=sin(h*(n-N-1))/(h*(n-N-1)); end;
end;
for n=1:2*N+1
   for m=1:2*N+1
      Z(n,m)=A(n)*A(m);
   end;
end;
% Задание площадки
[X,Y]=meshgrid([-N:1:N]);
% Вывод графика в аксонометрии в подокно 1
subplot(3,3,1),plot3(X,Y,Z);
% вывод трехмерного графика с функциональной окраской в подокно 2
subplot(3,3,2),mesh(X,Y,Z);
% вывод трехмерного графика с функциональной окраской и проекцией в подокно 3
```



```
subplot(3,3,3),meshc(X,Y,Z);
% вывод трехмерного графика с проекцией в подокно 4
subplot(3,3,4),surf(X,Y,Z);
% Вывод контурного графика в подокно 5
subplot(3,3,5),contour(X,Y,Z)
% Вывод объемного контурного графика в подокно 6
subplot(3,3,6),contour3(X,Y,Z)
% Вывод объемного графика с освещением в подокно 7
subplot(3,3,7),surfl(X,Y,Z)
```



4. Решение уравнений в MatLab.

Подготовка к работе.

1. По указанной литературе изучить:
   - правила локализации решения,
   - решение одного уравнения,
   - решение системы из двух уравнений.
2. Разработать алгоритмы решения задач из варианта задания.
3. Составить программы решения задач.

Контрольные вопросы

1. Задание функции пользователя.
2. Локализация решений уравнения.
3. Решение нелинейного уравнения с использованием функции fzero.
4. Вывод полученных решений уравнения.
5. Локализация решений системы из двух уравнений.
6. Решение системы из двух уравнений.
7. Вывод полученных решений системы уравнений.

Задание к работе

Задача 1. Решение нелинейного уравнения.

- Создать Mat-функцию для функции f1(x).
- Создать файл программы. Ввести текст заглавия задачи, как комментарий. Ввести в него аргументы в заданных пределах.
- Вывести y(x)=f1(x) в виде XY графика. По нему определить приближенно корни уравнения y(x)=0. Если корни на графике не просматриваются, то изменить пределы изменения аргумента и повторить операции.
- Для каждого корня найти точное значение, используя функцию fzero.
- Сформировать строку с результатами и вывести ее в заголовок окна графика.

Задача 2. Решение системы из двух нелинейных уравнений.

- Создать Mat-функции для функций f2(x) и f3(x) = f1(x) - f2(x).
- Создать файл программы. Ввести текст заглавия задачи, как комментарий. Ввести в него аргументы в заданных пределах.
- Вывести f1(x) и f2(x) в виде XY графиков. По нему определить приближенно корни системы уравнений, как координаты точек пересечения графиков f1(x) и f2(x).Если корни на графике не просматриваются, то изменить пределы изменения аргумента и повторить операции.
- Для каждого корня найти точное значение, используя функцию fzero к переменной f3(x).
- Сформировать строку с результатами и вывести ее в заголовок окна графика.

Варианты заданий

№	f1(x)- полином 3-ей степени с коэффициентами a				f2(x)
	a3	a2	a1	a0	333
1	0	-1	4	-1	0.2exp(x)-20
2	0	2	-2	-15	40|cos(x)|
3	0	1	4	-1	10ln(x+5.5)
4	0	9	-8	-70	100|sin(x)|
5	0	-4	4	50	70cos(x)
6	.1	-5	4	40	60exp(|0.1*x|)-100
7	.2	-3	2	30	20sin(2x)
8	.3	-6	1	50	exp(|x|)sin(2x)
9	.4	-9	1	70	exp(|x|)cos(3x)
10	.5	-7	5	60	-60|cos(x)|
11	-.1	-4	9	60	15log(x+5.1)
12	-.2	-6	-7	55	-50ln(x+5.1)
13	-.3	-9	-8	75	-100|cos(x)|
14	-.4	7	8	-75	100sin(x/2)
15	-.5	1	4	-1	40cos(x/2)

Методические указания

1. При решении нелинейного уравнения оно формируется из функций задания, как f1(x)=0.
2. При решении системы из двух нелинейных уравнений из функций задания формируется уравнение f3(x) = f1(x) - f2(x) = 0. Функции из задания надо определить, как функции пользователя, создав для них новые Mat-функции. Это упростит обращения к ним при решении уравнений.
3. В качестве имен функций можно выбрать fun1, fun2 и fun3. Mat-функции надо создавать в новом окне редактора. Формат Mat-функции:

   function [var1 var2 …] = <имя функции>(список параметров)
   var1=<выражение>
   var2=<выражение>

   function - зарезервированное слово,
   [var1 var2 …] - вектор имен возвращаемых функцией значений.
4. В нашем случае возможное такое описание Mat-функции:

   function = fun1(x)
   f1=<выражение>
5. Локализация корней. Уравнение или система уравнений может иметь несколько корней, каждый из которых ищется отдельно. При этом для каждого корня надо задать диапазон аргумента, в котором он находится (только один!).
6. Это делается путем локализации корня. Для этого надо просчитать значения функций в заданном интервале и построить их графики. Начальное значение для решения одного уравнения - точка пересечения графиком функции оси X. График выводится

процедурой, в которой аргументы - переменная x и анализируемая функция. С помощью grid on график делается с координатной сеткой:

plot(x,fun1(x));grid on;

7. Начальное значение для решения системы из двух уравнений - точка взаимного пересечения графиков функций. Графики выводятся процедурой, в которой для каждого графика следует группа параметров:

plot(x,fun1(x),x,fun2(x));grid on;

8. Функция fzero. Используется для нахождения корня нелинейного уравнения. Формат этой функции:

<имя результата>=fzero('имя функции',[левый предел: правый предел])

Пример использования:

```
% Вектор аргумента
x=[a:h:b];
% График локализации корней
plot(x,fun1(x));grid on;
% Найти первый корень
x1=fzero('fun1(x)',[-4 -3]);
```

Пример выполнения

Задание Функция 1 $f1(x) = -0.85*x^3-2x^2+7x+2$

Функция 2 $f2(x) = 6\cos(x) - 5$

Для нахождения корней выражений будем использовать процедуры MatLab, для которых нужно выражения оформить, как Mat-функции. Создадим в редакторе MatLab новые встроенные Mat-функции с именами fun1, fun2 и fun3, тексты которых сохраним в файлах: fun1.m, fun2.m и fun3.m. Третья функция нужна для решения системы из двух уравнений по тому же алгоритму, что для одного уравнения, но с преобразованием двух уравнений в одно разностное.

Файл fun1.m function f1=fun1(x)

```
f1=-0.85*x.^3-2*x.^2+7*x+2
```

Файл fun2.m function f2=fun2(x)

```
f2=6*cos(x)-5
```

Файл fun3.m function f3=fun3(x)

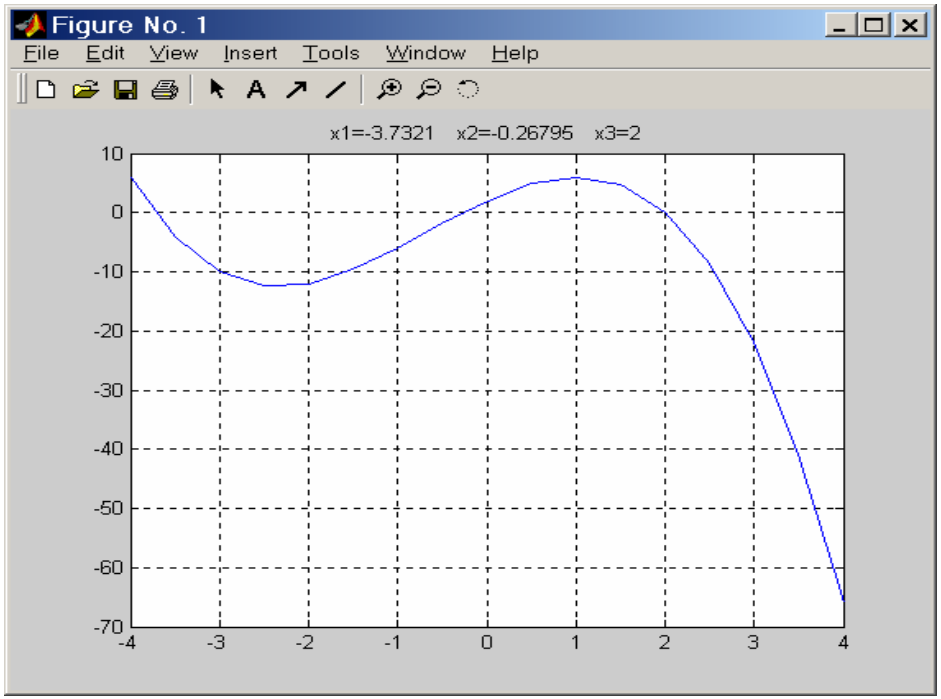
f3=-0.85*x.^3-2*x.^2+7*x+2-6*cos(x)+5

Задача 1

```
% Задача1
% Нахождение корней выражения
% Пределы и шаг аргумента
a=-4;
b=4;
h=0.5;
```

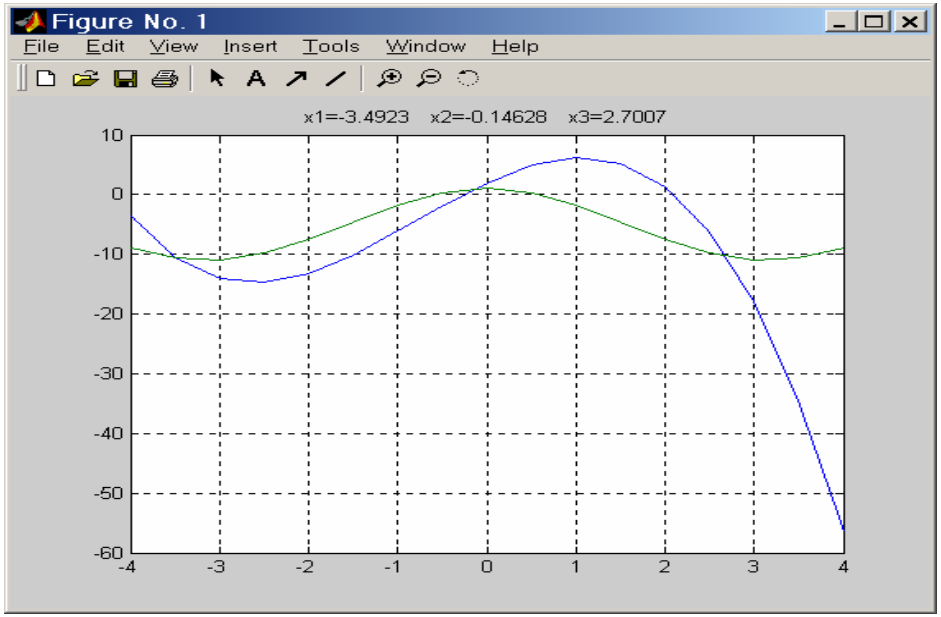


```
% Вектор аргумента
x=[a:h:b];
% График локализации корней
plot(x,fun1(x));grid on;
% Найти первый корень
x1=fzero('fun1(x)',[-4 -3]);
% Найти второй корень
x2=fzero('fun1(x)',[-1 0]);
% Найти третий корень
x3=fzero('fun1(x)',[1 3]);
% Получить строку результатов
Result=strcat('x1=',num2str(x1),'   x2=',num2str(x2),'   x3=',num2str(x3));
% Включить его в график в форме заголовка
title(Result)
```



Задача 2

```
% Задача2
% Решение системы нелинейных уравнений
% Пределы и шаг аргумента
a=-4;
b=4;
h=0.5;
% Вектор аргумента
x=[a:h:b];
% График локализации корней
plot(x,fun1(x),x,fun2(x));grid on;
% Найти первый корень
x1=fzero('fun3(x)',[-4 -3])
% Найти второй корень
x2=fzero('fun3(x)',[-1 0]);
% Найти третий корень
x3=fzero('fun3(x)',[2 3]);
% Получить строку результатов
Result=strcat('x1=',num2str(x1),'   x2=',num2str(x2),'   x3=',num2str(x3));
% Включить его в график в форме заголовка
title(Result)
```



5. Символьные вычисления в MatLab

Подготовка к работе

1. По указанной литературе изучить правила:
   - упрощения выражений,
   - раскрытия скобок в выражениях,
   - факторизации выражений,
   - подстановки подвыражений,
   - символического дифференцирования и интегрирования,
   - разложения в ряд Тейлора,
   - преобразования в элементарные дроби,
   - преобразований Фурье, Лапласа и z-.
2. Подготовить ожидаемые решения для задач путем ручного вывода формул (или найдя их по математическим справочникам).

Контрольные вопросы

1. Задание символьных переменных с помощью апострофа и функции sym.
2. Функция syms создания группы символьных объектов.
3. Функция расширить (разложить по степеням) - expand.
4. Функция свернуть (упростить) - simple.
5. Функция упростить - simplify.
6. Функция факторизовать (разложить на простые множители) - factor.
7. Функция собрать по степеням - collect.
8. Функция подстановки подвыражения переменной - subexpr.
9. Функция дифференцировать - diff.
10. Функция интегрировать - int.
11. Функция найти предел - limit.
12. Функция разложить в ряд Тейлора -teylor.
13. Функция решения уравнений в символьной форме - solver.
14. Функции работы с матрицами.
15. Функция отображения графиков символьных функций
16. Преобразования Фурье, Лапласа, z-.

Задание к работе

Задача 1. Развертка/свертка.
- Ввести выражение f1(x) и развернуть его.
- Полученное выражение свернуть. Сравнить результат с f1(x).

Задача 2. Дифференцировать/интегрировать.
- Ввести выражение f1(x) и найти производную по x.
- Для полученного выражения найти неопределенный интеграл. Сравнить с f1(x).

Задача 3. Разложить в ряд Тейлора.
- Ввести выражение f2(x) и найти его разложение в ряд Тейлора.
- Построить XY график для f2(x) и его разложения в ряд Тейлора F2(x).

Задача 4. Работа с командой funtool.

- Задать функцию f1 и выполнить с ней операции задачи 2.
- Задать функцию f2 и выполнить с ней операцию символьного дифференцирования.

Варианты заданий

№	f1(x)	f2(x)
1	$(1+x)^2$	ax^3+bx^2+cx+d
2	$(1-x)^2$	sin(ax)
3	$(a+x)^2$	cos(ax)
4	$(a-x)^2$	sec(x)
5	$(1+x)^3$	exp(ax)
6	(1-x)3	x(ln(x)-1)
7	(a+x)3	-csc(x)
8	(a-x)3	$1/(1+x^2)$
9	$(1+x)^4$	1/(a+bx)
10	$(1-x)^4$	$1/(1-x^2)$
11	$(a+x)^4$	$-\cos^3(x)/3$
12	$(a-x)^4$	$\sin^3(x)/3$
13	$(1+x)^5$	$x^2(\ln(x)-0.5)/2$
14	$(1-x)^5$	$-(\ln(x)+1)/x$
15	$(a+x)^5$	$\ln^2(x)/2$
16	$(a+x)^5$	$\ln^3(x)/3$

Методические указания

1. Обратите внимание: в задании употреблены имена стандартных функций, принятые в MatLab. Они могут не совпадать с принятыми в математике. В именах функций имеет значение высота букв.
2. Результат символьных преобразований выводится в командное окно с новым символьным именем. В отличие от вывода результатов несимвольных преобразований выводимое значение размещается без абзацного отступа.
3. Развертка и свертка выражений. Под разверткой понимается запись выражения в развернутой форме (с открытыми скобками). Под сверткой понимается обратное действие.
   - Начать надо с указания символьных переменных. Для этого применяется функция: syms перечень имен переменных через пробел
   - Затем надо ввести исходную функцию.
   - Далее следует выполнить операцию развертки и получить результат с именем f1_new. Для этого используется функция expand(имя переменной).
   - Затем над f1_new нужно выполнить операцию свертки и получить результат с именем f1_old. Для этого используется функция simple(имя переменной).
   - Признаки правильного выполнения операций - при свертке результата развертки восстанавливается исходная функция.
4. Дифференцирование и интегрирование выражений. При дифференцировании выражения находится производная по выбранной переменной. При интегрировании выражения находится неопределенный интеграл (первообразная) по выбранной переменной. Константа по умолчанию - нуль.
   - Начать надо с указания символьных переменных. Для этого применяется функция: syms перечень имен переменных через пробел
   - Затем надо ввести исходную функцию.
   - Далее следует выполнить операцию дифференцирования и получить результат с именем f1_new. Для этого используется функция diff(f1,'x',n). здесь f1- имя функции, 'x' - имя переменной (вводится, как строка, в апострофах), по которой производится дифференцирование, n - порядок производной.
   - Затем над f1_new нужно выполнить операцию интегрирования и получить результат с именем f1_old. Для этого используется функция int(f1_new,'x'). здесь f1_new - имя функции, 'x' - имя переменной (вводится, как строка), по которой производится интегрирование.
   - Признаки правильного выполнения операций - при интегрировании результата дифференцирования восстанавливается исходная функция.
5. Разложение в ряд Тейлора. При этом для заданного выражения находится ряд Тейлора с остаточным членом, величина которого зависит от точности, выбираемой при выполнении операции. Остаточный член отбрасывается.
   - Начать надо с указания символьных переменных. Для этого применяется функция: syms перечень имен переменных через пробел
   - Затем надо ввести исходную функцию.
   - Далее следует выполнить операцию разложения в ряд Тейлора и получить результат с именем f1_new. Для этого используется функция taylor(f1,n,'x',a). Здесь f1 - имя функции, переменной, n - порядок остаточного члена, , 'x' - имя переменной (вводится, как строка, в апострофах), по которой производится разложение, a - значение переменной, для которого делается разложение (если оно пропускается, то предполагается a=0).
   - Затем над f1_new нужно выполнить операцию свертки и получить результат с именем f1_old. Для этого используется функция simple(имя переменной).
   - Признаки правильного выполнения операции - в окрестности точки a графики исходной и полученной функций совпадают. Для построения графиков символьных функций имеется процедура ezplot(f2,-h,h);grid on. Здесь f - имя символьной функции, (-h h) - нижний и верхний предел значений аргумента, grid on - включает в графике координатную сетку. В заголовок графического окна помещается описание функции. По этой причине в одно графическое окно можно вывести только один график.
6. Для работы с символьными функциями предусмотрена функция оболочка - fuhtool. Она представляет собой интерактивный графический калькулятор, позволяющий быстро построить две функции одной переменной f(x) и g(x). выводятся три автономных окна: два графических и управляющее.
   - Графики отображаются в отдельных окнах с заголовками.
   - Управляющее окно содержит:
     - Два поля ввода функций.
     - Поле ввода пределов переменной x в формате [min,max].

- Поле ввода масштабирующего коэффициента a.

– Управление осуществляется кнопками, сгруппированными в 4 ряда:
  - Первый - тип символьного преобразования f(x).
  - Второй - тип масштабирования f(x): f+a, f-a, f*a, f/a, f^a, f(x+a), f(x*a).
  - Третий - тип замены f(x) на комбинацию f(x) и g(x).
  - Четвертый - управляющие операции:
    → Insert - ввести f(x) в библиотеку встроенных функций,
    → Cycle - циклически вызвать f(x) из библиотеки встроенных функций,
    → Delete - удалить f(x) из библиотеки встроенных функций,
    → Reset - установить утилиту в исходное состояние,
    → Help - вызов справки,
    → Demo - демонстрация использования,
    → Close - закрыть.

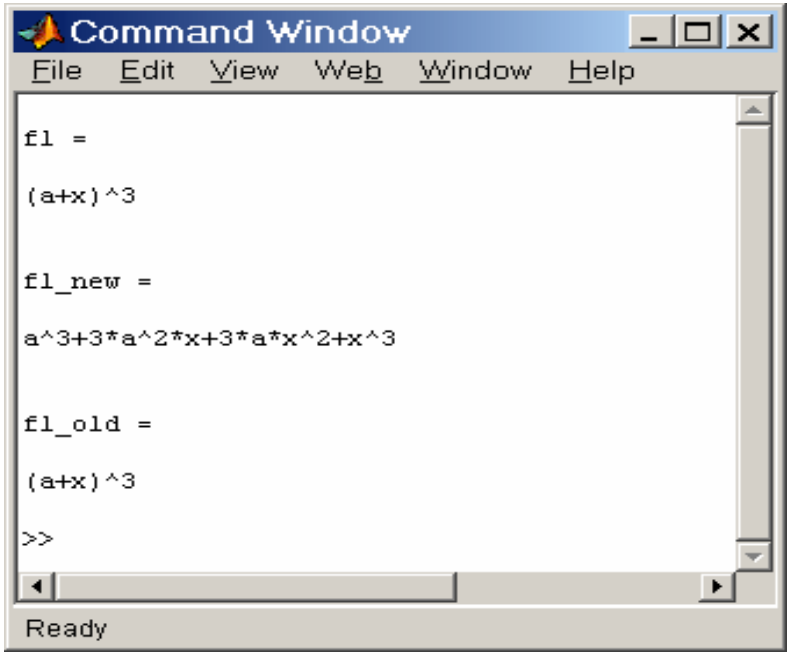
Пример выполнения

Задание f1(x)=$(a+x)^3$

f2(x)=sin(x)/x

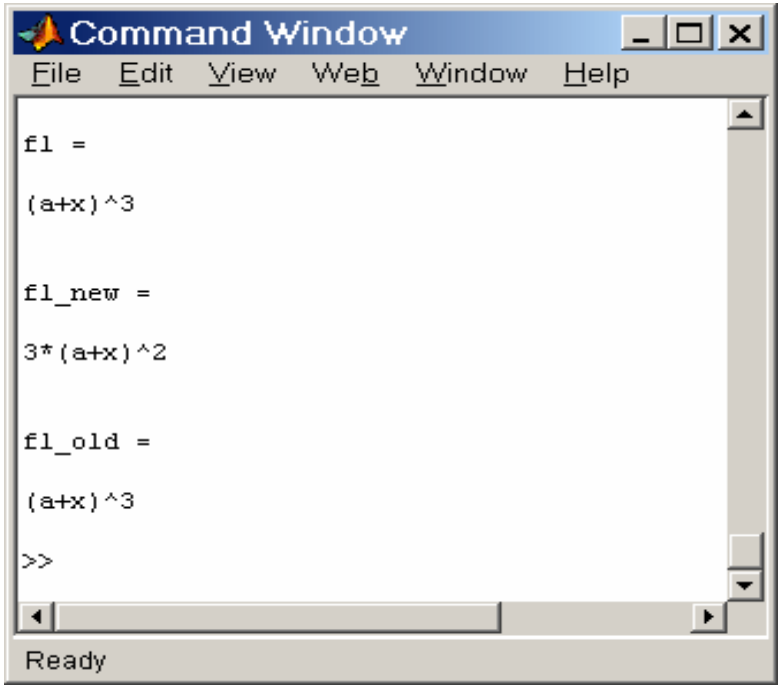
Задача 1

```
% Задача 1
% Определить символьные переменные
syms a x;
% Функция
f1=(a+x)^3
% Расширить ее
f1_new=expand(f1)
% Свернуть расширенное
f1_old=simple(f1_new)
```

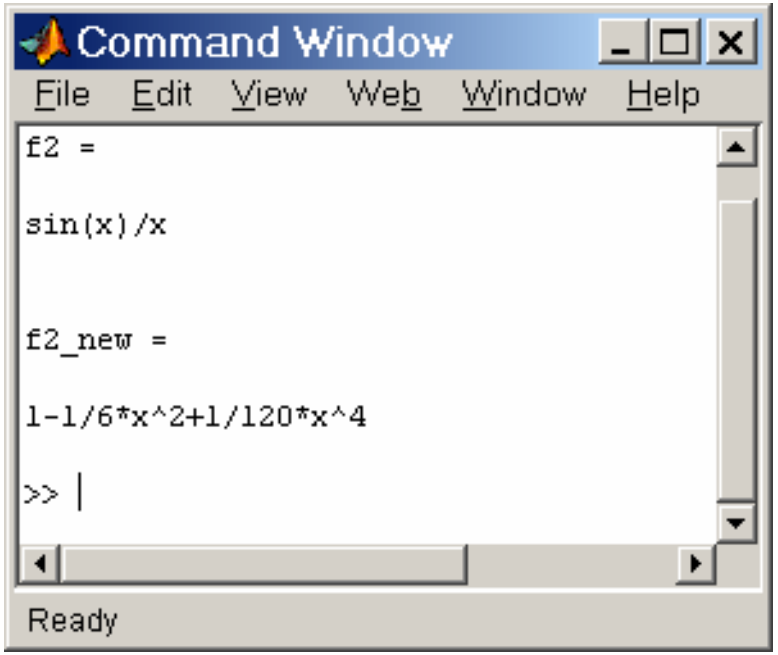
Задача 2

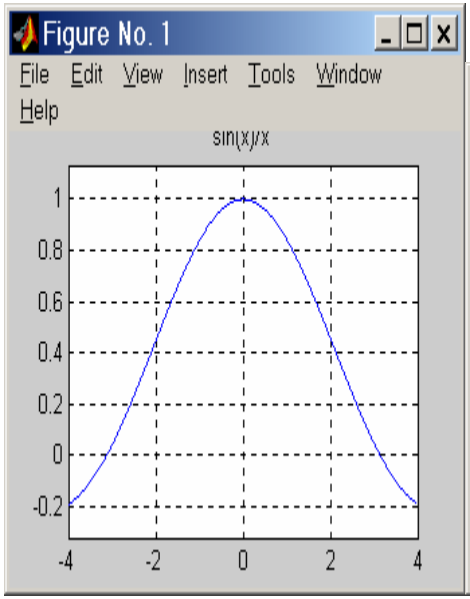
```
% Задача 2
% Определить символьные переменные
syms a x;
% Функция
f1=(a+x)^3
% Найти первую производную по x
f1_new=diff(f1,'x',1)
% Найти неопределенный интеграл по x
f1_old=int(f1_new,'x')
```

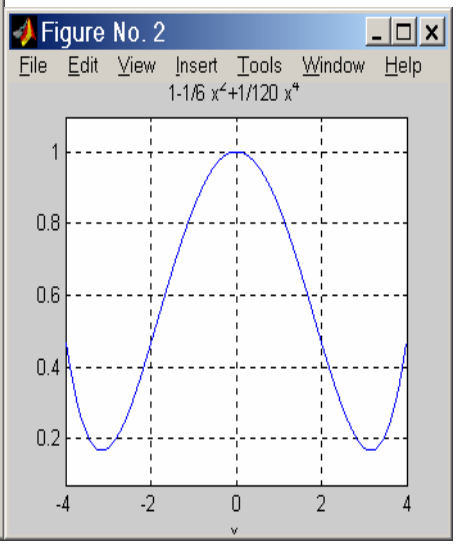


Задача 3

```
% Задача 3
% Определить символьные переменные
syms x;
% Функция
f2=sin(x)/x
% Найти разложение Тейлора по x в точке 0
f2_new=taylor(f2,5,'x',0)
% Диапазон просмотра
h=4;
% График f2
figure(1);
ezplot(f2,-h,h);grid on
% График f2_new
figure(2);
ezplot(f2_new,-h,h);grid on
```

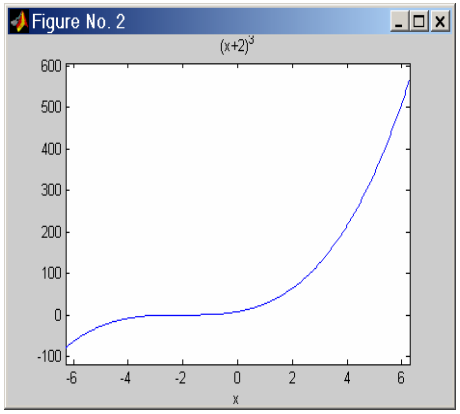



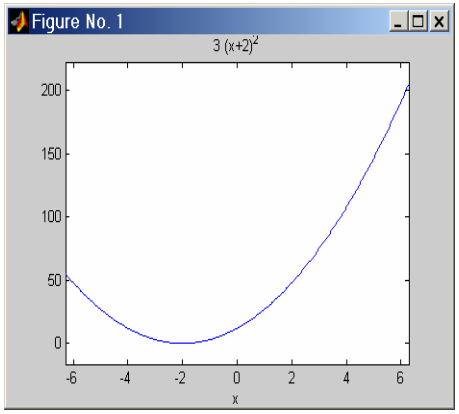


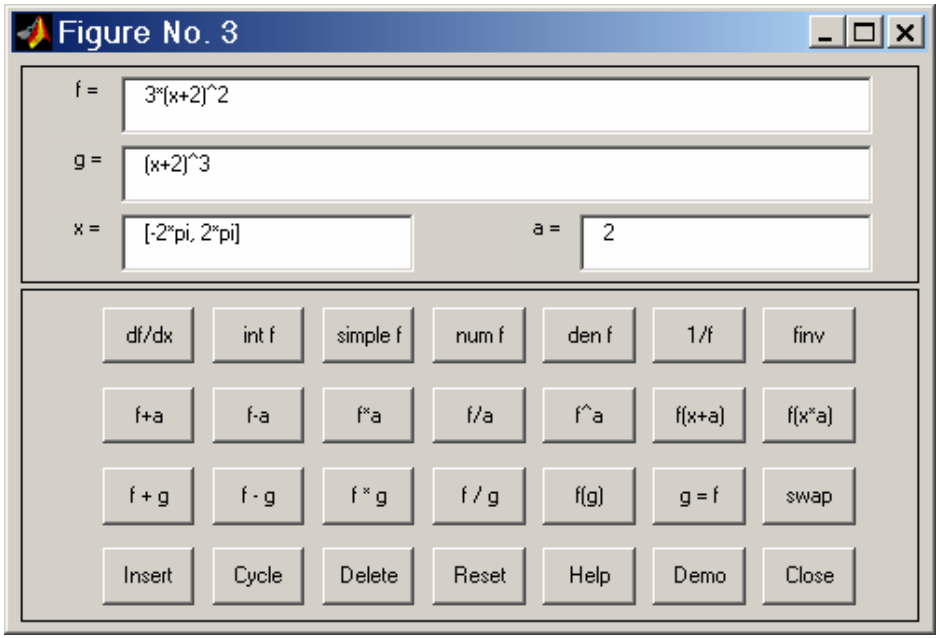


Задание 4

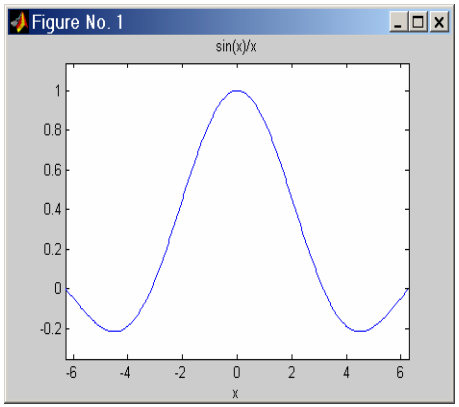
Работа с функцией f1(x)

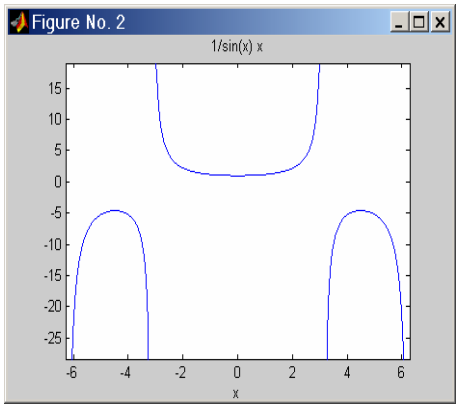


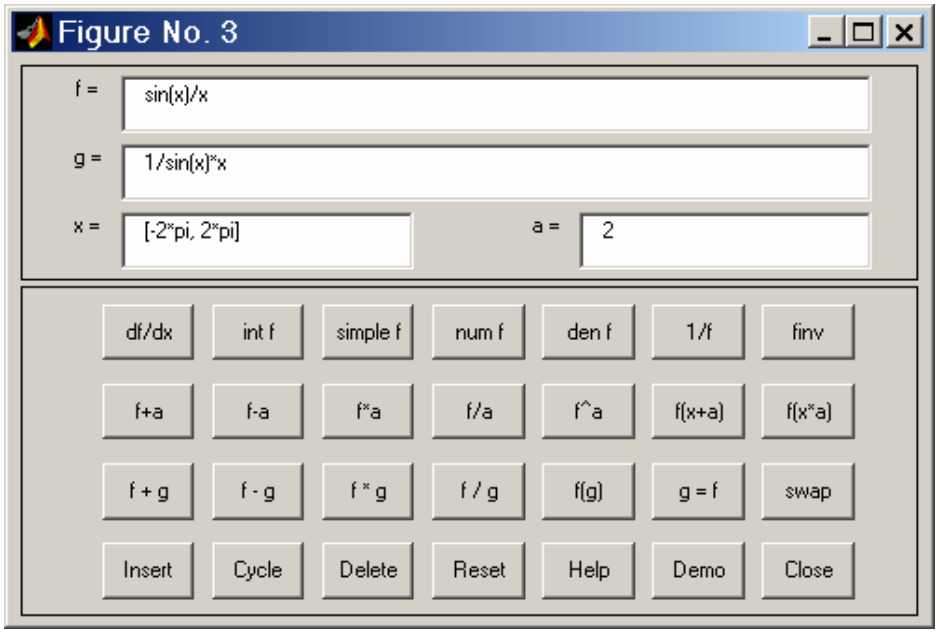




Работа с функцией f2(x)







Внимание: funtool использует средства символьной математики. Поэтому при вычислении sin(x)/x неопределенность при x=0 раскрывается. Отрабатываются также бесконечно большие значения.

6. Моделирование устройства с помощью Simulink

Подготовка к работе

1. По указанной литературе изучить:
   - основы Simulink,
   - правила создания моделей в Simulink,
   - правила моделирования в Simulink,
   - иерархическую библиотеку Simulink.
2. Разработать структуру модели устройства для варианта задания.

Контрольные вопросы

1. Назначение Simulink.
2. Правила построения моделей в Simulink.
3. Правила моделирования в Simulink.
4. Структура иерархической библиотеки Simulink.
5. Блоки из папки Continuous библиотеки Simulink.
6. Блоки из папки Discrete библиотеки Simulink.
7. Блоки из папки Functions & Tables библиотеки Simulink.
8. Блоки из папки Math библиотеки Simulink.
9. Блоки из папки Nonlinear библиотеки Simulink.
10. Блоки из папки Signal & Systems библиотеки Simulink.
11. Блоки из папки Sources библиотеки Simulink.
12. Блоки из папки Sinks библиотеки Simulink.
13. Блоки из Communication Blockset.
14. Блоки из DSP Blockset.
15. Блоки из Motorola DSP Blockset.
16. Блоки из Communication Blockset.
17. Другие наборы блоков.

Задание к работе

Задача 1. Простая модель устройства в соответствии с вариантом задания.

- Создать модель. В ней сигнал от источника поступает на функциональный блок. Регистратор с двумя входами позволяет наблюдать сигналы на входе и выходе функционального блока.
- Провести ее моделирование.

Задача 2. Расширенная модель устройства в соответствии с вариантом задания.

- Создать модель. В ней к модели задачи 1 добавляется параллельная ветвь с вторым функциональным блоком. Регистратор с тремя входами позволяет наблюдать сигналы на входах и выходах функциональных блоков.
- Провести ее моделирование.

Варианты заданий

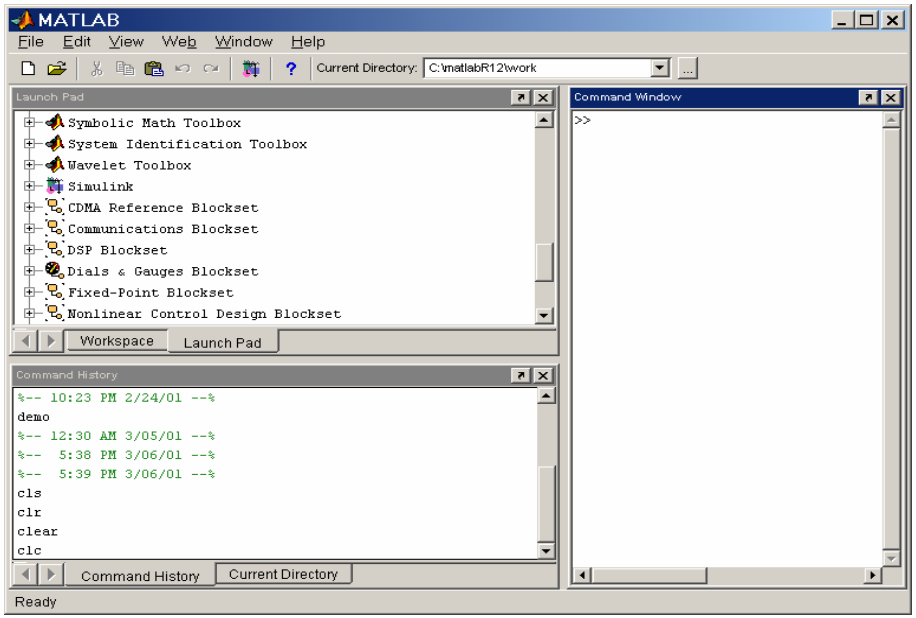
№	Источник сигнала	Блоки	
		Первый	Дополнительный
1	Sine Wave Синус	Gain Усиление	Derivate Дифференциатор
2	Pulse Generator Импульсы	Saturation Ограничитель	Integrator Интегратор
3	Repeating Sequence Пила	Quantizer Квантизатор	Gain Усиление
4	Ram Линейно нарастающий	Derivate Дифференциатор	Saturation Ограничитель
5	Chirp Signal Переменной частоты	Integrator Интегратор	Quantizer Квантизатор
6	Sine Wave Синус	Transport delay Задержка	Derivate Дифференциатор
7	Pulse Generator Импульсы	Dead Zone Мертвая зона	Integrator Интегратор
8	Repeating Sequence Пила	Gain Усиление	Transport delay Задержка
9	Ram Линейно нарастающий	Saturation Ограничитель	Dead Zone Мертвая зона
10	Chirp Signal Переменной частоты	Quantizer Квантизатор	Gain Усиление
11	Sine Wave Синус	Derivate Дифференциатор	Saturation Ограничитель
12	Pulse Generator Импульсы	Integrator Интегратор	Quantizer Квантизатор
13	Repeating Sequence Пила	Transport delay Задержка	Derivate Дифференциатор
14	Ram Линейно нарастающий	Dead Zone Мертвая зона	Transport delay Задержка
15	Chirp Signal Переменной частоты	Derivate Дифференциатор	Saturation Ограничитель

В таблице названия функциональных блоков даны на русском и английском языках (так, как они названы в браузере библиотеки блоков Simulink).

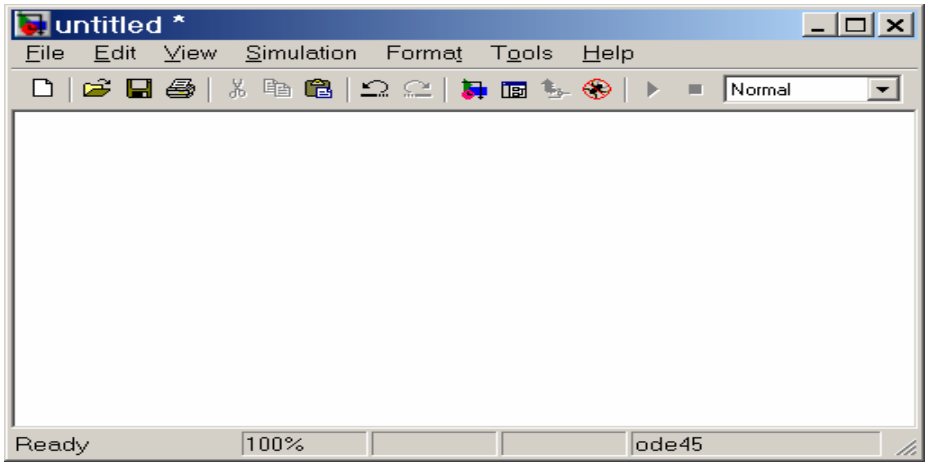
Методические указания

1. Модель устройства содержит источник сигнала, функциональные блоки и средства наблюдения за поведением системы (дисплей, численный индикатор и др.).
2. Во всех вариантах задания нужно использовать дисплей с одним входом в задаче 1 и с двумя входами в задаче 2.
3. Первое действие - запустить Matlab. При этом возникает стартовое диалоговое окно, в котором докированы три встроенных окна: Command Window (командное) - справа,

Launch Pad (Средства запуска) - в левом верхнем углу, Command Hustory (История команд) - в левом нижнем углу. Каждое подокно можно освободить из дока.

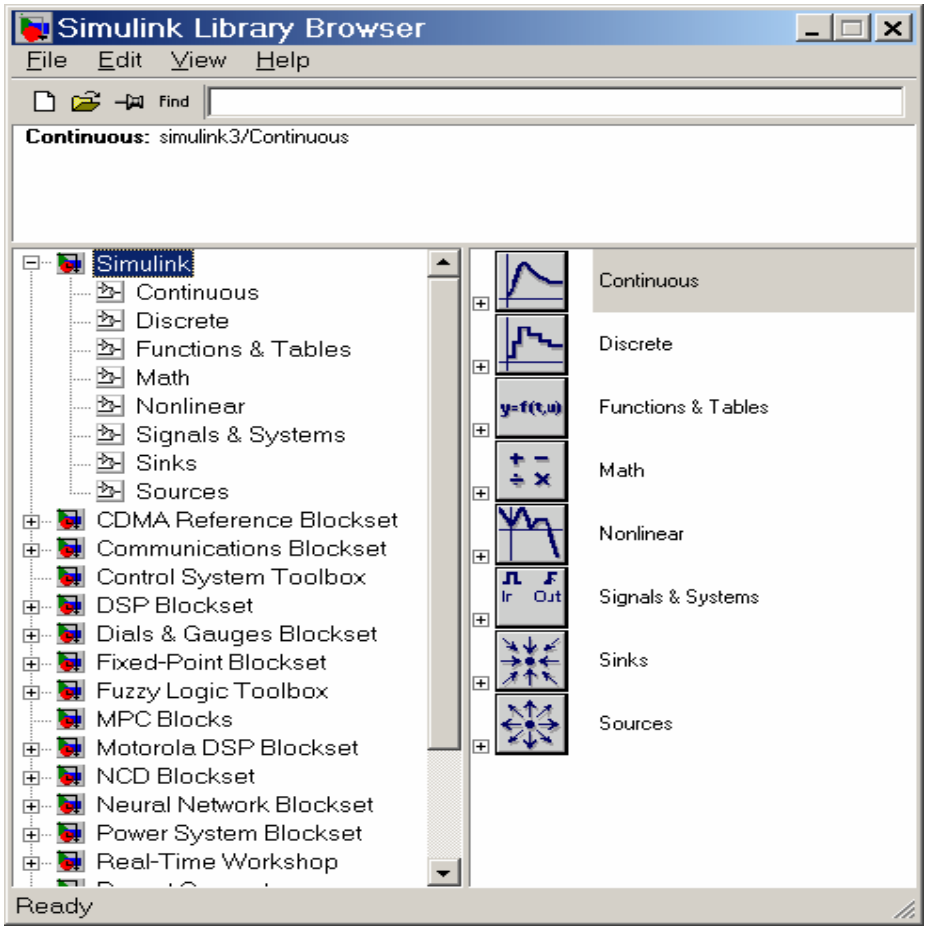


4. Для создания модели нужно выполнить действие File => New => Model. Это приводит к запуску программы Simulink, которая создает пустое окно модели.



5. Далее нужно вызвать браузер библиотеки компонент, используя меню или кнопку в панели инструментов Library Brouser. Окно браузера содержит две панели: слева иерархическое дерево библиотеки, справа - содержимое выбранной в левой панели

папки с блоками. В папке могут быть подбиблиотеки и блоки. Каждый блок и подбиблиотека имеют визуальный семантический образ и надпись.



6. Разместите окна браузера и модели таким образом, чтобы они не перекрывали друг друга. Теперь можно формировать модель визуальным методом.
7. Скопируйте мышью из браузера в окно модели нужные блоки и удобно разместите их. При переносе блока в модель там создается экземпляр блока с именем, совпадающим с надписью под блоком (при необходимости, когда однотипных блоков в модели несколько, в имя блока добавляется номер).
8. Соедините блоки коннекторами. Для этого нужно протаскивать мышь от одной соединяемой точки к другой. При отпускании кнопки мыши в модели отображается коннектор со стрелкой.

9. Установите для каждого блока свойства. Для этого нужно на блоке сделать двойной щелчок мышью, что приведет к появлению окна со свойствами блока. Установите нужные свойства в полях окна.

Пример выполнения

Задание

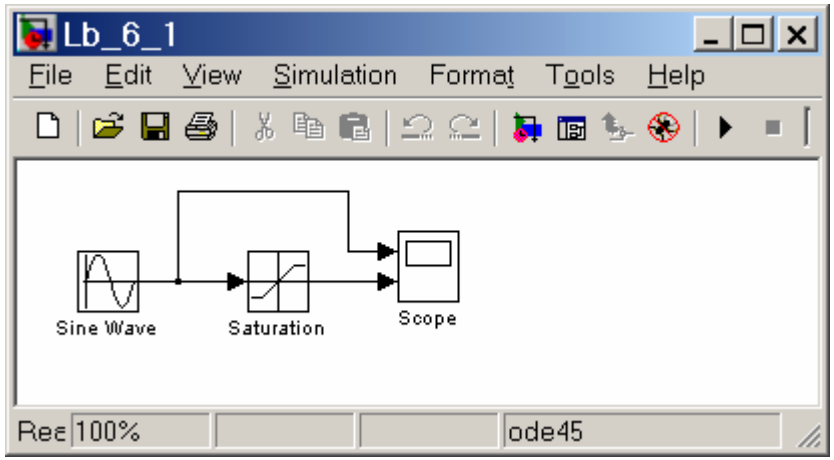
Задача 1. Двусторонний ограничитель синусоидального сигнала.
- Создать модель.
- Провести ее моделирование.

Задача 2. Двусторонний ограничитель синусоидального сигнала и блок мертвой зоны
- Создать модель.
- Провести ее моделирование.

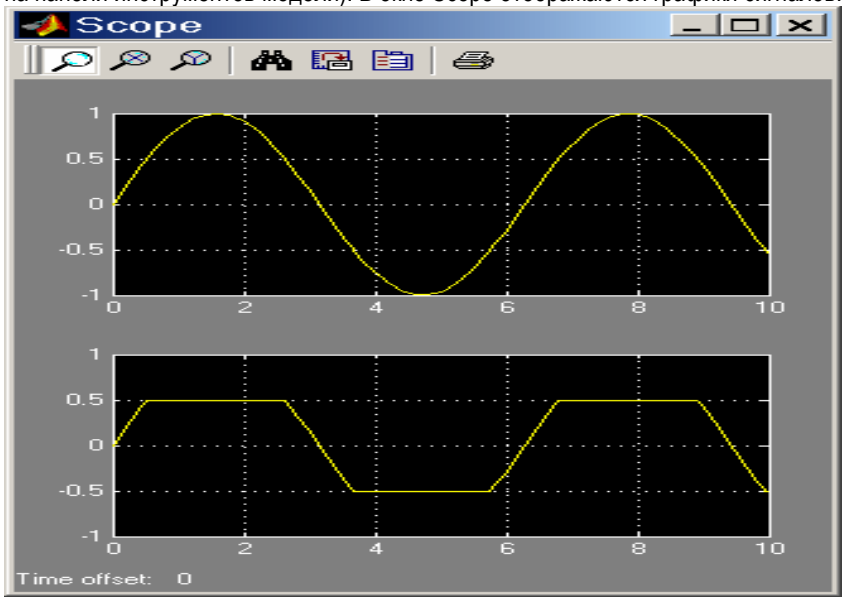
Задача 1

1. Создать на экране дисплея пустое окно модели и вызвать браузер библиотеки блоков.
2. Открыть в браузере папку с блоками источников, используя кнопку подбиблиотеки Sources (Источники). Из подбиблиотеки Sources левой кнопкой мыши перетащить в окно модели блок Sine Wave (генератор синусоиды) и там отпустить в удобном месте.
3. Двойным щелчком по блоку Sine Wave в модели вызвать окно со свойствами блока. В его полях выбрать параметры. В данном случае установить амплитуду и частоту (фазу и время отсчета можно не менять).
4. Открыть в браузере окно нелинейных блоков, используя кнопку подбиблиотеки Nonlinear (Нелинейные). Из подбиблиотеки Nonlinear левой кнопкой мыши перетащить в окно модели блок Saturation (ограничитель) и там отпустить в удобном месте.
5. Двойным щелчком по блоку Saturation в модели вызвать окно со свойствами блока. В нем установить верхний и нижний пределы ограничения.
6. Открыть в браузере окно блоков регистраторов, используя кнопку подбиблиотеки Sinks (Регистраторы). Из подбиблиотеки Sinks левой кнопкой мыши перетащить в окно модели блок Scope и там отпустить в удобном месте.
7. Двойным щелчком по блоку Scope в модели вызвать его демонстрационное окно. Разместить это окно на экране в удобном месте, перемещая его за заголовок левой кнопкой мыши.
8. Кнопкой Properties (Свойства) окна Scope вызвать окно свойств, в котором установить число осей 2 (для входного и выходного сигналов ограничителя).
9. Левой (или правой) кнопкой мыши соединить блоки. При нажатой левой кнопке курсор имеет форму крестика, который надо позиционировать по помеченным входам и выходам блоков. Начать надо с помеченного выхода одного блока и отпустить кнопку на помеченном входе другого. Первый вход регистратора соединить со входом ограничителя, второй - с выходом.

10. Результат - модель устройства и пустое окно регистратора.



11. Включить симулирование (моделирование) командой Simulation => Start (или кнопкой на панели инструментов модели). В окне Scope отображаются графики сигналов.



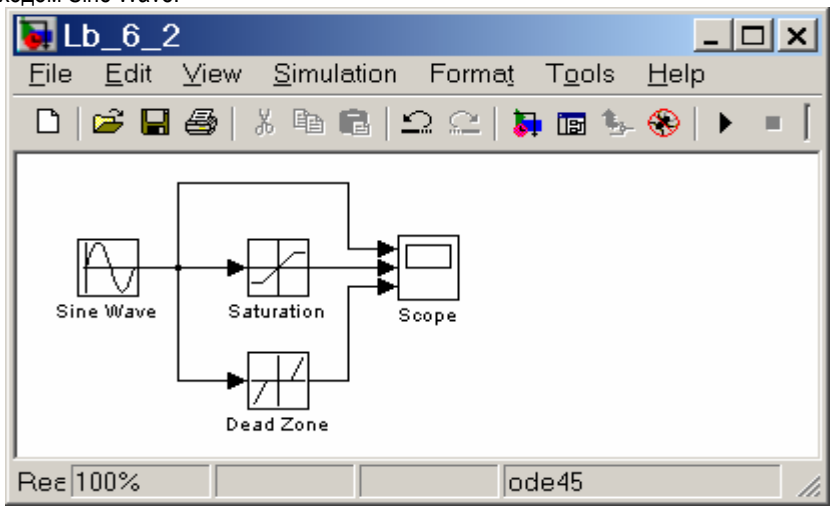
Если результат не совпадает с ожидаемым, то нужно изменить параметры модели. Команда Simulation => Simulation parameters вызывает окно параметров модели, в котором можно сделать изменения. Можно регулировать параметры, размещенные на 5 вкладках:

- Solver - решатель (время начала и конца, пошаговый или непрерывный режим, используемый математический метод, шаг моделирования, погрешности вычислений и др.).
- Workspace I/O - рабочая область (параметры загрузка и сохранения, опции сохранения).
- Diagnostics - диагностика (параметры и области, опции конфигурации, действия).
- Advanced - Расширения (признаки оптимизации).
- Real-Time Workshop - работа в реальном времени (выбор конфигурации целевого объекта и параметров для нее).

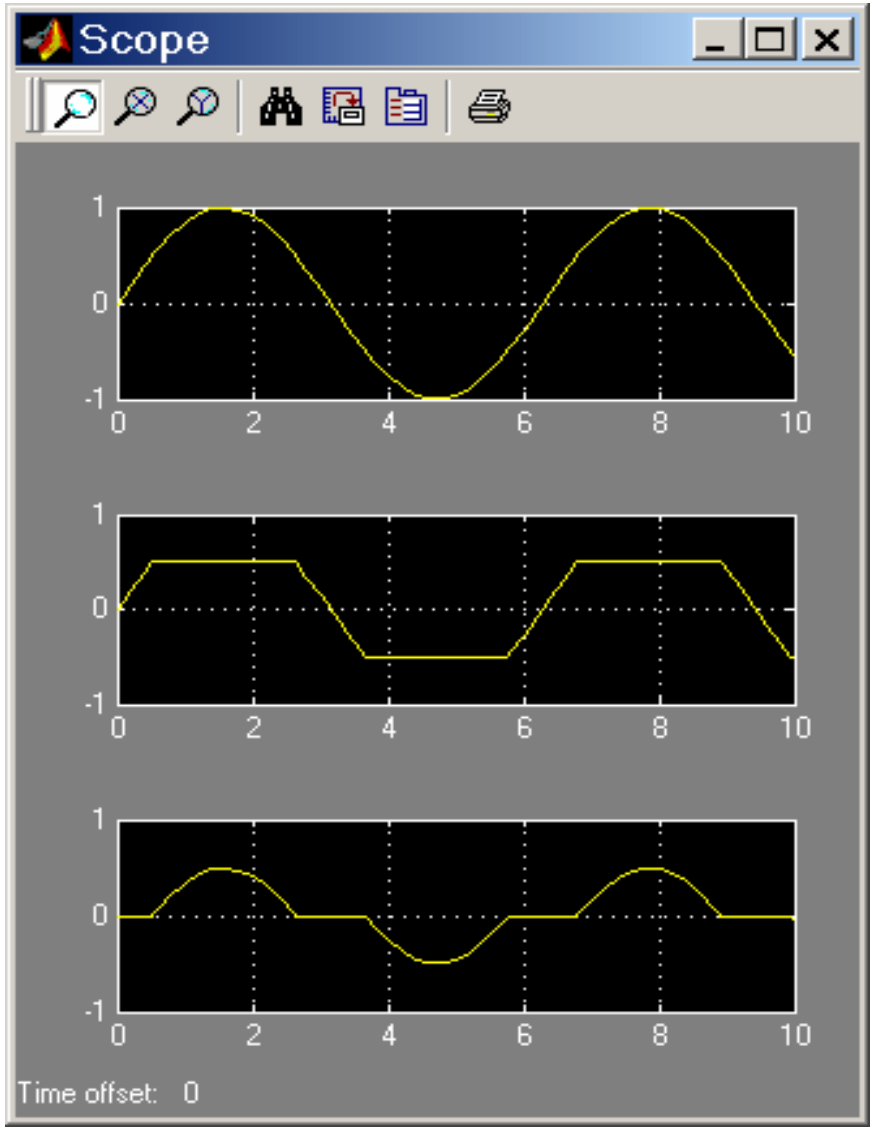
12. Если график получился в неудобном масштабе, то масштаб можно изменить. для этого лучше выбрать режим Autoscale (автомасштабирование) в локальном меню, вызываемым щелчком правой кнопки мыши по графику. Ниже видны результаты автомасштабирования.

Задача 2

1. Добавим в модель в параллельную ветвь блок Dead Zone (Мертвая зона). Установим его параметры.
2. У регистратора Scope изменим число осей на 3.
3. Соединим третий вход Scope с выходом Dead Zone. Соединим вход Dead Zone с выходом Sine Wave.



4. Выполнить моделирование. В окне Scope графики выходных сигналов.



7. Моделирование системы с помощью Simulink

Подготовка к работе

1. По указанной литературе изучить:
   - правила создания моделей систем в Simulink,
   - правила моделирования систем в Simulink,
   - иерархическую библиотеку Simulink.
   - состав Communication Blockset в Simulink.
2. Разработать структуру модели системы для варианта задания.

Контрольные вопросы

1. Назначение Simulink.
2. Правила построения моделей систем в Simulink.
3. Правила моделирования систем в Simulink.
4. Структура иерархической библиотеки Simulink.
5. Блоки из папки Sources библиотеки Simulink.
6. Блоки из папки Sinks библиотеки Simulink.
7. Блоки из Communication Blockset для обработки сигналов.
8. Блоки из Communication Blockset для генерации помех.

Задание к работе

Задача 1. Модель системы связи с заданными типами модуляции и помех:

- Создать модель. В ней сигнал от источника поступает на модулятор. Выходной сигнал модулятора передается в канал связи, где на него накладывается аддитивный шум. Выход канала связи поступает на демодулятор, восстанавливающий модулирующий сигналю. Регистратор с пятью входами позволяет наблюдать сигналы в разных точках системы.
- Провести ее моделирование.

Варианты заданий

№	Тип модуляции	Тип генератора помехи
1	DSB AM Passband Двухполосная амплитудная	Rician Noise Generator Райесовский шум
2	DSBSC AM Passband Балансная амплитудная	Rayleigh Noise Generator Релейевский шум
3	SSB AM Passband Однополосная амплитудная	Uniform Noise Generator Нормальный шум
4	FM Passband Частотная	Gaussian Noise Generator Гауссовский шум
5	PM Passband Фазовая	Rician Noise Generator Райесовский шум
6	DSB AM Passband Двухполосная амплитудная	Rayleigh Noise Generator Релейевский шум
7	DSBSC AM Passband Балансная амплитудная	Uniform Noise Generator Нормальный шум
8	SSB AM Passband Однополосная амплитудная	Gaussian Noise Generator Гауссовский шум
9	FM Passband Частотная	Rician Noise Generator Райесовский шум
10	PM Passband Фазовая	Rayleigh Noise Generator Релейевский шум
11	DSB AM Passband Двухполосная амплитудная	Uniform Noise Generator Нормальный шум
12	DSBSC AM Passband Балансная амплитудная	Gaussian Noise Generator Гауссовский шум
13	SSB AM Passband Однополосная амплитудная	Rician Noise Generator Райесовский шум
14	FM Passband Частотная	Rayleigh Noise Generator Релейевский шум
15	PM Passband Фазовая	Uniform Noise Generator Нормальный шум

Методические указания

1. Модель системы содержит источник сигнала и помехи, функциональные блоки и средства наблюдения за поведением системы (модулятор, демодулятор, дисплей, численный индикатор и др.).
2. Во всех вариантах задания нужно использовать дисплей с пятью входами.
3. Первое действие - запустить Matlab. При этом возникает стартовое диалоговое окно, в котором докированы три встроенных окна: Command Window (командное) - справа, Launch Pad (Средства запуска) - в левом верхнем углу, Command Hustory (История команд) - в левом нижнем углу. Каждое подокно можно освободить из дока.

4. Для создания модели нужно выполнить действие File => New => Model. Это приводит к запуску программы Simulink, которая создает пустое окно модели.
5. Далее нужно вызвать браузер библиотеки компонент, используя меню или кнопку в панели инструментов Library Brouser. Окно браузера содержит две панели: слева иерархическое дерево библиотеки, справа - содержимое выбранной в левой панели папки с блоками. В папке могут быть подбиблиотеки и блоки. Каждый блок и подбиблиотека имеют визуальный семантический образ и надпись.
6. Разместите окна браузера и модели таким образом, чтобы они не перекрывали друг друга. Теперь можно формировать модель визуальным методом.
7. Скопируйте мышью из браузера в окно модели нужные блоки и удобно разместите их. При переносе блока в модель там создается экземпляр блока с именем, совпадающим с надписью под блоком (при необходимости, когда однотипных блоков в модели несколько, в имя блока добавляется номер).
8. Соедините блоки коннекторами. Для этого нужно протаскивать мышь от одной соединяемой точки к другой. При отпускании кнопки мыши в модели отображается коннектор со стрелкой.
9. Установите для каждого блока свойства. Для этого нужно на блоке сделать двойной щелчок мышью, что приведет к появлению окна со свойствами блока. Установите нужные свойства в полях окна.

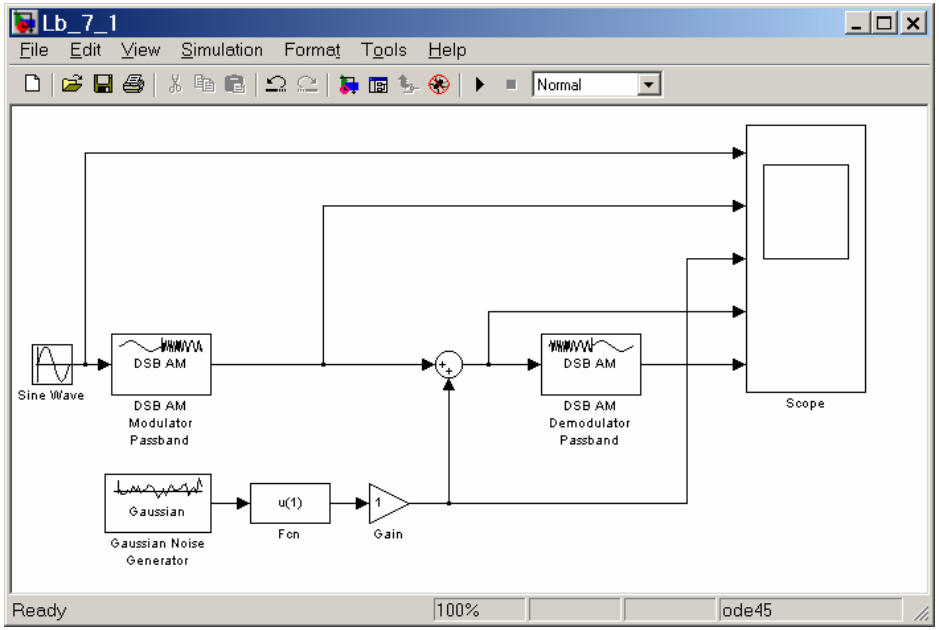
Пример выполнения

Задание. Создать модель аналоговой системы передачи с амплитудной модуляцией по каналу связи с гауссовым шумом.

Решение

1. Создать на экране дисплея пустое окно модели и вызвать браузер библиотеки блоков.
2. Открыть в браузере папку с блоками источников, используя кнопку подбиблиотеки Sources (Источники). Из подбиблиотеки Sources левой кнопкой мыши перетащить в окно модели блок Sine Wave (генератор синусоиды) и там отпустить в удобном месте.
3. Двойным щелчком по блоку Sine Wave в модели вызвать окно со свойствами блока. В его полях выбрать параметры. В данном случае установить амплитуду и частоту (фазу и время отсчета можно не менять).
4. Выбрать в браузере папку Communications Blockset (Коммуникационные блоки).
5. В нем открыть папку Modulation (Модуляция), а в ней папку Analog Passband Modulation (Аналоговая полосовая модуляция).
6. Из подбиблиотеки Analog Passband Modulation левой кнопкой мыши перетащить в окно модели блок DSB AM Modulator Passband (Модулятор двухполосной АМ с полосовым фильтром) и там отпустить в удобном месте.
7. Аналогично скопировать в окно модели блок DSB AM Demodulator Passband (Демодулятор двухполосной АМ с полосовым фильтром).

8. Двойным щелчком по блоку Saturation в модели вызвать окно со свойствами блока. В нем установить верхний и нижний пределы ограничения.
9. Выбрать в браузере папку Communications Blockset. В нем открыть папку Comm Sources (Коммуникационные источники).
10. Из подбиблиотеки Comm Sources левой кнопкой мыши перетащить в окно модели блок Gaussian Noise Generator (Генератор гауссового шума) и там отпустить в удобном месте.
11. Выбрать в браузере папку Simulink. В ней открыть папку Functions & Tables (Функции и тиаблицы).
12. Из подбиблиотеки Functions & Tables левой кнопкой мыши перетащить в окно модели блок Fcn (Функция) и там отпустить в удобном месте. Этот блок введет функцию преобразования u(1) векторного выхода блока Gaussian Noise Generator в скаляр. Для задания функции вызовите окно свойств блока двойным щелчком по нему и введите u(1) в поле функции.
13. Выбрать в браузере папку Simulink. В ней открыть папку Math (Математика).
14. Из подбиблиотеки Math левой кнопкой мыши перетащить в окно модели блок Gain (Усиление) и там отпустить в удобном месте. Этот блок будет использован для задания уровня шума в канале связи.
15. Из подбиблиотеки Math левой кнопкой мыши перетащить в окно модели блокSum (Сумматор) и там отпустить в удобном месте. Этот блок будет использован для сложение сигналов в канале связи.
16. .Выбрать в браузере папку Simulink. В ней открыть папку регистраторов Sinks (Регистраторы). Из подбиблиотеки Sinks левой кнопкой мыши перетащить в окно модели блок Scope и там отпустить в удобном месте.
17. Двойным щелчком по блоку Scope в модели вызвать его демонстрационное окно. Разместить это окно на экране в удобном месте, перемещая его за заголовок левой кнопкой мыши.
18. Кнопкой Properties (Свойства) окна Scope вызвать окно свойств, в котором установить число осей 5 (для модулирующего сигнала, модулированного сигнала, сигнала шума, сигнала на выходе канала связи, выходного сигнала демодулятора).
19. Левой (или правой) кнопкой мыши соединить блоки. При нажатой левой кнопке курсор имеет форму крестика, который надо позиционировать по помеченным входам и выходам блоков. Начать надо с помеченного выхода одного блока и отпустить кнопку на помеченном входе другого. Входы регистратора соединять с узлами модели в которых действуют сигналы, указанные выше и в том же порядке.
20. Результат - модель устройства и пустое окно регистратора.



21. Включить симулирование (моделирование) командой Simulation => Start (или кнопкой на панели инструментов модели). В окне Scope отображаются графики сигналов.
22. Осуществить моделирование для двух случаев
    - Помех нет. Для этого установить усиление блока Gain, равным нулю.
    - Помеха есть. Для этого установить усиление блока Gain, не равное нулю. Регулируя это значение можно проследить за поведение системы в разных условиях. Ниже приведены состояния регистратора без помех (слева) и с помехами (справа).

На приведенных графиках выходной сигнал демодулятора содержит остаточную высокочастотную компоненту. Это связано с использованием в демодуляторе не очень качественного фильтра.