

ГУАП

КАФЕДРА № 44

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Ст. преподаватель		Е.К. Григорьев
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

МОДЕЛИРОВАНИЕ И ОБРАБОТКА СИГНАЛОВ В ПАКЕТЕ MATLAB

по курсу: МОДЕЛИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №		Е.Д.Тегай
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

Цель работы

Ознакомиться с методикой моделирования и обработки периодических дискретных сигналов с помощью пакета MATLAB.

Вариант задания

Содержание индивидуального варианта продемонстрировано на рисунке 1. Для удобства восприятия необходимые данные выделены жёлтым цветом.

№	11	12	13	14	15	16	17	18	19	20
A_1	200	300	400	500	600	500	400	300	200	300
ω_1	60	50	40	30	20	60	50	40	30	20
φ_1										
A_2	500	400	300	200	300	400	300	200	400	200
ω_2	40	60	50	60	50	40	30	50	20	30
φ_2										

Рисунок 1 – Индивидуальное задание

Ход работы

Перед началом работы необходимо импортировать файл variant19.xlsx в

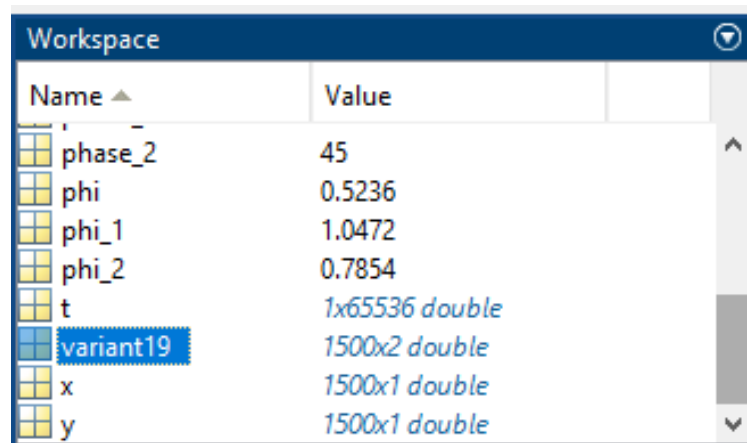


Рисунок 2 – Импорт данных

Затем по исходным данным необходимо построить график сигнала. При этом следует учесть, что частота дискретизации, с которой записан сигнал –

в секунду. После импорта при написании кода отсчёты были записаны в отдельные вектор столбцы с помощью команд:

где x – время, в которое был записан отсчет, y – значение сигнала. Итого получаем программу, код которой показан далее.

Код программы

```
% Создание отдельных векторов для времени (x) и значений сигнала (y)
x = variant19(:, 1); % Где 1 - номер столбца, содержащего время
y = variant19(:, 2); % А 2 - номер столбца, содержащего значения сигнала

% Построение графика отсчётов
plot(x, y);
xlabel('Время');
ylabel('Значение сигнала');
title('График отсчетов сигнала');
grid on;
```

Результат работы программы

График отсчётов продемонстрирован на рисунке 3.

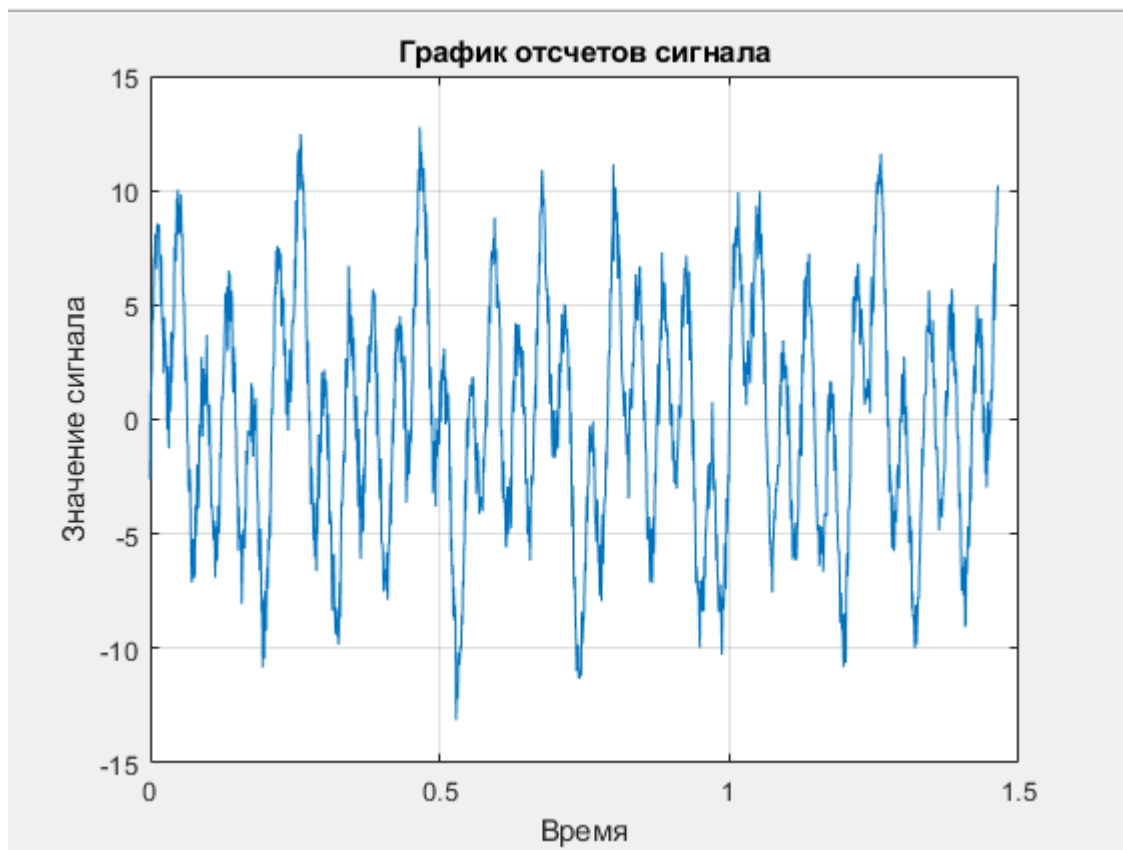


Рисунок 3 – График отсчётов

Ознакомимся более подробно с функцией `fft`. Данная функция является аббревиатурой, обозначающей быстрое преобразование Фурье. Она используется для выполнения дискретного преобразования Фурье (далее – ДПФ) или его обратной операции на заданном наборе данных. Синтаксис имеет вид:

$$Y$$

где X – входные данные (сигнал), а fY – результат преобразования Фурье сигнала X .

$$f$$

Функция `fft` выполняет прямое ДПФ входных данных. Для ряда значений сигнала во временной области X функция `fft` вычисляет его частотное представление в области частот Y . Это позволяет анализировать спектральные характеристики сигнала, такие как частоты компонентов сигнала и их амплитуды.

Теперь необходимо, используя БПФ, получить амплитудный и фазовый спектры сигнала, а затем построить графики. Код полученной программы и результаты её работы показаны далее.

Вообще, амплитудный и фазовый спектры являются способами представления частотных характеристик сигнала после преобразования Фурье. Рассмотрим более подробно амплитудный спектр.

Амплитудный спектр показывает амплитуду каждой составляющей частоты в сигнале. Он позволяет определить, какие частоты присутствуют в сигнале и с какой амплитудой. Затем рассмотрим более подробно фазовый спектр.

Фазовый спектр показывает фазу (угол) каждой составляющей частоты в сигнале. Он позволяет определить фазовые сдвиги между различными частотами в сигнале.

Листинг программы

```
% Создание отдельных векторов для времени (x) и значений сигнала (y)
x = variant19(:, 1); % Где 1 - номер столбца, содержащего время
y = variant19(:, 2); % А 2 - номер столбца, содержащего значения сигнала

% Построение графика отсчётов
plot(x, y);
```

```

xlabel('Время');
ylabel('Значение сигнала');
title('График отсчетов сигнала');
grid on;

N = length(x);
% Вычислим амплитудный спектр заново, чтобы можно было извлечь информацию
SPECTRUM = 2/N * fftshift(fft(y,N));

% Разрешение по частоте
fs = 1024;
df = fs/N;
sample_index = -N/2:N/2-1;
% Преобразуем индексы в частоту
f = sample_index * df;

% Построим график амплитудного спектра с учетом нововведений:
figure()
plot(f,abs(SPECTRUM),'LineWidth',2)
xlabel('Частота')
ylabel('|Y[k]|')
grid on
title('Амплитудный спектр сигнала')

% Создадим копию спектра
spectrum2 = SPECTRUM;

% Зададим порог, ниже которого все элементы будут обнуляться
treshold = max(abs(spectrum2))/10;
spectrum2(abs(SPECTRUM)<treshold) = 0;

phase_spectrum = atan2(imag(spectrum2),real(spectrum2))*180/pi;

figure()
plot(f,phase_spectrum,'LineWidth',2)
xlabel('Частота')
ylabel('phi(f)')
title('Фазовый спектр сигнала')
grid on

```

Результат работы программы

Графики полученных амплитудного и фазового спектров продемонстрированы на рисунках 4 – 5 и 6 – 7 соответственно.

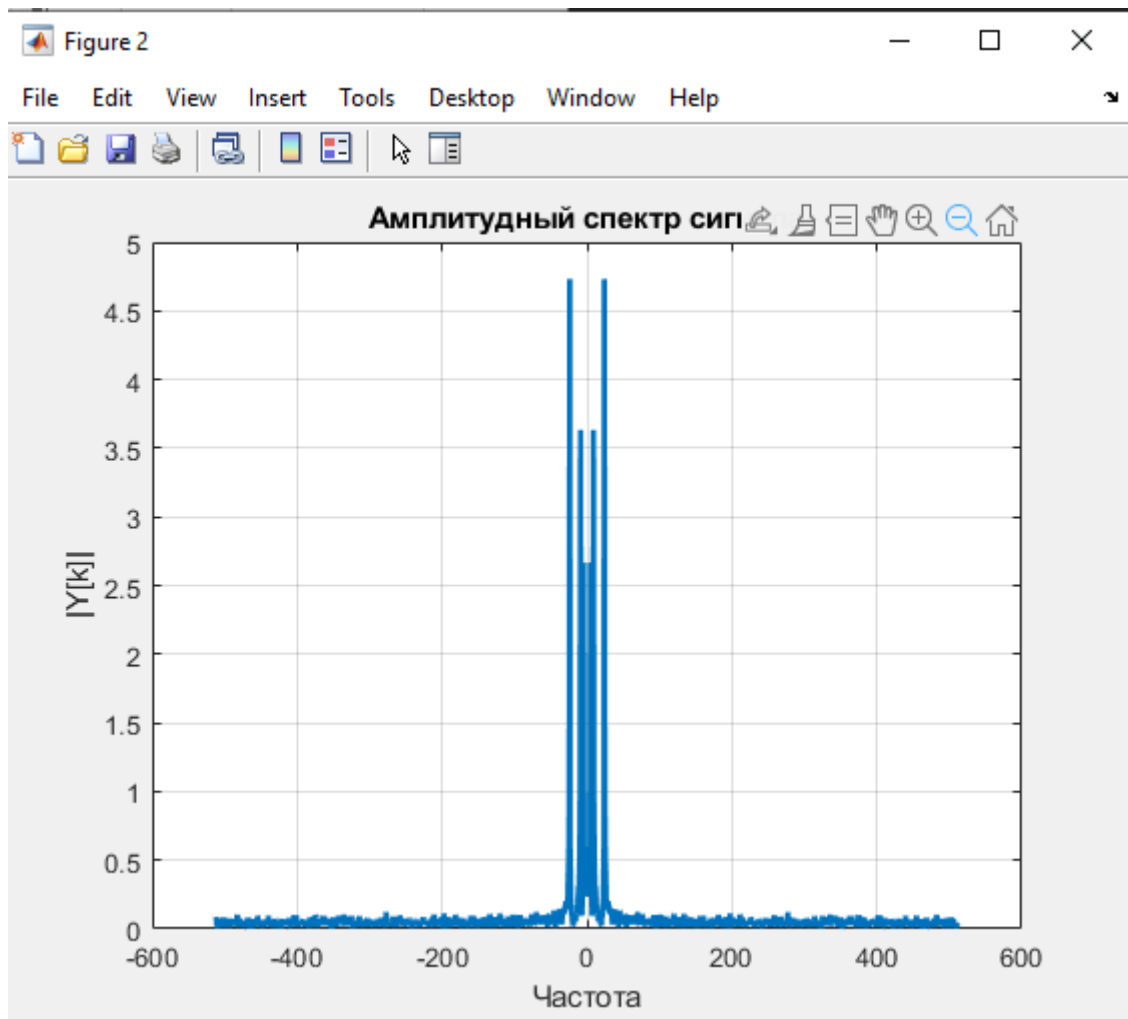


Рисунок 4 – Амплитудный спектр

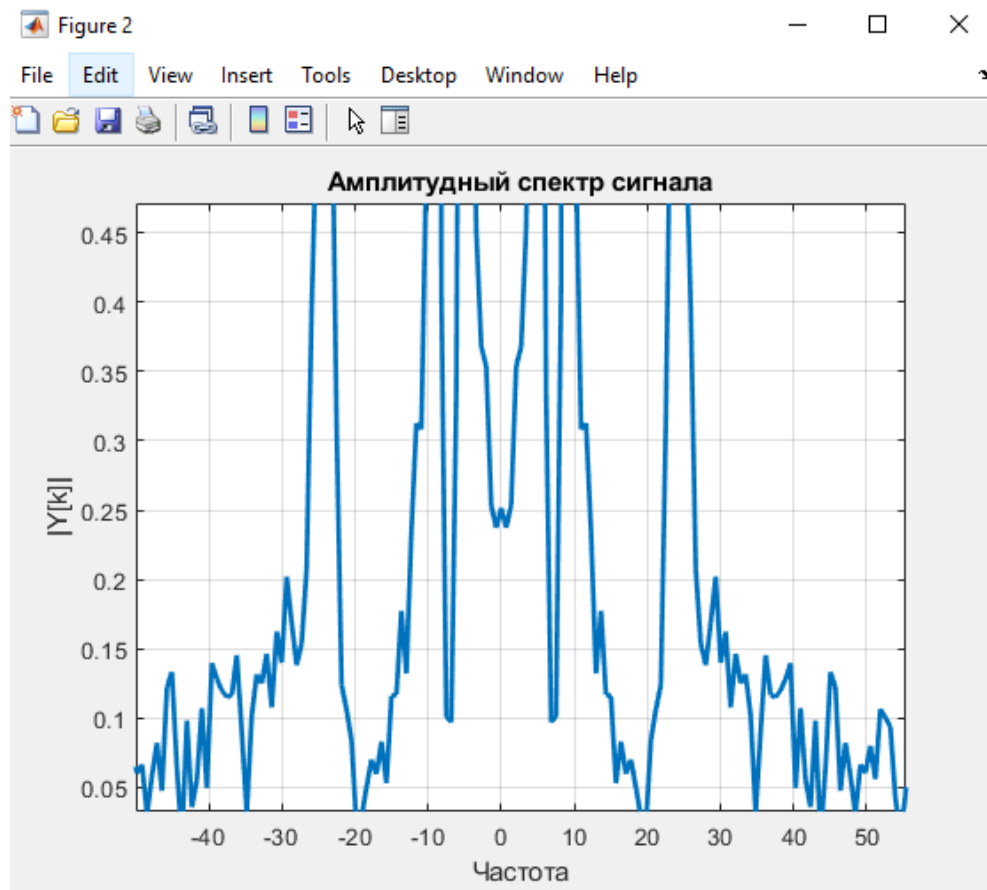


Рисунок 5 – Амплитудный спектр

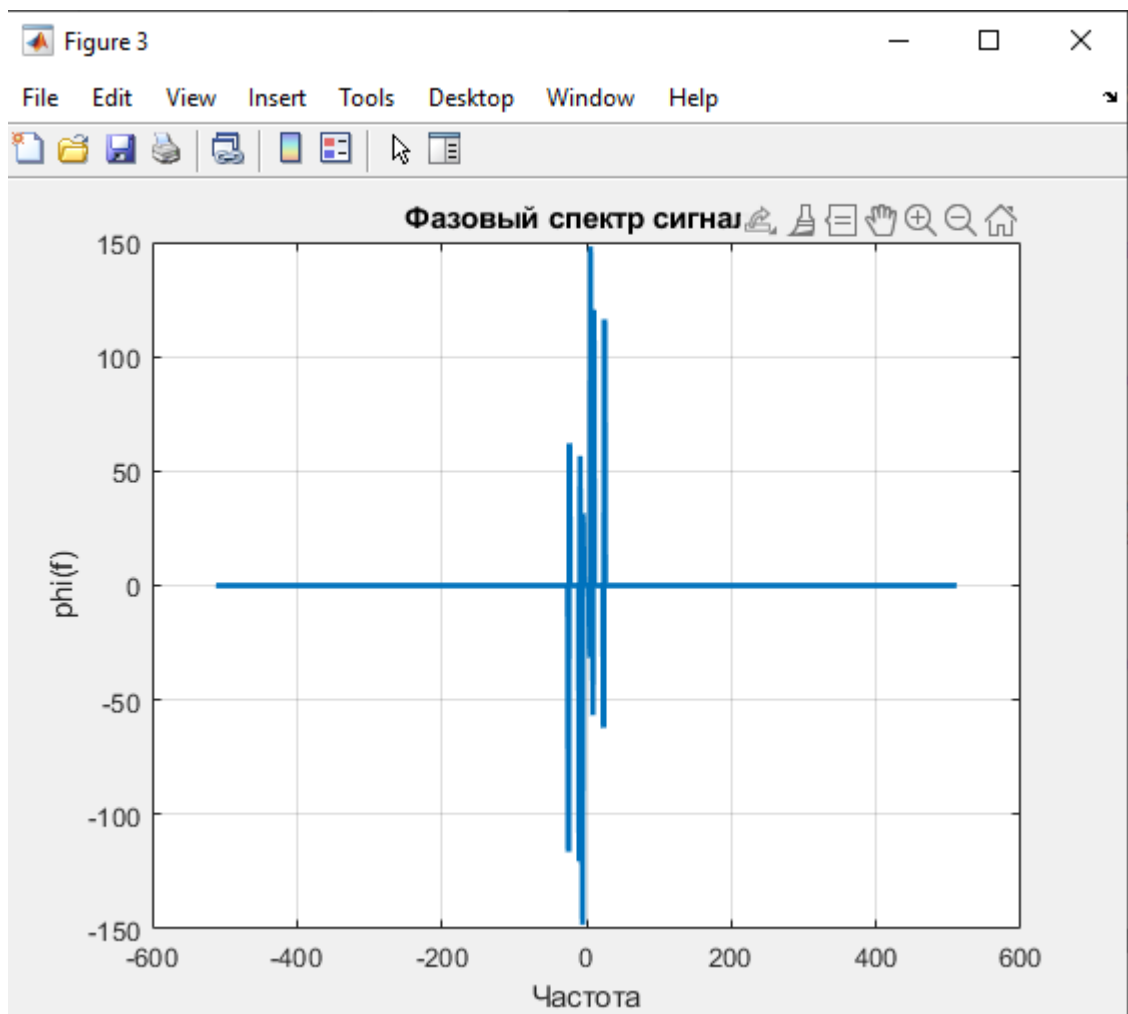


Рисунок 6 – Фазовый спектр

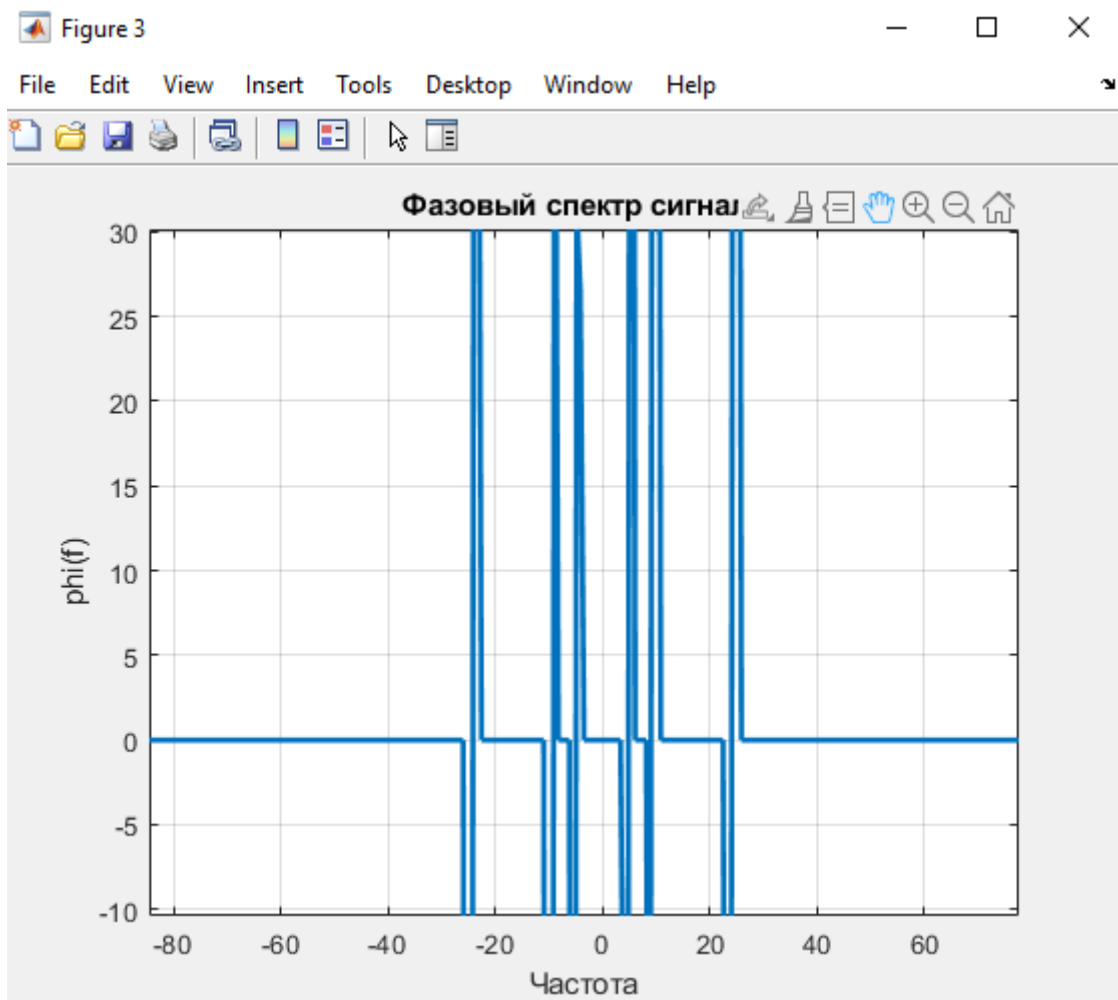


Рисунок 7 – Фазовый спектр

Проанализируем полученные результаты. Грубо говоря, на графике амплитудного спектра получилось несколько ярко выделяющихся высоких «горок». Значит, что сигнал содержит только несколько дискретных частотных компонентов или имеет ограниченный спектр.

Рассмотрим график фазового спектра. Он состоит из строго вертикальных и диагональных линий (это можно увидеть, если сильно увеличить график, хотя может показаться, что график состоит из единичных вертикальных полос). Вертикальные линии могут появиться, если сигнал содержит фазовые сдвиги между различными частотными компонентами. Диагональные же линии говорят о линейной зависимости фазы от частоты. Причём угол наклона каждой диагонали свой.

Далее необходимо примерно определить амплитуды и частоты гармонических сигналов, входящих в сигнал по варианту. Первое можно

определить по высоте пиков на полученном графике амплитудного спектра. Возьмём «пики» с амплитудами 3.63252, 4.7325 и 2.6664. Вообще, грубо говоря, это и есть те самые сильно выделяющиеся пики на фоне шума.

Частоту также можно определить по графику амплитудного спектра. Она является значением по оси абсцисс относительно пика «горки». Итого получаем: 10.24, 24.576 и 5.46133 соответственно. Результаты продемонстрированы в таблице 1.

Таблица 1

Амплитуда	Частота

Приступим к выполнению второй части лабораторной работы. Для этого создадим ещё один m-файл и зададим соответствующие значения двух гармонических сигналов по варианту согласно рисунку 1.

Затем необходимо сгенерировать сигнал, представляющий собой сумму двух синусоид и гауссовского шума с нулевым математическим ожиданием и СКО, равным единице. Код программы показан ниже.

Код программы

```
clear all
close all
clc

% Параметры синусоид
A1 = 200;
omega1 = 30;
phi1 = 0;

A2 = 400;
omega2 = 20;
phi2 = 0;

% Частота дискретизации и продолжительность сигнала
Fs = 2^10;
T = 1;

t = 0:1/Fs:T-1/Fs; % Вектор времени

% Генерация гауссовского шума
```

```

noise = randn(size(t));

% Сигнал - сумма синусоид и гауссовского шума
signal = A1*sin(2*pi*omega1*t + phi1) + A2*sin(2*pi*omega2*t + phi2) + noise;

% Построение графика сигнала
plot(t, signal);
xlabel('Время');
ylabel('Значение сигнала');
title('Сигнал: Сумма двух синусоид и гауссовского шума');
grid on;

```

Результат работы программы

Полученный график показан на рисунке 8.

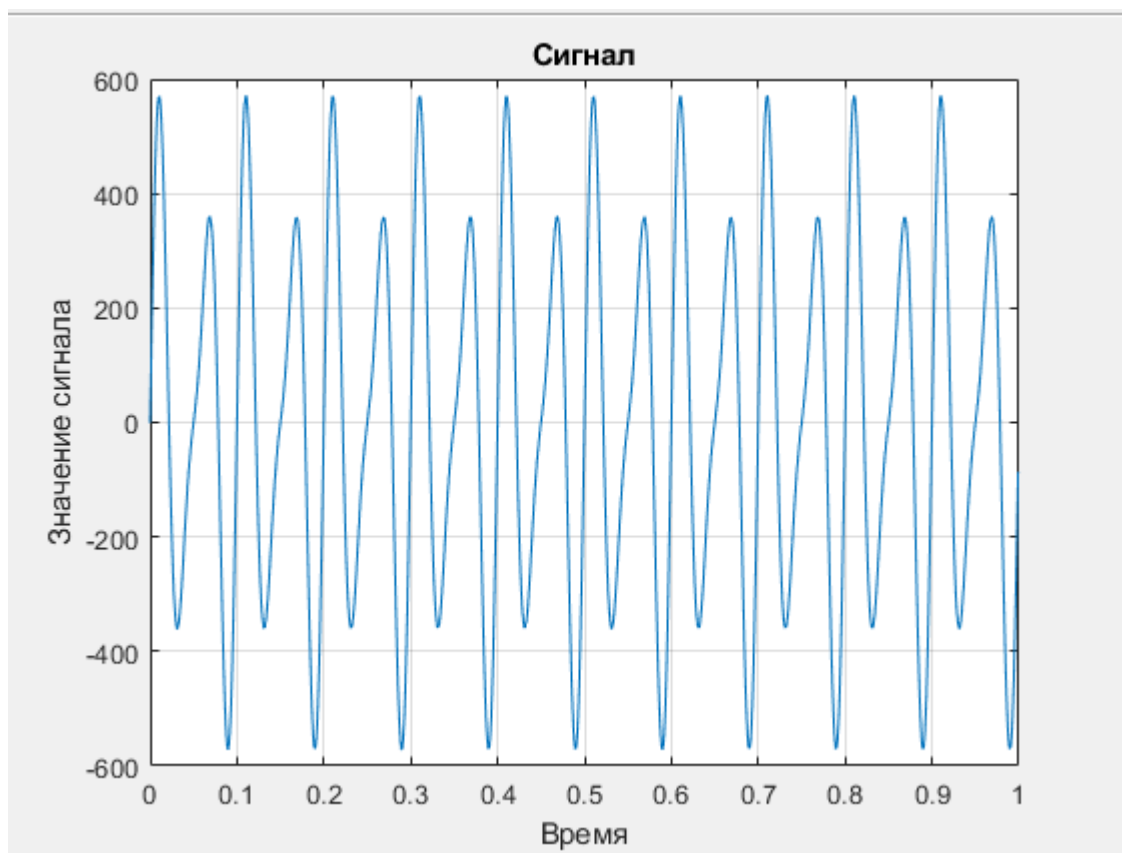


Рисунок 8 – График сигнала

Далее необходимо, используя, БПФ, получить амплитудный спектр смеси гармонических сигналов с шумом и оценить значения амплитуды и частоты по амплитудному спектру. Код продемонстрирован ниже. Следует отметить, что в коде используется модуль для отображения одностороннего амплитудного спектра. Такая форма записи предпочтительна, потому что реальные сигналы симметричны относительно нулевой частоты, и это

позволяет сократить размер данных в 2 раза и избежать дублирования информации.

Код программы

```
clear all
close all
clc

% Параметры синусоид
A1 = 200;
omega1 = 30;
phi1 = 0;

A2 = 400;
omega2 = 20;
phi2 = 0;

% Частота дискретизации и продолжительность сигнала
Fs = 2^10;
T = 1;

t = 0:1/Fs:T-1/Fs; % Вектор времени

% Генерация гауссовского шума
noise = randn(size(t));

% Сигнал - сумма синусоид и гауссовского шума
signal = A1*sin(2*pi*omega1*t + phi1) + A2*sin(2*pi*omega2*t + phi2) + noise;

% Построение графика сигнала
plot(t, signal);
xlabel('Время');
ylabel('Значение сигнала');
title('Сигнал');
grid on;

% БПФ (Быстрое Преобразование Фурье) смеси сигналов
N = length(signal);
spectrum = fft(signal, N);
spectrum = abs(spectrum(1:N/2+1)); % Получение одностороннего амплитудного спектра

% Разрешение по частоте
df = Fs/N;
frequencies = (0:(N/2)) * df;

% Построение графика амплитудного спектра
figure;
plot(frequencies, spectrum, 'LineWidth', 2);
xlabel('Частота (Гц)');
ylabel('Амплитуда');
title('Амплитудный спектр смеси гармонических сигналов с шумом');
grid on;
```

Результат работы программы

Полученный спектр продемонстрирован на рисунке 9.

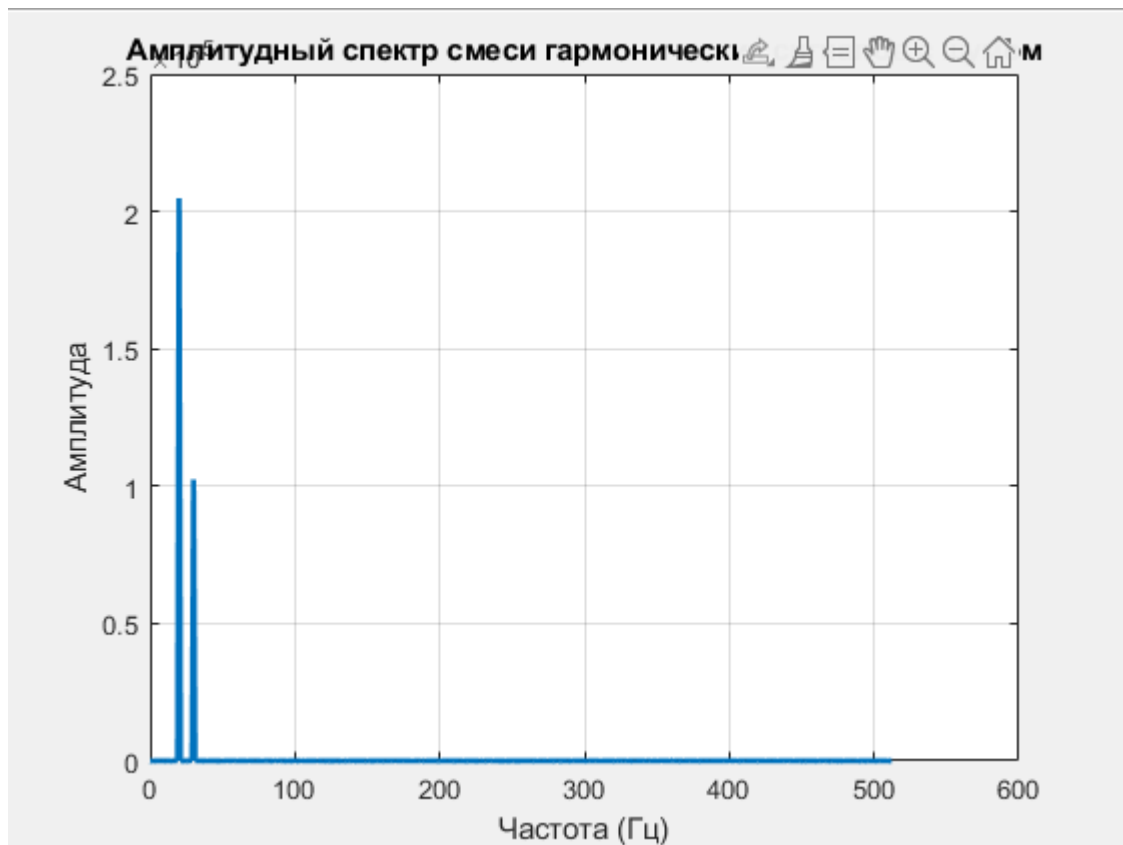


Рисунок 9 – Амплитудный спектр

Оценим значения амплитуды и частоты по амплитудному спектру.

Итого получаем:

Амплитуда – 204787, 102371

Частота – 20, 30

Увеличим СКО шума до 6 с шагом 0.5 и снова построим амплитудный спектр. Код продемонстрирован ниже.

Код программы

```
clear all
close all
clc

% Параметры синусоид
A1 = 200;
omega1 = 30;
phi1 = 0;

A2 = 400;
omega2 = 20;
phi2 = 0;

% Частота дискретизации и продолжительность сигнала
Fs = 2^10;
T = 1;

t = 0:1/Fs:T-1/Fs; % Вектор времени
```

```

% Увеличение СКО шума с шагом 0.5
for std_dev = 1:0.5:6
    % Генерация гауссовского шума
    noise = std_dev * randn(size(t));

    % Сигнал - сумма синусоид и гауссовского шума
    signal = A1*sin(2*pi*omega1*t + phi1) + A2*sin(2*pi*omega2*t + phi2) + noise;

    % БПФ (Быстрое Преобразование Фурье) смеси сигналов
    N = length(signal);
    spectrum = fft(signal, N);
    spectrum = abs(spectrum(1:N/2+1)); % Получение одностороннего амплитудного
    спектра

    % Разрешение по частоте
    df = Fs/N;
    frequencies = (0:(N/2)) * df;

    % Оценка значений амплитуды и частоты
    [~, idx1] = max(spectrum);
    a1 = spectrum(idx1);
    f1 = frequencies(idx1);

    [~, idx2] = max(spectrum(spectrum~=a1));
    a2 = spectrum(idx2);
    f2 = frequencies(idx2);

    % Вывод результатов
    fprintf('Стандартное отклонение шума: %.1f\n', std_dev);
    fprintf('Амплитуда первой компоненты: %.2f\n', a1);
    fprintf('Частота первой компоненты: %.2f\n', f1);
    fprintf('Амплитуда второй компоненты: %.2f\n', a2);
    fprintf('Частота второй компоненты: %.2f\n\n', f2);

    % Построение графика амплитудного спектра
    figure;
    plot(frequencies, spectrum, 'LineWidth', 2);
    xlabel('Частота (Гц)');
    ylabel('Амплитуда');
    title(sprintf('Амплитудный спектр смеси гармонических сигналов с шумом
    (СКО=%.1f)', std_dev));
    grid on;
end

```

Результат работы программы

Стандартное отклонение шума: 1.0

Амплитуда первой компоненты: 204822.39

Частота первой компоненты: 20.00

Амплитуда второй компоненты: 35.18

Частота второй компоненты: 29.00

Стандартное отклонение шума: 1.5

Амплитуда первой компоненты: 204815.31

Частота первой компоненты: 20.00

Амплитуда второй компоненты: 38.42

Частота второй компоненты: 29.00

Стандартное отклонение шума: 2.0

Амплитуда первой компоненты: 204808.08

Частота первой компоненты: 20.00

Амплитуда второй компоненты: 62.60

Частота второй компоненты: 29.00

Стандартное отклонение шума: 2.5

Амплитуда первой компоненты: 204761.68

Частота первой компоненты: 20.00

Амплитуда второй компоненты: 75.66

Частота второй компоненты: 29.00

Стандартное отклонение шума: 3.0

Амплитуда первой компоненты: 204755.21

Частота первой компоненты: 20.00

Амплитуда второй компоненты: 107.37

Частота второй компоненты: 29.00

Стандартное отклонение шума: 3.5

Амплитуда первой компоненты: 204882.86

Частота первой компоненты: 20.00

Амплитуда второй компоненты: 42.11

Частота второй компоненты: 29.00

Стандартное отклонение шума: 4.0

Амплитуда первой компоненты: 204838.76

Частота первой компоненты: 20.00

Амплитуда второй компоненты: 60.62

Частота второй компоненты: 29.00

Стандартное отклонение шума: 4.5

Амплитуда первой компоненты: 204836.99

Частота первой компоненты: 20.00

Амплитуда второй компоненты: 20.51

Частота второй компоненты: 29.00

Стандартное отклонение шума: 5.0

Амплитуда первой компоненты: 204747.49

Частота первой компоненты: 20.00

Амплитуда второй компоненты: 91.12

Частота второй компоненты: 29.00

Стандартное отклонение шума: 5.5

Амплитуда первой компоненты: 204895.19

Частота первой компоненты: 20.00

Амплитуда второй компоненты: 58.81

Частота второй компоненты: 29.00

Стандартное отклонение шума: 6.0

Амплитуда первой компоненты: 204736.79

Частота первой компоненты: 20.00

Амплитуда второй компоненты: 229.50

Частота второй компоненты: 29.00

Из этого можно сделать вывод о том, что амплитуда и частота первой компоненты остаются относительно постоянными, в то время как амплитуда

второй компоненты увеличивается с увеличением СКО шума. Частота второй компоненты остаётся постоянной на протяжении всех выводов.

Теперь построим график зависимости ошибки определения амплитуды сигнала в зависимости от СКО шума. Код продемонстрирован ниже.

Код программы

```
clear all
close all
clc

% Параметры сигналов
A1 = 200;
omega1 = 30;
phi1 = 0;

A2 = 400;
omega2 = 20;
phi2 = 0;

% Частота дискретизации и продолжительность сигнала
Fs = 2^10;
T = 1;

t = 0:1/Fs:T-1/Fs; % Вектор времени

% Инициализация массива для хранения ошибок
errors = [];

% Увеличение СКО шума с шагом 0.5
for std_dev = 1:0.5:500
    % Генерация гауссовского шума
    noise = std_dev * randn(size(t));

    % Сигнал - сумма синусоид и гауссовского шума
    signal = A1*sin(2*pi*omega1*t + phi1) + A2*sin(2*pi*omega2*t + phi2) + noise;

    % БПФ (Быстрое Преобразование Фурье) смеси сигналов
    N = length(signal);
    spectrum = fft(signal, N);
    spectrum = abs(spectrum(1:N/2+1)); % Получение одностороннего амплитудного спектра

    % Оценка значений амплитуды и частоты
    [~, idx1] = max(spectrum);
    a1 = spectrum(idx1);
    f1 = idx1;

    % Оценка ошибки амплитуды первой компоненты
    error = abs(a1 - A1);

    % Добавление ошибки в массив
    errors = [errors, error];
end

errors_length = length(errors);
disp(errors_length);
```

```
% Построение графика зависимости ошибки от стандартного отклонения шума
std_dev_range = 1:0.5:500;
figure;
plot(std_dev_range, errors, 'LineWidth', 2);
xlabel('Стандартное отклонение шума');
ylabel('Ошибка амплитуды первой компоненты');
title('Зависимость ошибки от СКО шума');
grid on;
```

Результат работы программы

Полученный график изображён на рисунке 10.

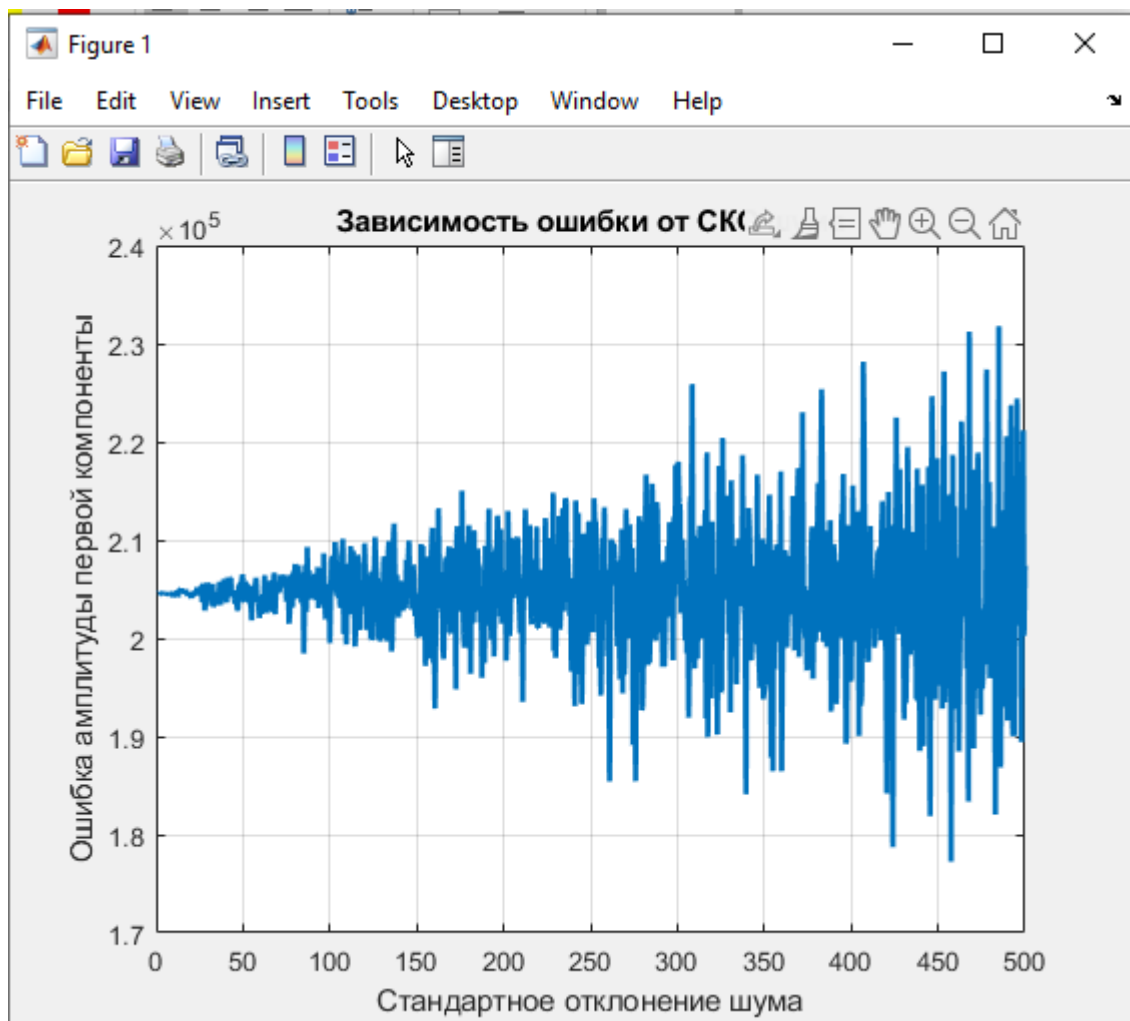


Рисунок 10 – График зависимости ошибки от СКО

Из полученного графика можно сделать вывод о том, что ошибка амплитуды первой компоненты сигнала увеличивается с увеличением СКО. Это означает, что при более высоком уровне шума точность определения амплитуды сигнала ухудшается.

Самому высокому пику на графике из рисунка 10 соответствует по оси абсцисс значение 485. График показан на рисунке 11.

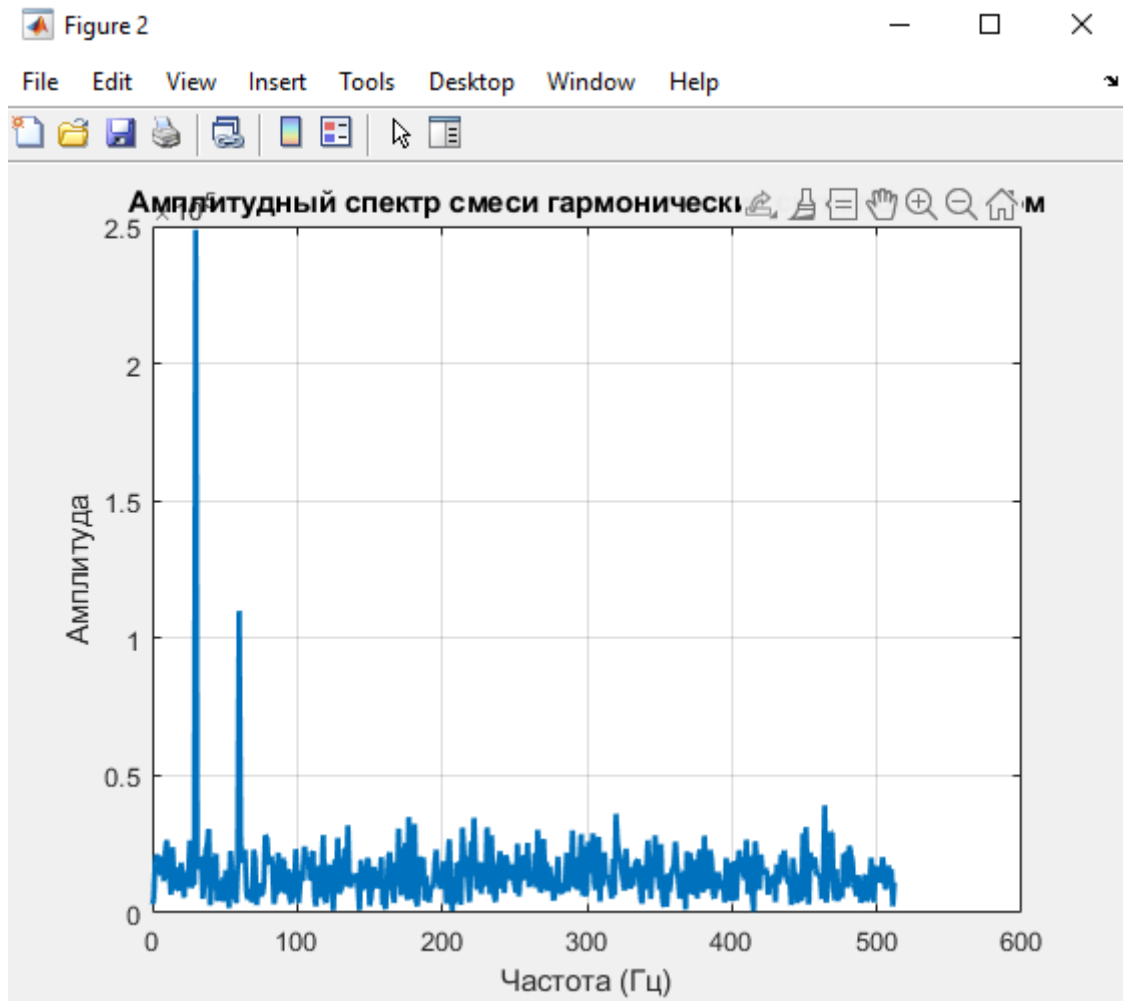


Рисунок 11 – Сигнал с шумом

Выводы

В данной лабораторной работе было проведено ознакомление с методикой моделирования и обработки периодических дискретных сигналов с

П

О

М

О

Щ

Б

Ю

П

а

к