Отчёт «Дискретизация сигнала и его спектральный анализ»

Радько В. А., 2 группа 4 курс

Вариант 16

Выделение гармонических частот сигнала

Дан дискретный сигнал $x(t)$, применим к нему быстрое дискретное преобразование Фурье: **`fft`**. Поскольку полученный вектор будет лежать в $\mathbb{C}^n$, найдём модуль каждой координаты, тогда получим X и построим его график.

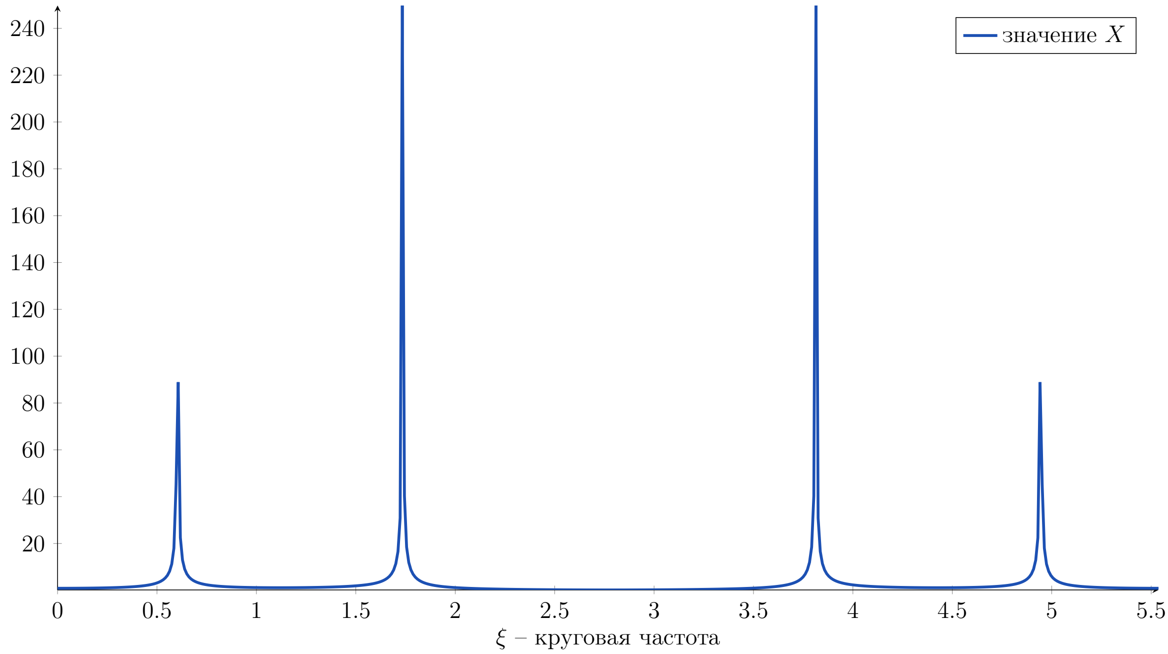

Рис. 1: График частот сигнала

На графике чётко видно два пика, на глаз они около значений 0.6 и 1.75. Найдём точные аргументы этих пиков: $\omega_1 = 0.60681$, $\omega_2 = 1.7337$. Частота в герцах равна $\nu = \dfrac{\omega}{2\pi}$, тогда $\nu_1 = 0.096576$Гц., $\nu_2 = 0.27593$Гц..

Добавление шума

Пусть A^* – амплитуда шумового сигнала e. Сам шум – это вектор из n равномерно распределённых на $(-A^*, A^*)$ случайных величин. Для генерации шума я воспользовался функцией **`unifrnd`** из пакета Octave.

Наложим шум на наш исходный сигнал и найдём такую пороговую амплитуду A^*, что при дальнейшем увеличении амплитуды шума невозможно будет распознать первый, низкий пик. Графики с низкой и высокой амплитудой шума на рисунках 2 и 3.

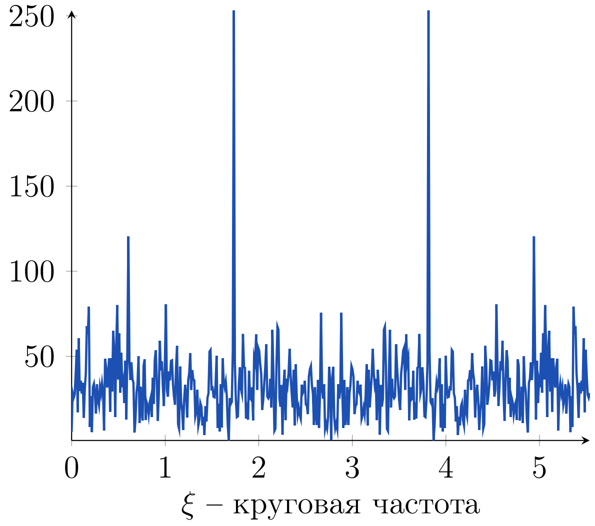

Рис. 2: $\mathcal{F}(x+e)$ при $A = 2.6$

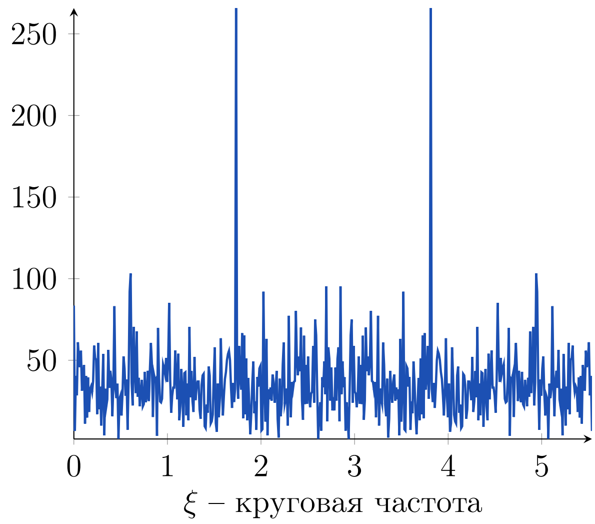

Рис. 3: $\mathcal{F}(x+e)$ при $A = 2.9$

Алиасинг

Добавим к исходному сигналу новый сигнал $y(t) = \hat{A}\cos\Omega_1 t + 2\hat{A}\cos\Omega_2 t$. Найдём близкие к максимуму значения $\Omega_{1,2}$, но чтобы условие Найквиста всё ещё выполнялось. Затем начнём увеличивать Ω_2 до тех пор, пока пики не поменяются местами. Изобразим распределения частот на графике, построенном при помощи преобразования Фурье до и после увеличения Ω_2 (рисунки 4, 5).

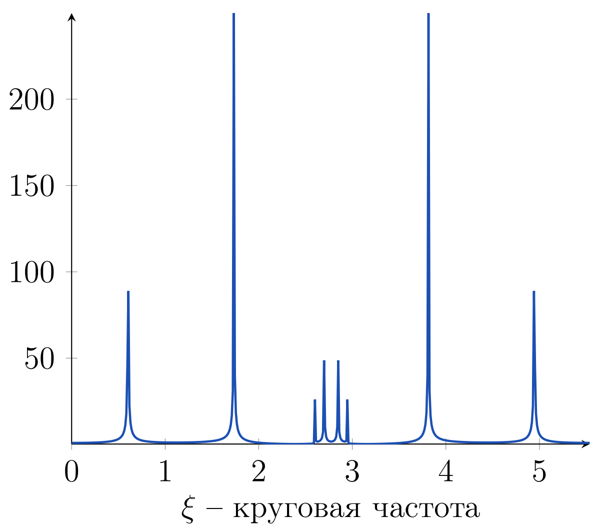

Рис. 4: Значения Y до увеличения Ω_2

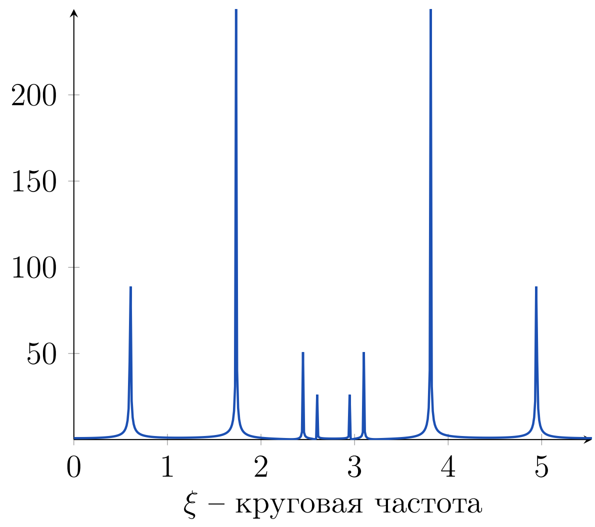

Рис. 5: Значения Y после увеличения Ω_2

На рисунке 4 значения частот $\Omega_1 = 2.6$, $\Omega_2 = 2.7$. На рисунке 5 значения частот $\Omega_1 = 2.6$, $\Omega_2 = 3.1$. Запишем условие Найквиста и посмотрим, есть ли алиасинг.

$$\xi_0 \cdot \Delta t < \pi$$

То есть $\xi_0 = \dfrac{\pi}{\Delta t}$ – это граничное значение частоты, выше которой возникает алиасинг. Для моего сигнала с шагом дискретизации $\Delta t = 1.1325$ получается значение $\xi_0 = 2.774$. Следовательно на рисунке 4 алиасинга не происходит, а на рисунке 5 - алиасинг есть.

Изменение шага дискретизации

Изменим шаг дискретизации, $\Delta t' = 2\Delta t$, тогда $n' = n/2$. Однако $\Delta\xi$ не изменится, $\Delta\xi' = \dfrac{2\pi}{n'\Delta t'} = \dfrac{2\pi}{2\Delta t\frac{n}{2}} = \dfrac{2\pi}{n\Delta t} = \Delta\xi$. Так же найдём новое граничное значение частоты из условия Найквиста $\xi_0' = \dfrac{\pi}{\Delta t'} = \dfrac{\pi}{2\Delta t} = \dfrac{\xi_0}{2}$, в моём варианте $\xi_0' = 1.387$.

Сравнивая ξ_0, полученный в прошлом задании с ξ_0', можно сделать вывод, что все частоты, находящиеся в промежутке $(\xi_0', \xi_0) = (1.387, 2.774)$ при уменьшении частоты дискретизации, будут накладываться на другие частоты и приводить к алиасингу.

В исходном сигнале мы имеем частоты $\omega_1 = 0.60681$, $\omega_2 = 1.7337$, видно, что $\omega_2 \in (1.387, 2.774) \Rightarrow$ алиасинг возникает.

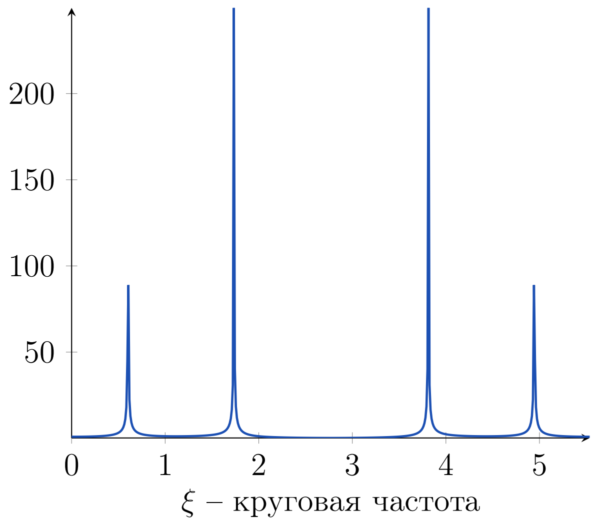

Рис. 6: Частоты до уменьшения шага дискретизации

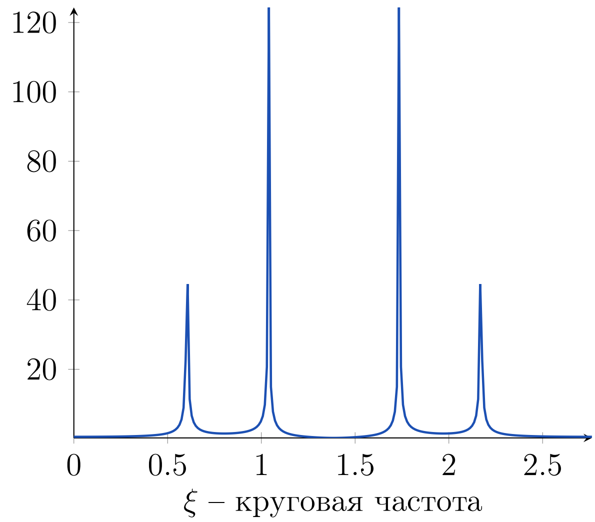

Рис. 7: Частоты после уменьшения шага дискретизации

Листинг 1: read_signal.m

```
function [signal, dt] = read_signal(filename)
    v = load(filename);
    [n, m] = size(v);
    dt = v(1);
    signal = v([2:n], :);
end
```

Листинг 2: task1.m

```
clear
[x, dt] = read_signal('data/Sa16.tx');
[n, m] = size(x);

t = [0:dt:(n-1)*dt]';

dxi = 2*pi/n/dt;
xi = [0:dxi:(n-1)*dxi]';

xi0 = pi/dt

X = abs(fft(x));

% search local maximum in first quarter (for small peak)
[max_value, max_index] = max(X([1:n/4], :));
omega1 = (max_index-1)*dxi

% search maximum in first half (for large peak)
[max_value, max_index] = max(X([1:n/2], :));
omega2 = (max_index-1)*dxi



% plot of the signal x
% plot(t, x);

% plot of the Fourier transformation of x
plot(xi, X);

% save graph in file for latex usage
% SaveX = [xi, X];
% save('data/X.graph', 'SaveX');
```

Листинг 3: `task2.m`

```
clear
[x, dt] = read_signal('data/Sa16.tx');
[n, m] = size(x);

dxi = 2*pi/n/dt;
xi = [0:dxi:(n-1)*dxi]';

% generate noise of amplitude A
A = 2.6
noise = unifrnd(-A, A, n, m);

y = x + noise;
Y = abs(fft(y));

% plot of the Fourier transformation of y
plot(xi, Y);

% save graph in file for latex usage
% SaveX = [xi, Y];
% save('data/XA28.graph', 'SaveX');
```

Листинг 4: `task3.m`

```
clear
[x, dt] = read_signal('data/Sa16.tx');
[n, m] = size(x);

t = [0:dt:(n-1)*dt]';

dxi = 2*pi/n/dt;
xi = [0:dxi:(n-1)*dxi]';

% define signal as function
function y=S(t, a, w1, w2)
    y = a*cos(w1*t) + 2*a*cos(w2*t);
end

% descrete signal
A = 0.1
omega1 = 2.6;
omega2 = 3.1;
```

```

y = S(t, A, omega1, omega2);
y += x;
Y = abs(fft(y));

% this plots signal y = x + (a*cos + 2*a*cos)
% plot(t, y);

% Fourier transformation of y
plot(xi, Y);

% SaveX = [xi, Y];
% save('data/Y1.graph', 'SaveX');
```

Листинг 5: task4.m

```
clear
[x, dt] = read_signal('data/Sa16.tx');
[n, m] = size(x);

dxi = 2*pi/n/dt;

t   = [0:dt :(n-1)*dt]';
xi  = [0:dxi:(n-1)*dxi]';

X = abs(fft(x));

SaveX = [xi, X];
save('data/pic6.graph', 'SaveX');

% and now we increase descrete step

dtau = 2*dt;

y = [];
for i=1:2:n
    y = [y; x(i)];
end
[n2, m2] = size(y);

tau = [0:dtau:(n2-1)*dtau]';
eta = [0:dxi :(n2-1)*dxi]';
```

```

xi0 = pi/dtau

Y = abs(fft(y));

SaveX = [eta, Y];
save('data/pic7.graph', 'SaveX');
```