Определение нот

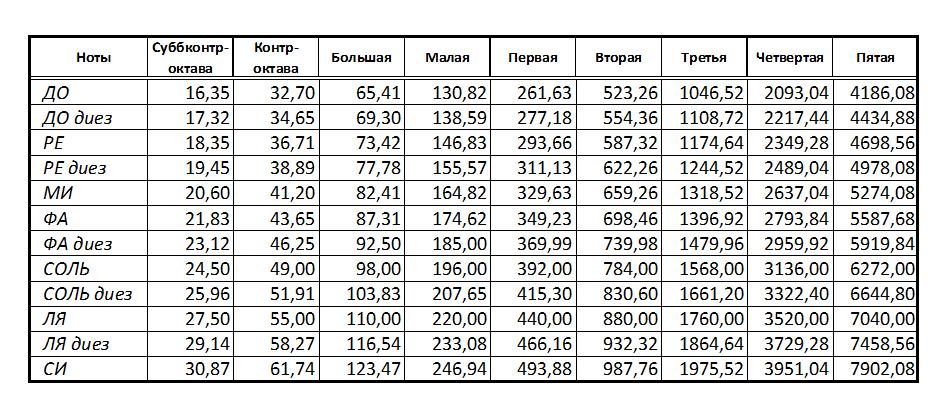
Начало программы

Очистка Workspace

clear all

close all

Определяю таблицу частот и нот



% Большая октава

Tab\_F\_1 = [65.41 69.3 73.42 77.78 82.41 87.31 92.5 98 103.83 110 116.54 123.47];

% Математически получаем частоты остальных октав

k = 2.^[0:6];

Tab\_F = k' .\* repmat(Tab\_F\_1, 7, 1);

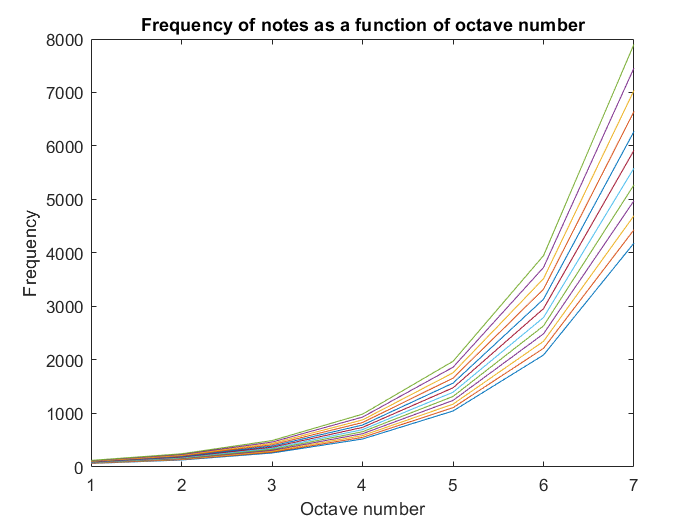
figure (1)

plot(Tab\_F)

xlabel("Octave number")

ylabel("Frequency")

title("Frequency of notes as a function of octave number")



Очистка от ненужного

clear Tab\_F\_1 k

Определяю матрицу сумм, в которую из спектра я буду складывать составляющие по частотам каждой ноты. КАждая ячейка соответствует мощности в спектре конкретной ноты конкретной октавы.

M = zeros(7, 12)

M = 7×12

0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0

Загрузка аудиофайла с 60 нотами подряд

path = 'D:\Desktop\Studie\Diploma\Audio data\wav\60 нот.wav';

[y, Fs] = audioread(path);

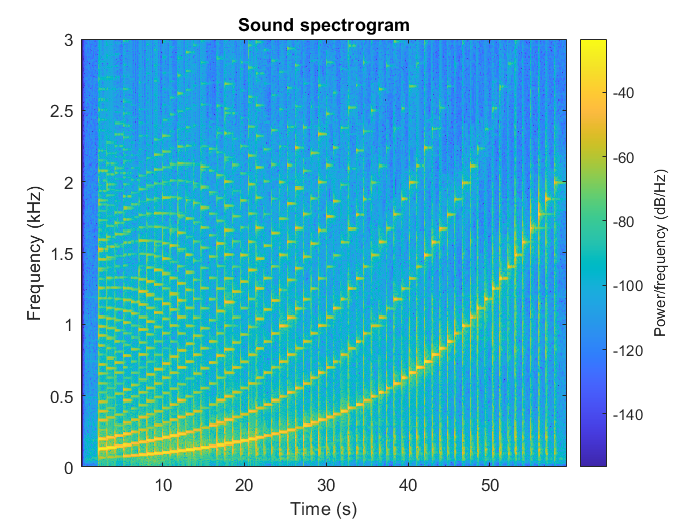
z = y(:,1);

figure (1)

spectrogram(z,6000, 2000,6000,Fs,'yaxis')

title('Sound spectrogram')

ylim([0 3])



Очистка от ненужного

clear y path

Определяю максимальную пачку отчетов спектр которого я могу брать для анализа, что бы заметить самую быструю ноту.

bpm = 120; % 120 таков в минуту

bps = bpm/60; % или 2 доли в секунду

t\_d = 1/bps; % что есть 0.5сек на долю

Note\_min = 1/16; % Условно минимальная продолжительность ноты

% 1/16 доли

T\_min = Note\_min \* t\_d

T\_min = 0.0313

Fs = 44100; % При частоте дискретизации 44 100Гц получаем

Samples\_num = T\_min \* Fs

Samples\_num = 1.3781e+03

Samples\_num = 1024 % С округлением в низ

Samples\_num = 1024

df = Fs/Samples\_num % В таком случае шаг частоты спектра 43Гц

df = 43.0664

Очистка от ненужного

clear bpm bps t\_d Note\_min T\_min df

Определение векторов "Vi" для извлечения амплитуд. АЛЬТЕРНАТИВА: находим массив "Indexes" индексов которые соответствуют нотам

Для начала выделю участок сигнала и проинтерполирую его спектр

Fs = 44100;

i = 400;

S = z(i\*Samples\_num: (i+1)\*Samples\_num -1);

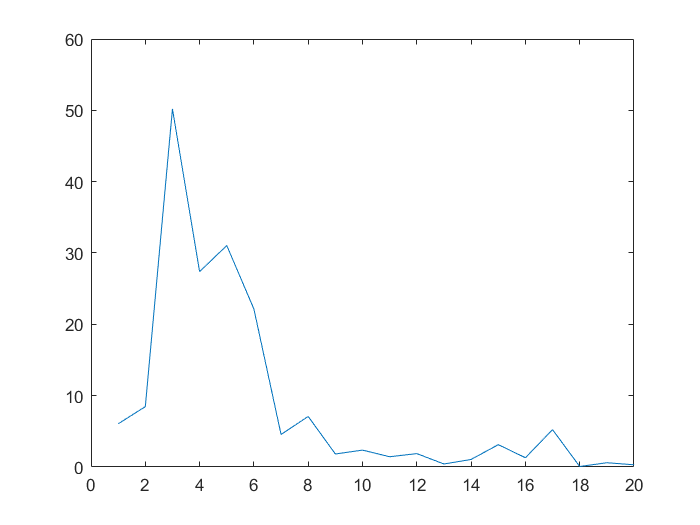
L = length(S);

df = Fs/L;

figure(1)

plot(abs(fft(S)))

xlim([0 20])



k\_interp = Fs/length(S) - 1;

zeros\_num = floor( k\_interp\*Samples\_num );

Zer = zeros(zeros\_num, 1);

S2 = vertcat (Zer, S);

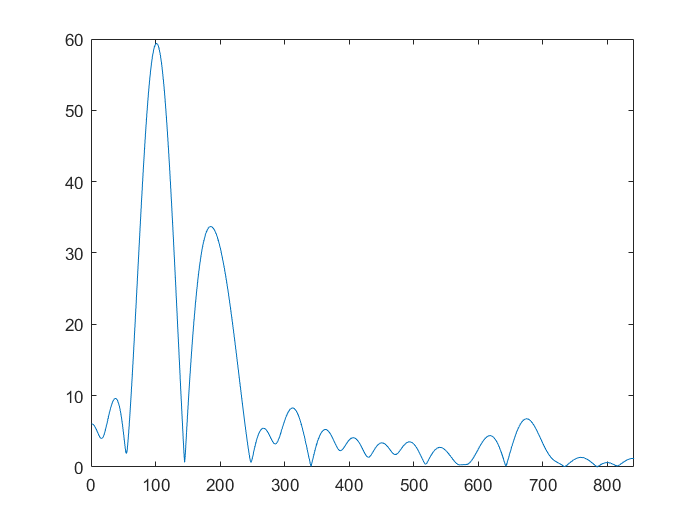
L2 = length(S2);

df2 = Fs/L2;

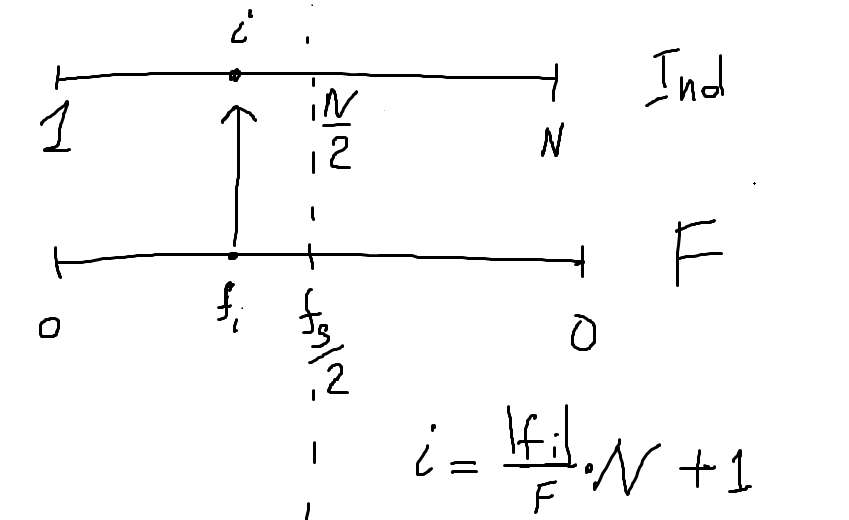
figure(2)

plot(abs(fft(S2)))

xlim([0 20] \* k\_interp)



Теперь, имея удобный спектр, найду индексы, которые соответствуют частотам нот



FlatTab\_F = reshape(Tab\_F.',1,[]) % FLAT!!!

FlatTab\_F = 1×84

103 ×

0.0654 0.0693 0.0734 0.0778 0.0824 0.0873 0.0925 ⋯

Indexes = floor(L2 \* FlatTab\_F / Fs + 1);

Spec = abs(fft(S2));

Spec = Spec(1:L2/2);

f = 0:df2:Fs/2-df2

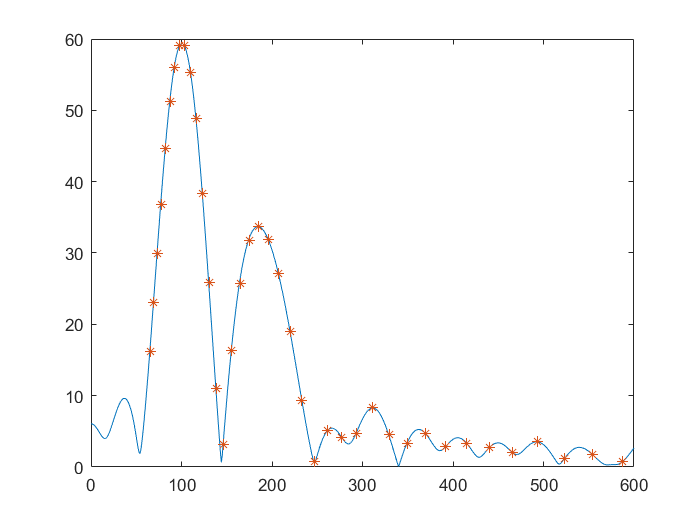
f = 1×22050

0 1 2 3 4 5 6 7 8 9 10 11 12 ⋯

figure (3)

plot(f, Spec, FlatTab\_F, Spec(Indexes), '\*')

xlim([0 600])



Проба записи матрицы М для 1го слайса

ИЗМЕНЯЙ "i" И НАБЛЮДАЙ 3Д ИЗОБРАЖЕНИЕ МАТРИЦЫ

i = 1...2500

i = 1350;

S = z(i\*Samples\_num: (i+1)\*Samples\_num - 1);

figure(1)

subplot(2,1,1)

plot(abs(fft(S)))

xlim([0 20])

k\_interp = Fs/length(S) - 1;

zeros\_num = floor( k\_interp\*Samples\_num );

Zer = zeros(zeros\_num, 1);

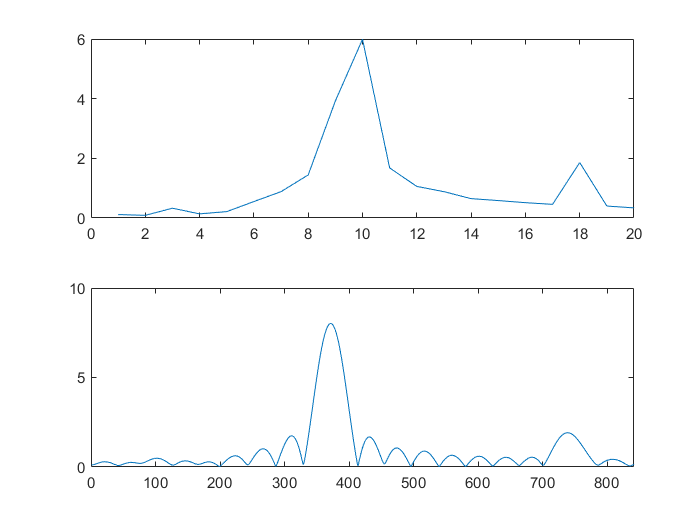
S2 = vertcat (Zer, S);

figure(1)

subplot(2,1,2)

plot(abs(fft(S2)))

xlim([0 20] \* k\_interp)



Indexes = floor(L2 \* Tab\_F / Fs + 1); % NOT FLAT!!!

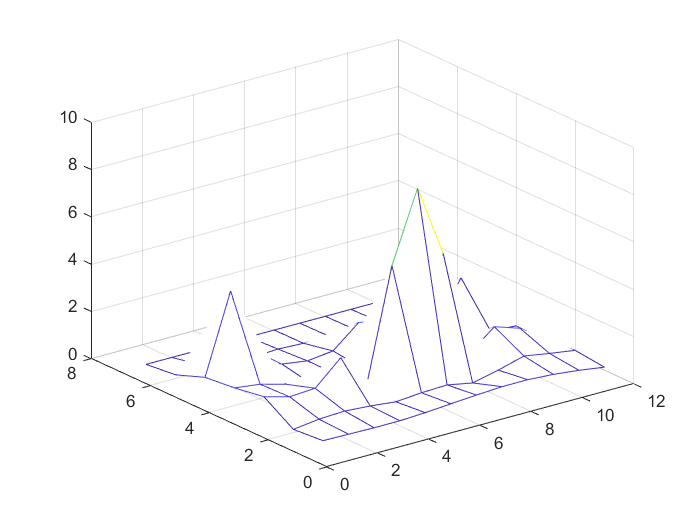
Spec = abs(fft(S2));

Spec = Spec(1:L2/2);

M = Spec(Indexes);

figure(2)

mesh(M);



Определение 1 ноты определением максимума матрицы М

i = 1350;

S = z(i\*Samples\_num: (i+1)\*Samples\_num - 1);

k\_interp = Fs/length(S) - 1;

zeros\_num = floor( k\_interp\*Samples\_num );

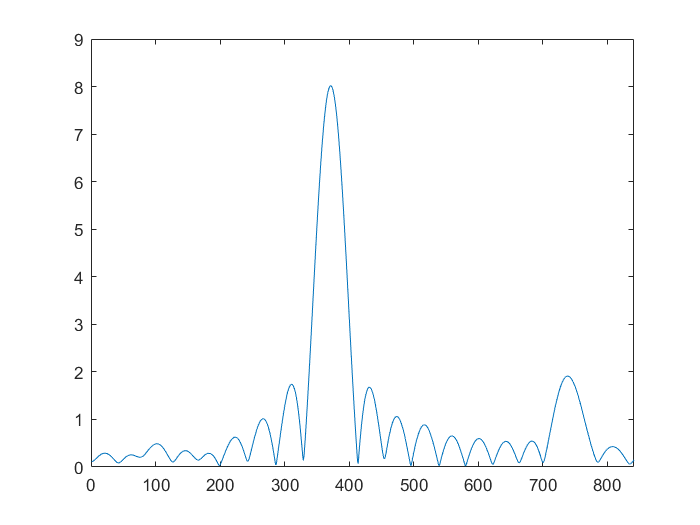
Zer = zeros(zeros\_num, 1);

S2 = vertcat (Zer, S);

figure(4)

plot(abs(fft(S2)))

xlim([0 20] \* k\_interp)



Indexes = floor(L2 \* Tab\_F / Fs + 1); % NOT FLAT!!!

Spec = abs(fft(S2));

Spec = Spec(1:L2/2);

M = Spec(Indexes);

Октава и нота. (линейный индекс берет конкретную ноту и идет по октавам, далее прыгает на другую ноту. По этому модуль по 7, т.к. прыжок совершает изменение индекса на длинну октав - 7)

[Val, Ind] = max(M,[],[1 2], 'linear') % Нахожу индекс максимума

Val = 8.0162

Ind = 45

octave\_num = mod(Ind, 7)

octave\_num = 3

note\_num = (Ind - octave\_num)/7 + 1

note\_num = 7

НЕ ГОТОВО!!!

Пройдусь по всему сигналу, вырезая из него пачки по 1024 отчета. Из этих пачек я буду находить амплитуды каждой ноты, определять максимальную и говорить, что она является основным тоном. (Учитываем что играется только одна нота)

Делаю обезок длинной Samples\_num. Использую перекрытие 1/2 по обрезкам тоже длинной Samples\_num. Передаю спектр обрезка в функцию которая вынимает гармоники и находит максимум. Эта функция возвращает мне частоту с максимальной амплитудой. Я добавляю ее в конец массива. В итоге получаю массив который показывает в какое время какую ноту играли.

%{

cur\_note = [0 0];

cuts = floor( length(z) / Samples\_num ) - 1; % Количество вырезок из сигнала

for i = 1:cuts

if (i > 1)

L = Samples\_num/2;

S = z(i\*Samples\_num-L : (i+1)\*Samples\_num-L - 1);

F = FillNoteMatrix(S);

cur\_note(end+1) = F;

end

S = z(i\*Samples\_num : (i+1)\*Samples\_num - 1);

F = FillNoteMatrix(S);

cur\_note(end+1) = F;

end

%}

Определяю функцию которая будет избирать из спектра амплитуды, которые находяться по частотам нот и складывать их в M

Интерполяцией переведу спектр в такой, что бы шаг частоты был равен 1

%{

function F = FillNoteMatrix (S)

k\_interp = Fs/length(S) - 1;

zeros\_num = floor( k\_interp\*Samples\_num);

Zer = zeros(zeros\_num, 1);

S2 = vertcat (Zer, S);

Spec = abs(fft(S2)); % Получаем спектр с шагом примерно 1Гц

N = length(Spec);

N2 = N/2;

Spec = Spec(1:N2);

for i = 1:length(M)

index = floor( N \* Tab\_F(i) / Fs ); % Перевод частоты в индекс

M(i) = Spec(index);

end

F; index = max(M);

end

%}