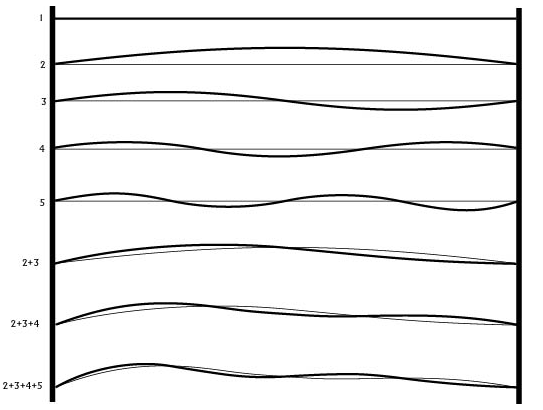
О рядах Фурье и методах гармонического анализа

В связи с чем возник этот метод и в чем его суть?

Предположим, есть некоторая функция, и требуется узнать, можно ли, и если можно, то как, представить эту функцию в виде наложения синусоид с разными амплитудами, фазами и частотами?

Впервые эта задача появилась в конце XVIII века в связи с решением вопроса о форме колебаний струны. Было известно и из теории, и из практики, что струна может колебаться по синусоиде, притом так, что на ее длине всегда умещается целое число полуволн, что обусловлено закреплением концов струны. Эти простейшие формы колебаний струны стали называть «гармониками», отсюда и название метода - «метод гармонического анализа».





См. комментарии к <http://www.forumklassika.ru/entry.php?b=3865>

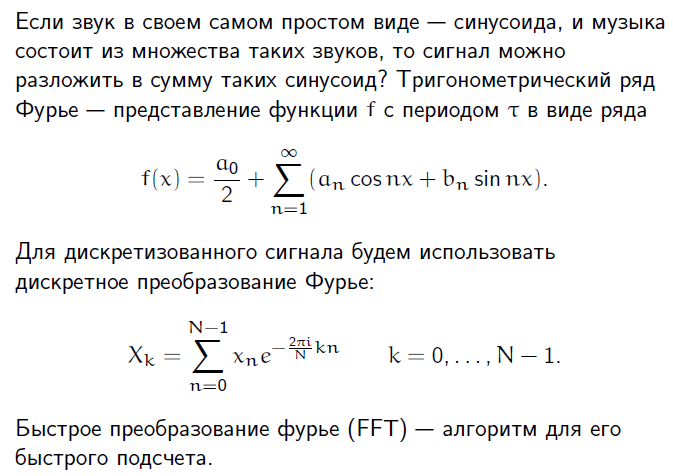
Но было известно и другое, что струна может совершать и другие, более сложные по форме колебания. Возникал вопрос: можно ли эти сложные формы представить, как сумму произвольного, возможно бесконечного числа простейших форм – синусоидальных гармоник?

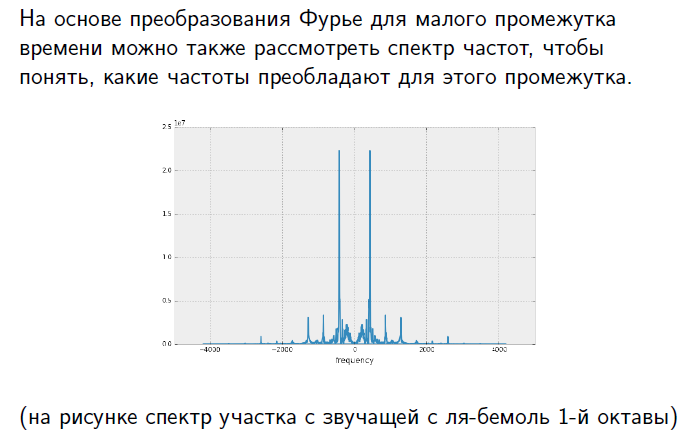
Впоследствии, в начале XIX в., Фурье применил аналогичный метод к решению совершенно другой задачи – задачи о распространении тепла в металлической пластине. Но настоящую инженерную популярность метод приобрел гораздо позже, когда стала развиваться электротехника. Все классические методы электротехники были разработаны для цепей синусоидального тока. Возникал вопрос, а что делать, если токи и напряжения в цепях имеют другую форму?

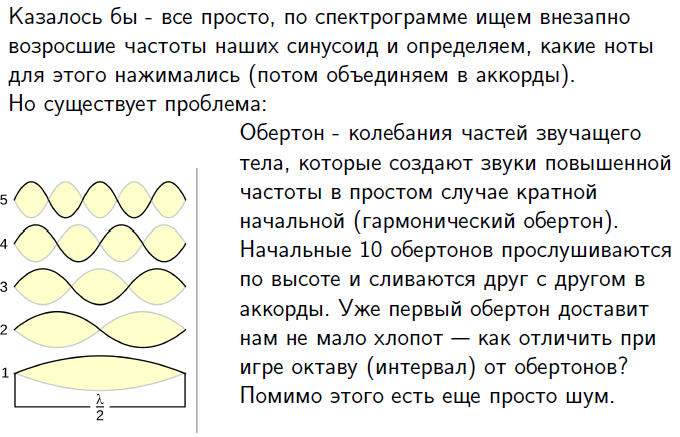
Решено было представить произвольный, хотя бы даже периодический сигнал, как сумму бесконечного числа гармоник.

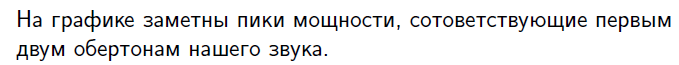
Популярность этого метода среди инженеров стала расти по мере того, как стала приобретать важность задач обработки сигналов. Метод Фурье основан на том, что исходный сигнал разлагался по ортогональным функциям – синусоидам кратной частоты. Преобразованный сигнал называется «спектром». Для периодических функций спектр - линейчатый, для непериодических – он непрерывный. Если в исходном сигнале появляются помехи – «шум», то на спектрограмме возникают дополнительные пики на частотах «шума», которые можно удалить, а затем с помощью обратного преобразования Фурье вернуться к исходной функции и получить чистый сигнал без помех.

С переходом на оцифрованные сигналы стали широко применяться электронные тюнеры(прибор для настройки духовых, смычковых и струнных инструментов). Пользователь, скачавший файл с электронным тюнером на свой смартфон, возможно и не догадывается, что он получил математическую программу с преобразованием Фурье. (Первый механический тюнер - Strobocon появился в 1936 году, и был основанный на стробоскопическом эффекте. С 1985 года и до настоящего времени его выпускает фирма Peterson).

****

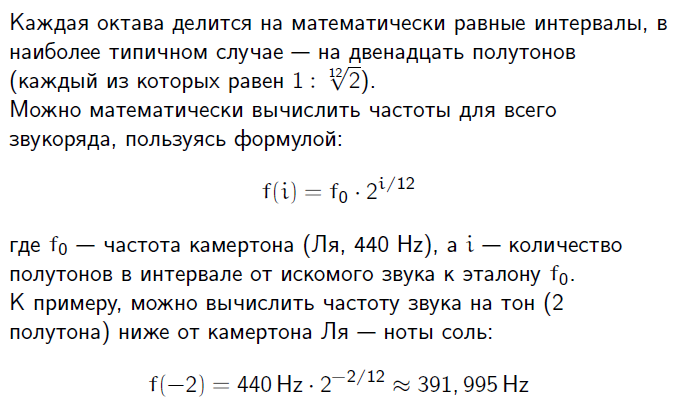
****

****

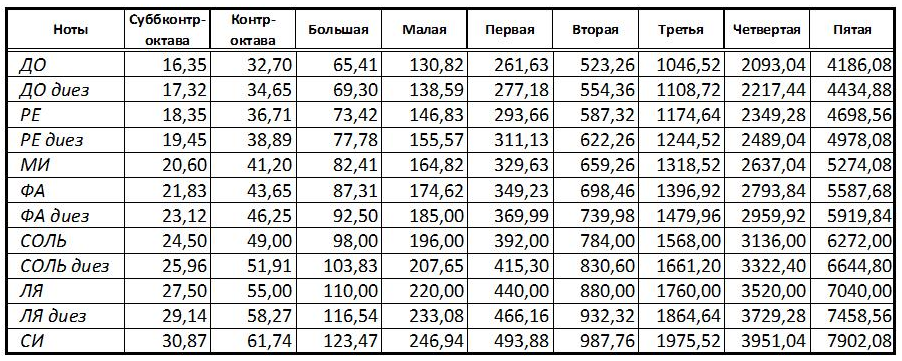
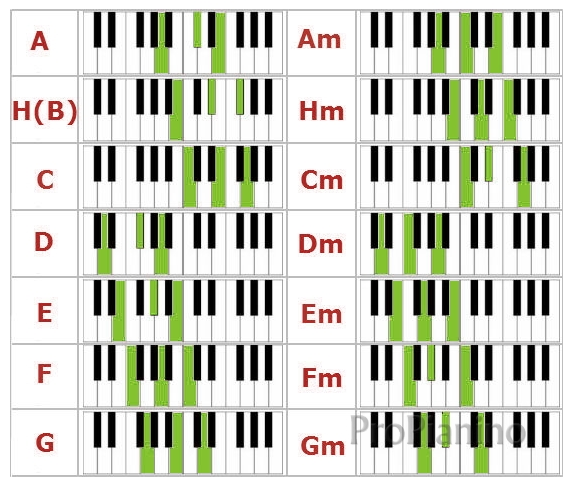
****

**Продолжение см.**

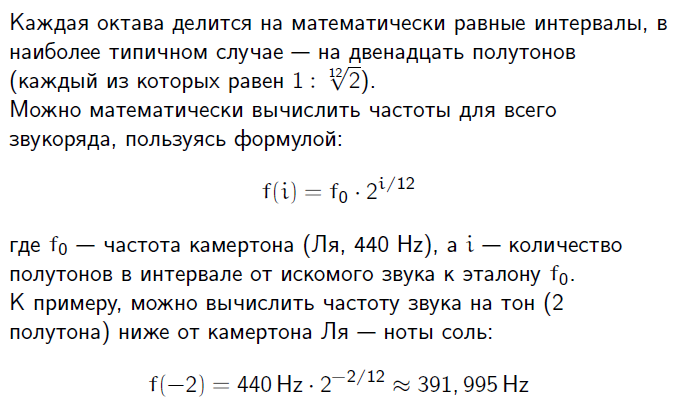
[www.machinelearning.ru/wiki/images/c/cf/NizhibitskyMusicSlides.pdf](http://www.machinelearning.ru/wiki/images/c/cf/NizhibitskyMusicSlides.pdf )

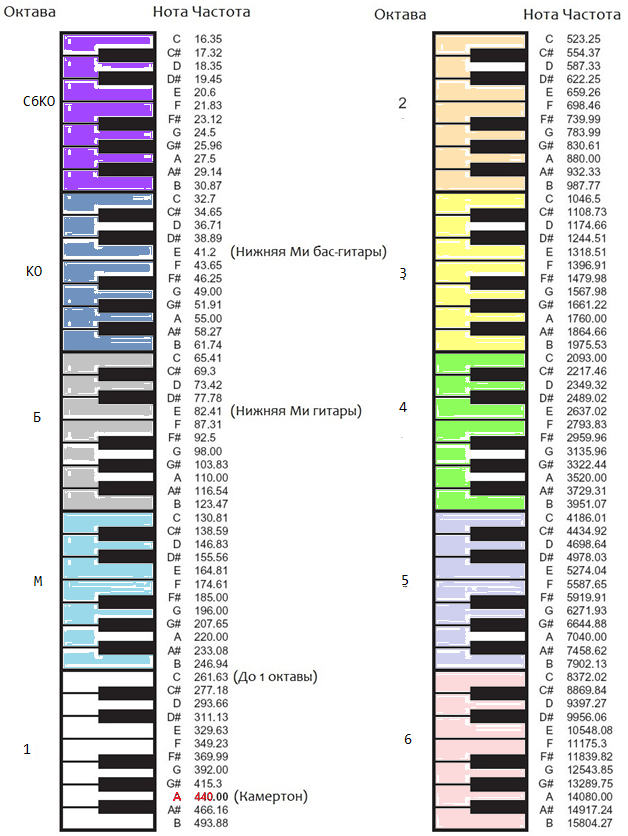


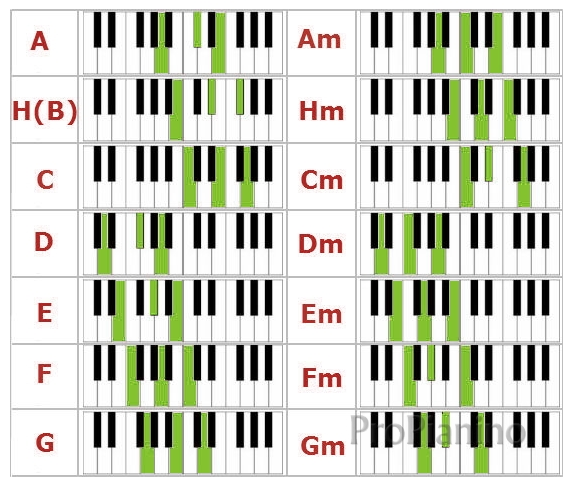
**Таблица соотношения нот и частот**



<http://muz-teoretik.ru/kak-poyavilis-nazvaniya-not/> происхождение названий нот – первые слоги начальных 6 строк в гимне св. Иоанну Крестителю





**Примеры построения аккордов на фортепьяно**

**Таблица буквенных обозначений тональностей и аккордов**

# Интернет – пример FFT

<https://matlab.ru/videos/vvedenie-v-cifrovuju-obrabotku-signalov-COS-prodolshenie>

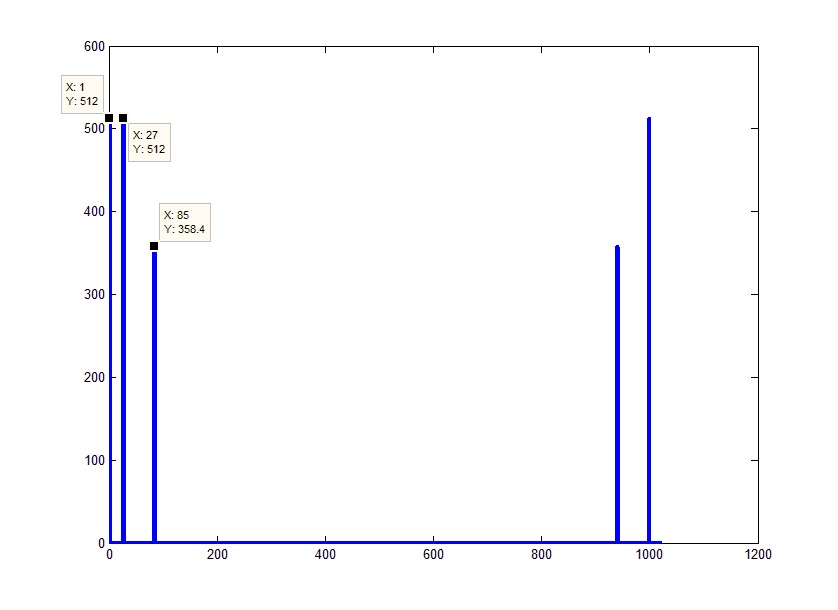
MATLAB и быстрое преобразование Фурье

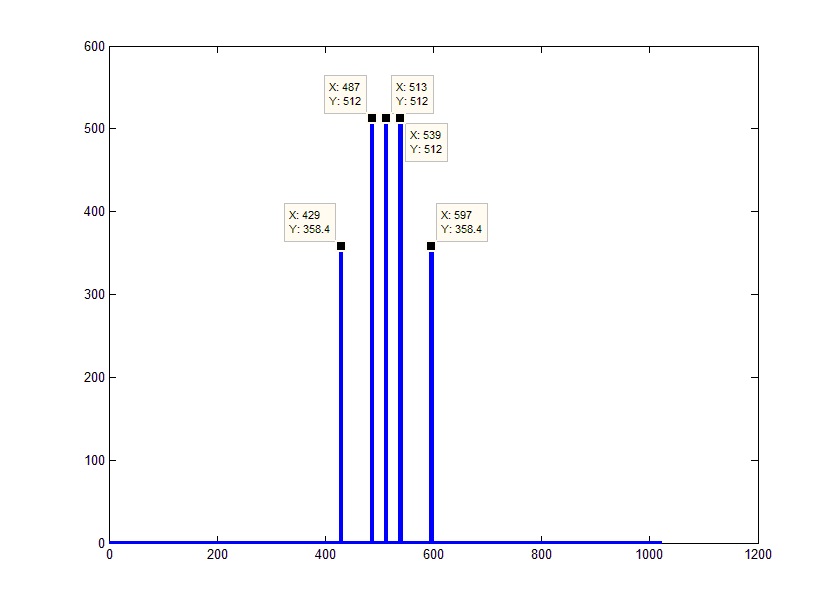
По работе неоднократно сталкивался с необходимостью быстро определить наличие в сигнале гармонических составляющих. Часто для примерной оценки достаточно воспользоваться алгоритмом быстрого преобразования Фурье. Тем более, что его реализации есть практически во всех математических пакетах и библиотеках, да и собственноручно реализовать не составит особого труда. Между тем, опыт показывает, что, при всей своей простоте, метод начинает вызывать некоторые вопросы, когда возникает необходимость не просто посмотреть наличие дискреток в сигнале, но и выяснить их абсолютные значения, т.е. нормализовать полученный результат.  
  
В этой статье я постараюсь объяснить, что же все-таки выдает в качестве результата fft (Fast Fourier transform) на примере MATLAB (и проведу небольшой ликбез по этому весьма полезному, на мой взгляд, языку).  
  
Можно смело удалять все, что накопилось в рабочем пространстве за активную сессию, добавив ключевое слово all:  
  
clear all % Очистка памяти  
  
*Фурье анализ идеально подходит для выделения гармонических сигналов на фоне помех*.

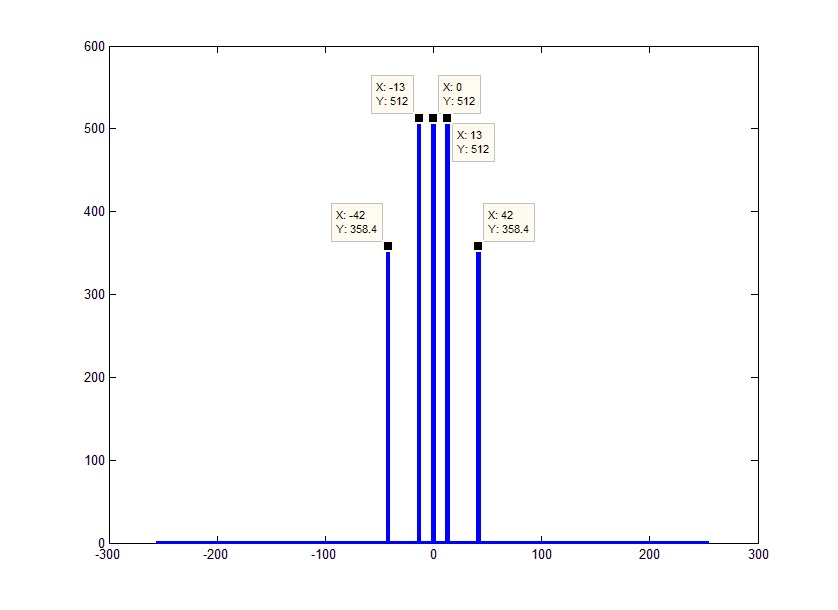
Для того чтобы продемонстрировать это, возьмем в качестве сигнала сумму некоторой постоянной величины и двух синусоид с разной частотой и амплитудой. Дисперсию шума An возьмем в 3 раза больше амплитуды первой синусоиды. Так же зададим количество частотных полос FftL, в которых должен будет проводить вычисления fft алгоритм.

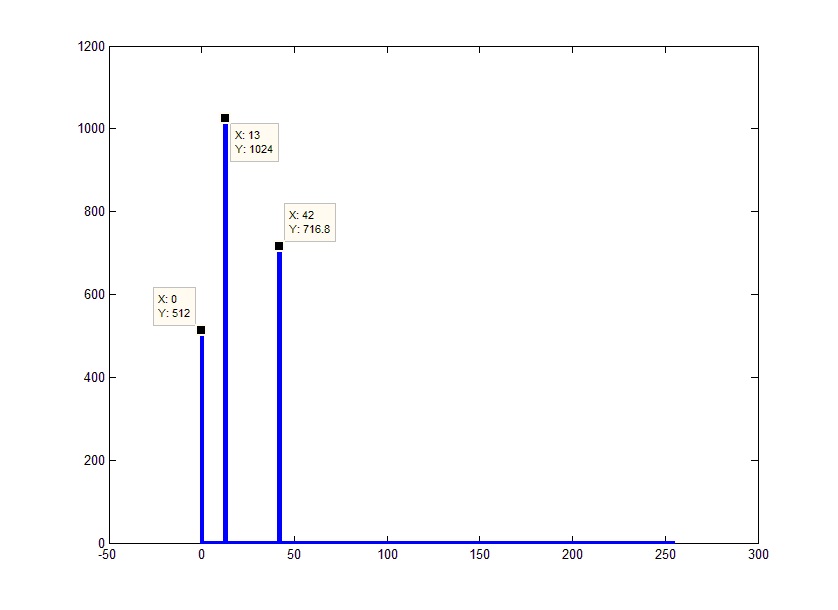
%% Параметры  
Tm=5; % Длина сигнала (с)  
Fd=512: % Частота дискретизации (Гц)  
Ak=0.5; % Постоянная составляющая (Ед)  
A1=1; % Амплитуда первой синусоиды (Ед)  
A2=0.7; % Амплитуда второй синусоиды (Ед)  
F1=13; % Частота первой синусоиды (Гц)  
F2=42; % Частота второй синусоиды (Гц)  
Phi1=0; % Начальная фаза первой синусоиды (Градусов)  
Phi2=37; % Начальная фаза второй синусоиды (Градусов)  
An=3\*A1; % Дисперсия шума (Ед)  
FftL=1024; % Количество линий Фурье спектра  
  
  
MATLAB (Matrix Laboratory), как следует из названия, предназначен прежде всего для работы с массивами, практически все алгоритмы счета в нем оптимизированы для работы с векторами. В частности, можно легко сгенерировать массив возрастающих (убывающих) величин с заданным шагом (в данном примере шаг – величина, обратная частоте дискретизации 1\Fd):  
  
%% Генерация рабочих массивов  
T=0:1/Fd:Tm; % Массив отсчетов времени  
  
  
Случайный Гауссов шум задается функцией **randn**, результатом применения которой является массив размерности, указываемой в параметрах. Зададим его строкой такой же длины, как наш массив моментов времени Т (используем функцию **length**):  
  
Noise=An\*randn(1,length(T));% Массив случайного шума, по длине равный массиву времени Т  
  
При умножении вектора на вектор (матрицу) добавленная перед этим оператором точка делает умножение поэлементным, равно как и деление, и возведение в степень. При умножении скаляра на вектор точка перед умножением не является обязательной:  
  
Signal=Ak+A1\*sind((F1\*360).\*T+Phi1)+A2\*sind((F2\*360).\*T+Phi2); % Массив сигнала (смесь 2х синусоид и постоянной составляющей) % cm  
  
Теперь перейдем к тому, ради чего и затевалась данная статья — к функции **fft**(). Аргументами стандартной функции MATLAB являются: сам сигнал (в нашем случае **Signal**), размерность результирующего вектора (**FftL**), а также индекс (**dim**), задающий направление преобразования вдоль строк или столбцов (если преобразование выполняется над матрицей). Так как наш сигнал представляет собой вектор, то индекс вполне можно опустить.

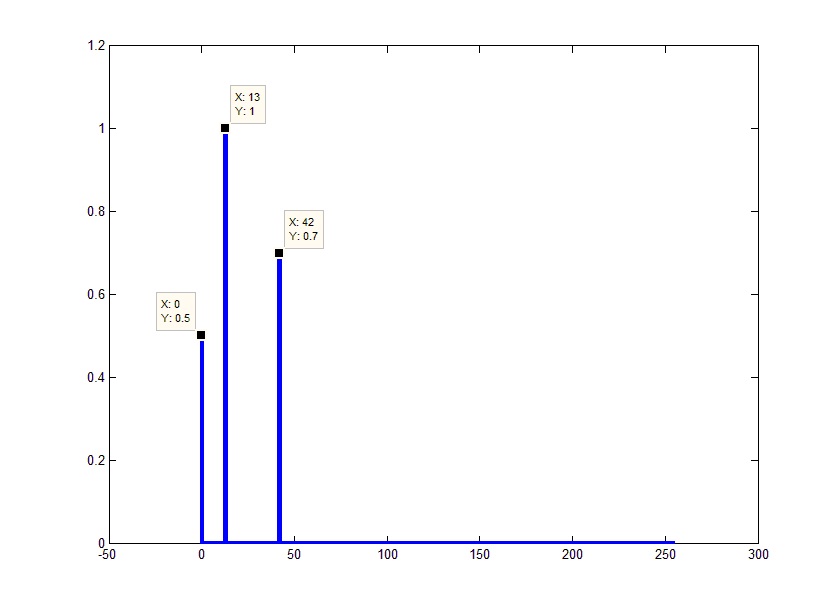
Рассмотрим сначала сигнал без примеси шума. В качестве *результата* мы получим вектор, состоящий из комплексных чисел. Это и есть представление нашего сигнала в частотной области в показательной форме. Т.е. модули этих комплексных чисел представляют амплитуды соответствующих частот (точнее полос частот - см. дальше), а аргументы – их начальные фазы. И если полученная фаза, однозначно вычисляется в радианах, то с амплитудой и частотами не все так просто.

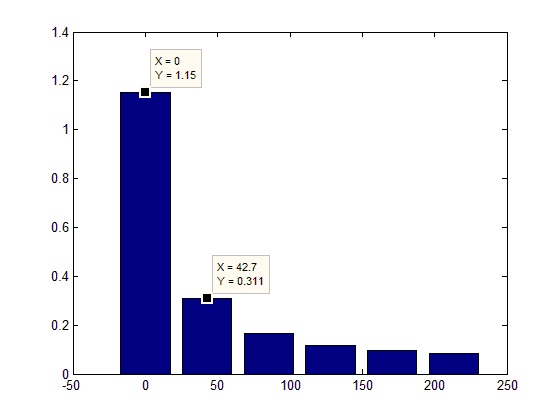
Например, если мы просто применим к нашему сигналу преобразование Фурье, и возьмем абсолютные значения вектора на выходе, то получим приблизительно следующую картинку:  
  
Для построения двухмерных графиков удобно использовать функцию plot. Основные параметры, используемые в этой функции – одномерные массивы точек, первый задает ось ординат, второй – значение функции в соответствующих точках. Если передать только один массив, то он будет отображен с фиксированным шагом 1.  
Если присмотреться к полученной картинке выяснится, что она несколько отличается от наших ожиданий. На приведенном графике 5 пиков вместо ожидаемых 3х (постоянная + 2 синусоиды), их амплитуды не совпадают с амплитудами исходных сигналов, и ось абсцисс вряд ли отображает частоты.  
  
Прежде всего, следует учитывать, что счет алгоритма устроен таким образом, что перебираются не только положительные, но и отрицательные частоты и правая часть графика является «зеркальным» отображением реального спектра. Т.е. на самом деле 0 (которому соответствует постоянная часть сигнала) должен приходиться на середину массива. Ситуацию можно поправить, совершив циклический сдвиг на половину длины массива. Для этих целей, в MATLAB существует функция сдвига **fftshift(),** смещающая первый элемент в середину массива:

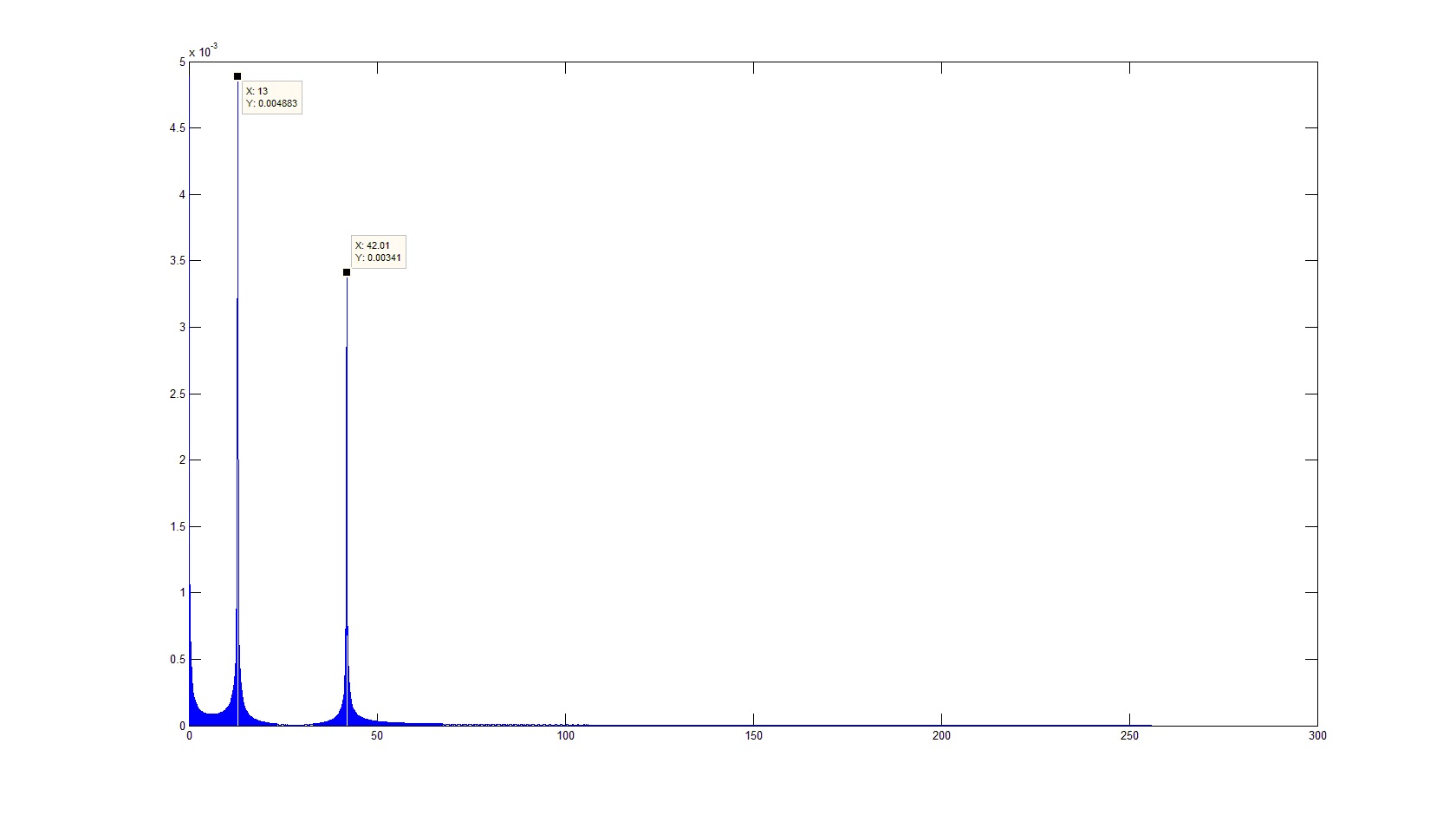


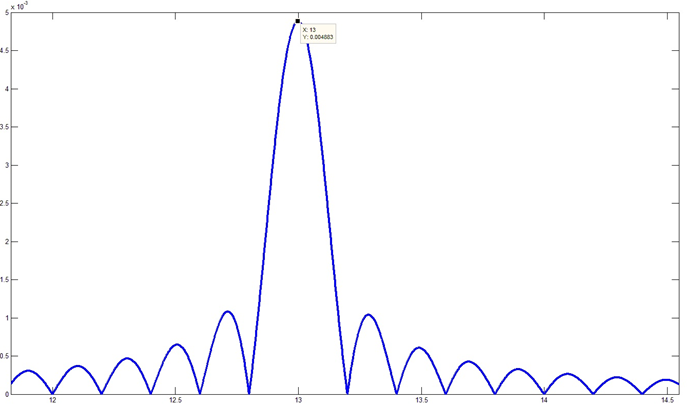
Теперь обратим внимание на ось значений. Согласно теореме отсчетов (так же известной как теорема Найквиста-Шеннона или более патриотично теорема Котельникова) спектр дискретного сигнала будет ограничен половиной диапазона частоты дискретизации (Fd). Или в нашем случае: -Fd/2 слева и Fd/2 справа. Т.е. весь полученный массив покрывает диапазон Fd частот. Отсюда, учитывая что мы знаем (вернее даже самостоятельно задали в качестве параметра) длину массива, получим частоты в виде массива значений от –Fd/2 до Fd/2 с шагом Fd/**FftL** (Надо иметь в виду, что на самом деле крайняя правая частота будет меньше границы на один отсчет т.е. Fd/2-Fd/FftL):  
  


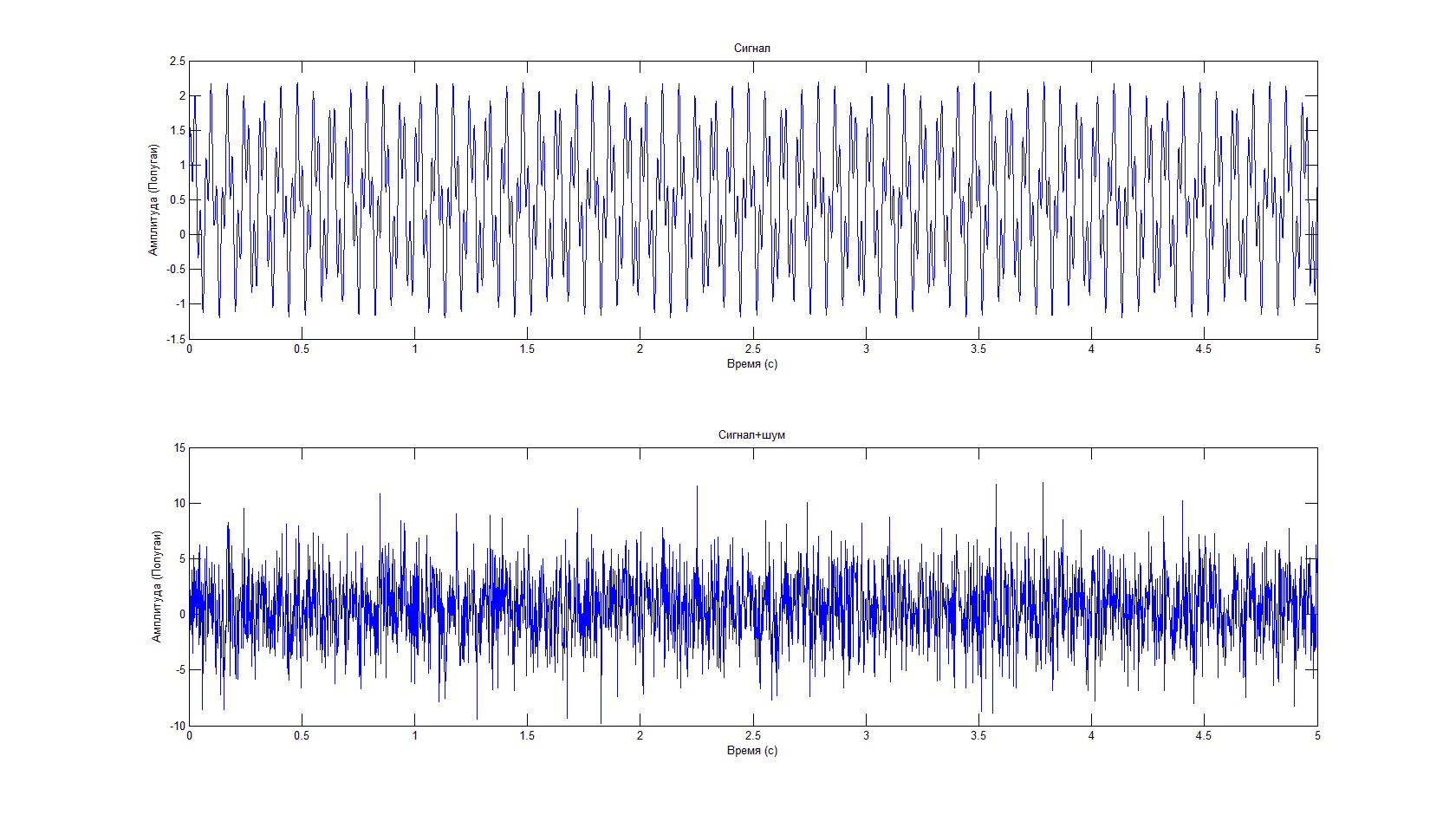
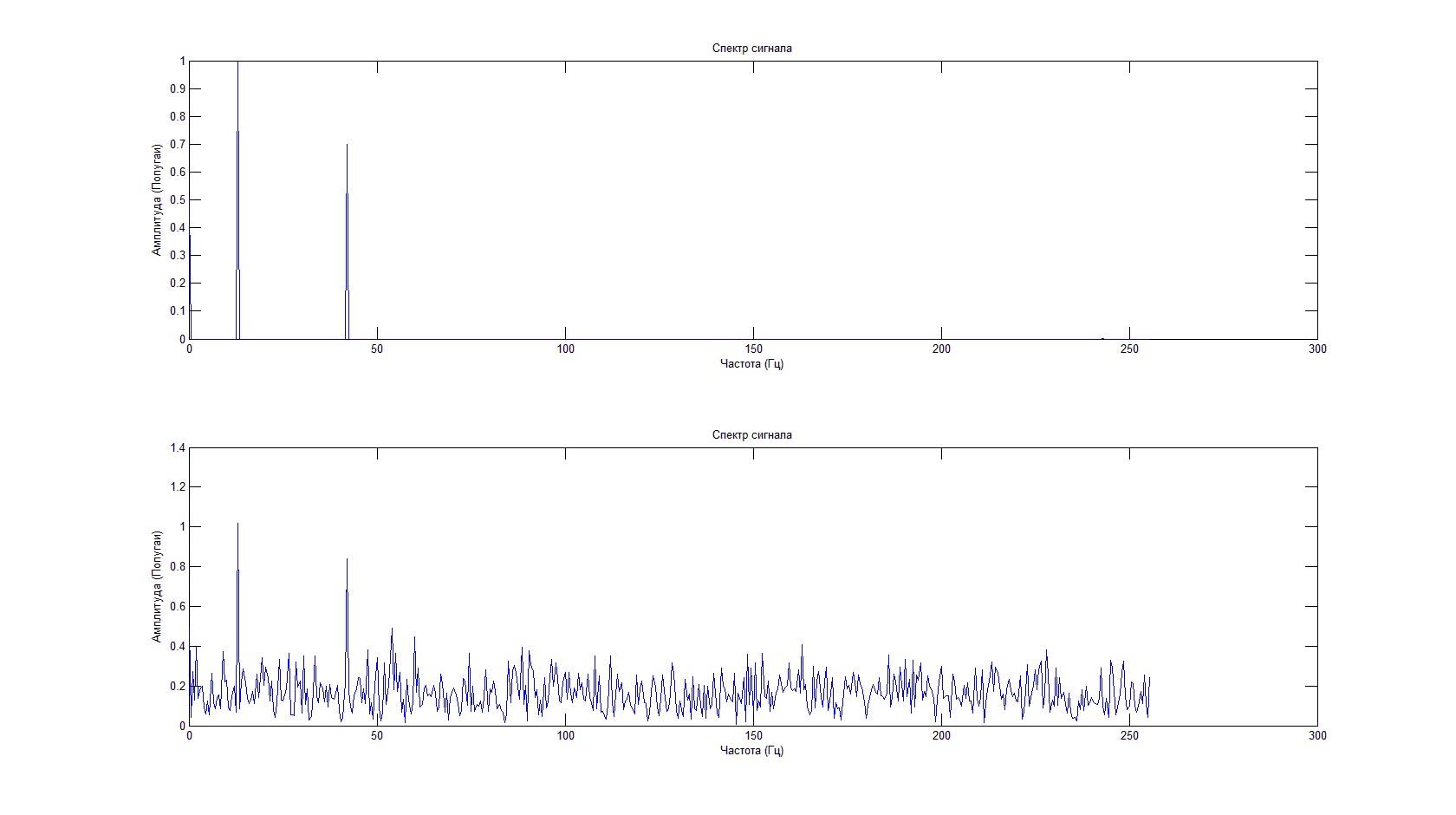
Если посмотреть на фазы частот, можно заметить, что они равны отрицательным фазам соответствующих отрицательных частот. Учитывая равенство амплитуд левой и правой частей спектра и соответствие их фаз с точностью до знака, весь спектр будет эквивалентен своей положительной части с удвоенной амплитудой. Исключение составляет только 0-й элемент, который не имеет зеркальной половины. Таким образом, можно избавиться от «непонятных» и зачастую ненужных отрицательных частот. Это можно было сделать сразу просто отбросив конец исходного массива и помножив амплитуду оставшихся элементов на 2 (за исключением постоянной составляющей):  


Теперь это уже похоже на тот результат, который мы ожидаем. Единственное, что смущает теперь – это амплитуды. Но с этим все достаточно просто. В виду того, что быстрое преобразование Фурье фактически представляет собой суммирование сигнала, перемноженного на ядро преобразования для каждой из частот (комплексную экспоненту), то реальный результат будет меньше полученного ровно *в количество суммирований* (частот). Иными словами, полученный результат надо разделить на количество элементов (не забываем, что под результатом понимается весь дискретный спектр, вместе с отброшенной частью, т.е. наше заданное число **FftL**):  
  


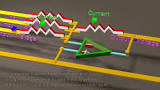
Стоит упомянуть еще одну вещь. В спектральном представлении вычисляется не значение сигнала на той частоте, которая попалась в ходе алгоритма (как мы помним, частоты следуют с шагом Fd/FftL), вычисляется значение в полосе, по ширине, равной шагу. Т.е. если в эту полосу попало несколько «дискреток», то они суммируются. Для примера можно уменьшить количество линий в результате работы алгоритма:   


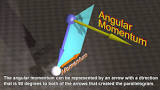
Однако это не означает, что стоит сразу бездумно увеличивать точность работы, т.к. это тоже приводит к негативным последствиям, т.к. если разрешение будет сопоставимо с частотой дискретизации сигнала, в спектр полезут гармоники «окна», которые имеют отношение не к реальному сигналу, а к его дискретному представлению.

Или приближая изображение, в окрестности одной из «дискреток»:  
  


Код для нормировки fft будет выглядеть приблизительно следующим образом:  
  
%% Спектральное представление сигнала  
FftS=abs(fft(Signal,FftL));% Амплитуды преобразования Фурье только сигнала  
FftS=2\*FftS./FftL;% Нормировка спектра по амплитуде (увеличиваем в 2 раза и делим на число задаваемых пиков)   
FftS(1)=FftS(1)/2;% Нормировка постоянной составляющей в спектре сигнала  
FftSh=abs(fft(Signal+Noise,FftL));% Амплитуды преобразования Фурье сигнала с шумом  
FftSh=2\*FftSh./FftL;% Нормировка спектра зашумленного сигнала по амплитуде  
FftSh(1)=FftSh(1)/2;% Нормировка постоянной составляющей в спектре  
  
Нам осталось только вывести результаты. Функция subplot позволяет разбить окно на несколько областей для отображения графиков.  
  
%% Построение графиков  
subplot(2,1,1);% Выбор области окна для построения  
plot(T,Signal);% Построение только сигнала  
title('Сигнал');% Заголовок графика  
xlabel('Время (с)');% Подпись по оси х графика  
ylabel('Амплитуда (Ед)'); % Подпись по оси у графика  
subplot(2,1,2);% Выбор области окна для следующего построения  
plot(T,Signal+Noise);% Построение сигнала с шумом  
title('Сигнал+шум');% Заголовок графика   
xlabel('Время (с)');% Подпись по оси х графика  
ylabel('Амплитуда (Ед)');% Подпись по оси у 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 только сигнала  
title('Спектр сигнала');% Заголовок графика  
xlabel('Частота (Гц)');% Подпись по оси х графика  
ylabel('Амплитуда (Ед)');% Подпись по оси у графика  
subplot(2,1,2);% Выбор области окна для построения  
plot(F,FftSh(1:length(F)));% Построение спектра Фурье сигнала с шумом  
title('Спектр сигнал+шум');% Заголовок графика  
xlabel('Частота (Гц)');% Подпись по оси х графика  
ylabel('Амплитуда (Ед);% Подпись по оси у графика  
  
Результат выполнения кода будет выглядеть следующим образом:  
  
  
  
Несмотря на то, что полезного сигнала не видно на фоне шума, **спектральная характеристика позволяет определить его частоту и амплитуду**.  
  
Надеюсь данный текст был вам полезен.

### [Картинки по запросу: eugene khutoryansky all videos](https://www.google.ru/search?q=eugene+khutoryansky+all+videos&newwindow=1&tbm=isch&tbo=u&source=univ&sa=X&ved=2ahUKEwjqxrv2s-jeAhWFkSwKHdI6ARcQsAR6BAgFEAE)

[](https://www.google.ru/search?q=eugene+khutoryansky+all+videos&newwindow=1&tbm=isch&source=iu&ictx=1&fir=zc3TZ097GOxIPM:,kOxKOu0x_XnLnM,_&usg=AI4_-kR1FY7vPjYZCSyUgzdkNOnXmQi2QQ&sa=X&ved=2ahUKEwjqxrv2s-jeAhWFkSwKHdI6ARcQ9QEwEnoECAUQBA#imgrc=zc3TZ097GOxIPM:)

[](https://www.google.ru/search?q=eugene+khutoryansky+all+videos&newwindow=1&tbm=isch&source=iu&ictx=1&fir=hVapBGCjqfPkAM:,LIbhgCBxpzJLRM,_&usg=AI4_-kQFvipOvMRwhG4Xy6SqZf9TKEpzGQ&sa=X&ved=2ahUKEwjqxrv2s-jeAhWFkSwKHdI6ARcQ9QEwE3oECAUQBg#imgrc=hVapBGCjqfPkAM:)

[](https://www.google.ru/search?q=eugene+khutoryansky+all+videos&newwindow=1&tbm=isch&source=iu&ictx=1&fir=dUhHZ6Wh7RJ7nM:,PABB_dZjmE4y0M,_&usg=AI4_-kSmo4db39fKHRzlRHGDcvApbdsYwA&sa=X&ved=2ahUKEwjqxrv2s-jeAhWFkSwKHdI6ARcQ9QEwFHoECAUQCA#imgrc=dUhHZ6Wh7RJ7nM:)

[](https://www.google.ru/search?q=eugene+khutoryansky+all+videos&newwindow=1&tbm=isch&source=iu&ictx=1&fir=53n1o0SpVacC1M:,E3zFtqbUcR8AAM,_&usg=AI4_-kT-cmQIInVQ3Q2Ew4xerit5sdUCeA&sa=X&ved=2ahUKEwjqxrv2s-jeAhWFkSwKHdI6ARcQ9QEwFXoECAUQCg#imgrc=53n1o0SpVacC1M:)

[](https://www.google.ru/search?q=eugene+khutoryansky+all+videos&newwindow=1&tbm=isch&source=iu&ictx=1&fir=3cGuATWRhHnDKM:,zk9HTYXJrdCgGM,_&usg=AI4_-kTqAuaPQPbzcoyVF2foSzCR5-vV4g&sa=X&ved=2ahUKEwjqxrv2s-jeAhWFkSwKHdI6ARcQ9QEwFnoECAUQDA#imgrc=3cGuATWRhHnDKM:)