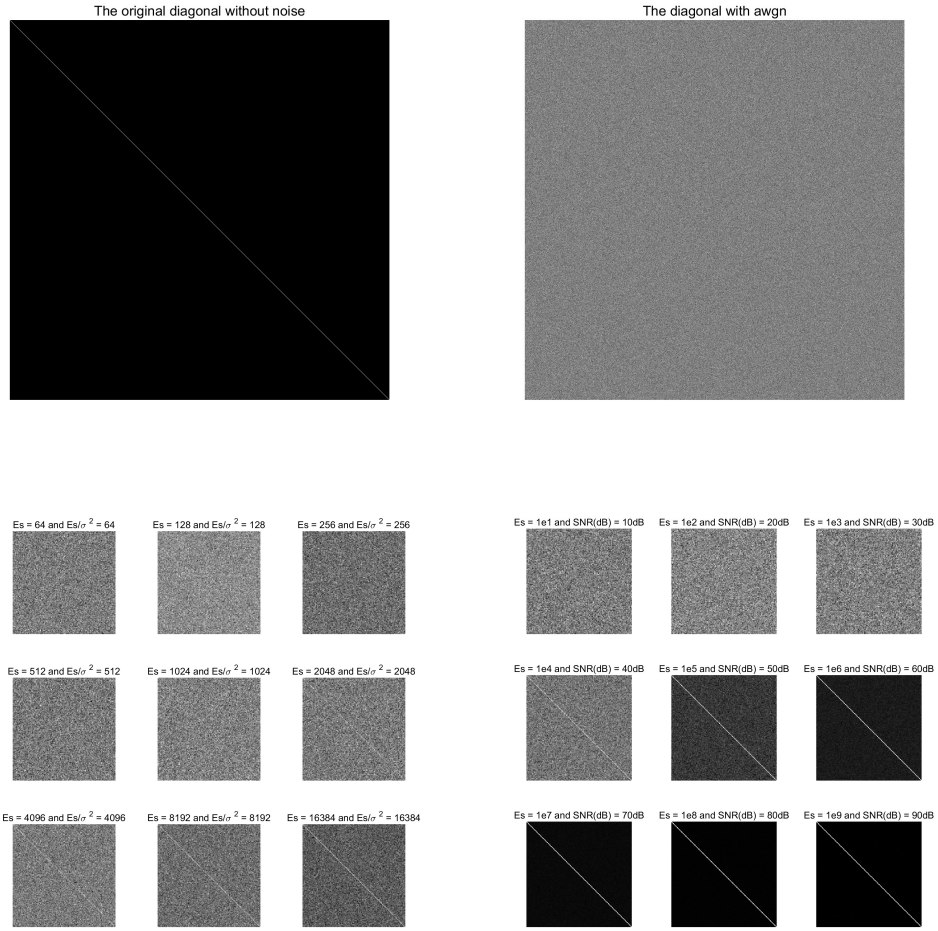
Problem analysis:

For this problem, several points require scrutiny.

- Total signal energy should be distributed to equal packs on each pixel.
- Gaussian white noise can be generated simply with the function “normrnd”.
- In order to find the adequate Es for the line, we need to change Es with reasonable ratio. We can simply make a wild guess that only when the signal power on each pixel is of the same order comparing to the noise power that the line is visible. According to this prediction, we can make adequate change to Es in problem d.
- Grayscale unifying. To draw the picture, we need to figure out some ways to unify the value of both Es and power of noise into grayscale. Fortunately, MATLAB have in-built functions and parameters which can solve this problem properly.
- Plotting and add notes to the figures.

Graphs 1-4:

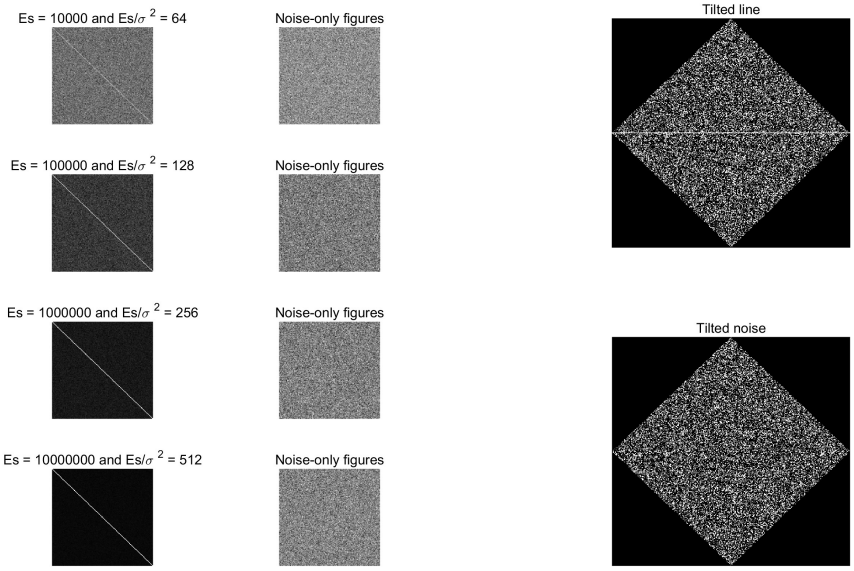


From my own perspective, when SNR reach 30dB, I can “just see” the signal diagonal. As the SNR goes bigger, the line become clearer and I can see it with confidence.

Graphs 5-6

After I checked the requirements I found that two more figures are needed to be added to this homework. The first is a contrast between noise-added diagonal and noise-only figure. The second is a "tilted" figure. Again, since I don't fully understand the usage of rotating them, I just do it as I think.

The noise-only figure seem to be less contrasted since I've unified the amplitude for diagonals, thus the relative contrast ratio is different between them.



Warning: The source code in PDF may contain '\n' added by system. Paste with caution.

Problem a)

- The goal is to generate an image with a uniform-distributed diagonal. Calculate the energy for each pixel and generate a matrix. The choose either to save it as an image file.

```
%% a image generation
image1 = diag(ones(1,1024));

Es = 4;
scale1 = sqrt(Es/1024);
image1 = image1.*scale1;
figure(1);
imshow(image1,[]); %colormap(gray);
disp(['The total signal energy of the image1 is '
num2str(sum(image1(:).^2))]) %expected to be 4
title('The original diagonal without noise')
```

```
% imagesave1 = getframe(gcf);
% imwrite(imagesave1,'imagesave1.bmp')
```

Problem b)

- Add Gaussian noise to the matrix and display it.

```
%% b noise add
sigma = 1; % so sigma^2 also = 1;
noise_gaussian1 = normrnd(0,sigma^2,1024,1024); % miu = 0; sigma^2 = 1;
image2 = image1 + noise_gaussian1;
figure(2);
imshow(image2,[]); % [] means auto scale to [low high];colormap(gray);
title('The diagonal with awgn')

% obviously cannot see anything but noise on the figure. i bet the Es is
% too small therefore the line is too dim.
```

Problem c) and d)

- Adjust the value of Es and see the difference between each try. Also printed out noise only images and rotated images.

```
%% c adjust the Es

% I've no idea about what is a subroutine, and I cannot get the point with
% "runs"... I'll just do it like what I think about the description

image1 = diag(ones(1,1024)); % re-definition of image1

% I plotted the figure by 1/64 of it original size to see the diagonal more
% clearly.

for i = 1:1:9
    Es = 2^(i+5); %add 5 just to make 1024 as the central subfigure.
    noise_gaussian2 = normrnd(0,1,1024,1024);
    scale2 = sqrt(Es/1024);
    image(:,:,i) = image1.*scale2 + noise_gaussian2;
end

figure(3);
for j = 1:9
    subplot(3,3,j)
```

```
        imshow(image(1:128,1:128,j),[]);axis square off; box off;
        title(['Es = ' num2str(2^(j+5)) ' and Es/\sigma ^2 = '
num2str(2^(j+5)/sigma^2)])
end

% at least for me, the SNR need to reach the level of 30dB so that I can
% distinguish the white line. I'm not so sure about the result since it is
% a little bit different from what I have learnt.


% I would like to use the dB representation of SNR to take a try.

for i = 1:1:9 %sqrt will be rational
        Es = 10^i;
        noise_gaussian3 = normrnd(0,1,1024,1024);
        scale3 = sqrt(Es/1024);
        image(:,:,i) = image1.*scale3 + noise_gaussian3;
end
figure(4)
for j = 1:9
        subplot(3,3,j)
        imshow(image(1:128,1:128,j),[]);axis square off; box off;
        title(['Es = 1e' num2str((j)) ' and SNR(dB) = '
num2str(10*log10(10^(j)/sigma^2)) 'dB'])
end

%% d add a noise contrast figure. Omitted it when reading the requirements.
% also add a tilted (rotated) figure since I don't know what does it mean.

for i = 1:1:4
        Es = 10^(i+3); %add 4 just to make figure more beautiful.
        noise_gaussian2 = normrnd(0,1,1024,1024);
        scale2 = sqrt(Es/1024);
        image(:,:,i) = image1.*scale2 + noise_gaussian2; % added noise figure
        imagenoise(:,:,i) = noise_gaussian2; % noise-only figure
end

figure(5);
for j = 1:4
        subplot(4,2,2*j - 1) %1/3/5/7 subs
        imshow(image(1:128,1:128,j),[]);axis square off; box off;
```

```
        title(['Es = ' num2str(10^(j+3)) ' and Es/\sigma ^2 = '
    num2str(2^(j+5)/sigma^2)])
    end

    for j = 1:4
        subplot(4,2,2*j)
        imshow(imagenoise(1:128,1:128,j),[]);axis square off; box off;
        title(['Noise-only figures'])
    end

    %% d2 just to show that I know how to tilt...but the imrotate have problem with bg
    color
    figure(6);
    subplot(211)

    tilt1 = imrotate(image(1:256,1:256,1),45);
    imshow(tilt1);axis square off;
    title(['Tilted line'])
    subplot(212)

    tilt2 = imrotate(imagenoise(1:256,1:256,1),45);
    imshow(tilt2);axis square off;
   title(['Tilted noise'])
```

MATLAB source code (which can also be found in .m file)

- Already re-typeset to fill the separated submission rule in announcement.