

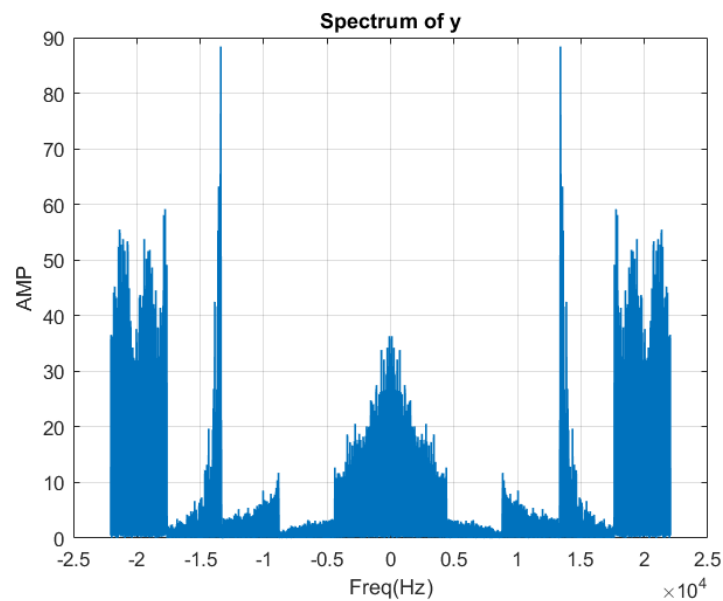
۱. فایل صوتی را لود کرده و طیف فرکانسی آن را رسم کنید.

مطابق زیر ابتدا فایل صوتی و فیلترها را لود کردم و بعد صدا را پلی کردم. سپس به کمک فانکشن تعریف شده FFT طیف فرکانسی صوت رمزنگاری شده را رسم کردم:

```
[y , fs] = audioread('C:\Users\pc
world\Desktop\S&S_FinalProject_14022\testmusic.wav');
load('C:\Users\pc world\Desktop\S&S_FinalProject_14022\filters.mat')
player = audioplayer(y , fs);
%play the sound
play(player)
```

## spectrum of sound

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% use FFT function defined in example
FFT(y,fs);title('Spectrum of y');
```



۲. هر کدام از باندهای فرکانسی را به دست آورید و طیف فرکانسی آنها و بازیابی شده اش را در یک figure رسم کنید (یعنی جمعاً ۵ تا).

برای این قسمت ابتدا معکوس ماتریس A را حساب کردم سپس باندهای ورودی را جدا کردم و بعد با ضرب معکوس A در سیگنال رمزنگاری شده باندهای سیگنال اصلی را بدست آوردم اما حالا نیاز است که با شیفต์ دادن باندها آن هارا تنظیم کنیم. اینجا هر کدام از باندها را در تابع cos با فرکانس مناسب (که میشود اختلاف پوزیشن ها ضرب در فرکانس مرکزی) که در ابتدای کد تعریف کرده بودم ضرب میکنم تا هر باند در جای خود باشد و بعد هر 10 باند را در یک figure رسم کردم مطابق زیر:

```
ts = 1/ fs;
t = 0:ts:(length(y)-1)*ts ;
band1_f= fs/10;
band2_f= 2*(fs/10);
band3_f= 3*(fs/10);
band4_f= 4*(fs/10);
c1 = cos (2*pi*band1_f*t) ;
c2 = cos (2*pi*band2_f*t) ;
c3 = cos (2*pi*band3_f*t) ;
```

### computing the inverse matrix

```
%%%%%%%%%%
Ainv = zeros(size(A));
Ainv = inv(A);
disp(Ainv)
```

```
0      0      0      20      0
0      0      0      0      1
1      0      0      0      0
0      0      1      0      0
0      1      0      0      0
```

### decomposing the input into 5 bands

```
%%%%%%%%%%
% Decompose the input into 5 bands using the provided filters
y1 = filter(Band1, y);
y2 = filter(Band2, y);
y3 = filter(Band3, y);
y4 = filter(Band4, y);
y5 = filter(Band5,y);
Y = [y1, y2, y3, y4, y5];
```

### Band1

```
x1 = zeros(length(y),1);
x1 = filter(Band1,x1);
```

## Band2

```
x2 = zeros(length(y) , 1);
x2 = filter(Band2,x2);
```

## Band3

```
x3 = zeros(length(y),1);
x3 = filter(Band3,x3);
```

## Band4

```
x4 = zeros(length(y),1);
x4 = filter(Band4,x4);
```

## Band5

```
x5 = zeros(length(y),1);
x5 = filter(Band5,x5);
```

## Decryption

```
X = [x1, x2, x3, x4, x5];
X = Y * Ainv';
x1 = X(:,1);
x2 = X(:,2);
x3 = X(:,3);
x4 = X(:,4);
x5 = X(:,5);
x1 = x1 .*c3';
x1 = filter(Band1,2*x1);
x2 = x2 .*c3';
x2 = filter(Band2,2*x2);
x3 = x3.*c2';
x3 = filter(Band3,2*x3);
x4 = x4 .*c1';
x4 = filter(Band4,2*x4);
x5 = x5 .*c3';
x5 = filter(Band5,2*x5);
```

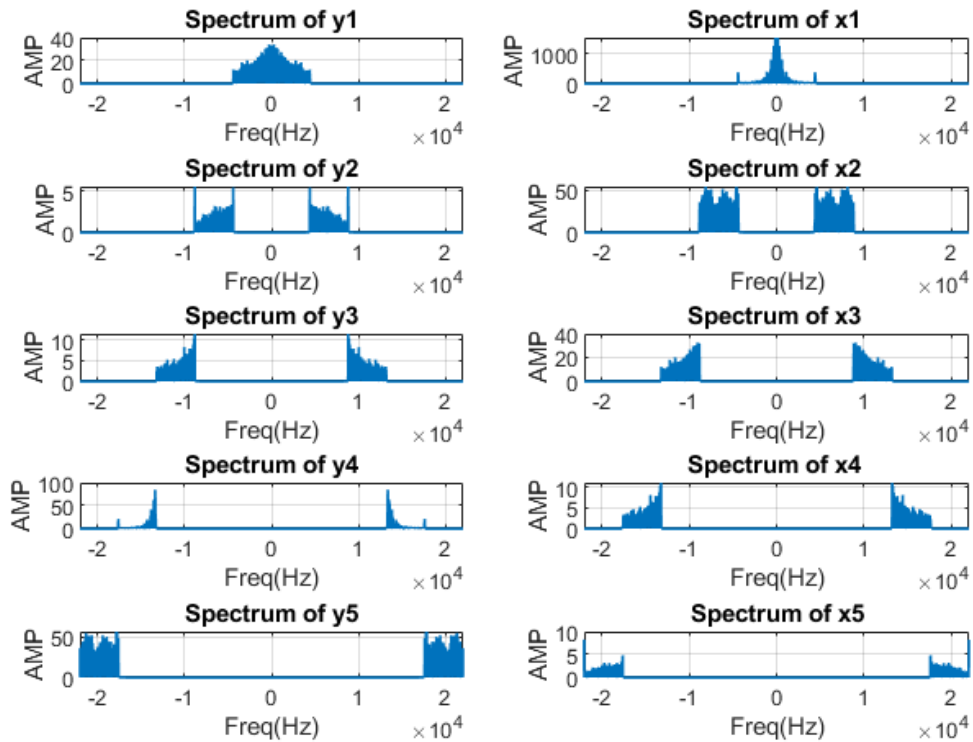
## frequency spectrum of xi and yi

```
% Plot the frequency spectrum of xi and yi in the same figure
figure;
subplot(5,2,1); FFT(y1, fs); title('Spectrum of y1');
```

```

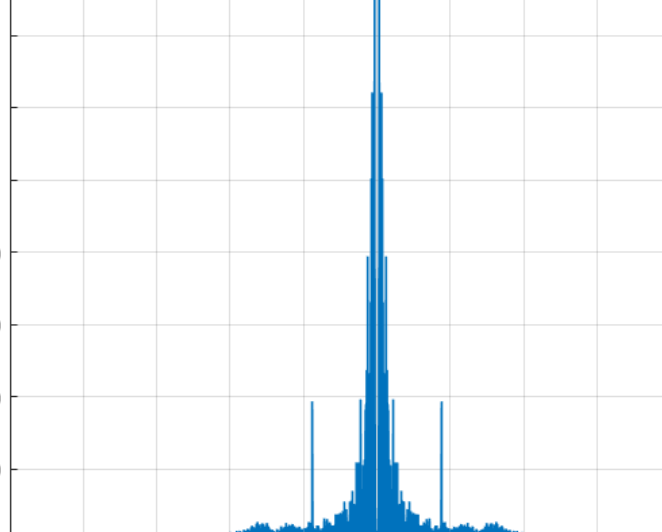
subplot(5,2,2); FFT(x1, fs); title('Spectrum of x1 ');
subplot(5,2,3); FFT(y2, fs); title('Spectrum of y2');
subplot(5,2,4); FFT(x2, fs); title('Spectrum of x2 ');
subplot(5,2,5); FFT(y3, fs); title('Spectrum of y3');
subplot(5,2,6); FFT(x3, fs); title('Spectrum of x3 ');
subplot(5,2,7); FFT(y4, fs); title('Spectrum of y4');
subplot(5,2,8); FFT(x4, fs); title('Spectrum of x4 ');
subplot(5,2,9); FFT(y5, fs); title('Spectrum of y5');
subplot(5,2,10); FFT(x5, fs); title('Spectrum of x5');

```



حال کافیت که باندهای بازیابی شده را جمع کنیم و آن را ذخیره کنیم و به کمک تابع FFT طیف فرکانسی آن را رسم کنیم:

## Spectrum of real sound



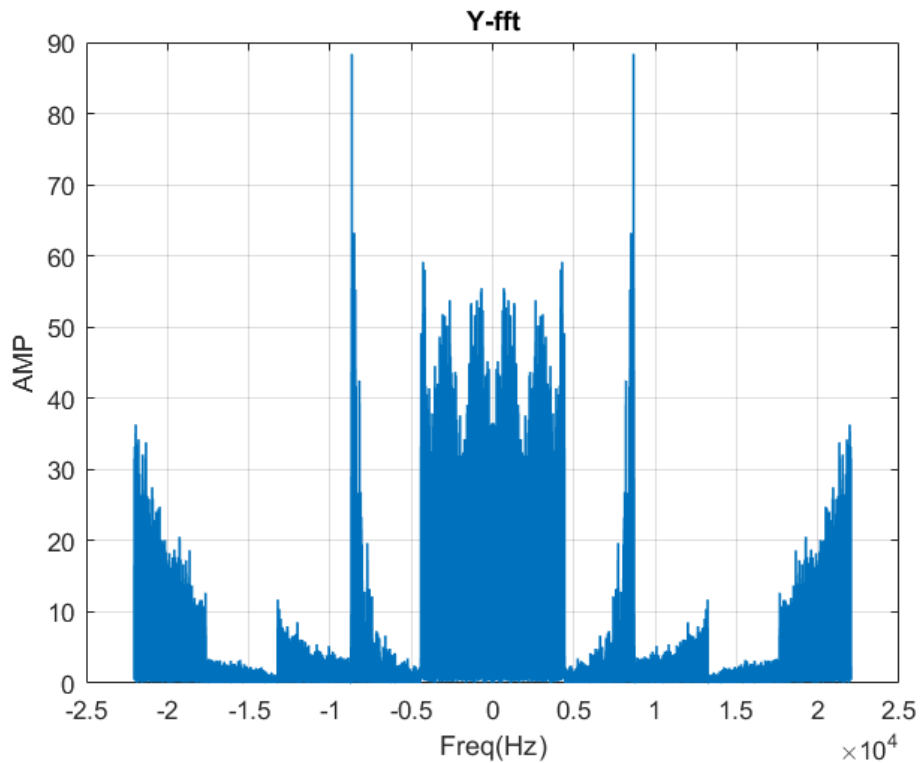
The plot, titled "Spectrum of x", shows the amplitude spectrum of a signal. The x-axis is labeled "Freq(Hz)" and ranges from -2.5 to 2.5, with a multiplier of  $\times 10^4$  at the bottom right. The y-axis is labeled "AMP" and ranges from 0 to 1600. The spectrum is centered at 0 Hz, with a primary peak reaching approximately 1500 in amplitude. There are two prominent side peaks at approximately  $\pm 0.4 \times 10^4$  Hz, each reaching an amplitude of about 400. The plot includes a light gray grid.

۴. یک روش حل دیگر می‌توانست این باشد که از سیگنال `fft` (یک دستور سریع است برای محاسبه تبدیل فوریه `fft` است) بگیریم و بعد المان‌های آن را جابجا کنیم و از طیف به دست آمده `inverse fft` بگیریم. آیا اینگونه هم امکانپذیر است؟ (اگر هست از این روش نیز مسئله را حل کنید و اگر نیست دلیل خود را ذکر کنید).

بله امکان پذیر است. در این قسمت ابتدا از سیگنال رمزنگاری شده ورودی `fft` گرفتیم و نمایش دادم مطابق زیر:

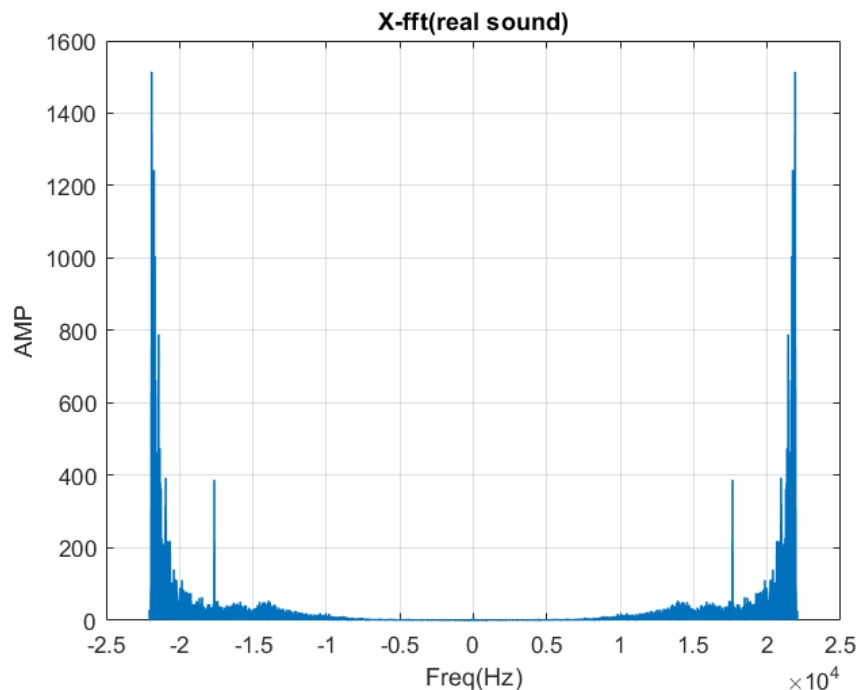
## fft method

```
Y_fft=fft(y);
N = numel(y);
freq = (-N/2:N/2-1)/N* fs;
% figure
plot(freq,(abs(Y_fft)),'LineWidth',1);
grid on
ylabel ('AMP')
xlabel('Freq(Hz)')
title('Y-fft');
```



از صوت بازیابی شده از قسمت قبل هم fft گرفتیم که پاسخ را چک کنیم مطابق زیر:

```
X_fft1=fft(x);
N = numel(x);
freq = (-N/2:N/2-1)/N* fs;
% figure
plot(freq,(abs(X_fft1)),'LineWidth',1);
grid on
ylabel ('AMP')
xlabel('Freq(Hz)')
title('X-fft(real sound)');
```

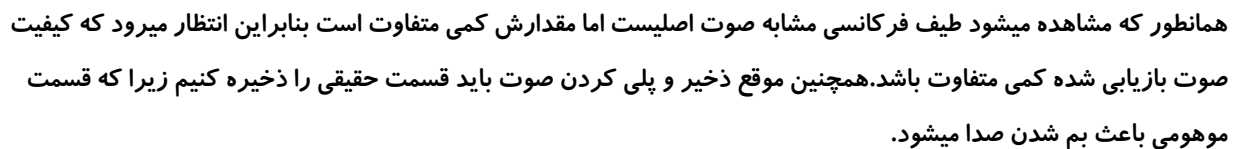


حالا امان هارا مطابق ماتریس معکوس جابجا کردم و چون سیگنال زوج است 10 قسمتی میشود مطابق زیر:

```
% changing the elements
X_fft(1:123480,1)=20*(Y_fft(370441:493920,1));
X_fft(123481:246960,1)=(Y_fft(493921:617400,1));
X_fft(246961:370440,1)=(Y_fft(1:123480,1));
X_fft(370441:493920,1)=(Y_fft(246961:370440,1));
X_fft(493921:617400,1)=(Y_fft(123481:246960,1));

X_fft(617401:740880,1)=(Y_fft(987841:1111320,1));
X_fft(740881:864360,1)=(Y_fft(864361:987840,1));
X_fft(864361:987840,1)=(Y_fft(1111321:1234800,1));
X_fft(987841:1111320,1)=(Y_fft(617401:740880,1));
```

```
N = numel(x);  
freq = (-N/2:N/2-1)/N* fs;  
% figure  
plot(freq,(abs(X_fft)),'LineWidth',1);  
grid on;  
ylabel ('AMP');  
xlabel('Freq(Hz)');  
title('X-fft(decoded sound)');
```



```
x_p=ifft(X_fft);
player = audioplayer(abs(x_p) , fs);
%play(player)
```



```
%%%%save vector x as a sound in the curent folder  
audiowrite('C:\Users\pc world\Desktop\S&S_FinalProject_14022\output_fft.wav' ,  
abs(x_p ), fs);
```

۷. (امتیازی) یک فایل صوتی با استفاده از متلب از صدای خود ضبط کرده و به همین روش آن را رمز و سپس رمز گشایی کنید.

در این قسمت ابتدا صدای خودم را به کمک متلب ضبط کردم و در پوشه پروژه ذخیره کردم. کد این قسمت جدا زده شده که مجدد ران نشود!

```
fs = 44100;
nBits = 16;
nChannels = 1;
recordTime = 28;
recObj = audiorecorder(fs, nBits, nChannels);
disp('Start speaking.')
Start speaking.
recordblocking(recObj, recordTime);
disp('End of recording.');
```

End of recording.

```
audioData = getaudiodata(recObj);
audiowrite('C:\Users\pc world\Desktop\S&S_FinalProject_14022\recorded.wav' ,
audioData , fs);
disp(['Audio saved to ' fileName]);
```

Audio saved to C:\Program Files\Polyspace\R2021a\bin\recordedAudio.wav

حال در فایلی جدید ابتدا فایل ضبط شده و فیلتر هارا لود کردم سپس ماتریس جدیدی برای رمزنگاری تعریف کردم. سپس باندهای صوت ورودی را جدا کردم و با ضرب ماتریس B در صوت ورودی ، صوت رمزنگاری شده را بدست آوردم وبعد باندهای آن را جدا کردم و بعد با شیفت مناسب به کمک ضرب در COS با فرکانس مناسب (مانند قسمت قبل)باند هارا بازیابی کردم. در نهایت کافی بود که باندهارا جمع کنم تا سیگنال رمزنگاری شده ایجاد شود و بعد آن را در پوشه پروژه ذخیره کردم. در این بین طیف فرکانسی هارا نیز رسم کردم مطابق زیر:

## Incrption

```
%loading the recorded audio
[y_p, fs_p] = audioread('C:\Users\pc
world\Desktop\S&S_FinalProject_14022\recorded.wav');
load('C:\Users\pc world\Desktop\S&S_FinalProject_14022\filters.mat')
player = audioplayer(y_p, fs_p);
% Play the sound
%play(player)
B = [0 0 0 10 0;
      0 0 10 0 0;
      0 1 0 0 0;
      0.05 0 0 0 0;
      0 0 0 0 1];
ts_p = 1 / fs_p;
t_p = 0:ts_p:(length(y_p) - 1) * ts_p;

band1_fp = fs_p / 10;
band2_fp = 2 * (fs_p / 10);
```

```

band3_fp = 3 * (fs_p / 10);
band4_fp = 4 * (fs_p / 10);

c1 = cos(2 * pi * band1_fp * t_p)';
c2 = cos(2 * pi * band2_fp * t_p)';
c3 = cos(2 * pi * band3_fp * t_p)';
c4 = cos(2 * pi * band4_fp * t_p)';

yp1 = filter(Band1, y_p);
yp2 = filter(Band2, y_p);
yp3 = filter(Band3, y_p);
yp4 = filter(Band4, y_p);
yp5 = filter(Band5, y_p);

Y_P = [yp1, yp2, yp3, yp4, yp5];

xp1 = zeros(length(y_p), 1);
xp2 = zeros(length(y_p), 1);
xp3 = zeros(length(y_p), 1);
xp4 = zeros(length(y_p), 1);
xp5 = zeros(length(y_p), 1);

xp1 = filter(Band1, xp1);
xp2 = filter(Band2, xp2);
xp3 = filter(Band3, xp3);
xp4 = filter(Band4, xp4);
xp5 = filter(Band5, xp5);

X_P = [xp1, xp2, xp3, xp4, xp5];
X_P = Y_P * B';

xp1 = X_P(:,1);
xp2 = X_P(:,2);
xp3 = X_P(:,3);
xp4 = X_P(:,4);
xp5 = X_P(:,5);

xp1 = xp1 .* c3;
xp1 = filter(Band1, 2*xp1);
xp2 = xp2 .* c1;
xp2 = filter(Band2, 2*xp2);

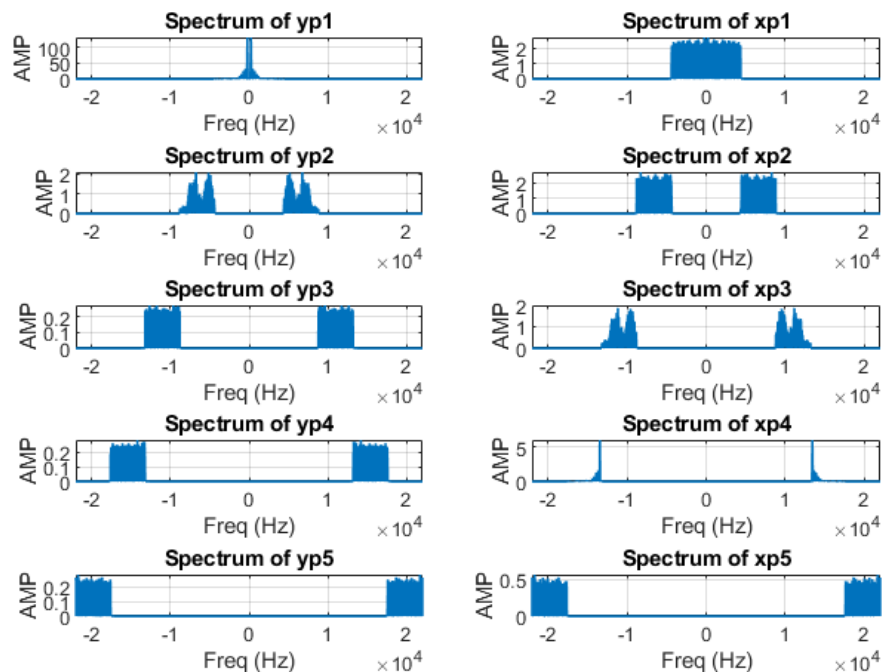
```

```

xp3 = xp3 .*c1;
xp3 = filter(Band3,2*xp3);
xp4 = xp4 .*c3;
xp4 = filter(Band4,2*xp4);
xp5 = filter(Band5,2*xp5);

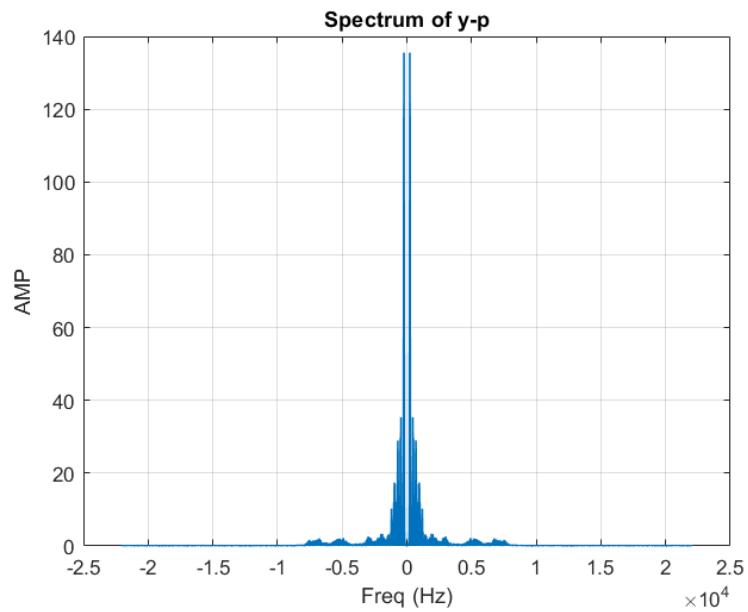
xp = xp1 + xp2 + xp3 + xp4 + xp5;
player = audioplayer(xp, fs_p);
% play(player)
audiowrite('C:\Users\pc
world\Desktop\S&S_FinalProject_14022\recorded_encrypted.wav', xp, fs_p);
figure;
subplot(5, 2, 1); FFT(yp1, fs_p); title('Spectrum of yp1');
subplot(5, 2, 2); FFT(xp1, fs_p); title('Spectrum of xp1');
subplot(5, 2, 3); FFT(yp2, fs_p); title('Spectrum of yp2');
subplot(5, 2, 4); FFT(xp2, fs_p); title('Spectrum of xp2');
subplot(5, 2, 5); FFT(yp3, fs_p); title('Spectrum of yp3');
subplot(5, 2, 6); FFT(xp3, fs_p); title('Spectrum of xp3');
subplot(5, 2, 7); FFT(yp4, fs_p); title('Spectrum of yp4');
subplot(5, 2, 8); FFT(xp4, fs_p); title('Spectrum of xp4');
subplot(5, 2, 9); FFT(yp5, fs_p); title('Spectrum of yp5');
subplot(5, 2, 10); FFT(xp5, fs_p); title('Spectrum of xp5');

```



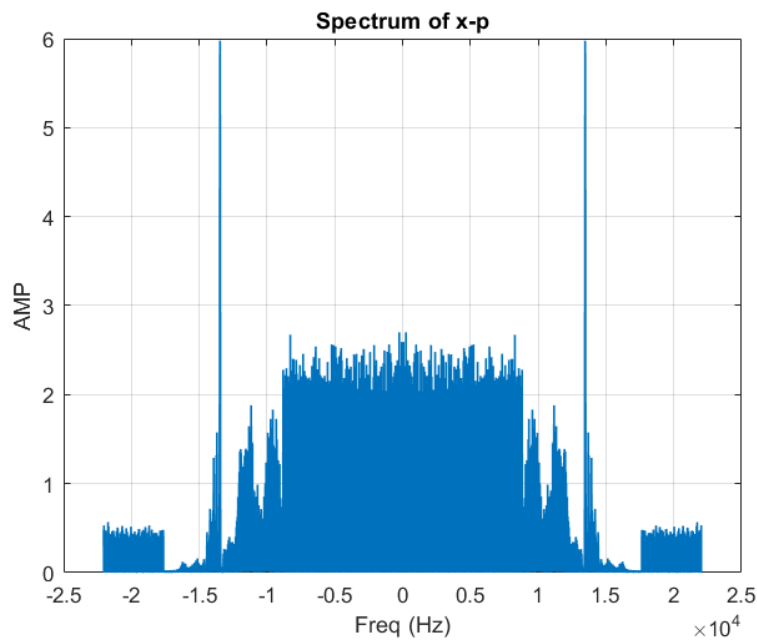
Spectrum of the real audio

```
figure;
FFT(y_p, fs_p); title('Spectrum of y-p');
```



## Spectrum of the encrypted audio

```
figure;
FFT(xp, fs_p); title('Spectrum of x-p');
```



حالا که سیگنال اصلی را رمزنگاری کردیم باید مانند قبل صوت اصلی را بازیابی کنیم. پس صوت رمزنگاری شده را لود کردیم و دقیقاً مطابق قسمت قبل عمل کردیم:

## Decryption

```
[y_z, fs_z] = audioread('C:\Users\pc
world\Desktop\S&S_FinalProject_14022\recorded_encrypted.wav');
Binv = inv(B);
disp(Binv)
```

```

      0      0      0  20.0000      0
      0      0  1.0000      0      0
      0  0.1000      0      0      0
  0.1000      0      0      0      0
      0      0      0      0  1.0000
```

```
yz1 = filter(Band1, y_z);
yz2 = filter(Band2, y_z);
yz3 = filter(Band3, y_z);
yz4 = filter(Band4, y_z);
yz5 = filter(Band5, y_z);
```

```
Y_Z = [yz1, yz2, yz3, yz4, yz5];
```

```
xz1 = zeros(length(y_z), 1);
xz2 = zeros(length(y_z), 1);
xz3 = zeros(length(y_z), 1);
xz4 = zeros(length(y_z), 1);
xz5 = zeros(length(y_z), 1);
```

```
xz1 = filter(Band1, xz1);
xz2 = filter(Band2, xz2);
xz3 = filter(Band3, xz3);
xz4 = filter(Band4, xz4);
xz5 = filter(Band5, xz5);
```

```
X_Z = [xz1, xz2, xz3, xz4, xz5];
X_Z = Y_Z * Binv';
```

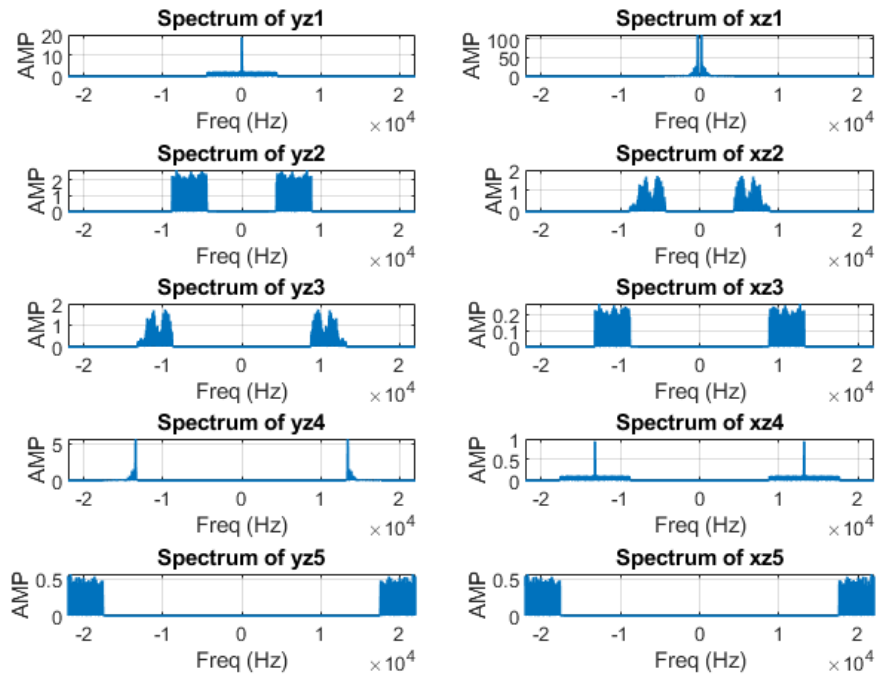
```
xz1 = X_Z(:,1);
xz2 = X_Z(:,2);
xz3 = X_Z(:,3);
xz4 = X_Z(:,4);
xz5 = X_Z(:,5);
ts_z = 1 / fs_z;
```

```
t_z = 0:ts_z:(length(y_z) - 1) * ts_z;

band1_fz = fs_z / 10;
band2_fz = 2 * (fs_z / 10);
band3_fz = 3 * (fs_z / 10);
band4_fz = 4 * (fs_z / 10);

cz1 = cos(2 * pi * band1_fz * t_z)';
cz2 = cos(2 * pi * band2_fz * t_z)';
cz3 = cos(2 * pi * band3_fz * t_z)';
cz4 = cos(2 * pi * band4_fz * t_z)';
xz1 = xz1 .*cz3;
xz1 = filter(Band1,2*xz1);
xz2 = xz2 .*cz1;
xz2 = filter(Band2,2*xz2);
xz3 = xz3 .*cz1;
xz3 = filter(Band3,2*xz3);
xz4 = xz4 .*cz3;
xz5 = filter(Band5,xz5);

figure;
subplot(5, 2, 1); FFT(yz1, fs_z); title('Spectrum of yz1');
subplot(5, 2, 2); FFT(xz1, fs_z); title('Spectrum of xz1');
subplot(5, 2, 3); FFT(yz2, fs_z); title('Spectrum of yz2');
subplot(5, 2, 4); FFT(xz2, fs_z); title('Spectrum of xz2');
subplot(5, 2, 5); FFT(yz3, fs_z); title('Spectrum of yz3');
subplot(5, 2, 6); FFT(xz3, fs_z); title('Spectrum of xz3');
subplot(5, 2, 7); FFT(yz4, fs_z); title('Spectrum of yz4');
subplot(5, 2, 8); FFT(xz4, fs_z); title('Spectrum of xz4');
subplot(5, 2, 9); FFT(yz5, fs_z); title('Spectrum of yz5');
subplot(5, 2, 10); FFT(xz5, fs_z); title('Spectrum of xz5');
```



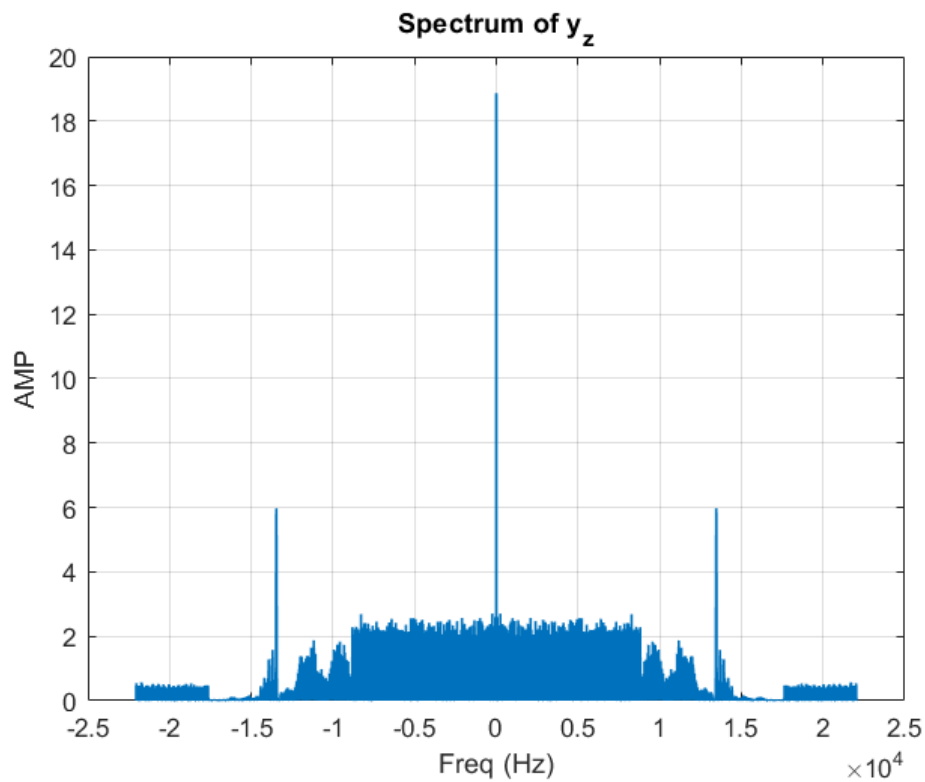
```
xz = xz1 + xz2 + xz3 + xz4 + xz5;
player = audioplayer(xz, fs_z);
% play(player)

audiowrite('C:\Users\pc
world\Desktop\S&S_FinalProject_14022\recorded_de_encrypted.wav', xz, fs_z);
```

## Spectrum of the encrypted audio

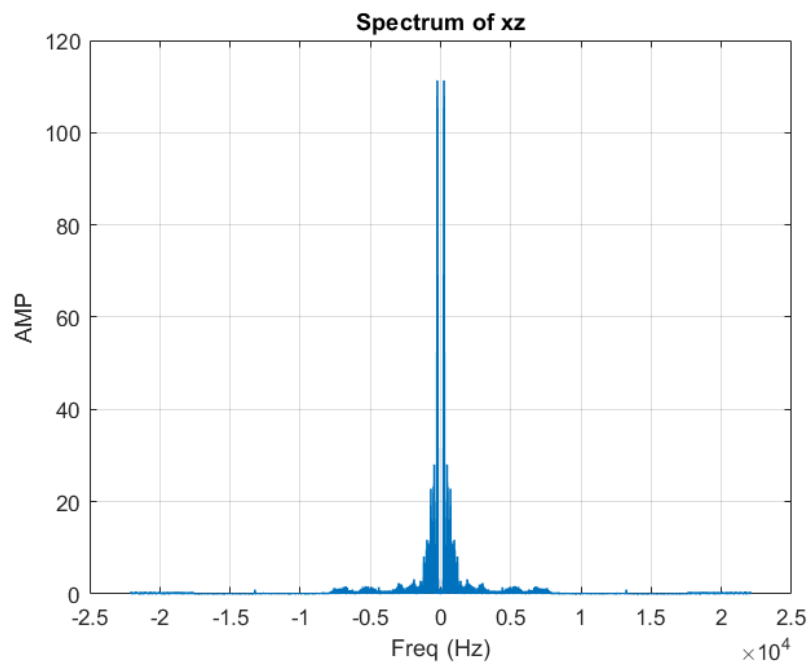
```
figure;
FFT(y_z, fs_z); title('Spectrum of y_z');
```





### Spectrum of the decoded audio(real audio)

```
figure;  
FFT(xz, fs_z); title('Spectrum of xz');
```



همانطور که مشاهده میشود صوت به خوبی بازپایی شده است.