# «Аналого-цифровое преобразование. Однородное квантование. Представление сигнала во временной и частотной областях.»

***Цель работы:*** Формирование навыков работы с дискретными сигналами в среде GNU Octave.

***Задачи лабораторной работы:***

* Формирование дискретных сигналов.
* Построение временной формы сигнала и его спектра.
* Квантование дискретных сигналов.

**Теоретический материал**

GNU Octave – это программное обеспечение для математических расчетов и моделирования, имеющее свой совместимый с Matlab высокоуровневый язык программирования. Для работы в нем необходимо понимание общих принципов работы с переменными, векторами и матрицами. В языке Octave доступны все базовые математические операции, а если их недостаточно, можно создавать свои функции с использованием циклов и условных операторов.

Для формирования дискретного сигнала в первую очередь необходимо задать вектор с временными отсчетами. Он определяет моменты взятия выборок непрерывного сигнала. Одна из наиболее существенных составляющих ошибки при измерениях с помощью АЦП – погрешность квантования – является результатом самого процесса преобразования. Погрешность квантования – это погрешность, вызванная значением шага квантования и определяемая как 1/2 величины наименьшего значащего разряда (LSB). Она не может быть исключена в аналого-цифровых преобразованиях, так как является неотъемлемой частью процесса преобразования, определяется разрешающей способностью АЦП и не меняется от АЦП к АЦП с равным разрешением.

Передаточная характеристика АЦП – это функция зависимости кода на выходе АЦП от напряжения на его входе. Такой график представляет собой кусочно-линейную функцию из 2N "ступеней", где N - разрядность АЦП. Каждый горизонтальный отрезок этой функции соответствует одному из значений выходного кода АЦП (см. рис. 1). Если соединить линиями начала этих горизонтальных отрезков (на границах перехода от одного значения кода к другому), то идеальная передаточная характеристика будет представлять собой прямую линию, проходящую через начало координат.

Рис. 2 иллюстрирует идеальную передаточную характеристику для 3-х разрядного АЦП с контрольными точками на границах перехода кода. Выходной код принимает наименьшее значение (000b) при значении входного сигнала от 0 до 1/8 полной шкалы (максимального значения кода этого АЦП).



Рисунок 1 – Идеальная передаточная характеристика 3-разрядного АЦП.



Рисунок 2 – Передаточная характеристика 3-разрядного АЦП со смещением на -1/2 LSB для уменьшения погрешности квантования на 1/2 LSB.

Также следует отметить, что АЦП достигнет значения кода полной шкалы (111b) при 7/8 полной шкалы, а не при значении полной шкалы. Таким образом, переход в максимальное значение на выходе происходит не при напряжении полной шкалы, а при значении, меньшем на наименьший значащий разряд (Least Significant Bit, LSB), чем входное напряжение полной шкалы. Передаточная характеристика может быть реализована со смещением -1/2 LSB. Это достигается смещением передаточной характеристики влево, что смещает погрешность квантования из диапазона -1... 0 LSB в диапазон -1/2 ... +1/2 LSB.

При квантовании с разрядностью 2 бита со смещением используется 4 уровня, поэтому диапазон амплитуд на входе квантователя можно разделить на 4 зоны. В каждой из них амплитуда сигнала должна устремиться к среднему значению. Данный процесс изображен на рисунке 3.



Рисунок 3 – Двухбитное квантование сигнала со смещением.

Рассмотрим процесс дискретизации на примере формирования гармонического сигнала и сигнала с линейной частотной модуляцией (ЛЧМ). Ход работы представлен на рисунке 1.



Рисунок 4 – Этапы выполнения работы.

**Ход выполнения работы**

1) Формирование дискретного сигнала.

Создайте новый проект в OCTAVE. Запишите следующие команды для очистки предыдущих значений:

clc

close all

clear all

pkg load signal

pkg load communications

set(0,'DefaultAxesFontSize',10);

Первая строка очищает командную окно, вторая закрывает все открытые окна, третья – очищает область переменных. Затем идет подключение пакетов «signal» и «communications», которые необходимы при вызове функций chirp() и quantiz(). Команда set() позволяет установить параметры шрифта при построении графиков с отображением текста.

Далее нужно задать основные параметры: частоту дискретизации, длительность символа и вектор временных значений t. Он задается диапазоном чисел от 1/Fs с шагом 1/Fs до длительности символа T.

Теперь перейдем к расчету отсчетов сигнала. Гармонический сигнал задается формулой

(1)

где – амплитуда сигнала, – угловая частота, – отсчеты по времени, – начальная фаза, а – частота гармоники.

ЛЧМ сигнал – это гармонический сигнал, частота которого линейно возрастает (или убывает) с течением времени:

(2)

Для формирования ЛЧМ сигнала в Octave используется функция chirp(). Синтаксис функции подразумевает указание четырех основных параметров (T, F0, T1, F1). T – это временной массив, F0 – частота в начале символа, T1 и F1 – значение времени и частоты в конце символа.

В общем виде данный фрагмент программы выглядит следующим образом:

%% 1 Формирование дискретного сигнала

fs = 128e3; % частота дискретизации

T = 12800/fs; % длительность символа

t = 1/fs:1/fs:T; % вектор временных значений

% формирование гармонического сигнала

s1 = ???;

% формирование ЛЧМ сигнала

s2 = chirp(t, 0, T, 5e3);

Сигнал s1 задайте по формуле 1 со следующими параметрами:

Амплитуда – 1 В;

Начальная фаза – 0 рад;

Частота – 5 кГц.

Для записи числа используется pi.

После выполнения это части программы интерпретатором будет сформировано две переменные, а также три вектора (t, s1 и s2) длиной 12800 отсчетов.

2) Построение временной формы сигнала.

Для построения графиков можно использовать следующую конструкцию:

%% 2 Построение временной формы сигнала

figure('Position',[100 100 360 480])

subplot(2,1,1)

plot(t,s1)

grid on

xlim([0 0.001])

ylim([-1.5 1.5])

title('Гармонический сигнал')

xlabel('Время, с')

ylabel('Амплитуда, В')

subplot(2,1,2)

plot(t,s2)

grid on

xlim([0 0.05])

ylim([-1.5 1.5])

title('ЛЧМ сигнал')

xlabel('Время, с')

ylabel('Амплитуда, В')

Разберем представленный код. Первая команда создает новое графическое окно размером 360х480 пикселей, нижний левый угол которого находится на расстоянии 100 пикселей вправо и 100 пикселей вверх от крайней нижней левой точки экрана (где расположена кнопка Пуск).

Команда subplot(2,1,1) используется для разбиения области построения графиков на несколько элементов, в данном случае они расположены в две строки и один столбец. Третья цифра указывает на текущую область построения, то есть следующая команда для построения графика выведет рисунок в окне с номером 1.

Команда plot(t,s1) позволяет построить график по векторам. Ось X представлена вектором t, ось Y – вектором s1. График строится точками по умолчанию, соединенными прямыми линиями. Команда grid on включает сетку на полученном графике.

Следующие команды относятся к форматированию графического окна. xlim() и ylim() устанавливают ограничение области построения по осям X и Y. Команда title() используется для подписи окна, а команды xlabel() и ylabel() – для подписи осей.

2) Расчет и построение спектра.

Для расчета спектральных составляющих используется дискретное преобразование Фурье. Чтобы построить полученное спектральное представление сигнала, нужно также рассчитать частотную шкалу. Для отображения амплитуды гармоник в соответствии с временным сигналом (то есть чтобы привести шкалу в Вольты) рассчитанные спектральные отсчеты нормируются путем умножения на два и деления на длину массива. Чтобы шкала полученных графиков соответствовала дБВ, от полученных значений необходимо взять 20\*log10.

%% 3 Расчет и построение спектра

k = 4;

fscale = -fs/2:fs/(length(t)\*k):fs/2-fs/(length(t)\*k); % частотная шкала

% гармонический сигнал

spec1 = fftshift(abs(fft(s1,length(t)\*k)));

spec1 = spec1\*2/length(t);

% ЛЧМ сигнал

spec2 = ???;

spec2 = ???;

% построение спектра гармонического сигнала

figure('Position',[460 100 360 480])

subplot(2,1,1)

plot(fscale,20\*log10(spec1))

grid on

xlim([-fs/2 fs/2])

ylim([-100 10])

title('Спектр гармонического сигнала')

xlabel('Частота, Гц')

ylabel('Амплитуда, дБВ')

% построение спектра ЛЧМ сигнала

subplot(2,1,2)

plot(fscale,???)

grid on

ylim([-100 10])

xlim([-fs/2 fs/2])

title('Спектр ЛЧМ сигнала')

xlabel('Частота, Гц')

ylabel('Амплитуда, дБВ')

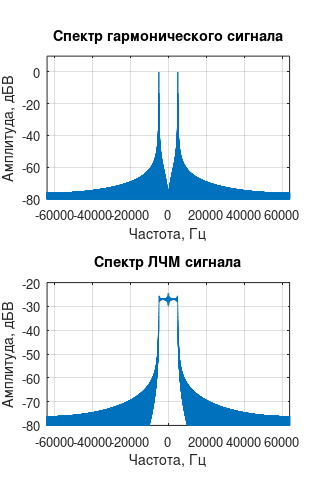
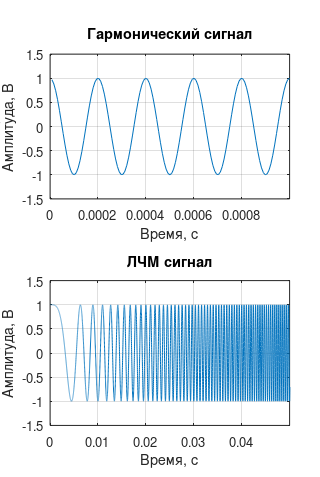


Рисунок 5 – Временное и спектральное представление сигналов.

4) Квантование.

Для квантования используется функция quantiz(). На вход функции подается дискретный сигнал и массив квантованных уровней. В примере ниже уровень соответствует сигналу амплитудой от -1 до 1. Полученные значения квантованного сигнала делятся на 2^(B-1), затем из результата вычитается 1. Если требуется квантование со смещением, описанное ранее, то к результату прибавляется смещение на половину шага квантования, то есть 1/2^B.

%% Квантование

B = 2;

table = (1:2^B)/2^(B-1)-1;

s1q = quantiz(s1, table)/2^(B-1)-1+1/2^B;

s2q = ???;

5)-6) Данные пункты выполните самостоятельно. Необходимо построить временную и спектральную форму квантованных сигналов. В результате должны получиться следующие графики:

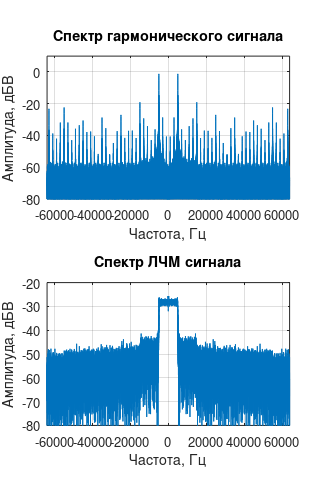
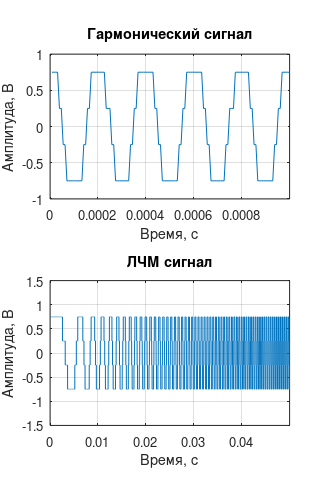


Рисунок 6 – Временное и спектральное представление сигналов после квантования.

**Работа с инструментом**

Для связи с генератором DG5071 и MSO5104 необходимы файлы скриптов MATLAB. Файлы скриптов можно получить двумя способами:

1. Скачать по ссылкам [DG5071\_код](https://raw.githubusercontent.com/AndrewMZ6/Instrument_control/main/MATLAB/DG_waveform_generator/DG.m) и [MSO5104\_код](https://raw.githubusercontent.com/AndrewMZ6/Instrument_control/main/MATLAB/MSO_oscilloscope/MSO.m) нажав правой кнопкой мыши и выбрав опцию «Сохранить как» («Save as») и назвав файлы DG.m и MSO.m соответственно.
2. Создать файлы в MATLAB с названиями DG.m и MSO.m а затем скопировать в эти файлы код из ссылок, приведенных в пункте 1.

**Размещение файлов в рабочей папке**

Для использования генератора и осциллографа, необходимо поместить файлы DG.m и MSO.m в папку, в которой находится скрипт, использующий эти файлы. Например, если такой скрипт называется main\_script.m, то файловая структура выглядит так, как показано на рисунке 1

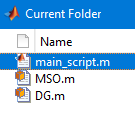


Рисунок 7 – Файловая структура рабочей папки

Вызов функций из файлов MSO.m и DG.m осуществляется обращением к ним по имени, рисунок 2

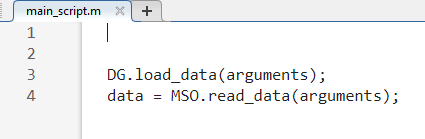


Рисунок 8 – Обращение к файлам DG.m и MSO.m

**Генератор DG5071**

**DG.load\_data**(connection\_ID, load\_data, Fs, amplitude)

connection\_ID – идентификатор инструмента.

load\_data – загружаемый вещественный сигнал.

Fs – частота дискретизации генератора (25 МГц, 125 МГц, 250 МГц, 500 МГц).

amplitude – значение амплитуды, выставляемые на инструменте в Вольтах.

**Осциллограф MSO5104**

data = **MSO.read\_raw\_bytes\_fs**(connection\_ID, channel\_number, points, Fs);

connection\_ID – идентификатор инструмента. Пример идентификатора инструмента:

connection\_ID = 'USB0::0x1AB1::0x0515::MS5A244909354::0::INSTR';

channel\_number – номер канала осциллографа.

points – количество точек, загруженных с осциллографа.

Fs – требуемая частота дискретизации осциллографа. Доступны следующие частоты дискретизации: 2 МГц, 5 МГц, 10 МГц, 20 МГц, 50 МГц, 100 МГц, 200 МГц, 500 МГц, 1 ГГц, 2 ГГц, 4 ГГц, 8 ГГц.

data – принятые данные с осциллографа.

**Загрузка данных на генератор**

Зададим начальные параметры сигнала, который будет загружен на генератор:

Таблица 1 – Параметры сигнала

|  |  |  |
| --- | --- | --- |
| Параметр | Название | Значение |
| fc | Частота несущей | 150 МГц |
| fs | Частота дискретизации | 500 МГц |
| points\_num | Количество точек | 16383 |

Код формирования сигнала синусоиды приведен ниже

% Формирование сигнала для загрузки на генератор

fc = 150e6;

fs = 500e6;

ts = 1/fs;

points\_num = 16383;

t = 0:ts:(points\_num – 1)\*ts;

signal = sin(2\*pi\*fc\*t);

Для построения графика нам необходимо сформировать частотную ось

% Формирование частотной оси для построения графика

freq\_step = fs/points\_num;

freq\_line = freq\_step:freq\_step:fs;

centered\_freq\_line = (freq\_line - fs/2)/1e6;

Построим график сформированного сигнала

% Построение графика сформированного сигнала

figure;

plot(centered\_freq\_line, abs(fftshift(fft(signal))));

xlabel('Частота, МГц');

grid on;

title('Спектр загружаемого сигнала');



Рисунок 9 – Спектр сформированного сигнала

% Загрузка сформированного сигнала на генератор

dg\_conn\_ID = 'USB0::0x1AB1::0x0640::DG5S244900056::0::INSTR';

amp = 0.5;

DG.load\_data(dg\_conn\_ID, signal, fs, amp);

**Получение данных с осциллографа**

Зададим параметры, используемые для получения данных с осциллографа

Таблица 2 – Параметры сигнала

|  |  |  |
| --- | --- | --- |
| Параметр | Название | Значение |
| channel\_num | Номер канала осциллографа | 1 |
| fs\_oscilloscope | Частота дискретизации осциллографа | 500 МГц |
| points\_num | Количество точек | 100 тыс. |

% Параметры получения данных с осциллографа

osci\_conn\_ID = 'USB0::0x1AB1::0x0515::MS5A244909354::0::INSTR';

channel\_num = 1;

points\_num = 100e3;

fs\_oscilloscope = 500e6;

Используя приведенные параметры можно произвести операцию получения данных с осциллографа

% Операция получения данных с осциллографа

data = MSO.read\_raw\_bytes\_fs(osci\_conn\_ID, channel\_num, points\_num, fs\_oscilloscope);

Для построения спектра нам снова необходимо сформировать частотную ось

% Формирование частотной оси для построения графика

freq\_step = fs\_oscilloscope/length(data);

freq\_line = freq\_step:freq\_step:fs\_oscilloscope;

centered\_freq\_line = (freq\_line - fs\_oscilloscope/2)/1e6 ;

Построим спектр принятого сигнала

% Построение спектра принятого сигнала

figure;

plot(centered\_freq\_line, abs(fftshift(fft(data))));

xlabel('Частота, МГц');

grid on;

title('Спектр принятого сигнала');



Рисунок 10 – Спектр принятого сигнала

Установите частоту дискретизации осциллографа fs\_oscilloscope 200 МГц и примите данные с осциллографа. Обратите внимание на то, как изменится частота принятого сигнала, рисунок 5.



Рисунок 11 – Спектр принятого сигнала

В ходе работы должны быть выполнены следующие операции:

* Формирование дискретных сигналов;
* Однородное квантование сигналов;
* Построение временной и спектральной формы сигналов до квантования и после;

**Контрольные вопросы лабораторной работы**

1. Что такое квантование?

2. Как построить временную форму сигнала и его спектр?

3. По полученным графикам определите, какому уровню соответствует шум квантования ЛЧМ сигнала при использовании двух, четырех и шести-битного АЦП на частоте 40 кГц.

**Задание на самостоятельную работу:**

Определите сигнал s3 как сумму s1 и s2, постройте временную и спектральную форму данного сигнала до и после квантования.

**Листинг программы**

clc

close all

clear all

pkg load signal

pkg load communications

set(0,'DefaultAxesFontSize',10);

%% 1 Формирование дискретного сигнала

fs = 128e3; % частота дискретизации

T = 12800/fs; % длительность символа

t = 1/fs:1/fs:T; % вектор временных значений

% формирование гармонического сигнала

s1 = cos(2\*pi\*5e3\*t);

% формирование ЛЧМ сигнала

s2 = chirp(t, 0, T, 5e3);

%% 2 Построение временной формы сигнала

figure('Position',[100 100 320 480])

subplot(2,1,1)

plot(t,s1)

grid on

xlim([0 0.001])

ylim([-1.5 1.5])

title('Гармонический сигнал')

xlabel('Время, с')

ylabel('Амплитуда, В')

subplot(2,1,2)

plot(t,s2)

grid on

xlim([0 0.05])

ylim([-1.5 1.5])

title('ЛЧМ сигнал')

xlabel('Время, с')

ylabel('Амплитуда, В')

%% 3 Расчет и построение спектра

k = 4;

fscale = -fs/2:fs/(length(t)\*k):fs/2-fs/(length(t)\*k); % частотная шкала

% гармонический сигнал

spec1 = fftshift(abs(fft(s1,length(t)\*k)));

spec1 = spec1\*2/length(t);

% ЛЧМ сигнал

spec2 = fftshift(abs(fft(s2,length(t)\*k)));

spec2 = spec2\*2/length(t);

% построение спектра гармонического сигнала

figure('Position',[420 100 320 480])

subplot(2,1,1)

plot(fscale,20\*log10(spec1))

grid on

xlim([-fs/2 fs/2])

ylim([-80 10])

title('Спектр гармонического сигнала')

xlabel('Частота, Гц')

ylabel('Амплитуда, дБВ')

% построение спектра ЛЧМ сигнала

subplot(2,1,2)

plot(fscale,20\*log10(spec2))

grid on

ylim([-80 -20])

xlim([-fs/2 fs/2])

title('Спектр ЛЧМ сигнала')

xlabel('Частота, Гц')

ylabel('Амплитуда, дБВ')

%% 4 Квантование сигнала

B = 2;

table = (1:2^B)/2^(B-1)-1;

s1q = quantiz(s1, table)/2^(B-1)-1+1/2^B;

s2q = quantiz(s2, table)/2^(B-1)-1+1/2^B;

%% 5 Построение временной формы квантованного сигнала

figure('Position',[740 100 320 480])

subplot(2,1,1)

plot(t,s1q)

grid on

xlim([0 0.001])

##ylim([-1.5 1.5])

title('Гармонический сигнал')

xlabel('Время, с')

ylabel('Амплитуда, В')

subplot(2,1,2)

plot(t,s2q)

grid on

xlim([0 0.05])

ylim([-1.5 1.5])

title('ЛЧМ сигнал')

xlabel('Время, с')

ylabel('Амплитуда, В')

%% 6 Расчет и построение спектра

k = 4;

fscale = -fs/2:fs/(length(t)\*k):fs/2-fs/(length(t)\*k); % частотная шкала

% гармонический сигнал

spec1 = fftshift(abs(fft(s1q,length(t)\*k)));

spec1 = spec1\*2/length(t);

% ЛЧМ сигнал

spec2 = fftshift(abs(fft(s2q,length(t)\*k)));

spec2 = spec2\*2/(length(t));

% построение спектра гармонического сигнала

figure('Position',[1060 100 320 480])

subplot(2,1,1)

plot(fscale,20\*log10(spec1))

grid on

xlim([-fs/2 fs/2])

ylim([-80 10])

title('Спектр гармонического сигнала')

xlabel('Частота, Гц')

ylabel('Амплитуда, дБВ')

% построение спектра ЛЧМ сигнала

subplot(2,1,2)

plot(fscale,20\*log10(spec2))

grid on

ylim([-80 -20])

xlim([-fs/2 fs/2])

title('Спектр ЛЧМ сигнала')

xlabel('Частота, Гц')