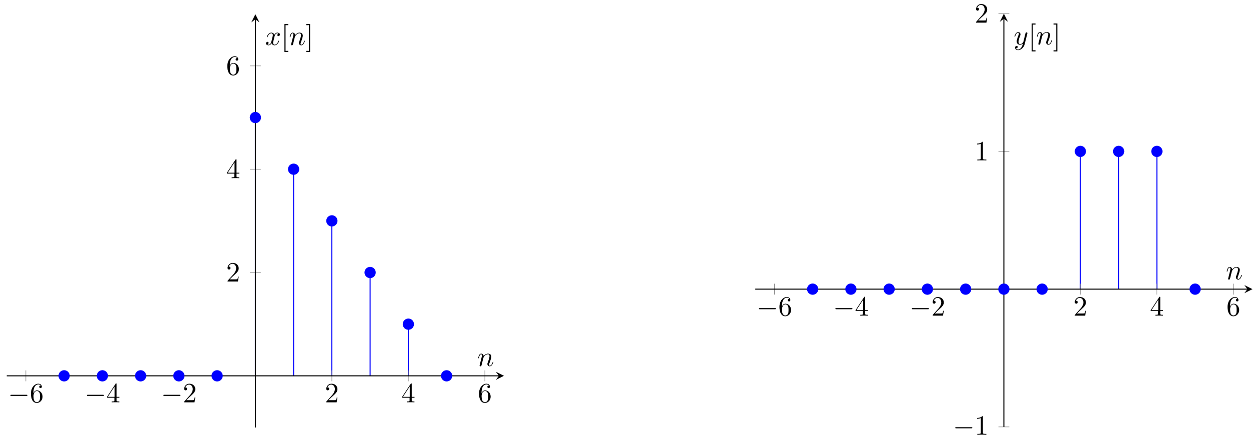
TTT4120 Assignment 1

Øyvind Skaaden (oyvindps)

September 2, 2020

Problem 1.

(a) The signals $x[n]$ og $y[n]$ are sketched in Figure 1.

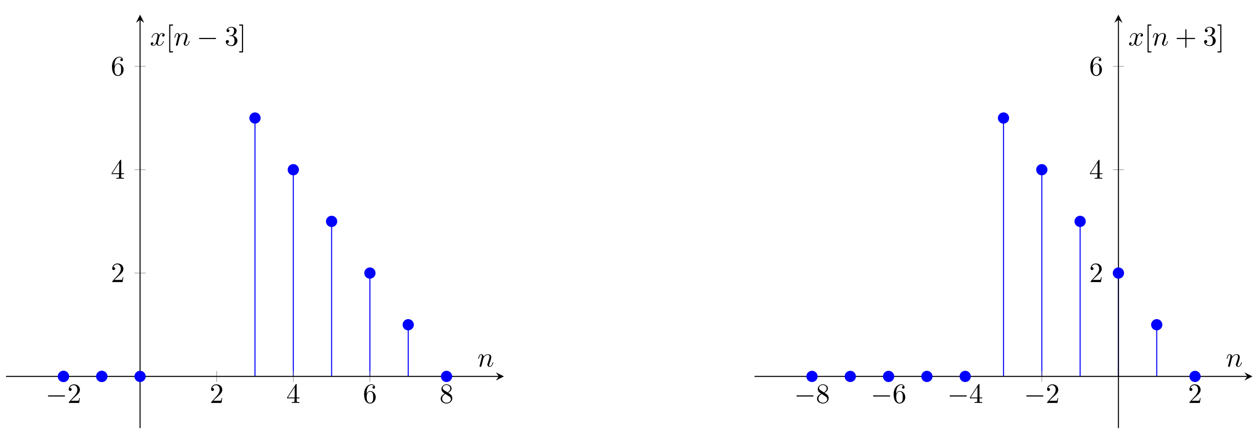


(a) Plot of $x[n]$.

(b) Plot of $y[n]$.

Figure 1: Signalene $x[n]$ og $y[n]$ plottet.

(b) The signals $x[n-3]$ and $x[n+3]$ are sketched in Figure 2.



(a) Plot of $x[n-3]$.

(b) Plot of $x[n+3]$.

Figure 2: Signalene $x[n-3]$ og $x[n+3]$ plottet.

(c) Plot for $x[-n]$.

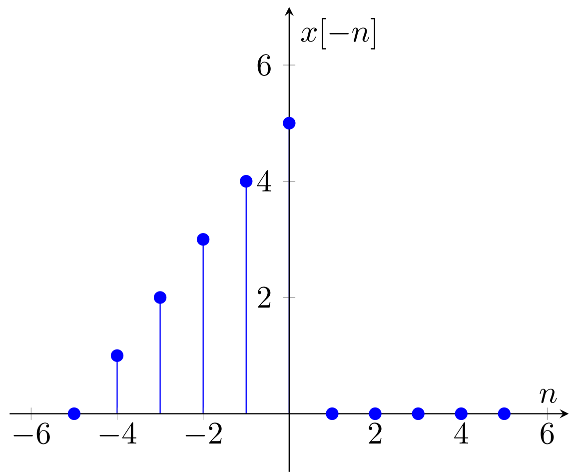


Figure 3: $x[-n]$

(d) Plot for $x[5-n] = x[-(n-5)]$.

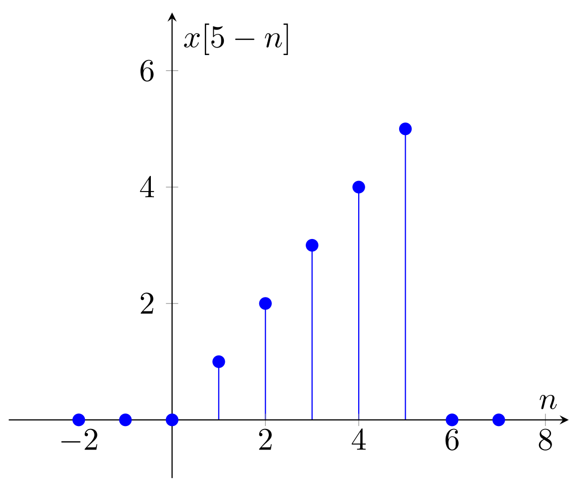


Figure 4: $x[5-n] = x[-(n-5)]$

(e) Plot for $x[n] \cdot y[n]$.

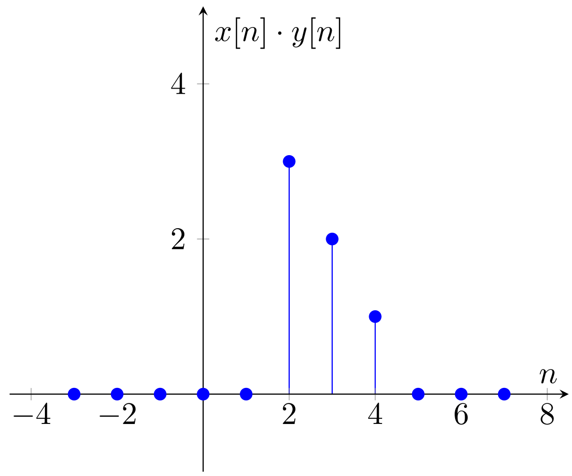


Figure 5: $x[n] \cdot y[n]$

(f) The sequence

$$x[n] = \begin{cases} 5-n & 0 \leq n \leq 4 \\ 0 & otherwise \end{cases} \tag{1}$$

can be described in simple form

$$x[n] = 5\delta[n] + 4\delta[n-1] + 3\delta[n-2] + 2\delta[n-3] + \delta[n-4] \tag{2}$$

(g) The sequence

$$y[n] = \begin{cases} 1 & 2 \leq n \leq 4 \\ 0 & otherwise \end{cases} \tag{3}$$

can be described in simple form

$$x[n] = u[n-2] - u[n-5] \tag{4}$$

(h) The energy in $x[n]$ are given by:

$$E_x = \sum_{n=-\infty}^{\infty} |x[n]|^2 \tag{5}$$

$$= \sum_{n=0}^{4} (n-5)^2 \tag{6}$$

$$= 25 + 16 + 9 + 4 + 1 = 55 \tag{7}$$

Problem 2.

(a) We have that the physical frequencies for F_1 must be of a value that makes $f_1 = \frac{F_1}{F_S} \in \left[-\frac{1}{2}, \frac{1}{2}\right)$. By inserting $F_S = 6000\text{Hz}$ we get that $F_1 \in [-3000\,\text{Hz}, 3000\,\text{Hz})$.

(b) Code used for generating the different signals in Listing 1.

```
import numpy as np
import sounddevice as sd

# Physical freq to "sample", must be integer
F_1 = 1000
# Volume must be between 0 and 100, may be float
volume = 20
# Sampling freq, must be integer
F_s = 6000
# Time how long a sample should last, must be integer
duration = 4

# Calculation of constants
volume = volume / 100
totalSamples = F_s * duration

# Comment out the last decleration if you want to use a fixed f1
f_1 = 0.3
#f_1 = F_1 / F_s

def GenSound(f1, noSamples, vol):
    x = np.empty(noSamples)
    for n in range(noSamples):
        x[n] = np.cos(2*np.pi * f1 * n) * vol
    return x

x = GenSound(f_1, totalSamples, volume)

print(x)

sd.play(x, F_s)
sd.wait()
sd.stop()
```

Listing 1: Code used to generate and play the tones in the following parts.

(c) When increasing the sampleling rate F_S, with a fixed $f_1 = 0.3$, the tone gets brighter.

If we look at the formula for the physical frequency, given f_1 and F_S.

$$F_1 = f_1 \cdot F_S \tag{8}$$

Whitch for $F_S = [1000\,\text{Hz}, 3000\,\text{Hz}, 12000\,\text{Hz}]$, gives $F_1 = [300\,\text{Hz}, 900\,\text{Hz}, 3600\,\text{Hz}]$ respectivly.

(d) This is the reverse problem from **(c)**. We use the formula (9), and use $F_S = 8000\,\text{Hz}$ and $F_1 = [1000\,\text{Hz}, 3000\,\text{Hz}, 6000\,\text{Hz}]$.

$$f_1 = \frac{F_1}{F_S} \tag{9}$$

From (9) we get $f_1 = \left[\frac{1}{8}, \frac{3}{8}, \frac{6}{8} = \frac{3}{4}\right]$. Since $f_1 \in \left[-\frac{1}{2}, \frac{1}{2}\right]$, and is "periodic", we can see that $f_1 = \frac{3}{4}$ is the same as $f_1 = 1 - \frac{3}{4} = -\frac{1}{4}$. When we calculate the actual frequencies, we get as in

$$F_{sampled} = \begin{cases} 1000\,\text{Hz} & \text{when } f_1 = \frac{1}{8} \\ 3000\,\text{Hz} & \text{when } f_1 = \frac{3}{8} \\ -2000\,\text{Hz} & \text{when } f_1 = -\frac{1}{4} \end{cases} \tag{10}$$

When we see a negative frequency, it just means that it is phase-shifted by π.

Problem 3.

(a) $y[n] = x[n] - x^2[n-1]$. Since we have an x^2-part, the equation is non linear. This is causal because we only use samples that are present or past. And this is also time-invariant, since none of the coefficients are dependent of n.

(b) $y[n] = nx[n] + 2x[n-2]$. Here, the equation only consists of linear combinations of $x[n]$, therefore the total equation is linear. Since we have one of the coefficients depending on n this is not time-invariant. This is also causal, because it only is dependent on samples from present or past.

(c) $y[n] = x[n] - x[n-1]$. This equation is a linear combination of $x[n]$'s, and therefore linear. This is time-invariant, because it does not have coefficients depending on n. This is causal, because it only uses samples from the present or past.

(d) $y[n] = x[n] + 3x[n+4]$. This equation is a linear combination of $x[n]$'s, and therefore linear. This is time-invariant, because it does not have coefficients depending on n. Since this is dependent on a sample from the future, $3x[n+4]$, the equation is non-causal.

Problem 4.

We have the following two difference equations.

$$y_1[n] = x[n] + 2x[n-1] + x[n-2] \tag{11}$$
$$y_2[n] = -0.9y[n-1] + x[n] \tag{12}$$

(a) The unit sample response is possible to obtain by sending in a unit sample into the equation.

For (11), we simplify and use the unit sample.

$$h_1[n] = \delta[n] + 2\delta[n-1] + \delta[n-2] \tag{13}$$
$$= \begin{cases} 1 & n = 0 \\ 2 & n = 1 \\ 1 & n = 2 \\ 0 & otherwise \end{cases} \tag{14}$$

For (12) we use the same trick, but this is dependent on the last processed sample.

$$h_2[n] = -0.9h_2[n-1] + \delta[n] \tag{15}$$

We know from the original equation that (12) is causal, therefore must for $n < 0$ the IR be zero.

$$h_2[n] = 0, \qquad n < 0 \tag{16}$$
$$\Rightarrow h_2[0] = -0.9h_2[-1] + \delta[0] = 0 + 1 = 1 \tag{17}$$

We can from this see that the IR is recursive, where it only depends on the previous sample $-0.9h[n-1]$.

By doing some repetitions, we can find a more generic formula.

$$h_2[0] = 1 \tag{18}$$
$$h_2[1] = -0.9 \cdot 1 = -0.9 \tag{19}$$
$$h_2[2] = -0.9 \cdot -0.9 = (-0.9)^2 \tag{20}$$
$$h_2[3] = -0.9 \cdot (-0.9)^2 = (-0.9)^3 \tag{21}$$
$$\vdots$$
$$h_2[n] = (-0.9)^n = (-1)^n \cdot (0.9)^n \qquad n \geq 0 \tag{22}$$
$$h_2[n] = (-1)^n \cdot (0.9)^n \cdot u[n] \tag{23}$$

The IR of $y_2[n]$ are an alternating exponentially decaying function.

(b) Since $h_1[n]$ has a finite amount of samples in the IR, this is a FIR.

$h_2[n]$ does not have a boundry of how long it could keep on going, so this is a IIR.

(c) To check if a system is stable or not. We need to check that (24) holds for each of the filters.

$$\sum_{n=-\infty}^{\infty} |h[n]| < \infty \tag{24}$$

Starting with $h_1[n]$.

$$\begin{aligned}
\sum_{n=-\infty}^{\infty} |h_1[n]| &= \sum_{n=0}^{2} |h_1[n]| && (25)\\
&= 1 + 2 + 1 = 4 && (26)\\
&\Rightarrow h_1[n] \text{ is stable.}
\end{aligned}$$

$h_2[n]$:

$$\begin{aligned}
\sum_{n=-\infty}^{\infty} |h_2[n]| &= \sum_{n=0}^{\infty} |(-0.9)^n| && (27)\\
&= \sum_{n=0}^{\infty} (0.9)^n && (28)\\
&= \frac{1}{1-0.9} \quad | \text{ using sum of infinite geometric series} && (29)\\
&= 10 && (30)\\
&\Rightarrow h_2[n] \text{ is stable.}
\end{aligned}$$

(d) In Figure 6 and 7, you can see the filter structure for the different equations.

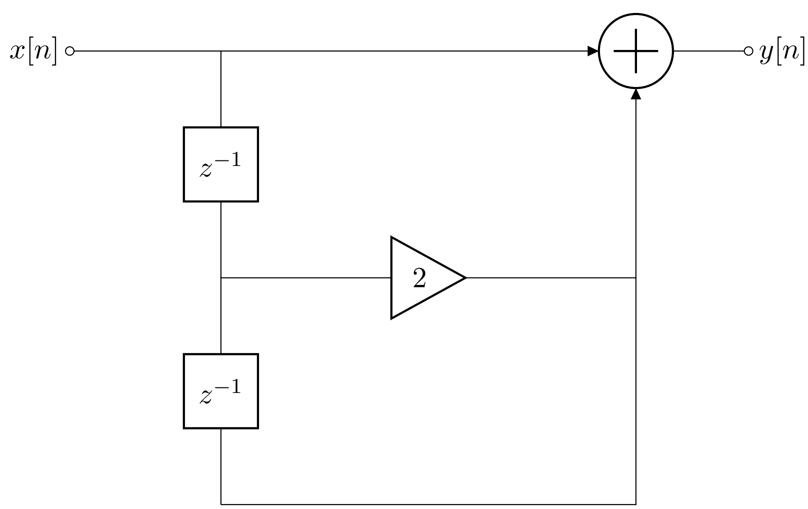


Figure 6: Filter structure of $y_1[n] = x[n] + 2x[n-1] + x[n-2]$.

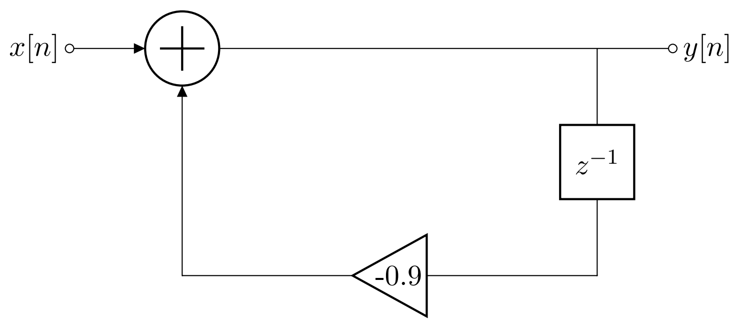


Figure 7: Filter structure of $y_2[n] = -0.9y[n-1] + x[n]$.

Problem 5.

(a) Plot for $y_1[n]$.

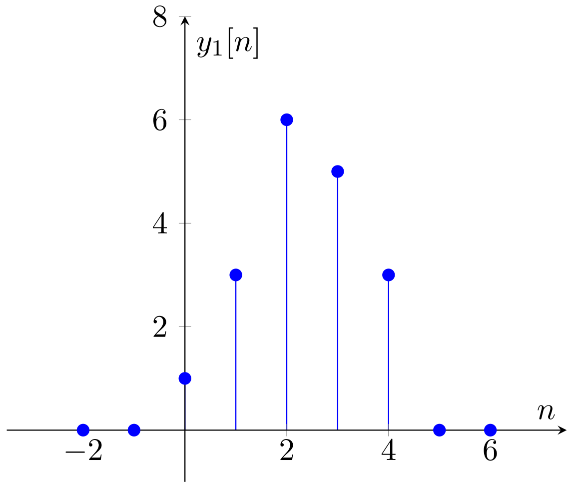


Figure 8: $y_1[n]$

(b) Plot for $y_2[n]$.

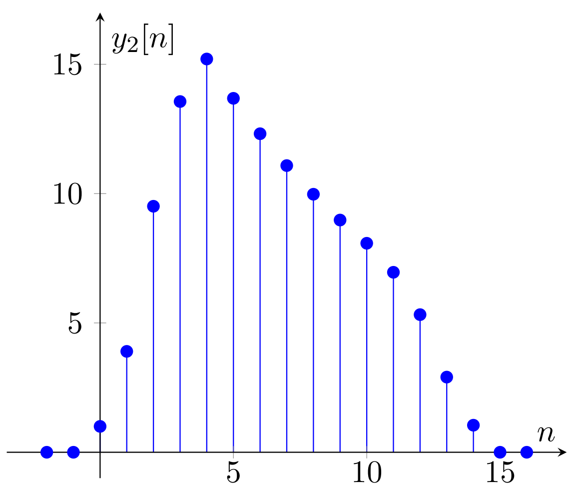


Figure 9: $y_2[n]$

(c)

(d) Since convolution is commutative, it does not matter in what order we use the filters. The output would still be like in Figure 9. The more interesting part is what happens at the point $y_1[n]$, when the order is changed. Plot for $y_{h_2}[n]$ after using h_2 instead of h_1.

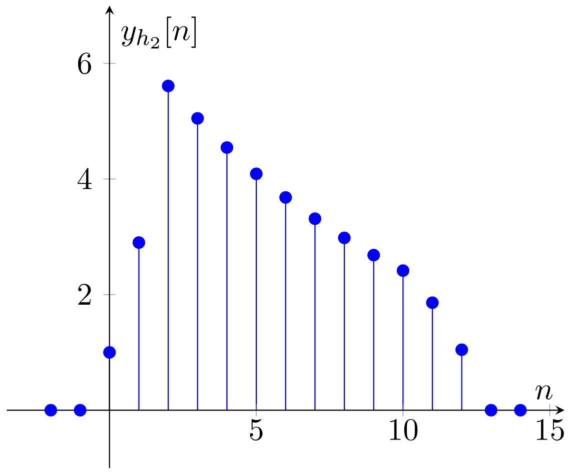


Figure 10: $y_{h_1}[n]$