COM-303 - Signal Processing for Communications

Solutions for Homework #5

Solution 1. LTI Systems

The system is not time-invariant. To see this consider the following signals:

$$\begin{aligned} x[n] &= \delta[n] \\ y[n] &= \delta[n-1] \end{aligned}$$

Now set

$$\begin{aligned} \mathscr{H}\{x[n]\} &= w[n] \\ \mathscr{H}\{y[n]\} &= r[n] \end{aligned}$$

It is easy to see that $w[n] = 0$ for all n, while $r[n] = \delta[n-1]$. However, since $y[n] = x[n-1]$ if the system was LTI we should have $r[n] = w[n-1]$, which is not true.

Solution 2. Convolution

(a) The discrete-time sequence $x[n]$ can be written as the self-convolution of a sequence $t[n]$ defined as

$$t[n] = \begin{cases} 1 & -(M-1)/2 \le n \le (M-1)/2 \\ 0 & \text{otherwise.} \end{cases}$$

We can verify this from the symmetry of $t[n]$ and by noticing that the sum corresponds to the size of the overlapping area between $t[k]$ and its copy shifted by n. When $|n| \ge M$ the two sequences do not overlap whereas the size of the overlapping area reaches its maximum M when $n = 0$.

Using Python and Numpy, we can easily verify the above result for $M = 11$ using the following code:

```
t = np.ones(11)
x = np.convolve(t, t)
plt.stem(x)
```

(b) The DTFT of the signal $t[n]$ is known and has been derived in class. Using the convolution theorem, we can write

$$\begin{aligned} X(e^{jw}) &= T(e^{j\omega})T(e^{j\omega}) \\ &= \left(\frac{\sin(\omega M/2)}{\sin(\omega/2)}\right)^2. \end{aligned}$$

Solution 3. System Properties

(a) linearity: YES
$\mathcal{H}\{a\,x_1[n]+b\,x_2[n]\} = a\,x_1[-n]+b\,x_2[-n] = a\mathcal{H}\{x_1[n]\}+b\mathcal{H}\{x_2[n]\}$.
time invariance: NO
$\mathcal{H}\{x[n-n_0]\} = x[-n-n_0] \neq y[n-n_0]$.
stability: YES
if $|x[n]| \leq M$, then $|\mathcal{H}\{x[n]\}| \leq M$.
causality: NO
impulse response: NOT APPLICABLE (system is not LTI)

(b) linearity: YES
$\mathcal{H}\{a\,x_1[n]+b\,x_2[n]\} = e^{-j\omega n}(a\,x_1[n]+b\,x_2[n]) = a\mathcal{H}\{x_1[n]\}+b\mathcal{H}\{x_2[n]\}$
time invariance: NO
$\mathcal{H}\{x[n-n_0]\} = e^{-j\omega n}x[n-n_0] = e^{j\omega n_0}y[n-n_0]$.
stability: YES
If $|x[n]| \leq M$, then $|\mathcal{H}\{x[n]\}| = |x[n]| \leq M$.
causality: YES
impulse response: NOT APPLICABLE (system is not LTI)

(c) linearity: YES
$\mathcal{H}\{a\,x_1[n]+b\,x_2[n]\} = \sum_{k=n-n_0}^{n+n_0}(a\,x_1[k]+b\,x_2[k]) = a\mathcal{H}\{x_1[n]\}+b\mathcal{H}\{x_2[n]\}$
time invariance: YES
$\mathcal{H}\{x[n-n_0]\} = \sum_{k=n-n_0}^{n+n_0} x[k-n_0] = \sum_{k=n-2n_0}^{n} x[k] = y[n-n_0]$
stability: YES
If $|x[n]| \leq M$, then $\mathcal{H}\{x[n]\} \leq |2n_0+1|M$
causality: NO
impulse response:

$$h[n] = \begin{cases} 1 & \text{if } |n| \leq |n_0|, \\ 0 & \text{otherwise.} \end{cases}$$

Solution 4. Ideal Filters

Consider a lowpass filter $h_{lp}[n]$ with bandwidth ω_b. If we consider the sequence

$$h[n] = 2\cos(\omega_0 n)h_{lp}[n]$$

the Modulation theorem tells us that its Fourier transform is

$$H(e^{j\omega}) = H_{lp}(e^{j(\omega-\omega_0)}) + H_{lp}(e^{j(\omega+\omega_0)}) = H_{bp}(e^{j\omega})$$

Therefore the impulse response of the bandpass filter is

$$h_{bp}[n] = 2\cos(\omega_0 n) h_{lp}[n] = 2\cos(\omega_0 n)\,\frac{\omega_b}{2\pi}\,\mathrm{sinc}\left(\frac{\omega_b}{2\pi}n\right)$$

Solution 5. Signals and Systems

(a) $\mathscr{H}\{\delta[n]\} = \delta[n]$; but $\mathscr{H}\{a\delta[n]\} = a^2\delta[n] \neq a\mathscr{H}\{\delta[n]\}$.

(b) Let $v[n] = \mathscr{H}\{x[n]\}$; let $w[n] = x[n-n_0]$; $\mathscr{H}\{w[n]\} = w^2[n] = x^2[n-n_0] = v[n-n_0]$.

(c) First of all, $\cos^2(\omega_0 n) = (1+\cos(2\omega_0 n))/2$ from the well-known trigonometric identity. So the output of the first block in the cascade contains a sinusoid at *double* the original frequency (but be careful: double in the 2π-periodic sense: if ω_0 is larger than $\pi/2$, then $2\omega_0$ will wrap around the $[-\pi, \pi]$ interval).

If $\omega_0 = 3\pi/8$, then $\mathscr{H}\{x[n]\} = (1+\cos((3\pi/4)n))/2$; since $\mathscr{G}$ is a highpass with cutoff frequency $\pi/2$, it will kill the frequency components below $\pi/2$ and therefore it will kill the constant term in the input. The only component that passes through is the cosine at $3\pi/4$. The final output is therefore $y[n] = \cos((3\pi/4)n)$.

(d) If $\omega_0 = 7\pi/8$, then $2\omega_0 = 7\pi/4 > \pi$. We can therefore bring back the frequency into the $[-\pi, \pi]$ interval. We have that $7\pi/4 = 2\pi - \pi/4$ and therefore $\cos((7\pi/4)n) = \cos((2\pi - \pi/4)n) = \cos((\pi/4)n)$. So in the end $\mathscr{H}\{x[n]\} = (1+\cos((\pi/4)n))/2$. Note that this time the frequency of the cosine is below $\pi/2$ and therefore $y[n] = 0$.

Solution 6. Linear Phase FIR Filters

If M is odd, then $C = (M-1)/2$ is an integer. We can then write

$$\begin{aligned}
\sum_{n=0}^{M-1} h[n]e^{-jn\omega} &= h[0] + h[1]e^{-j\omega} + \ldots + h[M-2]e^{-j(M-2)\omega} + h[M-1]e^{-j(M-1)\omega} \\
&= e^{-jC\omega}\big[(h[0]e^{jC\omega} + h[1]e^{j(C-1)\omega} + \ldots + h[C] + \ldots \\
&\quad + h[2C-1]e^{-j(C-1)\omega} + h[2C]e^{-jC\omega}\big] \\
&= e^{-jC\omega}[2h[0]\cos(C\omega) + 2h[1]\cos((C-1)\omega) + \ldots].
\end{aligned}$$

Since the term in brackets is real-valued, the phase is determined by the complex exponential only and it is linear.

Solution 7. The Gibbs Phenomenon

Code sample that plots the required figures:

```
from __future__ import division
import numpy as np
import matplotlib.pylab as plt


def ex3_func(omega, N):
    n = np.arange(-N, N + 1, step=1, dtype=float)[np.newaxis, :]
    h = 0.5 * np.sinc(n / 2.)
    # DTFT for the interval
    e_jw = np.exp(-1j * np.dot(omega, n))
    H = np.dot(e_jw, h.T)
    return H


if __name__ == '__main__':
    # define the range for omega: [1.4, 1.7]
    omega = (1.4 + 0.3 * np.linspace(0, 1, num=2000, dtype=float))[:, np.newaxis]
    count = 0
    for N in [20, 100, 200, 1000]:
        H = ex3_func(omega, N)
        plt.figure(num=count, dpi=90, figsize=(5, 3))
        plt.plot(omega, np.abs(H))
        plt.plot(omega, 1.09 * np.ones(omega.shape), 'r--', hold=True)
        plt.xlim([1.4, 1.7])
        plt.xlabel(r'$\omega$', fontsize=12)
        plt.ylabel(r'$\left|\hat{H}(e^{j\omega})\right|$', fontsize=12)
        plt.grid()
        plt.show()
        count += 1
```