# ECE53800 Digital Signal Processing I
# Matlab Homework 3:
# Frequency-Domain Sampling.
# Time-Domain Aliasing

Boris Diner

December 10, 2021

## 1 Problem 7.29

In this problem, we studied the consequences of frequency-domain sampling. Given the infinite duration signal $x[n] = a^{|n-D|}$ (for $a = 0.8$, $D = \frac{N-1}{2}$) and the theoretical expression for the DTFT of this signal

$$X(\omega) = \frac{1 - a^2}{1 - 2a\cos\omega + a^2}$$

we reconstructed the plot for the DTFT $X(\omega)$ from its samples $X({}^{2\pi k}/_{N})$, $k = 0, 1, \ldots, N-1$, for 2 different numbers of samples ($N = 21$ and $N = 101$).

Below are the results of reconstruction.

## 1.1 Part (a) and (b)

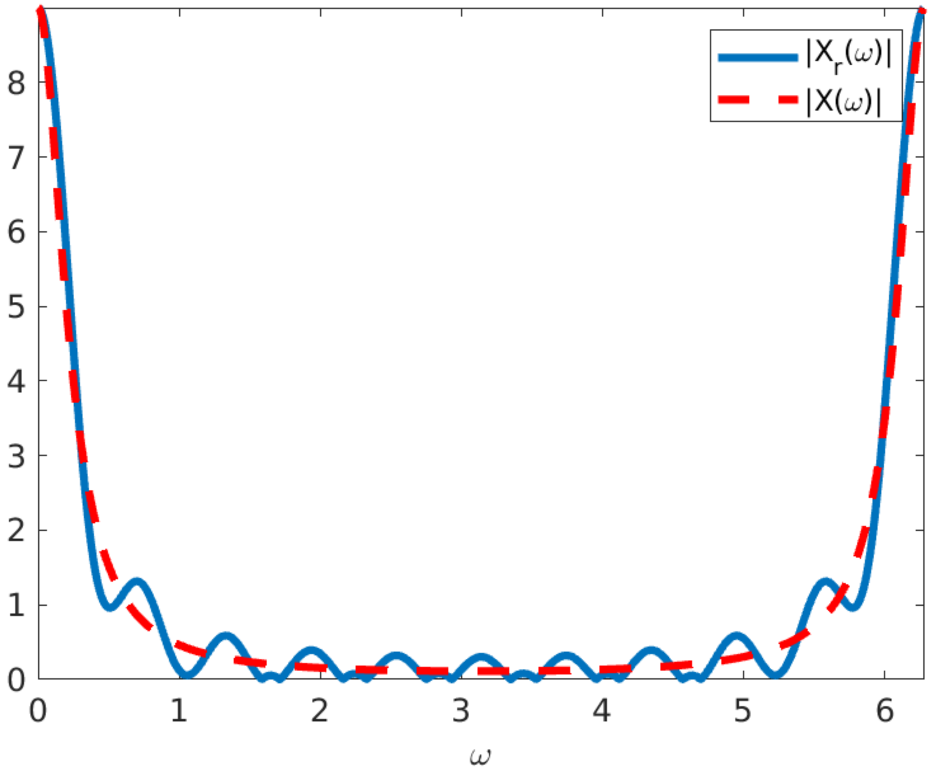


Figure 1: Reconstruction of the DTFT $X(\omega)$ from $N = 21$ samples of the DFT $X(2\pi k/N)$. $|X(\omega)|$ is the magnitude spectrum of the theoretical DTFT and $|X_r(\omega)|$ is the magnitude spectrum of the reconstructed DTFT

## 1.2 Part (a) and (c)

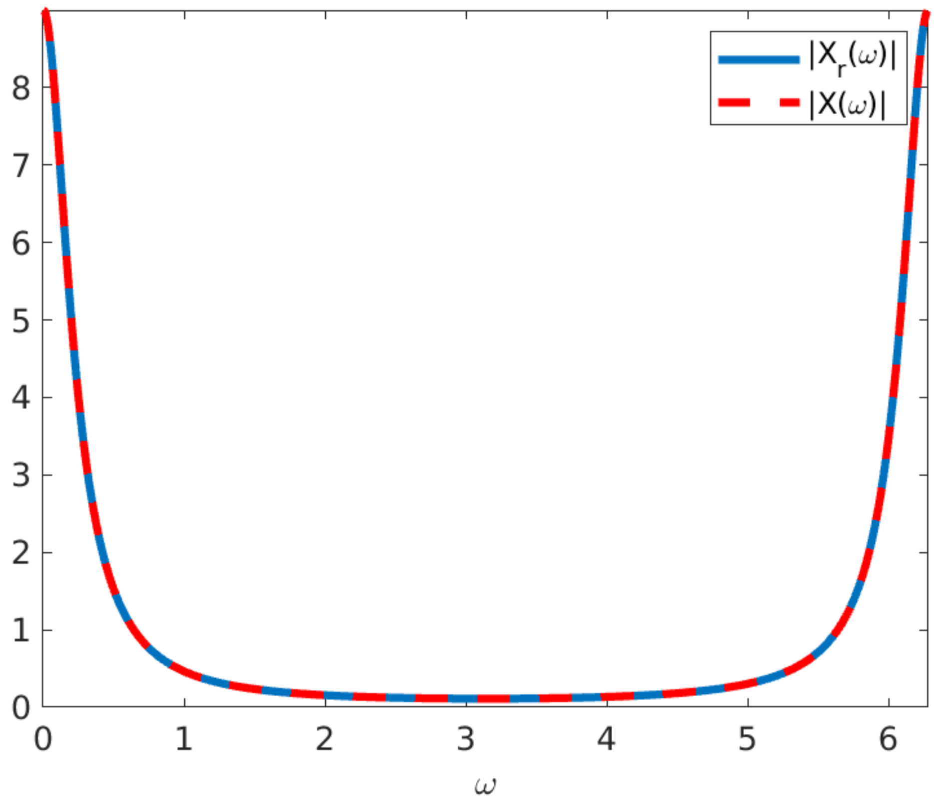


Figure 2: Reconstruction of the DTFT $X(\omega)$ from $N = 101$ samples of the DFT $X(^{2\pi k}/_{N})$. $|X(\omega)|$ is the magnitude spectrum of the theoretical DTFT and $|X_r(\omega)|$ is the magnitude spectrum of the reconstructed DTFT

## 1.3 Part (d)

We can observe from Figure 1 that $N = 21$ frequencies is not sufficient for the perfect reconstruction since there is significant difference between the theoretical curve and the reconstructed DTFT.

On the other hand, analysing the result for $N = 101$ depicted in Figure 2, we conclude that $N = 101$ is a good number of frequencies to reconstruct the DTFT $X(\omega)$. It is possible due to the fact the the given signal $x[n]$ is decaying as $n$ grows large.

## 1.4 Part (e)

Now using the time-domain aliasing formula $x_a[n] = x[n-N] + x[n] + x[n+N]$ for $N = 21$, $N = 101$, and $x[n] = a^{|n-D|}$, reconstructing the signal from $X_r(\omega)$ with help of the inverse DTFT and comparing these results with the theoretical curve $x[n] = a^{|n-D|}$ in the discrete time domain, we get the following results (see Figure 3 and Figure 4)

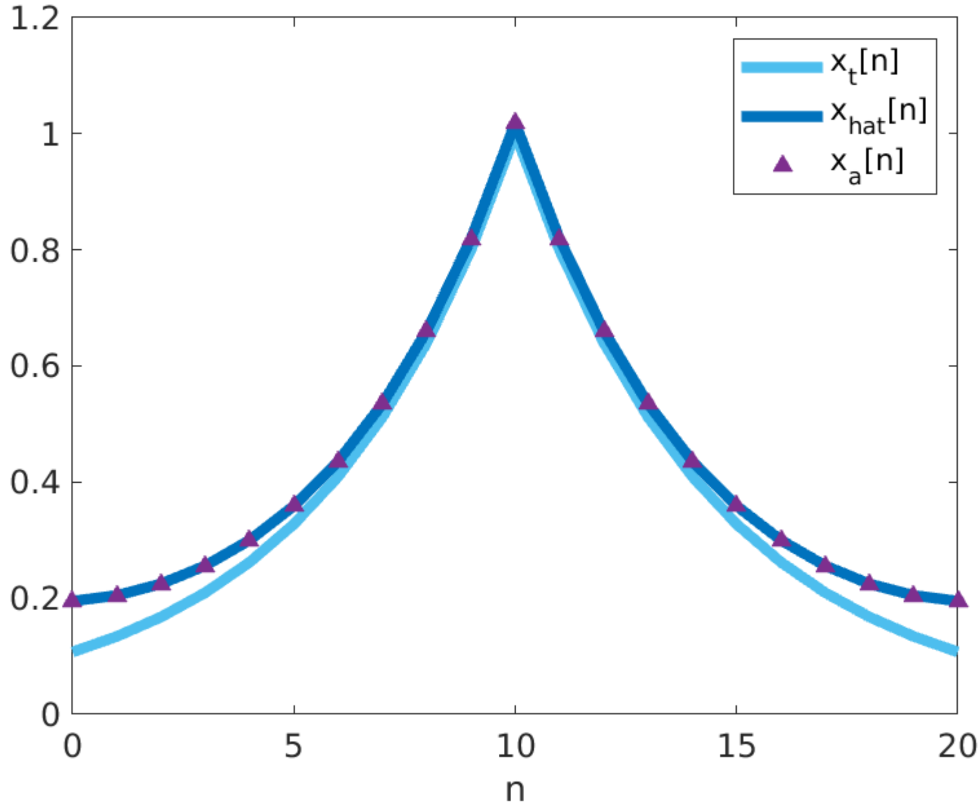


Figure 3: Time domain results. Here $x_t[n]$ is the theoretical curve, $x_{\text{hat}}[n]$ is the inverse DFT of $X_{21}(k)$, $x_{\text{a}}[n]$ is the signal obtained from the time-domain aliasing formula for $N = 21$

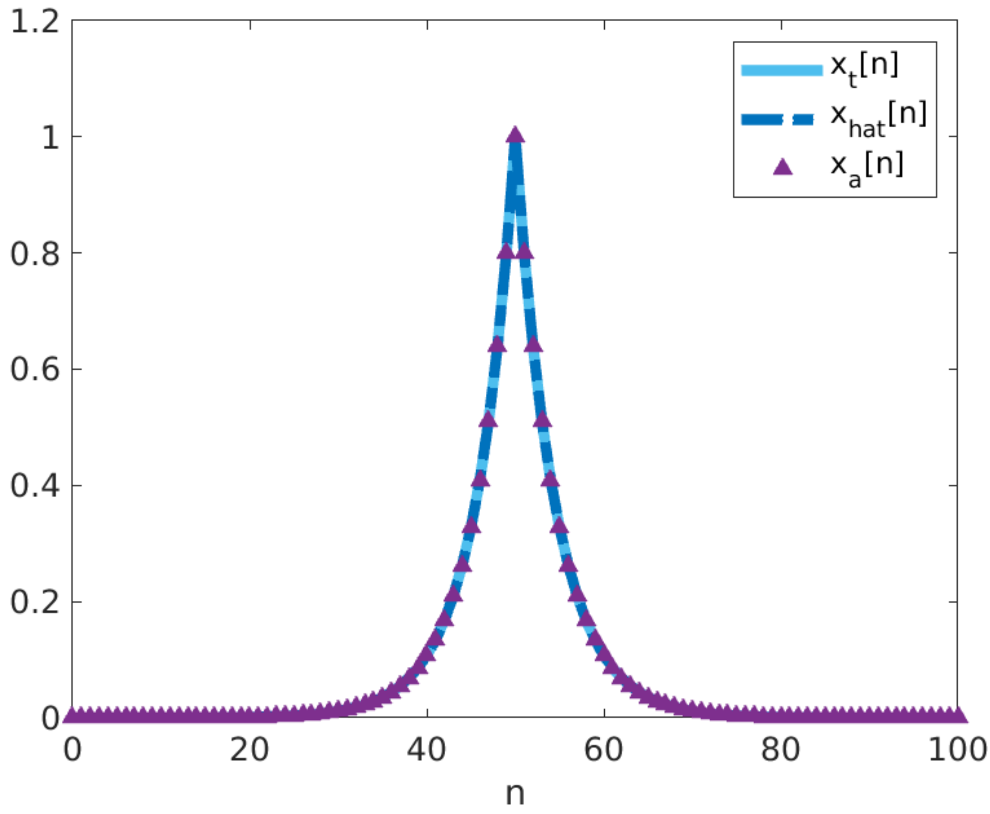


Figure 4: Time domain results. Here $x_t[n]$ is the theoretical curve, $x_{\text{hat}}[n]$ is the inverse DFT of $X_{101}(k)$, $x_{\text{a}}[n]$ is the signal obtained from the time-domain aliasing formula for $N = 101$

So we can observe time-domain aliasing issues in Figure 3 (for the case $N = 21$) while for the results presented in Figure 4 ($N = 101$), we can conclude that time-domain aliasing is negligible.

# 2 Problem 7.30

In problem 7.30, we considered the discrete-time signal

$$x[n] = \cos 2\pi f_1 n + \cos 2\pi f_2 n \tag{1}$$

where $f_1 = \frac{1}{128}$ and $f_2 = \frac{5}{128}$.

The signal from Equation 1 was then modulated by $x_c[n] = \cos 2\pi f_c n$ ($f_c = \frac{50}{128}$), so that $x_{am}[n] = \cos 2\pi f_c n \cdot (\cos 2\pi f_1 n + \cos 2\pi f_2 n)$

## 2.1 Part (a)

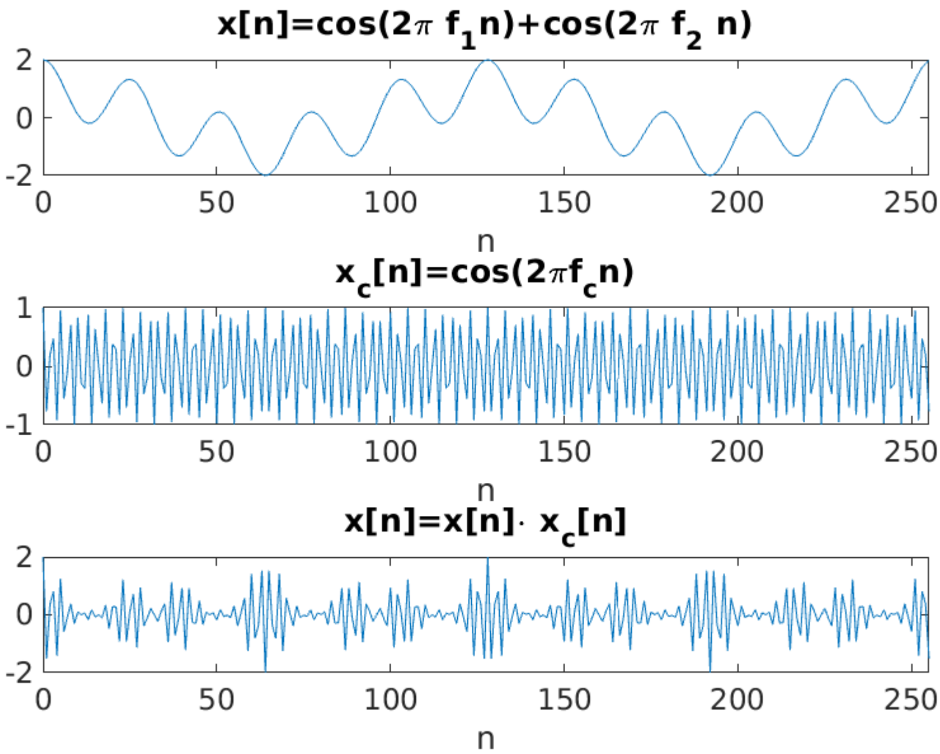


Figure 5: The signals $x[n]$, $x_c[n]$, and $x_{am}[n]$ for $0 \leq n \leq 255$

## 2.2 Parts (b), (c), and (d)

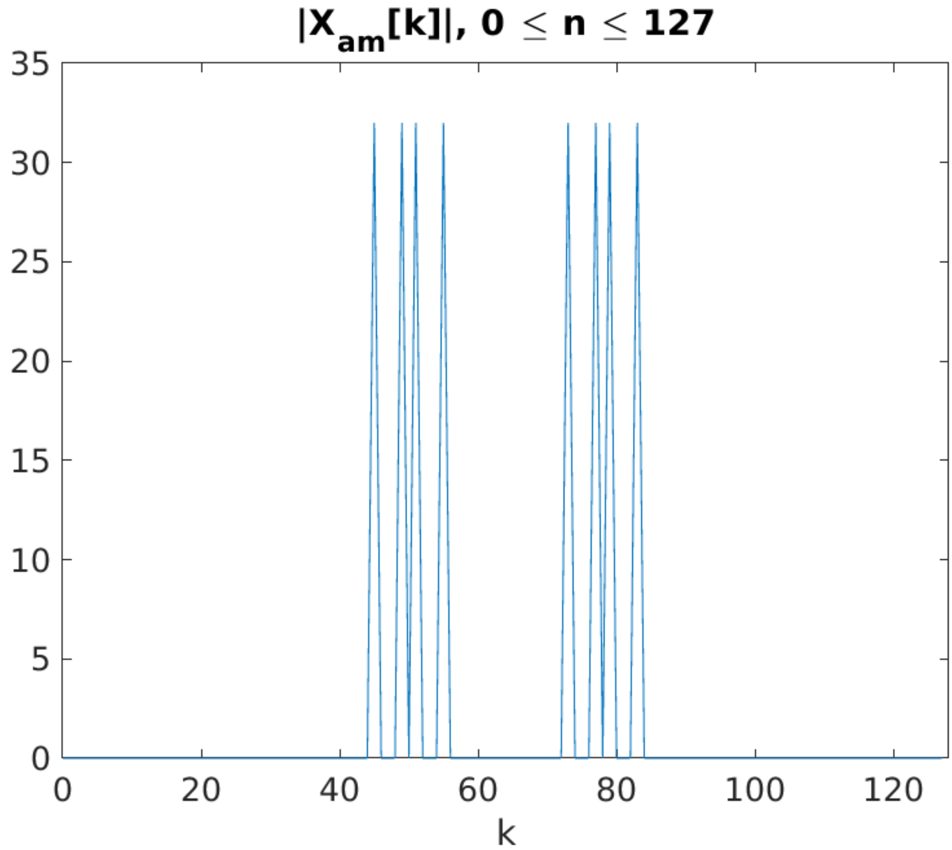


Figure 6: The 128-point DFT of the signal $x_{am}[n]$, $0 \leq n \leq 127$

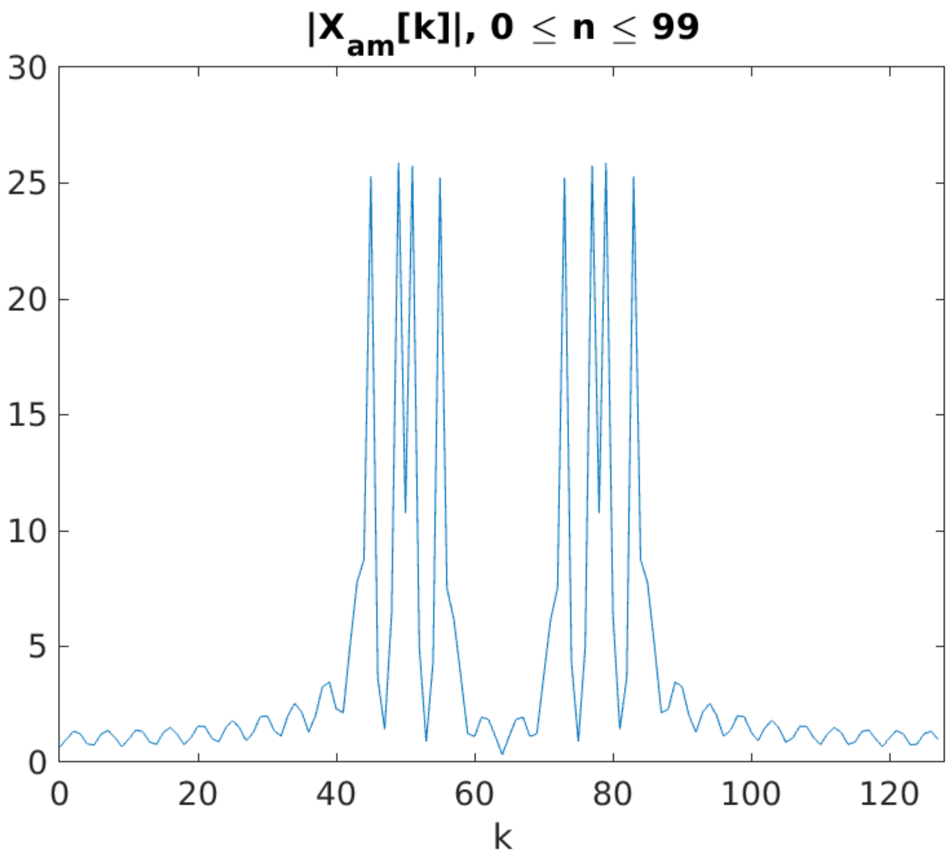


Figure 7: The 128-point DFT of the signal $x_{am}[n]$, $0 \leq n \leq 99$

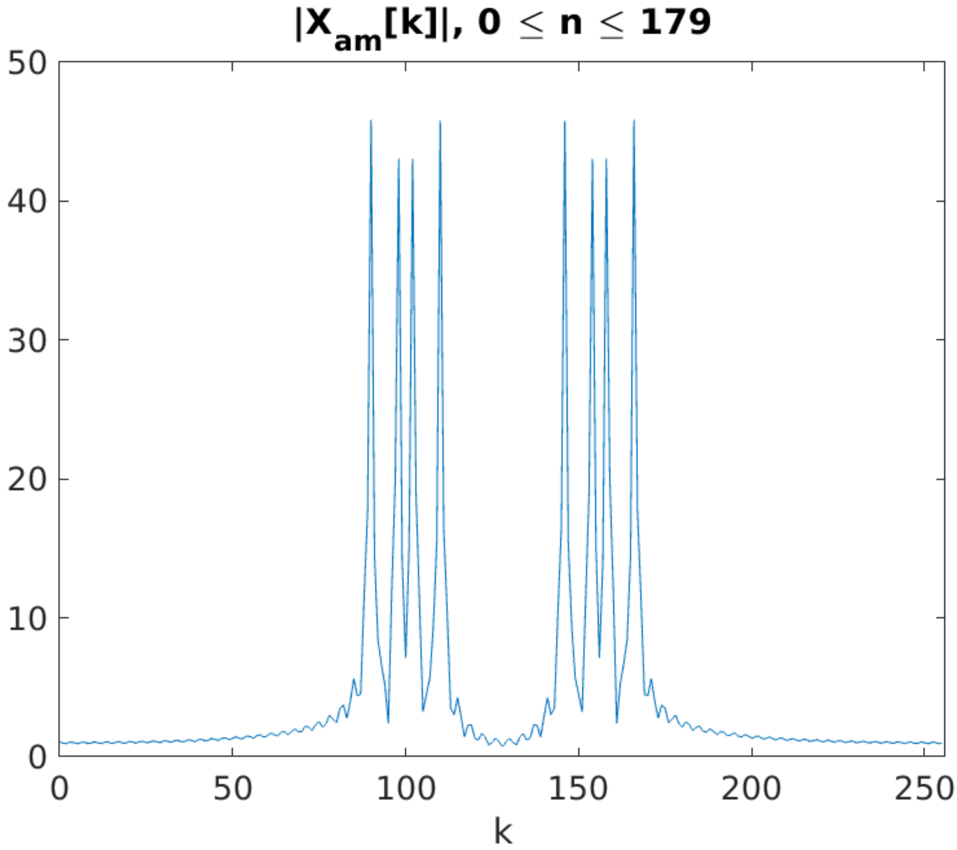


Figure 8: The 256-point DFT of the signal $x_{am}[n]$, $0 \leq n \leq 179$

## 2.3 Part (e)

We are given the discrete-time signal

$$x[n] = \cos 2\pi f_1 n + \cos 2\pi f_2 n$$

that modulates the amplitude of the carrier

$$x_c[n] = \cos 2\pi f_c n$$

So that the resulting amplitude-modulated signal is

$$x_{am}[n] = x[n] \cos 2\pi f_c n = \cos 2\pi f_c n \cdot (\cos 2\pi f_1 n + \cos 2\pi f_2 n) \qquad (2)$$

To find the DTFT of the signal (see Equation 2), we first note that

$$\cos 2\pi fn = \frac{e^{j2\pi fn} + e^{-j2\pi fn}}{2}$$

Denote $2\pi f_m \triangleq \omega_m$, then

$$\begin{aligned} x_{am}[n] &= x[n]\cos\omega_c n = \cos\omega_c n \cdot (\cos\omega_1 n + \cos\omega_2 n) = \\ &= \frac{e^{j\omega_c n} + e^{-j\omega_c n}}{2} \cdot \left( \frac{e^{j\omega_1 n} + e^{-j\omega_1 n}}{2} + \frac{e^{j\omega_2 n} + e^{-j\omega_2 n}}{2} \right) \end{aligned}$$

We also can use the fact that for $x[n] = e^{j\omega_k n}$

$$X(\omega) = \sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n} = \sum_{n=-\infty}^{+\infty} e^{-j(\omega-\omega_k)n} = 2\pi\delta(\omega - \omega_k + 2\pi m)$$

where $m$ is an integer.
For the interval $-\pi \le \omega \le \pi$, we get $X(\omega) = 2\pi\delta(\omega - \omega_k)$
So eventually we get the following expression

$$\begin{aligned} X_{am}(\omega) = \frac{\pi}{2}\{ &\delta(\omega - (\omega_1 + \omega_c)) + \delta(\omega + (\omega_1 + \omega_c))+ \\ &+ \delta(\omega - (\omega_c - \omega_1)) + \delta(\omega + (\omega_c - \omega_1))+ \\ &+ \delta(\omega - (\omega_c - \omega_2)) + \delta(\omega + (\omega_c - \omega_2))+ \\ &+ \delta(\omega - (\omega_2 + \omega_c)) + \delta(\omega + (\omega_2 + \omega_c))\} \end{aligned}$$

So for an infinite "length" sine waves, we have 8 $\delta$-peaks. We are also given that $f_1 = \frac{1}{128}$, $f_1 = \frac{5}{128}$, and $f_c = \frac{50}{128}$ that leads to the following locations of the $\delta$-peaks.

$$\omega_1 + \omega_c = 2\pi(f_1 + f_c) = \frac{2\pi \cdot (1+50)}{128} = \frac{2\pi \cdot 51}{128} \tag{3}$$

$$\omega_c - \omega_1 = 2\pi(f_c - f_1) = \frac{2\pi \cdot (50-1)}{128} = \frac{2\pi \cdot 49}{128} \tag{4}$$

$$\omega_c - \omega_2 = 2\pi(f_c - f_2) = \frac{2\pi \cdot (50-5)}{128} = \frac{2\pi \cdot 45}{128} \tag{5}$$

$$\omega_2 + \omega_c = 2\pi(f_2 + f_c) = \frac{2\pi \cdot (5+50)}{128} = \frac{2\pi \cdot 55}{128} \tag{6}$$

A finite length sine wave with the frequency $\omega_0 = \frac{2\pi}{N}$ of length $L$ can be written as

$$x_f[n] = e^{j\omega_0 n}\{u[n] - u[n-L]\}$$

The DTFT of the window $w[n] = u[n] - u[n-L]$ is

$$W(\omega) = e^{-j\frac{(L-1)}{2}\omega} \frac{\sin(\frac{L}{2}\omega)}{\sin\left(\frac{1}{2}\omega\right)}$$

Then by the modulation property of DTFT,

$$X(\omega) = e^{-j\frac{(L-1)}{2}\{\omega-\omega_0\}} \frac{\sin(\frac{L}{2}\{\omega-\omega_0\})}{\sin\left(\frac{1}{2}\{\omega-\omega_0\}\right)} \tag{7}$$

where $X(\omega)$ is the DTFT of a finite length sine wave.

To get the DFT of a finite length sine wave, we should sample the DTFT in Equation 7.

So in Figure 6, we observe the "lucky" case for which the length of the sine wave $L$ is equal to the number of frequency domain sampling points $N$ and the frequency of the sine wave is one of the frequencies we are sampling at. That is, we only see all 8 $\delta$-peaks, their locations are (see Equation 6) $k_1 = 45$, $k_2 = 49$, $k_3 = 51$, $k_4 = 55$. Since DFT is defined on the interval $[0, 2\pi]$, we have $k_5 = N - 45 = 128 - 45 = 83$, $k_6 = 128 - 49 = 79$, $k_7 = 128 - 51 = 77$, $k_8 = 128 - 55 = 73$.

The result obtained for $L = 100$ and $N = 128$ (see Figure 7) is an "unlucky" case and we observe spectral leakage. Peaks are still visible, yet we can see the energy spread over other frequencies as well.

Doubling the number of frequencies we are sampling at (see Figure 8), we can see that the situation improves slightly compared to the previous case, yet it is still an "unlucky" case ($L = 180$, $N = 256$) and we also observe spectral leakage.

# 3 The Matlab code

```
set(0,'defaultaxesfontsize',12);
clear
clf

% Problem 7.29
% Parameters
a=0.8;
Nfft=1024;

% The DTFT of x[n]
wd=linspace(0,2*pi-2*pi/Nfft,Nfft);
Xw = (1-a^2)./(1-2*a*cos(wd)+a^2);

% Reconstructed versions of the DTFT above
% for different number of its samples N
Xr_21 = rec_dtft(a,21,Nfft,wd);
Xr_101 = rec_dtft(a,101,Nfft,wd);

fig1=figure(1);
% The reconstructed spectrum from the formula for N
    =21
plot(wd,abs(Xr_21),'color','#0072BD','Linewidth',3);
hold on
% The DTFT
```

```
plot(wd,abs(Xw),'r--', 'Linewidth',3);
axis([0 2*pi 0 max(abs(Xw))]);
legend('|X_{r}(\omega)|','|X(\omega)|');
xlabel('\omega');
saveas(fig1, sprintf('lab3fig1.png'));

fig2=figure(2);
% The reconstructed spectrum from the formula for N=101
plot(wd,abs(Xr_101),'color','#0072BD','Linewidth',3);
hold on
% The DTFT
plot(wd,abs(Xw), 'r--', 'Linewidth',3);
axis([0 2*pi 0 max(abs(Xw))]);
legend('|X_{r}(\omega)|','|X(\omega)|');
xlabel('\omega');
saveas(fig2, sprintf('lab3fig2.png'));

% Time domain aliasing
[x_t_21, x_hat_21, x_a_21] = time_domain_aliasing(a,21);
[x_t_101, x_hat_101, x_a_101] = time_domain_aliasing(a,101);

n_21=0:20;

fig3=figure(3);
plot(n_21,x_t_21,'color','#4DBEEE','Linewidth',4);
hold on
plot(n_21,x_hat_21,'color','#0072BD', 'Linewidth',4);
hold on
plot(n_21,x_a_21,'^','MarkerFaceColor','#7E2F8E','MarkerEdgeColor','#7E2F8E');
```

```
hold off
legend('x_t[n]','x_{hat}[n]','x_a[n]');
xlabel('n');
saveas(fig3, sprintf('lab3fig3.png'));

n_101=0:100;

fig4=figure(4);
plot(n_101,x_t_101,'color','#4DBEEE','Linewidth',4);
hold on
plot(n_101,x_hat_101,'-.','color','#0072BD', 'Linewidth',4);
hold on
plot(n_101,x_t_101,'^','MarkerFaceColor','#7E2F8E','MarkerEdgeColor','#7E2F8E');
hold off
legend('x_t[n]','x_{hat}[n]','x_a[n]');
xlabel('n');
saveas(fig4, sprintf('lab3fig4.png'));

% Problem 7.30
% Frequencies
f1 = 1/128;
f2 = 5/128;
fc = 50/128;

% a)
n = 0:255;
x = cos(2*pi*f1*n) + cos(2*pi*f2*n);
x_c = cos(2*pi*fc*n);
x_am = x.*cos(2*pi*fc*n);

fig5=figure(5);
subplot(3,1,1);
plot(n, x);
```

```
xlabel('n');
xlim([0 n(end)]);
title('x[n]=cos(2\pi f_{1}n)+cos(2\pi f_{2} n)');
subplot(3,1,2);
plot(n,x_c);
xlabel('n');
xlim([0 n(end)]);
title('x_c[n]=cos(2\pif_cn)');
subplot(3,1,3);
plot(n,x_am);
xlabel('n');
xlim([0 n(end)]);
title('x[n]=x[n]\cdot x_c[n]');
saveas(fig5, sprintf('lab3fig5.png'));


% b), c), and d)
X_am_128 = fft(x_am(1:128),128);
X_am_128_t = fft(x_am(1:100),128);
X_am_256 = fft(x_am(1:180),256);

fig6=figure(6);
plot((0:127),abs(X_am_128));
title('|X_{am}[k]|, 0 \leq n \leq 127');
xlabel('k');
xlim([0 128]);
saveas(fig6, sprintf('lab3fig6.png'));

fig7=figure(7);
plot((0:127),abs(X_am_128_t));
title('|X_{am}[k]|, 0 \leq n \leq 99');
xlabel('k');
xlim([0 128]);
saveas(fig7, sprintf('lab3fig7.png'));

```

```
fig8=figure(8);
plot((0:255),abs(X_am_256));
title('|X_{am}[k]|, 0 \leq n \leq 179');
xlabel('k');
xlim([0 256]);
saveas(fig8, sprintf('lab3fig8.png'));

function Xr = rec_dtft(a,N,Nfft,wd)

D=(N-1)/2;

% initialize the reconstructed DTFT
Xr=zeros(1,Nfft);

for k=0:N-1
    wk=2*pi*k/N;
    wvec(1,k+1)=wk;
    % Sampling the "true" DTFT at N equispaced frequencies
    Xk=(1-a^2)./((1-2*a*cos(wk)+a^2)).*exp(-1i*wk*D);
    Xvec(1,k+1)=Xk;
    % Exercising the reconstruction formula
    Xr=Xr+Xk*(sin((wd-wk)*N/2).*exp(-1i*(wd-wk)*(N-1)/2))./(N*sin((wd-wk)/2));
end

end

function [x_t, x_hat, x_a] = time_domain_aliasing(a,N)

D = (N-1)/2;

% Original x[n]
```

```
n = 0:(N-1);
x_t = a.^(abs(n-D));
% Construct x_hat[n]
k = 0:N-1;
wk = 2*pi*k/N;
X_wk = (1-a^2)./(1-2*a*cos(wk)+a^2).*exp((-1i)*D*wk);
x_hat = real(ifft(X_wk,N));
% Construct x_a[n]
x_a = a.^(abs(n+N-D)) + a.^(abs(n-D)) + a.^(abs(n-N-D));

end
```