Министерство образования и науки Российской Федерации

Федеральное агентство по образованию

Государственное образовательное учреждение высшего профессионального образования «Магнитогорский государственный технический университет им. Г.И. Носова»

Кафедра электроснабжения промышленных предприятий



КОМПЬЮТЕРНЫЙ ПРАКТИКУМ ПО ДИСЦИПЛИНЕ «НАДЕЖНОСТЬ ЭЛЕКТРОСНАБЖЕНИЯ»

Методические указания к практическим занятиям для студентов специальности 140211 «Электроснабжение»

> Магнитогорск 2010

Составитель: А.Н. Шеметов

Компьютерный практикум по дисциплине «Надежность электроснабжения»: Методические указания к практическим занятиям для студентов специальности 140211 «Электроснабжение». – Магнитогорск: ГОУ ВПО «МГТУ им. Г.И. Носова», 2010. - 20 с.

Рецензент Т.Р. Храмшин

© Шеметов А.Н., 2010.

введение

В данной работе приводятся основы работы с программой визуального программирования *Simulink* математического пакета *MATLAB*, необходимые для производства практических расчетов надежности электроснабжения, а также в ходе курсового и дипломного проектирования.

Программа Simulink является приложением к пакету MATLAB. При моделировании с использованием Simulink реализуется принцип визуального программирования, в соответствии с которым, пользователь на экране из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты. При этом, в отличие от классических способов моделирования, пользователю не нужно досконально изучать язык программирования и численные методы математики, а достаточно только общих знаний, требующихся при работе на компьютере, и, естественно, знаний той предметной области, в которой он работает.

В ходе моделирования имеется возможность следить за процессами, происходящими в системе. Для этого используются специальные устройства наблюдения, входящие в состав библиотеки Simulink. Результаты моделирования могут быть представлены в виде графиков или таблиц.

Simulink является достаточно самостоятельным инструментом MATLAB и при работе с ним совсем не требуется знать сам MATLAB и остальные его приложения. С другой стороны доступ к функциям MATLAB и другим его инструментам остается открытым и их можно использовать в Simulink. При работе с Simulink пользователь имеет возможность модернизировать библиотечные блоки, создавать свои собственные, а также составлять новые библиотеки блоков.

Преимущество Simulink заключается также в том, что он позволяет пополнять библиотеки блоков с помощью подпрограмм написанных как на языке MATLAB, так и на языках C + +, Fortran и Ada.

В силу ограниченного объема и узкого круга решаемых задач, предлагаемая разработка не претендует на всеобъемлемость материала и не может заменить учебников или самоучителей, в достаточном количестве доступных для широкого круга пользователей.

1. Знакомство с программой МАТLAB

В этой части мы познакомимся с программой для проведения расчетов *MATLAB*. Данная программа обеспечивает эффективное проведение серьезных численных и аналитических расчетов.

Входной язык программы является текстовым. В программе используется командный режим работы, реализованный с помощью строчного редактора, который позволяет изменять вводимую информацию <u>только в одной</u> <u>строке</u>. Оформление текста, например, изменение размера символов (как в *MS Word* или *Mathcad*), в этой программе не предусматривается.

Однако, по сравнению с *Mathcad*, программа *MATLAB* обладает существенно большей библиотекой численных методов и более высокой скоростью вычислений.

1.1. Запуск программы и знакомство с рабочим окном

Чтобы начать знакомство с программой, запустим MATLAB.

- Нажмите кнопку Пуск (*Start*) на Панели задач (*Taskbar*) операционной системы *Windows*. На экране появится основное меню.

- Выберите команду **Программы** « **Matlab** « **MATLAB** из основного меню. На экране появится рабочее окно программы MATLAB (Puc 1.1).



Рис. 1.1. Рабочий стол системы МАТLAB.

Заголовок основного окна содержит название программы. Ниже заголовка окна расположена полоса меню с основными командами для работы с программой MATLAB. Под полосой меню находится панель инструментов, которая предназначена для быстрого выполнения команд.

Рабочий стол появляется по умолчанию при запуске системы. Вы можете изменить состав и размещение рабочих окон, открывая новые, закрывая имеющиеся, а также изменяя их размеры с помощью вертикальных и горизонтальных разделителей. Кроме того, можно отделять окна от рабочего стола и помещать их на рабочий стол, используя меню *View* инструментальной панели.

На рис. 1.1 вы видите три открытых окна: командное окно *Command Window*, в котором расположена командная строка, окно запуска приложений *Launch Pad*, кнопка развертки-свертки и окно предыстории *Command History*.

Командное окно *Command Window* предназначено для непосредственной работы с переменными, файлами и функциями системы *MATLAB*. Здесь отображаются вводимые математические выражения и результаты вычислений. При первом запуске программы в рабочем поле выводится текст приветствия. Справа и внизу рабочего окна расположены вертикальная и горизонтальная полосы прокрутки, предназначенные для перемещения содержимого окна в вертикальном и горизонтальном направлениях.

Окно предыстории *Command History* содержит строки операторов, которые вводились в командном окне с начала сеанса работы. Окно запуска *Launch Pad* предназначено для вызова программных продуктов, включенных в состав инсталлированной версии системы *MATLAB*.

Кроме того, на инструментальной панели системы *MATLAB* имеется информационное окно *Current Directory*, с помощью которого всегда обеспечивается доступ к списку ранее вызванных текущих каталогов, что позволяет быстро переходить от одного каталога к другому, получая доступ к интересующим файлам.

На рис. 1.2 показано окно рабочей области *Workspace*, которое содержит список переменных, накопленных в памяти в процессе работы, тип данных, размер массива *Size* и занимаемый им объем памяти *Bytes*.

Workspace				× 5
🛩 🖬 🗊 🖻	Stack: Base	-		
Name	Size	Bytes	Class	
mx 🔛	1x560	4480	double array	
У	1x560	4480	double array	
∢ ▶ Istanting and	Current Dir	octory 1	ounch Rod	

Рис. 1.2. Окно рабочей области МАТLАВ.

После завершения сеанса работы данные в рабочей области не сохраняются, поэтому пользователь должен предусмотреть их сохранение в виде файла данных. Это реализуется либо с помощью опции *Save Workspace As...* из меню *File*, либо с использованием команды *Save*. Любая из этих операций приводит к созданию специального двоичного файла с расширением *.*mat*.

В дальнейшем, чтобы прочитать *МАТ*-файл и разместить его в рабочей области, следует воспользоваться опцией *Import Data...* или использовать команду *Load*.

1.2. Простейшие вычисления

Для начала рассмотрим основные приемы работы с программой *MATLAB*: как выполнять простейшие вычисления, вводить переменные, матрицы и комментарии.

Вычислим простое математическое выражение.

Наберите на клавиатуре выражение **9*45** и нажмите клавишу *Enter*. В окне программы появится результат вычислений (Рис. 1.3).

Обратите внимание, что ввод данных предваряется значком «. В переменной *ans* хранится результат последней операции. Если указана переменная со значением результатов вычислений, то имя переменной *ans* на экране не отображается.

- ∦ ⊳	IATLA	۱B						_[
File	Edit	⊻iew	We	ĒΖ	<u>N</u> indo	w į	<u>H</u> elp		
D	2	*		C.	K)	CH.		Î	?
>> :	9*45								
ans	=								
.	405								
>>									_
∟									
<u> 📣 5</u>	Start								
		Dree	1 2	D.		0.70			

Рис. 1.3. Вычисление простого выражения.

Теперь зададим имя переменной и присвоим ей значение.

- Введите имя переменной **x**, нажмите клавишу = чтобы начать присваивание значения переменной и введите число, которое нужно присвоить. Получим **x=2**. Нажмите клавишу *Enter*. Переменной *x* будет присвоено значение 1, а в окне программы появится значение переменной.

Если вывод каждого присвоенного значения на экран не нужен, запретим отображение результатов вычислений, указав в конце строки символ «;».

- Введите строку x=2; и нажмите клавишу <u>*Enter*</u>. Переменной *x* будет присвоено новое значение, но на экране ничего отображаться не будет.

- Выведем значение переменной на экран. Для этого введите имя переменной **x** и нажмите клавишу *Enter*. В окне программы появится значение переменной 2.

Чтобы присвоить переменной одномерный массив, выполните следующие шаги:

- Введите строку **A=[1 2 3]**;, чтобы присвоить переменной *A* вектор из трех элементов: 1, 2, и 3. Нажмите клавишу *Enter*. Переменной *A* будет присвоен введенный вектор. Обратите внимание, что элементы вектора разделяются пробелами, а весь список элементов заключается в квадратные скобки.

При создании матрицы необходимо иметь в виду, что ее строчки разделяются точкой с запятой, а элементы строк - пробелами. Попробуем ввести матрицу.

- Наберите на клавиатуре строку B=[1 2; 3 4], чтобы присвоить матрицу из двух строчек и двух столбцов.

- Нажмите клавишу *Enter*. В окне программы появится введенная матрица в своем естественном представлении (рис. 1.4).



Рис. 1.4. Ввод матрицы.

1.3. Операторы, специальные символы, переменные и константы

Арифметические операторы:

+ plus	- сложение;
- minus	- вычитание;
* mtimes	- умножение матриц;
.* times	- поэлементное умножение для массивов;
^ mpower	- возведение матрицы в степень;
.^ power	- возведение в степень для массивов;
.' transpose	- транспонирование элементов массива;
' ctranspose	- транспонирование элементов матрицы;
[,] horzcat	 объединение элементов в строку;
[;] vertcat	- объединение элементов в столбец;
(), { },. subsasgn	- присваивание подмассива;
(), { },. subsref	- ссылка на подмассив;
subsindex	- индекс подмассива.

Специальные переменные и константы:

ans pi	 Результат выполнения последней операции; π = 3,141592653589793e+000;
i, j	- мнимая единица, $\sqrt{-1}$;
inf	- бесконечное значение, 🚥 ;
NaN	- нечисловое значение.

Тригонометрические функции:

sin -	синус;
asin -	арксинус;
cos -	косинус;
acos -	арккосинус;
tan -	тангенс;
atan -	арктангенс;
cot -	котангенс;
acot -	арккотангенс.

Трансцендентные функции:

exp - экспоненциальная функция; log - функция натурального логарифм log10 - десятичный логарифм; log2 - логарифм по основанию 2; pow2 - экспонента по основанию 2; sqrt - функция квадратного корня.	ıa;
--	-----

Функции обработки комплексных чисел:

abs	 абсолютное значение комплексного числа;
angle	- аргумент комплексного числа;
conj	- комплексно-сопряженное число;
imag	- мнимая часть комплексного числа;
real	- действительная часть комплексного числа.

1.4. Операции с матрицами

Программный пакет *MATLAB* основан на работе с матрицами, причем даже простые числа рассматриваются как матрицы единичного размера. В этом опыте мы расскажем об основных операциях с матрицами.

Чтобы указать элемент матрицы, нужно в круглых скобках через запятую ввести индексы элемента. По умолчанию строки и столбцы в индексах нумеруются с единицы.

- Наберите на клавиатуре строку **B**(2,2), для указания элемента матрицы **B**, расположенного во второй строке и втором столбце. Нажмите клавишу *Enter*. В окне программы появится значение указанного элемента 4.

- Теперь присвоим элементу матрицы новое значение. Введите строку C(1,1)=2;, чтобы присвоить элементу матрицы C, расположенному в первой строке и первом столбце, число 2. Нажмите клавишу *Enter*. Указанный элемент матрицы примет новое значение (рис. 1.5).

При выполнении арифметических операций с матрицами, они записываются в привычном представлении. Например, попробуем перемножить две матрицы.

- Введите строку **D=B*C** для умножения матриц **B** и **C**. Нажмите клавишу *Enter*. В окне программы появится результат умножения матриц (рис. 1.5).

Иногда в математических расчетах нужно выполнять операции поэлементно, например, как в электронных таблицах, перемножить соответствующие элементы матриц. Для проведения поэлементных вычислений перед знаком арифметической операции нужно ставить точку.

Давайте поэлементно перемножим две матрицы.

- Введите строку **E=B**. *С для поэлементного умножения матриц **B** и С. Нажмите клавишу *Enter*. В окне программы появится результат вычислений (рис. 1.6). Обратите внимание, что результаты обычного и поэлементного умножения матриц отличаются.

Для вычисления суммы элементов матрицы используются операции **sum (A)** и **cumsum (A)**, причем вторая функция выводит вектор, полученный поэлементным сложением всех элементов «нарастающим итогом» (рис. 1.5):

Command Window	7 X
>> A=[1 2 3];	
>> C=cunsum(A)	
C =	
1 3 6	
>>	
	▶

Рис. 1.5. Работа функции ситвит.

	~			
. 4			۰.	
	L			
	•		,	
	-	,		

AMATLAB	- D × 🛃	MATLAB		📣 MATLAB	
<u>File E</u> dit <u>V</u> iew We <u>b</u> <u>W</u> indow <u>H</u> elp	Eil	ile <u>E</u> dit <u>V</u> iew We <u>b W</u> indow	Help	<u>File E</u> dit <u>V</u> iew We <u>b</u>	<u>W</u> indow <u>H</u> elp
🗋 🖻 🕺 🖓 🖬 🛍 ၊ର ରା 🎁	? [) 🛩 X 🖻 🛍 🗠 🤉	× 🎁 ?	🗅 🗃 🕹 🖌 🖻 🖷	ມ ທ (🎁 🛛 ? 🗍
>> B=[1 2; 3 4]	>>	> B(2,2)		>> D=B*C	
В =	an	ns =		D =	
1 2		4		16 22	
3 4	>>> >>>	> C(1,1)=2		34 50	
>> C=[5 6; 7 8]	с	=		>> E=B.*C	
C =		2 6		E =	
5 6		7 8		2 12	
>>>	>>>	>		21 32	
		1	_	>>	
				A Stort	<u> </u>
4 Start		» start		- Start	

Рис. 1.6. Операции с матрицами.

В этом опыте мы рассмотрим, как строить графики и диаграммы. Сначала построим график функции.

- Введите строку **x=0.1:0.01:10**;, чтобы определить диапазон изменения аргумента на графике функции. Нажмите клавишу *Enter*.

- Наберите на клавиатуре строку **y=sin(x)**;, чтобы присвоить переменной **y** вектор значений функции **sin** по всем элементам **x** и нажмите клавишу *Enter* для завершения присваивания.

- Введите строку **plot(x,y)**;. Встроенная функция **plot** строит график функции **y(x)** в декартовой системе координат. Нажмите клавишу *Enter*. На экране появится новое окно с графиком (рис. 1.7, а), на горизонтальной и вертикальной осях которого отображаются диапазоны изменения аргумента и функции.

Чтобы закрыть окно графика, щелкните мышью на кнопке 🗵 в правом верхнем углу окна.



Рис. 1.7. Окно построения графика функции (а) и диаграммы массива данных (б).

Теперь построим диаграмму, отображающую массив данных.

- Введите строку **x**=[1 4 2 3];, чтобы ввести вектор, по которому будем строить диаграмму. Нажмите клавишу *Enter*.

- Наберите на клавиатуре строку **bar** (**x**) ;. Встроенная функция **bar** строит диаграмму, столбики которой определяются элементами вектора **x**. Нажмите клавишу *Enter*. На экране появится окно с диаграммой вектора (Рис. 1.7, б).

Для построения графика в полярной системе координат следует воспользоваться встроенной функцией **polar**.

1.6. Статистическая обработка данных

Исходные данные для статистической обработки в *MATLAB* задаются в виде матриц. Самый простой анализ данных, содержащихся в массиве, заключается в поиске его элементов с максимальным и минимальным значениями:

тах (A) – возвращает наибольший элемент, если **A** - вектор; или возвращает вектор-строку, содержащую максимальные элементы каждого столбца, если **A** – матрица;

min (A) – возвращает минимальный элемент, если **A** – вектор; или возвращает вектор-строку, содержащую минимальные элементы каждого столбца, если **A** – матрица (рис. 1.7, а).

Более полная статистическая обработка данных обычно сводится к нахождению их среднего значения, медианы (срединного значения) и стандартного отклонения. Для этого в системе *MATLAB* предусмотрены следующие встроенные функции:

mean (A) – определяет среднее арифметическое значение элементов массива, если **A** – вектор; или формирует вектор-строку, содержащую средние значения элементов каждого столбца, если **A** - двумерная матрица;

median (A) – возвращает медиану или вектор-строку медиан для каждого столбца матрицы **A**;

std (A) – определяет стандартное отклонение элементов массива.

A с помощью функции **hist** можно построить гистограмму, которая характеризует число попаданий элементов указанного вектора в определенные интервалы.

📣 M	ATL/	AB							- 10	1)	×
File	<u>E</u> dit	⊻iew	We	eb (<u>W</u> indo	w	Help	2			
D	2	¥	Ē	ß	K)	Ся		1	•	?	
a =											7
	1	á	2	3		4		5			
>> 1	ıax (a)									
ans	=										
I	6										
L											
>> 1	ain (a)									
ans	=										
	1										
>> 1	iean	(a)									
ans	=										
	3.5	000									
	-+4/	- 1									
/~ °	scal	a)									
ans	=										
	1.8	708									
┫										ſ	
<u> 📣 S</u>	start										

Рис. 1.6. Статистическая обработка данных.

1.6. Основы программирования в МАТLАВ. М-файлы.

До сих пор мы в основном использовали систему *MATLAB* в командном режиме непосредственного счета. Однако для решения серьезных задач возникает необходимость сохранения алгоритмов в виде программных кодов. Файлы, которые содержат коды языка *MATLAB*, называются М-файлами.

Для создания М-файла используется любой текстовый редактор или встроенный отладчик *MATLAB*, в котором текст программы набирается с использованием стандартных операций и команд.

Открыть редактор можно двумя способами:

a) из меню File выбрать опцию New, а затем M-File\$

б) использовать команду редактирования *edit* в рабочем окне *MATLAB*.

Например, команда *edit abcde* запускает редактор и открывает файл *abcde.m*.

Имя файла, содержащего М-функцию, составляется из имени функции и расширения ".m".

В системе MATLAB на имена М-функций налагаются те же ограничения, что и на имена переменных:

- их длина не должна превышать 31 символа;

- имена М-функций должны начинаться с буквы;

- остальные символы могут быть любой комбинацией букв, цифр и подчеркиваний.

В первой строке М-файла указывается его имя и (если есть) входные переменные. Например:

function f = fact (n)

Тело файла – текст программы, который и реализует вычисления над входными переменными и присваивает значения выходным аргументам. При необходимости текст дополняется комментариями, которые пишутся с новой строки и начинаются знаком «%». Причем первая строка комментария определяет назначение функции. Например:

```
function f = fact (n)
% FACT Вычисление факториала.
% fact(n) возвращает n! - факториал числа n
% Вычислить fact (n) = prod(1:n).
f = prod(1:n);
```

Как только такой файл создан, можно выполнить следующие действия: - вывести на экран имена файлов текущего каталога – команда **what**;

- вывести на экран текст М-файла – команда type ABCD1

- вызвать из файла функцию с заданными параметрами, например fact(5),

и получить результат вычислений ans= 120

2. Знакомство с приложением Simulink

После открытия основного окна пакета *MATLAB* нужно запустить программу *Simulink*. Это можно сделать одним из трех способов:

а) нажать кнопку \square (*Simulink*) на панели инструментов командного окна *MATLAB*;

б) в командной строке главного окна *MATLAB* напечатать *Simulink* и нажать клавишу *Enter* на клавиатуре;

в) выполнить команду *Open...* в меню *File* и открыть готовый файл модели с расширением ***.mdl**.

Использование первого и второго способов приводит к открытию окна обозревателя разделов библиотеки *Simulink* (рис. 2.1, а).



Рис 2.2. Окно обозревателя разделов библиотеки Simulink.

Список разделов библиотеки *Simulink* представлен в виде дерева. На рис. 2.1,а в левой части окна выделена основная библиотека, а в правой части – показаны ее разделы. При выборе соответствующего раздела библиотеки в правой части окна отображается его содержимое (рис. 2.1, б).

- *Continuous* линейные блоки.
- Discrete дискретные блоки.
- Functions & Tables функции и таблицы.
- Math блоки математических операций.
- Nonlinear нелинейные блоки.
- Signals & Systems сигналы и системы.
- Sinks регистрирующие устройства.
- Sources источники сигналов и воздействий.
- Subsystems блоки подсистем.

Для того чтобы развернуть или свернуть узел дерева, достаточно щелкнуть на его пиктограмме левой клавишей мыши.

2.1. Создание модели

Для создания модели в среде *Simulink* необходимо последовательно выполнить ряд действий:

1) Создать новый файл модели с помощью команды *File/New/Model*, или используя кнопку 🗅 на панели инструментов. Откроется пустое окно модели (рис. 2.2).



Рис 2.2. Пустое окно модели.

2) Расположить в окне модели необходимые блоки. Для этого – открыть соответствующий раздел библиотеки (например, *Sources*) и, указав курсором на требуемый блок, нажать и удерживая левую клавишу мыши, «перетащить» блок в созданное окно (рис 2.3).



Рис 2.3. Окно модели, содержащее блоки.

1	1
	n
	v

Для удаления блока необходимо выбрать блок (указать курсором на его изображение и нажать левую клавишу «мыши»), а затем нажать клавишу **Delete** на клавиатуре.

Для изменения размеров блока требуется выбрать блок, установить курсор в один из углов блока и, нажав левую клавишу «мыши», изменить размер блока (курсор при этом превратится в двухстороннюю стрелку).

3) Далее, если это требуется, можно изменить параметры блока, установленные программой «по умолчанию». Для этого необходимо дважды щелкнуть левой клавишей «мыши» на изображении блока. Откроется окно редактирования параметров (рис. 2.4). При задании численных параметров следует иметь в виду, что в качестве десятичного разделителя должна использоваться точка, а не запятая. После внесения изменений нужно закрыть окно кнопкой **OK**.

The second state of the se		
Noticiea	Block Parameters: Transfer Fcn	×
<u>File Edit View Simulation Forma</u>	- Transfer Fon	
	Matrix expression for numerator, vector expression for denominator. Output width equals the number of rows in the numerator. Coefficient are for descending powers of s.	nts
Transfer Fon	Parameters Numerator:	-
Ready	Denominator:	
	[1 1]	
	Absolute tolerance:	
	auto	
	OK Cancel Help Apply	

Рис 2.4. Блок, моделирующий передаточную функцию и окно редактирования его параметров.

4) После установки на схеме всех блоков из требуемых библиотек нужно выполнить соединение элементов схемы. Для этого необходимо указать курсором на «выход» первого блока, а затем, нажать и, не отпуская левую клавишу «мыши», провести линию к входу другого блока. После чего отпустить клавишу. В случае правильного соединения изображение стрелки на входе блока изменяет цвет (рис. 2.5).

👹 untitled *	
<u>E</u> ile <u>E</u> dit ⊻iew	Simulation Format Tools Help
🗅 🚔 🖬	🚳 🕹 🗈 🗠 🕩 🔳 Normal 💽 🛛 🖶 🕼 🖡 🖬 🕆 🛞
	Constant Transfer Fon
Ready	100% ode45
	D 25 E

Рис 2.5. Готовая схема модели.

- 1	_
	_ /
_	

Для создания **точки разветвления** в соединительной линии нужно подвести курсор к предполагаемому узлу и, нажав **правую** клавишу «мыши», протянуть линию. Для удаления линии требуется выбрать линию (так же, как это выполняется для блока) и нажать клавишу **Delete** на клавиатуре.

5) Для повышения наглядности модели удобно использовать текстовые надписи. Для создания надписи нужно курсором указать в окне модели место надписи и дважды щелкнуть левой клавишей мыши. После этого появится прямоугольная рамка с курсором ввода. Аналогичным образом можно изменить и подписи к блоками моделей.

6) После составления расчетной схемы необходимо сохранить ее в виде файла на диске, выбрав пункт меню *File/Save As...* в окне схемы и указав папку и имя файла. Следует иметь в виду, что имя файла не должно превышать 32 символов, должно начинаться с буквы и не может содержать символы кириллицы и спецсимволы. Это же требование относится и к пути файла (к тем папкам, в которых сохраняется файл).

Для последующего редактирования схемы можно пользоваться пунктом меню *Fille/Save*.

При повторных запусках программы загрузка схемы осуществляется с помощью меню *File/Open...* в окне обозревателя библиотеки или из основно-го окна *MATLAB*.

2.2. Выполнение расчета

Запуск расчета выполняется с помощью выбора пункта меню *Simulation/Start* или пиктограммой на панели инструментов. В ходе выполнения процесс расчета можно приостановить (*Simulation/Pause*), а затем продолжить (*Simulation/Continue*).

Для досрочного завершения расчета используется команда меню *Simulation/Stop* или пиктограмму на панели инструментов.

Результаты моделирования процесса отображаются с помощью блоков-приемников сигналов (*Sinks*):

- цифровой дисплей *Display* отображает значение сигнала в виде числа;

- осциллограф *Scope* строит графики исследуемых сигналов в функции времени и позволяет наблюдать за изменениями сигналов в процессе моделирования. Для того, чтобы открыть окно просмотра сигналов необходимо выполнить двойной щелчок левой клавишей "мыши" на изображении блока.

- графопостроитель **XУ Graph** строит график вида **Y(X)** одного сигнала в функции другого. Блок имеет два входа: верхний предназначен для подачи сигнала, который является аргументом (**X**), нижний – для подачи значений функции (**Y**).



Рис. 2.6. Применение блока *Display* с использованием различных форматов отображения данных (*Format*).



Рис. 2.7. Изображение блока *Scope* и окно для просмотра графиков.



Рис. 2.8. Примеры использования графопостроителя.



Библиографический список

1. Дьяконов В.П. МАТLАВ: учебный курс. – СПб.: Питер, 2001. – 560 с.

2. Черных И.В. Simulink: Среда создания инженерных приложений. - М.:

Диалог-Мифи, 2004. – 496 с.
 3. Компьютер для студентов, аспирантов и преподавателей. Самоучитель.
 – М.: Изд-во ТРИУМФ, 2001. – 656 с.