

Министерство образования Республики Беларусь

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Факультет компьютерных сетей и систем
Кафедра электронных вычислительных средств

Отчет по лабораторной работе № 1
**«Основы цифрового представления звуковых сигналов. Изменение
частоты дискретизации сигнала: децимация и интерполяция»**

Выполнили студенты гр. 150701
Кривальцевич Е.А.
XXX X.X.
XXX X.X.

Проверил
XXX X.X.

Минск 2025

1 ЦЕЛЬ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ

1 Изучить влияние параметров дискретизации и разрядности квантования сигналов в системах мультимедиа.

2 Разработать MATLAB-модель последовательного АЦП в соответствии с вариантом задания.

3 Изучить способы изменения частоты дискретизации (децимация и интерполяция) аудиосигналов.

4 Разработать MATLAB-модель, изменяющую частоту дискретизации сигнала в соответствии с вариантом задания.

2 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ РАБОТЫ

2.1 Задание 1

Необходимо разработать MATLAB-функцию последовательного АЦП в соответствии с алгоритмом его работы.

```
function [s_q] = serial_adc(x, N)
    s_q = zeros(size(x));

    for n = 1:length(x)
        data = x(n); % Initialization data
        BPR = 0; % Initialization BPR (bit-parallel representation)

        % Calculate BRP N times
        for i = 1:N
            BPR = BPR + 2^(-i); % Add 2^(-i) to BRP

            % Check carry state less then analog signal
            if abs(data) < BPR
                BPR = BPR - 2^(-i); % Sub 2^(-i) to BRP
            end
        end

        % Correct sign
        s_q(n) = sign(data) * BPR;
    end
end
```

2.2 Задание 2

При помощи разработанной функции serial_adc необходимо выполнить квантование гармонического сигнала в соответствии с вариантом. При помощи функций stem и stairs построить графики квантованного сигнала $s_q(n)$ для указанных в варианте значений N и наложить их на график функции $s(n)$.

$$s(n) = \cos\left(\frac{2\pi 20}{12000}n\right) + \sin\left(\frac{2\pi 450}{12000}n - \frac{\pi}{8}\right)$$

```

clearvars
clc
addpath('functions')

%% Create signal and Quantized them
N1 = 3;
N2 = 12;
n = 1:200;

s_n = cos(2 * pi * 20 .* n / 12000) + sin(2 * pi * 450 .* n / 12000 - pi / 8);

s_n = s_n / max(abs(s_n));
Quantized_s_n1 = serial_adc(s_n, N1);
Quantized_s_n2 = serial_adc(s_n, N2);

```

Результат выполнения квантования представлен на рисунке 1.

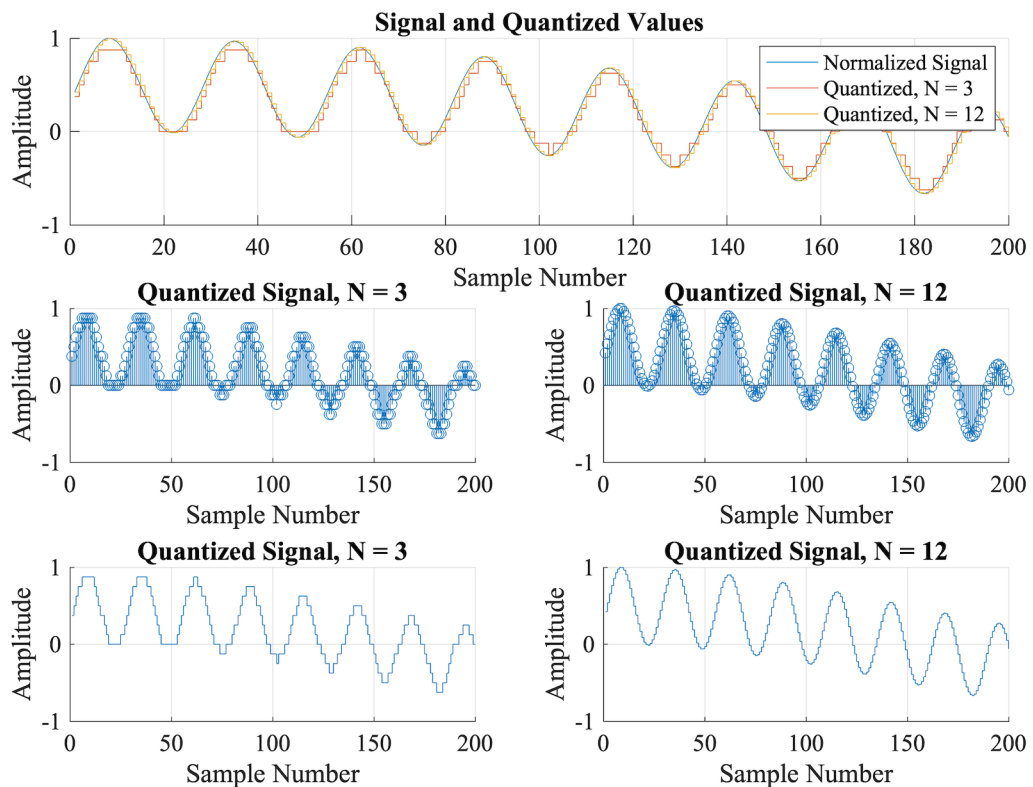


Рисунок 1 – Сигнал при $N = 3$ и $N = 12$

2.3 Задание 3

Необходимо построить графики ошибки квантования для квантованного сигнала $s_q(n)$ для указанных в варианте разрядностей N .

```

clearvars
clc
addpath('functions')

```

```

N1 = 3;
N2 = 12;

```

```

n = 1:200;

s_n = cos(2 * pi * 200 .* n / 12000) + sin(2 * pi * 90 .* n / 12000 - pi / 8);

s_n = s_n / max(abs(s_n));
Quantized_s_n1 = serial_adc(s_n, N1);
Quantized_s_n2 = serial_adc(s_n, N2);

%% Calculate mistake
e_n1 = s_n - Quantized_s_n1;
e_n2 = s_n - Quantized_s_n2;

```

Результат вычисления ошибки представлен на рисунке 2.

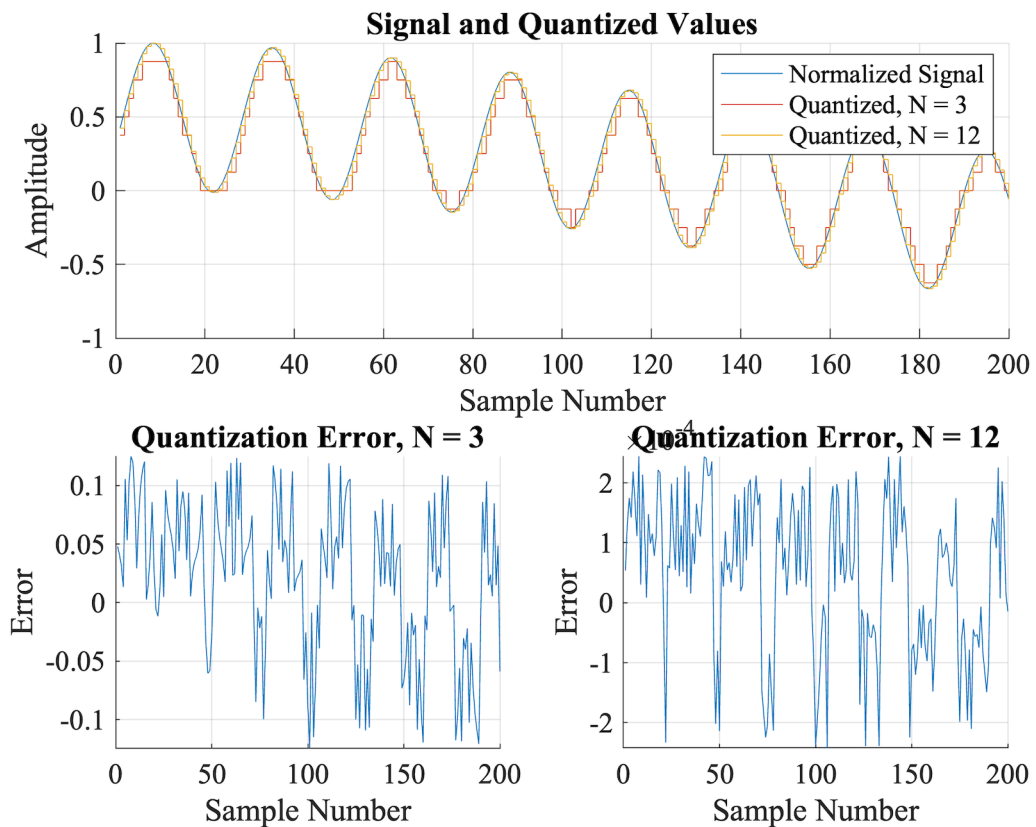


Рисунок 2 – Ошибка квантования для $N = 3$ и $N = 12$

2.4 Задание 4

Расчёт антиалайзингового фильтра для децимации и интерполяции. Порядок фильтра $N = 100$.

```

clearvars
clc

%% Filter options
N = 100; % filter order
Fc_interpolation = 0.1; % fcut
Fc_decimation = 0.1; % fcut

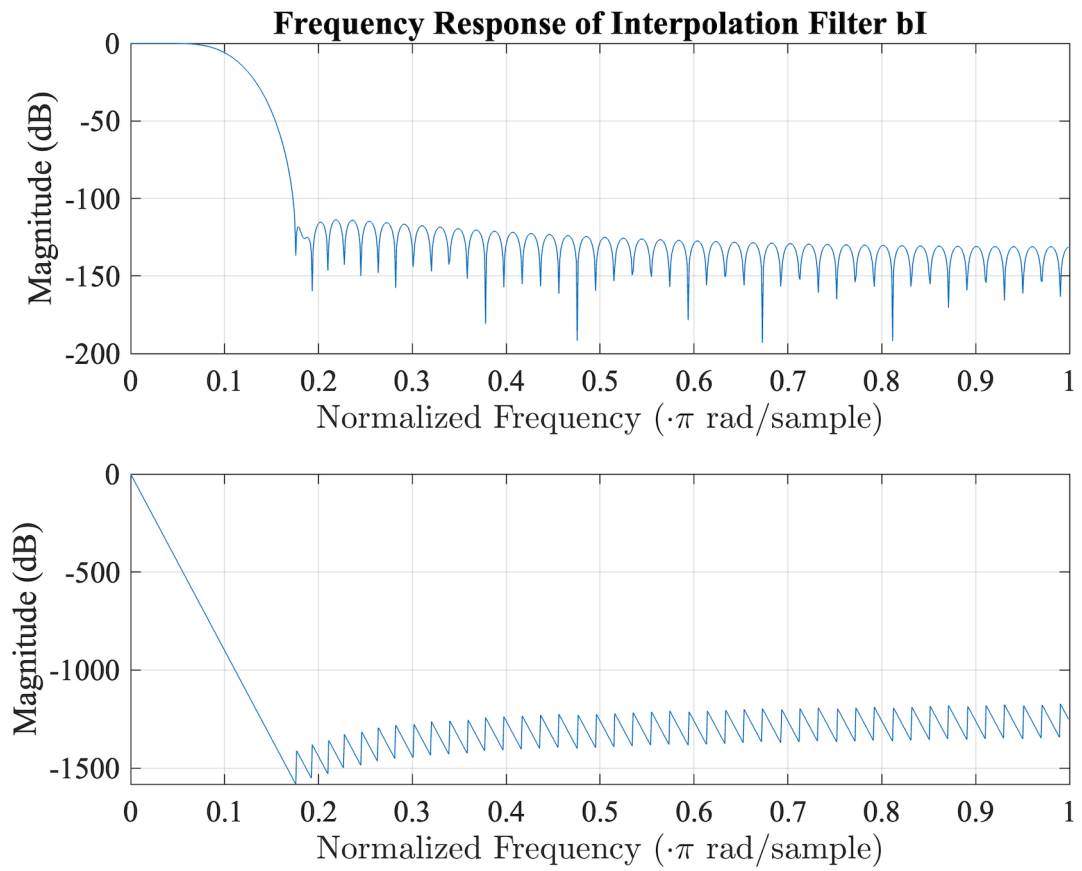
```

```
% Calculate filters
```

```
bI = fir1(N, Fc_interpolation, 'low', chebwin(N+1)); % Interpolation filter
```

```
bD = fir1(N, Fc_decimation, 'low', chebwin(N+1)); % Decimation filter
```

Результат построения фильтра представлен на рисунке 3.



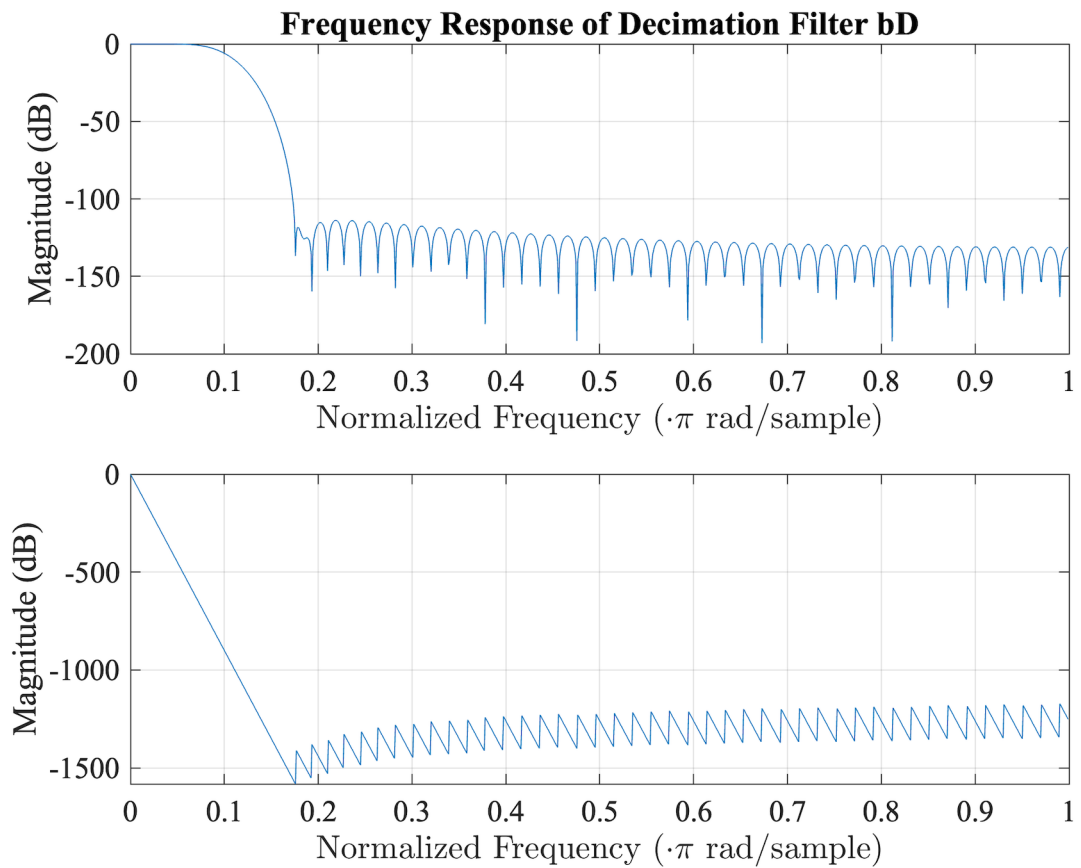


Рисунок 3 – Фильтр для $N = 3$ и $N = 12$

2.5 Задание 5

Для преобразования частоты дискретизации аудиосигнала с нецелым шагом необходимо разработать MATLAB-функцию расчёта параметра L для интерполяции и параметра M для децимации сигнала.

```
function [L, M] = find_resample_step(fs_old, fs_new)
    L = 1;
    M = 1;
    i = 1;

    while true
        l_tmp = fs_new * i / fs_old;

        if mod(l_tmp, 1) == 0
            L = l_tmp;
            M = i;
            break;
        end
        i = i + 1;
    end
end
```

2.6 Задание 6

Необходимо разработать MATLAB-функцию преобразования частоты дискретизации с нецелым шагом в соответствии с алгоритмом.

```
function [y] = resample_audio(x, fs_old, fs_new, bI, bD)
    [L, M] = find_resample_step(fs_old, fs_new);

    xI = upsample(x, L);
    yI = filter(bI, 1, xI);

    yD = filter(bD, 1, yI);
    y = downsample(yD, M);
end
```

2.7 Задание 7

При помощи разработанной функции `resample_audio` необходимо выполнить преобразование частоты дискретизации аудиосигнала в соответствии с вариантом. Аудиофайл «FA06_01.wav». Исходная частота дискретизации $f_s = 48000$ Гц. Требуемая частота дискретизации $f_s = 32000$ Гц.

```
clearvars
clc
addpath('functions')

%% Make dir
output_dir = 'wav/output';
input_dir = 'wav';

%% Check audio file
[x, fs_old] = audioread(fullfile(input_dir, 'FA06_01.wav'));
fs_new = 32000;
N = 100;

% Calculate L and aM
[L, M] = find_resample_step(fs_old, fs_new);
int_coef = 1 / L;
dec_coef = 1 / M;
if int_coef >= 1
    int_coef = int_coef - 0.0000000000000001;
end
if dec_coef >= 1
    dec_coef = dec_coef - 0.0000000000000001;
end

% Interpolation filter
bI = fir1(N-1, int_coef, 'low', chebwin(N, 60));

% Decimation filter
bD = fir1(N-1, dec_coef, 'low', chebwin(N, 60));
```

```
% resample
y = resample_audio(x, fs_old, fs_new, bI, bD);

% save result
audiowrite(fullfile(output_dir, 'output_audio_t7.wav'), y, fs_new);
```

2.8 Задание 8

Построить временное представление исходного сигнала и сигнала с изменённой частотой дискретизации.

```
clearvars
clc
addpath('functions')

figure;
% Plot spectrogram
subplot(2,2,1);
specgram(x, 512, fs_old, hann(512), 475);
title('Spectrogram of Original Signal');
set(gca, 'Clim', [-65 -15]);
xlabel('Time (s)');
ylabel('Frequency (Hz)');
set(gca, 'FontName', 'Times New Roman');
set(gca, 'FontSize', 14);

subplot(2,2,2);
specgram(y, 512, fs_new, hann(512), 475);
title('Spectrogram of Resampled Signal');
set(gca, 'Clim', [-65 -15]);
xlabel('Time (s)');
ylabel('Frequency (Hz)');
set(gca, 'FontName', 'Times New Roman');
set(gca, 'FontSize', 14);

% Plot time-domain representation
subplot(2,2,3);
plot((1:length(x))/fs_old, x);
xlim([0, length(x)/fs_old]);
title('Original Signal', 'FontName', 'Times New Roman', 'FontSize', 14);
xlabel('Time (s)');
ylabel('Amplitude');
set(gca, 'FontName', 'Times New Roman');
set(gca, 'FontSize', 14);

subplot(2,2,4);
plot((1:length(y))/fs_new, y);
xlim([0, length(y)/fs_new]);
title('Resampled Signal');
xlabel('Time (s)');
```



```
ylabel('Amplitude');
set(gca, 'FontName', 'Times New Roman');
set(gca, 'FontSize', 14);
```

Результат построения фильтра представлен на рисунке 3.

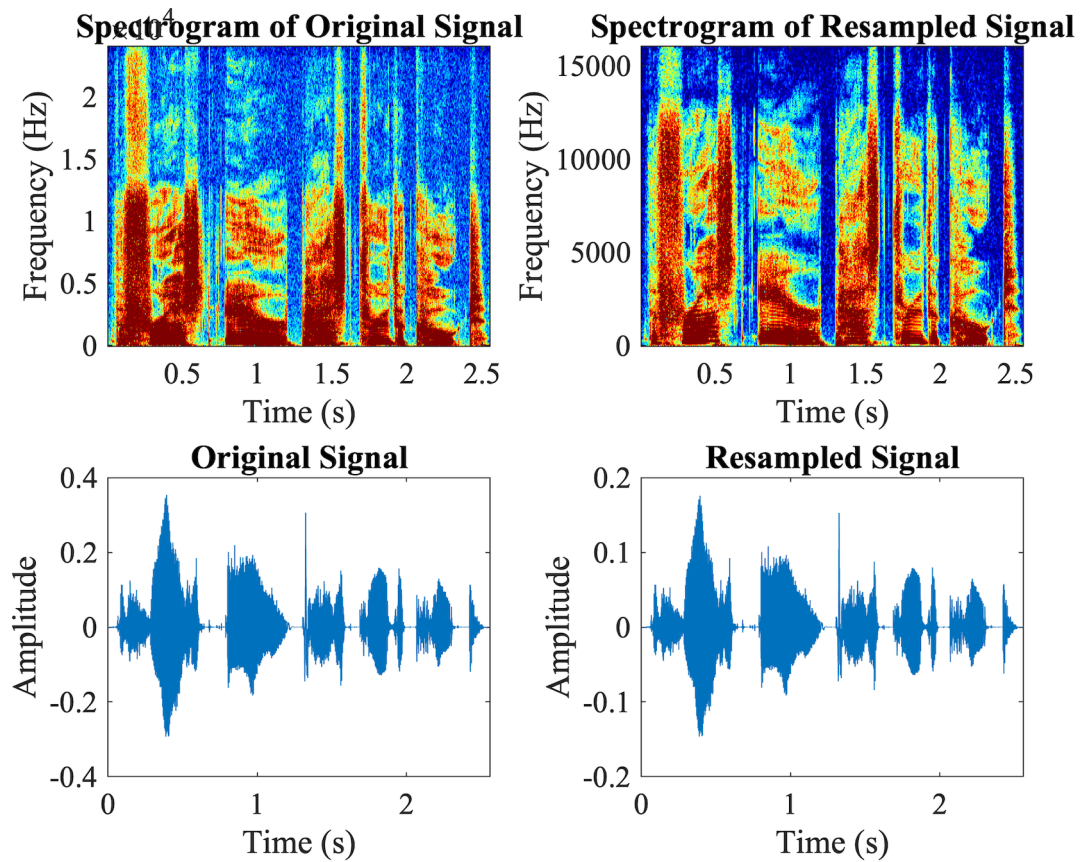


Рисунок 4 – Результат смены частоты дискретизации

3 ВЫВОД

В результате выполнения лабораторной работы были получены практические навыки работы с аудио сигналами. Рассмотрен принцип работы АЦП.