L.EEC025 - FUNDAMENTALS OF SIGNAL PROCESSING

Academic year 2022-2023, week 9
P2P exercises

Topics: Peer-to-peer learning/assessment exercises on filtering complex exponentials assessing the SNR before and after filtering

Peer-to-peer learning/assessment exercises (P2P L/A), valid for the week of November 14

NOTE: this week (November 14), Student "E3" of each group should explain P2P exercise 1, and Student "E4" of each group should explain P2P exercise 2. Detailed information on the P2P procedure is available on the dynamic email sent on Sept 22, 2022.

P2P Exercise 1

In this exercise, we evaluate how the frequency response of a discrete-time system affects (from system input to system output) an infinite-length real-valued sine (or co-sine) function, and validate numerically that evaluation in Matlab.

In a recent lecture, we have shown that complex exponentials are the eigenfunctions of linear and shift-invariant (LSI) discrete-time systems. That is, if an LSI discrete-time system is characterized by the impulse response $h[n]$, and is excited by the input sequence $x[n] = e^{jn\omega_0}$, then the output sequence is $y[n] = H(e^{j\omega_0})e^{jn\omega_0}$, where $H(e^{j\omega_0}) = \sum_{n=-\infty}^{+\infty} h[n]\, e^{-jn\omega}\Big|_{\omega=\omega_0}$ is the frequency response of the LSI system when it is evaluated for $\omega = \omega_0$.

a) Show that, in general terms, if $x[n] = \sin(n\omega_0)$, or if $x[n] = \cos(n\omega_0)$, and if $h[n]$ is real-valued (i.e. if $h[n]$ is complex-valued then the following is ***not*** true), then $y[n] = \left|H(e^{j\omega_0})\right|\sin\left(n\omega_0 + \angle H(e^{j\omega_0})\right)$, or $y[n] = \left|H(e^{j\omega_0})\right|\cos\left(n\omega_0 + \angle H(e^{j\omega_0})\right)$. These results presume that we write $H(e^{j\omega_0}) = \left|H(e^{j\omega_0})\right| e^{j\angle H(e^{j\omega_0})}$, where $\left|H(e^{j\omega_0})\right|$ represents the magnitude part of the frequency response of the system when it is evaluated at $\omega = \omega_0$, and $\angle H(e^{j\omega_0})$ represents the phase part of the frequency response of the system when it is evaluated at $\omega = \omega_0$.

P2P assessment: **3pt /5** if demonstration is clear and complete, and results are correct

b) We take as the LSI discrete-time system a causal second-order all-pole system that is governed by the difference equation $y[n] = x[n] + 2\alpha\cos(\beta)\, y[n-1] - \alpha^2 y[n-2]$, where α and β are as specified in the following Matlab code. We assume further that $\omega_0 =$ `OMEGA0= 0.7404`. Use the following Matlab code (which you need to complete) to find `MAG=`$\left|H(e^{j\omega_0})\right|$, and `PHI=`$\angle H(e^{j\omega_0})$.

```
ALFA=0.925; BETA=0.275*pi;
b = 1; a = please complete here ;
N=50; n=[0:N-1].';
OMEGA0= 0.7404;
[H, W]=freqz(b,a,[0 OMEGA0]);
MAG=abs(H(2));
PHI=angle(H(2));
```

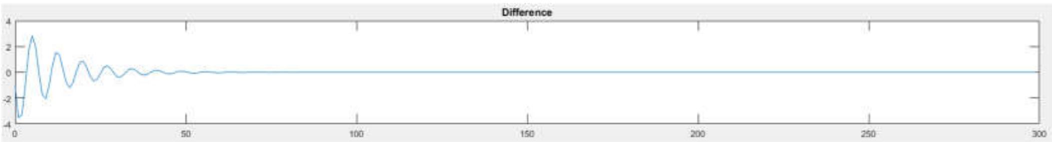
Note: the solution should be: `MAG = 5.1418` and `PHI 0.2159`

P2P assessment: **1pt /5** if explanation is clear and results are correct

c) Now, create a long sinusoidal sequence with 10^5 samples, obtain the (numerical) system output using the Matlab command `filter()`, program your analytical solution, and check the difference between numerical and analytical results using the following Matlab code:

```
N=1E5; n=[0:N-1];
xsine=sin(n*OMEGA0).'; % you may check that cos() also works
ysine=filter(b,a,xsine);
myysine= please complete here .'; % don't forget transposition .'
plotrange=[1:300];
figure(3)
subplot(4,1,1)
plot(n(plotrange), xsine(plotrange))
title('INPUT')
subplot(4,1,2)
plot(n(plotrange), ysine(plotrange))
title('Output (numerical)')
subplot(4,1,3)
plot(n(plotrange), myysine(plotrange))
title('Output (analytical)')
subplot(4,1,4)
plot(n(plotrange), ysine(plotrange)-myysine(plotrange))
     title('difference signal')
```

Note: you should obtain the following difference signal:



How do you explain to your Colleagues that the first few samples are not zero ?

P2P assessment: **1pt /5** if explanation is clear and results are correct

P2P Problem 2

In this problem, we show that if the input sine wave that is specified in P2P Problem 2 is contamined by noise, at a certain SNR, it comes out of the system with an improved SNR. We validate that conclusion numerically in Matlab, and find the theoretical gain in SNR.

a) Show that if the amplitudes of the samples of a random signal follow a uniform Probability Density Function (PDF) between $-v$ and $+v$, then its average power is given by $P_n = \frac{v^2}{3}$.

P2P assessment: **2pt /5** if explanation is clear and result is correct

b) Show that in case $x[n] = \sin(n\omega_0)$, and if the noise mentioned in **a)** is combined with $x[n]$, then the Signal to Noise Ratio (SNR), in dB, is given by $SNR = 10\log_{10}\frac{P_s}{P_n} = 10\log_{10}\frac{3}{2v^2}$.

P2P assessment: **1pt /5** if explanation is clear and result is correct

c) The following Matlab code (which is a continuation of the Matlab code in P2P exercise 1) generates noise whose samples' amplitudes follow a uniform PDF, shows how to combine the noise with the sine wave such that the resulting SNR is very approximately 10 dB, and shows that the auto-correlation function of the noise consists of an impulse.

```
SNR=10; MAXLAG=30;
v=sqrt(3)/(sqrt(2)*10.^(SNR/20));
xnoise=2*v*(rand(N,1)-0.5);
Ps=mean((abs(xsine)).^2);
Pn=mean((abs(xnoise)).^2);
10*log10(Ps/Pn)
% to confirm that rx[ell]=K*DELTA[ell]
[rx, lag]=xcorr(xnoise, MAXLAG);
rx=rx/length(xnoise); % average noise power
figure(4)
stem(lag, rx)
xlabel('$$\ell$$ (samples)','Interpreter', 'Latex');
ylabel('$$r_{X}[\ell]$$','Interpreter', 'Latex'); pause
```

Explain to your Colleagues:

- if the auto-correlation function is such that it consists of an impulse, how do we call this type of noise ? why ?
- the rationale that justifies how the above parameter "v" is obtained leading to a 10 dB SNR
- why running this code multiple times generates a practical SNR that is not exactly equal to, but is very close to, 10.0 dB.

P2P assessment: **1pt /5** if explanation is clear and complete (especially how "v" is obtained)

d) The following Matlab code (which is a continuation of the above Matlab code) allows to evaluate numerically the SNR of the signal after filtering:

```
ynoise=filter(b,a,xnoise);
Ps=mean((abs(ysine)).^2);
Pn=mean((abs(ynoise)).^2);
10*log10(Ps/Pn)
```

Running this piece of Matlab (plus the code in **a)**) reveals that the output SNR improves the input SNR by about 6 dB. How do you explain that to your Colleagues ?

P2P assessment: **1pt /5** if explanation is clear and complete

e) [**NOTE**: this last question, which is **not** mandatory, is more atypical than the previous ones, therefore, Student E4 should receive contributions/help/suggestions from the remaining group Students]

We state here, without proof, that if the impulse response of our LSI system is $h[n]$, and if the input auto-correlation is $r_x[\ell]$, then the output auto-correlation is given by $r_y[\ell] = h[\ell] * h^*[-\ell] * r_x[\ell]$. If we just consider the noise part of the input signal, it was shown in **a)** that $r_x[\ell] = K\delta[\ell]$, where K is the average power of the input noise.

(1) We have seen in **a)** that $K = P_{n\_inp} = \frac{v^2}{3}$, where v is the theoretical value that is specified in the above Matlab code.

(2) Find the theoretical value of the average power of the output noise $P_{n\_out} = r_y[0]$ where $r_y[0] = \sum_{n=-\infty}^{+\infty} |h[n]|^2 \ P_{n\_inp}$. This value may be conveniently computed using the Parseval theorem in the Z-domain. Remember that, in this case, in the Z-domain, the region of convergence consists of a ring, which means that when you apply the contour line integral and, therefore, the residue theorem, only two poles matter.

Note: the solution should be: $P_{n\_out} = \frac{1+\alpha^2}{1-\alpha^2} \cdot \frac{1}{1-2\alpha^2\cos(2\beta)+\alpha^4} P_{n\_inp}$

(3) Confirm that your theoretical output SNR is easily computed as:

```
Ps=(MAG^2)/2;
Pn=(v^2)/3;
Pn=(1+ALFA^2)/(1-ALFA^2)*1/(1-
     2*(ALFA^2)*cos(2*BETA)+ALFA^4)*Pn;
10*log10(Ps/Pn)
% = 16.1422   (and the input SNR is 10 dB)
```

Extra P2P Problem for all interested

In this problem, we look at the PDF of all signals of interest in these set of P2P problems. Use the following Matlab code to check the PDF of several signals:

```
[H X]=hist(x,50); equalize=50/(max(x)-min(x));
bar(X, H/sum(H)*equalize, 0.5);
ylabel('PDF'); xlabel('x[n] amplitude'); pause
```

run this piece of code after setting x to different alternatives :

```
x=xnoise;
x=ynoise;
x=xsine;
x=ysine;
x=xnoise+xsine;
x=ynoise+ysine;
```

Can you anticipate what cases correspond to the following two plots ? And how do you explain their different shapes ?