

УДК 004.424

БЫСТРОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ В MATLAB**Брыжинский Кирилл Алексеевич**
магистрант**Никулин Алексей Владимирович**
магистрант**Рыскин Константин Эдуардович**
магистрант

Мордовский государственный университет, Саранск

Аннотация. В данной статье рассматривается алгоритм быстрого преобразования Фурье с использованием языка программирования MATLAB.

Ключевые слова: алгоритм; программирование; MATLAB; быстрое преобразование Фурье.

FAST FOURIER TRANSFORM IN MATLAB**Bryzhinskiy Kirill Alekseevich**
undergraduate**Nikulin Aleksey Vladimirovich**
undergraduate**Ryskin Konstantin Eduardovich**
undergraduate
Mordovian State University, Saransk

Abstract. In this article the algorithm of fast Fourier transform with use of the MATLAB programming language is considered.

Keywords: algorithm programming; MATLAB; fast Fourier transform in MATLAB.

Зачастую необходимо оперативно установить наличие гармонических составляющих в сигнале. В большинстве случаев для первичной примерной оценки может использоваться алгоритм быстрого преобразования Фурье (БПФ). Стоит отметить, что он имеется почти во всех современных математических пакетах, а также библиотеках. Более того, самостоятельно реализовать его тоже можно без особых сложностей. Однако, при существующей простоте, данный метод вызывает определенные сложности, когда надо не только посмотреть наличие гармонических составляющих в сигнале, но и определить их абсолютные значения, то есть привести в норму полученный после выполненных расчетов результат.

В данной статье на примере MATLAB мы попробуем максимально детализировано объяснить, что именно и каким образом выдает в качестве результата алгоритм БПФ.

MATLAB предоставляет возможность сэкономить время на удалении лишних объектов в ручном режиме. Но при работе с крупномасштабными массивами данных MATLAB нередко может ссылаться на недостаточное количество свободной памяти. Чтобы освободить память, Вы можете использовать функцию `clear`. При этом обязательно необходимо указать имя объекта, который надо удалить.

Для начала стоит отметить, что все нужные данные и показатели изначально мы будем генерировать самостоятельно. Поэтому можно удалять все накопившееся в рабочем пространстве в течение активной сессии. Для этого достаточно просто добавить к процедуре `clear` приставку `all`:

```
clear all% Очистка памяти
```

Итак, в первую очередь, необходимо задать исходные данные. Алгоритм БПФ является оптимальным решением для выделения необходимых гармонических сигналов даже на фоне достаточно сильных по-

мех. Чтобы показать это, будем использовать в виде сигнала сумму конкретной постоянной, а также двух синусоид, которые обладают различной частотой и, конечно же, амплитудой. При этом дисперсию шума установим в три раза больше показателей амплитуды 1-ой синусоиды. Также укажем число частотных полос, которые нужно будет подсчитать используемому нами алгоритму БПФ. Хотелось бы обратить внимание, что в конце каждой строки абсолютно не обязательно надо ставить точку с запятой. В таком случае результат вычисления, а также задания переменных просто будут дополнительно отображаться в командной строке. Данную особенность Вы имеете возможность использовать для дополнительной отладки кода. Однако для более комфортной работы мы рекомендуется использовать точку с запятой.

```
%% Параметры
Tm=5;% Длина сигнала
Fd=512;% Частота дискретизации
Ak=0.5;% Постоянная составляющая
A1=1;% Амплитуда первой синусоиды
A2=0.7;% Амплитуда второй синусоиды
F1=13;% Частота первой синусоиды
F2=42;% Частота второй синусоиды
Phi1=0;% Начальная фаза первой синусоиды
Phi2=37;% Начальная фаза второй синусоиды
An=3*A1;% Дисперсия шума
FftL=1024;% Количество линий Фурье спектра
```

MATLAB предназначен в первую очередь для работы с массивами. Почти все алгоритмы счета в данном пакете прикладных программ оптимизированы для эффективной работы именно с векторами. Широкое разнообразие удобных и продуманных инструментов работы стимулирует представлять максимально возможное количество исходных данных именно в форме матриц. Например, имеется возможность с легкостью

сгенерировать массив величин, которые как убывают, так и возрастают, с установленным шагом:

```
%% Генерация рабочих массивов
T=0:1/Fd:Tm;% Массив отсчетов времени
```

Чтобы задать случайный Гауссов шум, используется функция под названием `randn`. С её помощью формируется массив размерности, который был заранее установлен в ее показателях. Для единообразия предлагается установить его в виде строки длиной, которая соответствует длине нашего массива отсчетов времен, вычисленной с помощью функции `length`:

```
Noise=An*randn(1,length(T));% Массив случайного шума длиной
равной массиву времени
```

Символ `*` применяется для обозначения проведения операции перемножения. Поскольку в большинстве случаев действия выполняется над векторами, то вполне логично, что и умножение в данном случае подразумевается векторное. Однако легко можно применять его для поэлементного перемножения. Для этого необходимо добавить перед вышеуказанным символом точку (`.*`). Хотелось бы обратить внимание, что при перемножении вектора на скаляр необязательно ставить точку перед символом умножения.

```
Signal=Ak+A1*sind((F1*360).*T+Phi1)+A2*sind((F2*360).*T+Phi2);
% Массив сигнала (смесь 2х синусоид и постоянной составляющей)
```

Теперь более детально рассмотрим функцию алгоритма БПФ. Элементами стандартной функции пакета прикладных программ MATLAB выступают сигнал, размерность вектора-результата и измерение.

Последний параметр устанавливает вдоль какого именно измерения находится сигнал, если на вход будет подан многомерный массив. Нередко данный аргумент абсолютно ошибочно ассоциируют с размерно-

стью преобразования Фурье. Это предположение является ложным. Хотя в Matrix Laboratory существуют реализации двумерного и даже многомерного алгоритмов БПФ. Поскольку наш сигнал является вектором, то при желании его можно без каких-либо последствий опустить.

Для начала рассмотрим сигнал, в котором шум отсутствует. Это позволит более точно понять суть используемого алгоритма. В виде результата появится вектор комплексных чисел, представляющий наш сигнал в частотном домене в форме показателей. То есть модули данных комплексных чисел представляют собой амплитуды соответствующих частот. Хотелось бы обратить внимание, что в свою очередь аргументы выступают их начальными фазами. При этом полученная фаза определяется в радианах. Важно подчеркнуть, что с амплитудой и частотами ситуация немного сложнее.

К примеру, если мы будем использовать к сигналу БПФ, и при этом будем брать абсолютные значения вектора на выходе, тогда у нас выйдет примерно такое изображение (см. рис. 1).

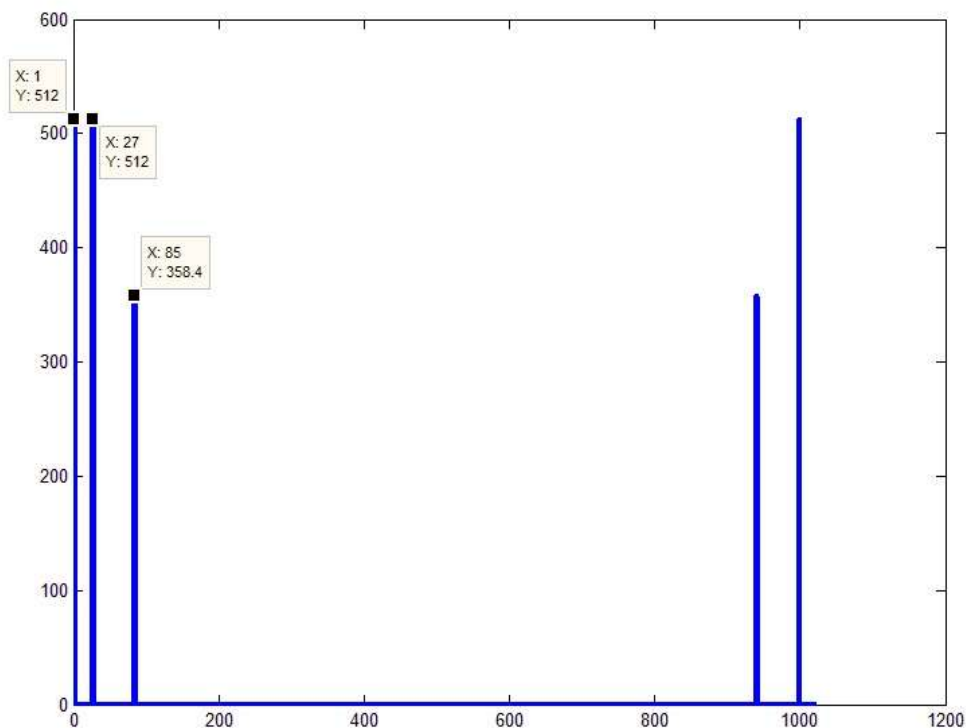


Рис. 1. Абсолютные значения вектора на выходе

Чтобы построить двухмерные графики мы рекомендуем применять функцию под названием plot. Главными параметрами, которые применяются в указанной функции, выступают одномерные массивы точек. Если передать один массив, тогда он отображается с фиксированным шагом 1.

Если более внимательно изучить полученное изображение то можно прийти к выводу, что оно немного отличается от существующих у нас ожиданий. На данном графике можно увидеть пять пиков вместо планируемых трёх. При этом их амплитуды абсолютно не совпадают с амплитудами имеющих исходных сигналов.

В первую очередь надо помнить, что счет алгоритма предусматривает переборку всех частот. То есть для работы используются и положительные и отрицательные частоты. В итоге правая сторона «зеркально» отображает реальный спектр. То есть ноль должен находиться именно на середине массива. Сложившуюся ситуацию можно без каких-либо лишних проблем и сложностей поправить. Для этого достаточно выполнить циклический сдвиг на $1/2$ длины массива. Для решения данной задачи в Матлаб имеется функция сдвига под названием fftshift(). Она смещает первый элемент аккуратно в середину массива (см. рис. 2).

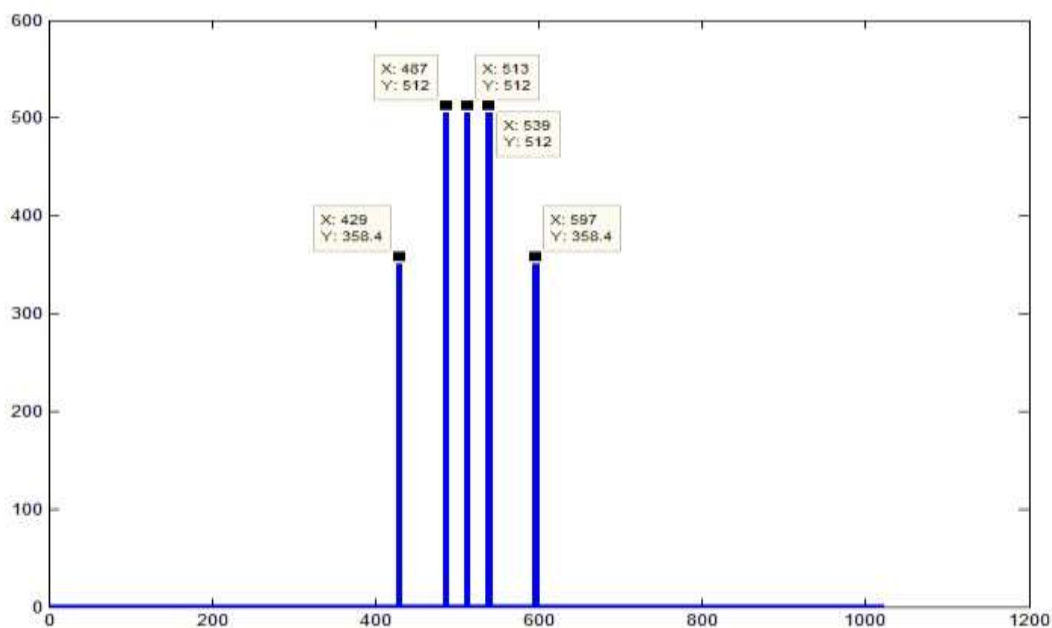


Рис. 2. Функция сдвига fftshift()

Теперь перейдем на ось значений.

По теореме отсчетов спектр дискретного сигнала должен быть ограничен $1/2$ частоты дискретизации (F_d). В рассматриваемом нами случае – это $F_d/2$ слева, а также $F_d/2$ справа. То есть весь полученный массив должен полностью покрывать F_d частот. Поскольку мы сами задали длину массива, мы должны получить частоты в виде полного массива значений в диапазоне от $-F_d/2$ до $F_d/2$ с шагом $F_d/F_{ft}L$ (см. рис. 3).

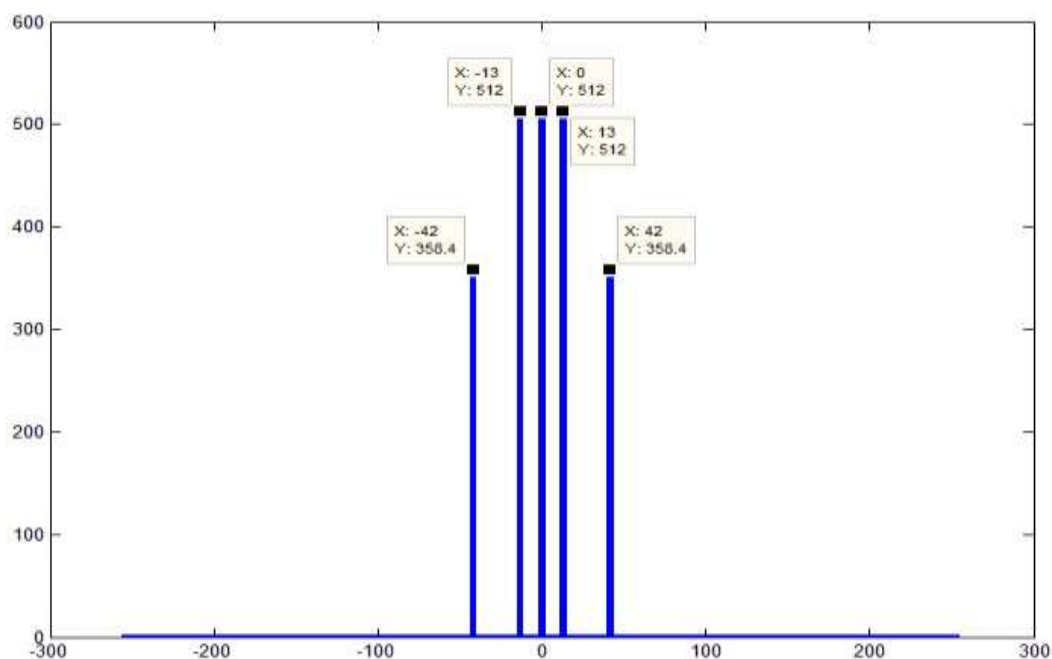


Рис. 3. массив значений в диапазоне от $-F_d/2$ до $F_d/2$ с шагом $F_d/F_{ft}L$

Если обратить более детальное внимание на фазы частот, то можно увидеть весьма интересную картину. Они абсолютно равны негативным фазам соответствующих негативных частот. С учетом равенства амплитуд левой, а также правой частей спектра, а также полного соответствия их фаз, весь спектр у нас получится эквивалентным собственной положительной части, которая при этом обладает удвоенной амплитудой. Из данного правила исключением будет лишь нулевой элемент, не имеющий своей зеркальной половины. Так, мы получаем отличную возможность избавиться от в большинстве случаев абсолютно ненужных отри-

цательных частот. Выполнить данную задачу можно было и сразу. Для этого достаточно было просто заранее отбросить конец исходного массива, а также умножить оставшиеся элементы на два (безусловно, без постоянной составляющей) (см. рис. 4).

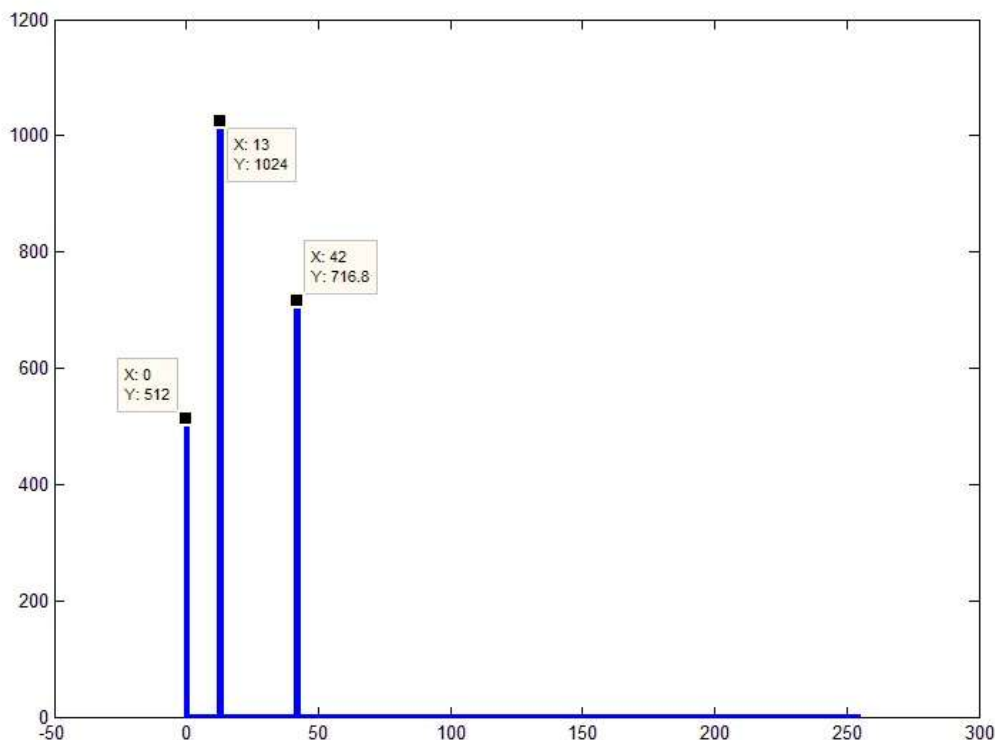


Рис. 4. Исключение конца исходного массива

В результате перед нами уже почти ожидаемый результат. Последней помехой остаются амплитуды, но с ними тоже можно без проблем справиться. Не секрет, что БПФ являет собой суммирование сигнала, который был умножен на ядро преобразования для каждой из частот. Поэтому реальный идеальный результат будет меньше того, который был получен ровно в число суммирований. То есть полученный результат необходимо просто разделить на число элементов в нём. Обращаем внимание, что под результатом следует понимать весь ответ, в который так же входит отброшенная часть (см. рис. 5).

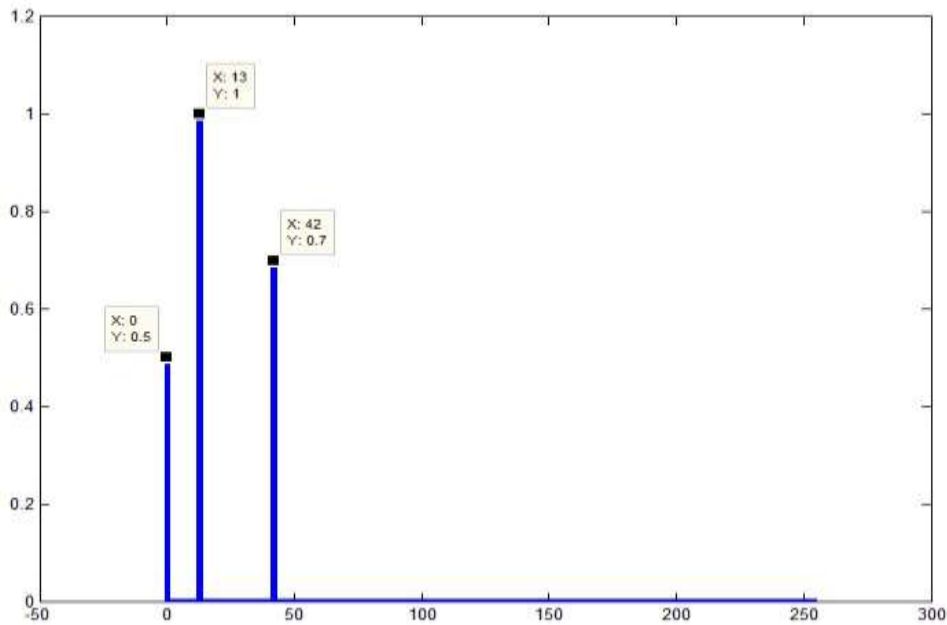


Рис. 5. Конечный результат Быстрого преобразования Фурье

Надо не забывать про еще одну крайне важную вещь. В спектральном представлении происходит вычисление не значения сигнала на частоте, на которую непосредственно попал алгоритм. В данном случае вычисляется значение в полосе. То есть если в указанную полосу попало сразу несколько гармонических составляющих, то они просто складываются. Для наглядности можно уменьшить число линий в результате выполнения алгоритма (см. рис. 6).

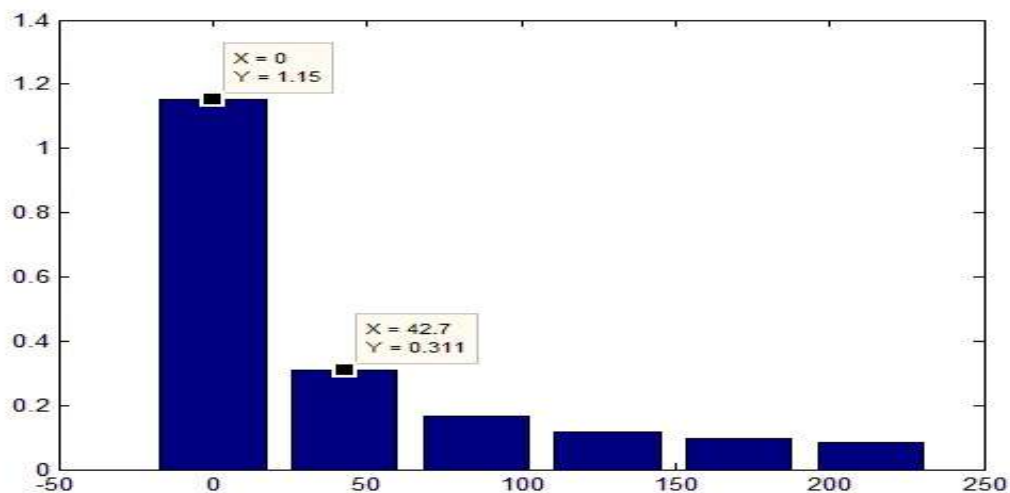


Рис. 6. Результат уменьшения числа линий в результате выполнения алгоритма

Но хотелось бы заранее предупредить возможную ошибку. Не следует необоснованно повышать точность работы, поскольку это тоже может стать причиной определенных негативных последствий. Так, к примеру, если выбранное разрешение будет сопоставимо с имеющейся частотой дискретизации сигнала, тогда в спектр начнут входить гармоники «окна», относящиеся не к настоящему сигналу, а всего лишь его дискретному представлению (см. рис. 7).

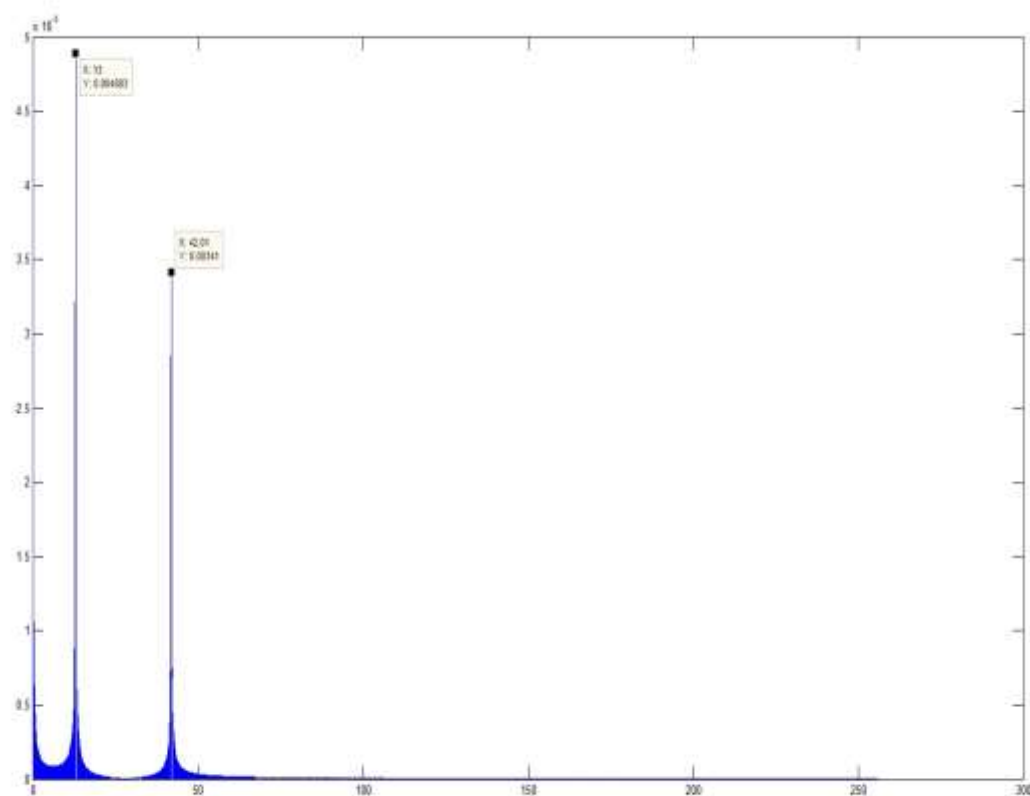


Рис. 7. Результат при ошибочном повышении точности работы

Либо более близко окрестности одной из гармонических составляющих (см. рис. 8).

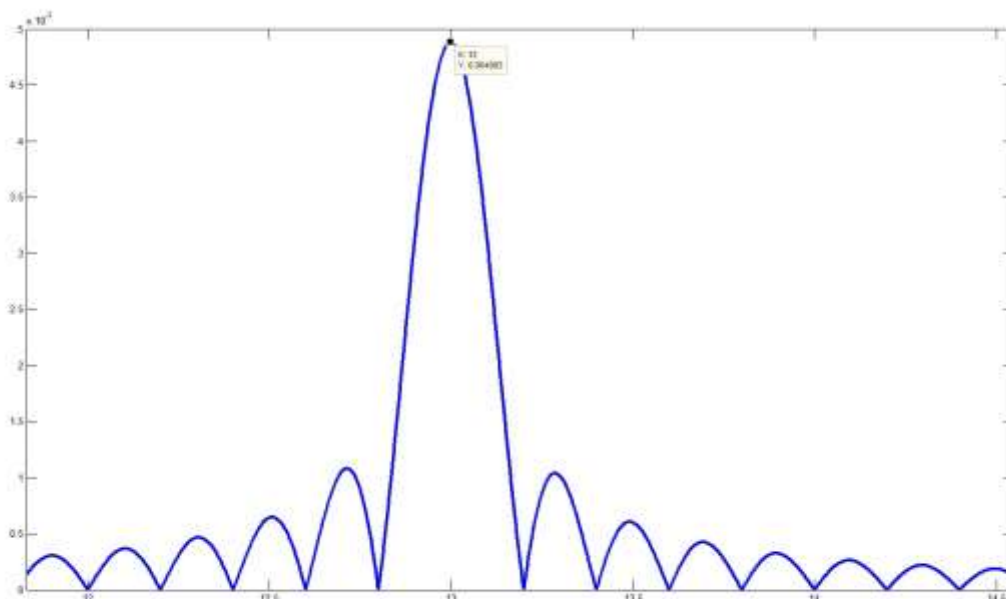


Рис. 8. Результат при ошибочном повышении точности работы

Код для нормировки fft должен выглядеть примерно так:

```
%% Спектральное представление сигнала
FftS=abs(fft(Signal,FftL));% Амплитуды преобразования Фурье
сигнала
FftS=2*FftS./FftL;% Нормировка спектра по амплитуде
FftS(1)=FftS(1)/2;% Нормировка постоянной составляющей в
спектре
FftSh=abs(fft(Signal+Noise,FftL));% Амплитуды преобразования
Фурье смеси сигнал+шум
FftSh=2*FftSh./FftL;% Нормировка спектра по амплитуде
FftSh(1)=FftSh(1)/2;% Нормировка постоянной составляющей в
спектре
```

Теперь осталось лишь грамотно и аккуратно вывести результаты. Для этого существует функция под названием subplot, которая предоставляет возможность разбить окно сразу на несколько областей для более удобного отображения полученных графиков.

```

%% Построение графиков
subplot(2,1,1);% Выбор области окна для построения
plot(T,Signal);% Построение сигнала
title('Сигнал');% Подпись графика
xlabel('Время (с)');% Подпись оси x графика
ylabel('Амплитуда (Попугаи)');% Подпись оси y графика
subplot(2,1,2);% Выбор области окна для построения
plot(T,Signal+Noise);% Построение смеси сигнал+шум
title('Сигнал+шум');% Подпись графика
xlabel('Время (с)');% Подпись оси x графика
ylabel('Амплитуда (Попугаи)');% Подпись оси y графика
F=0:Fd/FftL:Fd/2-1/FftL;% Массив частот вычисляемого спектра
Фурье
figure% Создаем новое окно
subplot(2,1,1);% Выбор области окна для построения
plot(F,FftS(1:length(F)));% Построение спектра Фурье сигнала
title('Спектр сигнала');% Подпись графика
xlabel('Частота (Гц)');% Подпись оси x графика
ylabel('Амплитуда (Попугаи)');% Подпись оси y графика
subplot(2,1,2);% Выбор области окна для построения
plot(F,FftSh(1:length(F)));% Построение спектра Фурье сигнала
title('Спектр сигнала');% Подпись графика
xlabel('Частота (Гц)');% Подпись оси x графика
ylabel('Амплитуда (Попугаи)');% Подпись оси y графика

```

Результат выполнения кода должен выглядеть так (см. рис. 9):

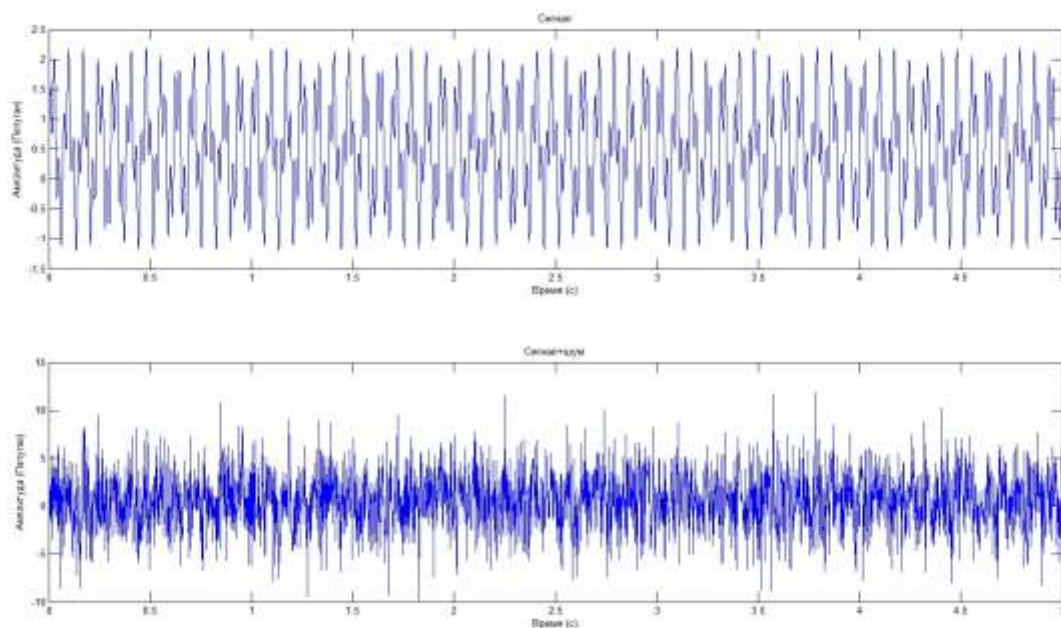


Рис. 9. Результат выполнения кода нормировки fft

На фоне шума нельзя увидеть полезный сигнал, однако спектральная характеристика предоставляет возможность точно установить его частоту и амплитуду.

В заключении хотелось бы отметить, что, несмотря на простоту использования метода БПФ, он является крайне удобным инструментом Фурье-анализа и может применяться во многих сферах деятельности человека.

Список использованных источников

1. Афонин В. В. Методы моделирования и оптимизации с примерами на языке C/C++ и Matlab / В.В. Афонин, В.В. Никулиню Саранск, 2015. Т. Часть I Методы моделирования.
2. Никулин А.В. Модель быстродействующего управляемого генератора квадратурных гармонических сигналов / А.В. Никулин, В.В. Никулин // Огарёв-Online. 2016. № 18 (83). С. 10.
3. Дубровин В.С. Способ формирования линейно изменяющегося сигнала / В.С. Дубровин, В.В. Никулин // Вимірювальна та обчислювальна техніка в технологічних процесах. 2013. № 4 (45). С. 72-76.
4. Дубровин В.С. Способ построения управляемых функциональных генераторов / В.С. Дубровин, В.В. Никулин // Наукоемкие технологии в космических исследованиях Земли. 2013. Т. 5. № 2. С. 16-23.
5. Дубровин В.С. Спектральный анализ квазилинейного сигнала треугольной формы / В.С. Дубровин, В.В. Никулин // XXXVII Огаревские чтения. Матер. науч. конф. 2009. С. 134-136.
6. Никулин А.В. Построение модели быстродействующего управляемого генератора квадратурных гармонических сигналов / А.В. Никулин, В.В. Никулин // Фундаментальные проблемы системной безопасности материалы III школы-семинара молодых ученых: в 2 ч. 2016. С. 135-139.