PRÁCTICA 6 RESUELTA + PLOTS: mti

Radiolocalització (Universitat Politècnica de Catalunya)

PRÁCTICA 6

MAIN_2

```
clear all;
clear all figures;

%% 7.1 Samples at the output of a detector: Gaussian white noise
%vector of noise
%--> i n p u t s   a j u s t a b l e s
N_samples=10001;
rep=100;
PRF=1*10^3 ;%kHz
pot_noise=1; %1W
mean_=0;

%--> e x e c u c i ó
sample_factor_noise=randn(1,N_samples)+mean_;
Pot_n=sum(abs(sample_factor_noise).^2)/N_samples;
noise=sqrt(pot_noise).*sample_factor_noise./sqrt(Pot_n);
figure(1);
plot(abs(fft(noise)));
figure(2);
histogram(noise,'Normalization','pdf');
xlabel('Noise');
ylabel('PDF of noise');

%matrix of noise
 %--> i n p u t s   a j u s t a b l e s
N_samples=10001;
PRF=1*10^3 ;%kHz
pot_noise=1; %1W
rep=100;
mean_=0;

%--> e x e c u c i ó
sample_noise=randn(rep,N_samples)+mean_;
for i=1:rep
Pot=sum(abs(sample_noise(i,:)).^2)/N_samples;
noise_2(i,:)=sqrt(pot_noise)*sample_noise(i,:)./sqrt(Pot);
end

noise_fft=abs(fft(noise_2,[],2));
noise_mean=mean(noise_fft(:,(1:N_samples)));
figure(3);
freq=(0:N_samples-1)/N_samples*PRF;
plot(noise_mean);
figure(4);
histogram(noise_mean,'Normalization','pdf');
xlabel('Noise');
ylabel('PDF of noise');
```

MAIN

```
clear all;
clear all figures;
%% 7.2 Single canceller
%--> i n p u t s   a j u s t a b l e s
N_samples=10001;
repetitions=100;
PRF=1*10^3 ;%kHz
```

```
pot_noise=1; %1W
mean_=0;

%--> e x e c u c i ó
sample_noise=randn(repetitions,N_samples)+mean_;

for i=1:repetitions
Pot=sum(abs(sample_noise(i,:)).^2)/N_samples;
noise_2(i,:)=sqrt(pot_noise)*sample_noise(i,:)./sqrt(Pot);
for j=2:N_samples
    MTI_filter_single(i,j-1)=noise_2(i,j)-noise_2(i,j-1);
end
end
MTI_filter_single_meanvalues=mean(MTI_filter_single(:,(1:length(MTI_fi
lter_single))));
MTI_fft_single=abs(fft(MTI_filter_single,[],2));
%MTI_mean_single1=sum(MTI_fft_single(:,(1:length(MTI_fft_single))))/re
p;
MTI_mean_single=mean(MTI_fft_single(:,(1:length(MTI_fft_single))));
MTI_mean_max_single=max(MTI_mean_single);

noise_meanvalues=mean(noise_2(:,(1:length(noise_2))));
noise_fft=abs(fft(noise_2,[],2));
noise_mean=mean(noise_fft(:,(1:N_samples)));

%% 7.2 PLOTS
%SAMPLES PLOTS
    %PLOTTING INPUT
figure(1);
subplot(1,2,1);
plot(noise_meanvalues);
xlabel('Number of Samples');
ylabel('Noise (V)');
title('MTI input for single canceller');

    %PLOTTING OUTPUT
subplot(1,2,2);
plot(MTI_filter_single_meanvalues);
xlabel('Number of Samples');
ylabel('Noise (V)');
title('MTI output for single canceller');

%SPECTRUM PLOTS
    %PLOTTING INPUT
figure(2);
subplot(1,2,1);
freq=(0:N_samples-1)/N_samples*PRF;
plot(freq,noise_mean);
xlabel('Frequency (Hz)');
ylabel('Noise (V)');
title('Spectrum MTI input for single canceller');

    %PLOTTING OUTPUT
subplot(1,2,2);
freq=(0:length(MTI_mean_single)-1)/N_samples*PRF;
plot(freq, MTI_mean_single);
xlabel('Frequency (Hz)');
ylabel('Noise (V)');
title('Spectrum MTI output for single canceller');

%% 7.3 Improving single canceller
```

```
%--> i n p u t s   a j u s t a b l e s
N_samples=10001;
repetitions=1000;
PRF=1*10^3 ;%kHz
pot_noise=1; %1W
mean_=0;

%--> e x e c u c i ó
sample_noise=randn(repetitions,N_samples)+mean_;

for i=1:repetitions
Pot=sum(abs(sample_noise(i,:)).^2)/N_samples;
noise_2(i,:)=sqrt(pot_noise)*sample_noise(i,:)./sqrt(Pot);
for j=2:N_samples
    MTI_filter_single(i,j-1)=noise_2(i,j)-noise_2(i,j-1);
end
end
MTI_filter_single_meanvalues=mean(MTI_filter_single(:,(1:length(MTI_fi
lter_single))));
MTI_fft_single=abs(fft(MTI_filter_single,[],2));
%MTI_mean_single1=sum(MTI_fft_single(:,(1:length(MTI_fft_single))))/re
p;
MTI_mean_single=mean(MTI_fft_single(:,(1:length(MTI_fft_single))));
MTI_mean_max_single=max(MTI_mean_single);

noise_meanvalues=mean(noise_2(:,(1:length(noise_2))));
noise_fft=abs(fft(noise_2,[],2));
noise_mean=mean(noise_fft(:,(1:N_samples)));

%% 7.3 PLOTS
%SPECTRUM PLOTS
    %PLOTTING INPUT
figure(3);
subplot(1,2,1);
freq=(0:N_samples-1)/N_samples*PRF;
plot(freq,noise_mean);
xlabel('Frequency (Hz)');
ylabel('Noise (V)');
title('Spectrum MTI input for single canceller');

    %PLOTTING OUTPUT
subplot(1,2,2);
freq=(0:length(MTI_mean_single)-1)/N_samples*PRF;
plot(freq, MTI_mean_single/MTI_mean_max_single);
xlabel('Frequency (Hz)');
ylabel('Noise (V)');
title('Spectrum MTI output for single canceller');
hold on;
y=2*abs(sin(1/PRF*pi*freq))/2;
plot(freq, y);
legend('Experimental Output', 'Theoretical Output');
hold off;

%% 7.4 Double canceller
%--> i n p u t s   a j u s t a b l e s
N_samples=10001;
repetitions=1000;
PRF=1*10^3 ;%kHz
pot_noise=1; %1W
mean_=0;
```

```
%--> e x e c u c i ó
sample_noise=randn(repetitions,N_samples)+mean_;

for i=1:repetitions
Pot=sum(abs(sample_noise(i,:)).^2)/N_samples;
noise_2(i,:)=sqrt(pot_noise)*sample_noise(i,:)./sqrt(Pot);
for j=3:N_samples
    MTI_filter_double(i,j-2)=noise_2(i,j)-2*noise_2(i,j-
1)+noise_2(i,j-2);
end
end

MTI_filter_double_meanvalues=mean(MTI_filter_double(:,(1:length(MTI_fi
lter_double))));
MTI_fft_double=abs(fft(MTI_filter_double,[],2));
MTI_mean_double=mean(MTI_fft_double(:,(1:length(MTI_fft_double))));
MTI_mean_max_double=max(MTI_mean_double);

noise_meanvalues_double=mean(noise_2(:,(1:length(noise_2))));
noise_fft_double=abs(fft(noise_2,[],2));
noise_mean_double=mean(noise_fft_double(:,(1:N_samples)));

%% 7.4 PLOTS
%SAMPLES PLOTS
figure(4);
subplot(1,2,1);
plot(noise_meanvalues_double);
xlabel('Number of Samples');
ylabel('Noise (V)');
title('MTI input for double canceller');

    %PLOTTING OUTPUT
subplot(1,2,2);
plot(MTI_filter_double_meanvalues);
xlabel('Number of Samples');
ylabel('Noise (V)');
title('MTI output for double canceller');

% SPECTRUM PLOTS
    %PLOTTING INPUT
figure(5);
subplot(1,2,1);
freq=(0:N_samples-1)/N_samples*PRF;
plot(freq,noise_mean_double);
xlabel('Frequency (Hz)');
ylabel('Noise (V)');
title('Spectrum MTI input for double canceller');

    %PLOTTING OUTPUT
subplot(1,2,2);
freq=(0:length(MTI_mean_double)-1)/N_samples*PRF;
plot(freq, MTI_mean_double/MTI_mean_max_double);
hold on;
y=4/4*abs(sin(1/PRF*pi*freq)).^2;
plot(freq, y);
hold off;
xlabel('Frequency (Hz)');
ylabel('Noise (V)');
title('Spectrum MTI output for double canceller');
legend('Experimental Output', 'Theoretical Output');
```