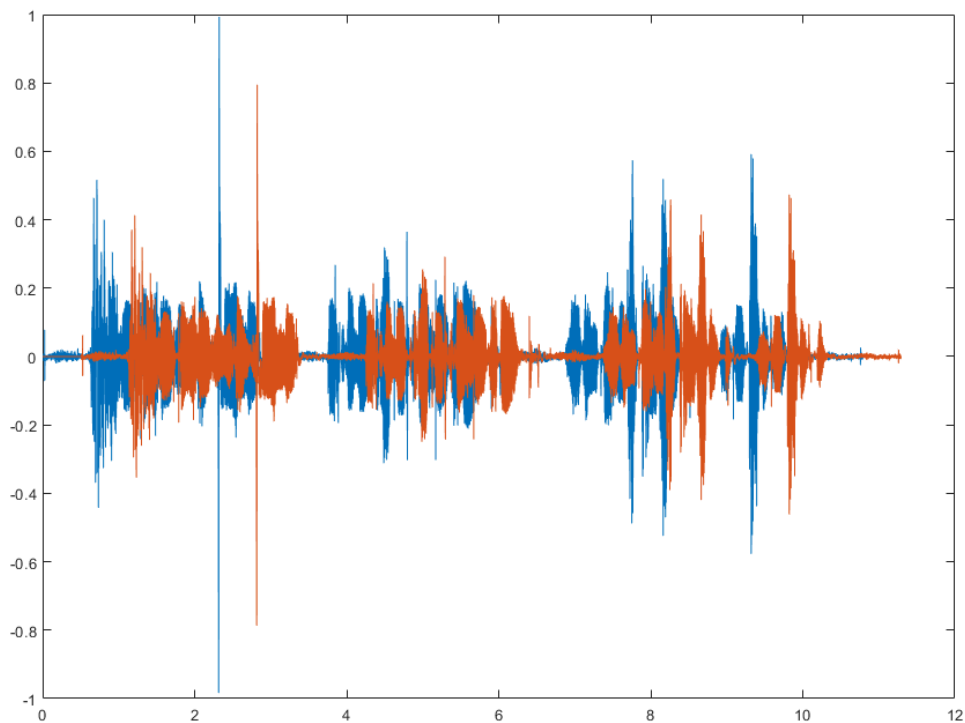
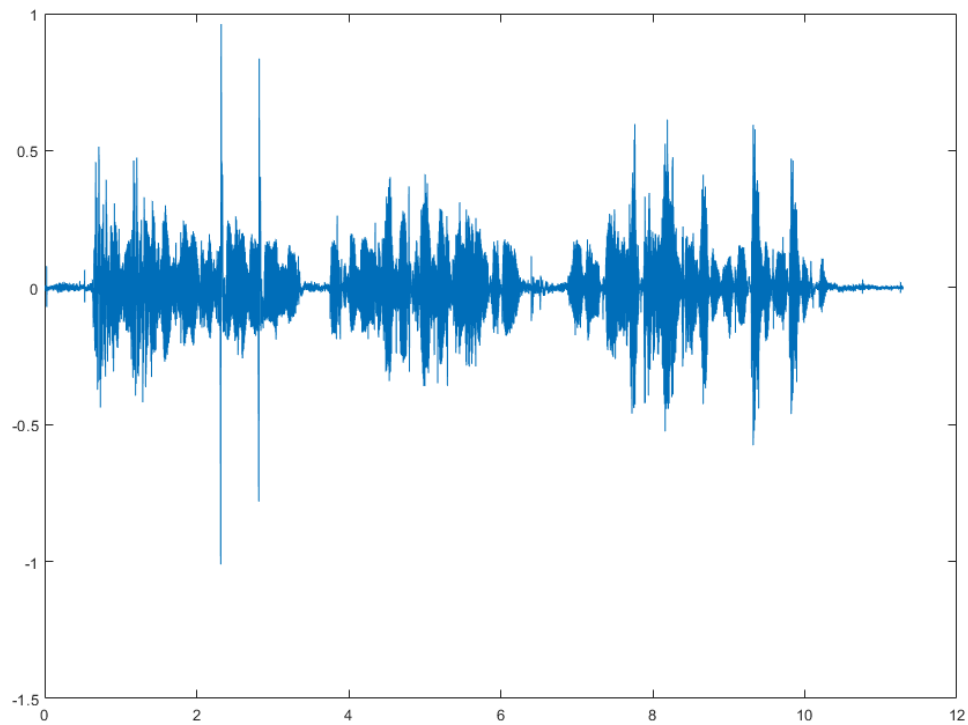


```
[data, fs] = audioread("Recording (4).m4a");  
data = data(:,1);  
data = data';
```

```
alpha = 0.8;  
Beta = round(0.5*fs);  
  
orig = [data,zeros(1,Beta)];  
echo = [zeros(1,Beta),data]*alpha;  
  
t = 0:1/fs:(length(orig)-1)/fs;  
dtecho = orig+echo;  
plot(t,orig)  
hold on  
plot(t,echo)  
hold off
```



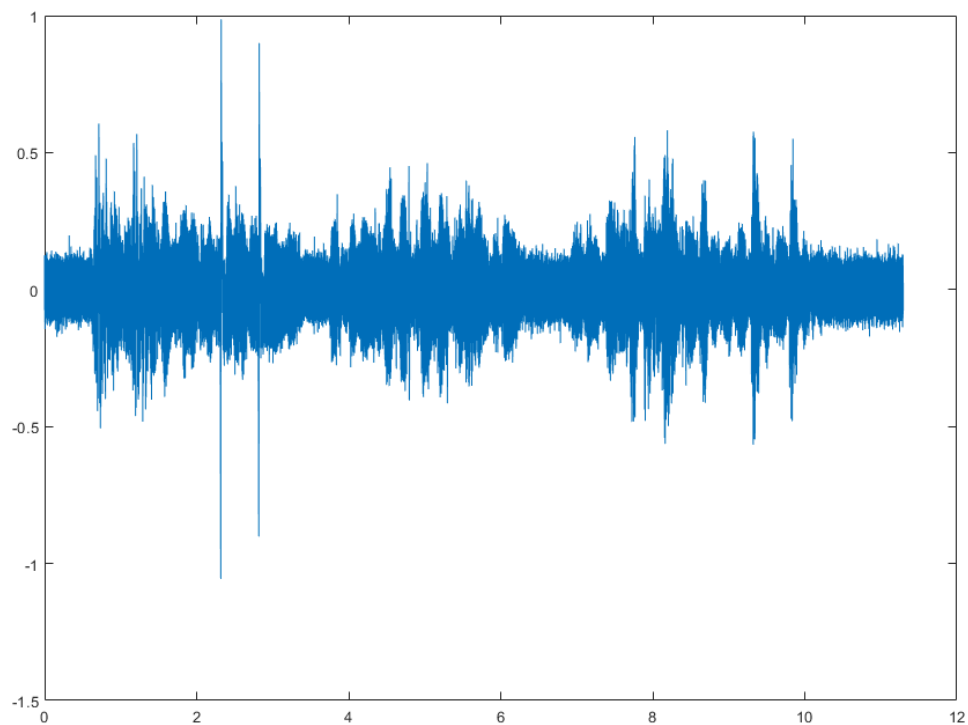
```
plot(t,dtecho)
```



```
noise =randn(1,length(t));  
noise = noise/max(noise)*0.2;  
dtechon = dtecho + noise;  
audiowrite("noisy_echoed_voice.wav",dtechon,fs);
```

Warning: Data clipped when writing file.

```
plot(t,dtechon)
```

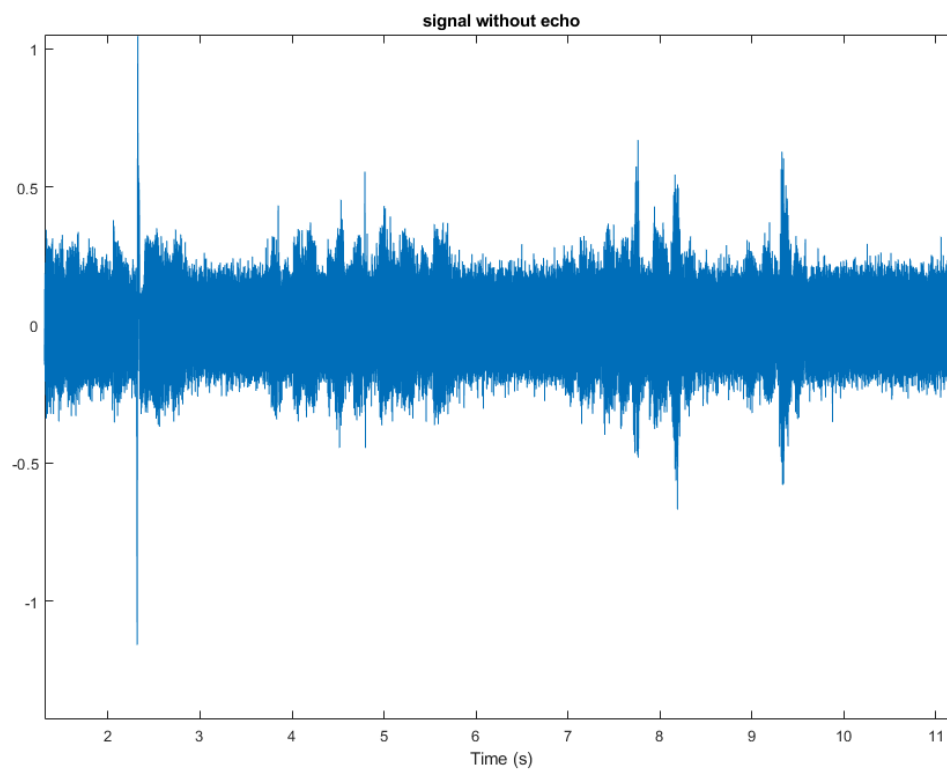


```
sound(dtechon,fs)
```

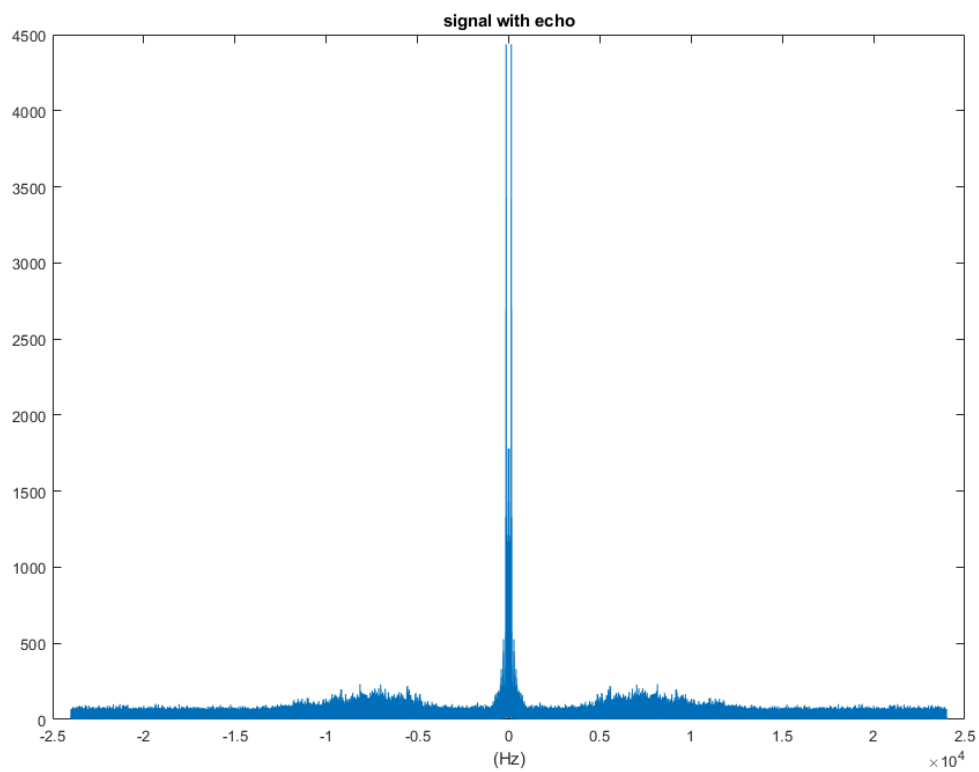
```
noecho = rmecho(dtechon,alpha,fs);
```

```
Beta is 0.500000  
Warning: Data clipped when writing file.
```

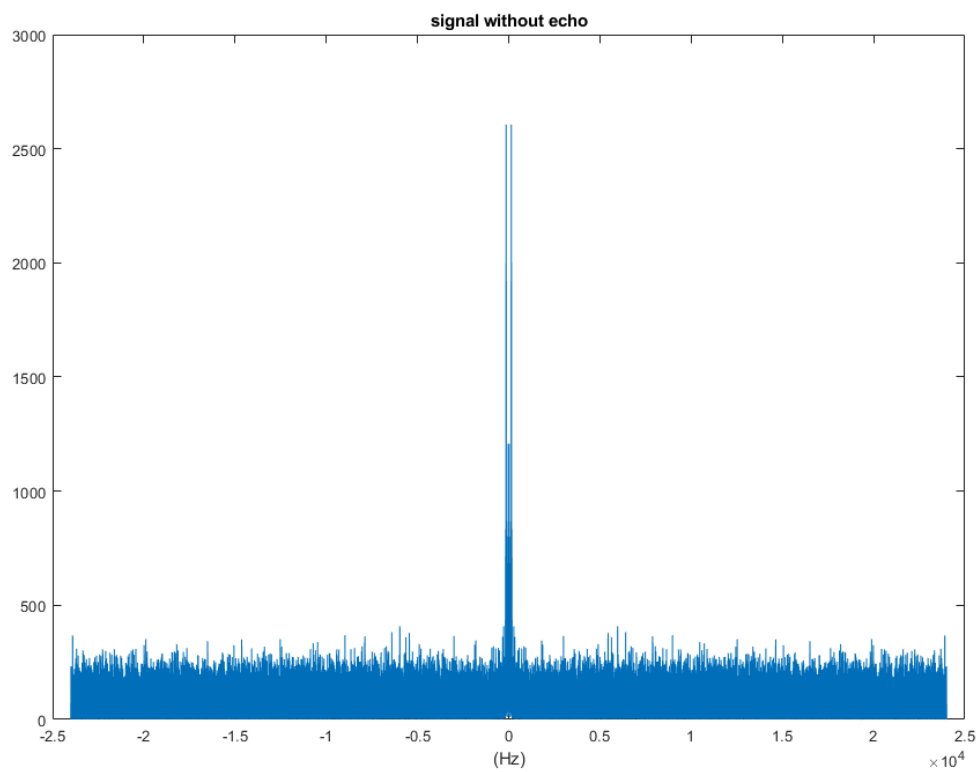
```
plot(t,noecho)  
title("signal without echo")  
xlabel('Time (s)')
```



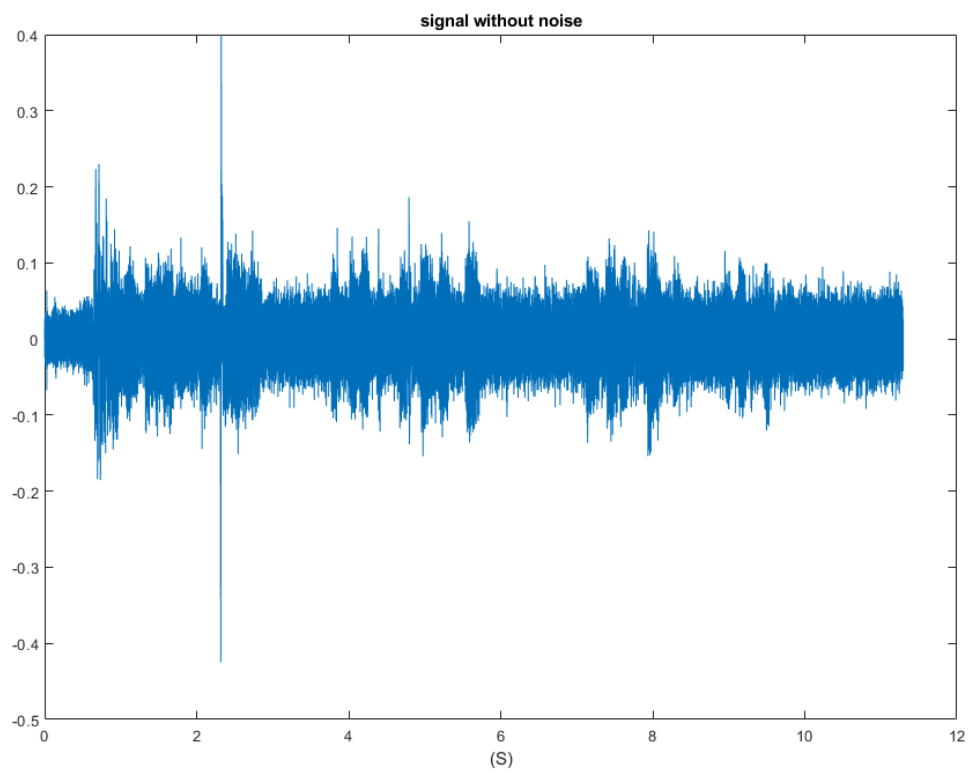
```
f= -fs/2:fs/(length(t)-1):fs/2;  
plot(f,abs(fftshift(fft(dtechon))))  
title("signal with echo")  
xlabel('(Hz)')
```



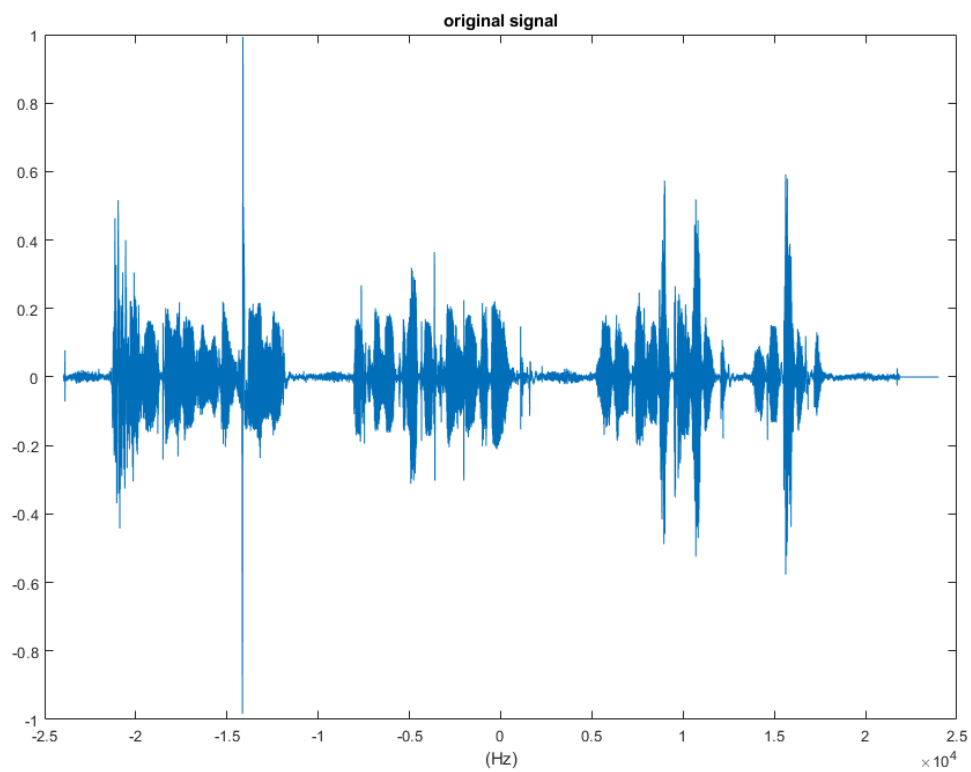
```
plot(f,abs(fftshift(fft(noecho))))  
title("signal without echo")  
xlabel('(Hz)')
```



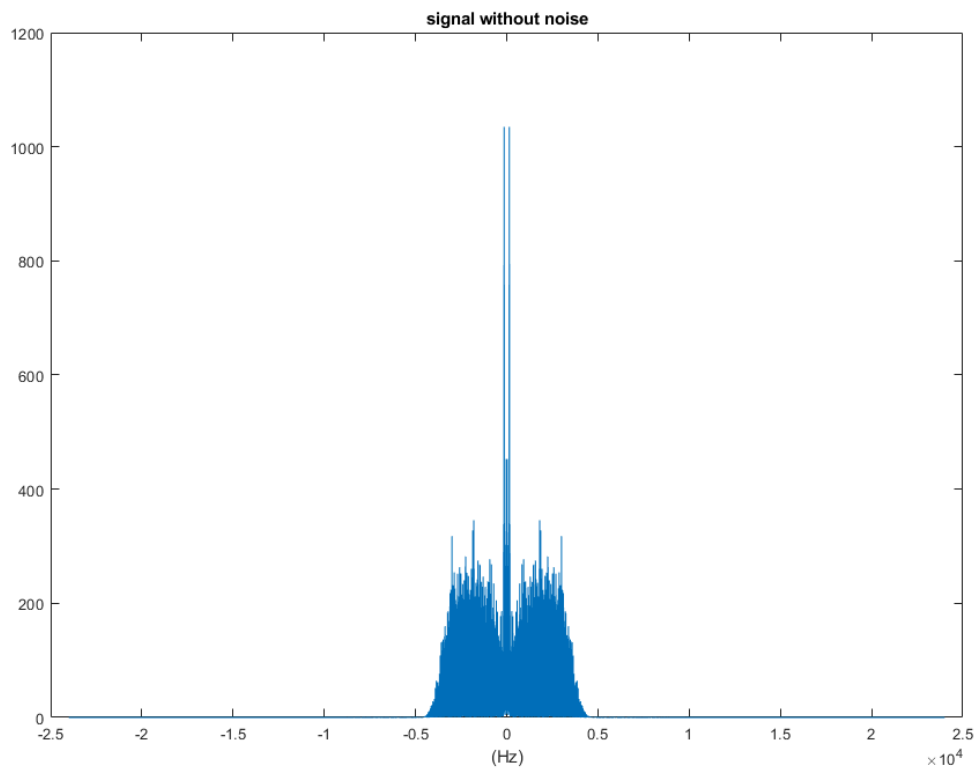
```
noechonoise = filter(b,1,noecho);  
plot(t,noechonoise)  
title("signal without noise")  
xlabel('(S)')
```



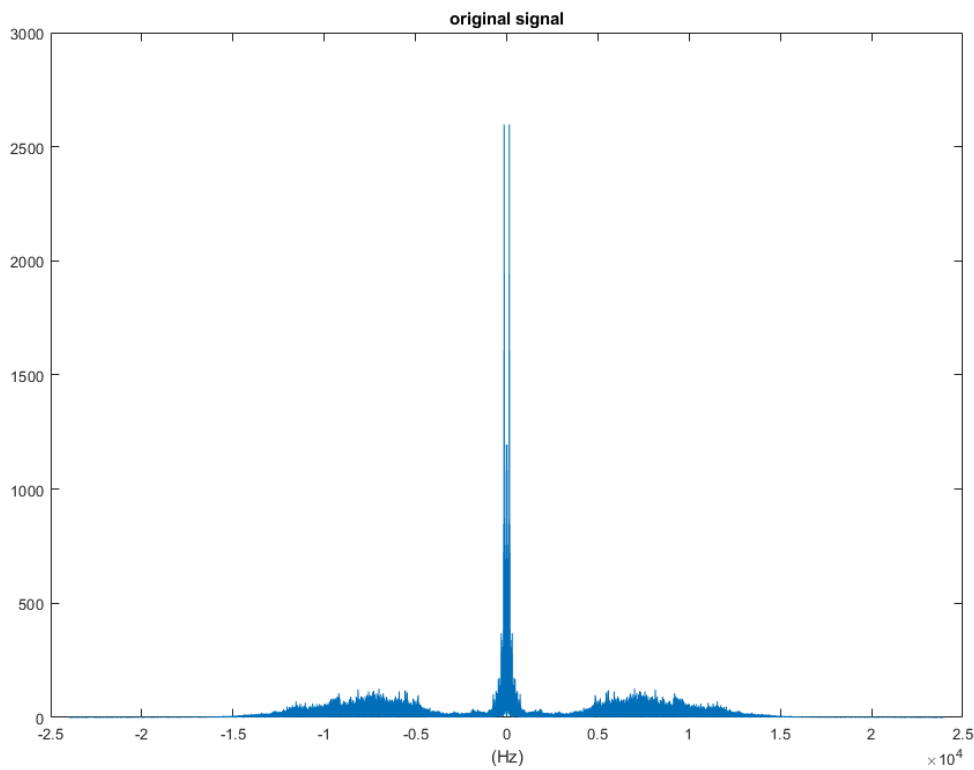
```
plot(f,orig)
title("original signal")
xlabel('(Hz)')
```



```
sound(noechonoise,fs)
plot(f,abs(fftshift(fft(noechonoise))))
title("signal without noise")
xlabel('(Hz)')
```

```
plot(f,abs(fftshift(fft(orig))))  
title("original signal")  
xlabel('(Hz)')
```



Part 2

```
pic = imread("house.jpg");
pic = rgb2gray(pic);
imshow(pic)
```

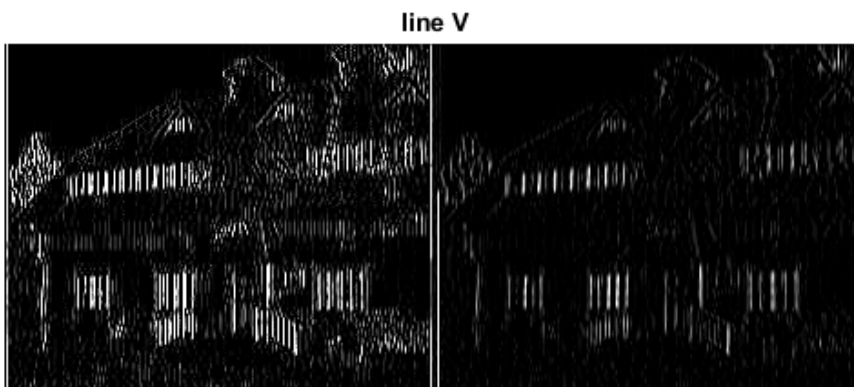


```
identity = [0,0,0;0,1,0;0,0,0];
idnt = conv2(pic,identity);
idnt = uint8(idnt);
imshow(idnt)
```

```
title("identity")
```

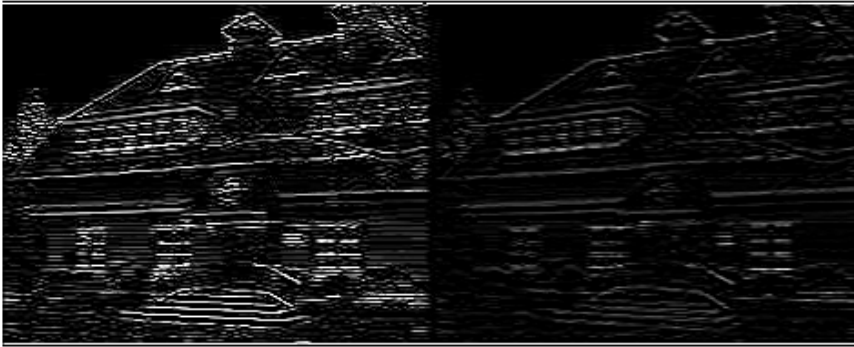


```
lineVk = [-1,2,-1;-1,2,-1;-1,2,-1];  
lineV = conv2(pic,lineVk);  
lineV = uint8(lineV);  
imshow(lineV)  
title("line V")
```



```
lineHk = [-1,-1,-1;2,2,2;-1,-1,-1];  
lineH = conv2(pic,lineHk);  
lineH = uint8(lineH);  
imshow(lineH)  
title("line H")
```

line H



```
avgmk = [0.1111,0.1111,0.1111;0.1111,0.1111,0.1111;0.1111,0.1111,0.1111];  
avgm = conv2(pic,avgmk);  
avgm = uint8(avgm);  
imshow(avgm)  
title("moving average")
```

moving average



```
gaussk = [0.0113,0.0838,0.0113;0.0838,0.6193,0.0838;0.0113,0.0838,0.0113];  
gauss = conv2(pic,gaussk);  
gauss = uint8(gauss);  
imshow(gauss)  
title("gaussian blur")
```

gaussian blur



```
outlinek = [-1,-1,-1;-1,8,-1;-1,-1,-1];  
outline = conv2(pic,outlinek);  
outline = uint8(outline);  
imshow(outline)  
title("outline")
```

outline



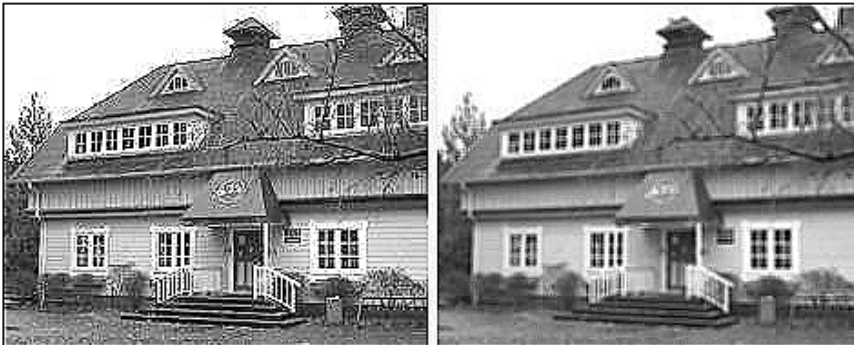
```
blurk = [0.0625,0.125,0.0625;0.125,0.25,0.125;0.0625,0.125,0.0625];  
blur = conv2(pic,blurk);  
blur = uint8(blur);  
imshow(blur)  
title("blur")
```

blur



```
sharpk = [0,-1,0;-1,5,-1;0,-1,0];  
sharp = conv2(pic,sharpk);  
sharp = uint8(sharp);  
imshow(sharp)  
title("sharpen")
```

sharpen



```
pic = imread("kobe.jpeg");  
pic = rgb2gray(pic);  
imshow(pic)
```



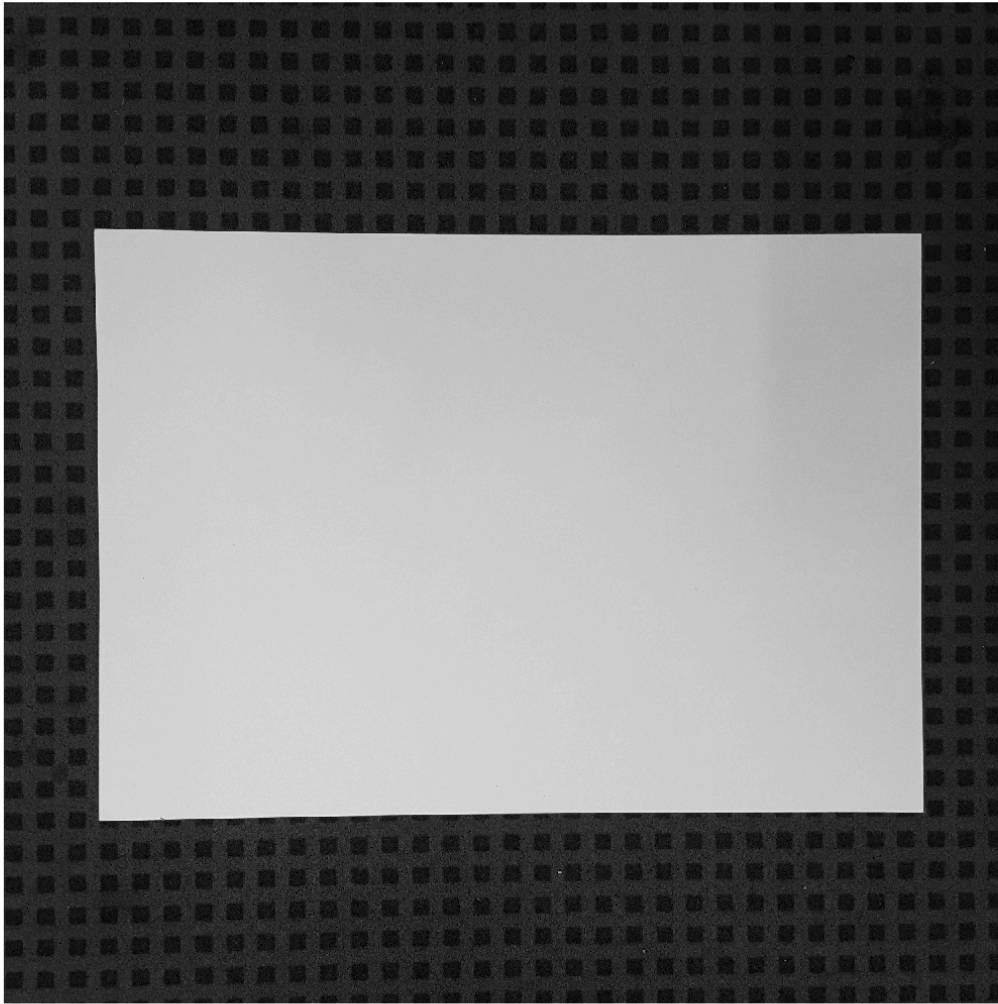
```
picres = imresize(pic,1/5);  
picres =imresize(picres,5,"Method","nearest");  
imshow(picres)
```



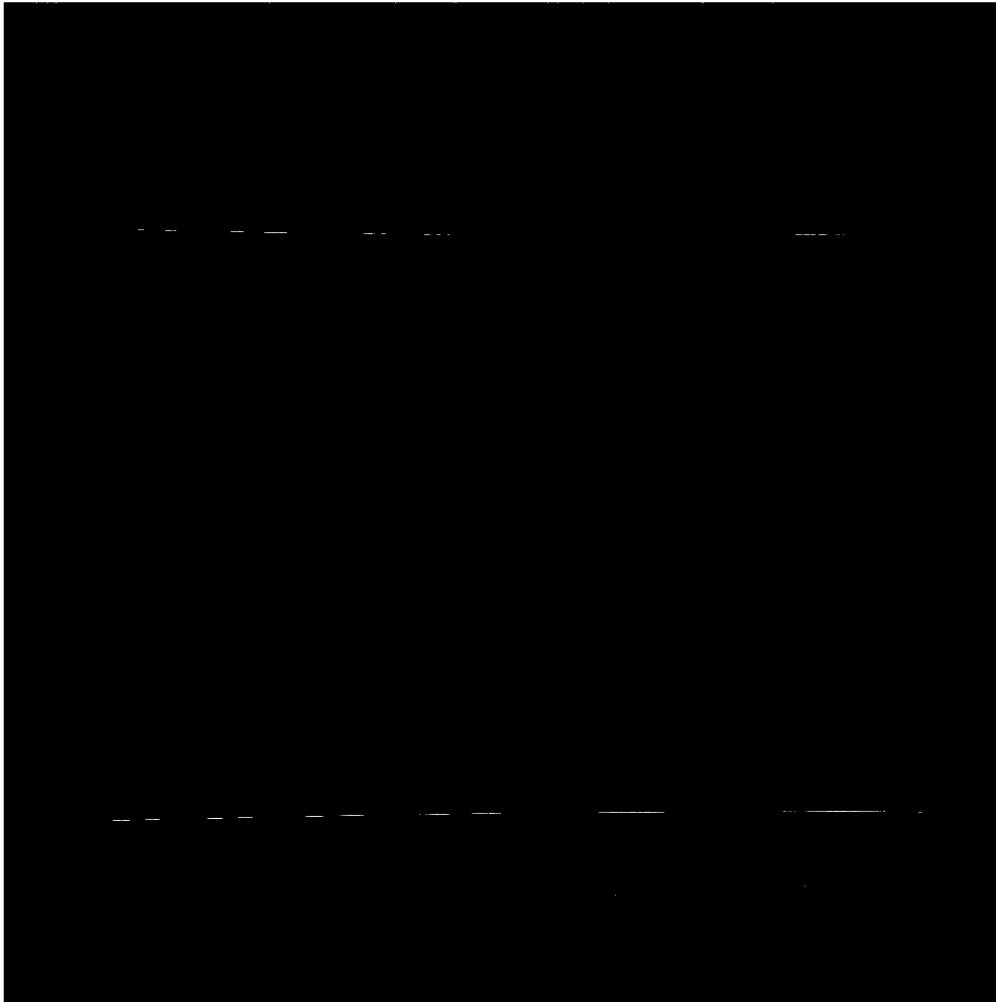
```
pic_recovered = conv2(picres,gaussk);  
for i = 0:20  
    %pic_recovered = conv2(pic_recovered,gaussk);  
    pic_recovered = conv2(pic_recovered,avgmk);  
end  
  
for i = 0:10  
    %pic_recovered = conv2(pic_recovered,sharpk);  
end  
  
pic_recovered = uint8(pic_recovered);  
imshow(pic_recovered)
```




```
pic = imread("page.jpg");  
pic = rgb2gray(pic);  
imshow(pic)
```



```
Hlines = conv2(lineHk,pic);  
Hlines(Hlines<150) = 0;  
Hlines = uint8(Hlines);  
imshow(Hlines)
```



```
[hl_row,hl_col] = find(Hlines);  
hl_row = sort(hl_row);  
hl_row = hl_row(hl_row >50 & hl_row<length(pic(:,1)) - 50);  
  
for i = 0:3  
    hl_row = hl_row(diff(hl_row) == 0 );  
end  
  
top = round(mean(hl_row(hl_row<hl_row(1)+ 100)))
```

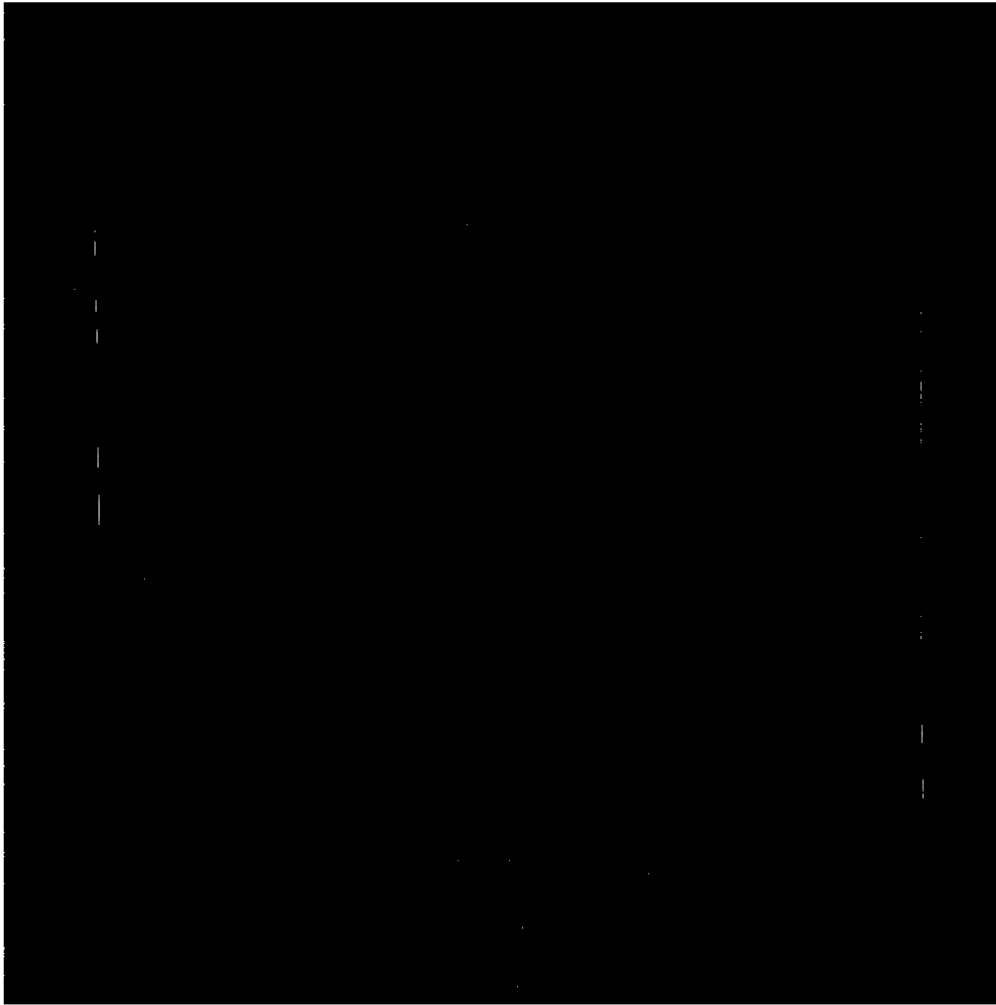
```
top = 690
```

```
down = round(mean(hl_row(hl_row>hl_row(1)+ 100)))
```

```
down = 2416
```

```
Vlines = conv2(lineVk,pic);
```

```
Vlines(Vlines<170) = 0;  
Vlines = uint8(Vlines);  
imshow(Vlines)
```



```
[vl_row,vl_col] = find(Vlines);  
vl_col = sort(vl_col);  
vl_col = vl_col(vl_col >50 & vl_col<length(pic(:,1)) - 50);  
  
for i = 0:3  
    vl_col = vl_col(diff(vl_col) == 0 );  
end  
left = round(mean(vl_col(vl_col<vl_col(1)+ 100)))
```

```
left = 282
```

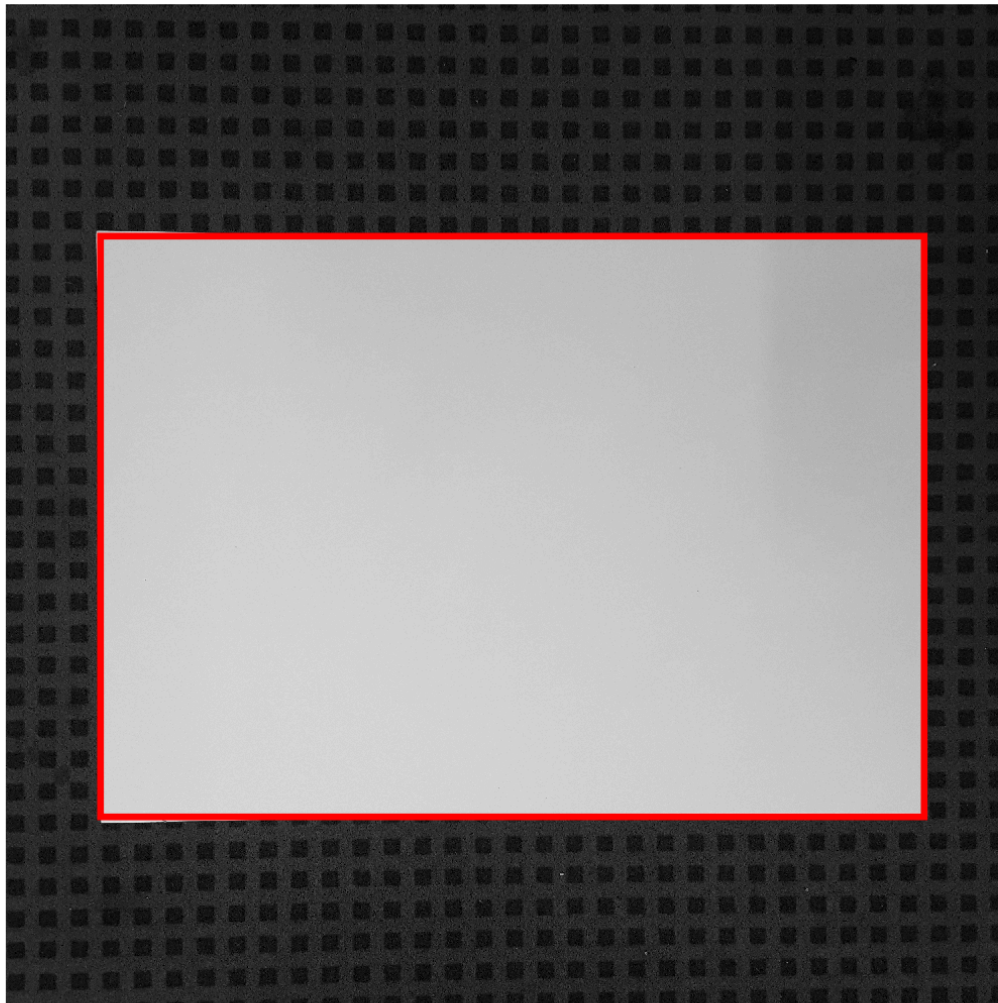
```
right = round(mean(vl_col(vl_col>vl_col(1)+ 100)))
```

```
right = 2730
```

```

H = down - top;
W = right - left;
imshow(pic)
rectangle('position',[left top W H],"EdgeColor","r","LineWidth",5)

```



```

function y = rmecho(x,alpha,fs)
c = rceps(x);

[px,locs] = findpeaks(c,'Threshold',0.01,'MinPeakDistance',0.02);
dl = locs(1)-1;
fprintf("Beta is %f" , dl/fs )
y = filter(1,[1 ,zeros(1,dl-1), alpha],x);
audiowrite("noisy_voice.wav",y,fs)

end

```