ECE 421 Introduction to Signal Processing

Final – Spring 2018

May 4, 2018

Please remember to justify your answers carefully.

There are 9 pages in total on this test.

Last name: ________________ First name: ________________

Student ID: ________________

Question 1 (warmup question)

This question includes two unrelated parts. In both of them, we consider the sequence

$$x(n) = 2\delta(n+1) + 3\delta(n-1).$$

a) Determine the autocorrelation sequence, $r_{xx}(l)$, that corresponds to $x(n)$.

Solution:

$r_{xx}(l) = \sum_n x(n)x(n-l)$
$r_{xx}(-2) = 3\delta(-1) \cdot 2\delta(-1) = 6$
$r_{xx}(0) = 13$
$r_{xx}(2) = 2\delta(1) \cdot 3\delta(1) = 6$
And all other terms are 0's.

b) Compute the Fourier transform, $X(\omega)$, that corresponds to $x(n)$.

Solution:

Since $\mathcal{F}(\delta(n+1)) = e^{2\pi fj} = e^{j\omega}$, $X(\omega) = 2e^{j\omega} + 3e^{-j\omega}$.

Question 2 (discrete Fourier transform)
Consider the finite duration discrete time signal, $x(n) = [\underline{1}, 3, 1, 3]$.

a) Compute the 4-point discrete Fourier transform (DFT), $X(k)$, of the signal $x(n)$.

Solution:

$X(k) = W_4 \cdot x(n)$, where

$$W_4 = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}.$$

More explicitly, the element at $l-$th row and $m-$th column is $e^{2\pi j/4\cdot(l-1)\cdot(m-1)}$.
$X(k) = [2, 0, -1, 0]$.
If you follow the convention used in MATLAB, 1/4 is not specified in the DFT. Thus $X(k) = [8, 0, -4, 0]$ will also be accepted. In this case, 1/4 will be specified in IDFT.

b) The signal is periodic. In this part, you will compute the DFT of one of the periods. To do so, (*i*) specify the period, k; (*ii*) specify one period of the signal, i.e., $\widetilde{x} = x(0), \ldots, x(k-1)$; and (*iii*) compute the k-point DFT of $\widetilde{x}$.

Solution:

$k = 2$, $\widetilde{x} = [1, 3]$;

$\widetilde{X}(k) = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \widetilde{x}(n)$, thus $\widetilde{X}(k) = [2, -1]$.
$\widetilde{X}(k) = [4, -2]$ will be accepted.

c) Write down MATLAB code that calculates the answers to parts a and b.
Solution:

```
x=[ 1 3 1 3];
Xk=fft(x);
xt=[ 1 3];
Xtk=fft(xt);
```

Question 3 (filter design using bilinear transformation)
An analog (continuous time, continuous amplitude) filter is proposed with the following transfer function in the Laplace domain,

$$H(s) = \frac{s}{s+3}.$$

Your goal in this question will be to map this filter to a discrete time filter.

a) Select an appropriate sampling parameter T to ensure that a bilinear transformation can map this analog filter to a digital one where the analog frequency $\Omega = 3$ in the denominator (i.e., $s+3$) corresponds to digital frequency $\omega = \frac{\pi}{2}$.

Solution:

$\Omega = 3, \omega = \frac{\pi}{2}$,
$\frac{\pi}{2} = 2atan\left(\frac{3T}{2}\right)$,
$T = \frac{2}{3}$.

b) Determine an expression for the transfer function, $H(z)$, after the analog filter is mapped to the discrete time domain using the time sampling parameter T you found in part a. (If you are not sure about your answer for part a, then you may assume that $T = 10$.)

Solution:

$s = \frac{2}{T}\frac{z-1}{z+1} = 3\frac{z-1}{z+1}$,
$H\left(z\right) = H\left(s = 3\frac{z-1}{z+1}\right) = \frac{3\frac{z-1}{z+1}}{3\frac{z-1}{z+1}+3} = \frac{z-1}{2z}$.

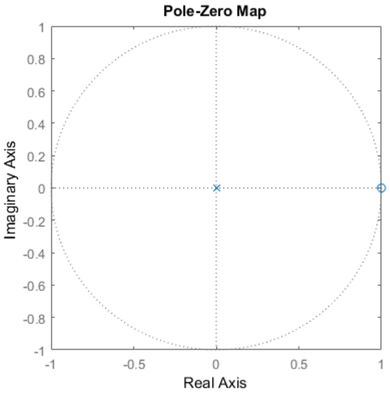


Figure 1: Pole Zero Plot

c) Identify any possible poles and zeros in the digital filter, and sketch a pole-zero plot. Make sure to clearly label your axes. Based on the pole-zero plot, what type of digital filter is this (e.g., low pass, band pass, high pass)? (If you are not sure about your answer for part b, then you may assume in parts c and d that $H(z) = 1 + z^{-2}$.)

Solution:

The gain is high at 0 (pole) and low at 1 (zero), which represents a high-pass filter. Refer to Figure 1 for the plot.

d) Convert the transfer function, $H(z)$, to a difference equation. (The result of the conversion should look something like $y(n) = 2y(n-1) - 3x(n) + 4x(n-1)$.)

Solution:

$2zY(z) = zX(z) - X(z)$,

$y(n) = \frac{1}{2}x(n) - \frac{1}{2}x(n-1)$.

Question 4 (MATLAB)

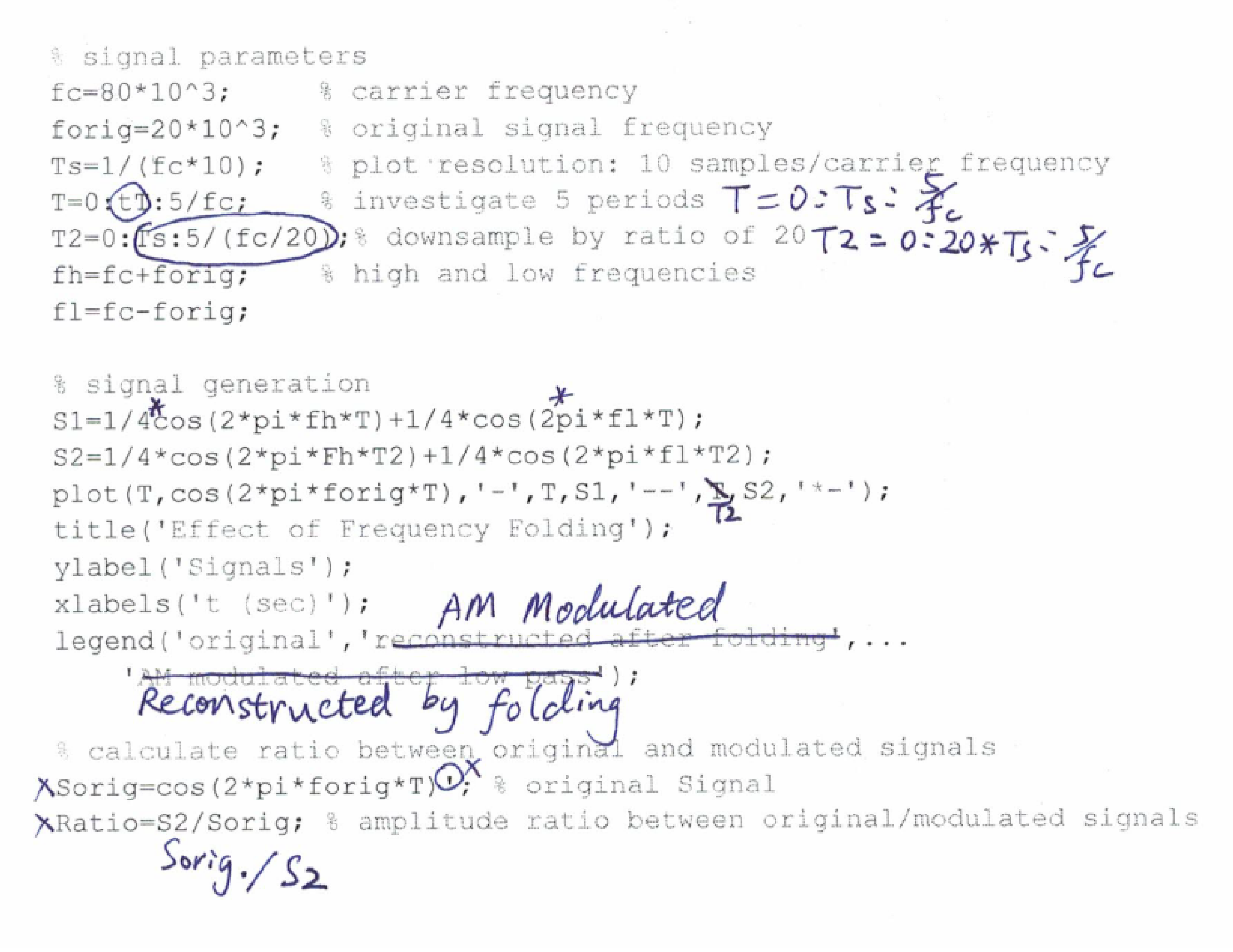
Recall from Project 1 that the final demodulated signal maintained the frequency and phase of the original signal due to frequency folding, but the amplitude was scaled by some ratio. The MATLAB script below implements AM demodulation using frequency folding; it generates a modulated signal S_1 and downsamples it by a ratio of 20 to obtain the demodulated signal S_2. After plotting the (*i*) original signal (prior to modulation) S_{orig}; (*ii*) modulated signal S_1; and (*iii*) demodulated signal S_2 *on the same graph*, the amplitude ratio is determined by dividing S_{orig} by S_2. There are multiple mistakes in the script, including coding errors (e.g., syntax) and implementation errors (e.g., wrong variables). Please identify at least 3 of these errors. For each error you identify, circle/underline the error, explain its nature, and how to correct it.

```
% signal parameters
fc=80*10^3;     % carrier frequency
forig=20*10^3;  % original signal frequency
Ts=1/(fc*10);   % plot resolution: 10 samples/carrier frequency
T=0:tT:5/fc;    % investigate 5 periods
T2=0:Ts:5/(fc/20);% downsample by ratio of 20
fh=fc+forig;    % high and low frequencies
fl=fc-forig;

% signal generation
S1=1/4cos(2*pi*fh*T)+1/4*cos(2pi*fl*T);
S2=1/4*cos(2*pi*Fh*T2)+1/4*cos(2*pi*fl*T2);
plot(T,cos(2*pi*forig*T),'-',T,S1,'--',T,S2,'*-');
title('Effect of Frequency Folding');
ylabel('Signals');
xlabels('t (sec)');
legend('original','reconstructed after folding',...
    'AM modulated after low pass');

% calculate ratio between original and modulated signals
Sorig=cos(2*pi*forig*T)'; % original Signal
Ratio=S2/Sorig; % amplitude ratio between original/modulated signals
```

Refer to Figure 2 for the errors in the code.



```
% signal parameters
fc=80*10^3;       % carrier frequency
forig=20*10^3;    % original signal frequency
Ts=1/(fc*10);     % plot resolution: 10 samples/carrier frequency
T=0:tT:5/fc;      % investigate 5 periods T=0:Ts:5/fc
T2=0:Ts:5/(fc/20);% downsample by ratio of 20 T2=0:20*Ts:5/fc
fh=fc+forig;      % high and low frequencies
fl=fc-forig;

% signal generation
S1=1/4*cos(2*pi*fh*T)+1/4*cos(2*pi*fl*T);
S2=1/4*cos(2*pi*Fh*T2)+1/4*cos(2*pi*fl*T2);
plot(T,cos(2*pi*forig*T),'-',T,S1,'--',T,S2,'*-');
                                       T2
title('Effect of Frequency Folding');
ylabel('Signals');
xlabels('t (sec)');       AM Modulated
legend('original','reconstructed after folding',...
    'AM modulated after low pass');
   Reconstructed by folding

% calculate ratio between original and modulated signals
xSorig=cos(2*pi*forig*T)'; % original Signal
xRatio=S2/Sorig; % amplitude ratio between original/modulated signals
       Sorig./S2
```

Figure 2: Code Debugging

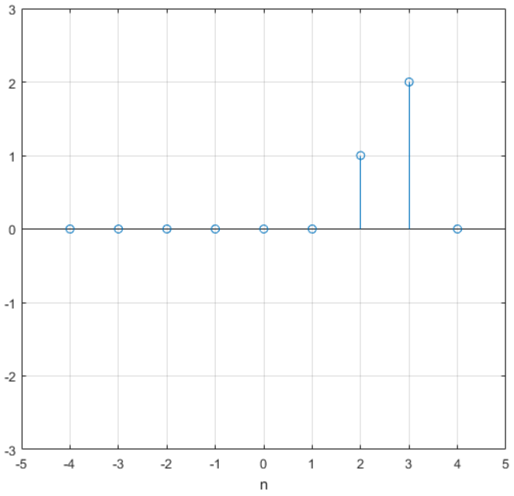


Figure 3: Impulse Response

Question 5 (LTI systems)
This question considers a linear time invariant (LTI) system that for an input $x(n) = \delta(n)$ generates the output $y(n) = \delta(n-2) + 2\delta(n-3)$.

a) Sketch the impulse response. As before, make sure to clearly label your axes.
Refer to Figure 3 for the impulse response.

b) Provide a difference equation relating the output $y(n)$ to the input $x(n)$. (As before, the result of the conversion should look something like $y(n) = 2y(n-1) - 3x(n) + 4x(n-1)$.)
Solution:

The impulse response of an FIR filter corresponds to its filter tap. Thus $y(n) = x(n-2) + 2x(n-3)$.

c) Suppose a constant signal $x(n) = 2$ is an input to the LTI system. What is $y(n)$? (Hint: the output will be identical for all n.)
$y(n) = 2 + 4 = 6$.

d) Instead of the constant input of part c, we now input a periodic version of the signal from question 2 to the LTI system. (That is, the input has the form $x(n) = [\ldots 1, 3, \underline{1}, 3, 1, 3, \ldots]$.) Please explain carefully how you could compute the output, $y(n)$. Full credit will be given for comprehensive explanations; computing the output will provide modest extra credit.

Solution:
Filtering process is convolution in time domain or multiplication in Fourier domain. From part c), we know the filter coefficients are $[0, 0, 1, 2]$.
We now consider the DFT following the convention in MATLAB (multiplying the 1/N factor at IDFT). The Fourier transform of the filter coefficients are $[3, -1 + 2i, 1, -1 - 2i]$. We are matching lengths of input $x(n)$ and filter coefficients while ensuring it is integer multiple of $x(n)$'s period. We already know the DFT for $[1, 3, 1, 3]$ is $[8, 0, -2, 0]$. Thus the DFT for the output is a element-wise product of the two Fourier series and we have [1.5,0,0.25,0]. Then we take the inverse DFT to get [7 5 7 5].

Notes:
1. Using normal convolution will not give you the correct answer because $x(n)$ is an infinite series.
2. If you just observe the periodic property of the output and found the solution numerically, I have given you full credits.
3. Note this method will only work following the convention of scaling $1/N$ for the IDFT process. But if you following the correct argument and used $1/N$ for the DFT process, I have given you full credits. But why only the first convention works? The hint is that due to Fourier basis being not orthonormal, when doing DFT/IDFT, we are effectively scaling the energy in the signal. But now we have two signal multiplied together, we need to be careful how many times we have actually scaled up/down the energy.