Практичні заняття 1

План:

1. Обчислення спектрограми за допомогою функції Matlab
2. Обчислення спектрограми за допомогою самостійно розробленої функції

1. Підготовка до обчислень спектрограми

Зчитуємо звуковий файл із диска, будуємо графік сигналу та прослуховуємо сигнал:

```
[x, fs] = audioread('speech.wav');
t = 1/fs:1/fs:length(x)/fs;
figure; plot(t,x);
xlabel('Time (sec)'); ylabel('Level');
soundsc(x,fs)
```

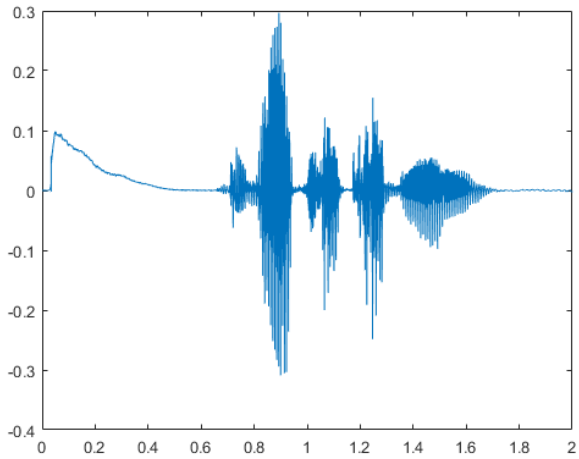


Рис. 1. Форма сигналу «Двадцать один»

Підготовка до обчислень спектрограми:

```
Tseg = 0.1;
Nseg = round(Tseg*fs);
window = hamming(Nseg);
nfft = Nseg;
noverlap = round(Nseg/2);
```

2. Обчислення спектрограми за допомогою функції Matlab

Синтаксис:

```
spectrogram(x,window,noverlap,nfft,fs)
```

або

```
S = spectrogram(x,window,noverlap,nfft,fs)
```

В першому випадку побудова графіка спектрограми (рис. 2) виконується автоматично:

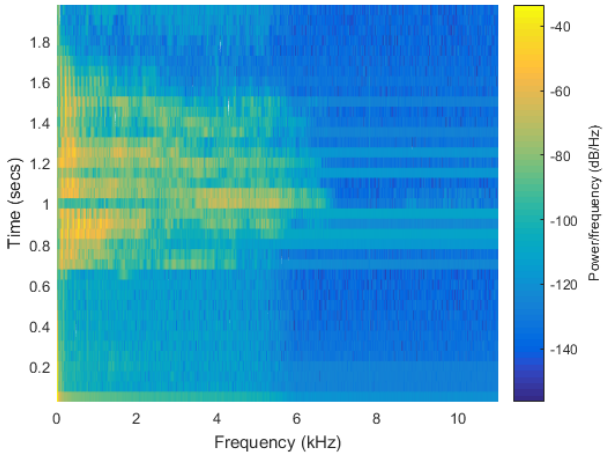


Рис. 2

Звертаємо увагу на графік рис. 2: частоти розташовані уздовж осі x, час - уздовж осі y. Це не завжди є зручним. Наприклад, в ряді програм, таких як Sound Forge, Audacity та інших, вісь часу розташована горизонтально, а вісь частот - вертикально.

Щоб отримати такий графік (рис. 3), треба додати в аргумент функції опцію '**yaxis**':

```
spectrogram(x, window, noverlap, nfft, fs, 'yaxis');
```

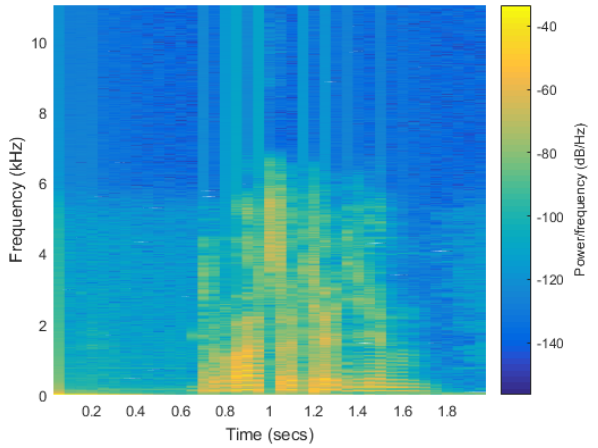


Рис. 3

Побудова 3D-графіка (рис. 4):

```
spectrogram(x, window, noverlap, nfft, fs);

view(-77,72)
shading interp
colorbar off
```

Побудова із використанням порогу –SNR, нижче якого значення спектрограми замінено темним кольором (рис. 5):

```
spectrogram(x, window, noverlap, nfft, fs, 'MinThreshold', -SNR, 'yaxis');
```

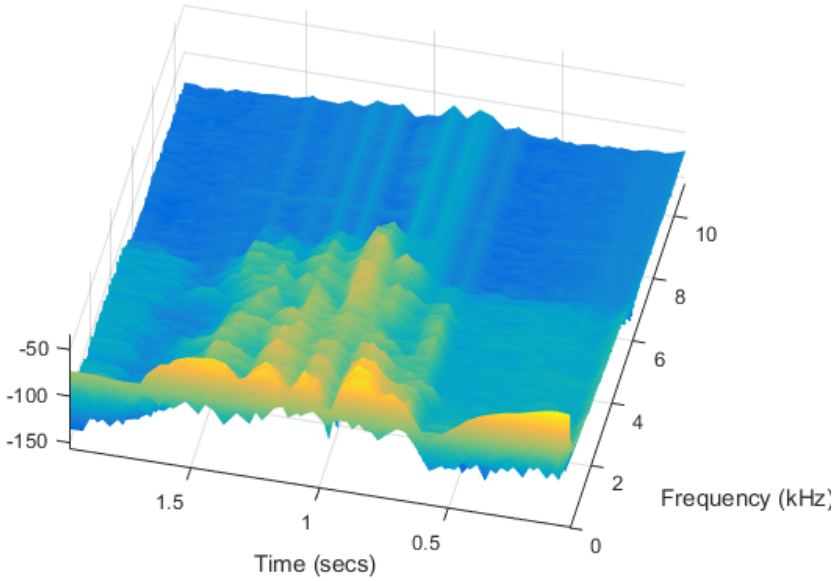
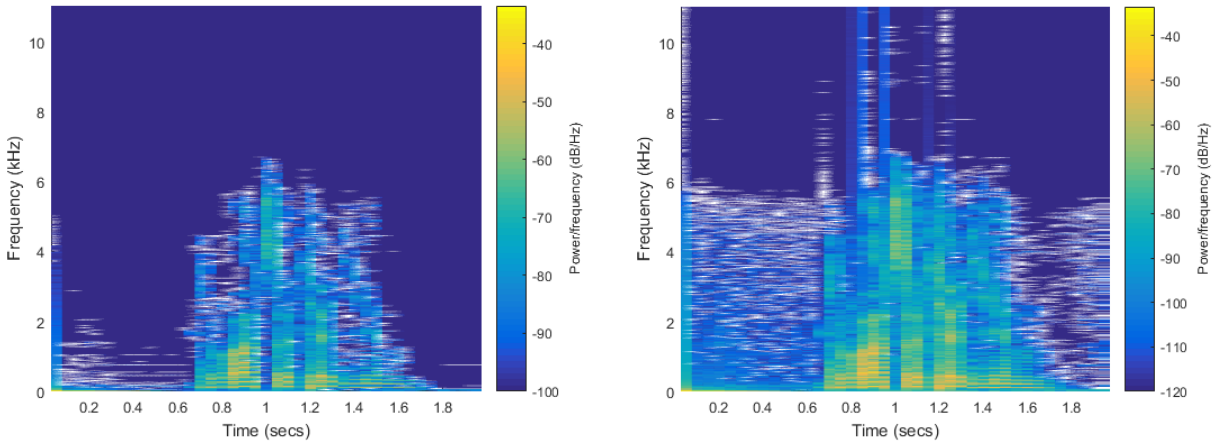


Рис. 4



а б

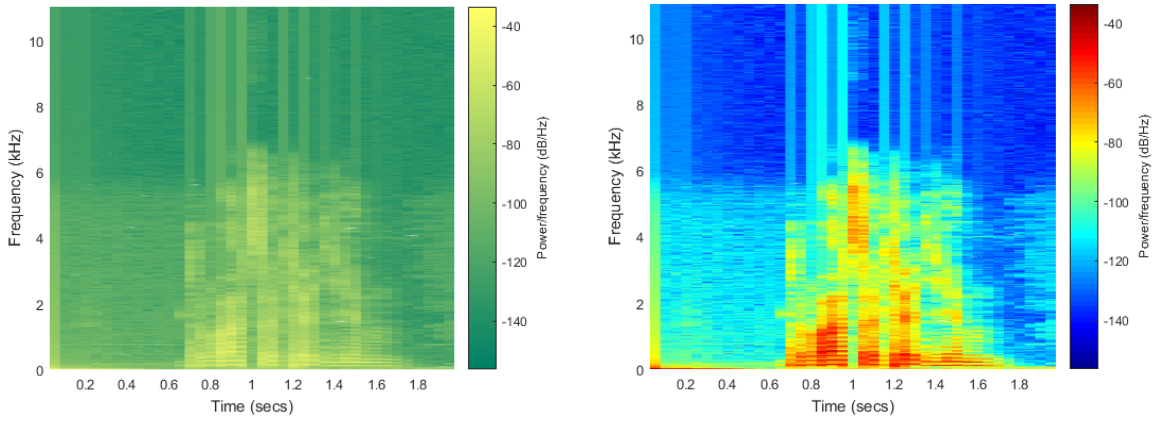
Рис. 5. SNR = 100 (а) та SNR=120 (б)

Управління кольоровою палітрою (рис. 6):

colormap name

де значеннями **name** можуть бути:

parula jet hsv hot cool spring summer
autumn winter gray bone copper pink
lines colorcube prism flag white



а б

Рис. 6. **summer** (а) та **jet** (б)

2. Обчислення спектрограми за допомогою самостійно розробленої функції

Аргументом на користь такої ідеї є прагнення спростити синтаксис. Результат обчислень представлено на рис. 7.

```
function S = myspecgram(filename,dt,df)
% === вычисление спектрограммы ===
%
% ВХОДНЫЕ ДАННЫЕ:
% filename - имя звукового файла
% dt - разрешение по времени
% df - разрешение по частоте
% ВЫХОДНЫЕ ДАННЫЕ:
% S - спектрограмма
%
% === чтение звукового файла ===
[x, fs] = audioread(filename);
%
% === подготовка к делению на фреймы ===
nfft = round(fs/df);        % длина фрейма = параметр БПФ = длина окна
shft = round(dt*fs);              % сдвиг фреймов, выраженный в кол-ве выборок
nfrm = floor((length(x)-nfft)/shft); % количество фреймов
%
% === деление x(n) на фреймы ===
xfrm = zeros(nfft,nfrm);  % выделение области
for k = 1:nfrm
   nach = 1+(k-1)*shft; kon = nach+nfft-1;   % номера границ фреймов
   xfrm(:,k) = x(nach:kon);                % матрица фреймов
end
%
% === взвешивание фреймов окном Хэмминга ===
w = hamming(nfft);
winmatr = repmat(w,1,nfrm);
xwin = winmatr.*xfrm;
%
% === БПФ от каждого фрейма ===
sp = fft(xwin);
S1 = sp.*conj(sp);
%
% === отбрасываем отрицат частоты, берем модуль и нормир. по макс. ===
S1 = S1(1:round(nfft/2),:);
Sabs = abs(S1);
Samax = max(max(Sabs));
S2 = Sabs/Samax;
```

```
S = im2uint16(S2);
%
% === вывод графиков ===
tx = 0:1/fs:(length(x)-1)/fs;
figure; subplot(2,1,1); plot(tx,x);
xlabel('Время, с'); ylabel('Уровень');

t = 0:dt:(size(S,2)-1)*dt;
f = 0:df:(size(S,1)-1)*df;
subplot(2,1,2); image(t,f,S);
set(gca,'YDir','normal');
colormap(jet);
xlabel('Время, с'); ylabel('Частота, Гц');
```

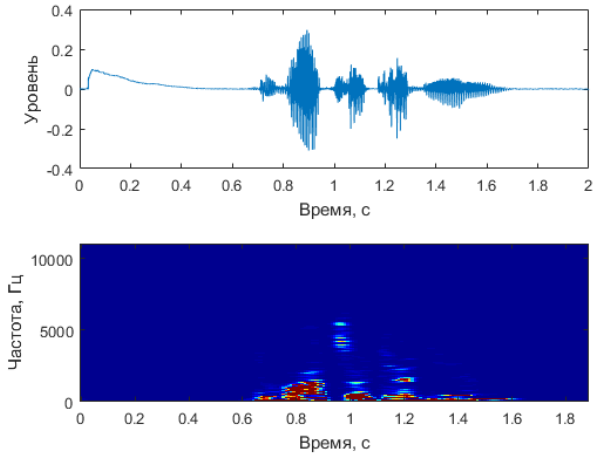


Рис. 7

Одним із недоліків отриманого результату є недостатній динамічний діапазон спектрограми. Щоб покращити справу, можна спробувати використати процедуру логарифмування:

```
 % === вывод спектрограммы 2 способ ===
S3 = (20 * log10(S2) +70)/70;
Slog = im2uint16(S3);
figure; image(t,f,Slog);
set(gca,'YDir','normal');
colormap(jet);
xlabel('Время, с'); ylabel('Частота, Гц');
```

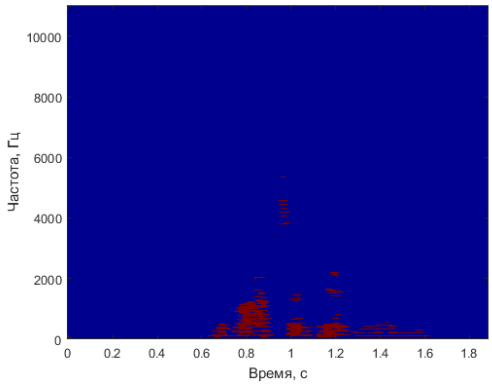


Рис. 8

Відповідний результат показано на рис. 8. Як бачимо, покращити якість картинки не вдалося. Тому домашнім завданням буде спроба домогтися кращого результату.