

# Лабораторная работа №2 по курсу «Радиотехнические устройства и системы»

## Построение графиков в GNU/Octave

Кузнецов В.В., ассистент кафедры ЭИУ1-КФ

11 октября 2013 г.

### 1 Цель работы

Целью лабораторной работы является ознакомление с базовыми принципами построения графиков в системе численной математики GNU/Octave для применения с целью проведения расчётов различных радиотехнических устройств.

Система GNU/Octave — это высокоуровневый язык программирования, предназначенный прежде всего для численных расчётов. Он предоставляет удобный интерфейс командной строки для численного решения линейных и нелинейных задач, а также для выполнения других численных экспериментов. С помощью GNU/Octave можно решать задачи в том числе генерации и обработки сигналов. Установить GNU/Octave для Linux можно в один клик через пакетный менеджер, а для Windows её можно бесплатно скачать с сайта разработчика <http://octave.sourceforge.net>.

Octave работает в режиме командной строки. GNU/Octave позволяет выполнять операции с действительными и комплексными числами, матрицами, решать системы линейных уравнений, обрабатывать данные, строить графики и диаграммы. Синтаксис команд Octave близок к языку C и повторяет среду Matlab.

В ходе выполнения лабораторной работы необходимо ознакомиться с принципами построения графиков в среде GNU/Octave.

При подготовке руководства использовались материалы с сайтов <http://mydebianblog.blogspot.com> (на русском языке) и [http://en.wikibooks.org/wiki/Octave\\_Programming\\_Tutorial/Getting\\_started](http://en.wikibooks.org/wiki/Octave_Programming_Tutorial/Getting_started) (на английском языке).

### 2 Построение графиков в Octave

#### 2.1 Одиночные графики

Для построения графиков в GNU/Octave используется два типа функций Matlab-совместимые и Gnuplot-совместимые. Далее будут рассмотрены Matlab-совместимые функции.

Для построения графиков служит функция `plot(x,y)`. Параметрами функции `plot` являются две матрицы-строки или матрицы столбца (вектора), содержащие координаты точек графика по  $x$  и  $y$  соответственно. Оба вектора должны иметь одинаковую длину (количество элементов). Вектора удобно генерировать с помощью функции `linspace` (см. лабораторную работу №1).

Построим график синусоидального сигнала с частотой 5 Гц. Для этого введём в командной строке Octave следующие команды:

```
octave:1> t=linspace(0,1,200); # вектор из 200 точек от 0 до 1 сек.
octave:2> f=5; # частота 5 Гц
octave:3> s=sin(2*pi*f*t); # синусоидальный сигнал
octave:4> plot(t,s); # строим график
```

В результате открывается графическое окно, в котором построена синусоида.

## 2.2 Несколько графиков в одном окне

Для построения нескольких графиков служит функция `plot(x1,y1,тип_линии,x2,y2,тип_линии)`. Параметры `x1`, `y1`, `x2`, `y2` — такие же векторы как и для одиночного графика. Тип линии — это строка специального вида, заключённая в двойные кавычки, которая устанавливает индивидуальный тип линии каждый для своего графика.

Например:

```
"-" — сплошная линия;
"--" — штриховая линия;
```

Построим дополнительно к графику из раздела 2.1 ещё один график. Для этого выполним следующие команды:

```
octave:5> f1=10; # частота второго сигнала 10 Гц
octave:6> s1=0.5*sin(2*pi*f1*t); # второй синусоидальный сигнал
octave:7> plot(t,s,"-",t,s1,"--"); # строим графики. По оси X - время. По оси Y
- напряжение. Первый график сплошной линией, а второй - штриховой.
```

В результате в графическом окне строятся две синусоиды (с частотой 5 Гц и 10 Гц). Одна — штриховой линией, а другая — сплошной.

## 2.3 Подписи по осям, заголовок графика и легенда

Для того, чтобы создать заголовок графика служит функция `title(str)`. Параметром функции является строка `str`, заключённая в двойные кавычки " ", как в языке C.

Чтобы создать подписи по осям  $x$  и  $y$ , служат функции `xlabel(str)` и `ylabel(str)` соответственно. Параметрами этих функций является строка `str`, заключённая в двойные кавычки " ", как в языке C.

Легенду создаёт функция `legend(str1,str2, ...)`. Функция должна иметь не менее одного параметра, каждый из которых является строкой. Каждая строка относится к своему графику.

```
octave:8> title("Sinusoidal voltage"); # Заголовок графика
octave:9> xlabel("time, sec"); # по оси X время
octave:10> ylabel("Voltage, V"); # по оси Y напряжение
octave:11> legend("Freq= 5 Hz","Freq=10 Hz"); # Легенда. Подпишем каждый график
```

В результате выполнения всех команд получается график синусоидальных напряжений с подписями по осям  $X$ ,  $Y$ , легендой и заголовком. (см. рис.1)

## 2.4 Разделение графического окна на части

Для разбиения графического окна на части служит функция `subplot(m, n, k)`

Графическое окно разбивается на  $m$  подграфиков по оси  $x$  и на  $n$  подграфиков по оси  $y$  и осуществляется переход в график с номером  $k$ . Всего графическое окно разбивается на

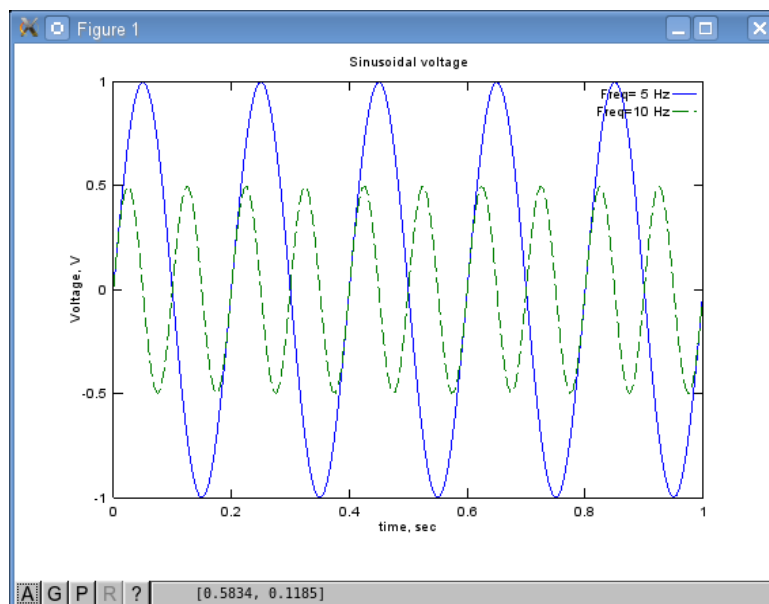


Рис. 1. Одиночный график в системе Octave

$m \cdot n$  графиков. Можно разбивать графическое окно в любых комбинациях. В том числе допускается  $m = 1$  и  $n = 1$ .

Графики нумеруются как показано на рис.2. Пример нумерации для разбиения окна 2x2 показан на рис.3

1	2	...	m
$m + 1$	$m + 2$	...	$m + m$
...	...	...	...
$m(n - 1) + 1$	$m(n - 1) + 2$	...	$m \cdot n$

Рис. 2. Схема нумерация подграфиков для команды subplot.

1	2
3	4

Рис. 3. Схема нумерация подграфиков для команды subplot(2,2,1).

Рассмотрим пример. Построим график четырёх синусоид в на четырёх подграфиках:

```
octave:12> f1=4;# Частота 1-й синусоиды 4 Гц
octave:13> f2=6;# 2-й синусоиды -- 6 Гц
octave:14> f3=10; # 3-й -- 10 Гц
octave:15> f4=1; # 4-й -- 1 Гц
octave:16> s1=sin(2*pi*f1*t); # 1-й график
octave:17> s2=sin(2*pi*f2*t); # 2-й график
octave:18> s3=sin(2*pi*f3*t); # 3-й график
octave:19> s4=sin(2*pi*f4*t); # 4-й график
octave:20> subplot(2,2,1); # делим графическое окно на 4 части 2x2
# и переходим в первое подокно
```

```

octave:21> plot(t,s1); # строим график
octave:22> title("4 Hz"); # подпишем график
octave:23> subplot(2,2,2);# переходим во 2-е подокно
octave:24> plot(t,s2);# строим график
octave:25> title("6 Hz");# подпишем график
octave:26> subplot(2,2,3);# переходим в 3-е подокно
octave:27> plot(t,s3);# строим график
octave:28> title("10 Hz");# подпишем график
octave:29> subplot(2,2,4);# переходим в 4-е подокно
octave:30> plot(t,s4);# строим график
octave:31> title("1 Hz");# подпишем график

```

В результате выполнения этих команд мы получим четыре графика в графическом окне, разделённом на 4 части 4.

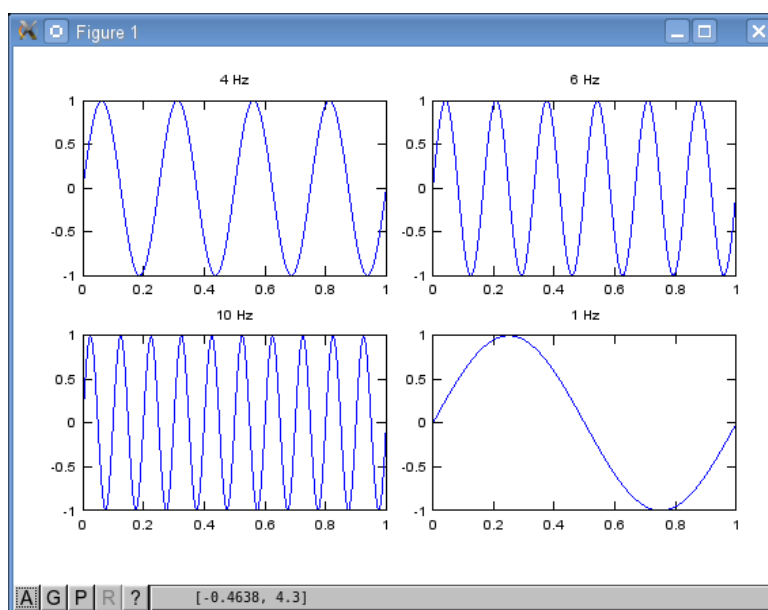


Рис. 4. Несколько графиков в одном окне в системе Octave с помощью команды subplot

## 2.5 Гистограмма

В GNU/Octave возможно построение гистограммы. Для этого служит функция `hist(X,N)`

Параметр `X` представляет собой вектор, содержащий числа, для которых нужно построить гистограмму.

Параметр `N` представляет собой число интервалов разбиения.

Рассмотрим пример. Построим гистограмму для вектора, содержащего 10000 случайных чисел. Для генерации случайных чисел служит функция `randn`.

```

octave:12> X=randn(10000,1); # генерируем вектор из 10000 случайных чисел
octave:13> hist(X,30); # строим гистограмму распределения случайных чисел

```

В результате открывается графическое окно, в котором построена гистограмма (см. рис.5)

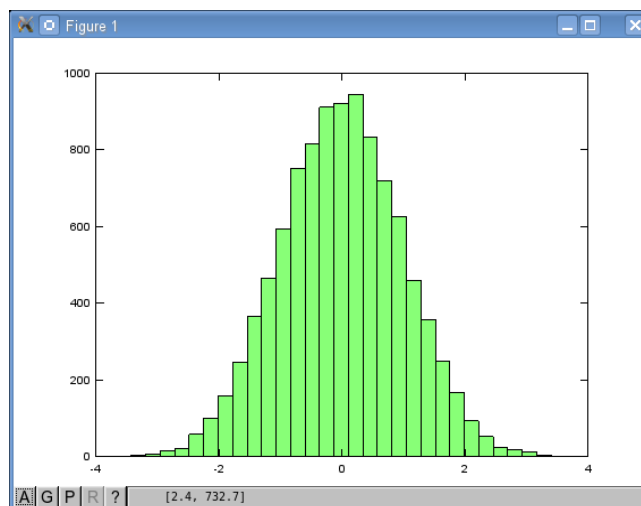


Рис. 5. Построение гистограммы при помощи команды `hist`

### 3 Экспорт графиков

Графики можно экспортировать в различные графические форматы. Для этого служит команда

```
print("имя_файла", "формат_вывода")
```

Параметр `имя_файла` содержит строку, заключённую в двойные кавычки, в которой записано имя файла, в который будет сохранено содержимое графического окна. Для ОС Windows здесь должен быть указан полный путь. Например `C:\mega_graphic.png`. При этом символ `\` в строке пути обязательно нужно удваивать, иначе Octave воспринимает его как управляющий символ.

Параметр `формат_вывода` является специальной строкой, которая содержит указание в каком графическом формате будет сохранён график. Доступны следующие варианты (перечислены только основные форматы, о дополнительных форматах см. официальную документацию):

- "-dps" — формат Postscript (векторная графика);
- "-deps" — формат Encapsulated Postscript (векторная графика);
- "-dtikz" — формат векторной графики TikZ для LaTeX;
- "-dpng" — PNG (растровая графика);
- "-dgif" — GIF (растровая графика);
- "-djpeg" — JPEG (растровая графика);
- "-dpdf" — PDF (векторная графика);

Рассмотрим пример. Выполним команду (графическое окно при этом должно быть открыто):

```
octave:12> print("sinusoid.png", "-dpng");
```

Для ОС Windows в первый параметр нужно подставить полный путь: например `"C:\\sinusoid.png"`.

В результате исполнения команды в текущем каталоге создаётся файл `sinusoid.png`, который содержит копию изображения в графическом окне.

### 4 Заключение

В результате выполнения лабораторной работы произведено ознакомление с принципами визуализации информации в системе GNU/Octave.