TTT4120 Digital Signal Processing

Problem Set 2

Peter Pham

2023-09-15

Contents

Problem 1 (3 points)

Two signals $x[n]$ and $y[n]$ are given by

$$x[n] = \begin{cases} 2 & n = 0 \\ 1 & n = \pm 1 \\ 0 & \text{otherwise,} \end{cases} \qquad y[n] = \begin{cases} 1 & -M \leq n \leq M \\ 0 & \text{otherwise} \end{cases}$$

(a) Show that the Fourier transform of $x[n]$ is given by... and sketch it for $\omega \in [-\pi, \pi]$.

$$X(\omega) = 2 + 2\cos\omega$$

We have that

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

In our case we can write

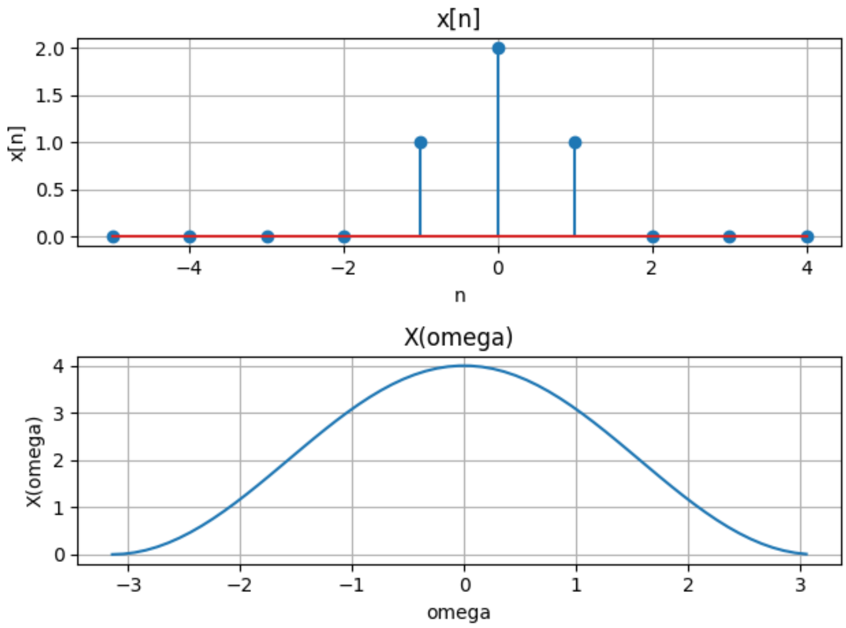
$$X(\omega) = \sum_{n=-1}^{1} x[n]e^{-j\omega n} = -e^{j\omega} + 2 + e^{-j\omega}$$

From Euler's formula we get that

$$\cos x = \frac{e^{ix} + e^{-ix}}{2}$$

by combining these two we get:

$$X(\omega) = 2 + 2\cos\omega$$



(b) Show that the Fourier transform of y[n] is given by

$$Y(\omega) = \frac{\sin\left(\omega\left(M+\frac{1}{2}\right)\right)}{\sin\left(\frac{\omega}{2}\right)}$$

and sketch it for $M = 10$ **and** $\omega \in [-\pi, \pi]$

$$Y(\omega) = \sum_{n=-M}^{M} e^{-j\omega n}$$

This is a finite geometric series. By using the general formula for a geometric series

$$\sum_{k=m}^{n} ar^k = \begin{cases} a(n-m+1) & \text{if } r = 1 \\ \frac{a\left(r^m - r^{n+1}\right)}{1-r} & \text{if } r \neq 1 \end{cases}$$

we get

$$Y(\omega) = \frac{e^{j\omega M} - e^{-j\omega(M+1)}}{1 - e^{-j\omega}}$$

by multiplying the numerator and the denominator by $e^{\frac{j\omega}{2}}$.

$$
\begin{aligned}
Y(\omega) &= \frac{e^{j\omega M} e^{j\omega/2} - e^{-j\omega(M+1)} e^{j\omega/2}}{(1 - e^{-j\omega})\, e^{j\omega/2}} \\
&= \frac{e^{j\omega\left(M+\frac{1}{2}\right)} - e^{-j\omega\left(M+\frac{1}{2}\right)}}{e^{j\omega/2} - e^{-j\omega/2}}
\end{aligned}
$$

Since we have expresions of the form $e^{j\Theta} - e^{-j\Theta}$, which can be simplidied using Euler's formula

$$
\begin{aligned}
e^{j\theta} &= \cos(\theta) + j\sin(\theta) \\
e^{-j\theta} &= \cos(-\theta) + j\sin(-\theta) = \cos(\theta) - j\sin(\theta)
\end{aligned}
$$

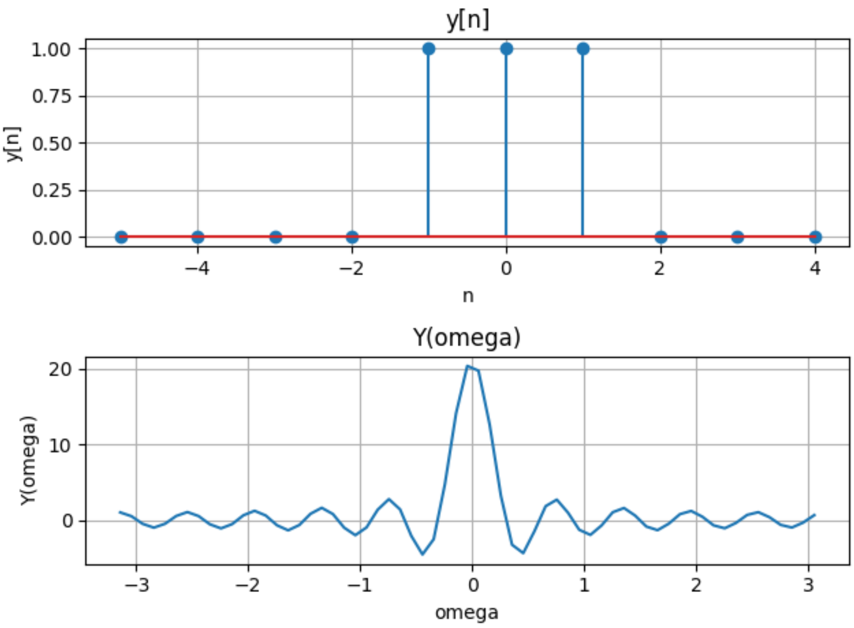
Therefore:

$$
e^{j\Theta} - e^{-j\Theta} = 2j\sin\Theta
$$

By substituing this back into the equation, we get

$$
Y(\omega) = \frac{2j\sin\left(\omega\left(M+\frac{1}{2}\right)\right)}{2j\sin\left(\frac{\omega}{2}\right)}
$$

$$
Y(\omega) = \frac{\sin\left(\omega\left(M+\frac{1}{2}\right)\right)}{\sin\left(\frac{\omega}{2}\right)}
$$



(c) Explain why the signals $x[n]$ and $y[n]$ have real valued spectra.

The signals have a real valued spectra because the sequencences $x[n]$ and $y[n]$ are real and symmetric.

(d) Let the signal

$$z[n] = \sum_{l=-\infty}^{\infty} x[n - lN]$$

be the periodic extension of $x[n]$. Assume that N is greater than the length of the signal, i.e. $N > 3$.
Sketch the signal $z[n]$.
Find the Fourier series coefficients $\{c_k\}$ of $z[n]$.
Sketch $\{c_k\}$ as a function of $\omega = 2\pi \frac{k}{N} \in [-\pi, \pi]$ for $N = 10$.

For a discrete-time periodic signal, the Fourier Series coefficients $\{c_k\}$ can be defined as:

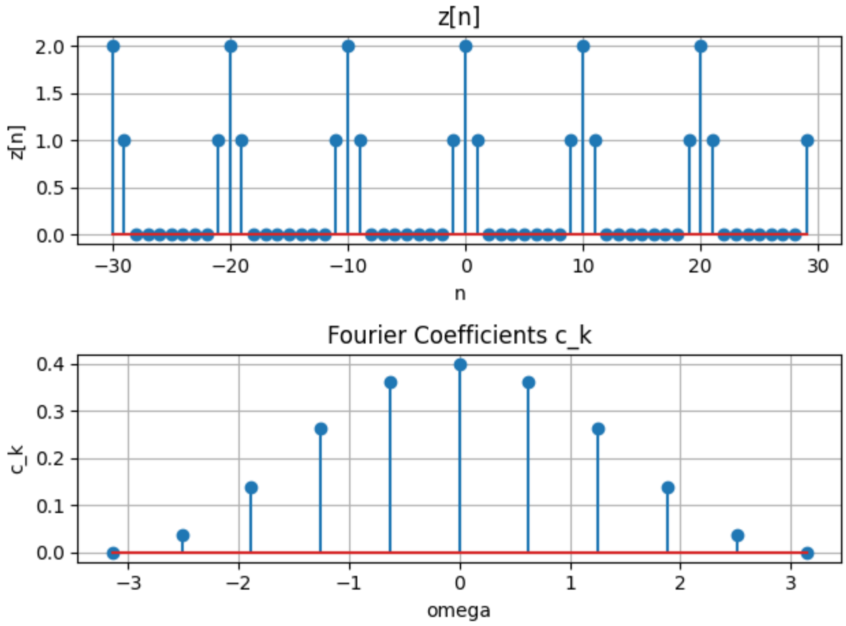
$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} z[n] e^{-j(2\pi kn)/N}$$

For $z[n]$, you can see that most terms in this sum will be zero, except for the few terms that correspond to the non-zero values of $x[n]$. Thus:

$$c_k = \frac{1}{N} \left(2 + e^{-j(2\pi k)/N} + e^{j(2\pi k)/N}\right)$$

Simplifying this using Euler's formula $e^{j\theta} + e^{-j\theta} = 2\cos(\theta)$, we get:

$$c_k = \tfrac{1}{N} \left(2 + 2\cos\left(\tfrac{2\pi k}{N}\right)\right)$$
$$c_k = \tfrac{2}{N} \left(1 + \cos\left(\tfrac{2\pi k}{N}\right)\right)$$



(e) Compare the spectra of $x[n]$ and $z(n)$, i.e. $X(\omega)$ and $\{c_k\}$. What is the relationship between the spectra?

$X(\omega)$ and $\{c_k\}$ are essentially two different ways of representing the frequency content of a signal. $X(\omega)$ is used for the original aperiodic $x[n]$, while $\{c_k\}$ is used for the periodic $z[n]$, which is formed by periodically repeating $x[n]$.

Problem 2 (1.5 points)

Let $x[n]$ be a signal with the Fourier transform $X(\omega)$. Find the fourier tranforms of the following signals: We have that

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

(a) $x_1[n] = x[n+3]$

The Fourier Transform of a time-shifted signal $x[n-m]$ is given by:

$$\mathcal{F}\{x[n-m]\} = e^{-j\omega m}X(\omega)$$

$$\Rightarrow X_1(\omega) = e^{-j\omega m}X(\omega) = e^{j3\omega}X(\omega)$$

(b) $x_2[n] = x[-n]$

The Fourier Transform of a time-reversed signal $x[-n]$ is given by:

$$\mathcal{F}\{x[-n]\} = X(-\omega)$$

$$\Rightarrow X_2(\omega) = X(-\omega)$$

(c) $x_3[n] = x[3-n]$

Here we have both time-shift and time-reversed

$$X_3(\omega) = e^{-j\omega m}X(\omega) = e^{j3\omega}X(-\omega)$$

(d) $x_4[n] = x[n] * x[n]$

Covolution in the time domain corresponds to multiplication in the frequency domain

$$X_4(\omega) = X(\omega) \cdot X(\omega) = X^2(\omega)$$

Problem 3 (3 points)

Two systems (from Problem Set 1) are given by the following difference equations

$$y[n] = x[n] + 2x[n-1] + x[n-2]$$
$$y[n] = -0.9y[n-1] + x[n].$$

(a) Find the frequency responses of these two systems.

To find the frequency response $H(\omega)$ of a system described by a difference equation, you can take the Fourier Transform of both sides of the equation and solve for $H(\omega)$, which is essentially $\frac{Y(\omega)}{X(\omega)}$.

For the first system:

The difference equation is:

$$y[n] = x[n] + 2x[n-1] + x[n-2]$$

Taking the Fourier Transform of both sides:

$$Y(\omega) = X(\omega) + 2e^{-j\omega}X(\omega) + e^{-2j\omega}X(\omega)$$

Simplifying:

$$Y(\omega) = X(\omega)(1 + 2e^{-j\omega} + e^{-2j\omega})$$

The frequency response $H_1(\omega)$ is:

$$H_1(\omega) = \frac{Y(\omega)}{X(\omega)} = 1 + 2e^{-j\omega} + e^{-2j\omega}$$

For the second system:

The difference equation is:

$$y[n] = -0.9y[n-1] + x[n]$$

Taking the Fourier Transform of both sides:

$$Y(\omega) = -0.9e^{-j\omega}Y(\omega) + X(\omega)$$

Solving for $Y(\omega)$:

$$Y(\omega) = \frac{X(\omega)}{1 + 0.9e^{-j\omega}}$$

The frequency response $H_2(\omega)$ is:

$$H_2(\omega) = \frac{Y(\omega)}{X(\omega)} = \frac{1}{1 + 0.9e^{-j\omega}}$$

Find the magnitude and phase responses of the two systems. Are they even or odd functions?

To find the magnitude and phase of a frequency response $H(\omega)$, we can express $H(\omega)$ in polar form: $H(\omega) = |H(\omega)|e^{j\angle H(\omega)}$ where $|H(\omega)|$ is the magnitude and $\angle H(\omega)$ is the phase.

System 1

The difference equation is $y[n] = x[n] + 2x[n-1] + x[n-2]$.

The Fourier Transform of the system, known as the frequency response $H_1(\omega)$, is given by:

$$H_1(\omega) = 1 + 2e^{-j\omega} + e^{-2j\omega}$$

To find the magnitude $|H_1(\omega)|$:

$$\begin{aligned}|H_1(\omega)| &= \sqrt{\mathrm{Re}[H_1(\omega)]^2 + \mathrm{Im}[H_1(\omega)]^2} \\ &= \sqrt{(1 + 2\cos(\omega) + \cos(2\omega))^2} \\ &= \sqrt{2 + 4\cos(\omega) + 4\cos^2(\omega) + 2\cos(2\omega)} \\ &= \sqrt{2(1 + 2\cos(\omega) + 2\cos^2(\omega) + \cos(2\omega))}\end{aligned}$$

To find the phase $\angle H_1(\omega)$:

$$\angle H_1(\omega) = 0$$

To determine whenever the function is even or odd

$$H_1(-\omega) = 1 + 2e^{j\omega} + e^{2j\omega}$$

$H_1(\omega)$ and $H_1(-\omega)$ are not equal, nor are $H_1(\omega)$ and $-H_1(-\omega)$- Thus, $H_1(\omega)$ is neither even nor odd.2.

System 2

The difference equation is $y[n] = -0.9y[n-1] + x[n]$.

The frequency response $H_2(\omega)$ is:

$$H_2(\omega) = \frac{1}{1 + 0.9e^{-j\omega}}$$

To find the magnitude $|H_2(\omega)|$:

$$|H_2(\omega)| = \left|\frac{1}{1+0.9(\cos(-\omega) - j\sin(-\omega))}\right|$$
$$= \frac{1}{\sqrt{(1-0.9\cos(\omega))^2 + (0.9\sin(\omega))^2}}$$
$$= \frac{1}{\sqrt{1-1.8\cos(\omega)+0.81}}$$

To find the phase $\angle H_2(\omega)$:

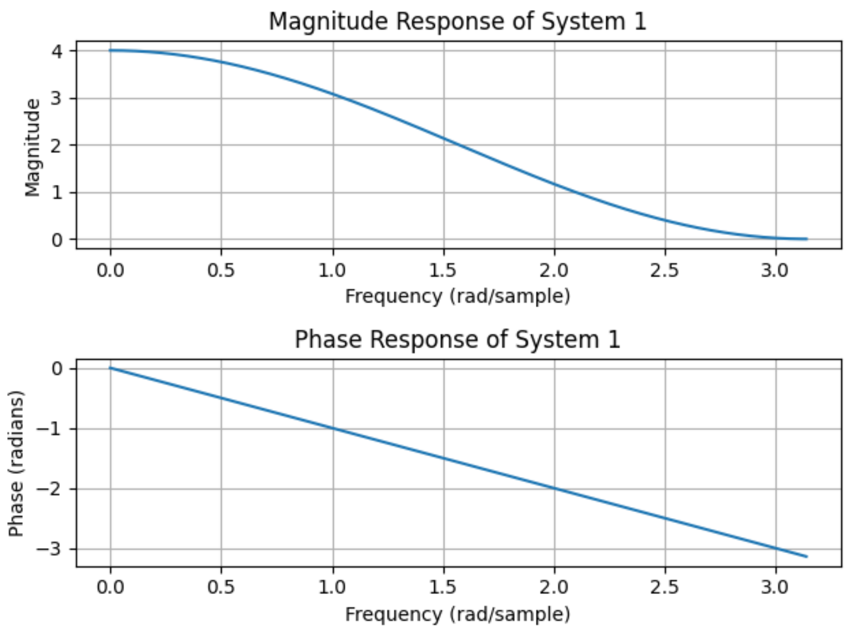
$$\angle H_2(\omega) = -\text{atan2}(0.9\sin(\omega), 1-0.9\cos(\omega))$$

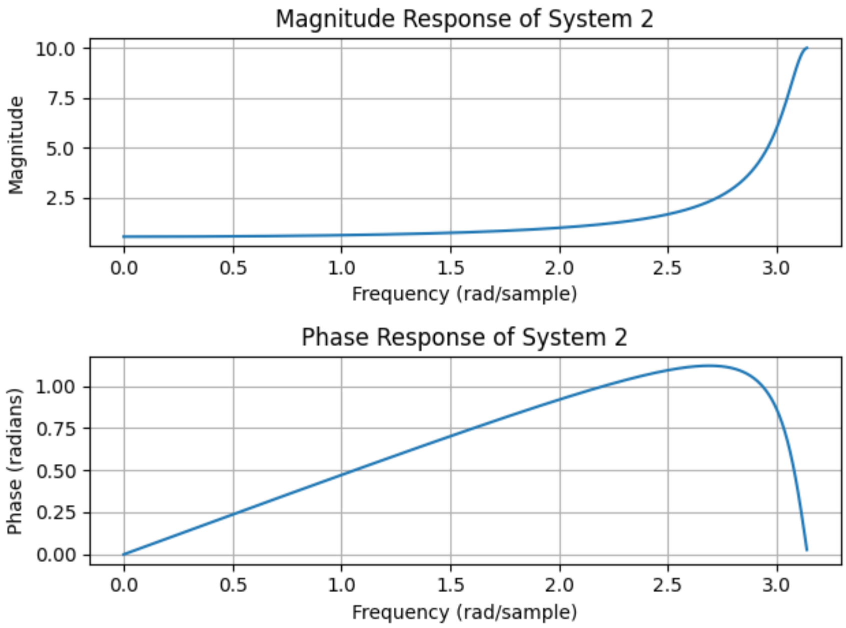
To determine whenever the function is even or odd

$$H_2(-\omega) = \frac{1}{1+0.9e^{j\omega}}$$

$H_2(\omega)$ and $H_2(-\omega)$ are not equal, nor are $H_2(\omega)$ and $-H_2(-\omega)$- Thus, $H_2(\omega)$ is neither even nor odd.

(c) Use Python (the functions scipy.signal.freqz, numpy.abs and numpy.angle) to find and plot the magnitude and phase responses of the systems.





(d) Determine whether each system represents a lowpass, bandpass, bandstop or highpass filter. Justify your answers.

From the plots, we can se that system 1 is a lowpass filter, while system 2 is a highpass filter. This is because the amplitude is high at a low frequency at system 1 and opposite at system 2.

The signal $x[n] = \frac{1}{2}\cos\left(\frac{\pi}{2}n + \frac{\pi}{4}\right)$ is passed through the two systems. Find the frequency, amplitude and phase of the corresponding output signals.

The signal $x[n] = \frac{1}{2}\cos\left(\frac{\pi}{2}n + \frac{\pi}{4}\right)$ can be represented as

$$x[n] = \frac{1}{2}\left(e^{j\left(\frac{\pi}{4}\right)}e^{j\left(\frac{\pi}{2}n\right)} + e^{-j\left(\frac{\pi}{4}\right)}e^{-j\left(\frac{\pi}{2}n\right)}\right)$$

The Fourier Transform $X(\omega)$ of $x[n]$ is:

$$X(\omega) = \frac{1}{4}\left(e^{j\left(\frac{\pi}{4}\right)}\delta\left(\omega - \frac{\pi}{2}\right) + e^{-j\left(\frac{\pi}{4}\right)}\delta\left(\omega + \frac{\pi}{2}\right)\right)$$

For the first system $y_1[n] = x[n] + 2x[n-1] + x[n-2]$

The frequency response $H_1(\omega)$ is:

$$H_1(\omega) = 1 + 2e^{-j\omega} + e^{-j2\omega} = 1 + 2\cos(\omega) - j2\sin(\omega) + \cos(2\omega) - j\sin(2\omega)$$

Thus, the output $Y_1(\omega)$ will be:

$$Y_1(\omega) = X(\omega) \cdot H_1(\omega)$$

$$Y_1(\omega) = \frac{1}{4}\left(e^{j\left(\frac{\pi}{4}\right)}H_1\left(\frac{\pi}{2}\right)\delta\left(\omega - \frac{\pi}{2}\right) + e^{-j\left(\frac{\pi}{4}\right)}H_1\left(-\frac{\pi}{2}\right)\delta\left(\omega + \frac{\pi}{2}\right)\right)$$

For the second system $y_2[n] = -0.9y_2[n-1] + x[n]$

The frequency response $H_2(\omega)$ is:

$$H_2(\omega) = \frac{1}{1 + 0.9e^{-j\omega}} = \frac{1}{1 + 0.9\cos(\omega) - j0.9\sin(\omega)}$$

The output $Y_2(\omega)$ will then be:

$$Y_2(\omega) = X(\omega) \cdot H_2(\omega)$$

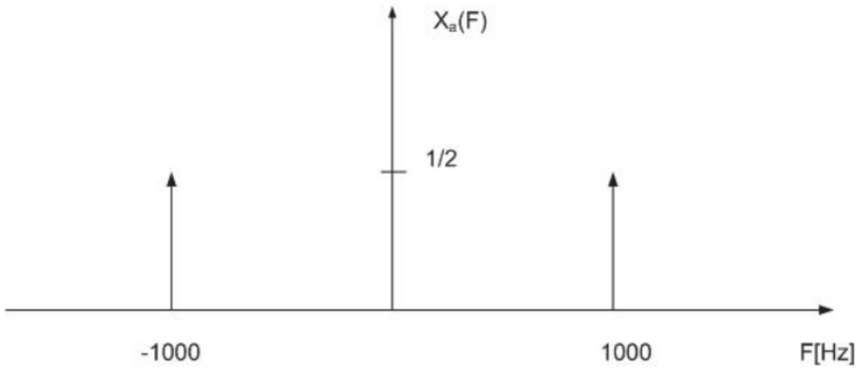
$$Y_2(\omega) = \frac{1}{4}\left(e^{j\left(\frac{\pi}{4}\right)}H_2\left(\frac{\pi}{2}\right)\delta\left(\omega - \frac{\pi}{2}\right) + e^{-j\left(\frac{\pi}{4}\right)}H_2\left(-\frac{\pi}{2}\right)\delta\left(\omega + \frac{\pi}{2}\right)\right)$$

Problem 4 (2 points)

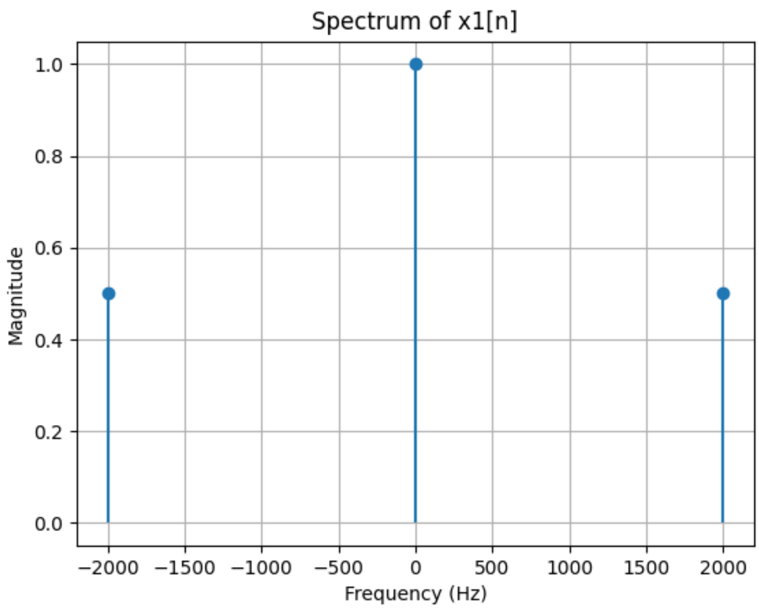
The analog signal $x_a(t)$ is given by

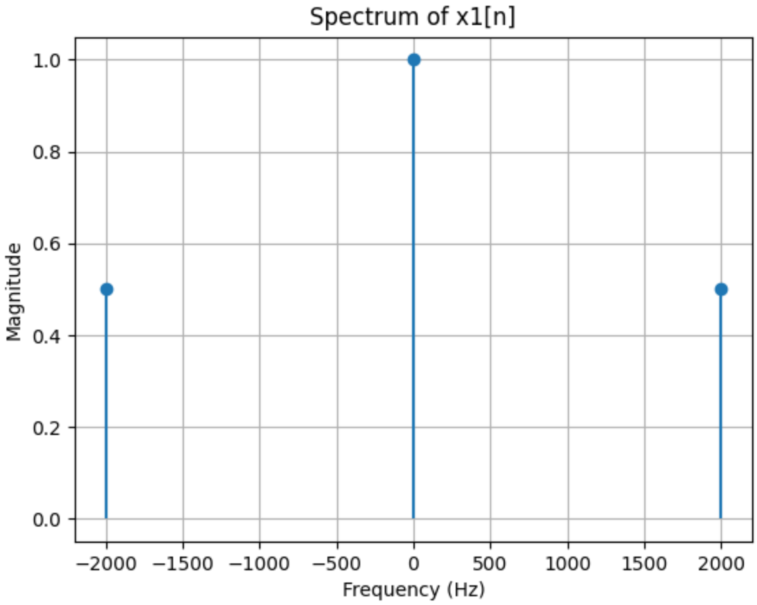
$$x_a(t) = \cos(2000\pi t).$$

The spectrum of $x_a(t)$ is shown in the figure below



(a) Two time-discrete signals, $x_1[n]$ and $x_2[n]$, are generated by sampling the signal $x_a[n]$ using the sampling frequencies $F_s = 4000$ Hz and $F_s = 1500$ Hz, respectively.Sketch the spectra of the two sampled signals, $X_1(f)$ and $X_2(f)$, for $f \in \left[-\frac{1}{2}, \frac{1}{2}\right]$.





(b) Use Matlab or Python to generate a segment of both time-discrete signals corresponding to the signal duration of 1 s**.Play the two signal segments using Matlab function sound or the Python command sounddevice.play with the corresponding sampling frequencies.Why do they sound differently when they both represent the same analog signal?**

No the first has a higher pitch than the second.

Appendix

Python Code for problem1

```python
import matplotlib.pyplot as plt
import numpy as np

# Function definitions
def x_n(n):
    if n == -1 or n == 1:
        return 1
    if n == 0:
        return 2
    else:
        return 0

def y_n(n, M):
    if -M <= n <= M:
        return 1
    else:
        return 0

def X_omega(omega):
    return 2 + 2 * np.cos(omega)

# Variables
omega = np.arange(-np.pi, np.pi, 0.1)
n_values = np.arange(-5, 5)
x_values = [x_n(n) for n in n_values]
y_values = [y_n(n, 1) for n in n_values]
X_values = [X_omega(om) for om in omega]

# Problem 1

# a)
def problem1a():
    plt.figure()
    plt.subplot(2, 1, 1)
    plt.stem(n_values, x_values)
    plt.xlabel('n')
```

```
    plt.ylabel('x[n]')
    plt.title('x[n]')
    plt.grid(True)

    plt.subplot(2, 1, 2)
    plt.plot(omega, X_values)
    plt.xlabel('omega')
    plt.ylabel('X(omega)')
    plt.title('X(omega)')
    plt.grid(True)

    plt.tight_layout()
    plt.show()
# problem1a()
# b)

def Y_omega(omega, M):
    return (np.sin(omega*(M+(1/2)))/np.sin(omega/2))
def problem1b():
    plt.figure()
    plt.subplot(2, 1, 1)
    plt.stem(n_values, y_values)
    plt.xlabel('n')
    plt.ylabel('y[n]')
    plt.title('y[n]')
    plt.grid(True)
   M=10
    Y_values = [Y_omega(om, M) for om in omega]
    plt.subplot(2, 1, 2)
    plt.plot(omega, Y_values)
    plt.xlabel('omega')
    plt.ylabel('Y(omega)')
    plt.title('Y(omega)')
    plt.grid(True)

    plt.tight_layout()
    plt.show()
# problem1b()

def z_n(n, N):
    n_mod = n % N
```

```
    if n_mod == 0:
        return 2
    elif n_mod == 1 or n_mod == N-1:
        return 1
    else:
        return 0

def c_k(k, N):
    return 2/N * (1 + np.cos(2 * np.pi * k / N))

def problem1d():
    N = 10
    n_values = np.arange(-30, 30)  # For example, from -30 to 30
    z_values = [z_n(n, N) for n in n_values]

    plt.figure()
    plt.subplot(2, 1, 1)
    plt.stem(n_values, z_values)
    plt.xlabel('n')
    plt.ylabel('z[n]')
    plt.title('z[n]')
    plt.grid(True)

    k_values = np.arange(-5, 6)  # From -N/2 to N/2
    omega_values = 2 * np.pi * k_values / N  # omega = 2*pi*k/N
    ck_values = [c_k(k, N) for k in k_values]

    plt.subplot(2, 1, 2)
    plt.stem(omega_values, ck_values)
    plt.xlabel('omega')
    plt.ylabel('c_k')
    plt.title('Fourier Coefficients c_k')
    plt.grid(True)

    plt.tight_layout()
    plt.show()

problem1d()
```

Python Code for problem2

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import freqz

# Frequency response for the first system: y[n] = x[n] + 2x[n-1] + x[n-2]
def system1():
    b = np.array([1, 2, 1])  # Coefficients for x[n]
    a = np.array([1])        # Coefficients for y[n]

    w, h = freqz(b, a, worN=1024)

    plt.figure()
    plt.subplot(2, 1, 1)
    plt.plot(w, np.abs(h))
    plt.xlabel('Frequency (rad/sample)')
    plt.ylabel('Magnitude')
    plt.title('Magnitude Response of System 1')
    plt.grid()

    plt.subplot(2, 1, 2)
    plt.plot(w, np.angle(h))
    plt.xlabel('Frequency (rad/sample)')
    plt.ylabel('Phase (radians)')
    plt.title('Phase Response of System 1')
    plt.grid()

    plt.tight_layout()
    plt.show()

# Frequency response for the second system: y[n] = -0.9y[n-1] + x[n]
def system2():
    b = np.array([1])        # Coefficients for x[n]
    a = np.array([1, 0.9])   # Coefficients for y[n]

    w, h = freqz(b, a, worN=1024)

    plt.figure()
    plt.subplot(2, 1, 1)
    plt.plot(w, np.abs(h))
    plt.xlabel('Frequency (rad/sample)')
    plt.ylabel('Magnitude')
```

```python
    plt.title('Magnitude Response of System 2')
    plt.grid()

    plt.subplot(2, 1, 2)
    plt.plot(w, np.angle(h))
    plt.xlabel('Frequency (rad/sample)')
    plt.ylabel('Phase (radians)')
    plt.title('Phase Response of System 2')
    plt.grid()

    plt.tight_layout()
    plt.show()

# Plot both systems
system1()
system2()
```

Python Code for problem5

```python
import matplotlib.pyplot as plt
import numpy as np
import sounddevice as sd

def plot_spectrum(Fs, title):
    plt.figure()
    f_values = np.array([-Fs/2, 0, Fs/2])
    X_values = np.array([0.5, 1, 0.5])

    plt.stem(f_values, X_values, basefmt=" ", use_line_collection=True)
    plt.xlabel('Frequency (Hz)')
    plt.ylabel('Magnitude')
    plt.title(title)
    plt.grid(True)

# For x1[n] with Fs1 = 4000 Hz
plot_spectrum(4000, 'Spectrum of x1[n]')

# For x2[n] with Fs2 = 1500 Hz
plot_spectrum(1500, 'Spectrum of x2[n] (Aliased)')

plt.tight_layout()
```

```python
plt.show()


# Time duration
T = 1.0  # seconds

# Sampling frequencies
Fs1 = 4000  # Hz for x1[n]
Fs2 = 1500  # Hz for x2[n]

# Generate time vectors
t1 = np.linspace(0, T, int(T * Fs1), endpoint=False)
t2 = np.linspace(0, T, int(T * Fs2), endpoint=False)

# Generate the signals
x1 = np.cos(2000 * np.pi * t1)
x2 = np.cos(2000 * np.pi * t2)

# Play the signals
print("Playing signal with Fs1 = 4000 Hz")
sd.play(x1, Fs1)
sd.wait()
print("Playing signal with Fs2 = 1500 Hz")
sd.play(x2, Fs2)
sd.wait()
```