ESE 531: Homework 3

Noah Schwab

February 14, 2022

Problem solutions with figures are shown below. Work and code is shown in attachments at end of document.

4.28

a) $X_c(j\Omega) = 2\pi e^{-j\pi/4}\delta[\Omega - 100\pi] + 2\pi e^{j\pi/4}\delta[\Omega + 100\pi] + \pi e^{j\pi/3}\delta[\Omega - 300\pi] + \pi e^{-j\pi/4}\delta[\Omega + 300\pi]$

See attached plot.

b) $x_r(t) = x_c(t) = 2\cos(100\pi t - \pi/4) + \cos(300\pi t + \pi/3)$

See attached plot.

c) $x_r(t) = 2\cos(100\pi t - \pi/4) + \cos(200\pi t - \pi/3)$

See attached plot.

d) A $= \frac{1}{2}$

$x_r(t) = \frac{1}{2} + 2\cos(100\pi t - \pi/4)$

See attached plot.

4.30

See attached plots.

4.32

a) See attached plots.

b) $y[n] = \delta[n]$

See attached plots.

4.40

a) See attached plots.

b) $\epsilon = \frac{1}{8}$

c) See attached plot.

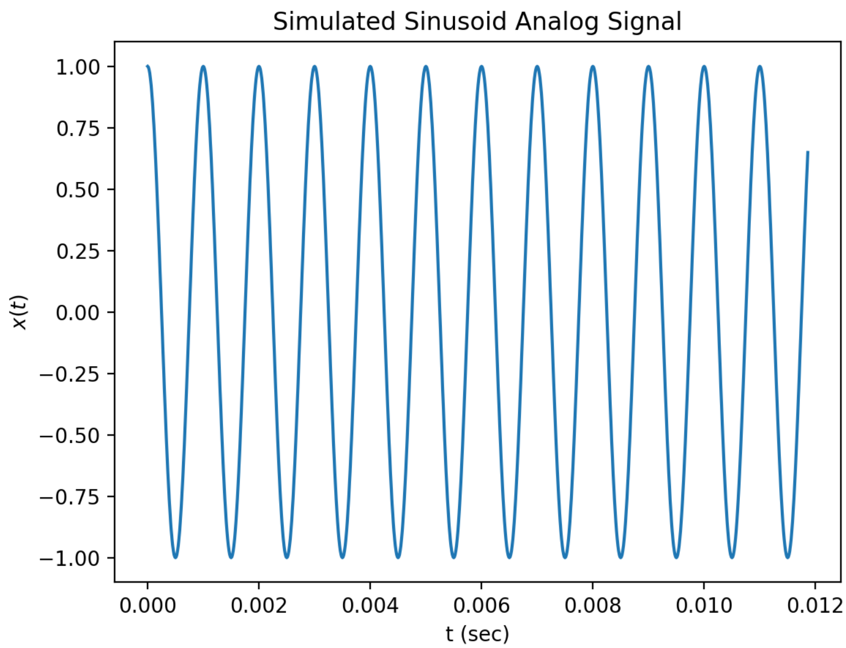
d) See attached plots.

4.49

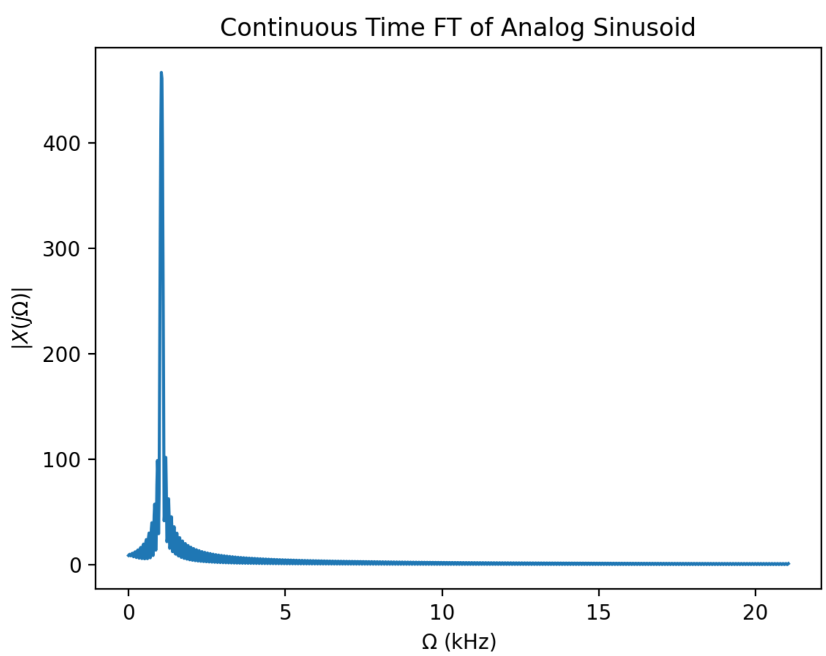
See attached plots.

Matlab Problem 1

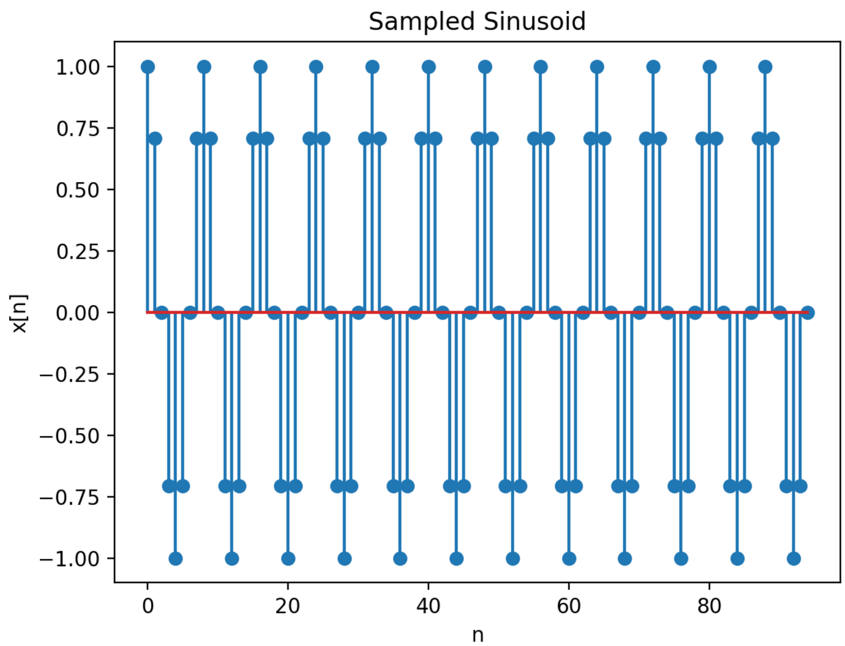
a) Simulated Sinusoid Analog Signal



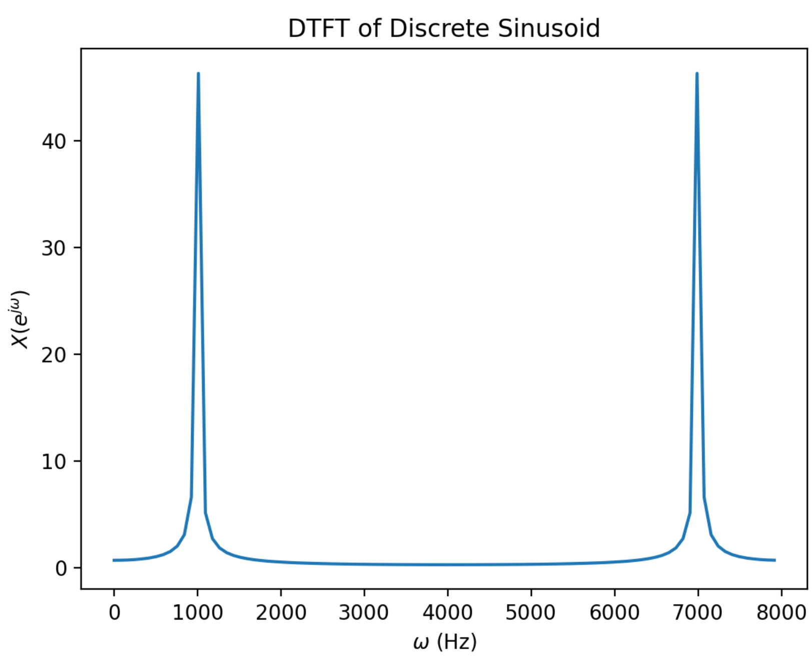
b) Fourier Transform of Analog Signal



c) Sampled Signal



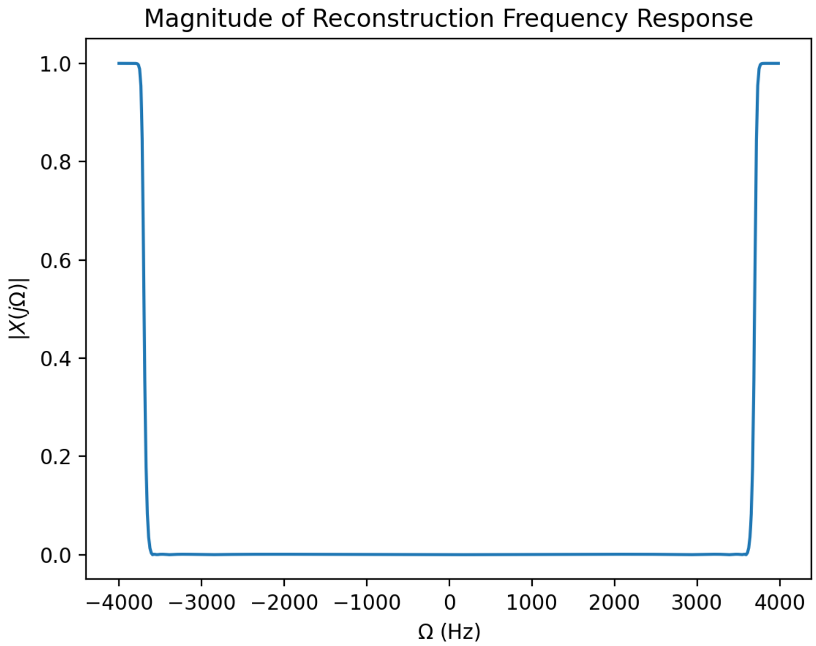
d) DTFT of Sampled Signal

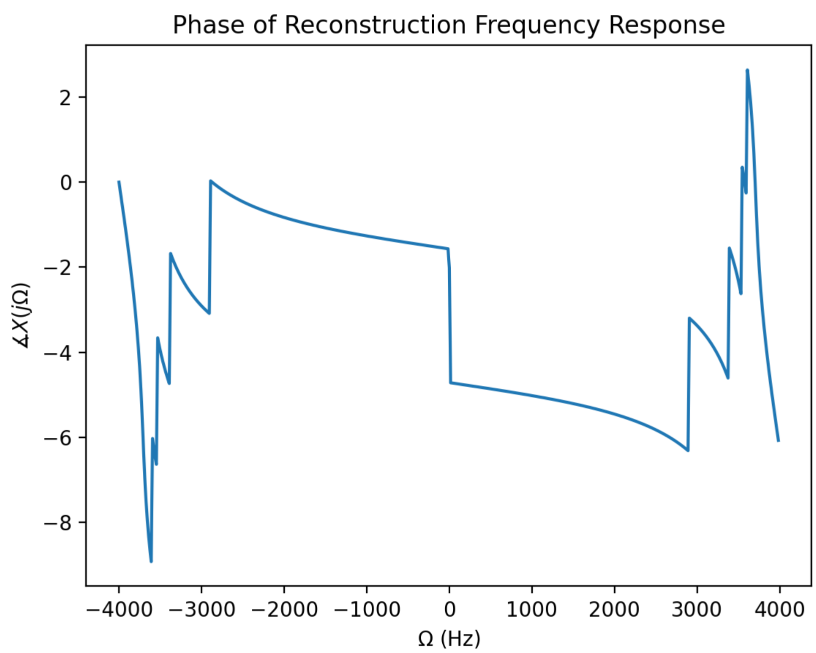


As we should expect, the DTFT of the sampled sinusoid is the same version as the continuous FT, except it is periodized in $f_s/2$.

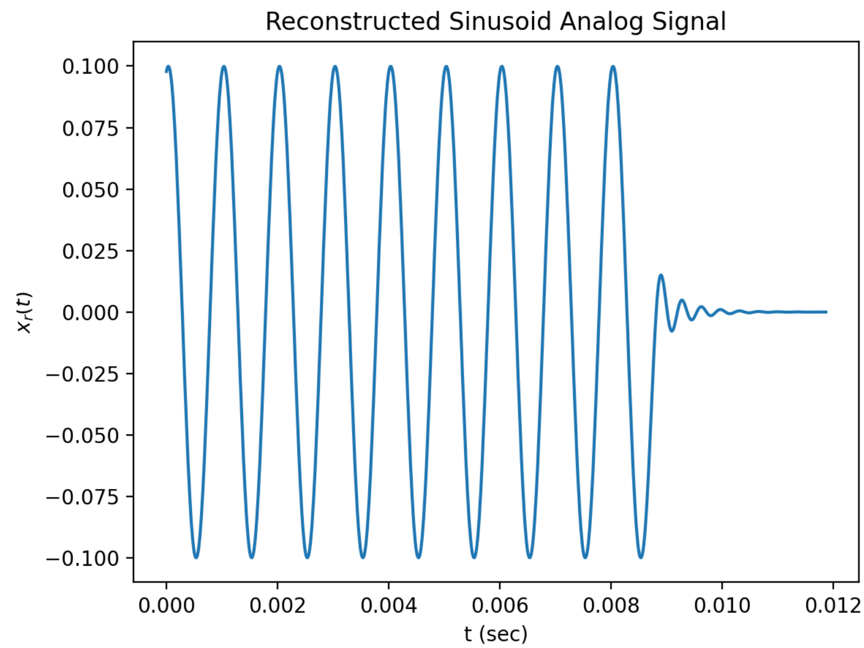
Matlab Problem 2

a) Frequency Response of Simulated Reconstruction Filter

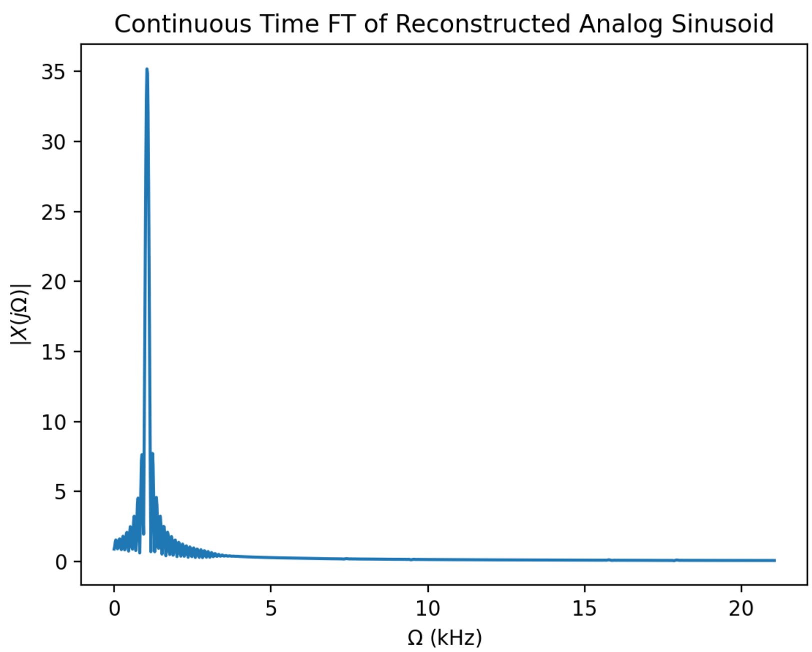




b) Reconstructed Signal and its FT



Reconstructed Sinusoid Analog Signal
$x_r(t)$
t (sec)
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
0.000
0.002
0.004
0.006
0.008
0.010
0.012



Continuous Time FT of Reconstructed Analog Sinusoid
$|X(j\Omega)|$
Ω (kHz)
35
30
25
20
15
10
5
0
0
5
10
15
20

Noah Schwab
ESE 531

04/13/22

HW 3: Reconstruction, DT/CT Systems, Re-Sampling

4.28

a) $X_c(j\Omega) = \int_{-\infty}^{\infty} x_c(t)\, e^{-j\Omega t}\, dt$

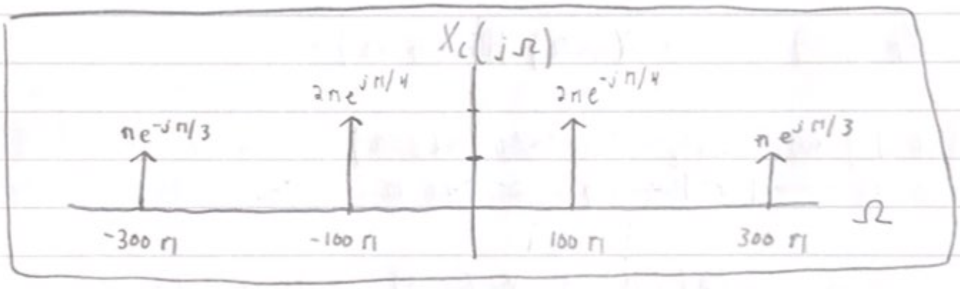
$= \int_{-\infty}^{\infty} 2\cos(100\pi t - \pi/4)\, e^{-j\Omega t}\, dt + \int_{-\infty}^{\infty} \cos(300\pi t + \pi/3)\, e^{-j\Omega t}\, dt$

We know the Fourier Transform pair:

$\cos(\alpha t + \phi) = \pi e^{j\phi} \delta[\Omega - \alpha] + \pi e^{-j\phi} \delta[\Omega + \alpha]$

$\Rightarrow$

$$\boxed{\begin{aligned} X_c(j\Omega) &= 2\pi e^{-j\pi/4} \delta[\Omega - 100\pi] + 2\pi e^{j\pi/4} \delta[\Omega + 100\pi] \\ &+ \pi e^{j\pi/3} \delta[\Omega - 300\pi] + \pi e^{-j\pi/3} \delta[\Omega + 300\pi] \end{aligned}}$$



b) $f_s = \frac{1}{T} = 500$ samples/sec, $\Omega_s = \frac{2\pi}{T} = 1000\pi$

$X_s(j\Omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c(j(\Omega - k\Omega_s))$

$= 500 \sum_{k=-\infty}^{\infty} X_c(j(\Omega - 1000\pi k))$

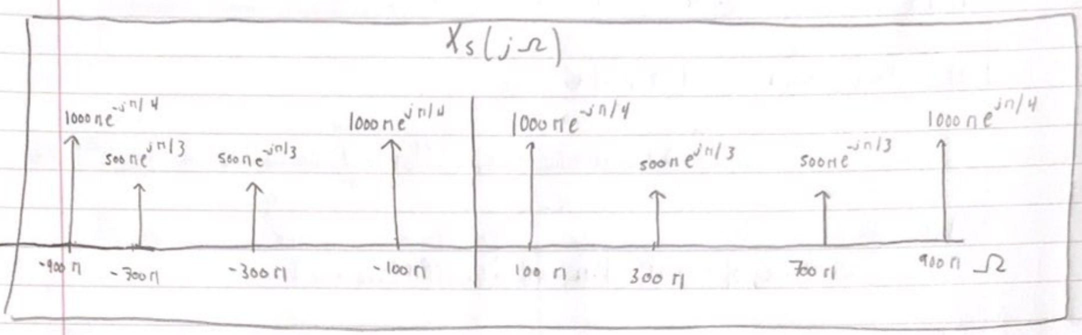
$X_c(j\Omega)$ is only non-zero at $\Omega = \pm 300\pi, \pm 100\pi$

$X_s(j\Omega) = 1000\pi e^{-j\pi/4} \delta[\Omega - 100\pi - 1000\pi k] + 1000\pi e^{j\pi/4} \delta[\Omega + 100\pi - 1000\pi$

$+ 500\pi e^{j\pi/3} \delta[\Omega - 300\pi - 1000\pi k] + 500\pi e^{-j\pi/3} \delta[\Omega + 300\pi - 1000$

for $-\infty < k < \infty$

In the range $-1000\pi \le \Omega \le 1000\pi$



To solve for $x_r(t)$, we can first compute its DTFT

$$X_r(j\Omega) = X_s(j\Omega)\, H_r(j\Omega)$$

$X_r(j\Omega)$ will only be $X_s(j\Omega)$ scaled by T in the range $|\Omega| \le \pi/T = 500\pi$, and zero elsewhere.

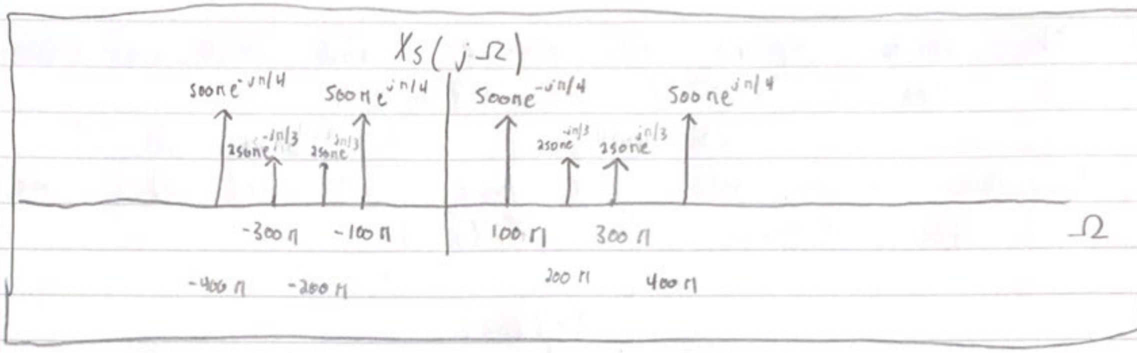
This means $X_r(j\Omega) = X_c(j\Omega)$. Therefore,

$$\boxed{x_r(t) = x_c(t) = 2\cos(100\pi t - \pi/4) + \cos(300\pi t + \pi/3)}$$

c) $F_s = \frac{1}{T} = 250$ samples/sec, $\Omega_s = 500\pi$

$$X_s(j\Omega) = 250 \sum_{k=-\infty}^{\infty} X_c(j(\Omega - 500\pi k))$$

$$X_s(j\Omega) = 500\pi e^{-j\pi/4}\delta[\Omega - 100\pi - 500\pi k] + 500\pi e^{j\pi/4}\delta[\Omega + 100\pi - 500\pi k] + 250\pi e^{j\pi/3}\delta[\Omega - 300\pi - 500\pi k] + 250\pi e^{-j\pi/3}\delta[\Omega + 300\pi - 500\pi k]$$



$$X_r(j\Omega) = X_s(j\Omega)\, H_r(j\Omega)$$

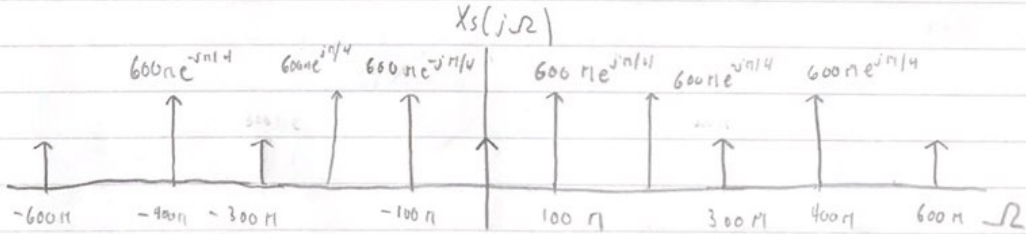
$X_r(j\Omega)$ is $X_s(j\Omega)$ scaled by T in the range $|\Omega| \leq 250\pi$ and zero elsewhere.

As we can see, there has been aliasing. We can use Fourier transform pairs:

$$\boxed{x_r(t) = 2\cos\left(100\pi t - \frac{\pi}{4}\right) + \cos\left(200\pi t - \frac{\pi}{3}\right)}$$

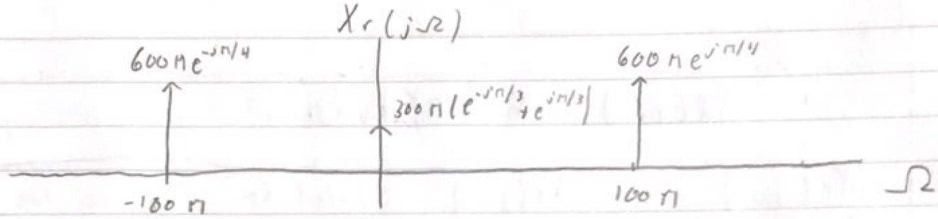
d) For $x_r(t) = A + 2\cos(100\pi t - \frac{\pi}{4})$, where A is a constant, to be satisfied, $X_r(j\Omega)$ must be filtered such that only the spikes at $\pm 100\pi$ remain after $x_s(t)$ is passed to $H_r(j\Omega)$.

If we sample at $f_s = \frac{1}{T} = 150$ samples/sec, $X_s(j\Omega)$ becomes



The shorter spikes all overlap, and have a value of $300\pi(e^{-j\pi/3} + e^{j\pi/3})$ at $\pm 300\pi k$.

Therefore, when we reconstruct w/ $H_r(j\Omega)$, we get the following for $X_r(j\Omega)$



$\Rightarrow$

$$X_r(t) = 2\cos(100\pi t - \pi/4) + \cos(\pi/3)$$

$$\boxed{X_r(t) = \frac{1}{2} + 2\cos(100\pi t - \pi/4)}$$

$$\boxed{A = \frac{1}{2}}$$

4.30

$X_c(j\Omega)$ is bandlimited w/ $\Omega_N = \pi \times 10^4$. Therefore, if $\Omega_N \leq \pi/T$, then $H_{eff}(j\Omega) = \begin{cases} H(e^{j\Omega T}), & |\Omega| < \pi/T \\ 0, & |\Omega| \geq \pi/T \end{cases}$

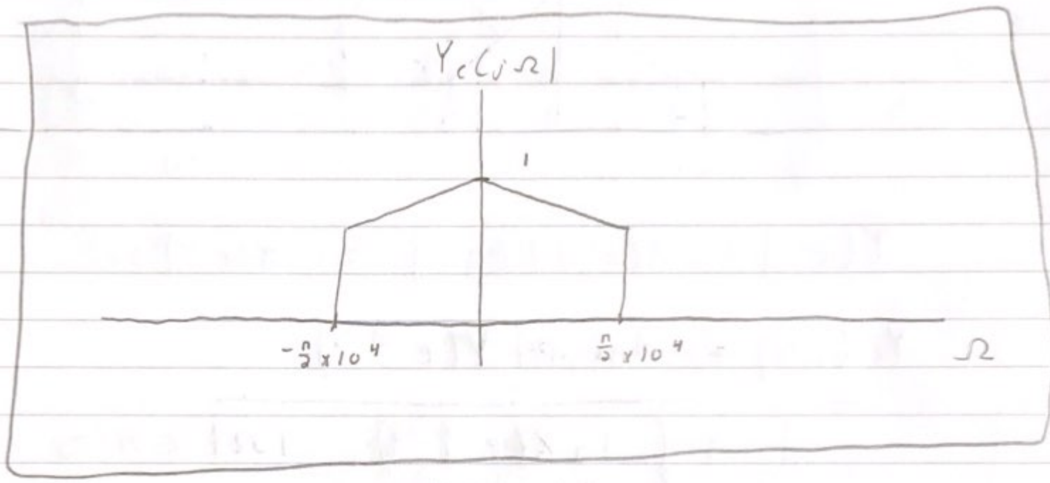
Otherwise, we will have to go through the whole system

$w = \Omega T$

a) $\frac{1}{T_1} = \frac{1}{T_2} = 10^4$

In this case, $\frac{\pi}{T} = \pi \times 10^4 = \Omega_N$, so Nyquist is satisfied.

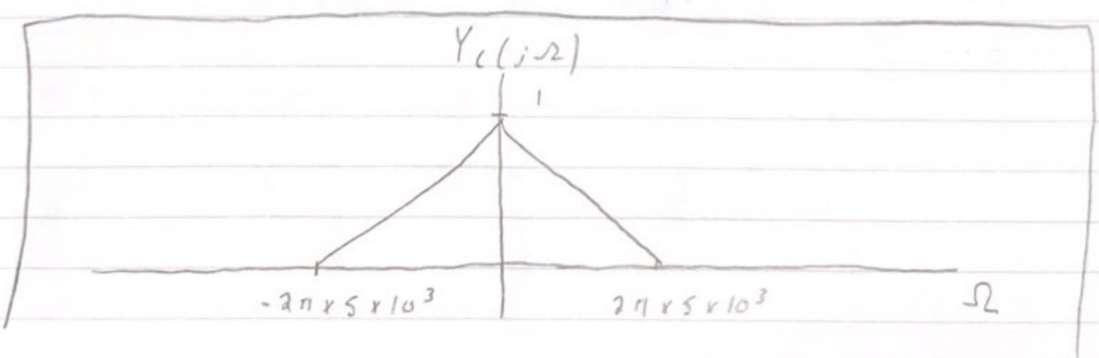
$$H_{eff}(j\Omega) = \begin{cases} 1, & |\Omega| < \frac{\pi}{2} \times 10^4 \\ 0, & \text{otherwise} \end{cases}$$



b) $\frac{1}{T_1} = \frac{1}{T_2} = 2 \times 10^4$

$\frac{\pi}{T} = 2\pi \times 10^4 > \Omega_N = \pi \times 10^4$, Nyquist is satisfied.

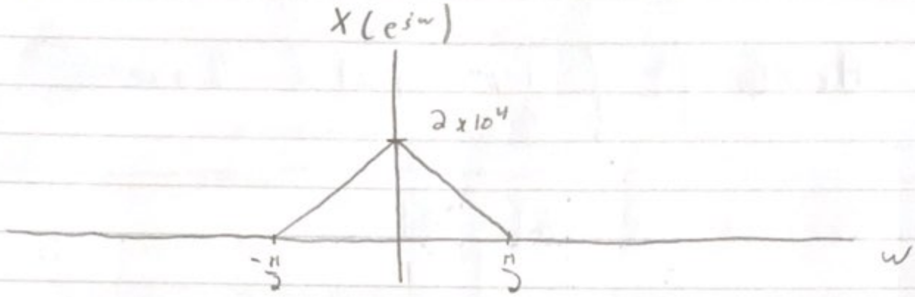
$$H_{eff}(j\Omega) = \begin{cases} 1, & |\Omega| < \pi \times 10^4 \\ 0, & \text{otherwise} \end{cases}$$



c) $\frac{1}{T_1} = 2 \times 10^4$, $\frac{1}{T_2} = 10^4$

Since $T_1 \neq T_2$, we must go through cascade

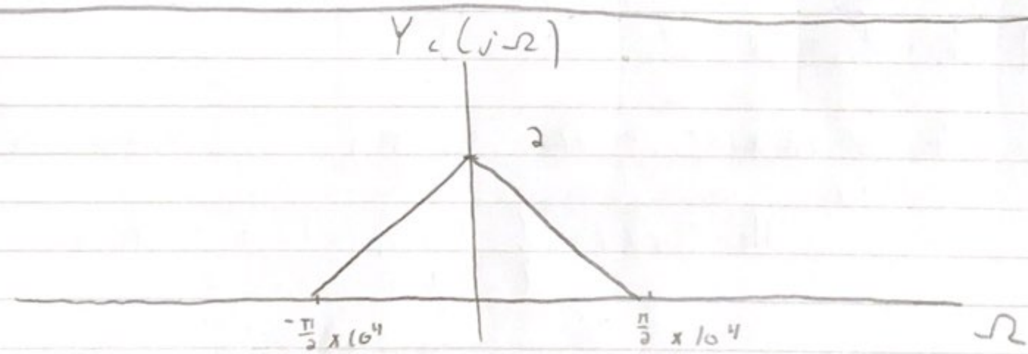
$\Omega_s = 4\pi \times 10^4$, $\Omega_N = \pi \times 10^4$



$$Y(e^{j\omega}) = X(e^{j\omega}) H(e^{j\omega}) = X(e^{j\omega})$$

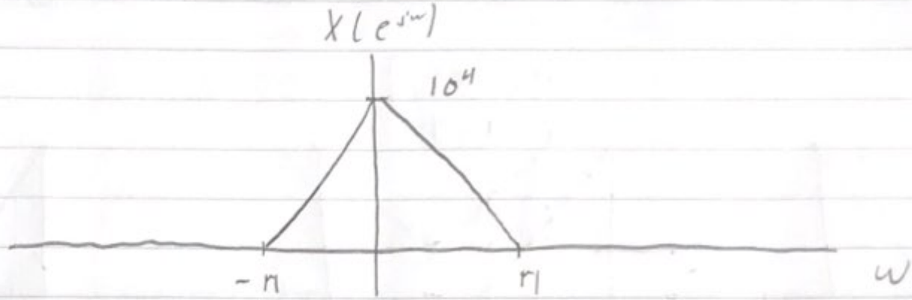
$$Y_c(j\Omega) = H_r(j\Omega) Y(e^{j\Omega T_2})$$

$$= \begin{cases} T_2 X(e^{j\Omega T_2}), & |\Omega| < \pi/T_2 \\ 0, & \text{otherwise} \end{cases}$$

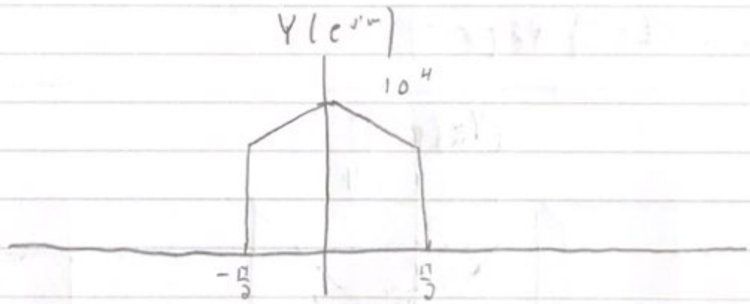


d) $\frac{1}{T_1} = 10^4$, $\frac{1}{T_2} = 2 \times 10^4$

$\Omega_s = 2\pi \times 10^4$, $\Omega_N = \pi \times 10^4$

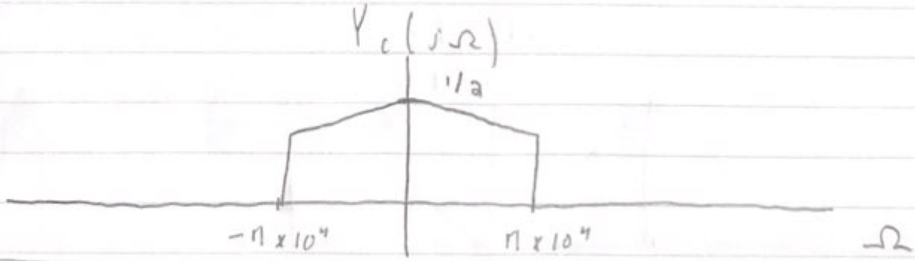


$Y(e^{j\omega}) = H(e^{j\omega}) X(e^{j\omega})$



$Y_c(j\Omega) = H_r(j\Omega) \, Y(e^{j\Omega T_2})$

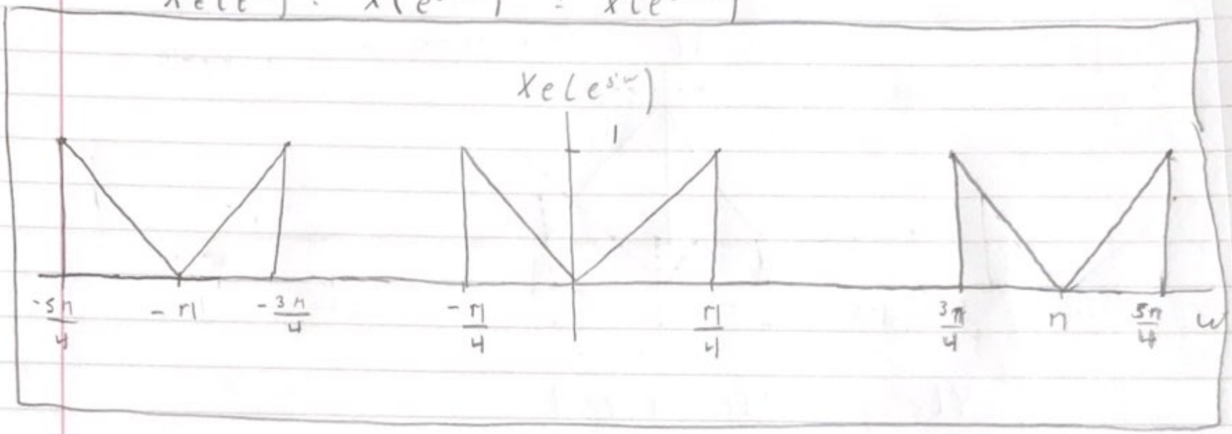
$$= \begin{cases} T_2 \, Y(e^{j\Omega T_2}), & |\Omega| < \pi / T_2 \\ 0, & \text{otherwise} \end{cases}$$



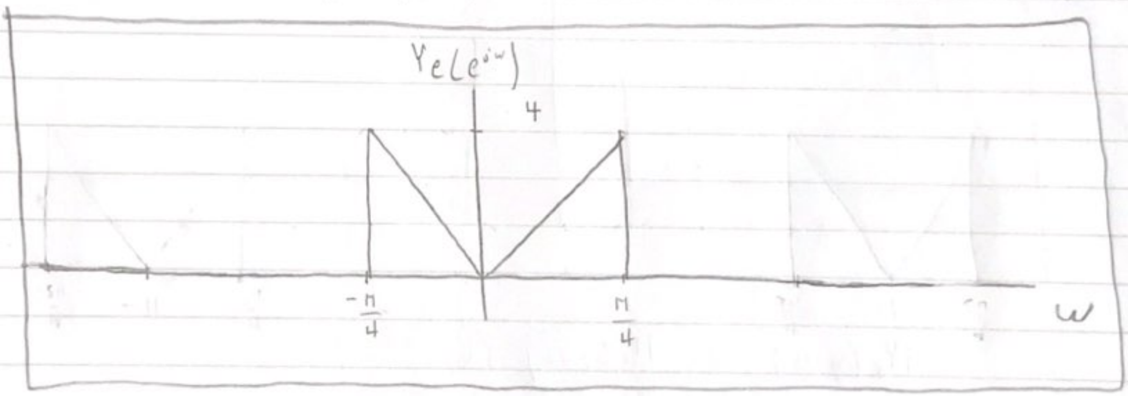
4.32

a) $L = 2, \quad M = 4$

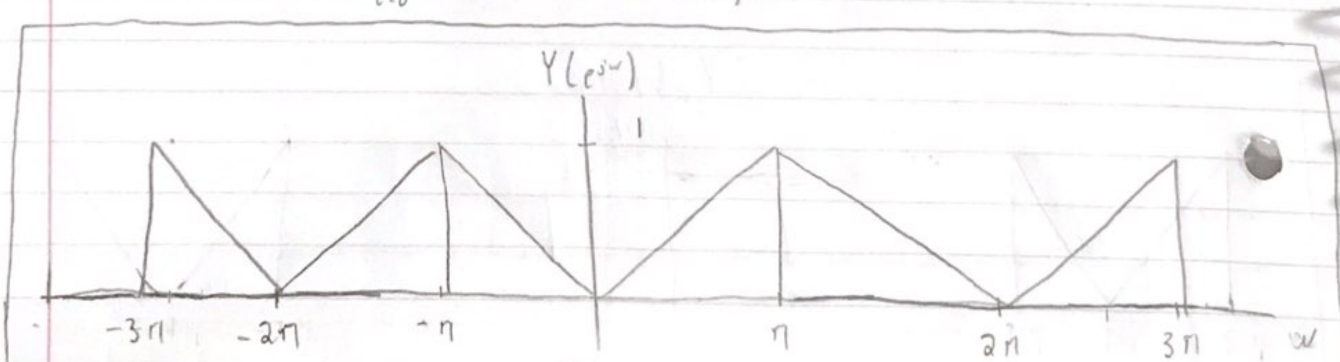
$$X_e(e^{j\omega}) = X(e^{j\omega L}) = X(e^{j2\omega})$$



$$Y_e(e^{j\omega}) = H(e^{j\omega})\, X_e(e^{j\omega})$$

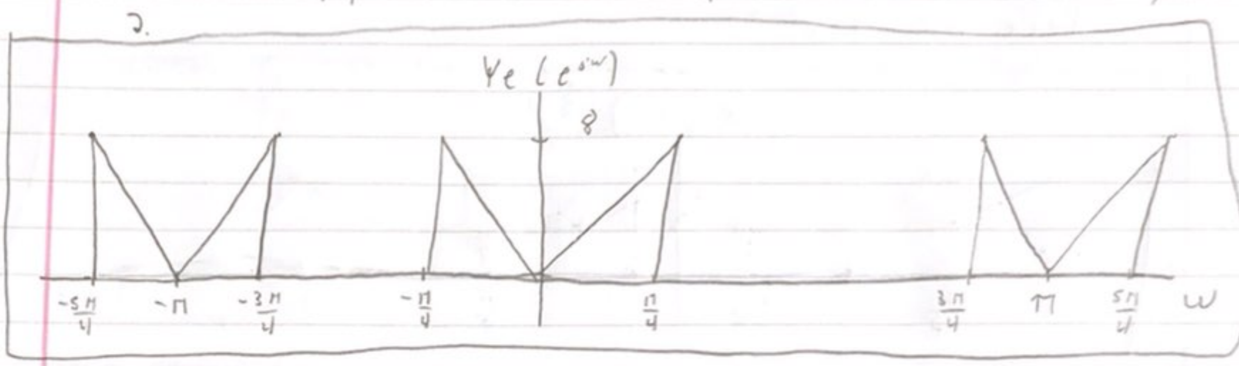


$$Y(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} Y_e\left(e^{j(\omega/M - 2\pi i/M)}\right)$$

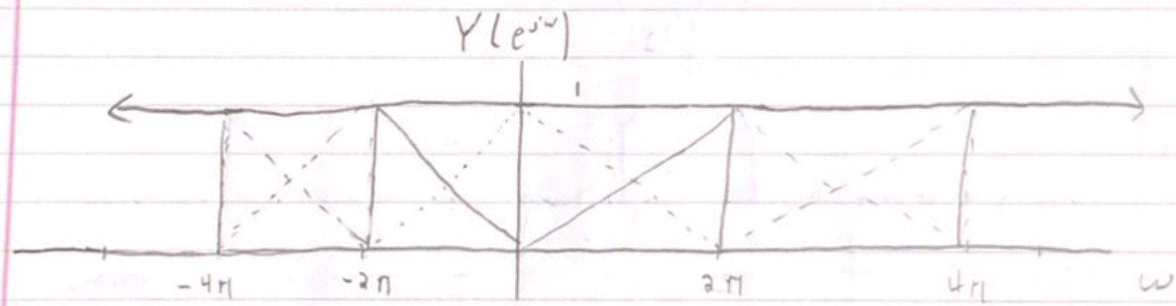


b) $L = 2, \quad M = 8$

$X_e(e^{jw})$ remains the same, and $Y_e(e^{jw})$ has the same shape, but the amplitude is scaled by 2.



$$Y(e^{jw}) = \frac{1}{M} \sum_{i=0}^{M-1} Y_e(e^{j(w/M - 2\pi i/M)})$$



As we can see, due to the expansion and periodization, $Y(e^{jw}) = 1$ for all w

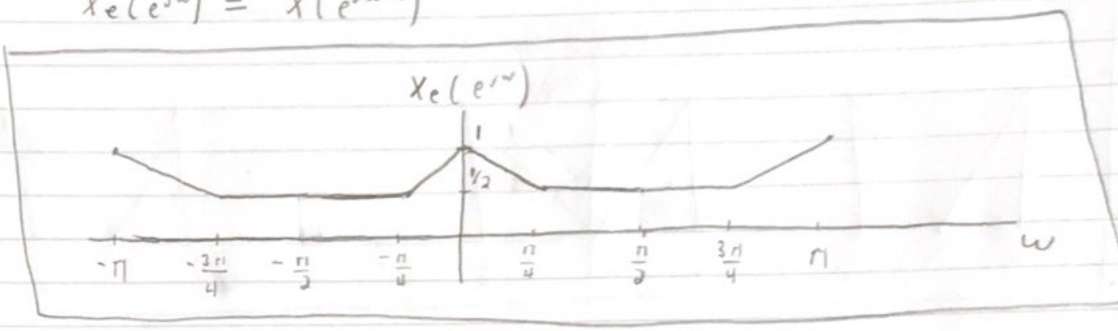
Using the transform pair:

$$\boxed{y[n] = \delta[n]}$$

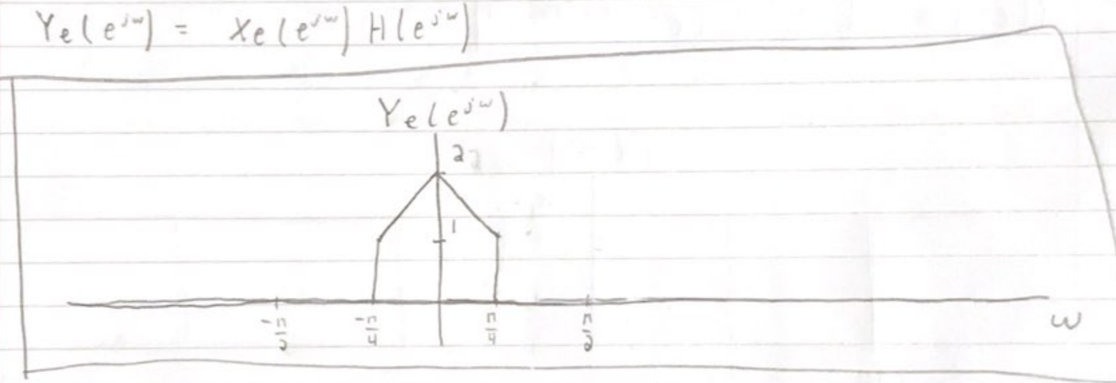
4.40

a) $M = 2$

$X_e(e^{j\omega}) = X(e^{j\omega M})$

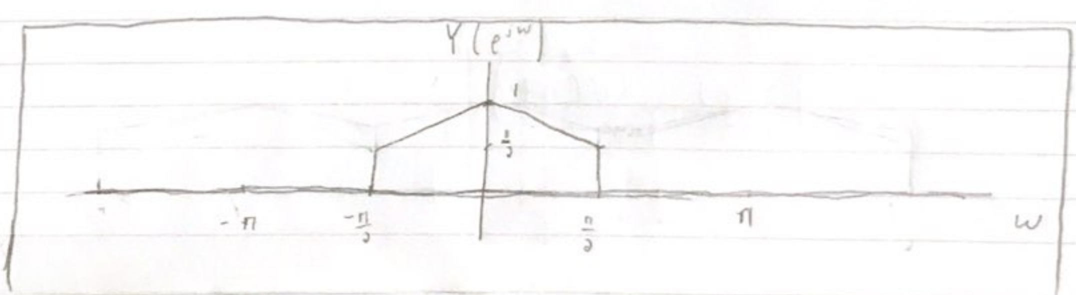


$Y_e(e^{j\omega}) = X_e(e^{j\omega}) H(e^{j\omega})$



$$Y(e^{j\omega}) = \frac{1}{M} \sum_{i=0}^{M-1} Y_e(e^{j(\omega/M - 2\pi i/M)})$$

$$= \frac{1}{2}\left(Y_e(e^{j\omega/2}) + Y_e(e^{j(\omega/2 - \pi)})\right)$$



b) To find $\mathcal{E}$, we can define a function $z[n] = x[n] - y[n]$. The DTFT of $z[n]$, $Z(e^{j\omega})$, is defined by

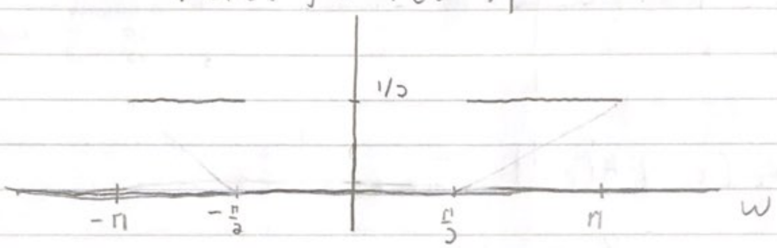
$$Z(e^{j\omega}) = X(e^{j\omega}) - Y(e^{j\omega})$$

Additionally, Parseval's theorem tells us that

$$|z[n]|^2 = |Z(e^{j\omega})|^2$$

Therefore, $$\mathcal{E} = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(e^{j\omega}) - Y(e^{j\omega})|^2 \, d\omega$$

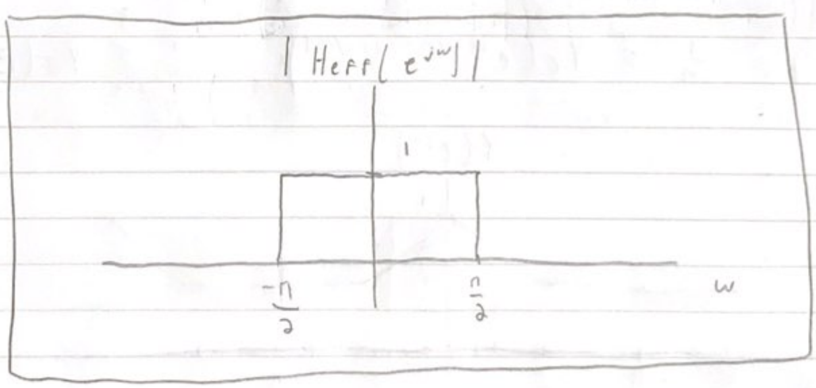
$$|X(e^{j\omega}) - Y(e^{j\omega})|$$



$$\mathcal{E} = \frac{1}{2\pi} \cdot 2\int_{\frac{\pi}{2}}^{\pi} \left|\tfrac{1}{2}\right|^2 d\omega = \frac{1}{4\pi}\left(\pi - \frac{\pi}{2}\right) = \frac{1}{8}$$

$$\boxed{\mathcal{E} = \frac{1}{8}}$$

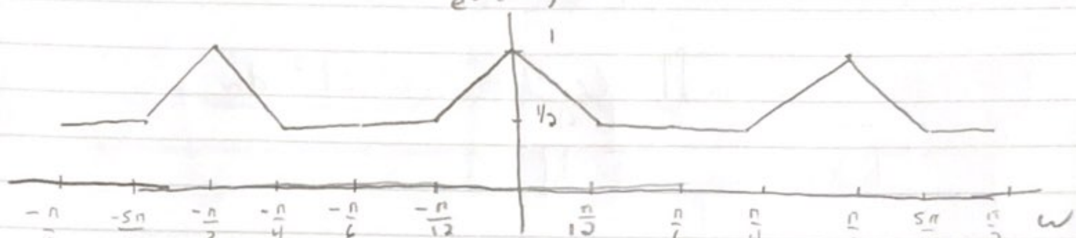
c) Heff is simply a low-pass filter of height 1:



d) In this case, downsampling by $M=6$ will result in aliasing, so we must compute the cascade to find $H_{eff}(e^{j\omega})$

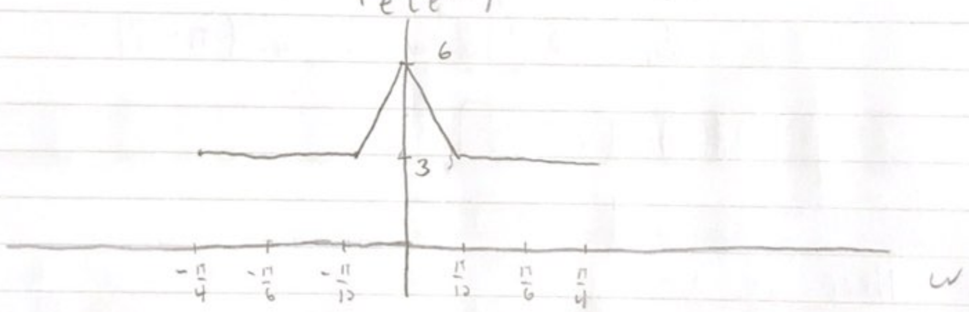
$$X_e(e^{j\omega}) = X(e^{j\omega M})$$

$X_e(e^{j\omega})$



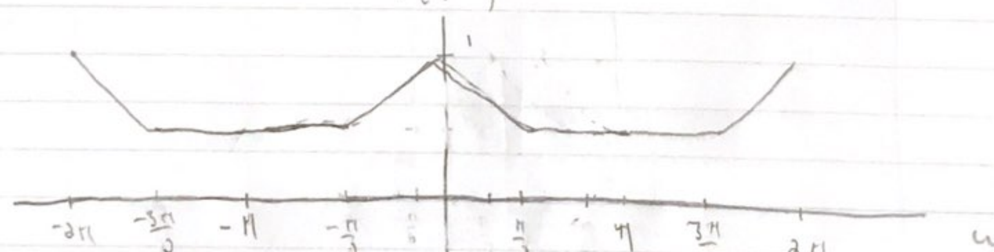
$$Y_e(e^{j\omega}) = X_e(e^{j\omega})\,H(e^{j\omega})$$

$Y_e(e^{j\omega})$



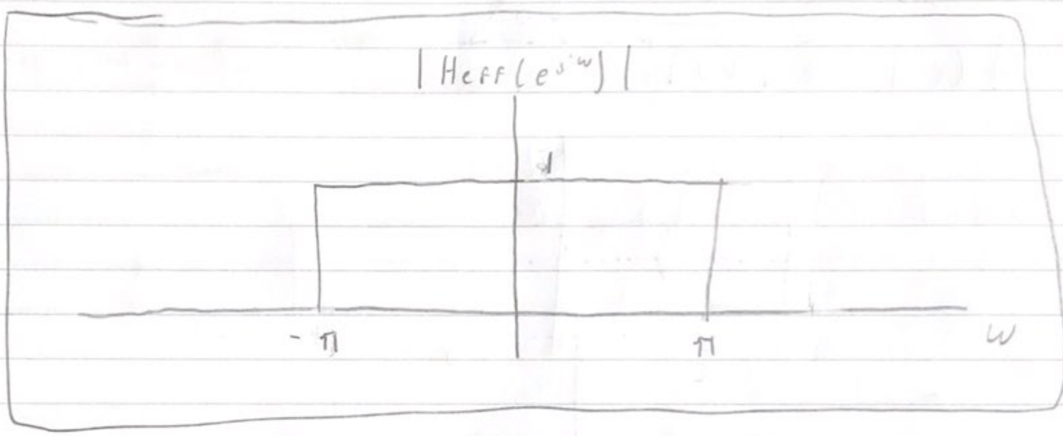
$$Y(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} Y_e\left(e^{j(\omega/M - 2\pi i/M)}\right) = \frac{1}{6}\sum_{i=0}^{5} Y_e\left(e^{j(\omega/6 - 2\pi i/6)}\right)$$

$Y(e^{j\omega})$



We are told this is an LTI system, so

$$|H_{eff}(e^{jw})| = \begin{cases} 1, & |w| \leq \pi \\ 0, & \text{elsewhere} \end{cases}$$

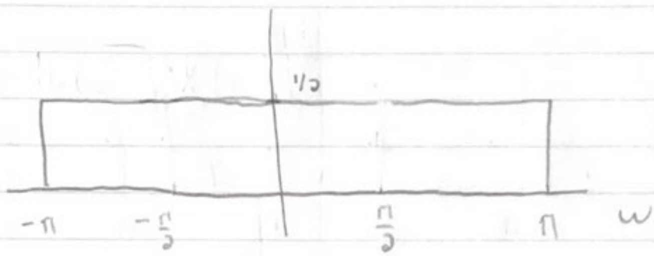


4.49

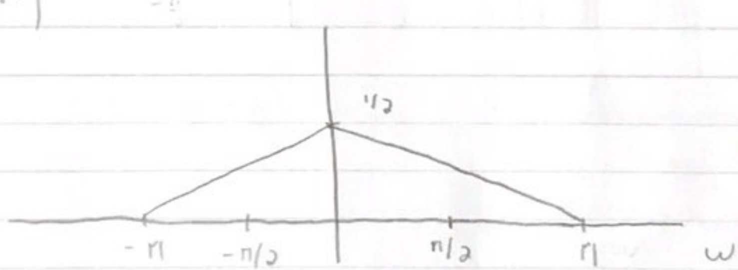
If we assume a bandlimited signal input, then we can see that the decimation and interpolation processes by the same rate cancel each other out in both loops. Therefore, the result $G(e^{j\omega})$ can be found by summing the result of both processes

Upper cascade:

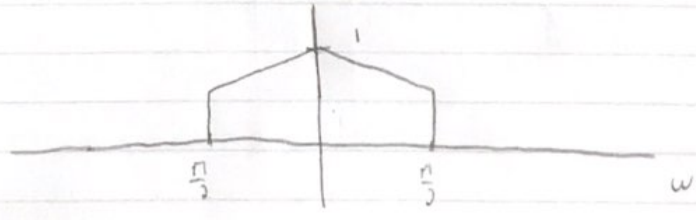
$H_0(e^{j\omega}) \rightarrow \boxed{\downarrow 2}$



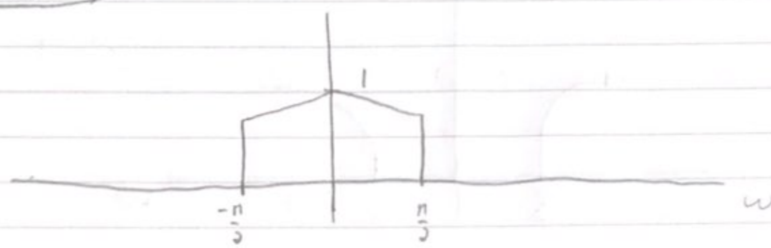
$\rightarrow Q_0(e^{j\omega})$



$\rightarrow \boxed{\uparrow 2}$

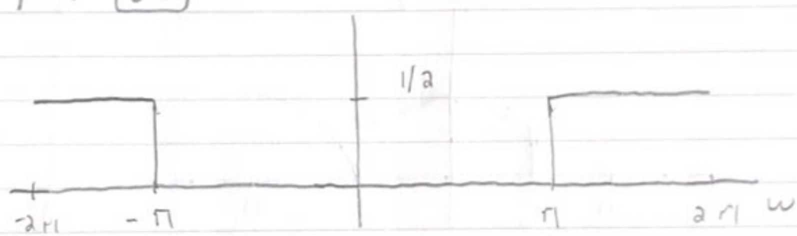


$\rightarrow \boxed{H_0(e^{j\omega})}$

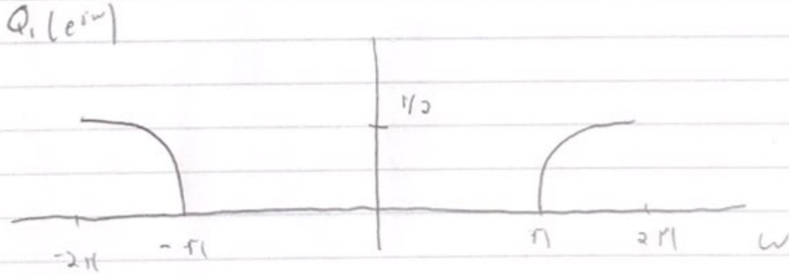


<u>Lower cascade</u>

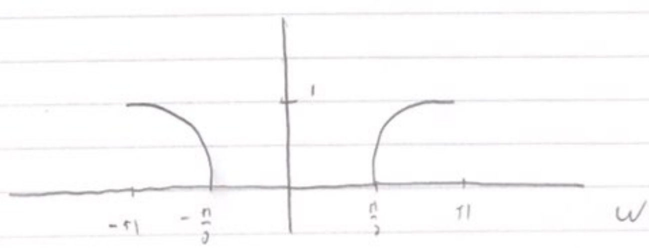
$H_1(e^{j\omega}) \rightarrow \boxed{\downarrow 2}$



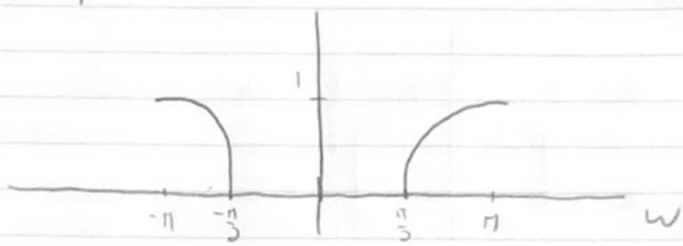
$\rightarrow Q_1(e^{j\omega})$



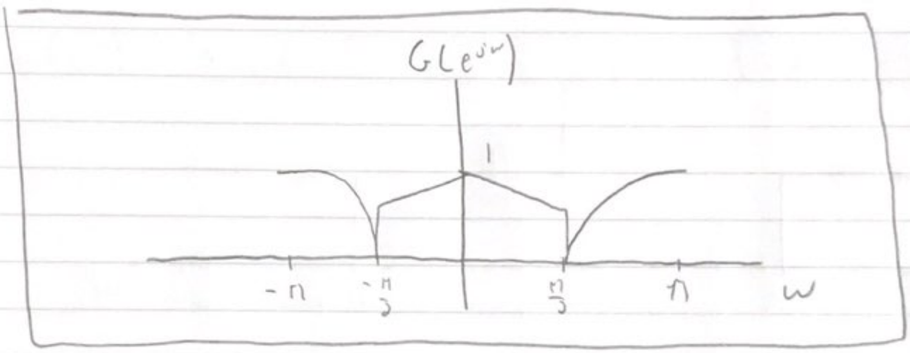
$\rightarrow \boxed{\uparrow 2}$



→ $H_1(e^{jw})$



Summation :



```python
# ESE 531: HW3 Problem 2

# libraries
import numpy as np
import matplotlib.pyplot as plt

# part a

# generate a simulated sinusoid analog signal

# parameters
fsim = 80000 # Hz
N_samples = 950
T = N_samples / fsim # sec
fo = 1000 # Hz

x_t = np.cos([2 * np.pi * fo * t for t in np.arange(0, T, 1/fsim)])

# plot analog signal
plt.plot(np.arange(0, T, 1/fsim), x_t)
plt.xlabel("t (sec)")
plt.ylabel("$x(t)$")
plt.title("Simulated Sinusoid Analog Signal")
plt.show()

# part b

# compute and plot continuous FT of the x(t)
def fmagplot(xa, dt):
   L = len(xa)
   Nfft = round(2 ** (np.log2(5 * L)))
   Xa = np.fft.fft(xa, Nfft)
   r = np.arange(0, Nfft/4)
   ff = r / Nfft / dt
   return ff , np.abs(Xa[:len(r)])

x, y = fmagplot(x_t, T)

plt.plot(x, y)
plt.title("Continuous Time FT of Analog Sinusoid")
plt.xlabel("$\Omega$ (kHz)")
plt.ylabel("$|X(j\Omega)|$")
plt.show()

# part c

# generate sampled signal
fs = 8000 # Hz
L = int(fsim / fs)
x_n = np.array([x_t[n] for n in np.arange(0, len(x_t), L)])

plt.stem(x_n)
plt.title("Sampled Sinusoid")
plt.xlabel("n")
plt.ylabel("x[n]")
plt.show()

# part d

# compute and plot the DTFT of discrete signal
X_d = np.fft.fft(x_n, len(x_n))
# X_d = np.roll(X_d, int(len(X_d)/2))

plt.plot(np.arange(0, fs - 1, fs/(len(X_d))), np.abs(X_d))
plt.title("DTFT of Discrete Sinusoid")
plt.xlabel("$\omega$ (Hz)")
plt.ylabel("$X(e^{j\omega})$")
```

```
plt.show()
```

```python
# ESE 531: HW3 Problem 3

# libraries
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import cheby2
from scipy.signal import freqz

# parameters
fsim = 80000 # Hz
N_samples = 950
T = N_samples / fsim # sec
fo = 1000 # Hz

# continuous signal
x_t = np.cos([2 * np.pi * fo * t for t in np.arange(0, T, 1/fsim)])

# generate sampled signal
fs = 8000 # Hz
L = int(fsim / fs)
x_n = np.array([x_t[n] for n in np.arange(0, len(x_t), L)])

# part a

# design and implement reconstruction filter
fsim = 80000 # Hz
fs = 8000 # Hz
fcut = 2 * (fs / 2) / fsim # Hz

b, a = cheby2(9, 60, fcut)
w, h = freqz(b, a, whole=True)
w -= np.pi
w *= fs / (2 * np.pi)

# plot the magnitude
plt.plot(w, np.abs(h))
plt.title("Magnitude of Reconstruction Frequency Response")
plt.xlabel("$\Omega$ (Hz)")
plt.ylabel("$|X(j\Omega)|$")
plt.show()

# plot the angle
plt.plot(w, np.unwrap(np.angle(h)))
plt.title("Phase of Reconstruction Frequency Response")
plt.xlabel("$\Omega$ (Hz)")
plt.ylabel("$\measuredangle X(j\Omega)$")
plt.show()

# part b

# zero insert operation
x_prime = np.zeros(len(x_t))

# create zero-padded signal
for i, val in enumerate(x_n):
    x_prime[int(i * len(x_t) / len(x_n))] = x_n[i]

# apply cheby2 filter to generate reconstructed output
x_r = np.convolve(x_prime, np.fft.ifft(h), mode='same')

# plot reconstructed signal
plt.plot(np.arange(0, T, 1/fsim), x_r)
plt.xlabel("t (sec)")
plt.ylabel("$x_r(t)$")
plt.title("Reconstructed Sinusoid Analog Signal")
plt.show()
```

```python
# compute and plot continuous FT of the x(t)
def fmagplot(xa, dt):
    L = len(xa)
    Nfft = round(2 ** (np.log2(5 * L)))
    Xa = np.fft.fft(xa, Nfft)
    r = np.arange(0, Nfft/4)
    ff = r / Nfft / dt
    return ff , np.abs(Xa[:len(r)])

x, y = fmagplot(x_r, T)

plt.plot(x, y)
plt.title("Continuous Time FT of Reconstructed Analog Sinusoid")
plt.xlabel("$\Omega$ (kHz)")
plt.ylabel("$|X(j\Omega)|$")
plt.show()
```