Ex02 — Discrete-time signals and systems

Delivery deadline: **9 September** 2022.

We highly recommend you to collaborate and interact with each other when working on these tasks. Still, the exercises shall be solved independently and must be assessed in the weekly submission sessions. It is expected that everyone is familiar with the approval requirements for exercises. This is described on the course homepage: `http://www.uio.no/studier/emner/matnat/ifi/IN3190/h22/curriculum-related/`

Observe: The sum of points in total is 14, but the maximum score obtainable in one week is 10.

Goals:

- The course IN3190/4190 requires you to be familiar with the notation and operations relevant for discrete time series (digital signals).
- The *impulse response* is an important concept in digital signal processing because it can determine all properties of a linear time-invariant (LTI) system. We will work more on impulse responses later, but here you will have a first look into how it can be evaluated when a difference equation is provided that describes the relation between the input signal $x(n)$ and the output signal $y(n)$.
- We also would like you to work on some basic coding which in addition lets you build some in-depth understanding of the discrete *convolution* operator.

Exercise 1—Basic operations — 3 Points

The sequences $x_1(n)$ and $x_2(n)$ are given as:

$$\begin{aligned} x_1(n) &= \{-3, 2, \underset{\uparrow}{2}, 1, 0, 4, -1\} \\ x_2(n) &= \{-1, 2, -3, \underset{\uparrow}{-3}, 0\} \end{aligned}$$

Perform the following operations:

a) $y_1(n) = x_1(n) + x_2(n)$.

b) $y_2(n) = \frac{1}{3}x_1(n) - \frac{2}{3}\, x_2(n)$.

c) $y_3(n) = x_1(n)\, x_2(n)$. *Note: by this notation we mean element-wise multiplication between $x_1(n)$ and $x_2(n)$.*

Exercise 2— System classification — 3 Points

In the systems described by the difference equations below, $x(n)$ is the input and $y(n)$ is the output.

Determine whether each system is:

- Linear / non-linear
- Time invariant / Time variant
- Dynamic / static
- Causal / non-causal

a) $y(n) = x(n) + y(n-1)$

b) $y(n) - y(n+1) = 12x(n+2)$

c) $y(n) = x(n) + 2^n y(n)$

a) Linear, time-invariant, dynamic, causal

b) Linear, time-invariant, dynamic, non-causal

c) Linear, time-varying, static, causal

Exercise 3— Impulse response — 3 Points

Find the impulse response $h(n)$ to the system described by the difference equation (1) below.

You can do this helped by recursion where determine what $y(n)$ becomes, starting for $n = 0$ and then continuing up to $n = 4$.

The impulse response $h(n)$ hence is found by solving for the $y(n)$ output found for the case when the input $x(n)$ is equal to an impulse $\delta(n)$. This impulse $\delta(n)$ is zero everywhere apart from at $n = 0$, where $\delta(0) = 1$.

Hence, your task is to find $y(n)$ from Eq. (1), for $n = 0, 1, 2, 3$ given that the input $x(n) = \delta(n)$.

When you find $y(1)$, you can use the result you found for $y(0)$.

When you find $y(2)$, you can use the result you found for $y(0)$ and $y(1)$, and so on.

The difference equation describing the system (the relation between input and output) is:

$$y(n) - 3y(n-1) + 6y(n-2) = x(n-1). \tag{1}$$

Note that the case of the input $x(n)$ being equal to the impulse $\delta(n)$ and the output $y(n)$ being equal to the impulse response $h(n)$, we can write Eq. (1) as

$$h(n) - 3h(n-1) + 6h(n-2) = \delta(n-1). \tag{2}$$

We assume that $y(0) = 0$. Hence, the system is *relaxed* and $y(n)$ does not have signal (is equal to zero) for all $n \leq 0$.

$h(n) = \{\overset{\downarrow}{0}, 1, 3, 3, -9\}$

Exercise 4— Matlab/Python — 5 Points

Implement a function `function c = my_convolve(a,b)`, in Matlab, or `def my_convolve(a, b)` in Python, that takes as input two vectors or numpy arrays of arbitrary length a and b and which returns the convolution $c = a * b$. The function shall be implemented using for loops and shall follow the operator definition closely. Hence, you don't have to consider performance when implementing.

Check that your function provides the correct output as the builtin Matlab / numpy convolve functions, `conv(a,b)` / `numpy.convolve(a,b)`.