ESE 531: Homework 1

Noah Schwab

January 29, 2022

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

2.23

a) $T(x[n]) = (\cos \pi n)x[n]$

1) Stable
2) Causal
3) Linear
4) Not time-invariant

b) $T(x[n]) = x[n^2]$

1) Stable
2) Not causal
3) Linear
4) Not time-invariant

c) $T(x[n]) = x[n] \sum_{k=0}^{\infty} \delta[n-k]$

1) Stable
2) Causal
3) Linear
4) Not time-invariant

d) $T(x[n]) = \sum_{k=n-1}^{\infty} x[k]$

1) Unstable
2) Not causal
3) Linear
4) Time-invariant

2.36

a) $h[n] = (0.8)^n u[n] + (0.8)^{n-2} u[n-2]$

b) $y[n] = 0.8y[n-1] + x[n] + x[n-2]$

c) $\omega_0 = \frac{\pi}{2}$; $A = 40$

2.54

a) $h[n] = \alpha^n u[n] + \beta \alpha^{n-1} u[n-1]$

b) $H(e^{j\omega}) = \frac{1+\beta e^{-j\omega}}{1-\alpha e^{-j\omega}}$, $|\alpha| < 1$

c) $y[n] - \alpha y[n-1] = x[n] + \beta x[n-1]$

d) $|\alpha| < 1$ and $|\beta| < \infty$

2.64

a) $H_1(e^{j\omega}) = \begin{cases} 1 & : 0.8\pi \leq |\omega| \leq \pi \\ 0 & : |\omega| < 0.8\pi \end{cases}$

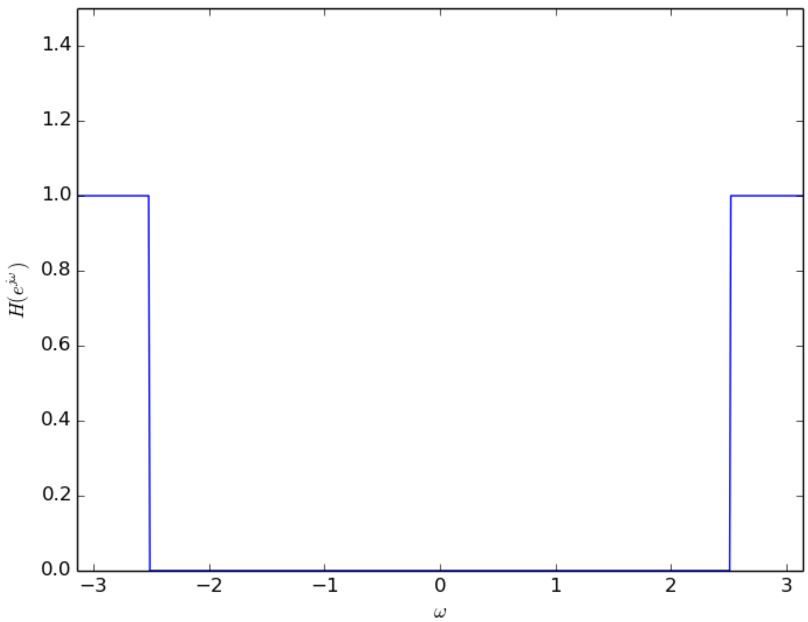


Figure 1: High-pass Filter

b) $H_2(e^{j\omega}) = \begin{cases} 1 & : 0.3\pi \leq |\omega| \leq 0.7\pi \\ 0 & : |\omega| < 0.3\pi,\ 0.7\pi < |\omega| \leq \pi \end{cases}$

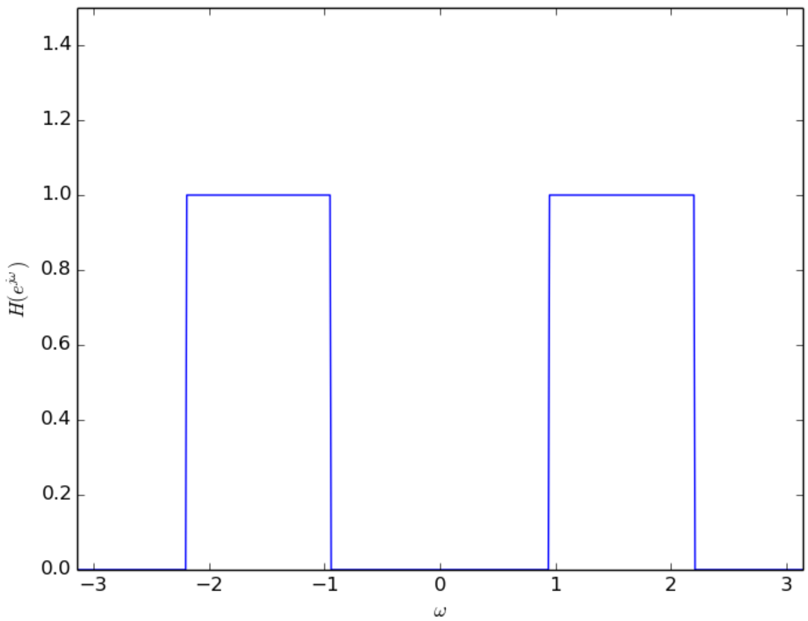


Figure 2: Band-pass Filter

c) $$H_3(e^{j\omega}) = \begin{cases} 0.1 & : |\omega| \leq 0.1\pi \\ \frac{\omega}{2\pi} + 0.15 & : -0.3\pi \leq \omega < -0.1\pi \\ \frac{-\omega}{2\pi} + 0.15 & : 0.1\pi < \omega \leq 0.3\pi \\ 0 & : 0.3\pi < |\omega| \leq \pi \end{cases}$$

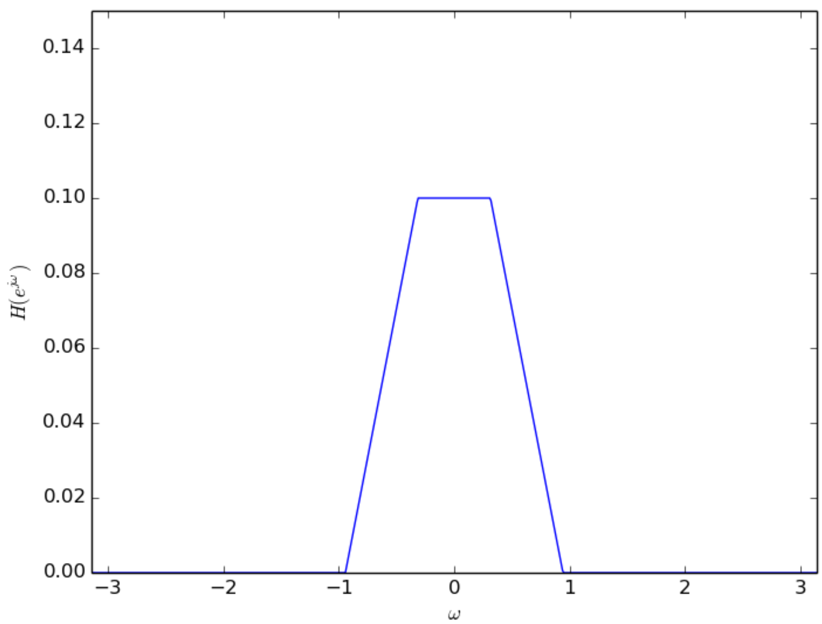


Figure 3: Trapezoidal Low-pass Filter

2.76

a) The product of LTI systems is not guaranteed to be an overall LTI system.

b) $Y(e^{j\omega})$ is only guaranteed to be zero in the region $0.6\pi < |\omega| \leq \pi$

Matlab Problem

I wrote all code in Python (see attached).

a) $s[n] = 2n(0.9)^n$

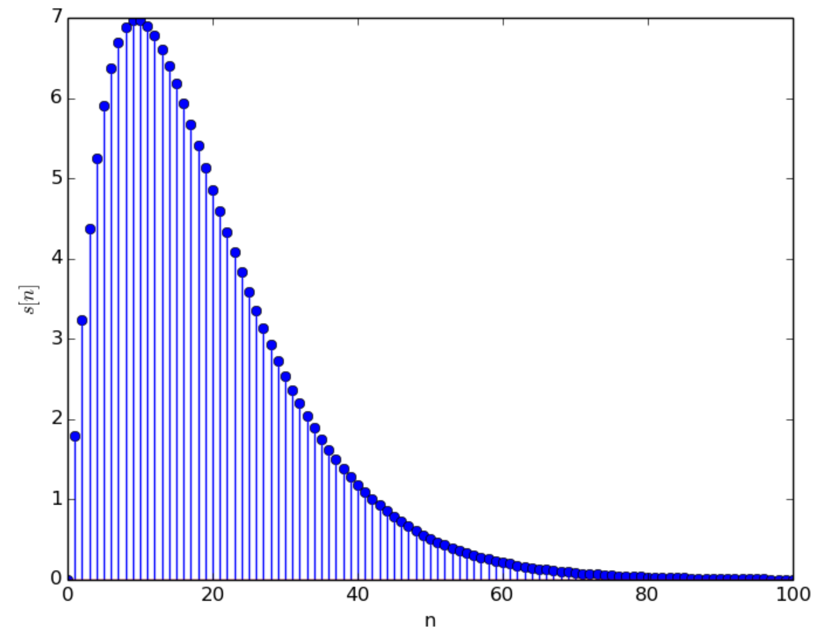


Figure 4: $s[n] = 2n(0.9)^n$

b) $w[n] =$ independent random Gaussian, $\mu = 0, \sigma^2 = 1$

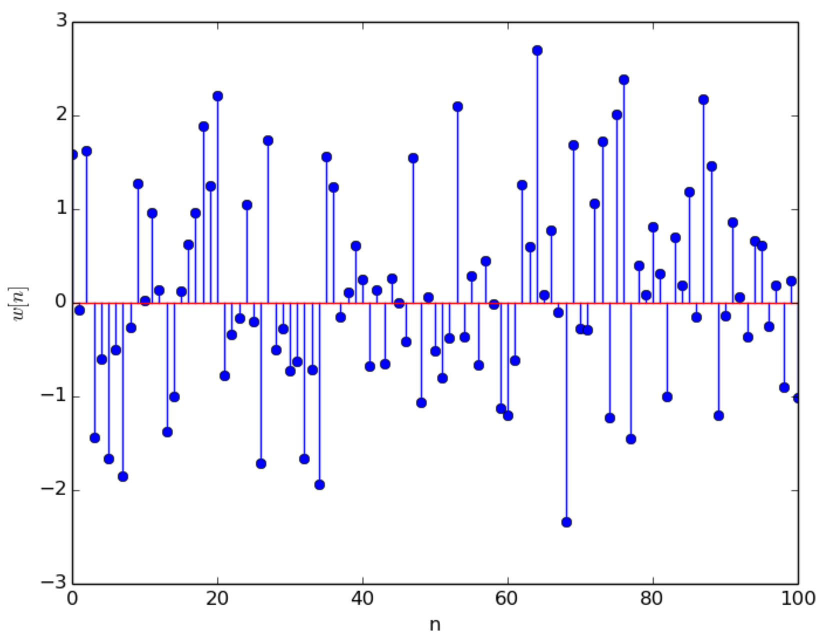


Figure 5: $w[n] =$ independent random Gaussian, $\mu = 0, \sigma^2 = 1$

c) $x[n] = s[n] + w[n]$

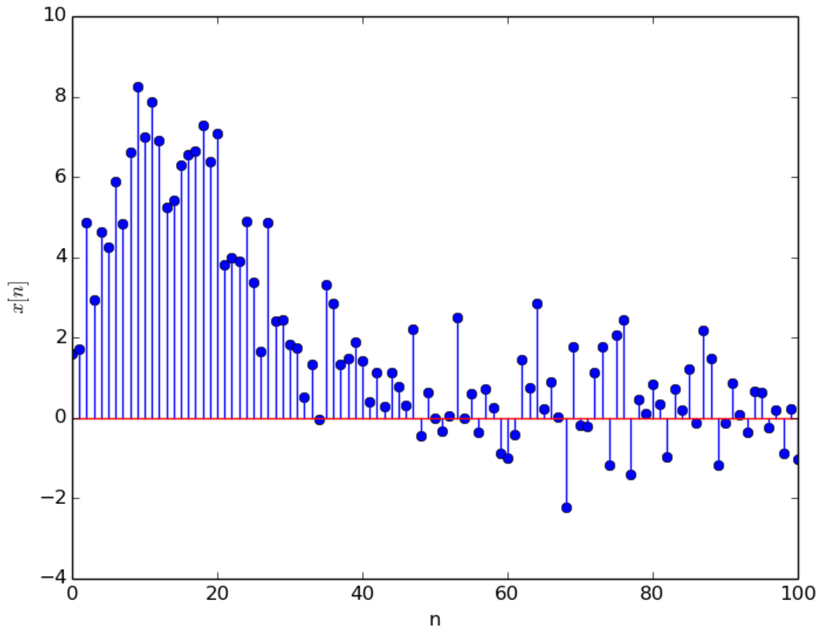


Figure 6: $x[n] = s[n] + w[n]$

d) $y[n]$ and $s[n]$

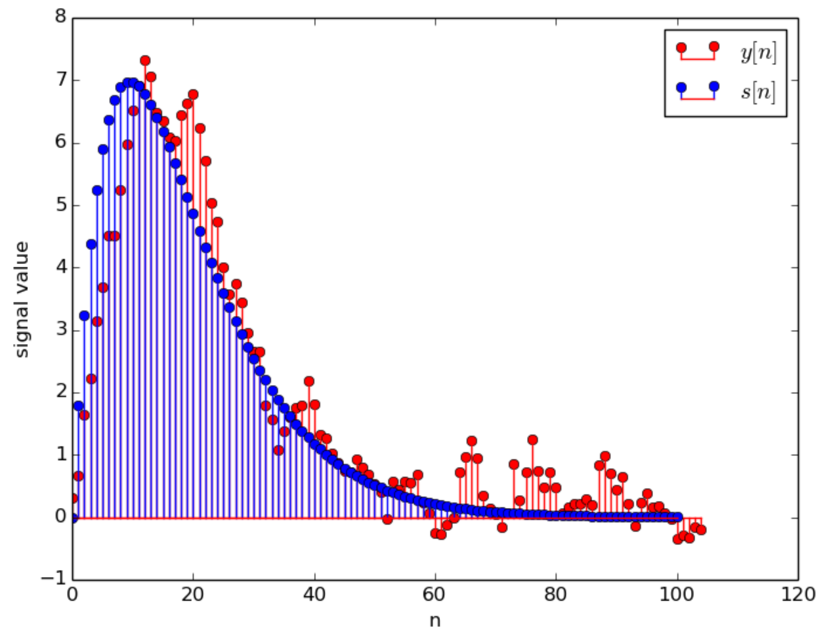


Figure 7: $y[n]$ and $s[n]$

e) Moving Average Filter applied to Interfered Signal

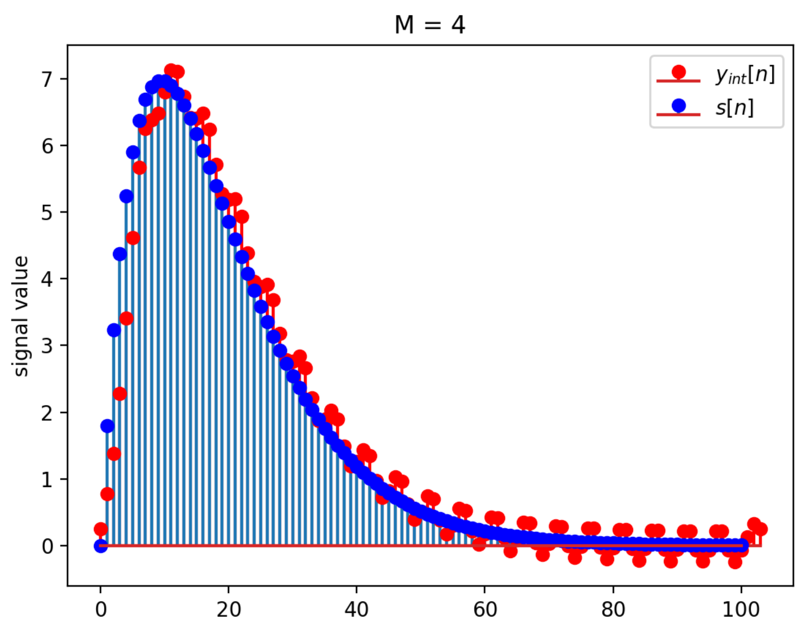


Figure 8: $x_{int}[n]$ filtered by 4-point MA

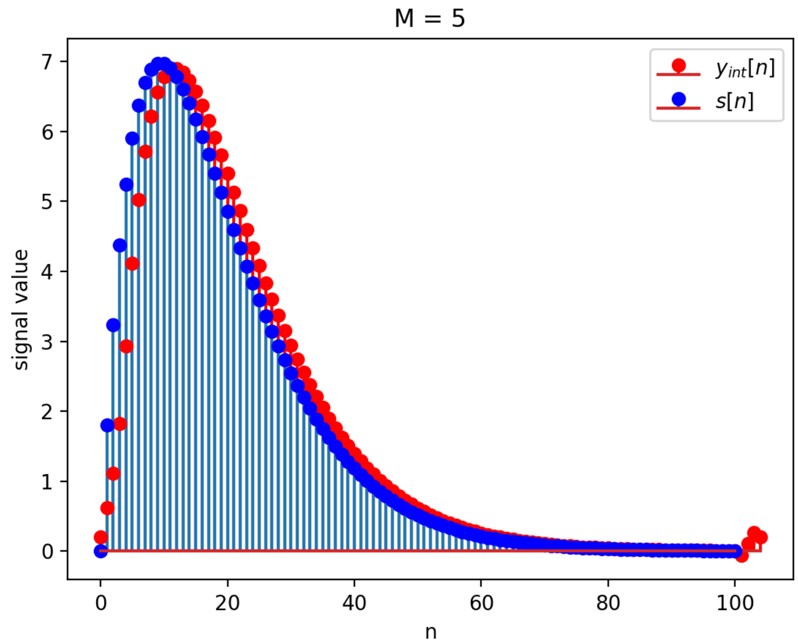


Figure 9: $x_{int}[n]$ filtered by 5-point MA

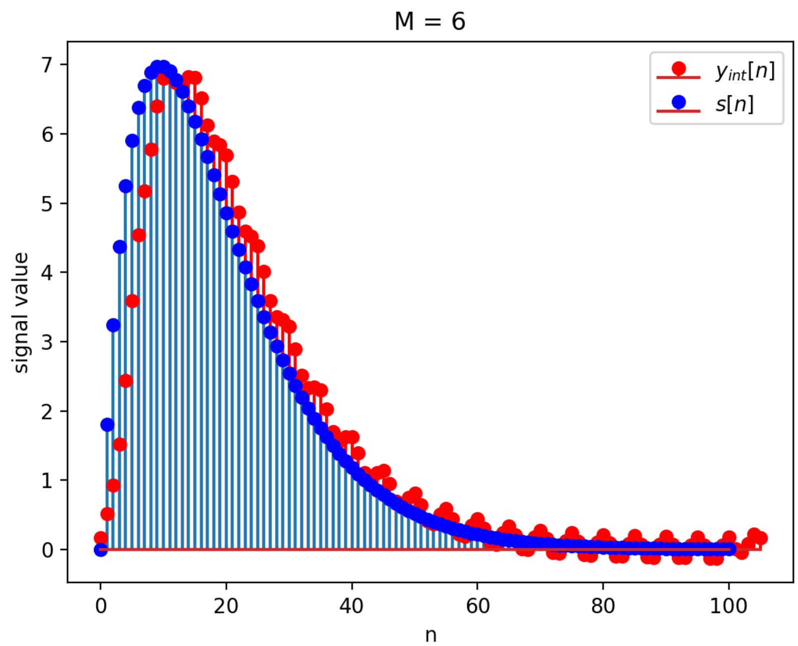


Figure 10: $x_{int}[n]$ filtered by 6-point MA

While the interference is not completely removed, the filter does a relatively good job smoothing over the interfered signal, as we can see it converges on the original, clean signal. It appears the M=5 is the most effective at removing the interference, and that M=4 and M=6 result in the signal oscillating about the true, original values. I am guessing that the window size being odd vs. even may play a role here.

Noah Schwab
ESE 531

01/30/2

HW 1: Discrete-time Signals and Systems

2.23

a) $T(x[n]) = (\cos \pi n)\, x[n]$

(1) $\cos \pi n$ evaluates to either $+1$ or -1 if n is even or odd, respectively. Therefore, if $x[n]$ is bounded, the system output is also bounded since the output is identical to the input, aside a sign flip at odd indices. Therefore, this system is stable.

(2) The system output only depends on the current iteration, $x[n]$. There is no indication that it looks into the future. Therefore it is causal.

(3) To prove linearity, we must demonstrate
$$T(\alpha x_1[n] + \beta x_2[n]) = \alpha T(x_1[n]) + \beta T(x_2[n])$$

Consider $y_1[n] = T(x_1[n])$ and $y_2[n] = T(x_2[n])$
$$y'[n] = T(x'[n])$$
$$x'[n] = \alpha x_1[n] + \beta x_2[n]$$

Substitution : $y'[n] = T(\alpha x_1[n] + \beta x_2[n])$

Apply system: $= (\cos \pi n)(\alpha x_1[n] + \beta x_2[n])$

$= \alpha(\cos \pi n)\, x_1[n] + \beta(\cos \pi n)\, x_2[n]$

$= \alpha T(x_1[n]) + \beta T(x_2[n])$

$\Rightarrow T(\alpha x_1[n] + \beta x_2[n]) = \alpha T(x_1[n]) + \beta T(x_2[n])$

Therefore, the system is linear

(4) To prove time-invariance, we must demonstrate
$$T(x[n-q]) = y[n-q], \quad \text{where } T(x[n]) = y[n]$$

Consider $x'[n] = x[n-q]$ and $y'[n] = T(x'[n])$

Substitution: $y'[n] = (\cos \pi n)\, x'[n] = (\cos \pi n)\, x[n-q]$

This does not satisfy $T(x[n-q]) = y[n-q]$

$$y[n-q] = (\cos(\pi(n-q)))\,x[n-q] \neq (\cos \pi n)\,x[n-q]$$

We can provide a simple counterexample.
Consider $q = 1$ and $x[n] = \delta[n]$.

$$T(\delta[n]) = (\cos(\pi n))\,\delta[n] = \delta[n]$$

$$T(\delta[n-1]) = (\cos(\pi n))\,\delta[n-1] = -\delta[n-1]$$

Therefore $T(x[n-q]) \neq y[n-q]$. It is not time-invariant

b) $T(x[n]) = x[n^2]$

(1) If $x[n]$ is bounded, then $x[n^2]$ must also be bounded. If we assume that $|x[n]| \leq B_x$ for all n, then $x[n^2]$ will also be upper bounded since the transformation does not affect the range of the output, only its domain. It is stable

(2) The output $x[n^2]$ depends on the value of the input signal at an index greater than n for all indices except $n=0$ and $n=1$. Therefore, it is not causal.

(3) Consider $y_1[n] = T(x_1[n])$ and $y_2[n] = T(x_2[n])$

$$y'[n] = T(x'[n])$$
$$x'[n] = \alpha x_1[n] + \beta x_2[n]$$

$$\begin{aligned} y'[n] &= T(\alpha x_1[n] + \beta x_2[n]) \\ &= \alpha x_1[n^2] + \beta x_2[n^2] \\ &= \alpha T(x_1[n]) + \beta T(x_2[n]) \\ &= \alpha y_1[n] + \beta y_2[n] \end{aligned}$$

Therefore, it is Linear

(4) Consider $x'[n] = x[n-q]$ and $y'[n] = T(x'[n])$

$$y'[n] = T(x'[n]) = x'[n^2] = x[n^2-q]$$

This does not satisfy $T(x[n-q]) = y[n-q]$

$$y[n-q] = x[(n-q)^2] = x[n^2-2nq+q^2] \neq x[n^2-q]$$

Simple counter example

Consider $q=2$ and $x[n] = \delta[n]$

$$T(\delta[n]) = \delta[n^2] = \delta[n]$$

$$T(\delta[n-2]) = \delta[n^2-2] = \delta[n-4]$$

Therefore, $T(x[n-q]) \neq y[n-q]$. This is not time-invariant.

c) $T(x[n]) = x[n] \sum_{k=0}^{\infty} \delta[n-k]$

(1) If $x[n]$ is bounded such that $|x[n]| \leq B_x < \infty$ for all n, then the output must also be bounded. This is due to the fact that the transformation yields the same signal as the input, except all values prior to $n=0$ are zeroed out. It is stable.

(2) Since the summation is only defined for positive k, the transformation only involves knowledge of signal values at the current and prior iterations. Therefore, it is causal.

(3) Consider $y_1[n] = T(x_1[n])$ and $y_2[n] = T(x_2[n])$

$$y'[n] = T(x'[n])$$
$$x'[n] = \alpha x_1[n] + \beta x_2[n]$$

$$\begin{aligned} y'[n] &= T(\alpha x_1[n] + \beta x_2[n]) \\ &= (\alpha x_1[n] + \beta x_2[n]) \sum_{k=0}^{\infty} \delta[n-k] \\ &= \alpha x_1[n] \sum_{k=0}^{\infty} \delta[n-k] + \beta x_2[n] \sum_{k=0}^{\infty} \delta[n-k] \\ &= \alpha T(x_1[n]) + \beta T(x_2[n]) = \alpha y_1[n] + \beta y_2[n] \end{aligned}$$

Therefore, it is linear

(4) Consider $x'[n] = x[n-q]$ and $y'[n] = T(x'[n])$

$$y'[n] = T(x'[n]) = x'[n] \sum_{k=0}^{\infty} \delta[n-k] = x[n-q] \sum_{k=0}^{\infty} \delta[n-k]$$

$$y[n-q] = T(x[n-q]) = x[n-q] \sum_{k=0}^{\infty} \delta[(n-q)-k]$$

This does not satisfy $T(x[n-q]) = y[n-q]$

$$y[n-q] = x[n-q] \sum_{k=0}^{\infty} \delta[(n-q)-k] \neq x[n-q] \sum_{k=0}^{\infty} \delta[n-k]$$

Simple counter example:

consider $x[n] = \delta[n]$ and $q = -1$

$$T(\delta[n]) = \delta[n] \sum_{k=0}^{\infty} \delta[n-k] = \delta[n]$$

$$T(\delta[n+1]) = \delta[n+1] \sum_{k=0}^{\infty} \delta[n-k] = 0$$

Therefore, $T(x[n-q]) \neq y[n-q]$. It is not time-invariant.

d) $T(x[n]) = \sum_{k=n-1}^{\infty} x[k]$

(1) This transformation yields the sum of the signal values from the previous iteration to ∞. Therefore, a bounded input may yield an unbounded output, such as $x[n] = u[n]$, which diverges as $n \to \infty$.

Therefore, it is unstable.

(2) The output signal value at index n depends on input signal values ahead in time. Therefore it is not causal.

(3) Consider $y_1[n] = T(x_1[n])$ and $y_2[n] = T(x_2[n])$

$y'[n] = T(x'[n])$

$x'[n] = \alpha x_1[n] + \beta x_2[n]$

$$y'[n] = T(\alpha x_1[n] + \beta x_2[n])$$
$$= \sum_{k=n-1}^{\infty} (\alpha x_1[k] + \beta x_2[k])$$
$$= \sum_{k=n-1}^{\infty} \alpha x_1[k] + \sum_{k=n-1}^{\infty} \beta x_2[k]$$

$$= \alpha \sum_{k=n-1}^{\infty} x_1[k] + \beta \sum_{k=n-1}^{\infty} x_2[k]$$
$$= \alpha T(x_1[n]) + \beta T(x_2[n])$$
$$= \alpha y_1[n] + \beta y_2[n]$$

Therefore, it is linear

(4) consider $x'[n] = x[n-q]$ and $y'[n] = T[x'[n]]$

$$y'[n] = T(x'[n]) = \sum_{k=n-1}^{\infty} x'[k] = \sum_{k=n-1}^{\infty} x[k-q]$$

$$y[n-q] = T(x[n-q]) = \sum_{k=n-q-1}^{\infty} x[k] = \sum_{k=n-1}^{\infty} x[k-q]$$

Therefore, $T(x[n-q]) = y[n-q]$. It is time-invariant.

2.36

$$H(e^{jw}) = \frac{(1-je^{-jw})(1+je^{-jw})}{1-0.8e^{-jw}} = \frac{1+e^{-j2w}}{1-0.8e^{-jw}} = \frac{1}{1-0.8e^{-jw}} + \frac{e^{-j2w}}{1-0.8e^{-jw}}$$

a) To find the impulse response, we must apply the iFT to the frequency response

$$h[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{jw}) e^{jwn} \, dw$$

Using the last term for the frequency response, Theorem 1 from Table 2.2 tells us that

$$h[n] = iFT\left(\frac{1}{1-0.8e^{-jw}}\right) + iFT\left(\frac{e^{-2jw}}{1-0.8e^{-jw}}\right)$$

Using Table 2.3 of Fourier Transform pairs, we can evaluate the iFTs w/o direct integration

$$iFT\left(\frac{1}{1-0.8e^{-jw}}\right) = (0.8)^n u[n]$$

$$iFT\left(\frac{e^{-j2w}}{1-0.8e^{-jw}}\right) = (0.8)^{n-2} u[n-2]$$

We were able to evaluate the second by recognizing it is identical to the first but w/ a modulation of $n_d = 2$. See theorem 3 in Table 2.2

Therefore,

$$\boxed{h[n] = (0.8)^n u[n] + (0.8)^{n-2} u[n-2]}$$

b) The frequency response can be expressed as the ratio of the output FT over the input FT. In other words, a difference equation

$$H(e^{jw}) = \frac{Y(e^{jw})}{X(e^{jw})} = \frac{1+e^{-j2w}}{1-0.8e^{-jw}}$$

$$\Rightarrow \quad Y(e^{jw}) - 0.8e^{-jw}Y(e^{jw}) = X(e^{jw}) + e^{-j2w}X(e^{jw})$$

Citing the Time shifting and Frequency shifting Theorem (Theorem 2, Table 2.2) and Linearity of the FT

$$y[n] - 0.8\,y[n-1] = x[n] + x[n-2]$$

$$\boxed{y[n] = 0.8\,y[n-1] + x[n] + x[n-2]}$$

c) system input: $x[n] = 4 + 2\cos(w_0 n) \quad -\infty < n < \infty$

The output can be expressed in terms of the frequency response:

$$y[n] = 4H(e^{j0}) + 2\,|H(e^{jw_0})|\cos(w_0 n + \theta)$$

where $\theta = \angle H(e^{jw_0})$ [see Example 2.15]

For $y[n]$ to be constant, $|H(e^{jw_0})| = 0$

Using the second form of the frequency response.

$$\frac{1+e^{-j2w_0}}{1-0.8e^{-jw_0}} = 0$$

$$\Rightarrow \quad 1 = -e^{-j2w_0}$$

Solving for w_0:

$$-\cos(-2w_0) - i\sin(-2w_0) = 1$$

$$\left.\begin{aligned} \cos(2w_0) &= -1 \\ \sin(2w_0) &= 0 \end{aligned}\right\} \quad \boxed{w_0 = \frac{\pi}{2}}$$

$$A = y[n]\Big|_{w_0=\frac{\pi}{2}} = 4H(e^{j0}) = 4\left(\frac{1+1}{1+0.8}\right)$$

$$\boxed{A = 40}$$

2.54

a) Using the block diagram we can find the overall impulse response.

$$h[n] = (\delta[n] + h_1[n]) * h_2[n]$$
$$= (\delta[n] + \beta\delta[n-1]) * \alpha^n u[n]$$
$$= \delta[n] * \alpha^n u[n] + \beta\delta[n-1] * \alpha^n u[n]$$

$$\boxed{h[n] = \alpha^n u[n] + \beta\alpha^{n-1} u[n-1]}$$

b) Apply FT to the impulse response to find frequency response. Table 2.3, Fourier Transform Pair 4 yields a solution;

$$H(e^{jw}) = \frac{1}{1-\alpha e^{-jw}} + \left(\frac{\beta}{1-\alpha e^{-jw}}\right) e^{-jw}$$

$$\boxed{H(e^{jw}) = \frac{1+\beta e^{-jw}}{1-\alpha e^{-jw}}} \quad \text{for } |\alpha| < 1$$

c) $$H(e^{jw}) = \frac{Y(e^{jw})}{X(e^{jw})} = \frac{1+\beta e^{-jw}}{1-\alpha e^{-jw}}$$

$$\Rightarrow Y(e^{jw})(1-\alpha e^{-jw}) = X(e^{jw})(1+\beta e^{-jw})$$
$$= \boxed{y[n] - \alpha y[n-1] = x[n] + \beta x[n-1]}$$

d) As we can see from part (a), $h[n] = 0$ for $n < 0$, therefore this system is causal.

To test stability, we can evaluate the sum
$$B_h = \sum_{n=-\infty}^{\infty} |h[n]|$$

In order for $B_h < \infty$, $\boxed{|\alpha| < 1 \text{ and } \beta < \infty}$. These are the stability conditions.

2.64

$$H_{1p}(e^{jw}) = \begin{cases} 1, & |w| < 0.2\pi \\ 0, & 0.2\pi \le |w| \le \pi \end{cases}$$

a) $h_1[n] = (-1)^n h_{1p}[n] = e^{j\pi n} h_{1p}[n]$

A modulation in the time-domain corresponds to a shift in the frequency domain.

$$H_1(e^{jw}) = H_{1p}(e^{j(w-\pi)}) = \begin{cases} 1, & |w-\pi| < 0.2\pi \\ 0, & 0.2\pi \le |w-\pi| \le \pi \end{cases}$$

Considering periodicity of frequency response

$$\boxed{H_1(e^{jw}) = \begin{cases} 1, & 0.8\pi \le |w| \le 1.2\pi \\ 0, & \text{else} \end{cases}}$$

See figure in write-up. This is an ideal high-pass filter.

b) $h_2[n] = 2h_{1p}[n]\cos(0.5\pi n)$

Multiplication in the time-domain corresponds to a convolution in the frequency domain. The frequency response of $\cos(\frac{\pi}{2}n)$ is as follows:

$$\sum_{k=-\infty}^{\infty} [\pi\delta(w - \tfrac{\pi}{2} + 2\pi k) + \pi\delta(w + \tfrac{\pi}{2} + 2\pi k)]$$

In the range $[-\pi, \pi]$, this corresponds to two unit impulses at $w = \pm\frac{\pi}{2}$, with amplitudes of π. The convolution w/ a shifted impulse is a shift of the signal. Therefore,

$$H_2(e^{jw}) = 2H_{1p}(e^{jw}) * (\pi\delta(w - \tfrac{\pi}{2}) + \pi\delta(w + \tfrac{\pi}{2}))$$

$$= \frac{2\pi H_{1p}(e^{j(\omega-\frac{\pi}{2})}) + 2\pi H_{1p}(e^{j(\omega+\frac{\pi}{2})})}{2\pi}$$

$$\boxed{H_2(e^{j\omega}) = \begin{cases} 1, & 0.3\pi \le |\omega| \le 0.7\pi \\ 0, & |\omega| < 0.3\pi,\ 0.7\pi < |\omega| < \pi \end{cases}}$$

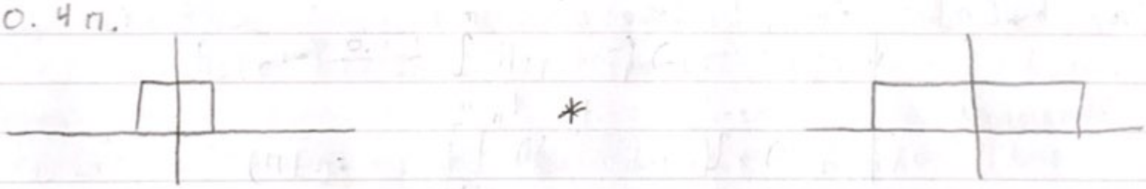
See figure in write-up. This is a <u>band-pass filter</u>

c) $h_3[n] = \frac{\sin(0.1\pi n)}{\pi n} h_{1p}[n]$

Again, we must use the time-convolution theorem. We can recognize that $\frac{\sin(0.1\pi n)}{\pi n}$ has a Fourier Transform that is a window centered at $\omega = 0$ with a width of 0.2π:

$$G(e^{j\omega}) = \begin{cases} 1, & |\omega| < 0.1\pi \\ 0, & 0.1\pi \le |\omega| \le \pi \end{cases}$$

$H_{1p}(e^{j\omega})$ is also a window, but with a width of 0.4π.



$$\boxed{H_3(e^{j\omega}) = \begin{cases} 0.1, & |\omega| \le 0.1\pi \\ \frac{\omega}{2\pi} + 0.15, & -0.3\pi \le \omega < -0.1\pi \\ -\frac{\omega}{2\pi} + 0.15, & 0.1\pi < \omega \le 0.3\pi \\ 0, & 0.3\pi < |\omega| \le \pi \end{cases}}$$

See figure in write-up This is a <u>trapezoidal low-pass filter</u>.

2.76

a) $y[n] = y_1[n]\, y_2[n] = S_1(x[n])\, S_2(x[n])$

To determine if the overall system is LTI, we must demonstrate it is both linear and time-invariant.

Since S_1 and S_2 are both linear, the following is true:

$$x[n] = \alpha z_1[n] + \beta z_2[n]$$

$$S_1(\alpha z_1[n] + \beta z_2[n]) = \alpha S_1(z_1[n]) + \beta S_1(z_2[n])$$

$$S_2(\alpha z_1[n] + \beta z_2[n]) = \alpha S_2(z_1[n]) + \beta S_2(z_2[n])$$

For the overall system S to be linear, we must prove $S(\alpha z_1[n] + \beta z_2[n]) = \alpha S(z_1[n]) + \beta S(z_2[n])$

$$S(\alpha z_1[n] + \beta z_2[n]) = S_1(\alpha z_1[n] + \beta z[n])\, S_2(\alpha z_1[n] + \beta z_2[n])$$

$$= (\alpha S_1(z_1[n]) + \beta S_1(z_2[n]))(\alpha S_2(z_1[n]) + \beta S_2(z_2[n]))$$

$$= \alpha^2 S_1(z_1[n])\, S_2(z_1[n]) + \alpha\beta S_1(z_1[n])\, S_2(z_2[n]) + \alpha\beta S_1(z_2[n])\, S_2(z_1[n]) + \beta^2 S_1(z_2[n])\, S_2(z_2[n])$$

From this, there is no clear way of reducing the expression to $\alpha S_1(z_1[n])\, S_2(z_1[n]) + \beta S_1(z_2[n])\, S_2(z_2[n])$.

Instead, we will provide a counter example.
Consider S_1 and S_2 are both identity transformations with $h_1[n] = h_2[n] = \delta[n]$. In this case,
$y_1[n] = y_2[n] = x[n]$

The overall system would yield
$y[n] = y_1[n]\, y_2[n] = x[n]\, x[n] = x[n]^2$

Clearly, $y[n] = x[n]^2$ is not a linear system.

$\boxed{\text{Therefore, the product of LTI systems is not guaranteed to be an overall LTI system}}$

b) We can use properties of the Fourier Transform to solve this problem.

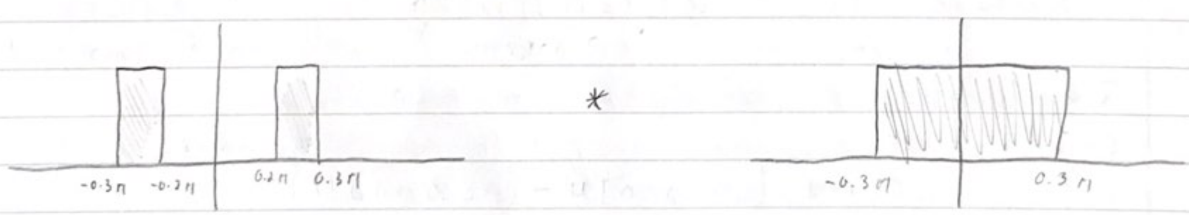
$$y[n] = y_1[n]\, y_2[n]$$
$$y_1[n] = x[n] * h_1[n]$$
$$y_2[n] = x[n] * h_2[n]$$

Substitution: $y[n] = (x[n] * h_1[n]) \times (x[n] * h_2[n])$

A convolution in time is a multiplication in frequency, and a multiplication in time is a convolution in frequency.

$$Y(e^{jw}) = X(e^{jw})H_1(e^{jw}) * X(e^{jw})H_2(e^{jw})$$

$$= \left\{ \begin{matrix} \text{unspecified} & 0.2\pi \le |w| \le 0.3\pi \\ 0, & \text{elsewhere} \end{matrix} \right\} * \left\{ \begin{matrix} \text{unspecified} & |w| < 0.3\pi \\ 0, & \text{elsewhere} \end{matrix} \right\}$$



By visualizing the convolution between the two spectra, we can see that the only region guaranteed to be zero is $\boxed{|w| > 0.6\pi}$.

```python
# This script plots the frequency response of a low-pass filter and transformed versions
import numpy as np
import matplotlib.pyplot as plt

# frequency array
w_array = np.arange(-np.pi, np.pi + 1, 0.01)

# low-pass filter
H1p = []
for w in w_array:
    if np.abs(w) < 0.2 * np.pi:
        H1p.append(1)
    else:
        H1p.append(0)

# plot low-pass filter
plt.plot(w_array, H1p)
plt.xlim(-np.pi, np.pi)
plt.ylim(0, 1.5)
plt.xlabel('$\omega$')
plt.ylabel('$H(e^{j\omega})$')
plt.show()

# part (a)
H1 = []
for w in w_array:
    if (np.abs(w) < 1.2 * np.pi) and (np.abs(w) > 0.8 * np.pi):
        H1.append(1)
    else:
        H1.append(0)

# plot H1
plt.plot(w_array, H1)
plt.xlim(-np.pi, np.pi)
plt.ylim(0, 1.5)
plt.xlabel('$\omega$')
plt.ylabel('$H(e^{j\omega})$')
plt.show()

# part (b)
H2 = []
for w in w_array:
    if (np.abs(w) < 0.7 * np.pi) and (np.abs(w) > 0.3 * np.pi):
        H2.append(1)
    else:
        H2.append(0)

# plot H2
plt.plot(w_array, H2)
plt.xlim(-np.pi, np.pi)
plt.ylim(0, 1.5)
plt.xlabel('$\omega$')
plt.ylabel('$H(e^{j\omega})$')
plt.show()

# part (c)
H3 = []
for w in w_array:
    if (np.abs(w) < 0.1 * np.pi):
        H3.append(0.1)
    elif (w >= -0.3 * np.pi) and (w <= -0.1 * np.pi):
        H3.append(w / (2 * np.pi) + 0.15)
    elif (w >= 0.1 * np.pi) and (w <= 0.3 * np.pi):
        H3.append(-w / (2 * np.pi) + 0.15)
    else:
        H3.append(0)
```

```
# plot H3
plt.plot(w_array, H3)
plt.xlim(-np.pi, np.pi)
plt.ylim(0, 0.15)
plt.xlabel('$\omega$')
plt.ylabel('$H(e^{j\omega})$')
plt.show()
```

```python
# ESE 531: HW1 Problem 2
# M-point Moving Average Filter

# import libraries
from turtle import color
import numpy as np
import matplotlib.pyplot as plt

# PART A

# generate defined signal
s = np.array([2 * n * 0.9 ** n for n in range(101)])

# PART B

# random gaussain noise
w = np.array([np.random.normal(0, 1) for n in range(101)])

# PART C

# define signal x as s plus noise
x = s + w

# plot all three signals
plt.stem(s)
plt.xlabel('n')
plt.ylabel('$s[n]$')
plt.show()

plt.stem(w)
plt.xlabel('n')
plt.ylabel('$w[n]$')
plt.show()

plt.stem(x)
plt.xlabel('n')
plt.ylabel('$x[n]$')
plt.show()


# PART D

# Moving Average Function
'''
Arguments:
    x: input signal
    m: window size of moving average
Output:
    y: averaged signal
'''
def moving_avg(x, m):

    # the impulse response of the moving average filter is a scaled and shifted window
    h = np.ones(m) / m

    # to apply the filter, we simply convolve the signal with the impulse response
    y = np.convolve(x, h)

    return y

# apply 5-point moving average filter to x[n]
y = moving_avg(x, 5)
plt.stem(y, label='$y[n]$', markerfmt='ro', linefmt='r-')
plt.stem(s, label='$s[n]$', markerfmt='bo')
plt.xlabel('n')
plt.ylabel('signal value')
plt.legend()
```

```
plt.show()

# PART E

# generate interference signal
f = 0.2
w_int = np.cos([2 * np.pi * f * n for n in range(101)])

# interfered signal
x_int = s + w_int

# Filter interfered signal with moving average filter of variable window sizes
m_list = [4, 5, 6]
for m in m_list:
    y_int = moving_avg(x_int, m)
    plt.stem(y_int, label='$y_{int}[n]$', markerfmt='ro', linefmt='r-')
    plt.stem(s, label='$s[n]$', markerfmt='bo')
    plt.title('M = {}'.format(m))
    plt.xlabel('n')
    plt.ylabel('signal value')
    plt.legend()
    plt.show()
```