

# Семинар №1 по курсу «Основы цифровой обработки сигналов»

## Математическая модель сигнала

Кузнецов В.В., ассистент кафедры ЭИУ1-КФ

2 августа 2013 г.

## 1 Цель работы

Целью семинара является построение осциллограмм различных радиотехнических сигналов на основе математической модели сигнала при помощи программы Gnuplot.

## 2 Введение в Gnuplot

Если его теперь запустить, вызвав в консолике gnuplot то он нас поприветствует примерно таким образом:

```
G N U P L O T Version 4.0 patchlevel 0 last modified Thu Apr 15 14:44:22 CEST 2004
System: Linux 2.6.15.7
```

```
Copyright (C) 1986 - 1993, 1998, 2004 Thomas Williams, Colin Kelley and many others
```

```
This is gnuplot version 4.0. Please refer to the documentation for command syntax
changes. The old syntax will be accepted throughout the 4.0 series, but all save files use
the new syntax.
```

```
Type 'help' to access the on-line reference manual. The gnuplot FAQ is available from
http://www.gnuplot.info/faq/
```

```
Send comments and requests for help to
```

```
Send bugs, suggestions and mods to
```

```
Terminal type set to 'x11' gnuplot>
```

То есть запустилась интерактивная часть программы. Здесь нам предлагают набирать команды и сразу видеть их результат - что не очень удобно, особенно если вы в первый раз работаете с gnuplot.

Но у gnuplot есть и другой режим работы - потоковый. На мой взгляд, для освоения и работы он более эффективен. Для того, чтобы быстро начать с ним работу и сразу посмотреть, на что он способен, мы построим небольшой, но показательный график. Всё, что нам потребуется - это ваш любимый текстовый редактор с подсветкой синтаксиса и немного терпения.

Приступим В потоковом режиме работы gnuplot воспринимает сразу все команды, которые вы ему пошлёте, и сразу строит график по данным из файла. В ряде случаев это удобнее, а если требуется строить множество графиков, то потоковый режим работы GNUplot вас просто спасёт.

Итак, ниже я привожу разбор построения типичного графика, параметры которого записанного в виде скрипта, что выделено тёмно-красным цветом, а комментарии серым

цветом. Пишите в файл строчки так, как они написаны здесь, без переносов и абзацев - это важно!

Создаём новый файл, называем, например, `simple.graph` и начинаем в него записывать команды `gnuplot`. Признаком того, что файл относится к скрипту `GNuplot`, является первая строка в файле: Далее выведем график в виде файла формата постскрипт (который можно скормить `LaTeX`) `set terminal postscript eps enhanced` задаётся имя выводимого файла с графиком, имя в кавычках `set output -/RMSresearch.ps` Ставим поддержку кириллицы, а то не увидим в подписях `set encoding koi8r` Устанавливаем метки по осям, размер и тип шрифта: сейчас по оси `X` `set xlabel` Устанавливаем метки по осям, размер и тип шрифта: теперь по оси `Y` `set ylabel "Величина СКО, отн.ед"` `font "Helvetica,18"` Устанавливаем предел значений по оси `Y` `set yrange [0:1]` Задаём стиль линии (`line1`): сплошная прямая (`lt` - от `linetype`) с круглыми точками (`pt` - от `pointstyle`) типа 7 `set style line 1 lt 1 pt 7` Теперь, задав все параметры, строим график, беря данные из файла `RMSresult` (для примера), где мы берём только второй столбец. Легенду для данных второго столбца называем `СКО` (среднеквадратичное отклонение) и заставляем строить график, используя тип линии, который мы задали. `plot -/RMSresult"using 2 title "СКО"with linespoints linestyle 1`

Осталось привести содержимое файла `RMSresult` - это просто поток чисел, разделённых пробелами (снятые параметры физического эксперимента):

```
1.0000000e-04 6.1881100e-01
1.0000000e-03 5.5289577e-01
1.0000000e-02 3.5527980e-01
1.0000000e-01 2.4352333e-01
1.0000000e+00 2.5852181e-01
1.0000000e+01 2.5291911e-01
1.0000000e+02 2.5109706e-01
1.0000000e+03 2.5524945e-01
1.0000000e+04 2.6013621e-01
1.0000000e+05 2.4596297e-01
```

Теперь, когда у нас есть всё необходимое для строительства графика, в командной строке пишем: Всё, должен появиться график `RMSresearch.ps`, который можно просмотреть любым графическим редактором. Выглядеть он должен примерно так:

Ну вот, мы и построили первый график в `gnuplot`. Причём, график довольно навороченный - с греческими буквами в подписи, правильно подписанными осями и прочим.

### 3 Математическая модель сигнала

Сигналом называют процесс изменения во времени физического состояния какого-либо объекта, служащий для отображения, регистрации и передачи сообщений. Сигналом могут быть например напряжение, ток, напряжённость поля. В большинстве случаев носителями радиотехнических сигналов являются электромагнитные колебания. Рассмотрим классификацию радиотехнических сигналов.

Математической моделью сигнала обычно служит функциональная зависимость, аргументом которой является время. Например зависимость напряжения в цепи от времени.

Если физический процесс, порождающий сигнал развивается во времени таким образом, что его можно измерять в любые моменты времени, то сигнал такого класса называется *аналоговым*. Аналоговый сигнал можно наглядно представить его графиком

во времени — осциллограммой. Осциллограмма может быть непрерывной или иметь точки разрыва.

Примером сигналов, широко применяемых в радиотехнике являются модулированные сигналы. Модулированные сигналы применяются для передачи информации по радиоканалу. Используются сигналы с амплитудной, частотной и фазовой модуляцией.

В амплитудно-модулированном (АМ) сигнале в такт с модулирующим сигналом меняется амплитуда несущей. Математическая модель (зависимость уровня сигнала от времени) для АМ-сигнала имеет вид:

$$u(t) = U(t) \cos(\omega_0 t + \varphi_0) \quad (1)$$

Где  $U(t)$  — огибающая;

При амплитудной модуляции связь между огибающей и модулирующим полезным сигналом  $s(t)$  выражается уравнением:

$$U(t) = U_m [1 + M s(t)] \quad (2)$$

Где  $U_m$  — амплитуда несущей в отсутствии модулирующего сигнала,  $M$  — коэффициент амплитудной модуляции.

При частотной модуляции (ЧМ) амплитуда несущей остаётся неизменной, а в такт с модулирующим сигналом изменяется частота несущей. ЧМ сигнал описывается следующим уравнением (где  $M$  — индекс модуляции):

$$u(t) = U_m \cos[(\omega_0 + M s(t))t] \quad (3)$$

При фазовой модуляции (ФМ) в зависимости от амплитуды модулирующего сигнала  $s(t)$  меняется начальная фаза сигнала. ФМ сигнал описывается следующим уравнением (где  $M$  — индекс модуляции).

$$u(t) = U_m \cos(\omega_0 t + M s(t)) \quad (4)$$

В данные выражения везде входит круговая частота  $\omega$ , которая связана с частотой  $f$  как:

$$\omega = 2\pi f \quad (5)$$

## 4 Задание для выполнения

В процессе выполнения семинара необходимо построить в программе Gnuplot осциллограммы АМ, ЧМ и ФМ сигналов в соответствии с заданной частотой несущей  $f$ , частотой модулирующего сигнала  $f_m$  и коэффициентом модуляции  $M$ . Индекс модуляции для ЧМ и ФМ подобрать по наилучшей наглядности получившейся осциллограммы.

Исходные данные для выполнения задания приведены в таблице 1.

Амплитуду сигналов везде принять равной 1 В.

Пример скрипта Gnuplot, для выполнения задания приведён в листинге 1. Gnuplot предназначен для построения инженерных и научных графиков и распространяется бесплатно. Программа работает под управлением ОС Linux и Windows. Родной ОС для неё является Linux и на этой платформе Gnuplot обладает наивысшей стабильностью и скоростью работы. Gnuplot входит во все современные дистрибутивы Linux и его можно установить в один клик через пакетный менеджер. Скачать установщик Gnuplot для Windows можно с официального сайта [gnuplot.sourceforge.net](http://gnuplot.sourceforge.net)

Таблица 1: Исходные данные

№варианта	$f$ , Гц	$f_m$ , Гц	$M$
1	100000	10000	0.1
2	120000	8000	0.2
3	90000	6000	0.8
4	100000	12000	0.3
5	160000	11000	0.4
6	180000	10000	0.75
7	200000	9000	0.66
8	110000	10000	1.2
9	150000	10000	0.5
10	160000	15000	0.8
11	180000	12000	0.6
12	220000	22000	0.66
13	200000	20000	0.15
14	220000	20000	1.1
15	160000	18000	1.5

Листинг 1: Пример скрипта Gnuplot для выполнения задания.

```

1  #!/usr/bin/gnuplot -persist
2  # Задаём параметры сигнала в виде переменных.
3  # Здесь нужно вписать свои параметры по вариантам.
4  M_AM=0.55 #коэффициент модуляции АМ - сигнала
5  M_FM=20000 #коэффициент модуляции ЧМ - сигнала
6  M_PM=1.5 #коэффициент модуляции ФМ - сигнала
7  f=200000 #частота несущей, Гц
8  fm=10000 #частота модулирующего сигнала, Гц
9  # задаём функции графики которых будем строить. Синтаксис -- как в
   языке C
10 s(x)=cos(2*pi*fm*x) #модулирующий сигнал, вместо переменной времени t
   используем
11 переменную x
12 U(x)=1+M_AM*s(x) # огибающая
13 U_RF(x)=cos(2*pi*f*x) # несущая
14 U_AM(x)=U(x)*U_RF(x) # АМ - сигнал
15 U_FM(x)=cos(2*pi*(f+M_FM*s(x))*x) # ЧМ - сигнал
16 U_PM(x)=cos(2*pi*f*x+M_PM*s(x)) # ФМ - сигнал
17 set multiplot layout 4,1 # разбиваем окно на 4 графика по вертикали и
   один
18 # график по горизонтали
19 #set terminal png # раскомментировать эту строку если хотим чтобы
   график вывелся
20 # не на экран а в файл PNG. Ещё можно выводить график в файлы PDF,SVG
   ,EPS
21 set xrange [0:200e-6] #отрезок времени по оси x 0 - 200 мксек
22 set yrange [-3:3] #растяжка по Y от -3 до 3
23 set xlabel "Время, сек" # Подпись оси X
24 set format x "%.1e" # У чисел на оси X оставить один знак после

```

```

запятой
25 set ylabel "Напряжение, В" # Подпись по оси Y
26 set ytics 2 # Два деления по оси X
27 #set output "1.png" # раскомментировать эту строку если хотим чтобы
    график
28 вывелся
29 # не на экран а в файл PNG. Ещё можно выводить график в файлы PDF,SVG
    ,EPS
30 # График выведется в файл с именем 1.png в текущем каталоге
31 # set sample 1000 # Пользователям Windows раскомментировать эту
    строку
32 plot s(x) title "Огибающая" # строим график Огибающей. title задаёт
    легенду
33 plot U_AM(x) title "АМ сигнал" # строим график АМ-сигнала
34 plot U_FM(x) title "ЧМ сигнал" # строим график ЧМ-сигнала
35 plot U_PM(x) title "ФМ сигнал", U_RF(x) title "несущая"
36 # Строим график ФМ сигнала и несущей
37 # на одном графике. Функции график которых хотя построить указаны
    через запятую

```

Несмотря на то, что Gnuplot не имеет графического интерфейса пользователя, пользоваться им очень просто.

Gnuplot может работать в интерактивном режиме и команды из скрипта можно вводить прямо в консоль Gnuplot. Но удобнее заранее записать команды в скрипт и затем передать этот скрипт Gnuplot'у.

Скрипт из листинга набирается в любом текстовом редакторе, который может сохранять текст с кодировкой UTF-8 (Unicode). Подойдёт даже обычный Notepad. Не следует использовать редактор Word и ему подобные офисные программы, так как он сохраняет текст в бинарный формат. Из текстового редактора нужно сохранить скрипт в текстовый файл, например с именем Sem1.plt. Кодировку нужно обязательно выбрать UTF-8 для избежания проблем с русскими буквами. Затем пользователям Linux нужно сделать файл исполняемым и запустить его как программу. Сделать это можно как из консоли, так и в файловом менеджере. В консоли для этого нужно выполнить следующие две команды:

```

chmod a+x Sem1.plt
./Sem1.plt

```

Первая команда устанавливает права на исполнения файла, а вторая запускает скрипт. Первую команду нужно выполнить только один раз.

В системе Windows нужно запустить ярлык Gnuplot на рабочем столе и выбрать в меню File->Open наш скрипт.

В результате выполнения скрипта появится окно с четырьмя графиками, или, если мы раскомментируем строки 19 и 27, то графики автоматически сохранятся в файл PNG с заданным именем.