

Министерство образования Республики Беларусь  
БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

---

Кафедра «Интеллектуальные системы»

**Моделирование каналов передачи информации  
в среде MATLAB+SIMULINK**

Учебно-методическое пособие  
для студентов специальности  
1-55.01.02 «Интегральные сенсорные системы»

Минск 2012

УДК 621.37:621.391

Составители и разработчики **Баркалин В.В., Чашинский А.С., Глушак Д.А.**

Учебно-методическое пособие «Моделирование каналов передачи информации в среде MATLAB+SIMULINK» разработано в соответствии с образовательными стандартами высшего образования Республики Беларусь по специальности 1-55.01.02 «Интегральные сенсорные системы» и утвержденной программой курса «Физика каналов передачи информации и теория связи».

Учебное пособие предназначено для развития у студентов навыков использования среды моделирования MATLAB для решения технических задач. В пособии описаны высокоуровневый интерпретируемый язык программирования, инструменты для символьных вычислений, функции обработки сигналов, а также методы создания и визуализации моделей.

Одобрено заседанием кафедры «Интеллектуальные системы».

Белорусский национальный технический университет  
пр-т Независимости, 65, г. Минск, Республика Беларусь  
Тел.(017)292-77-52 факс (017)292-91-37  
E-mail: is@bntu.by  
Регистрационный № БНТУ/МСФ82-26.2012

© Баркалин В.В., Чашинский А.С., Глушак Д.А. 2012  
© Глушак Д.А., компьютерный дизайн, 2012  
© БНТУ, 2012

## СОДЕРЖАНИЕ

	Стр.
Содержание .....	3
1 Состав и интерфейс системы программирования MATLAB. Работа в командном окне MATLAB .....	4
2 Лабораторная работа №1. Работа с матрицами в системе MATLAB. Работа с графическими командами и функциями системы MATLAB .....	12
3 Лабораторная работа №2. Работа с пакетом расширения Symbolic Math системы MATLAB .....	16
4 Лабораторная работа №3. Программирование в среде MATLAB: операторы и массивы.....	27
5 Лабораторная работа №4. Знакомство с пакетом Simulink системы MATLAB.....	36
6 Лабораторная работа №5. Разработка модели бинарного симметричного канала связи с помехами .....	39
7 Лабораторная работа №6. Разработка моделей амплитудной, частотной и фазовой модуляции несущей в пакете расширения Simulink системы MATLAB .....	44
8 Лабораторная работа №7. Преобразование Фурье в системе MATLAB .....	49
9 Лабораторная работа №8. Разработка генераторов сигналов в пакете расширения Simulink системы MATLAB .....	53
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ .....	57

# 1. СОСТАВ И ИНТЕРФЕЙС СИСТЕМЫ ПРОГРАММИРОВАНИЯ MATLAB. РАБОТА В КОМАНДНОМ ОКНЕ MATLAB

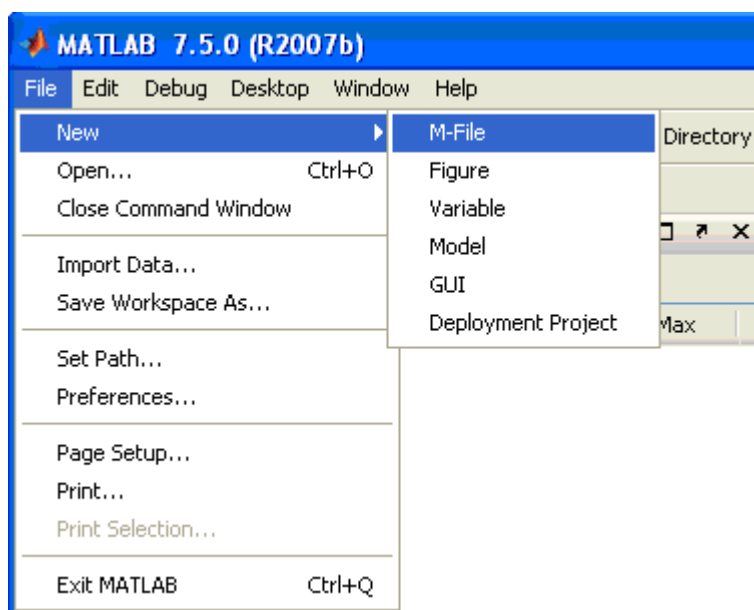
**Цель работы:** ознакомиться с составом и интерфейсом системы программирования MATLAB. Изучить синтаксис команд языка MATLAB и работу с ними.

Работа рассчитана на 2 академических часа

## Теоретические сведения

### *Командное окно. Инструментальная панель*

Командное окно системы MATLAB показано на рисунке 1.1. Здесь же показано ниспадающее меню File.



**Рисунок 1.1 – Командное окно MATLAB**

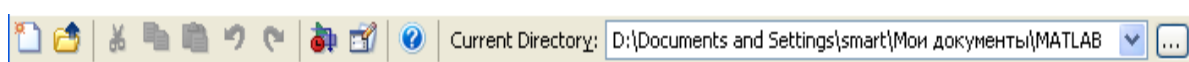
В таблице 1.1 приведены опции меню File.

Таблица 1.1 – Опции Меню File

<i>Опция</i>	<i>Назначение</i>
New	Открыть в редакторе/отладчике новый файл. Открыть графическое окно
Open	Открыть в редакторе/отладчике указанный файл
Close Command Window	Заккрыть командное окно
Import Data	Вызов окна загрузки данных Recognized Data Files
Save Workspace As	Вызов окна сохранения MAT-файла
Set Path	Вызов средства просмотра путей доступа Path Browser
Preferences	Выбор характеристик
Print Setup	Установка опций принтера
Print	Установка опций вывода на печать
Print Selection	Печать выделенного фрагмента

**И  
нс  
т  
ру  
ме  
н  
т  
ал  
ьн  
ая  
па  
не  
ль**

Инструментальная панель (рисунок 1.2) командного окна системы MATLAB позволяет обеспечить простой доступ к операциям над М-файлами.



**Рисунок 1.2 – Инструментальная панель MATLAB**

Эти операции включают:

- создание нового М-файла (New File);
- открытие существующего М-файла (Open File);
- удаление фрагмента (Cut);
- копирование фрагмента (Copy);
- вставка фрагмента (Paste);
- восстановление только выполненной операции (Undo);
- вызов Simulink Library Browser (Simulink)
- вызов окна GUIDE Quick Start (GUIDE)
- просмотр путей доступа (Path Browser);
- текущая помощь (Help).

### ***Рабочая область***

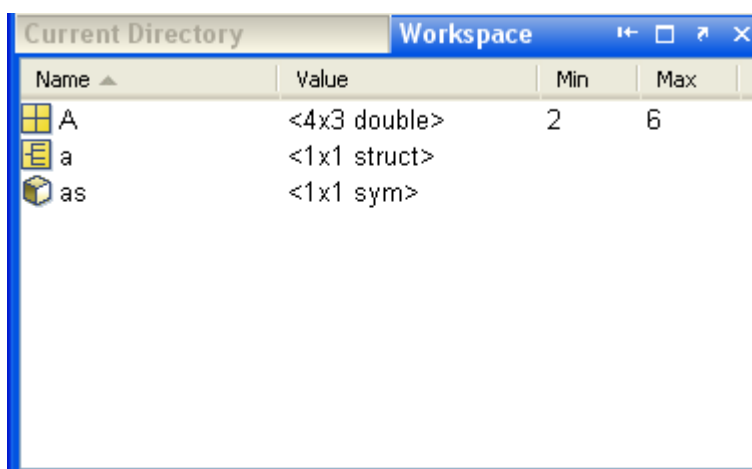
*Рабочая область* системы MATLAB - это область памяти, в которой размещены переменные системы. Содержимое этой области можно просмотреть из командной строки с помощью команд `who` и `whos`. Команда `who` выводит только имена переменных, а команда `whos` - информацию о размерах массивов и типе переменной.

Рассмотрим в качестве примера 5 массивов различного типа:

- `A` – трехмерный массив чисел удвоенной точности;
- `B` – массив разреженной структуры;
- `C` – массив ячеек;
- `S` – массив символов;
- `patient` – массив записей.

Специальное средство просмотра `Workspace Browser` обеспечивает представление команды `whos` в виде графического интерфейса. Для того чтобы открыть `Workspace Browser` надо либо выбрать опцию `Show Workspace` из меню `File menu`, либо воспользоваться кнопкой `Workspace Browser` инструментальной панели.

В результате этих операций на терминал будет выведено следующее окно (рисунок 1.3).



**Рисунок 1.3 – Рабочая область MATLAB**

В этом окне можно выполнить следующие операции:

- удалить переменную, если выделить ее и нажать кнопку Delete;
- закрыть окно с помощью кнопки Close.

Кроме того, можно изменять размеры колонок посредством перемещения их границ с помощью мыши. Можно выполнить переименование переменной, если сначала выделить ее, затем однократно щелкнуть левой клавишей мыши (заметим, что двойной щелчок никакого действия не оказывает). После короткой задержки появляется поле, в котором можно указать новое имя; и наконец, следует нажать клавишу Enter, чтобы подтвердить завершение операции.

**Загрузка и сохранение рабочей области**

Команды `save` и `load` позволяют в любой момент времени сохранить содержимое рабочей области или загрузить новые данные в процессе сеанса работы.

**Список путей доступа.** Для поиска М-файлов система MATLAB использует механизм путей доступа, поскольку М-файлы записываются в каталоги или папки файловой системы.

Например, при поиске файла с именем *smart* MATLAB выполняет следующие действия:

- просматривает, не является ли *smart* именем переменной;
- просматривает, не является ли *smart* встроенной функцией;
- ищет в текущем каталоге М-файл с именем *smart.m*;
- ищет М-файл с именем *smart.m* во всех каталогах списка путей доступа.

**Работа со списком путей доступа.** В процессе сеанса работы можно вывести на терминал или внести изменения в список путей доступа, используя следующие функции:

- `path` выводит на экран списка путей доступа;
- `path(s)` заменяет существующий список списком `s`;
- `addpath /home/lib` и `path(path, '/home/lib')`-- добавляют новый каталог в список путей доступа;
- `rmpath /home/lib`-- удаляет путь `/home/lib` из списка.

Список путей доступа, используемый по умолчанию, определен в файле `pathdef.m`, который размещен в каталоге `local`; этот файл выполняется при каждом запуске системы MATLAB.

Кроме работы из командной строки существует средство просмотра путей доступа `Path Browser`, которое поддерживает удобный графический интерфейс для просмотра и изменения списка путей. Однако более предпочтительно вносить непосредственные изменения в М-файл `pathdef.m`, используя какой-либо текстовый редактор, в том числе и редактор/отладчик системы MATLAB.

**Текущий каталог.** Система MATLAB использует понятие текущего каталога при работе с М- и МТ-файлами во время сеанса работы. Начальный текущий каталог определен в файле запуска, который ассоциирован с ярлыком запуска системы MATLAB, расположенном на рабочем столе. Щелчок правой кнопки мыши, установленной на этом ярлыке, и выбор опции `Properties` позволяет изменить начальный каталог, используемый по умолчанию. Для вывода текущего каталога на экран терминала предназначена команда `cd`. Для изменения текущего каталога следует использовать команду `cd <новый путь доступа>`.



Для редактирования М-файла используется команда edit <имя файла>.

### ***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
/ mrdivide	Правое деление матриц
./ rdivide	Правое деление для массивов

### ***Операторы отношения***

==	eq	Тождественно
~=	ne	Не тождественно
<	lt	Меньше
>	gt	Больше
<=	le	Меньше или равно
>=	ge	Больше или равно

### ***Логические операторы***

&	and	Логическое И
	or	Логическое ИЛИ
~	not	Логическое НЕТ
xor		Логическое ИСКЛЮЧИТЕЛЬНОЕ ИЛИ
any		Истинно, если хотя бы 1 элемент вектора не равен нулю
all		Истинно, если все элементы вектора не равны нулю

### ***Специальные символы***

:	Сечение массива
()	Указание последовательности выполнения операций
[]	Формирование массива
{}	Многомерные массивы
.	Десятичная точка (разделитель)
...	Продолжение строки
;	Подавление вывода эхо-результата
%	Комментарий
!	Вызов команды операционной системы
=	Присваивание
'	Кавычка
.' transpose	Транспонирование элементов массива
' ctranspose	Транспонирование элементов матрицы

### ***Специальные переменные и константы***

pi	3.141592653589793e+000
i, j	Мнимая единица, $\sqrt{-1}$
inf	Бесконечное значение, $\infty$
NaN	Нечисловое значение

### ***Справочные команды***

help	Текущая справка в командной строке
demo	Демонстрационные примеры

### ***Управление рабочей областью***

Who	Список текущих переменных
whos	Список текущих переменных с подробностями
clear	Удаление переменных и функций из памяти
Load	Считывание переменных из MAT-файла
Save	Запись переменных в MAT-файл
Quit	Завершение работы в системе MATLAB

### ***Управление командами и функциями***

what	Список файлов в текущем каталоге
Type	Просмотр текста M-файла
Edit	Редактирование текста M-файла
lookfor	Поиск M-файлов по ключу
which	Месторасположение функций и файлов

### ***Управление путями доступа***

Path	Определить/установить путь доступа
addpath	Добавить каталог к пути доступа
rmpath	Удалить каталог из пути доступа
editpath	Отредактировать путь доступа

### **Контрольные вопросы:**

1. Что выполняют данные справочные команды и функции: DEMO, INFO, VER, HELP, LOOKFOR, WHAT, WHICH, PATH, QUIT?
2. Какие команды работают с рабочей областью?
3. Какие форматы вывода числовой информации существуют в MATLAB?
4. Какие арифметические и логические операторы представлены в MATLAB?
5. Что именно обозначают данные специальные символы, переменные и константы: [ ] ( ) =, ; : %, ANS, i, j, INF, NaN, PI?

## 2. ЛАБОРАТОРНАЯ РАБОТА №1. РАБОТА С МАТРИЦАМИ В СИСТЕМЕ MATLAB. РАБОТА С ГРАФИЧЕСКИМИ КОМАНДАМИ И ФУНКЦИЯМИ СИСТЕМЫ MATLAB

**Цель работы:** ознакомиться с работой с матрицами в системе MATLAB. Изучить графические возможности пакета MATLAB.

Работа рассчитана на 4 академических часа

### Теоретические сведения

Подробные описания команд и примеры их использования можно получить при вводе команды подсказки `help <имя_функции>`

### *Математические функции Тригонометрические функции*

Sin	Синус
Sinh	Гиперболический синус
Asin	Арксинус
tan	Тангенс
Atan	Арктангенс

### *Математические функции Трансцендентные функции*

Exp	Экспоненциальная функция
log	Функция натурального логарифма
log10	Логарифм по основанию 10
log2	Логарифм по основанию 2
Pow2	Экспонента по основанию 2
Sqrt	Функция квадратного корня

### ***Математические функции Функции обработки комплексных чисел***

abs	Абсолютное значение комплексного числа
conj	Комплексно-сопряженное число
imag	Мнимая часть комплексного числа
real	Действительная часть комплексного числа

### ***Элементарная графика Двумерные графики***

plot	График в линейном масштабе
polar	График в полярных координатах

### ***Элементарная графика Трехмерные графики***

plot3	Построение линий и точек в трехмерном пространстве
contour	Изображение линий уровня для трехмерной поверхности
meshgrid	Формирование двумерных массивов X и Y
mesh	Трехмерная сетчатая поверхность
surf	Затененная сетчатая поверхность

### ***Элементарная графика Задание осей координат***

axis	Масштабирование и вывод осей координат
grid	Управление выводом сетки
subplot	Разбиение графического окна

### ***Линейная алгебра Матричный анализ***

rank	Вычисление ранга матрицы
det	Вычисление определителя матрицы

## *Линейная алгебра Решение систем линейных уравнений*

inv	Вычисление обратной матрицы
eig	Полная проблема собственных значений

### **Задания для выполнения**

#### ***Вариант 1***

1. Задать матрицу  $A$   $3 \times 3$  и столбец свободных членов  $b$ . Решить систему  $Ax=b$ .

$$A=[1 \ 3 \ 5; 4 \ 2 \ 1; 3 \ 1 \ 1]; b=[29; 19; 13].$$

2. Задать матрицу  $A$   $3 \times 3$  и строку свободных членов  $b$ . Решить систему  $xA=b$ .

3. Чему равно  $1/0$  в MATLABe?

4.  $M = \begin{pmatrix} 8 & 2 & 1 \\ 9 & 0 & 5 \\ 7 & 8 & 9 \end{pmatrix}$ . Каков результат действия оператора  $M(:,1:2)=[ \ ]$ ?

5. Построить графики двух функций  $y=\cos(x)$ ,  $y=\sin(x)/x$  в одной системе координат.

6. Построить график поверхности  $z=x.*\sin(x+y)+y$  и ее проекции на плоскость  $xu$  в виде контурного графика.

7. Написать файл-функцию для задания функции  $z=x^2 + 5x + tg(y)$ .

8.  $M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ . Каков результат действия оператора  $M(2,:)= [ \ ]$ ?

#### ***Вариант 2***

1. Задать матрицу  $A$   $3 \times 3$  и столбец свободных членов  $b$ . Решить систему  $Ax=b$ .

$$A= [2,7,5; 7,1,3; 9,8,1]; b=[20;9;13].$$

2. Задать матрицу  $A$   $3 \times 3$  и строку свободных членов  $b$ . Решить систему  $xA=b$ .

3. Чему равно  $0/0$  в MATLABe?

4.  $M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ . Каков результат действия оператора  $M(:,2)=[ ]$ ?

5. Построить графики двух функций  $y=\sin(x)$ ,  $y=\sin(x)/x$  в одной системе координат.

6. Построить график поверхности  $z=5*y*\cos(2*x-y)$  и ее проекции на плоскость  $xu$  в виде контурного графика.

7. Написать файл-функцию для задания функции  $z=x+\sin(y)$ .

8.  $M = \begin{pmatrix} 9 & 4 & 3 \\ 4 & 6 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ . Каков результат действия оператора  $M(:,1:2)=[ ]$ ?

### **Контрольные вопросы**

1. Что выполняют команды: `sinh`, `log`, `real`, `plot`, `rank`, `eig`, `meshgrid`, `subplot`, `axis`, `grid`?

2. Какие математические функции присутствуют в среде MATLAB?

3. Какие графические команды и функции использует MATLAB?

### 3. ЛАБОРАТОРНАЯ РАБОТА №2. РАБОТА С ПАКЕТОМ РАСШИРЕНИЯ SYMBOLIC MATH СИСТЕМЫ MATLAB

**Цель работы:** ознакомиться с пакетом расширения Symbolic Math системы MATLAB. Изучить возможности данного пакета MATLAB.

Работа рассчитана на 4 академических часа

#### Теоретические сведения

##### *Функция создания символьных переменных sym*

Для создания символьных переменных используется функция `sym`:

`x = sym('x')` — возвращает символьную переменную с именем 'x' и записывает результат в `x`.

##### *Обращение матрицы – inv*

Обращением квадратной матрицы  $X$  называют результат деления единичной матрицы  $E$  того же размера, что и исходная матрица, на эту матрицу, то есть  $X^{-1} = E/X$ . Для обращения (инвертирования) матрицы в символьном виде используется функция `inv`:

» `syms abcd;`

» `inv([a b;c d])`

`ans =`

`[d/(a*d-b*c), -b/(a*d-b*c)]`

`[-c/(a*d-b*c), a/(a*d-b*c)]`



### ***Вычисление детерминанта матрицы — det***

Функция  $\det(X)$  вычисляет детерминант квадратной матрицы в символьном виде.

#### **Пример**

```
» syms abcd;
» det([a b; c d])
ans =
a*d-b*c
```

### ***Вычисление ранга матрицы — rank***

Для вычисления ранга квадратной матрицы используется функция `rank`:

- `rank(A, tol)` – число сингулярных значений матрицы  $A$ , определенных с погрешностью `tol`;
- `rank(A)` – аналогично предшествующей функции при `tol=max(size(A))*norm(A)*eps`.

#### **Пример**

```
» syms abcd;
» rank([a b;c d])
ans =
2
```

### ***Функция вычисления производных — diff***

Для вычисления в символьном виде производных от выражения  $S$  служит функция `diff`, записываемая в формате `diff(S,'v')` или `diff(S,sym(v))`. Она возвращает символьное значение первой производной от символьного выражения или массива

символьных выражений  $S$  по переменной  $v$ . Эта функция возвращает  $S'(v) = \frac{dS}{dv}$

`diff(S,n)` — возвращает  $n$ -ю ( $n$  — целое число) производную от символьного выражения или массива символьных выражений  $S$  по переменной  $v$ ;

$\text{diff}(S, v, n)$  и  $\text{diff}(S, n, 'v')$  — возвращает  $n$ -ю производную  $S$  по переменной  $v$ , то есть значение.

$$S^n(v) = \frac{d^n S}{dv^n}$$

### Примеры

»  $x = \text{sym}('x'); y = \text{sym}('y');$

»  $\text{diff}(x^y)$

ans =

$x^y * y / x$

»  $\text{diff}(x^y, x)$

ans =

$x^y * y / x$

»  $\text{simplify}(\text{ans})$

ans =

$x^{(y-1)} * y$

»  $\text{diff}(\sin(y*x), x, 3)$

ans =

$-\cos(y*x) * y^3$

»  $\text{diff}([x^3 \sin(x) \exp(x)], x)$

ans =

$[3*x^2, \cos(x), \exp(x)]$

### Функция вычисления интегралов – *int*

В практической работе часто возникает необходимость вычисления неопределенных и определенных интегралов вида

$$I = \int f(x) dx \quad \text{и} \quad I = \int_a^b f(x) dx$$

Здесь  $f(x)$  — подынтегральная функция независимой переменной  $x$ ,  $a$  - нижний и  $b$  — верхний пределы интегрирования для определенного интеграла.

• `int(S)` — возвращает символьное значение неопределенного интеграла от символьного выражения или массива символьных выражений  $S$  по переменной, которая автоматически определяется функцией `findsym`. Если  $S$  — скаляр или матрица, то вычисляется интеграл по переменной 'x'.

• `int(S,v)` — возвращает неопределенный интеграл от  $S$  по переменной  $v$ .

• `int(S,a,b)` — возвращает определенный интеграл от  $S$  с пределами интегрирования от  $a$  до  $b$ , причем пределы интегрирования могут быть как символьными, так и числовыми.

• `int(S,v,a,b)` — возвращает определенный интеграл от  $S$  по переменной  $v$  с пределами от  $a$  до  $b$ .

Примеры применения этой функции приводятся ниже:

```
» x=sym('x');
```

```
» int(x^2,x)
```

```
ans =
```

```
1/3*x^3
```

```
>> int(sin(x)^3,x)
```

```
ans =
```

```
-1/3*sin(x)^2*cos(x)-2/3*cos(x)
```

```
» int(log(2*x),x)
```

```
ans =
```

```
log(2*x)*x-x
```

```
» int((x^2-2)/(x^3-1),x,2,5)
```

```
ans =
```

```
-2/3*log(2)+2/3*log(31)+2/3*3^(1/2)*atan(11/3*3^(1/2))-...
```

```
2/3*log(7)-2/3*3^(1/2)*atan(5/3*3^(1/2))
```

## Функция вычисления пределов — *limit*

Вычисление пределов функций представляет собой важный раздел математического анализа.

Число  $L$  называется пределом функции  $F(x)$  в точке  $a$ , если при  $x$ , стремящемся к  $a$  (или  $x \rightarrow a$ ) значение функции неограниченно приближается к  $L$ . Это обозначается следующим образом:

$$\lim_{x \rightarrow a} F(x) = L.$$

Предел может быть конечным числом, положительной или отрицательной бесконечностью.

Есть функции (например, разрывные в точке  $x = a$ ), у которых нет предела в самой точке  $x = a$ , но есть предел при  $x \rightarrow a - 0$  или при  $x \rightarrow a + 0$ , где под 0 подразумевается очень малое число. В первом случае говорят о существовании предела *слева* от точки  $x = a$ , а во втором — *справа* от этой точки. Если эти пределы равны, то существует предел функции в точке  $x = a$ .

Для вычисления пределов аналитически (символьно) заданной функции  $F(x)$  служит функция `limit`, которая записывается в следующих вариантах:

- `limit(F,x,a)` — возвращает предел символьного выражения  $F$  в точке  $x \rightarrow a$ ;
- `limit(F,a)` — возвращает предел для независимой переменной, определяемой функцией `findsym`;
- `limit(F)` — возвращает предел при  $a=0$ ;
- `limit(F,x,a,'right')` или `limit(F,x,a,'left')` — возвращает предел в точке  $a$  справа или слева.

Ниже представлены примеры применения этих функций:

```
» syms a x
```

```
» limit(sin(a*x)/x)
```

```
ans =
```

```
a
```

```
» limit(2*sin(x)/x)
```

```
ans =
```

```
2
```

» limit(2+sin(x)/x,0)

ans =

3

» limit(tan(x),pi/2)

ans =

NaN

» limit(tan(x),x,pi/2.'right')

ans =

-inf

» limit([sin(x)/x, (1+x)/(2+x)],x,0)

ans

[1, 1/2]

### ***Функция разложения выражения в ряд Тейлора — taylor***

В задачах аппроксимации и приближения функций  $f(x)$  важное место занимает их разложение в ряд Тейлора в окрестности точки  $a$ :

$$f(x) = \sum_{n=0}^{\infty} (x-a)^n \frac{f^n(a)}{n!}$$

Частным случаем этого ряда при  $a = 0$  является ряд Маклорена:

$$f(x) = \sum_{n=0}^{\infty} (x)^n \frac{f^n(0)}{n!}$$

Для получения разложений аналитических функций в ряд Тейлора (и Маклорена) служит функция `taylor`:

• `taylor(f)` — возвращает шесть членов ряда Маклорена (ряд Тейлора в точке  $x = 0$ ). В любом варианте разложения можно задавать число членов ряда  $n$ , точку  $a$ , относительно которой ищется разложение, и переменную  $x$ , по которой ищется разложение, например `taylor(f,n,x,a)`.

• `taylor(f, n)` — возвращает члены ряда Маклорена до  $(n - 1)$ -го порядка.

• `taylor(f, a)` — возвращает ряд Тейлора в окрестности точки  $a$ .

• `taylor(f, x)` — возвращает ряд Тейлора для переменной  $x$ , определяемой функцией `findsym(f)`.

Примеры разложения функций в ряд:

» `x=sym('x');`

» `F=sin(x);`

» `taylor(F)`

`ans =`

$x - \frac{1}{6}x^3 + \frac{1}{120}x^5$

» `taylor(F,10)`

`ans =`

$x - \frac{1}{6}x^3 + \frac{1}{120}x^5 - \frac{1}{5040}x^7 + \frac{1}{362880}x^9$

» `taylor(exp(x),1)`

`ans =`

1

» `taylor(cos(x),-pi/2,6)`

`ans =`

$x + \frac{1}{2}\pi - \frac{1}{6}(x + \frac{1}{2}\pi)^3 + \frac{1}{120}(x + \frac{1}{2}\pi)^5$

### ***Решение алгебраических уравнений — solve***

Для решения систем алгебраических уравнений и одиночных уравнений служит функция `solve`:

• `solve(expr1,expr2,...,exprN,var1,var2,...,varN)` — возвращает значения переменных  $var1$ , при которых соблюдаются равенства, заданные выражениями `expr1`. Если в выражениях не используются знаки равенства, то полагается `exprI=0`;

•..... `solve(expr1,expr2,...,exprN)` — аналогична предшествующей функции, но переменные, по которым ищется решение, определяются функцией `findsym`.

При отсутствии аналитического решения и числе неизвестных, равном числу уравнений, ищется только одно *численное* решение, а не все решения. Результат решения возможен в следующих формах:

• для одного уравнения и одной переменной решение возвращается в виде

одномерного или многомерного массива ячеек;

- при одинаковом числе уравнений и переменных решение возвращается в упорядоченном по именам переменных виде;

- для систем с одним выходным аргументом решение возвращается в виде массива записей.

Примеры решения уравнений:

```
» syms x y;
```

```
» solve(x^3-1,x)
```

```
ans =
```

```
[ 1]
```

```
[-1/2+1/2*i*3^(1/2)]
```

```
[-1/2-1/2*i*3^(1/2)]
```

```
» solve(x^2-x-9,x)
```

```
ans =
```

```
[1/2+1/2*37^(1/2)]
```

```
[1/2-1/2*37^(1/2)]
```

```
» syms a b c
```

```
» solve(a*x^2+b*x+c)
```

```
ans =
```

```
[1/2/a*(-b+(b^2-4*a*c)^1/2)]
```

```
[1/2/a*(-b-(b^2-4*a*c)^1/2)]
```

```
» S=solve('x+y=3','x*y^2=4',x,y)
```

```
S =
```

```
x:[3x1 sym]
```

```
y:[3x1 sym]
```

```
» S.x
```

```
ans =
```

```
[4]
```

```
[1]
```

```
[1]
```

```

» S.y
ans =
[-1]
[2]
[2]
» solve('sin(x)=0.5',x)
ans =
.52359877559829887307710723054658

```

Обратите внимание на то, что для задания уравнений в явном виде используются строковые выражения, когда уравнения заключаются в апострофы.

### ***Решение дифференциальных уравнений — dsolve***

Для решения дифференциальных уравнений в форме Коши MATLAB имеет следующую функцию:

- `dsolve(eqnl', 'eqn2', ...)` — возвращает аналитическое решение системы дифференциальных уравнений с начальными условиями. Они задаются равенствами `eqnl` (вначале задаются уравнения, затем начальные условия).

По умолчанию независимой переменной считается переменная 't', обычно обозначающая время. Можно использовать и другую переменную, добавив ее в конец списка параметров функции `dsolve`. Символ `D` обозначает производную по независимой переменной, то есть  $d/dt$ , при этом `D2` означает  $d^2/dt^2$  и т. д. Имя независимой переменной не должно начинаться с буквы `D`.

Начальные условия задаются в виде равенств `'y(a)=b'` или `'Du(a)=b'`, где `y` — независимая переменная, `a` и `b` — константы. Если число начальных условий меньше, чем число дифференциальных уравнений, то в решении будут присутствовать произвольные постоянные `C1`, `C2` и т. д. Правила вывода подобны приведенным для функции `solve`.

Примеры применения функции `dsolve`:

```

» dsolve('D2x=-2*x')
ans =

```



$$C1*\cos(2^{(1/2)*t})+C2*\sin(2^{(1/2)*t})$$

» dsolve('D2y=-2\*x+y', 'y(0)=1, 'x')

ans =

$$(2*x*\exp(x)+(-C2+1)*\exp(x)^2+C2)/\exp(x)$$

### Задания для выполнения

#### Вариант 1

1. Найти формулу обращения матрицы  $A = \begin{pmatrix} p & o \\ r & s \end{pmatrix}$ .

2. Найти собственные значения числовой матрицы A.

3. Получить аналитическое выражение для отношения производных функций

$$y = x^3 \sqrt{1+x}, \quad y = 2 \sin x + e^{x^2}.$$

4. Вычислить интеграл  $\int_0^a \int_0^b \int_0^c (x^2 + y^3) z dx dy dz$ .

5. Найти десять первых членов разложения функции  $\sin(x)$  в ряд Маклорена.

6. Решить уравнение  $x^5 + 4 = 0$ .

7. Решить систему уравнений  $x + y = 3$ ,  $xy^2 = 4$ .

8. Найти преобразование Фурье от  $\exp(-x^2)$ .

9. Найти 13 значащих цифр числа  $\pi$  (используйте арифметику произвольной точности).

10. Изобразить движение точки в трехмерном пространстве, если заданы зависимости ее координат от времени:  $x = \cos(t)$ ,  $y = \sin(t)$ ,  $z = t^2$ .

11. Построить поверхность, задаваемую уравнением  $z = \text{real}(\text{arcsec}(x + iy))$ .

#### Вариант 2

1. Найти формулу обращения матрицы  $A = \begin{pmatrix} e & f \\ j & k \end{pmatrix}$ .

2. Найти собственные значения числовой матрицы A.

3. Получить аналитическое выражение для отношения производных функций

$$y = x^{1/2} \sqrt{1+x}, \quad y = 2 \cos x + e^{x^4}.$$

4. Вычислить интеграл  $\int_0^a \int_0^b \int_0^c (x^3 + y^2)xyz dx dy dz$ .
5. Найти пять первых членов разложения функции  $\cos(x)$  в ряд Маклорена.
6. Решить уравнение  $x^5 - 2 = 0$ .
7. Решить систему уравнений  $x^2 + y = 5$   $x^2 y = 9$ .
8. Найти преобразование Фурье от  $\exp(-x^2)$ .
9. Найти 10 значащих цифр числа  $\pi$  (используйте арифметику произвольной точности).
10. Изобразить движение точки в трехмерном пространстве, если заданы зависимости ее координат от времени:  $x = \cos(t)$ ,  $y = \sin(t)$ ,  $z = t^2$ .
11. Построить поверхность, задаваемую уравнением  $z = \text{real}(\text{arcsec}(ix - y))$ .

### **Контрольные вопросы**

Что означают данные команды пакета расширения Symbolic Math: SYM, SYMS, FINDSYM, PRETTY, ISVARNAME, VECTORIZE, DIGITS, VPA, DIAG, TRIU, TRIL, INV, DET, RANK, EIG, POLY, DIFF, INT, SUBS, SIMPLE, COMPOSE, EZPLOT, EZPLOT3, EZPOLAR, EZSURF, EZSURFC?

## 4. ЛАБОРАТОРНАЯ РАБОТА №3. ПРОГРАММИРОВАНИЕ В СРЕДЕ MATLAB: ОПЕРАТОРЫ И МАССИВЫ

**Цель работы:** ознакомиться с языком программирования в среде MATLAB. Изучить синтаксис написания операторов if, while, for, switch, break.

Работа рассчитана на 4 академических часа

### Теоретические сведения

#### *Создание M-файлов. M-сценарии. M-функции*

M-файлы являются обычными текстовыми файлами, которые создаются с помощью текстового редактора. Система MATLAB поддерживает специальный встроенный редактор/отладчик.

Открыть редактор можно двумя способами:

- из меню File выбрать опцию New, а затем M-File.
- использовать команду редактирования edit.

*M-сценарии.* Сценарии являются самым простым типом M-файла – у них нет входных и выходных аргументов. Они используются для автоматизации многократно выполняемых вычислений. Сценарии оперируют данными из рабочей области и могут генерировать новые данные для последующей обработки в этом же файле. Данные, которые используются в сценарии, сохраняются в рабочей области после завершения сценария и могут быть использованы для дальнейших вычислений.

### Пример

Следующие операторы вычисляют радиус-вектор rho для различных тригонометрических функций от угла theta и строят последовательность графиков в полярных координатах.

% M-file petals - сценарий построения лепесткового графика

```
theta = -pi:0.01:pi;
```

```
rho(1, :) = 2*sin(5*theta).^2;
```

```

rho(2, :) = cos(10*theta).^3;
rho(3, :) = sin(theta).^2;
rho(4, :) = 5*cos(3.5*theta).^3;
for i = 1:4
    polar (theta, rho(i, :))
    pause
end

```

М-файл petals.m является сценарием. Ввод команды petals.m в командной строке системы MATLAB вызывает выполнение операторов этого сценария.

После того, как сценарий отобразит первый график, необходимо нажать клавишу Return, чтобы перейти к следующему графику. В сценарии отсутствуют входные и выходные аргументы; программа petals.m сама создаёт переменные, которые сохраняются в рабочей области системы MATLAB. Когда выполнение завершено, переменные (i, theta и rho) остаются в рабочей области. Для того чтобы увидеть этот список, следует воспользоваться командой whos.

*М-функции.* М-функции являются М-файлами, которые допускают наличие входных и выходных аргументов. Они работают с переменными в пределах собственной рабочей области, отличной от рабочей области системы MATLAB.

### **Пример**

Функция average - это М-файл, который вычисляет среднее значение элементов вектора.

```

function y = average (x)
% AVERAGE Среднее значение элементов вектора.
% AVERAGE(X), где X - вектор. Вычисляет среднее значение элементов
вектора.
% Если входной аргумент не является вектором, генерируется ошибка.
[m,n] = size(x);
if (~(m == 1) | (n == 1)) | (m == 1 & n == 1))
    error('Входной массив должен быть вектором')
end

```

```
y = sum(x)/length(x);      % Собственно вычисление
```

Функция `average` допускает единственный входной и единственный выходной аргументы. Для того чтобы вызвать функцию `average`, надо ввести следующие операторы:

```
z = 1:99;  
average(z)  
ans = 50
```

### ***Структура М-функции.***

М-функция состоит из:

- строки определения функции;
- первой строки комментария;
- собственно комментария;
- тела функции;
- строчных комментариев;

***Строка определения функции.*** Строка определения функции сообщает системе MATLAB, что файл является М-функцией, а также определяет список входных аргументов.

### **Пример**

Строка определения функции `average` имеет вид:

```
function y = average(x)
```

Здесь:

- `function` - ключевое слово, определяющее М-функцию;
- `y` - выходной аргумент;
- `average` - имя функции;
- `x` - входной аргумент.

Каждая функция в системе MATLAB содержит строку определения функции, подобную приведенной.

Если функция имеет более одного выходного аргумента, список выходных аргументов помещается в квадратные скобки. Входные аргументы помещаются в круглые скобки.

### **Пример**

```
function [x, y, z] = sphere(theta, phi, rho)
```

*Первая строка комментария.* Ниже приведены несколько строк комментария

### **Пример**

```
% Функция average(x) вычисляет среднее значение элементов вектора x.  
% Если входной аргумент не является вектором, выдается ошибка.
```

При вводе команды подсказки `help <имя_функции>`, система MATLAB отображает строки комментария, которые размещаются между строкой определения функции и первой пустой строкой, либо началом программы. Команда `help <имя_функции>` игнорирует комментарии, размещенные вне этой области.

**Тело функции.** Тело функции содержит код языка MATLAB, который выполняет вычисления и присваивает значения выходным аргументам. Операторы в теле функции могут состоять из вызовов функций, программных конструкций для управления потоком команд, интерактивного ввода/вывода, вычислений, присваиваний, комментариев и пустых строк.

### **Пример**

Тело функции `average` включает ряд простейших операторов программирования:

```
[m, n] = size(x);  
if ~(m == 1 | n == 1) | (m == 1 & n == 1)  
    Error('Input должно быть вектором')  
end
```

```
y = sum(x)/length(x);
```

Комментарии отмечаются знаком (%). Строка комментария может быть размещена в любом месте М-файла, в том числе и в конце строки.

## Пример

```
% Найти сумму всех элементов вектора x
```

```
y = sum(x)
```

```
% Использована функция sum.
```

**Имена М-функций.** 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. Если имя файла и имя функции в строке определения функции разные, то используется имя файла, а внутреннее имя игнорируется. Хотя имя функции, определенное в строке определения функции, может и не совпадать с именем файла, настоятельно рекомендуется использовать одинаковые имена.

**Операторы ветвления и операторы цикла.** Для управления порядком выполнения операций служат операторы управления. К ним относятся операторы ветвления и операторы цикла.

К операторам ветвления в М-языке относятся условный оператор и оператор переключения. Условный оператор использует ключевые слова: if ("если"), else ("иначе") elseif ("иначе если"), end ("конец всей конструкции") и в общем виде представлено как:

```
if условие1
```

```
...
```

```
elseif условие2
```

```
...
```

```
else
```

```
...
```

```
end
```

Область действия условного оператора начинается ключевым словом if, а заканчивается ключевым словом end. Под условием понимается произвольное выражение (чаще всего это выражение включает в себя операции сравнения и логические операции), истинность или ложность которого понимается как отличие или равенство нулю.

Если условие истинно, то выполняются команды, стоящие после строки с ключевым словом `if`. Если условие ложно, то эти команды пропускаются и переходят либо к следующему за условным оператору (первая форма), либо проверяют ещё одно условие в строке с ключевым словом `elseif` (третья форма условного оператора), либо выполняются без дополнительных проверок команды, стоящие после строки с ключевым словом `else` (вторая из приведённых выше форм).

В тех случаях, когда значением условных выражений будет массив, истинность условия наступает, когда истинны (не равны нулю) все элементы массива. Если хоть один элемент такого массива будет равен нулю, то условие считается ложным. Кроме того, ложность имеет место при использовании пустых массивов.

Приведём иллюстрирующий работу условного оператора фрагмент кода

```
A = [ 1 2; 4 0 ];
```

```
if A
```

```
    b = 1;
```

```
else
```

```
    b = 2;
```

в результате выполнения которого переменная `b` получит значение 2, так как матрица `A` содержит один нулевой элемент, и всё условие считается ложным.

Запись `if A` по своему действию полностью эквивалентна записи `if A~=0` и записи `if all( A(:) )`.

Оператор переключения использует ключевые слова `switch` ( "переключить" ), `case` ( "случай" ), `otherwise` ( "иначе" ) и имеет следующую конструкцию:

```
switch выражение
```

```
    case значение1
```

```
    ...
```

```
    case { значение2, значение3 }
```

```
    ...
```

```
    ...
```



otherwise

...

end

Сначала вычисляется вырабатывающее скалярное числовое значение выражение, а затем полученный результат сравнивается с набором значений значение1, значение2, значение3 и так далее. В случае совпадения с одним из значений, выполняется нижестоящая ветка. Если нет совпадения ни с каким из перечисленных значений, то выполняется ветка, стоящая после ключевого слова otherwise.

В зависимости от способа определения условия останова повторов различают два вида операторов цикла в М-языке системы MATLAB.

Первый из них использует ключевые слова for ( "для" ) и end. Он имеет вид

while выражение

...

end

Здесь повтор участка кода, обозначенного многоточием, продолжается пока выражение истинно (не равно нулю). В случае массивов истинность наступает, когда все элементы массива истинны.

Следующий фрагмент вычисляет сумму отрезка ряда:

S=0; k=1; u=1;

while u > 1e-8

S = S + u;

k = k + 1; u = 1/k^2;

end

Условием останова служит требование к очередным слагаемым быть больше некоторого заранее определённого числа: как только слагаемое станет меньше этого числа, суммирование прекратится.

Другой вариант оператора цикла использует ключевые слова for ("для") и end.

Он имеет вид:

```
for varName = выражение
```

```
...
```

```
end
```

где varName - произвольно выбираемое программистом имя так называемой переменной цикла.

В следующем фрагменте кода осуществляется сложение 57 членов ряда:

```
S=0;
```

```
for k = 1 : 1 : 57
```

```
    S = S + 1/k^2;
```

```
end
```

При каждом новом проходе переменная цикла k увеличивается на единицу.

### **Задания для выполнения**

1. Ввести одномерный массив из  $n$  элементов. Вычислить сумму отрицательных чисел, их количество и сумму всех положительных чисел, среднее арифметическое ненулевых элементов. Проверить, есть ли во введенном с клавиатуры массиве элементы с одинаковым значением. Найти количество элементов, меньших заданного числа и напечатать их.

2. Написать программу, которая проверяет, находится ли введенное с клавиатуры число в массиве. Массив должен вводиться во время работы программы. Определить, представляют ли элементы введенного с клавиатуры массива возрастающую последовательность, вычислить среднее арифметическое элементов массива без учета минимального и максимального элементов массива. Сформировать новый массив, в котором первым элементом будет последний элемент вторым - предпоследний и т.д.

### **Контрольные вопросы**

1. Какие операторы существуют в среде MATLAB?
2. Каков синтаксис операторов `if` и `for`?
3. Каков синтаксис оператора `switch`?
4. Чем отличаются операторы `for` и `while`?
5. Что означает `if A~=0`?


## 5. ЛАБОРАТОРНАЯ РАБОТА №4. ЗНАКОМСТВО С ПАКЕТОМ SIMULINK СИСТЕМЫ MATLAB

**Цель работы:** рассмотреть возможности использования пакета расширения Simulink. Ознакомиться с некоторыми элементами Simulink.

Работа рассчитана на 2 академических часа

### Теоретические сведения

*Запуск Simulink.* Для запуска программы необходимо предварительно запустить пакет MATLAB. После открытия основного окна программы MATLAB нужно запустить программу *Simulink*. Это можно сделать одним из трех способов:


Нажать кнопку  (*Simulink*) на панели инструментов командного окна *MATLAB*.

В командной строке главного окна *MATLAB* напечатать *Simulink* и нажать клавишу *Enter* на клавиатуре.

Выполнить команду *Open...* в меню *File* и открыть файл модели (*mdl* - файл).

Последний вариант удобно использовать для запуска уже готовой и отлаженной модели, когда требуется лишь провести расчеты и не нужно добавлять новые блоки в модель. Использование первого и второго способов приводит к открытию окна обозревателя разделов библиотеки *Simulink*.

*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 (здесь и далее, с помощью символа “/”, указаны пункты меню программы, которые необходимо последовательно выбрать для выполнения указанного действия).

2. Расположить блоки в окне модели. Для этого необходимо открыть соответствующий раздел библиотеки (Например, *Sources* - *Источники*). Далее,

указав курсором на требуемый блок и нажав на левую клавишу “мыши” - “перетащить” блок в созданное окно. Клавишу мыши нужно держать нажатой.

Для удаления блока необходимо выбрать блок, 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, нужно изменить параметры блока, установленные программой “по умолчанию”. Для этого необходимо дважды щелкнуть левой клавишей “мыши”, указав курсором на изображение блока. Откроется окно редактирования параметров данного блока. 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*.

4. После установки на схеме всех блоков из требуемых библиотек нужно выполнить соединение элементов схемы. Для соединения блоков необходимо указать курсором на “выход” блока, а затем, нажав 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. 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), а затем нажать клавишу *Delete* на клавиатуре.

5. 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). При последующем редактировании схемы можно пользоваться пунктом меню *File/Save*. При повторных запусках программы

*SIMULINK* загрузка схемы осуществляется с помощью меню *File/Open...* в окне обозревателя библиотеки или из основного окна *MATLAB*.

### **Задания для выполнения**

Используя Simulink создать две различные модели (каждая модель должна содержать физический смысл какого-либо процесса). В отчете должно быть краткое описание модели, представлена сама модель и полученные графики, отражающие смысл модели.

Для выполнения лабораторной работы используйте материалы:  
<http://MATLAB.exponenta.ru/simulink/book1/index.php>

### **Контрольные вопросы**

1. Какие существуют методы создания модели?
2. Какова процедура редактирования параметров блока?
3. Как удалить блок из модели?
4. Какие существуют блоки источников сигналов?
5. Какие существуют блоки приемников сигналов?

## 6. ЛАБОРАТОРНАЯ РАБОТА №5. РАЗРАБОТКА МОДЕЛИ БИНАРНОГО СИММЕТРИЧНОГО КАНАЛА СВЯЗИ С ПОМЕХАМИ

**Цель работы:** объединить полученные знания по программированию в среде MATLAB и пакету Simulink.

Работа рассчитана на 4 академических часа

### **Теоретические сведения:**

#### ***Осциллограф Scope***

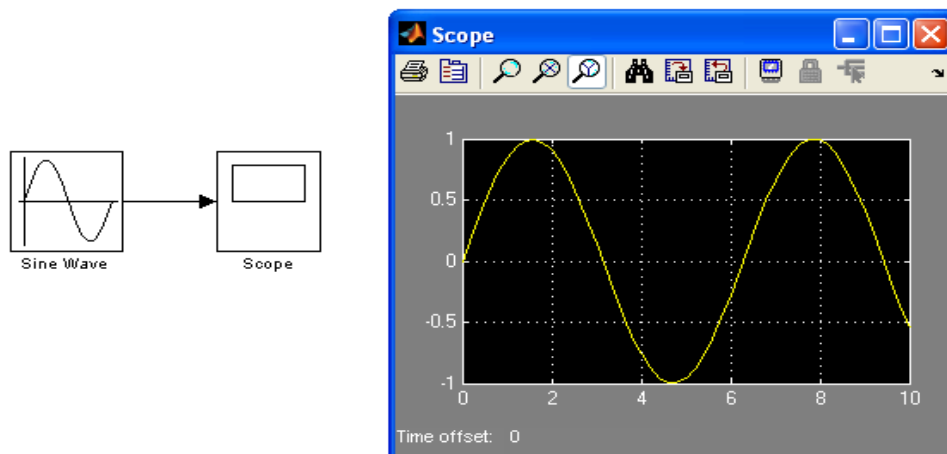
##### *Назначение*

Строит графики исследуемых сигналов в функции времени.

Панель содержит 11 кнопок:

1. Print – печать содержимого окна осциллографа.
2. Parameters – доступ к окну настройки параметров.
3. Zoom – увеличение масштаба по обеим осям.
4. Zoom X-axis – увеличение масштаба по горизонтальной оси.
5. Zoom Y-axis – увеличение масштаба по вертикальной оси.
6. Autoscale – автоматическая установка масштабов по обеим осям.
7. Save current axes settings – сохранение текущих настроек окна.
8. Restore saved axes settings – установка ранее сохраненных настроек окна.
9. Floating scope – перевод осциллографа в “свободный” режим.
10. Lock/Unlock axes selection – закрепить/разорвать связь между текущей координатной системой окна и отображаемым сигналом. Инструмент доступен, если включен режим Floating scope.
11. Signal selection – выбор сигналов для отображения. Инструмент доступен, если включен режим Floating scope.

Изображение блока и окно для просмотра графиков показаны на рисунке 6.1.



**Рисунок 6. 1 – Осциллограф Scope**

Для того, чтобы открыть окно просмотра сигналов необходимо выполнить двойной щелчок левой клавишей “мыши” на изображении блока.

### ***Блок считывания данных из рабочего пространства From Workspace***

#### *Назначение*

Получение данных из рабочего пространства MATLAB.

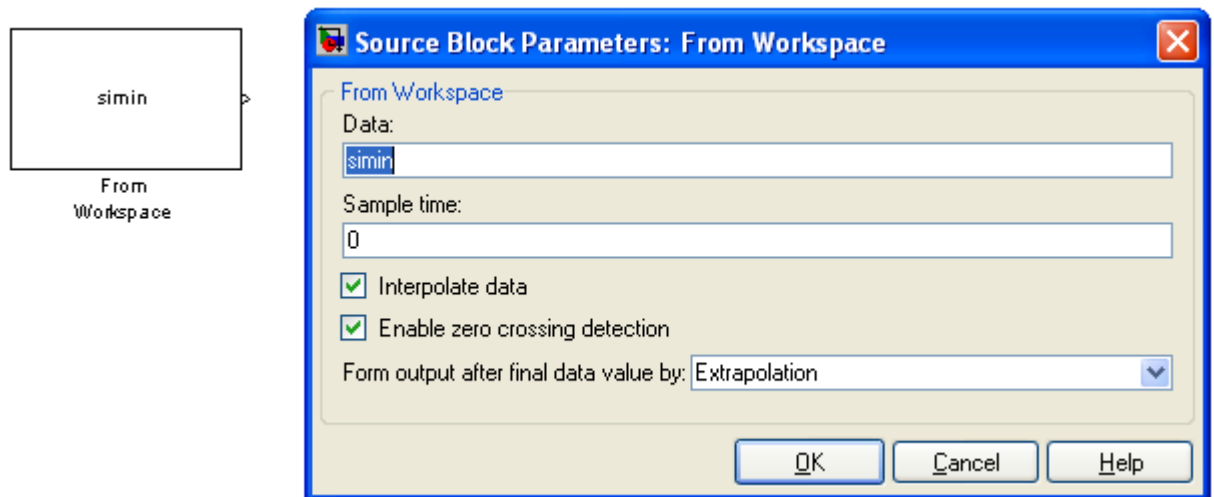
#### *Параметры*

1. Data – Имя переменной (матрицы или структуры) содержащей данные.
2. Sample time - Шаг изменения выходного сигнала блока.
3. Interpolate data — Интерполяция данных для значений модельного времени не совпадающих со значениями в переменной Data.
4. Form output after final data value by – Вид выходного сигнала по окончании значений времени в переменной Data:
  - Extrapolate – Линейная экстраполяция сигналов;
  - SettingToZero – Нулевые значения сигналов;
  - HoldingFinalValue – Выходные значения сигналов равны последним значениям;



- CyclicRepetition – Циклическое повторение значений сигналов. Данный вариант может использоваться, только если переменная Data имеет формат Structure without time.

На рисунке 6.2 показан блок *From Workspace*.



**Рисунок 6.2 – Блок считывания данных из рабочего пространства From Workspace**

### ***Блок сохранения данных в рабочей области To Workspace***

#### ***Назначение***

Блок записывает данные, поступающие на его вход, в рабочую область MATLAB.

#### ***Параметры***

- Variable name – имя переменной, содержащей записываемые данные.
- Limit data points to last – максимальное количество сохраняемых расчетных точек по времени (отсчет ведется от момента завершения моделирования). В том случае, если значение параметра Limit data points to last задано как inf, то в рабочей области будут сохранены все данные.
- Decimation – кратность записи данных в рабочую область.
- Sample time – шаг модельного времени. Определяет дискретность записи данных.

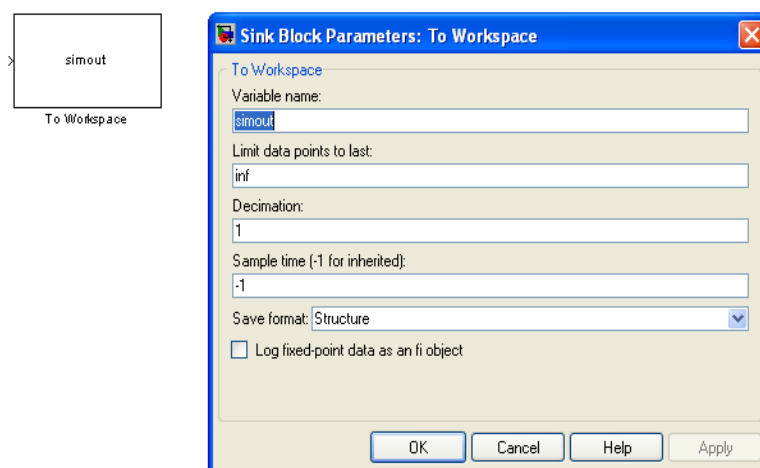
• Save format – формат сохранения данных. Может принимать значения:

1. Matrix – матрица. Данные сохраняются как массив, в котором число строк определяется числом расчетных точек по времени, а число столбцов – размерностью вектора подаваемого на вход блока. Если на вход подается скалярный сигнал, то матрица будет содержать лишь один столбец.

2. Structure – структура. Данные сохраняются в виде структуры, имеющей три поля: time – время, signals – сохраняемые значения сигналов, blockName – имя модели и блока To Workspace. Поле time для данного формата остается не заполненным.

3. Structure with Time – структура с дополнительным полем (время). Для данного формата, в отличие от предыдущего, поле time заполняется значениями времени.

На рисунке 6.3 показан блок *To Workspace*.



**Рисунок 6.3 – Блок сохранения данных в рабочей области To Workspace**

**Элементы Simulink для выполнения данного задания:**

1. Блок считывания данных из рабочего пространства From Workspace
2. Осциллограф Scope
3. Блок сохранения данных в рабочей области To Workspace
4. Двоичный симметричный канал связи (BSC)

### **Задания для выполнения**

1. Задать последовательность символов из некоторого алфавита с пятью буквами. Длина последовательности символов должна составлять 7 букв.

2. Написать программу в виде m-файла для преобразования этой последовательности в бинарную последовательность нулей и единиц. Каждому символу сопоставить при этом бинарную последовательность из четырех двоичных символов.

Вариант 1: a-0001, b-0010, c-0011, d-0100, e-0101;

Вариант 2: r-1111, s-1110, t-1101, u-1100, v-1011;

3. Преобразовать полученную последовательность  $x(i)$ ,  $i=1, \dots, 4N$ , где  $N$  – число символов в исходной последовательности, в двумерный массив  $y(i,j)$ , где  $y(i,2)=x(i)$ ,  $y(i,1)=t(i)$ , где  $t(i)$ -множество равноотстоящих друг от друга моментов времени.

4. Построить модель SIMULINK для передачи массива  $y$  через двоичный симметричный канал связи (BSC) с достаточно высоким уровнем помех. В качестве входного и выходного блока для BSC использовать блоки From Workspace и To Workspace, последний из которых создает массив  $z$  переданных кодов.

Вариант 1: Ошибка канала составляет 21%;

Вариант 2: Ошибка канала составляет 13%.

5. Преобразовать массив  $z$  в последовательность символов. Подсчитать долю неправильно переданных символов и сравнить ее с величиной вероятности ошибок в блоке BSC. Сформулировать вывод и написать краткий отчет по работе.

### **Контрольные вопросы**

1. Чем отличается M-функция от M-сценария?
2. Какова структура M-функции?
3. Как в среде MATLAB создается M-File?

## **7. ЛАБОРАТОРНАЯ РАБОТА №6. РАЗРАБОТКА МОДЕЛЕЙ АМПЛИТУДНОЙ, ЧАСТОТНОЙ И ФАЗОВОЙ МОДУЛЯЦИИ НЕСУЩЕЙ В ПАКЕТЕ РАСШИРЕНИЯ SIMULINK СИСТЕМЫ MATLAB**

**Цель работы:** рассмотреть возможности использования пакета расширения Simulink. Ознакомиться с некоторыми элементами Simulink.

Работа рассчитана на 6 академических часов.

### **Теоретические сведения**

#### ***Блок переключателя Switch***

##### *Назначение*

Выполняет переключение входных сигналов по сигналу управления.

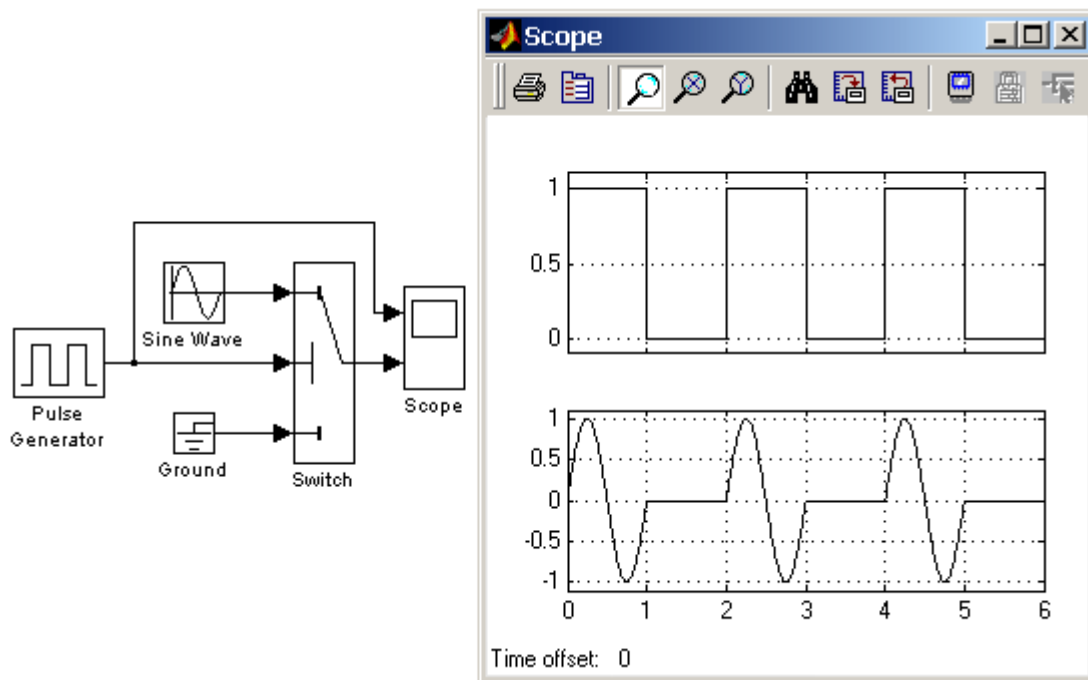
##### *Параметры*

Threshold – Порог управляющего сигнала.

Блок работает следующим образом.

Если сигнал управления, подаваемый на средний вход больше, чем величина порогового значения Threshold, то на выход блока проходит сигнал с первого (верхнего) входа. Если сигнал управления станет меньше, чем пороговое значение, то на выход блока будет поступать сигнал со второго (нижнего) входа.

На рисунке 7.1 показан пример работы блока Switch. В том случае, когда сигнал на управляющем входе ключа равен 1, на выход блока проходит гармонический сигнал, если же управляющий сигнал равен нулю, то на выход проходит сигнал нулевого уровня от блока Ground. Пороговое значение управляющего сигнала задано равным 0.5.



**Рисунок 7.1 – Применение переключателя Switch**

### ***Блок ручного переключателя Manual Switch***

#### *Назначение*

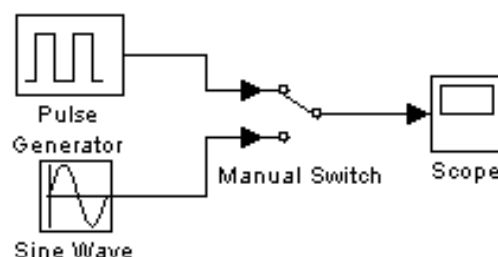
Выполняет переключение входных сигналов по команде пользователя.

#### *Параметры*

Нет.

Командой на переключение является двойной щелчок левой клавишей “мыши” на изображении блока. При этом изображение блока изменяется, показывая, какой входной сигнал в данный момент проходит на выход блока. Переключение блока можно выполнять как до начала моделирования, так и в процессе расчета.

На рисунке 7.2 показан пример использования блока Manual Switch.



**Рисунок 7.2 – Пример использования блока Manual Switch**

## ***Блок многовходового переключателя Multiport Switch***

### *Назначение*

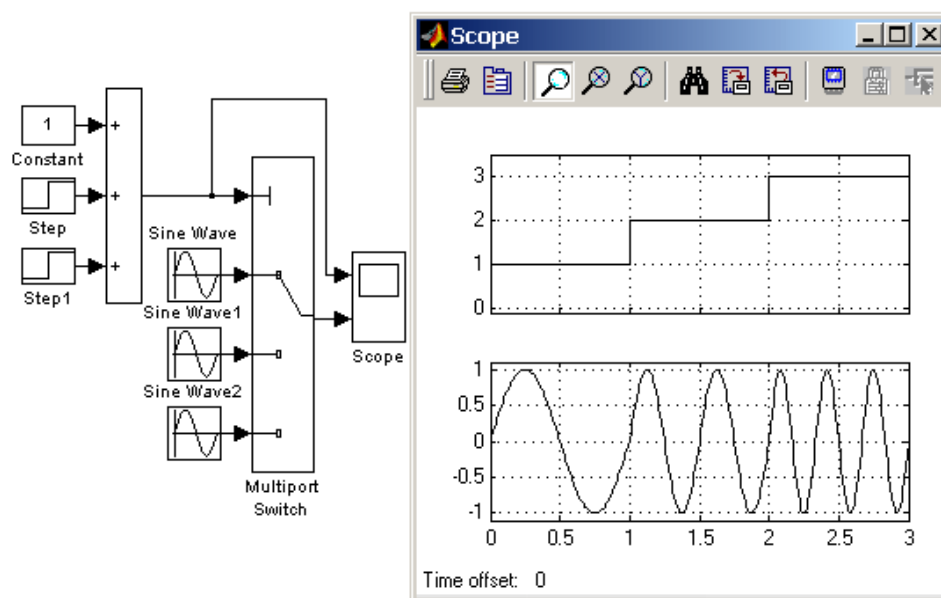
Выполняет переключение входных сигналов по сигналу управления, задающему номер активного входного порта.

### *Параметры*

Number of inputs – Количество входов.

Блок многовходового переключателя Multiport Switch, пропускает на выход сигнал с того входного порта, номер которого равен текущему значению управляющего сигнала. Если управляющий сигнал не является сигналом целого типа, то блок Multiport Switch производит отбрасывание дробной части числа, при этом в командном окне MATLAB появляется предупреждающее сообщение.

На рисунке 7.3 показан пример работы блока Multiport Switch. Управляющий сигнал переключателя имеет три уровня и формируется с помощью блоков Constant, Step, Step1 и Sum. На выход блока Multiport Switch, в зависимости от уровня входного сигнала, проходят гармонические сигналы, имеющие разные частоты.



**Рисунок 7.3 – Применение переключателя Multiport Switch**

### **Элементы Simulink для выполнения данного задания:**

- Источник линейно изменяющегося воздействия Ramp;
- Источник постоянного сигнала Constant;
- Осциллограф Scope;
- Блок сохранения данных в рабочей области To Workspace;
- Блок считывания данных из рабочего пространства From Workspace;
- Блок вычисления суммы Sum;
- Блок умножения Product;
- Блок вычисления тригонометрических функций Trigonometric Function;
- Блок переключателя Switch Case;
- Блок многовходового переключателя Multiport Switch;
- Блок ручного переключателя Manual Switch.

### **Задания для выполнения**

1. Построить модель SIMULINK, моделирующую сигнал вида  $s(t) = A_m E(t) \cos(\omega_0 t + \psi(t) + \varphi_0)$ , представляющий собой гармоническое колебание с несущей частотой  $\omega_0$  и амплитудой  $A_m$ , модулированное заданными медленно меняющимися функциями  $E(t), \psi(t)$ .

2. Выбрав некоторое значение эталонного времени  $T$ , в соответствии с заданной кодовой двоичной последовательностью сформировать амплитудный модулирующий сигнал вида

$$E(t) = \begin{cases} E_1, & \text{если передается 0} \\ E_2, & \text{если передается 1} \end{cases}$$

Частоту  $\omega_0$  принять равной  $\pi/5$ . Фазу  $\psi(t)$  при этом считать равной нулю.

3. В новом окне создать аналогичную модель для фазовой модуляции, считая

$$\psi(t) = \begin{cases} 0, & \text{если передается пара символов 00} \\ \frac{\pi}{2}, & \text{если передается пара символов 01} \\ \pi, & \text{если передается пара символов 10} \\ \frac{3\pi}{2}, & \text{если передается пара символов 11} \end{cases}$$

4. Разработать систему кодирования и модель формирования сигнала для частотной модуляции с двумя несущими частотами  $w_1$  и  $w_2$ , соответствующими значению передаваемого бита 0 и 1 соответственно, т.е.

$$w(t) = \begin{cases} w_1, & \text{если передается 0} \\ w_2, & \text{если передается 1} \end{cases}$$

### **Контрольные вопросы**

1. Как на модели продемонстрировать амплитудную, частотную и фазовую модуляцию сигнала?
2. Что определяет скорость изменения выходного сигнала?
3. Какие параметры имеет блок Ramp?
4. Как работает блок многовходового переключателя?
5. Чем отличается алгоритм работы блока переключателя от блока ручного переключателя?



## 8. ЛАБОРАТОРНАЯ РАБОТА №7. ПРЕОБРАЗОВАНИЕ ФУРЬЕ В СИСТЕМЕ MATLAB

**Цель работы:** изучить преобразование Фурье в пакете MATLAB и Symbolic Math Toolbox.

Работа рассчитана на 4 академических часа

### Теоретические сведения

**Преобразование Фурье. Одномерное дискретное прямое преобразование Фурье – *fft***

*Синтаксис*

$Y = \text{fft}(X)$

$Y = \text{fft}(X, n)$

*Описание*

Дискретные прямое преобразования Фурье для одномерного массива  $x$  длины  $N$  определяются следующим образом:

$$X(k) = \sum_{j=1}^N x(j) e^{2\pi j / N (j-1)(k-1)} .$$

Функция  $Y = \text{fft}(X)$  вычисляет для массива данных  $X$  дискретное преобразование Фурье, используя FFT-алгоритм быстрого Фурье-преобразования. Если массив  $X$  двумерный, вычисляется дискретное преобразование каждого столбца.

Функция  $Y = \text{fft}(X, n)$  вычисляет  $n$ -точечное дискретное преобразование Фурье. Если  $\text{length}(X) < n$ , то недостающие строки массива  $X$  заполняются нулями; если  $\text{length}(X) > n$ , то лишние строки удаляются.

## Пример

Основное назначение преобразования Фурье - выделить частоты регулярных составляющих сигнала, зашумленного помехами. Рассмотрим данные, поступающие с частотой 1000 Гц. Сформируем сигнал, содержащий регулярные составляющие с частотами 50 Гц и 120 Гц и случайную аддитивную компоненту с нулевым средним.

```
t = 0:0.001:0.6;  
x = sin(2 * pi * 50 * t) + sin(2 * pi * 120 * t);  
y = x + 2 * randn(size(t));  
plot(y(1:50)), grid
```

Реализуя одномерное преобразование Фурье этого сигнала на основе 512 точек и построив график спектральной плотности, можно выделить две частоты, на которых амплитуда спектра максимальна. Это частоты 120 и 50 Гц.

```
Y = fft(y, 512);  
Pyy = Y.*conj(Y)/512;  
f = 1000 * (0:255)/512;  
figure(2), plot(f, Pyy(1:256)), grid
```

## ***Преобразование Фурье в Symbolic Toolbox. Прямое преобразование Фурье – `fourier`***

Прямым преобразованием Фурье является следующее преобразование:

$$F(\omega) = \int_{-\infty}^{\infty} f(x)e^{-i\omega x} dx,$$

где  $f(x)$  — скалярная функция независимой переменной  $x$ . Оно реализуется функцией `fourier`:

• `F=fourier(f)` — возвращает  $F(\omega)$  и определяет независимую переменную с помощью функции `findsym` (по умолчанию это  $x$ ). Если  $f = f(\omega)$ , то возвращается функция  $F = F(t)$ .

•  $F = \text{fourier}(f, v)$  — аналогична ранее приведенной функции, но заменяет аргумент возвращаемой функции  $F$  (по умолчанию —  $w$ ), на  $v$ , то есть реализует преобразование Фурье по формуле

$$F(v) = \int_{-\infty}^{\infty} f(x) e^{-ivx} dx$$

•  $\text{fourier}(f, u, v)$  — аналогична исходной функции, но заменяет аргумент  $x$  в  $f(x)$  на  $u$ , а аргумент в  $F(w)$  на  $v$ , то есть дает следующее преобразование:

$$F(v) = \int_{-\infty}^{\infty} f(u) e^{-ivu} du$$

Примеры прямого преобразования Фурье:

» `syms f F x w u v`

» `fourier(0, 1*x)`

`ans =`

`1/5*i*pi*Dirac(1,w)`

» `F=fourier(sin(x),v)`

`F =`

`-i*pi*Dirac(v-1)+i*pi*Dirac(v+1)`

» `syms t`

» `f=1/t^2;`

» `F=fourier(f,v)`

`F =`

`pi*v*(Heaviside(-v)-Heaviside(v))`

» `fourier(exp(-x^2),x,t)`

`ans =`

`pi^(1/2)*exp(-1/4*t^2)`

### **Задания для выполнения:**

С помощью функции `fft` системы MATLAB построить фурье-образы следующих функций:

$$g(t) = ae^{-\alpha|t|}$$

$$g(t) = ae^{-\alpha|t|} \cos \omega_0 t$$

при различных значениях параметров  $a > 0, \alpha > 0, \omega_0 > 0$ .

Определить, при каком значении параметров  $\alpha, \omega_0$  получается одногорбое и двугорбое изображение.

Функции задавать как массивы не менее 2000 точек.

Проверить результат путем точного вычисления преобразования в SYMBOLIC MATH TOOLBOX.

### **Контрольные вопросы**

1. Назначение преобразования Фурье?
2. Какие аргументы входят в функцию `fft`?
3. Какие аргументы входят в функцию `fourier`?
4. Как можно охарактеризовать частотный состав измеряемого сигнала?
5. В чем разница между прямым и обратным преобразованием Фурье?

## 9. ЛАБОРАТОРНАЯ РАБОТА №8. РАЗРАБОТКА ГЕНЕРАТОРОВ СИГНАЛОВ В ПАКЕТЕ РАСШИРЕНИЯ SIMULINK СИСТЕМЫ MATLAB

**Цель работы:** Изучить моделирование амплитудной, частотной и фазовой модуляции несущей в пакете расширения Simulink системы MATLAB

Работа рассчитана на 4 академических часа.

### Теоретические сведения

*Источник случайного сигнала с нормальным распределением **Random Number***

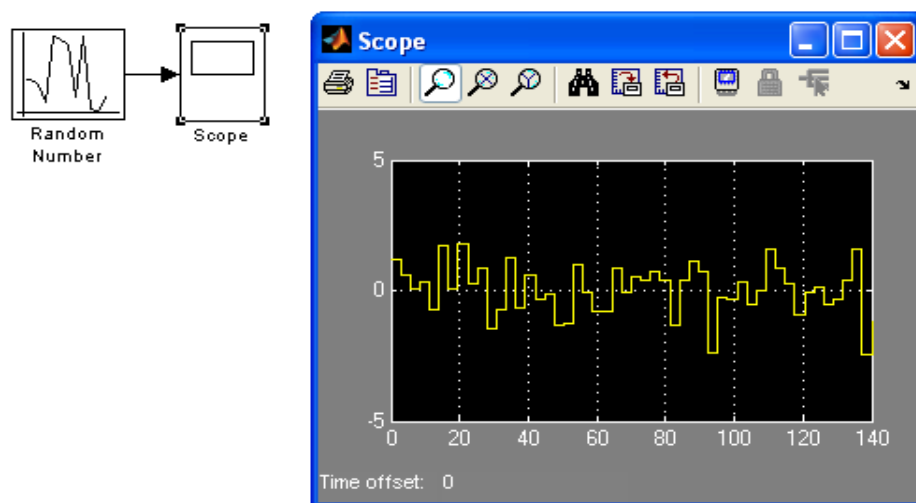
#### *Назначение*

Формирование случайного сигнала с нормальным распределением уровня сигнала.

#### *Параметры*

1. Mean - Среднее значение сигнала
2. Variance- Дисперсия (среднеквадратическое отклонение).
3. Initial seed – Начальное значение.

Пример использования блока представлен на рисунке 9.1.



**Рисунок 9.1 – Источник случайного сигнала с нормальным распределением**

## ***Источник случайного сигнала с равномерным распределением Uniform Random Number***

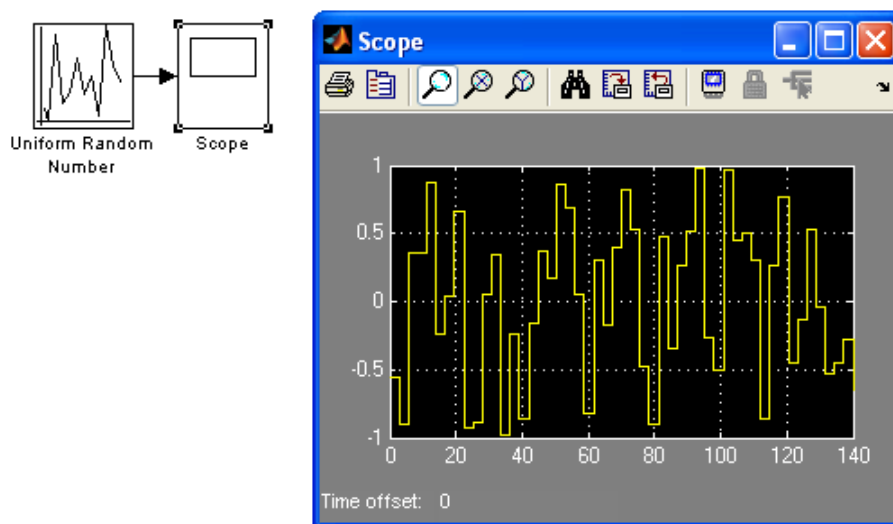
### *Назначение*

Формирование случайного сигнала с равномерным распределением.

### *Параметры*

1. Minimum – Минимальный уровень сигнала.
2. Maximum – Максимальный уровень сигнала.
3. Initial seed – Начальное значение.

Пример использования блока и график его выходного сигнала представлен на рисунке 9.2.



**Рисунок 9.2 - Источник случайного сигнала с равномерным распределением**

### ***Элементы Simulink для выполнения данного задания:***

Источник линейно изменяющегося воздействия Ramp;

Источник случайного сигнала с равномерным распределением Uniform Random Number;

Источник случайного сигнала с нормальным распределением Random Number Integrator;

Блок скалярного умножения Dot Product;

Источник постоянного сигнала Constant;

Осциллограф Scope;

Блок сохранения данных в рабочей области To Workspace;

Блок считывания данных из рабочего пространства From Workspace;

Блок вычисления суммы Sum;

Блок умножения Product;

Блок вычисления тригонометрических функций Trigonometric Function;

Блок переключателя Switch Case.

### **Задания для выполнения:**

Построить модель-заготовку SIMULINK, моделирующую сигнал вида

$$s(t) = A_m E(t) \cos(\omega_0 t + \psi(t) + \varphi_0),$$

представляющий собой гармоническое колебание с несущей частотой  $\omega_0$  и амплитудой  $A_m$ , модулированное заданными медленно меняющимися функциями  $E(t)$ ,  $\psi(t)$ . Начальная фаза  $\varphi_0$  представляет собой случайную величину, равномерно распределенную в интервале  $]-\pi, \pi[$ .

На основе модели п.1 построить модель амплитудно-моделированного сигнала, задаваемого соотношением

$$E(t) = \begin{cases} 1 + m_{AM} \lambda(t), & \lambda(t) \geq -\frac{1}{m_{AM}} \\ 0, & \lambda(t) < -\frac{1}{m_{AM}} \end{cases} ;$$

$$\psi(t) = 0,$$

где  $\lambda(t)$  - стационарный гауссовский процесс с нулевым математическим ожиданием и заданной корреляционной функцией

$$\text{а) } R_\omega(\tau) = e^{-2|\tau|},$$

$$\text{б) } R_\omega(\tau) = e^{-\alpha|\tau|} \cos(\omega_1 \tau).$$

На основе модели п.1 построить модель фазо-модулированного сигнала, задаваемого соотношением

$$E(t) = 1;$$

$$\psi(t) = M_{\phi M} \lambda(t)$$

где  $\lambda(t)$  - стационарный гауссовский процесс с нулевым математическим ожиданием и заданной корреляционной функцией

$$\text{а) } R_{\omega}(\tau) = e^{-2|\tau|},$$

$$\text{б) } R_{\omega}(\tau) = e^{-\alpha|\tau|} \cos(\omega_1 \tau)$$

На основе модели п.1 построить модель частотно-модулированного сигнала, задаваемого соотношением

$$E(t) = 1;$$

$$\psi(t) = M_{\text{чМ}} \int_0^t \omega(t) dt$$

где  $\omega(t)$  - стационарный гауссовский процесс с нулевым математическим ожиданием и заданной корреляционной функцией

$$\text{а) } R_{\omega}(\tau) = e^{-2|\tau|},$$

$$\text{б) } R_{\omega}(\tau) = e^{-\alpha|\tau|} \cos(\omega_1 \tau).$$

### Контрольные вопросы

1. Когда физическая величина подчиняется нормальному распределению?
2. Из каких параметров состоит нормальное распределение?
3. Какие параметры имеет блок Random Number?
4. Чем отличается распределение Гаусса от равномерного распределения?
5. Какие параметры входят в блок Uniform Random Number?



## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Дьяконов, В.П. Математические пакеты расширения MATLAB. Специальный справочник. / В.П. Дьяконов, В.В. Круглов. – СПб.: Питер, 2001. – 480 с.
2. Потемкин, В.Г. Введение MATLAB / В.Г. Потемкин – М.: Диалог-МИФИ, 2000. – 247 с.
3. Солонина, А.И. Цифровая обработка сигналов. Моделирование в MATLAB / А.И. Солонина – СПб.: BHV, 2008. – 816 с.
4. Дьяконов, В.П. MATLAB R2006/2007/2008. Simulink 5/6/7. Основы применения / В.П. Дьяконов – М.: Солон, 2008. – 800 с.
5. Поршневу, С.В. MATLAB 7: Основы работы и программирования. Учебник / С.В. Поршневу – М.: Бином-Пресс, 2008. – 320 с.
6. Половко, А. М. MATLAB для студента / А. М. Половко, П. Н. Бутусов. – СПб.: БХВ-Петербург, · 2005. – 320 с.
7. Тихонов, В.И. Статистическая радиотехника / В.И. Тихонов. - М.: Радио и Связь, 1982.- 624 с.
8. Сергиенко, А.Б. Цифровая обработка сигналов / А.Б. Сергиенко. – СПб.: Питер, 2002. – 608 с.