

## Лабораторна робота № 1

**Тема:** алгоритмічна компенсація випадкових похибок інформаційних сигналів у вимірювальному каналу

**Мета роботи:** отримати навички обробки випадкових похибок інформаційних сигналів у вимірювальному каналі за допомогою програми MATLAB.

### 1.1. Основні теоретичні відомості

Отримання повних апіорних відомостей про вимірювальну інформацію є складною задачею, так як в загальному випадку початкова вимірювальна інформація (двовимірний масив  $f_0(x, y)$ ) невідома, а для вимірювань доступне тільки спотворене відеозображення  $f_H(n, m)$  на виході ПФВЗ. Тому для отримання апіорної інформації про вимірювальну інформацію пропонується використовувати методики, що розроблені в підрозділі 2.4.

Для оцінки ефективності застосування методів алгоритмічної компенсації похибок потрібно визначити критерії, за якими оцінюються результати компенсації. Відомі типові критерії оцінки вірності передачі та відтворення аналогових і цифрових відеозображень. Ці критерії орієнтовані на оцінку амплітудних похибок та візуальної якості відеозображень [161, 162, 216, 310, 315, 316]. Вказані критерії обчислюють середньоквадратичне значення похибки відтворення амплітуди відеосигналу на виході вимірювального каналу. Це є загальна оцінка похибок всього відеозображення. При цьому похибка відтворення кожного дискретного значення амплітуди відеосигналу розглядається незалежно від похибок відтворення сусідніх дискретних значень. Такий підхід дозволяє з високою точністю дати загальну оцінку похибок яскравості і кольору ОВ, що наявні на сформованому відеозображенні.

Однак, при оцінці похибок відеозображень з вимірювальною інформацією про ГП необхідно в першу чергу враховувати викривлення форми перепадів амплітуди відеосигналу, що відповідають контурам ОВ. Оскільки в амплітудних критеріях вказані викривлення не враховуються безпосередньо, то оцінка похибок відеозображень з вимірювальною інформацією на основі амплітудних критеріїв має низьку точність та об'єктивність. Окрім того, кількісну оцінку похибок ГП доцільно виражати в одиницях просторових координат відеозображення, а не в одиницях амплітуди відеосигналу.

					<i>МММТ.420.004.004 – ЗЛІ</i>	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

## 1.2. Виконання роботи

Варіант	Початкові дані для розрахунку					
	Корисний сигнал вимірювальної інформації			Випадкові викривлення (шум)		
	Параметр кореляційної функції $\alpha_x$ , (д.т.) <sup>-1</sup>	Параметр кореляційної функції $\alpha_y$ , (д.т.) <sup>-1</sup>	Середньо-квадратичне значення амплітуди $\sigma_{f2}$ , д.р.	Параметр кореляційної функції $\alpha_{x\xi}$ , (д.т.) <sup>-1</sup>	Параметр кореляційної функції $\alpha_{y\xi}$ , (д.т.) <sup>-1</sup>	Середньо-квадратичне значення амплітуди $\sigma_\xi$ , д.р.
4	0,064	0,037	28,6	1,552	0,881	3,9

Примітки: 1. Канал  $Y$  відповідає яскравості кольорового відеозображення, що обчислюється за формулою  $Y = 0,30R + 0,59G + 0,11B$ , де  $R$ ,  $G$  і  $B$  – відеосигнали в каналах червоного, зеленого та синього кольору.

2. Частота зрізу  $\omega_{10}$  визначається за умови  $|W_{\text{КВ}}(j\omega_1, j\omega_2)| = 0,5$  при  $\omega_2 = 0$ , частота зрізу  $\omega_{20}$  – за умови  $|W_{\text{КВ}}(j\omega_1, j\omega_2)| = 0,5$  при  $\omega_1 = 0$ .

3. \* – частота зрізу лежить за межами смуги пропускання  $0 \dots 0,5$  (д.т.)<sup>-1</sup> для цифрового відеозображення, що представлено дискретними відліками.

### 1.2.1. Програма для виконання фільтрації.

```
clear all;
Dir_Test='D:\My_doc 02-11-10\MM_3\Test\';
Dir_Im='D:\My_doc 02-11-10\MM_3\CFA_Natural_Stone\';
FileMask='.bmp'; Type_Real_Im=1;
Ideal_Im_name='D:\My_doc 02-11-10\MM_3\Ideal_Image\Ideal_1982_8s_12s_RGB.bmp';
M=1024; N=1024;
Y_Background=27/255;
Y_Figure=227/255;

Threshold_Segm=(Y_Background+Y_Figure)/2;
N_Im=3;
N_Frame_Filt=10;
Type_Figure=2;
Boun=10;
Type_Noise=1;
AlfaX_OV=0.700;
NG=50;
L_PK=12;
PSF_Size=L_PK+2;
SNR=40;
PDov=0.95;
Filtered_Method=3;
Filt_M_Rect=0.25;
Filt_N_Rect=0.25;
Filt_M_Rect=round(Filt_M_Rect*M); Filt_N_Rect=round(Filt_N_Rect*N);
Filt_N_Mask=3;
F_Order_1D=1;
Im_Show=0;
Delta_Show=1;
Im_Write=0;
X_Center=round(N/2); Y_Center=round(M/2);
RectXMax=round(M/sqrt(2)-2*Boun-2*L_PK);
```

					<i>МММТ.420.004.004 – ЗЛІ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

RectYMin=round((M/sqrt(2)-2*Boun-2*L_PK)/10);
EllipsAMax=round((M/sqrt(2)-2*Boun-2*L_PK)/2);
EllipsBMin=round((M/sqrt(2)-2*Boun-2*L_PK)/2/3);
A_SKO_Im_M=zeros(N_Im,1); A_SNR_dB_Im_M=zeros(N_Im,1); A_PSNR_dB_Im_M=zeros(N_Im,1);
A_SKO_Im_R=zeros(N_Im,1); A_SNR_dB_Im_R=zeros(N_Im,1); A_PSNR_dB_Im_R=zeros(N_Im,1);
B_M_XOR=zeros(N_Im,1); B_R_XOR=zeros(N_Im,1);
S_Figure=zeros(N_Im,1); S_Figure_M=zeros(N_Im,1); S_Figure_R=zeros(N_Im,1);
Centroid_X=zeros(N_Im,1); Centroid_X_M=zeros(N_Im,1); Centroid_X_R=zeros(N_Im,1);
Centroid_Y=zeros(N_Im,1); Centroid_Y_M=zeros(N_Im,1); Centroid_Y_R=zeros(N_Im,1);
W_Figure=zeros(N_Im,1); W_Figure_M=zeros(N_Im,1); W_Figure_R=zeros(N_Im,1);
H_Figure=zeros(N_Im,1); H_Figure_M=zeros(N_Im,1); H_Figure_R=zeros(N_Im,1);
ED_Figure=zeros(N_Im,1); ED_Figure_M=zeros(N_Im,1); ED_Figure_R=zeros(N_Im,1);
% rand('state',sum(100*clock)); randn('state',sum(100*clock));
rand('state',0); randn('state',0);
ListFileInDir=dir(Dir_Im);
NumberOfFileInDir=length(ListFileInDir(:,1));
NumberOfFileToMask=0;
ListFileToMask=[];
for i=1:NumberOfFileInDir,
    if strfind(ListFileInDir(i).name, upper(FileMask))>=1
        NumberOfFileToMask=NumberOfFileToMask+1;
        ListFileToMask(NumberOfFileToMask).name=ListFileInDir(i).name;
    end;
    if strfind(ListFileInDir(i).name, lower(FileMask))>=1
        NumberOfFileToMask=NumberOfFileToMask+1;
        ListFileToMask(NumberOfFileToMask).name=ListFileInDir(i).name;
    end;
end;
h_waitbar = waitbar(0, 'Обробка зображен');
for i_Im=1:N_Im
    waitbar(i_Im/N_Im,h_waitbar);

    Im_Frame=zeros(M,N);
    Im_Frame_M=zeros(M,N);
    Im_Frame_R=zeros(M,N);

    switch Type_Figure
        case 1
            RectX=round(RectXMax-(RectXMax-RectYMin)*rand);
            RectY=round(RectXMax-(RectXMax-RectYMin)*rand);
            RectTheta=180*rand;
            Im_Frame_J=zeros(RectY, RectX)+Y_Figure;
            Im_Frame_J = imrotate(Im_Frame_J,RectTheta, 'bicubic', 'loose');
            [M_J, N_J]=size(Im_Frame_J);
            Im_Frame(round(Y_Center-M_J/2+1):round(Y_Center+M_J/2), round(X_Center-
N_J/2+1):round(X_Center+N_J/2))=Im_Frame_J;
            Im_Frame(Im_Frame==0)=Y_Background;
            Im_Frame=mat2gray(Im_Frame, [0 1]);
        case 2
            EllipsA=round(EllipsAMax-(EllipsAMax-EllipsBMin)*rand);
            EllipsB=round(EllipsAMax-(EllipsAMax-EllipsBMin)*rand);
            EllipsTheta=180*rand;
            Im_Frame_J=zeros(EllipsB*2, EllipsA*2);
            for j=1:EllipsB*2
                for i=1:EllipsA*2
                    if abs(i-EllipsA)<=EllipsA*sqrt(1-(EllipsB-j)^2/EllipsB^2)
                        Im_Frame_J(j,i)=Y_Figure;
                    end
                end
            end
            Im_Frame_J = imrotate(Im_Frame_J,EllipsTheta, 'bicubic', 'loose');

```

						Арк.
					<i>МММТ.420.004.004 – 3Л1</i>	5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

[M_J, N_J]=size(Im_Frame_J);
Im_Frame(round(Y_Center-M_J/2+1):round(Y_Center+M_J/2), round(X_Center-
N_J/2+1):round(X_Center+N_J/2))=Im_Frame_J;
Im_Frame(Im_Frame<Y_Background)=Y_Background;
Im_Frame=mat2gray(Im_Frame, [0 1]);
case 3
if i_Im<=NumberOfFileToMask
Im_Frame=rgb2gray(im2double(imread(Ideal_Im_name)));

Im_Frame_M=rgb2gray(im2double(imread(strcat(Dir_Im,ListFileToMask(i_Im).name))));
M=size(Im_Frame,1); N=size(Im_Frame,2);
Im_Frame_R=zeros(M,N);
else
N_Im=i_Im-1; break;
end;
end; % switch Type_Figure

if Type_Figure<3
PSF_DVC=fspecial('gaussian',PSF_Size,L_PK/6);
Im_Frame=imfilter(Im_Frame, PSF_DVC, 'circular');
switch Type_Noise
case 0
Disp_Gaussian=1/10^(2*SNR/20);
Im_Frame_M=imnoise(Im_Frame,'gaussian',0,Disp_Gaussian);
case 1
gamma1=rand(NG,1); gamma2=rand(NG,1); gamma4=rand(NG,1)*2*pi;
psbp=rand(NG,1);
for lll=1:NG
if psbp(lll)>=0.5 psbp(lll)=1; else psbp(lll)=-1; end
end;
ksi_k=psbp.*sqrt(-log(gamma1));
e1=cos(gamma4); e2=sin(gamma4); % iðð x iðð y
eta_mod=AlfaX_OV.*sqrt(1./gamma2.^2-1);
eta1_k=eta_mod.*e1; eta2_k=eta_mod.*e2;
h2 = waitbar(0, strcat('Wait ... generated Im Noise ', num2str(i_Im)));
Im_Noise=zeros(M,N);
for lll=1:NG
waitbar(lll/NG,h2);
for jjj=1:M for iii=1:N

Im_Noise(jjj,iii)=Im_Noise(jjj,iii)+sqrt(2)*ksi_k(lll)*sin(eta1_k(lll)*iii+eta2_k(lll)*j
jj+pi/4);
end; end;
end;
close(h2);
Im_Noise=1/10^(SNR/20)/sqrt(NG)*Im_Noise;
Im_Frame_M=Im_Frame+Im_Noise;
end % switch Type_Noise
end; % if Type_Figure<3

switch Filtered_Method
case 1
Snoise2=abs(fft2(Im_Frame_M-Im_Frame, M*2, N*2)).^2;
Simage2=abs(fft2(Im_Frame, M*2, N*2)).^2;
HF=zeros(M*2,N*2);
HF=fftshift(Simage2./(Simage2+Snoise2));
h17=fspecial('average',17);
HF=imfilter(HF,h17);
HF=zeros(M*2,N*2);

```

						Арк.
					<i>МММТ.420.004.004 – 3Л1</i>	6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

HF((M-2*Filt_M_Rect+1):(M+2*Filt_M_Rect),(N
2*Filt_N_Rect+1):(N+2*Filt_N_Rect))=1;
Fourie_Im=fftshift(fft2(Im_Frame_M, M*2, N*2));
iFourie_Im=real(ifft2(fftshift(HF.*Fourie_Im)));
Im_Frame_R=iFourie_Im(1:M, 1:N);
case 2
HF=zeros(M,N);
HF((round(M/2)-Filt_M_Rect+1):(round(M/2)+Filt_M_Rect),...
(round(N/2)-Filt_N_Rect+1):(round(N/2)+Filt_N_Rect))=1;
[f1 f2]=freqspace([M N], 'meshgrid');
hf=real(fsamp2(f1,f2,HF,[Filt_N_Mask Filt_N_Mask])); hf=hf/sum(sum(hf));
Im_Frame_R=imfilter(Im_Frame_M,hf);
case 3
Snoise2=abs(fft2(Im_Frame_M-Im_Frame, M*2, N*2)).^2;
Simage2=abs(fft2(Im_Frame, M*2, N*2)).^2;
HF=zeros(M*2,N*2);
HF=fftshift(Simage2./(Simage2+Snoise2));
h17=fspecial('average',17);
HF=imfilter(HF,h17);
Fourie_Im=fftshift(fft2(Im_Frame_M, M*2, N*2));
iFourie_Im=real(ifft2(fftshift(HF.*Fourie_Im)));
Im_Frame_R=iFourie_Im(1:M, 1:N);
case 4
Snoise=abs(fft2(Im_Frame_M-Im_Frame)).^2;
Simage=abs(fft2(Im_Frame)).^2;
HF=zeros(M,N);
HF=fftshift(Simage./(Simage+Snoise));
h17=fspecial('average',17);
HF=imfilter(HF,h17);
[f1 f2]=freqspace([M N], 'meshgrid');
hf=real(fsamp2(f1,f2,HF,[Filt_N_Mask Filt_N_Mask])); hf=hf/sum(sum(hf));
Im_Frame_R=imfilter(Im_Frame_M,hf);
case 5
F_1D=linspace(0,1,round(N/2));
HF_1D=zeros(1,round(N/2));
HF_1D(1:Filt_N_Rect)=1;
[B_1D,A_1D]=yulewalk(F_Order_1D,F_1D,HF_1D);
for jjj=1:M
Im_Frame_R(jjj,:) = filter(B_1D, A_1D, Im_Frame_M(jjj,:));
end;
case 6
F_1D=linspace(0,1,round(N/2));
Snoise1=abs(fft(Im_Frame_M(round(M/2),:)-Im_Frame(round(M/2),:), N)).^2;
Simage1=abs(fft(Im_Frame(round(M/2),:), N)).^2;
HF_1D=Simage1./(Simage1+Snoise1);
h17=fspecial('average',17);
HF_1D=imfilter(HF_1D(1:round(N/2)),h17);
[B_1D,A_1D]=yulewalk(F_Order_1D,F_1D,HF_1D);
B_1D=B_1D*sum(A_1D)/sum(B_1D);
for jjj=1:M
Im_Frame_R(jjj,:) = filter(B_1D, A_1D, Im_Frame_M(jjj,:));
end;
case 7
Im_Frame_R=zeros(M,N);
h1_waitbar = waitbar(0,'Obrobka neruchomych zobrazen');
if Type_Figure<3
for iii=1:N_Frame_Filt
waitbar(iii/N_Frame_Filt,h1_waitbar);
switch Type_Noise

```

Змн.	Арк.	№ докум.	Підпис	Дата

МММТ.420.004.004 – 3ЛІ

Арк.

7

```

case 0
    Disp_Gaussian=1/10^(2*SNR/20);
    Im_Frame_M=imnoise(Im_Frame,'gaussian',0,Disp_Gaussian);
case 1
    gamma1=rand(NG,1); gamma2=rand(NG,1); gamma4=rand(NG,1)*2*pi;
    psbp=rand(NG,1);
    for lll=1:NG
        if psbp(lll)>=0.5 psbp(lll)=1; else psbp(lll)=-1; end;
    end;
    ksi_k=psbp.*sqrt(-log(gamma1));
    e1=cos(gamma4); e2=sin(gamma4); % ið x ið y
    eta_mod=AlfaX_OV.*sqrt(1./gamma2.^2-1);
    eta1_k=eta_mod.*e1; eta2_k=eta_mod.*e2;
    h2 = waitbar(0,
    strcat('Wait ... generated Im Noise ',num2str(iii)));
    Im_Noise=zeros(M,N);
    for lll=1:NG
        waitbar(lll/NG,h2);
        for jjj=1:M for iii=1:N
            Im_Noise(jjj,iii)=Im_Noise(jjj,iii)+sqrt(2)*ksi_k(lll)
            *sin(eta1_k(lll)*iii+eta2_k(lll)*jjj+pi/4);
        end; end;
    end;
    close(h2);
    Im_Noise=1/10^(SNR/20)/sqrt(NG)*Im_Noise;
    Im_Frame_M=Im_Frame+Im_Noise;
end; % switch Type_Noise
Im_Frame_R=Im_Frame_R+Im_Frame_M;
end;
else
    for iii=1:N_Frame_Filt
        waitbar(iii/N_Frame_Filt,h1_waitbar);
        if iii<=NumberOfFileToMask
            Im_Frame_M=rgb2gray(im2double(imread(
                strcat(Dir_Im,ListFileToMask(iii).name)
            )));
            Im_Frame_R=Im_Frame_R+Im_Frame_M;
        else
            N_Frame_Filt=iii-1; break;
        end;
    end;
end;
Im_Frame_R=Im_Frame_R/N_Frame_Filt;
close(h1_waitbar);
end; % switch Filtered_Method

Pixel_Value=zeros(M,N); Pixel_Value_M=zeros(M,N); Pixel_Value_R=zeros(M,N);
if Y_Figure<Threshold_Segm
    Pixel_Value(1+Boun:M-Boun,1+Boun:N-Boun)=roicolor(Im_Frame(1+Boun:M-Boun,1+Boun:N-
Boun), 0, Threshold_Segm);
    Pixel_Value_M(1+Boun:M-Boun,1+Boun:N-Boun)=roicolor(Im_Frame_M(1+Boun:M-
Boun,1+Boun:N-Boun), 0, Threshold_Segm);
    Pixel_Value_R(1+Boun:M-Boun,1+Boun:N-Boun)=roicolor(Im_Frame_R(1+Boun:M-
Boun,1+Boun:N-Boun), 0, Threshold_Segm);
else
    Pixel_Value(1+Boun:M-Boun,1+Boun:N-Boun)=roicolor(Im_Frame(1+Boun:M-Boun,1+Boun:N-
Boun), Threshold_Segm, 1);
    Pixel_Value_M(1+Boun:M-Boun,1+Boun:N-Boun)=roicolor(Im_Frame_M(1+Boun:M-
Boun,1+Boun:N-Boun), Threshold_Segm, 1);
    Pixel_Value_R(1+Boun:M-Boun,1+Boun:N-Boun)=roicolor(Im_Frame_R(1+Boun:M-
Boun,1+Boun:N-Boun), Threshold_Segm, 1);

```

						<i>MMMT.420.004.004 – 3Л1</i>	Арк.
							8
Змн.	Арк.	№ докум.	Підпис	Дата			

```

end;

[L num]=bwlabel(Pixel_Value, 8);
[L_M num_M]=bwlabel(Pixel_Value_M, 8);
[L_R num_R]=bwlabel(Pixel_Value_R, 8);

feats=imfeature(L, 'Area', 'Centroid', 8);
feats_M=imfeature(L_M, 'Area', 'Centroid', 8);
feats_R=imfeature(L_R, 'Area', 'Centroid', 8);

Areas=zeros(num); CentX=zeros(num); CentY=zeros(num);
Areas_M=zeros(num_M); CentX_M=zeros(num_M); CentY_M=zeros(num_M);
Areas_R=zeros(num_R); CentX_R=zeros(num_R); CentY_R=zeros(num_R);
for i=1:1:num
    Areas(i)=feats(i).Area; CentX(i)=feats(i).Centroid(1);
CentY(i)=feats(i).Centroid(2);
end;
for i=1:1:num_M
    Areas_M(i)=feats_M(i).Area; CentX_M(i)=feats_M(i).Centroid(1);
CentY_M(i)=feats_M(i).Centroid(2);
end;
for i=1:1:num_R
    Areas_R(i)=feats_R(i).Area; CentX_R(i)=feats_R(i).Centroid(1);
CentY_R(i)=feats_R(i).Centroid(2);
end;

[rows cols]=size(Pixel_Value);
idx=find(Areas>1000&CentX>50&CentY>50&CentX<(cols-50)&CentY<(rows-50));
idx_M=find(Areas_M>1000&CentX_M>50&CentY_M>50&CentX_M<(cols-50)&CentY_M<(rows-50));
idx_R=find(Areas_R>1000&CentX_R>50&CentY_R>50&CentX_R<(cols-50)&CentY_R<(rows-50));
Pixel_Value=ismember(L, idx);
Pixel_Value_M=ismember(L_M, idx_M);
Pixel_Value_R=ismember(L_R, idx_R);

[L num]=bwlabel(Pixel_Value, 8);
[L_M num_M]=bwlabel(Pixel_Value_M, 8);
[L_R num_R]=bwlabel(Pixel_Value_R, 8);

feats=imfeature(L, 'Area', 'Centroid', 'BoundingBox', 'EquivDiameter', 8);
feats_M=imfeature(L_M, 'Area', 'Centroid', 'BoundingBox', 'EquivDiameter', 8);
feats_R=imfeature(L_R, 'Area', 'Centroid', 'BoundingBox', 'EquivDiameter', 8);

A_SKO_Im_M(i_Im)=255.*sqrt(sum(sum((Im_Frame(Boun+1:M-Boun, Boun+1:N-Boun)-
Im_Frame_M(Boun+1:M-Boun, Boun+1:N-Boun)).^2))/((M-2*Boun)*(N-2*Boun)));
A_SKO_Im_R(i_Im)=255.*sqrt(sum(sum((Im_Frame(Boun+1:M-Boun, Boun+1:N-Boun)-
Im_Frame_R(Boun+1:M-Boun, Boun+1:N-Boun)).^2))/((M-2*Boun)*(N-2*Boun)));

A_SNR_dB_Im_M(i_Im)=10*log10(sum(sum(Im_Frame(Boun+1:M-Boun, Boun+1:N-
Boun).^2))/sum(sum((Im_Frame(Boun+1:M-Boun, Boun+1:N-Boun)-Im_Frame_M(Boun+1:M-
Boun, Boun+1:N-Boun)).^2)));
A_SNR_dB_Im_R(i_Im)=10*log10(sum(sum(Im_Frame(Boun+1:M-Boun, Boun+1:N-
Boun).^2))/sum(sum((Im_Frame(Boun+1:M-Boun, Boun+1:N-Boun)-Im_Frame_R(Boun+1:M-
Boun, Boun+1:N-Boun)).^2)));

A_PSNR_dB_Im_M(i_Im)=10*log10(1.0*(M-2*Boun)*(N-2*Boun)/sum(sum((Im_Frame(Boun+1:M-
Boun, Boun+1:N-Boun)-Im_Frame_M(Boun+1:M-Boun, Boun+1:N-Boun)).^2)));
A_PSNR_dB_Im_R(i_Im)=10*log10(1.0*(M-2*Boun)*(N-2*Boun)/sum(sum((Im_Frame(Boun+1:M-
Boun, Boun+1:N-Boun)-Im_Frame_R(Boun+1:M-Boun, Boun+1:N-Boun)).^2)));

```

					<i>МММТ.420.004.004 – 3ЛІ</i>	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

S_Figure(i_Im)=feats.Area;
Centroid_X(i_Im)=feats.Centroid(1); Centroid_Y(i_Im)=feats.Centroid(2);
W_Figure(i_Im)=feats.BoundingBox(3); H_Figure(i_Im)=feats.BoundingBox(4);
S_Figure_M(i_Im)=feats_M.Area;
Centroid_X_M(i_Im)=feats_M.Centroid(1); Centroid_Y_M(i_Im)=feats_M.Centroid(2);
W_Figure_M(i_Im)=feats_M.BoundingBox(3); H_Figure_M(i_Im)=feats_M.BoundingBox(4);
S_Figure_R(i_Im)=feats_R.Area;
Centroid_X_R(i_Im)=feats_R.Centroid(1); Centroid_Y_R(i_Im)=feats_R.Centroid(2);
W_Figure_R(i_Im)=feats_R.BoundingBox(3); H_Figure_R(i_Im)=feats_R.BoundingBox(4);
ED_Figure(i_Im)=feats.EquivDiameter; ED_Figure_M(i_Im)=feats_M.EquivDiameter;
ED_Figure_R(i_Im)=feats_R.EquivDiameter;

B_M=xor(Pixel_Value, Pixel_Value_M);
B_R=xor(Pixel_Value, Pixel_Value_R);
if Im_Show==1
    figure; subplot(2, 3, 1); imshow(Pixel_Value); title('Segm. pochatkovogo
zobrazhennya');
    subplot(2, 3, 2); imshow(Pixel_Value_M); title('Segm. zobrazhennya z
vykryvlennyamy');
    subplot(2, 3, 5); imshow(Pixel_Value_R); title('Segm. filtr. zobrazhennya');
    subplot(2, 3, 3); imshow(B_M); title('Vidminnist pochatkovogo i vykr.');
```

```

    subplot(2, 3, 6); imshow(B_R); title('Vidminnist pochatkovogo i filtr.');
```

```

end;
B_M_XOR(i_Im)=sum(sum(B_M(Boun+1:M-Boun,Boun+1:N-Boun)));
B_R_XOR(i_Im)=sum(sum(B_R(Boun+1:M-Boun,Boun+1:N-Boun)));

if Im_Write==1
    Name_Frame=strcat(Dir_Test, 'Test_', num2str(i_Im, '%d'), '.bmp');
    imwrite(Im_Frame, Name_Frame);
    Name_Frame=strcat(Dir_Test, 'Test_Noise_', num2str(i_Im, '%d'), '.bmp');
    imwrite(Im_Frame_M, Name_Frame);
    Name_Frame=strcat(Dir_Test, 'Filtered_', num2str(i_Im, '%d'), '.bmp');
    imwrite(Im_Frame_R, Name_Frame);
    Name_Frame=strcat(Dir_Test, 'Segm_Test_', num2str(i_Im, '%d'), '.bmp');
    imwrite(Pixel_Value, Name_Frame);
    Name_Frame=strcat(Dir_Test, 'Segm_Test_Noise_', num2str(i_Im, '%d'), '.bmp');
    imwrite(Pixel_Value_M, Name_Frame);
    Name_Frame=strcat(Dir_Test, 'Segm_Filtered_', num2str(i_Im, '%d'), '.bmp');
    imwrite(Pixel_Value_R, Name_Frame);
end;

if Im_Show==1
    figure; subplot(2, 2, 1); imshow(Im_Frame); title('Pochatkove zobrazhennya');
    subplot(2, 2, 2); imshow(Im_Frame_M); title('Zobrazhennya z vykryvlennyamy');
    subplot(2, 2, 3); imshow(Im_Frame_R); title('Filtrovane zobrazhennya');
    subplot(2, 2, 4); imshow(Pixel_Value_R); title('Segm. filtr. zobrazhennya');
```

```

end;

end; % for i_Im=1:N_Im
close(h_waitbar);

if Delta_Show==1

    Max_A_SKO_Im_M=max(A_SKO_Im_M); Max_A_SKO_Im_R=max(A_SKO_Im_R);
    Mean_A_SKO_Im_M=mean(A_SKO_Im_M); Mean_A_SKO_Im_R=mean(A_SKO_Im_R);
    Min_A_SKO_Im_M=min(A_SKO_Im_M); Min_A_SKO_Im_R=min(A_SKO_Im_R);
    Delta_Dov_M=sort(A_SKO_Im_M, 'descend'); Delta_Dov_R=sort(A_SKO_Im_R, 'descend');
    MaxDov_A_SKO_Im_M=Delta_Dov_M(1+round(N_Im*(1-PDov)));
    MaxDov_A_SKO_Im_R=Delta_Dov_R(1+round(N_Im*(1-PDov)));

```

						Арк.
					<i>МММТ.420.004.004 – 3Л1</i>	10
Змн.	Арк.	№ докум.	Підпис	Дата		



```

Max_A_SNR_dB_Im_M=max(A_SNR_dB_Im_M); Max_A_SNR_dB_Im_R=max(A_SNR_dB_Im_R);
Mean_A_SNR_dB_Im_M=mean(A_SNR_dB_Im_M); Mean_A_SNR_dB_Im_R=mean(A_SNR_dB_Im_R);
Min_A_SNR_dB_Im_M=min(A_SNR_dB_Im_M); Min_A_SNR_dB_Im_R=min(A_SNR_dB_Im_R);
Delta_Dov_M=sort(A_SNR_dB_Im_M, 'descend'); Delta_Dov_R=sort(A_SNR_dB_Im_R,
'descend');
MaxDov_A_SNR_dB_Im_M=Delta_Dov_M(1+round(N_Im*(1-PDov)));
MaxDov_A_SNR_dB_Im_R=Delta_Dov_R(1+round(N_Im*(1-PDov)));

Max_A_PSNR_dB_Im_M=max(A_PSNR_dB_Im_M); Max_A_PSNR_dB_Im_R=max(A_PSNR_dB_Im_R);
Mean_A_PSNR_dB_Im_M=mean(A_PSNR_dB_Im_M); Mean_A_PSNR_dB_Im_R=mean(A_PSNR_dB_Im_R);
Min_A_PSNR_dB_Im_M=min(A_PSNR_dB_Im_M); Min_A_PSNR_dB_Im_R=min(A_PSNR_dB_Im_R);
Delta_Dov_M=sort(A_PSNR_dB_Im_M, 'descend'); Delta_Dov_R=sort(A_PSNR_dB_Im_R,
'descend');
MaxDov_A_PSNR_dB_Im_M=Delta_Dov_M(1+round(N_Im*(1-PDov)));
MaxDov_A_PSNR_dB_Im_R=Delta_Dov_R(1+round(N_Im*(1-PDov)));

Delta_S_Figure_M=(S_Figure_M-S_Figure);
D1_S_Figure_M=Delta_S_Figure_M.*100./S_Figure;
Delta_S_Figure_R=(S_Figure_R-S_Figure);
D1_S_Figure_R=Delta_S_Figure_R.*100./S_Figure;
Delta_Centroid_X_M=Centroid_X_M-Centroid_X;
D1_Centroid_X_M=Delta_Centroid_X_M.*100./Centroid_X;
Delta_Centroid_X_R=Centroid_X_R-Centroid_X;
D1_Centroid_X_R=Delta_Centroid_X_R.*100./Centroid_X;
Delta_Centroid_Y_M=Centroid_Y_M-Centroid_Y;
D1_Centroid_Y_M=Delta_Centroid_Y_M.*100./Centroid_Y;
Delta_Centroid_Y_R=Centroid_Y_R-Centroid_Y;
D1_Centroid_Y_R=Delta_Centroid_Y_R.*100./Centroid_Y;

Max_Delta_S_Figure_M=max(abs(Delta_S_Figure_M));
Max_Delta_S_Figure_R=max(abs(Delta_S_Figure_R));
Delta_Dov_M=sort(abs(Delta_S_Figure_M), 'descend');
Delta_Dov_R=sort(abs(Delta_S_Figure_R), 'descend');
MaxDov_Delta_S_Figure_M=Delta_Dov_M(1+round(N_Im*(1-PDov)));
MaxDov_Delta_S_Figure_R=Delta_Dov_R(1+round(N_Im*(1-PDov)));
Mean_Delta_S_Figure_M=mean(Delta_S_Figure_M);
Mean_Delta_S_Figure_R=mean(Delta_S_Figure_R);
Std_Delta_S_Figure_M=std(Delta_S_Figure_M,1);
Std_Delta_S_Figure_R=std(Delta_S_Figure_R,1);

Max_Delta_Centroid_M=max(abs([Delta_Centroid_X_M; Delta_Centroid_Y_M]));
Max_Delta_Centroid_R=max(abs([Delta_Centroid_X_R; Delta_Centroid_Y_R]));
Delta_Dov_M=sort(abs([Delta_Centroid_X_M; Delta_Centroid_Y_M]), 'descend');
Delta_Dov_R=sort(abs([Delta_Centroid_X_R; Delta_Centroid_Y_R]), 'descend');
MaxDov_Delta_Centroid_M=Delta_Dov_M(1+round(N_Im*(1-PDov)));
MaxDov_Delta_Centroid_R=Delta_Dov_R(1+round(N_Im*(1-PDov)));
Mean_Delta_Centroid_M=mean([Delta_Centroid_X_M; Delta_Centroid_Y_M]);
Mean_Delta_Centroid_R=mean([Delta_Centroid_X_R; Delta_Centroid_Y_R]);
Std_Delta_Centroid_M=std([Delta_Centroid_X_M; Delta_Centroid_Y_M],1);
Std_Delta_Centroid_R=std([Delta_Centroid_X_R; Delta_Centroid_Y_R]);

Max_D1_S_Figure_M=max(abs(D1_S_Figure_M));
Max_D1_S_Figure_R=max(abs(D1_S_Figure_R));
Delta_Dov_M=sort(abs(D1_S_Figure_M), 'descend');
Delta_Dov_R=sort(abs(D1_S_Figure_R), 'descend');
MaxDov_D1_S_Figure_M=Delta_Dov_M(1+round(N_Im*(1-PDov)));
MaxDov_D1_S_Figure_R=Delta_Dov_R(1+round(N_Im*(1-PDov)));
Mean_D1_S_Figure_M=mean(D1_S_Figure_M); Mean_D1_S_Figure_R=mean(D1_S_Figure_R);
Std_D1_S_Figure_M=std(D1_S_Figure_M,1); Std_D1_S_Figure_R=std(D1_S_Figure_R,1);

```

Змн.	Арк.	№ докум.	Підпис	Дата

МММТ.420.004.004 – 3Л1

Арк.

11

```

Max_D1_Centroid_M=max(abs([D1_Centroid_X_M; D1_Centroid_Y_M]));
Max_D1_Centroid_R=max(abs([D1_Centroid_X_R; D1_Centroid_Y_R]));
Delta_Dov_M=sort(abs([D1_Centroid_X_M; D1_Centroid_Y_M]), 'descend');
Delta_Dov_R=sort(abs([D1_Centroid_X_R; D1_Centroid_Y_R]), 'descend');
MaxDov_D1_Centroid_M=Delta_Dov_M(1+round(N_Im*(1-PDov)));
MaxDov_D1_Centroid_R=Delta_Dov_R(1+round(N_Im*(1-PDov)));
Mean_D1_Centroid_M=mean([D1_Centroid_X_M; D1_Centroid_Y_M]);
Mean_D1_Centroid_R=mean([D1_Centroid_X_R; D1_Centroid_Y_R]);
Std_D1_Centroid_M=std([D1_Centroid_X_M; D1_Centroid_Y_M],1);
Std_D1_Centroid_R=std([D1_Centroid_X_R; D1_Centroid_Y_R]);

Delta_W_Figure_M=(W_Figure_M-W_Figure);
D1_W_Figure_M=Delta_W_Figure_M.*100./W_Figure;
Delta_W_Figure_R=(W_Figure_R-W_Figure);
D1_W_Figure_R=Delta_W_Figure_R.*100./W_Figure;
Delta_H_Figure_M=(H_Figure_M-H_Figure);
D1_H_Figure_M=Delta_H_Figure_M.*100./H_Figure;
Delta_H_Figure_R=(H_Figure_R-H_Figure);
D1_H_Figure_R=Delta_H_Figure_R.*100./H_Figure;

Max_Delta_L_Figure_M=max(abs([Delta_W_Figure_M; Delta_H_Figure_M]));
Max_Delta_L_Figure_R=max(abs([Delta_W_Figure_R; Delta_H_Figure_R]));
Delta_Dov_M=sort(abs([Delta_W_Figure_M; Delta_H_Figure_M]), 'descend');
Delta_Dov_R=sort(abs([Delta_W_Figure_R; Delta_H_Figure_R]), 'descend');
MaxDov_Delta_L_Figure_M=Delta_Dov_M(1+round(N_Im*(1-PDov)));
MaxDov_Delta_L_Figure_R=Delta_Dov_R(1+round(N_Im*(1-PDov)));
Mean_Delta_L_Figure_M=mean([Delta_W_Figure_M; Delta_H_Figure_M]);
Mean_Delta_L_Figure_R=mean([Delta_W_Figure_R; Delta_H_Figure_R]);
Std_Delta_L_Figure_M=std([Delta_W_Figure_M; Delta_H_Figure_M],1);
Std_Delta_L_Figure_R=std([Delta_W_Figure_R; Delta_H_Figure_R],1);

Max_D1_L_Figure_M=max(abs([D1_W_Figure_M; D1_H_Figure_M]));
Max_D1_L_Figure_R=max(abs([D1_W_Figure_R; D1_H_Figure_R]));
Delta_Dov_M=sort(abs([D1_W_Figure_M; D1_H_Figure_M]), 'descend');
Delta_Dov_R=sort(abs([D1_W_Figure_R; D1_H_Figure_R]), 'descend');
MaxDov_D1_L_Figure_M=Delta_Dov_M(1+round(N_Im*(1-PDov)));
MaxDov_D1_L_Figure_R=Delta_Dov_R(1+round(N_Im*(1-PDov)));
Mean_D1_L_Figure_M=mean([D1_W_Figure_M; D1_H_Figure_M]);
Mean_D1_L_Figure_R=mean([D1_W_Figure_R; D1_H_Figure_R]);
Std_D1_L_Figure_M=std([D1_W_Figure_M; D1_H_Figure_M],1);
Std_D1_L_Figure_R=std([D1_W_Figure_R; D1_H_Figure_R],1);

B_M_XOR=B_M_XOR./(ED_Figure.*pi); B_R_XOR=B_R_XOR./(ED_Figure.*pi);
Max_B_M_XOR=max(abs(B_M_XOR)); Max_B_R_XOR=max(abs(B_R_XOR));
Delta_Dov_M=sort(abs(B_M_XOR), 'descend'); Delta_Dov_R=sort(abs(B_R_XOR),
'descend');
MaxDov_B_M_XOR=Delta_Dov_M(1+round(N_Im*(1-PDov)));
MaxDov_B_R_XOR=Delta_Dov_R(1+round(N_Im*(1-PDov)));
Mean_B_M_XOR=mean(B_M_XOR); Mean_B_R_XOR=mean(B_R_XOR);
Min_B_M_XOR=min(B_M_XOR); Min_B_R_XOR=min(B_R_XOR);

end;

```

Змн.	Арк.	№ докум.	Підпис	Дата

МММТ.420.004.004 – 3Л1

Арк.

12

1.2.2. Виконуємо програму та отримаємо розрахунки фільтрації зображення.

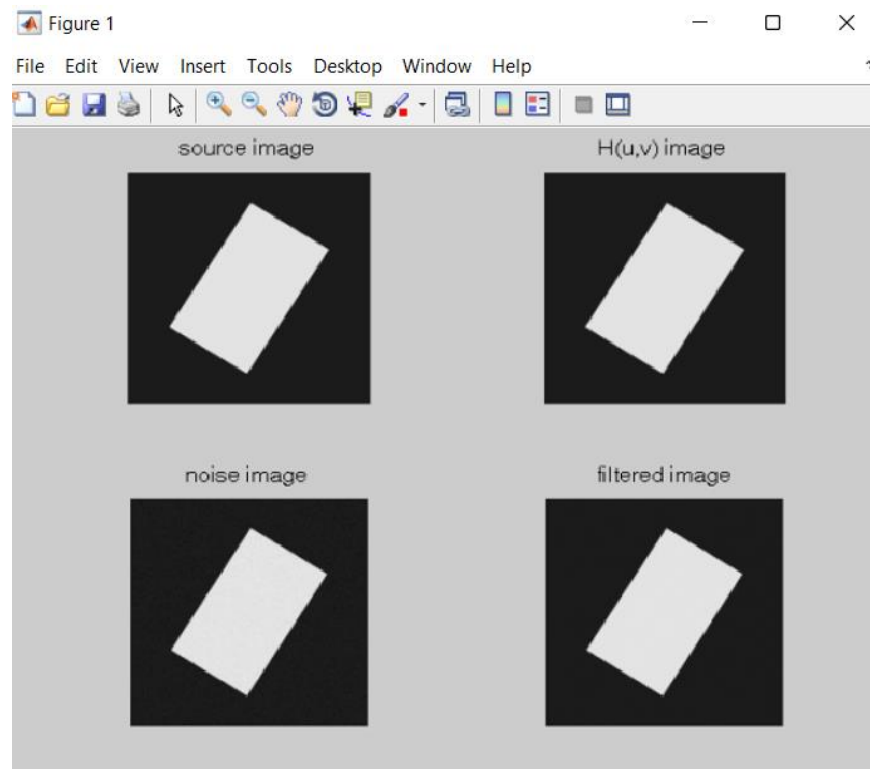


Рис. 1.1. Результат виконання програми

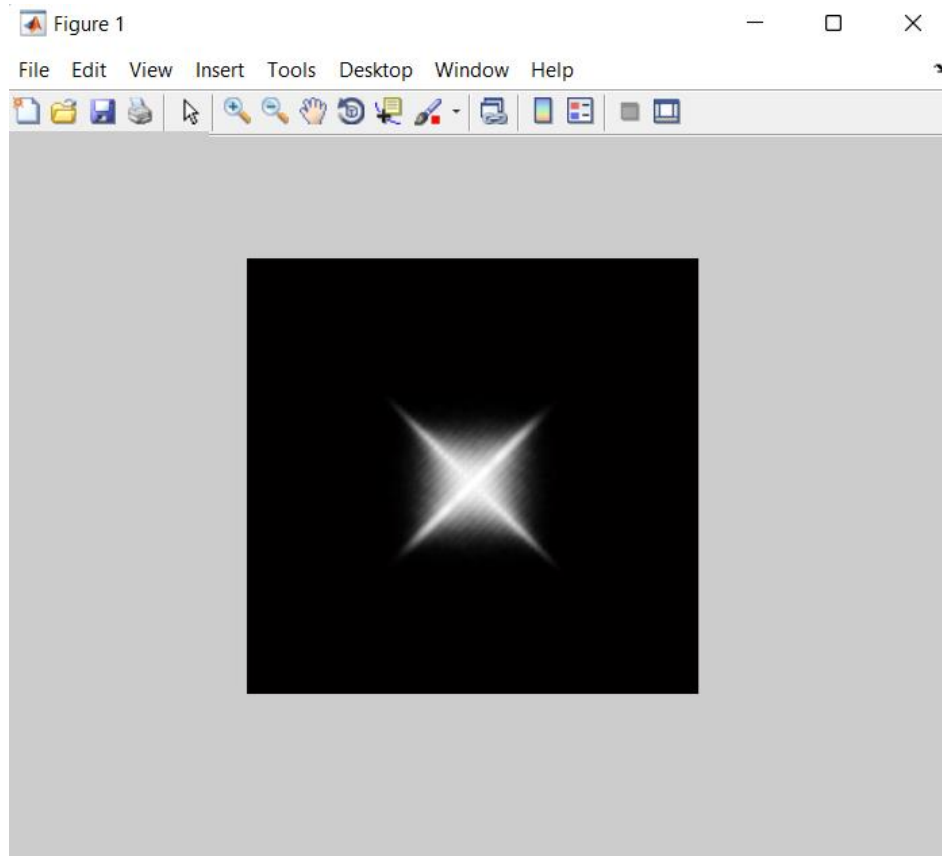


Рис. 1.2. Спектр вихідного зображення

Змн.	Арк.	№ докум.	Підпис	Дата

МММТ.420.004.004 – ЗЛІ

Арк.

13

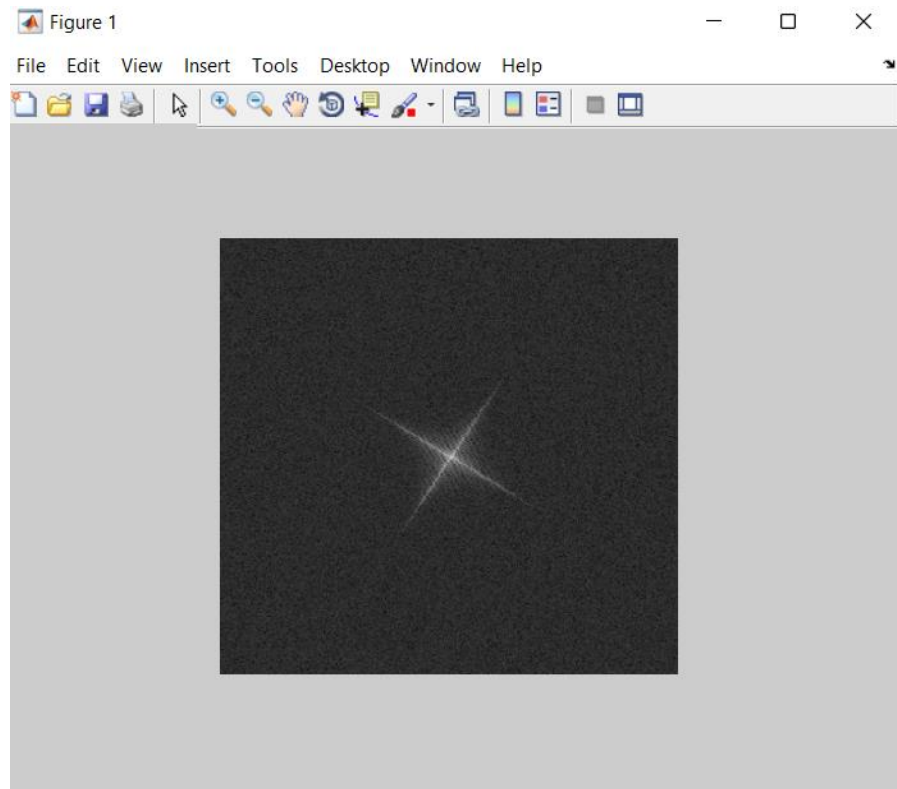


Рис. 1.4. Спектр зображення з шумом

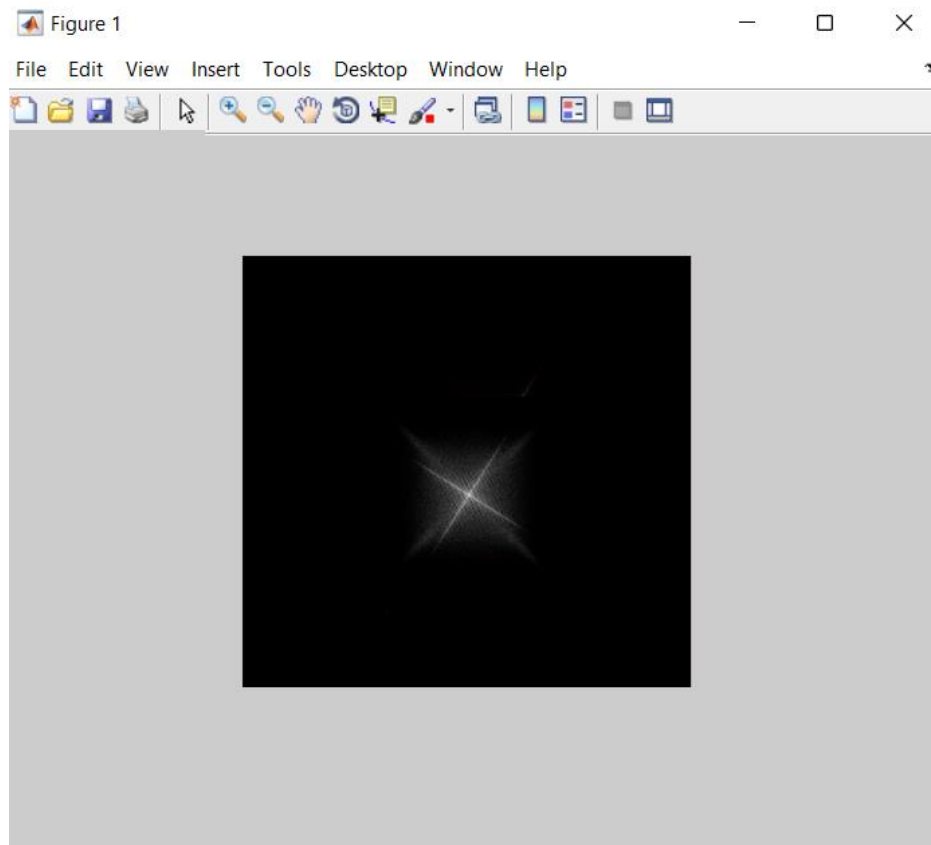


Рис. 1.5. Спектр зображення після фільтрації

### 1.2.3. Розрахунки програмою MATLAB.

Розмір зображення рядка та стовпці д.т. 1024 x 1024  
Фігура 1 – прямокутник  
Кількість кадрів 30  
ширина погр кривої (перепадку яскравості в рядку) д.т. 12  
розмір маски фільтра д.т. 3  
вів змаза під час руху відеокамери <=20 д.т. 0  
кут змаза під час руху відеокамери град. 0  
сигнал шум у дБ 40  
Ампл погріш без фільт після фільт  
СКО серед д.у. 2.550 1.440  
SNR серед дБ 28.518 33.842  
PSNR серед дБ 40.001 45.325  
Параметри похибки геометричних ознак д.т.  
Погріш без фільт, д.т. площа% центр x центр y ширина висота  
Максимальна 0.031 0.017 0.030 2.000 1.000  
Макс дов 0.021 0.017 0.022 1.000 1.000  
Серед зн -0.001 0.001 -0.001 0.067 0.067  
СКО 0.010 0.008 0.010 0.573 0.249  
Погріш непадання контуру без фільт після фільт  
макс д.т./довжину контуру 0.088 0.057  
макс дов д.т./довжину конт 0.077 0.054  
серед д.т./довжину контуру 0.060 0.037  
СКО д.т./довжину контуру 0.010 0.010

**Висновок:** на данній лабораторній роботі було отримано навички обробки випадкових похибок інформаційних сигналів у вимірювальному каналі за допомогою програми MATLAB.

					<i>МММТ.420.004.004 – ЗЛІ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15