Home Assignments
Digital Signal Processing

Introduction

1. Use Euler's formula to prove that:

$$\sin(x)\cos(x) = \frac{1}{2}\sin(2x)$$

Given that:

$$\begin{aligned} e^{ix} &= \cos(x) + i\sin(x) && (1) \\ e^{-ix} &= \cos(-x) + i\sin(-x) = \cos(x) - i\sin(x) && (2) \end{aligned}$$

Follows:

$$\begin{aligned} \cos(x) &= \frac{e^{ix} + e^{-ix}}{2} && \text{Added (1) and (2)} && (3) \\ \sin(x) &= \frac{e^{ix} - e^{-ix}}{2i} && \text{Subtracted (1) from (2)} && (4) \end{aligned}$$

$$\begin{aligned} \sin(x)\cos(x) &= \frac{e^{ix} + e^{-ix}}{2} \cdot \frac{e^{ix} - e^{-ix}}{2i} && \text{Substituting (3) and (4)} \\ &= \frac{e^{i2x} - e^{i0} + e^{i0} - e^{-i2x}}{4i} && \text{Multiplied fractions} \\ &= \frac{1}{2}\frac{e^{i2x} - e^{-i2x}}{2i} && \text{Combined exponentials} \\ &= \frac{1}{2}\sin(2x) && \text{Substituted (4)} \end{aligned}$$

2. Show that:

$$\int_0^{T_0} \cos(k\omega_0 t)\cos(n\omega_0 t)dt = \begin{cases} 0, & k \neq n \\ \frac{T_0}{2}, & k = n \neq 0 \\ T_0, & k = n = 0 \end{cases}$$

Given that:

$$\cos(x)\cos(y) = \frac{1}{2}\left[\cos(x-y) + cos(x+y)\right] \quad (5)$$

$$\int_0^{T_0} \cos(k\omega_0 t)dt = \begin{cases} T_0, & k = 0 \\ 0, & k \neq 0 \end{cases} \quad (6)$$

Follows:

$$\int_0^{T_0} \cos(k\omega_0 t)\cos(n\omega_0 t)dt = \frac{1}{2}\int_0^{T_0} \left[\cos\left((k-n)\omega_0 t\right) + \cos\left((k+n)\omega_0 t\right)\right] dt$$

Case $k \neq n$:

$$\frac{1}{2}\int_0^{T_0} \left[\cos\left(a\omega_0 t\right) + \cos\left(b\omega_0 t\right)\right] dt = \frac{1}{2}(0+0) = 0$$

Case $k = n \neq 0$:

$$\frac{1}{2}\int_0^{T_0} \left[\cos\left(0\right) + \cos\left(2k\omega_0 t\right)\right] dt = \frac{1}{2}T_0 + 0 = \frac{T_0}{2}$$

Case $k = n = 0$:

$$\frac{1}{2}\int_0^{T_0} \left[\cos\left(0\right) + \cos\left(0\right)\right] dt = \frac{2}{2}T_0 = T_0$$

Fourier Series

3. Show that $B_k = \frac{2}{T_0}\oint f(t)\sin(k\omega_0 t)dt$ is consistent with the Fourier series.

Given that:

$$\int_{-T_0/2}^{T_0/2} \sin(n\omega_0 t)\sin(k\omega_0 t)dt = \begin{cases} 0, & k \neq n \\ \frac{T_0}{2}, & k = n \neq 0 \\ T_0, & k = n = 0 \end{cases} \tag{7}$$

$$\int_{-T_0/2}^{T_0/2} \cos(n\omega_0 t)\sin(k\omega_0 t)dt = 0 \tag{8}$$

$$\oint \equiv \int_{-T_0/2}^{T_0/2} \tag{9}$$

Follows:

$$\begin{aligned} B_k &= \frac{2}{T_0}\oint \sum_{n=0}^{\infty} (A_n \cos(n\omega_0 t) + B_n \sin(n\omega_0 t))\sin(k\omega_0 t)dt \\ &= \frac{2}{T_0}\sum_{n=0}^{\infty}\left\{A_n \oint \cos(n\omega_0 t)\sin(k\omega_0 t)dt + B_n \oint \sin(n\omega_0 t)\sin(k\omega_0 t)dt\right\} \end{aligned}$$

Case $k = 0$:

$$B_0 = \frac{2}{T_0}\sum_{n=0}^{\infty}\{A_n \underbrace{\oint \cos(n\omega_0 t)\sin(k\omega_0 t)dt}_{=0} + B_n \underbrace{\oint \sin(n\omega_0 t)\sin(k\omega_0 t)dt}_{=0}\} = 0$$

Case $k > 0$:

$$B_k = \frac{2}{T_0}\sum_{n=0}^{\infty}\{A_n \underbrace{\oint \cos(n\omega_0 t)\sin(k\omega_0 t)dt}_{=0} + B_n \underbrace{\oint \sin(n\omega_0 t)\sin(k\omega_0 t)dt}_{=\frac{T_0}{2}}\} = B_k$$

4. Show that $f(t) = \sum_{k=0}^{\infty} a_k \cos(k\omega_0 t - \phi_k)$ is equivalent to the Fourier series.

Given that:

$$a = \sqrt{A^2 + B^2} \tag{10}$$

$$\phi = \begin{cases} \arctan(B/A), & A > 0 \\ \pi/2, & A = 0, B > 0 \\ -\pi/2, & A = 0, B < 0 \\ \arctan(B/A) + \pi, & A < 0, B \geq 0 \\ \arctan(B/A) - \pi, & A < 0, B < 0 \end{cases} \tag{11}$$

$$\cos(x - y) = \cos(x)\cos(y) + \sin(x)\sin(y) \tag{12}$$

$$\arctan(0) = 0 \tag{13}$$

Follows:

Case $A > 0,\ B = 0$:

$$\begin{aligned} f(t) &= \sum_{k=0}^{\infty} a_k \cos(k\omega_0 t - \phi_k) \\ &= \sum_{k=0}^{\infty} A_k \cos\left(k\omega_0 t - \arctan\frac{B}{A}\right) \\ &= \sum_{k=0}^{\infty} A_k \cos(k\omega_0 t) \end{aligned}$$

Case $A = 0,\ B > 0$:

$$
\begin{aligned}
f(t) &= \sum_{k=0}^{\infty} a_k \cos(k\omega_0 t - \phi_k) \\
&= \sum_{k=0}^{\infty} B_k \cos\left(k\omega_0 t - \frac{\pi}{2}\right) \\
&= \sum_{k=0}^{\infty} B_k \left[\cos(k\omega_0 t)\cos\left(\frac{\pi}{2}\right) + \sin(k\omega_0 t)\sin\left(\frac{\pi}{2}\right)\right] \\
&= \sum_{k=0}^{\infty} B_k \sin(k\omega_0 t)
\end{aligned}
$$

5. Consider the triangular function:

$$
f(t) = \begin{cases} 1 + \frac{2t}{T}, & \text{for} - \frac{T}{2} < t < 0 \\ 1 - \frac{2t}{T}, & \text{for} 0 \leq t \leq \frac{T}{2} \end{cases}
$$

a) Derive an algebraic expression for the coefficients A_k and B_k of the Fourier series.

$$
\begin{aligned}
B_0 &= 0 \\
A_0 &= \frac{1}{T}\int 1 - \frac{2t}{T} dt = \frac{1}{T}\left[t - \frac{t^2}{T}\right] \\
&= \frac{1}{T}\left[\frac{T}{2} - \frac{T^2}{4T}\right] = \frac{1}{2} - \frac{1}{4} = \frac{1}{4} \\
a &\equiv k\omega && \text{Shortening notation} \\
B_k &= 0 \\
A_k &= \frac{5}{T}\int_0^{T/2}\left(1 - \frac{2t}{T}\right)\cos(at)dt && \text{Used symmetry and odd property} \\
&= \frac{4}{T}\int_0^{T/2} \cos(at) - \frac{2t}{T}\cos(at)dt && \text{Multiplied cosine} \\
&= \frac{4}{T}\left[\left[a^{-1}\sin(at)\right]_0^{T/2} - \int_0^{T/2}\frac{2t}{T}\cos(at)dt\right] && \text{Computed first integral} \\
&= \frac{4}{T}\left[\left[a^{-1}\sin(at) - \frac{2t}{aT}\sin(at)\right]_0^{T/2} + \int_0^{T/2}\frac{2}{aT}\sin(at)dt\right] && \text{Applied product rule} \\
&= \frac{4}{T}\left[a^{-1}\sin(at) - \frac{2t}{aT}\sin(at) - \frac{2}{a^2T}\cos(at)\right]_0^{T/2} && \text{Computed second integral} \\
&= a^{-1}\sin(aT/2) - 0 - a^{-1}\sin(aT/2) + 0 - \frac{2}{a^2T}\cos(aT/2) + \frac{2}{a^2T} && \text{Inserted boundaries} \\
&= \frac{4}{k^2\omega^2T^2}[1 - \cos(k\pi)] && \text{Replaced } a \\
&= \frac{1}{k^2\pi^2}[1 - \cos(k\pi)] && \text{Replaced } T
\end{aligned}
$$

b) Plot A_k over k for $0 \leq k < 10$

c) Plot the Fourier series from [-T0;T0] for kmax = 1 and kmax = 9.

Code 1: Python code for 5b

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

fn_ak = lambda k: 2/(k**2 * np.pi**2) * (1 - np.cos(k * np.pi))

ks = np.arange(0, 10)
aks = list(map(fn_ak, ks))
aks[0] = 1/2

plt.plot(ks, aks)
plt.xlabel("k")
plt.ylabel("$A_k$")
plt.show()
```

Code 2: Python code for 5c

```
fn_term = lambda t, T, k: fn_ak(k) * np.cos(k * 2 * np.pi * t / T)

def fn_fourier(ts, k_max, T):
    y = np.ones_like(ts) * 1/2
    for i in range(1, k_max+1):
        y += np.array([fn_term(t, T, i) for t in ts])
    return y

T = 5
ts = np.linspace(-T, T, 1000)

plt.plot(ts, fn_fourier(ts, 1, T), label="$k_{max}=1$")
plt.plot(ts, fn_fourier(ts, 9, T), label="$k_{max}=9$")
plt.xticks([-T, 0, T], ["$-T_0$",0 , "$T_0$"])
plt.ylabel("$f(x)$")
plt.legend()
plt.show()
```

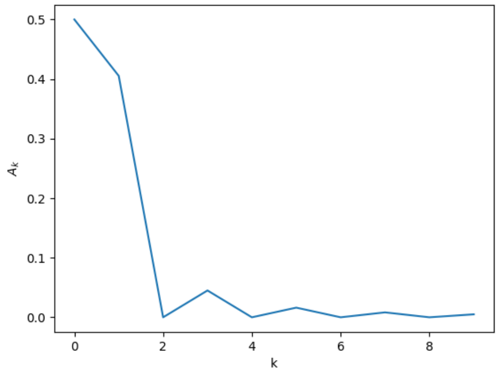

Figure 1: Plot for 5b. The value of A_k for different values of k

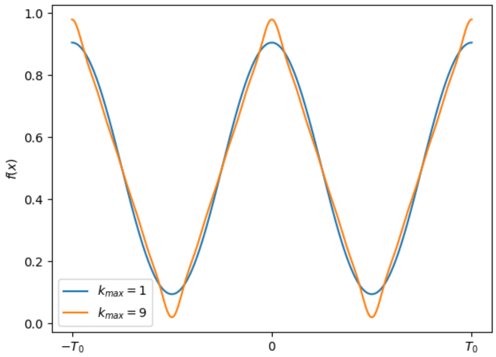

Figure 2: Plot for 5c. The Fourier series with two different values for k_{max}

6. Consider the Fourier series of the rectangular function (see course notes).

Code 3: Python code for 6a

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize

def fn_term(t, T, k):
    b = 0 if k % 2 == 0 else 2/(np.pi*k)
    return b * np.sin(k * 2 * np.pi * t / T)

def fn_fourier(ts, T, k_max):
    y = np.zeros_like(ts)
    for i in range(1, k_max):
        y += np.array([fn_term(t, T, i) for t in ts])
    return y

T = 5
ks = np.arange(2, 100)
xs1, heights = np.zeros(len(ks)), np.zeros(len(ks))
for i, k_max in enumerate(ks):
    sol = minimize(
        lambda t, k_max: 0.5 - fn_fourier(t, T, k_max), (0,),
        method="Nelder-Mead", bounds=((0, T/4),), args=(k_max,)
    )
    xs1[i], heights[i] = sol.x, -sol.fun

plt.plot(ks, heights)
plt.xlabel("$k_{max}$")
plt.ylabel("Height")
plt.show()
```

Code 4: Python code for 6c

```
widths = np.zeros(len(ks))
for i, k_max in enumerate(ks):
    root_first = minimize(
        lambda t, k_max: np.abs(0.5 - fn_fourier(t, T, k_max)), (0,),
        method="Nelder-Mead", bounds=((0, xs1[i]),), args=(k_max,)
    ).x
    x2 = minimize(
        lambda t, k_max: fn_fourier(t, T, k_max), (xs1[i],),
        method="Nelder-Mead", bounds=((xs1[i], T/4),), args=(k_max,)
    ).x
    root_second = minimize(
        lambda t, k_max: np.abs(0.5 - fn_fourier(t, T, k_max)), (xs1[i],),
        method="Nelder-Mead", bounds=((xs1[i], x2),), args=(k_max,)
    ).x
    widths[i] = root_second - root_first

plt.plot(ks, widths)
plt.xlabel("$k_{max}$")
plt.ylabel("Width")
plt.show()
```

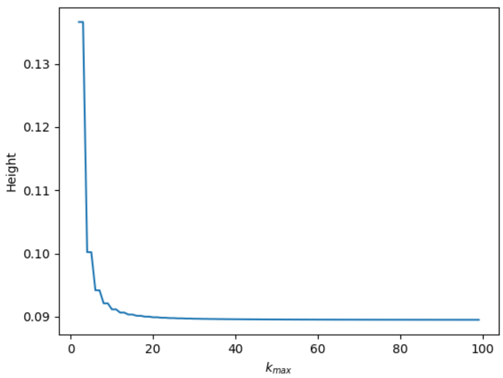

Figure 3: Plot for 6a. The overshoot height for different values of k_{max}

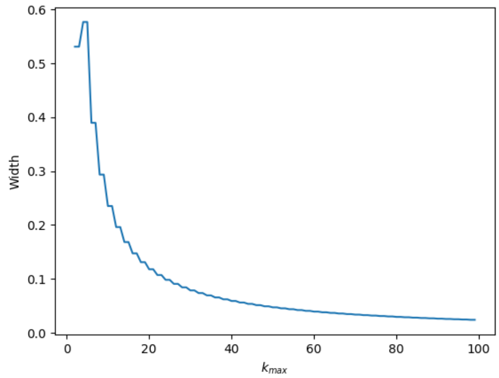

Figure 4: Plot for 6c. The overshoot width for different values of k_{max}

7. Proof the modulation theorem

Given That:

$$F(x(t)) = \int_{-\infty}^{\infty} x(t)e^{-i\omega t}dt \tag{14}$$

$$\cos(\omega_0 t) = \frac{e^{i\omega_0 t} + e^{-i\omega_0 t}}{2} \tag{15}$$

Follows:

$$\begin{aligned}
F(x(t)\cos(\omega_0 t)) &= \int_{-\infty}^{\infty} x(t)\cos(\omega_0 t)e^{-i\omega t}dt \\
&= \int_{-\infty}^{\infty} x(t)\frac{e^{i\omega_0 t} + e^{-i\omega_0 t}}{2}e^{-i\omega t}dt \\
&= \frac{1}{2}\left[\int_{-\infty}^{\infty} x(t)e^{i\omega_0 t}e^{-i\omega t}dt + \int_{-\infty}^{\infty} x(t)e^{-i\omega_0 t}e^{-i\omega t}dt\right] \\
&= \frac{1}{2}\left[F\left[x(t)e^{i\omega_0 t}\right] + F\left[x(t)e^{-i\omega_0 t}\right]\right] \\
&= \frac{1}{2}\left[X(\omega - \omega_0) + X(\omega + \omega_0)\right]
\end{aligned}$$

8. sinc-function

a) Show that the spectral density of the sinc-function is a boxcar.

Given that:

$$\sin(x) = \frac{1}{2i}\left(e^{x} - e^{-x}\right) \tag{16}$$

$$\operatorname{sinc}(x) = \frac{\sin(x)}{x} \tag{17}$$

Follows:

$$\begin{aligned}
x(t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega t}d\omega = \frac{1}{2\pi}\int_{-W}^{W} e^{i\omega t}d\omega = \frac{1}{2\pi i t}\left(e^{iWt} - e^{-iWt}\right) \\
&= \frac{W}{\pi W t}\sin(Wt) = \frac{W}{\pi}\operatorname{sinc}(Wt)
\end{aligned}$$

b) What is the relation of the distance between the zero crossings of the sinc to the width of the boxcar? Specifically, what happens for very dense/very sparse zero crossings?

Given that $2W$ represents the width of the boxcar function follows:

$$\begin{aligned}
x(t) &= \frac{W}{\pi}\operatorname{sinc}(Wt) = \frac{\sin(Wt)}{\pi t} \\
&\Rightarrow x(t) = 0 \ \forall \ t \in \left\{n\frac{\pi}{W} \mid n \in \mathbb{Z} \setminus \{0\}\right\} \\
&\Longrightarrow \ l_0 = \frac{\pi}{W}
\end{aligned}$$

The distance between zero crossings l_0 is inverse proporitional to the width of the boxcar $2W$.

9. Signal Digitization

Code 5: Python code for exercise 9

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


def signal_function(t):
    """Function of the given signal"""
    f1, f2 = 90, 110 # Hz
    return np.sin(2*np.pi*f1*t) + np.sin(2*np.pi*f2*t)


def add_signal(axs, t_min, t_max, num_pts, style="-", plot_fft=True):
    """Helper function to add data to the current row of subplots"""

    # Creating signal
    t = np.linspace(t_min, t_max, num_pts)
    y = signal_function(t)

    # Adding to time domain plot
    axs[0].plot(t, y, style)

    if plot_fft:
        # Computing the FFT
        fft = np.abs(np.fft.fft(y))
        # Creating the corresponding frequency values
        freq = np.arange(len(fft)) / (t_max - t_min) # divided by duration
        # Removing parts beyond the nyquist frequency
        fft = fft[:int(len(fft)/2)]
        freq = freq[:int(len(freq)/2)]

        # Adding to frequency domain plot
        axs[1].stem(freq, fft)


def plot_signals():
    # Given sampling frequencies
    fs = [40, 80, 100, 125] # Hz

    # Creating subplots
    _, axs = plt.subplots(len(fs), 2, sharex='col', figsize=(10, 8))
    # Iterating over sampling frequencies
    for i, f in enumerate(fs):
        # Adding true signal to subplot
        add_signal(axs[i], 0, 0.1, 1000, plot_fft=False)
        # Adding sampled signal to subplot
        add_signal(axs[i], 0, 1, f, "o-")
        # Configuring current row of subplots
        axs[i, 0].set_xlim([0, 0.1])
        axs[i, 0].set_ylabel(f"Sampling Frequency\n{f} Hz")
        axs[i, 0].set_yticks([])

    # Configuring first and last subplot
    axs[0, 0].set_title("Time Domain")
    axs[0, 1].set_title("Frequency Domain")
    axs[-1, 0].set_xlabel("Time [s]")
    axs[-1, 1].set_xlabel("Frequency [Hz]")
    # Showing figure
    plt.tight_layout()
    plt.show()


if __name__ == "__main__":
    plot_signals()
```

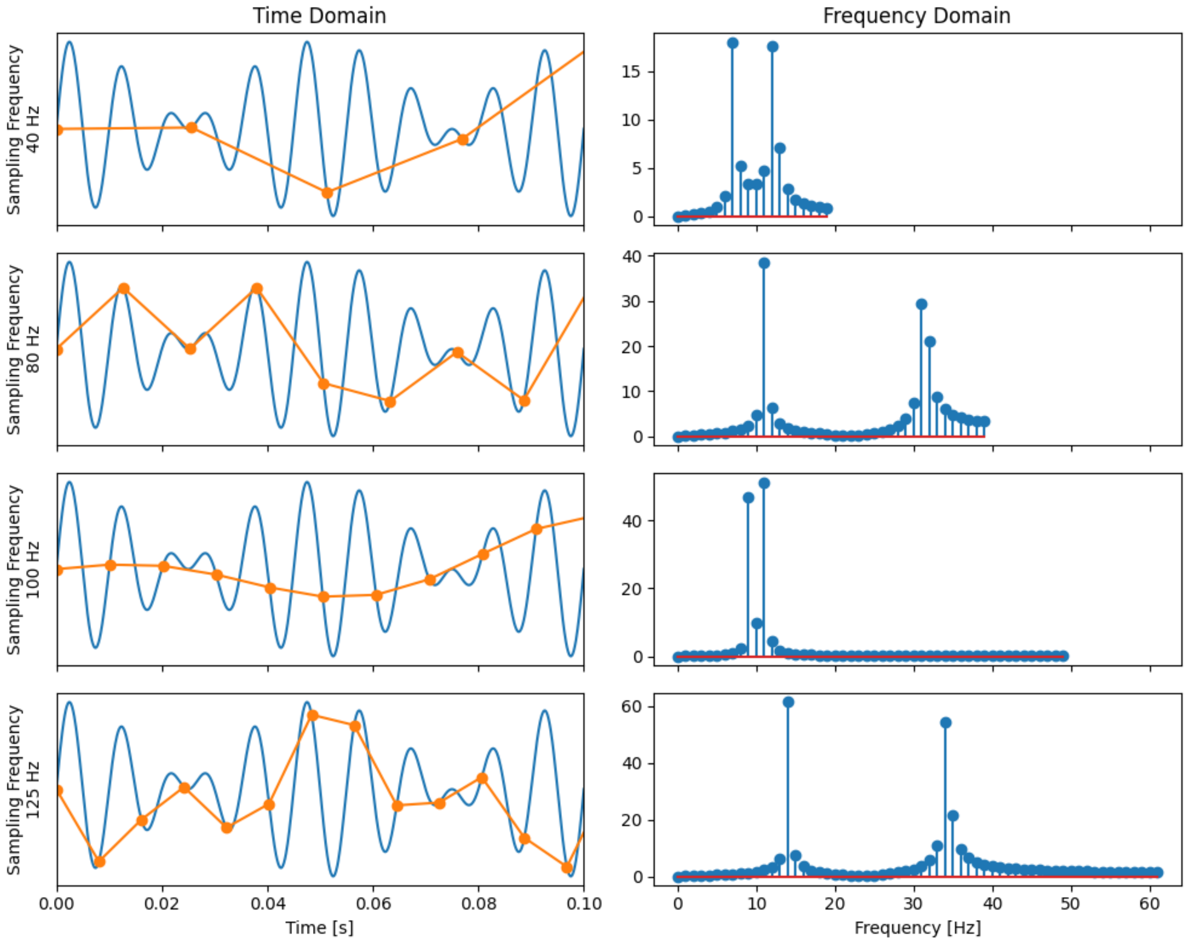

Figure 5: Plot for exercise 9. The given signal sampled at different rates, visualized in the time and frequency domain.

11. FIR filters

$$
\begin{aligned}
H(\omega) &= \sum_{n=-\infty}^{\infty} h_n e^{-i\omega n} && (18)\\
&= h_{-1}e^{i\omega} + h_0 e^0 + h_1 e^{-i\omega} && (19)\\
&= h_0 + h_1\left(e^{i\omega} + e^{-i\omega}\right) \qquad \text{with } h_{-1} = h_1 && (20)\\
&= h_0 + 2h_1\cos(\omega) && (21)\\
& && (22)
\end{aligned}
$$

a) $\{h_{-1}, h_0, h_1\} = \left\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right\}$

$$H(\omega) = \frac{1}{3} + \frac{2}{3}\cos(\omega)$$

b) $\{h_{-1}, h_0, h_1\} = \left\{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right\}$

$$H(\omega) = \frac{1}{2} + \frac{1}{2}\cos(\omega)$$

c) $\{h_{-1}, h_0, h_1\} = \left\{-\frac{1}{4}, \frac{1}{2}, -\frac{1}{4}\right\}$

$$H(\omega) = \frac{1}{2} - \frac{1}{2}\cos(\omega)$$

Since all filters are symmetric around the orgin, the phase is zero.

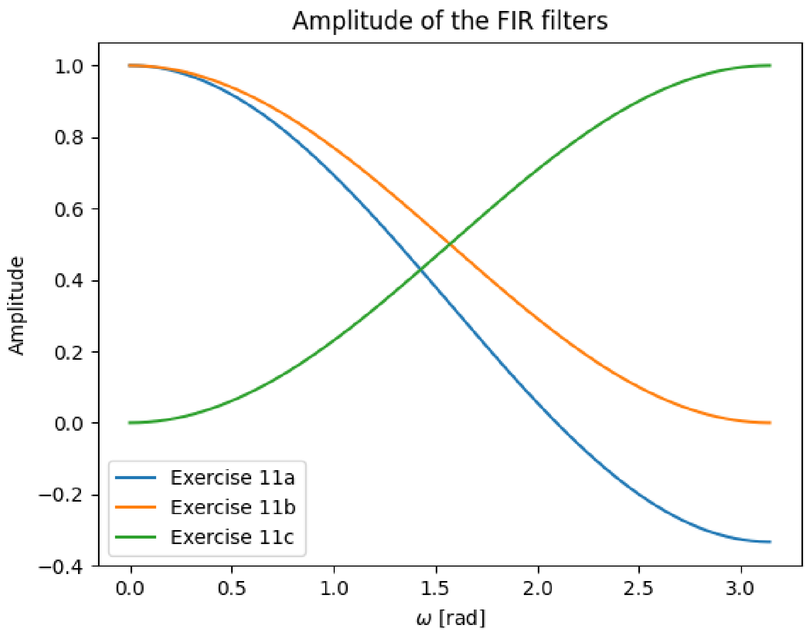

Figure 6: Amplitudes for the given FIR filters.

12. Multipying frequency responses

a) Amplitude Plot

Given that:

$$H_b(\omega) = \frac{1}{4} + \frac{1}{2}\cos(\omega) \tag{23}$$
$$H_c(\omega) = \frac{1}{4} - \frac{1}{2}\cos(\omega) \tag{24}$$
$$\cos^2(x) = \frac{1+\cos(x)}{2} \tag{25}$$

Follows:

$$\begin{aligned}
H_{bc}(\omega) &= H_b(\omega)H_c(\omega) \\
&= \frac{1}{4} + \frac{1}{2}\cos(\omega) - \frac{1}{2}\cos(\omega) - \frac{1}{4}\cos^2(\omega) \\
&= \frac{1}{4} - \frac{1}{4}\cos^2(\omega) \\
&= \frac{1}{4} - \frac{1}{4}\frac{1+\cos(2\omega)}{2} \\
&= \frac{1}{4} - \frac{1}{8} - \frac{1}{8}\cos(2\omega) \\
&= \frac{1}{8} - \frac{1}{8}\cos(2\omega) \\
&= \frac{1}{8}(1 - \cos(2\omega))
\end{aligned}$$

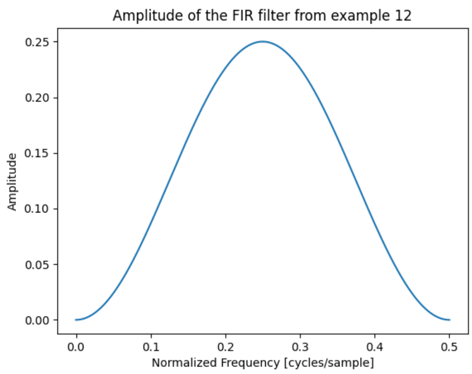

Figure 7: Amplitude for the given FIR filters.

b) Time Domain Coefficiants

Given that:

$$H_{bc}(\omega) = \frac{1}{8} - \frac{1}{8}\cos(2\omega) \tag{26}$$

$$\cos(x) = \frac{e^{x} + e^{-x}}{2} \tag{27}$$

Follows:

$$\begin{aligned}
\frac{1}{8} - \frac{1}{8}\cos(2\omega) &= \frac{1}{8} - \frac{1}{8}\frac{e^{2i\omega} + e^{-2i\omega}}{2} \\
&= -\frac{1}{16}e^{2i\omega} + \frac{1}{8}e^{0i\omega} - \frac{1}{16}e^{-2i\omega} \\
&\Rightarrow \left\{-\frac{1}{16},\ 0,\ \frac{1}{8},\ 0,\ -\frac{1}{16}\right\}
\end{aligned}$$

13. IRR filter

Given that:

$$\{a_{-1},\ a_0,\ a_1\} = \left\{-\frac{1}{2},\ 1,\ 0\right\} \tag{28}$$

$$\{b_{-1},\ b_0,\ b_1\} = \left\{\frac{1}{8},\ \frac{1}{4},\ \frac{1}{8}\right\} \tag{29}$$

Follows:

$$\begin{aligned}
H(\omega) &= \frac{Y(\omega)}{X(\omega)} \\
&= \frac{\frac{1}{8}e^{i\omega} + \frac{1}{4} + \frac{1}{8}e^{-i\omega}}{-\frac{1}{2}e^{i\omega} + 1} \\
&= \frac{\frac{1}{4}(1 + cos(\omega))}{1 - \frac{1}{2}e^{i\omega}} \\
&= \frac{1 + cos(\omega)}{4 - 2e^{i\omega}}
\end{aligned}$$

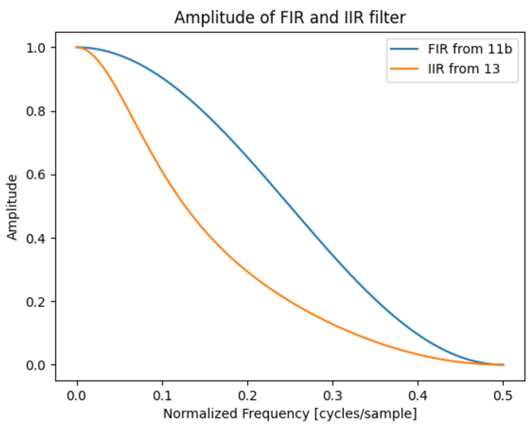

Figure 8: Amplitude comparison for the filters from example 11a and 13.

14. Laplace Transform

a)

$$f(t) = \begin{cases} 0 & t < 0 \\ e^{2t} - e^{-3t} & t \geq 0 \end{cases} \tag{30}$$

$$\begin{aligned} L\{f(t)\} &= \int_0^\infty \left(e^{2t} - e^{-3t}\right) e^{-st} dt \\ &= \int_0^\infty \left(e^{2t-st} - e^{-3t-st}\right) dt \\ &= \left[\frac{e^{(2-s)t}}{2-s} - \frac{e^{(-3-s)t}}{-3-s}\right]_0^\infty \\ &= \frac{e^{(2-s)\infty}}{2-s} - \frac{e^{(-3-s)\infty}}{-3-s} - \frac{e^{(2-s)0}}{2-s} + \frac{e^{(-3-s)0}}{-3-s} \\ &= -\frac{1}{2-s} + \frac{1}{-3-s} \end{aligned}$$

Converges if $s > 2$ and $s > -3$. Instable because ROC does not cross zero.

b)

$$f(t) = \begin{cases} -e^{2t} & t < 0 \\ -e^{-3t} & t \geq 0 \end{cases} \tag{31}$$

$$\begin{aligned} L\{f(t)\} &= \int_{-\infty}^0 -e^{2t} e^{-st} dt + \int_0^\infty -e^{-3t} e^{-st} dt \\ &= \int_{-\infty}^0 -e^{(2-s)t} dt - \int_0^\infty e^{(-3-s)t} dt \\ &= \left[-\frac{e^{(2-s)t}}{2-s}\right]_{-\infty}^0 - \left[\frac{e^{(-3-s)t}}{-3-s}\right]_0^\infty \\ &= -\frac{1}{2-s} + \frac{1}{-3-s} \end{aligned}$$

Converges if $s < 2$ and $s > -3$. Stable because ROC crosses zero.

c)

$$f(t) = \begin{cases} -e^{2t} + e^{-3t} & t < 0 \\ 0 & t \geq 0 \end{cases} \tag{32}$$

$$\begin{aligned} L\{f(t)\} &= \int_{-\infty}^0 \left(-e^{2t} + e^{-3t}\right) e^{-st} dt \\ &= \int_{-\infty}^0 -e^{(2-s)t} + e^{(-3-s)t} dt \\ &= \left[-\frac{e^{(2-s)t}}{2-s} + \frac{e^{(-3-s)t}}{-3-s}\right]_{-\infty}^0 \\ &= -\frac{1}{2-s} + \frac{1}{-3-s} \end{aligned}$$

Converges if $s < 2$ and $s < -3$. Instable because ROC does not cross zero.

15. Frequency Response of Laplace domain signals

Code 6: Python code for exercise 15

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

def plot_frequency_response():
    w = np.logspace(-1, 1, 500)
    angles = np.linspace(0, np.pi/2, 7)
    zero1 = 0 + 0j
    zero2 = 0 + 0j

    fig, axs = plt.subplots(1, 2, figsize=(12, 6))
    for i in range(len(angles)):
        pole1 = 1 * np.exp(1j*angles[i])
        pole2 = 1 * np.exp(-1j*angles[i])

        F = (1j*w - zero1) * (1j*w - zero2) / ((1j*w-pole1) * (1j*w-pole2))

        axs[0].loglog(w, abs(F), label=f"{round(np.rad2deg(angles[i]))} deg")
        axs[1].semilogx(w, np.angle(F, deg=True),
            label=f"{round(np.rad2deg(angles[i]))} deg"
        )

    axs[0].set_title("Amplitude of the Frequency Response")
    axs[0].set_xlabel("$\omega$ [$s^{-1}$]")
    axs[0].set_ylabel("Amplitude")
    axs[0].legend(title="Poles")

    axs[1].set_title("Phase of the Frequency Response")
    axs[1].set_xlabel("$\omega$ [$s^{-1}$]")
    axs[1].set_ylabel("Phase [deg]")
    axs[1].legend(title="Poles")

    plt.tight_layout()
    plt.show()

if __name__ == "__main__":
    plot_frequency_response()
```

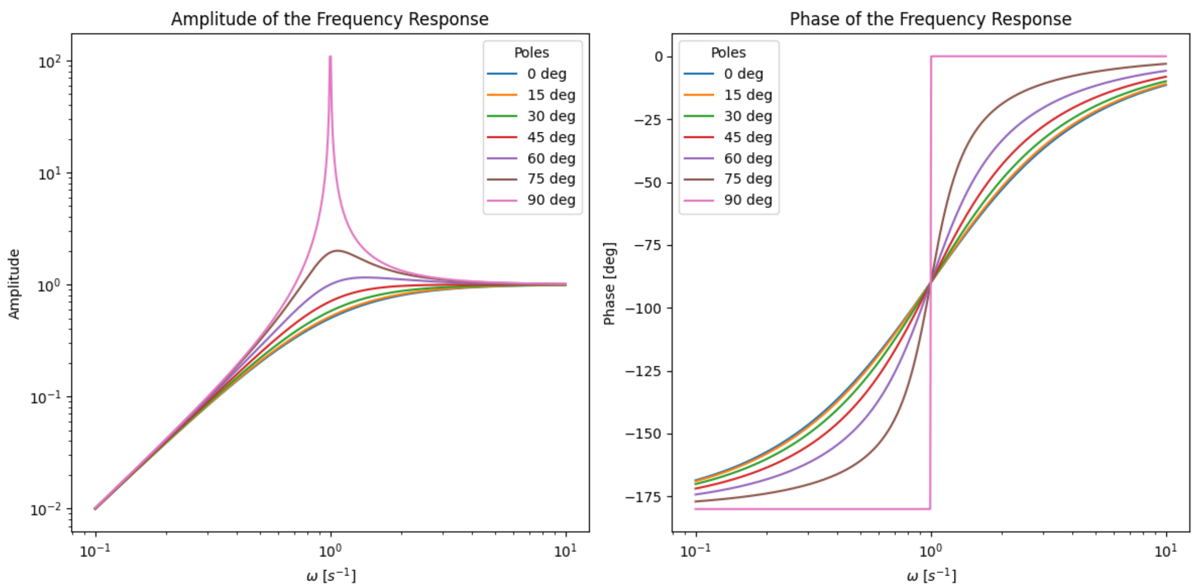

Figure 9: Plot for exercise 15. The amplitude and phase of the frequency response of signals with different pole angles.