Ex02 — Discrete-time signals and systems

Delivery deadline: **29 August** 2025.

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/h25/curriculum-related/`

Observe: The sum of points in total is 16, 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).
- For some of the problems, we provide the answer already in the task text. For these cases, you shall show how you can arrive at this solution.
- Note that the Manolakis and Ingle book uses the square bracket notation for discrete sequences, for example, $x[n]$. Some other literature instead use the notation $x(n)$ using conventional parentheses. In the current task set, we will see both notations.
- For the even / odd decomposition of a signal, remember that for an even sequence you have $x_e[n] = x_e[-n]$ and for an odd sequence you have $x_o[n] = -x_o[-n]$.
- For the varius *System classification* aspects, you can find an overview of these criteria in the *Curriculum topic list* PDF available from the Course web sub-page on Curriculum & instruction resources: `https://www.uio.no/studier/emner/matnat/ifi/IN3190/h24/curriculum-related/index.html`.
- 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 $y(n)$.
- You will get the chance to download time series data from a real earthquake, recorded at a seismic instrument located on our campus. You will make some basic plotting and time shifting of these data.
- 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—Decomposition into even and odd sequence 2 Points

Any signal $x(n)$ can be decomposed into an even part $x_e(n)$ and an odd part $x_o(n)$. Show that the following relations:

$$x_e(n) = \frac{1}{2}\left(x(n) + x(-n)\right) \tag{1}$$

$$x_o(n) = \frac{1}{2}\left(x(n) - x(-n)\right) \tag{2}$$

are consistent with $x(n) = x_e(n) + x_o(n)$.

Exercise 3— 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) $\boxed{\text{Linear, time-invariant, dynamic, causal}}$

b) $\boxed{\text{Linear, time-invariant, dynamic, non-causal}}$

c) $\boxed{\text{Linear, time-varying, static, causal}}$

Exercise 4— Impulse response 5 Points

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

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

This difference equation provides our relation between an input $x(n)$ and an output $y(n)$. The system is *relaxed* and $y(n)$ does not have signal (is equal to zero) for all $n < 0$. Therefore, for example, we have that $y(-1) = 0$.

Hint 1: To find the impulse response, you can insert an impulse as input: $x(n) = \delta(n)$. Then the output will $y(n)$ is the impulse response, which we denote $h(n)$. In other words, for the case when the input $x(n)$ is an impulse $\delta(n)$, then the output $y(n)$ becomes equal to the impulse response $h(n)$.
With this perspective, if we set $x(n) = \delta(n)$ and $y(n) = h(n)$ in Eq. (3), we can write:

$$h(n) - 3h(n-1) + 6h(n-2) = \delta(n-1). \tag{4}$$

Hint 2: You can solve the problem using recursion. Then you determine $h(n)$ for all n, starting with $n = 0$ and then continuing up to $n = 4$. The impulse response $h(n)$ is then found by solving for the $y(n)$ output 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. (3) (or Eq. (4)), for $n = 0, 1, 2, 3$ given that the input is set to $x(n) = \delta(n)$.

- When you determine $y(0)$, you can use that $y(-1) = 0$ and $y(-2) = 0$, since the system is relaxed and $y(n) = 0$ for all $n < 0$.
- When you determine $y(1)$, you can use the result found for $y(0)$ and $y(-1)$.
- When you determine $y(2)$, you can use the result found for $y(0)$, and $y(1)$, and so on.

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

Exercise 5—Basic data analysis **3 Points**

A Python code was run in order to generate and save a set of 100 time series. Each of these is a recording taken from a seismometer located at the Oslo University campus, and each of theses includes waves arriving from a significant earthquake. Figure 1 displays the geographical distribution of the events and their magnitudes.

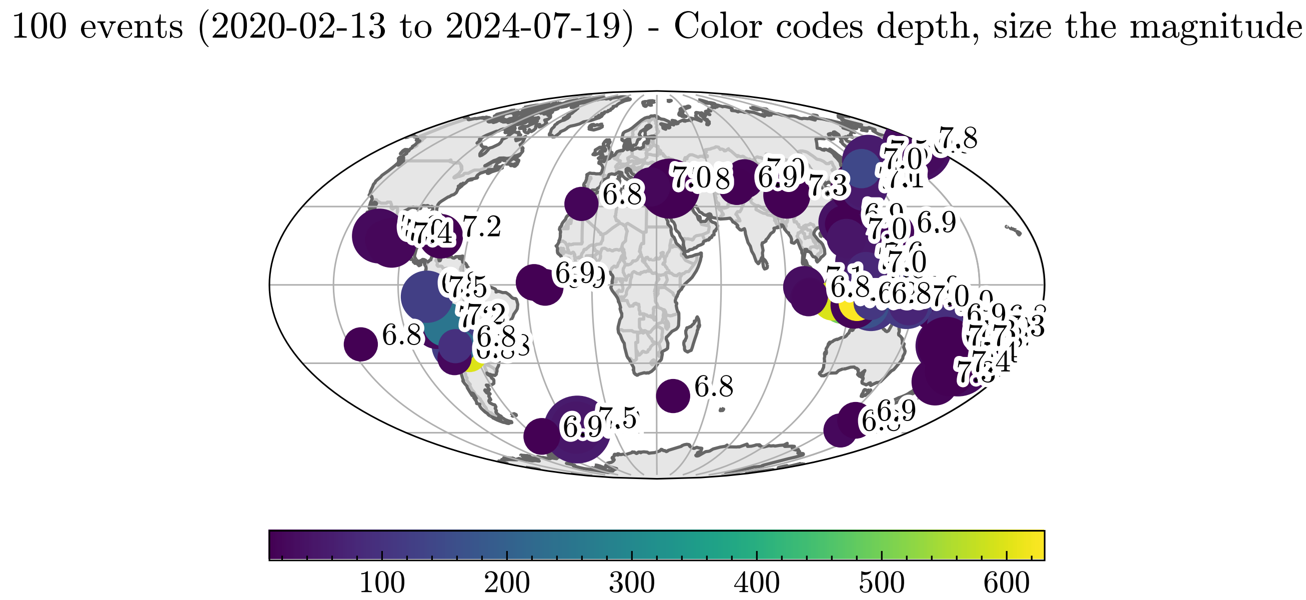

Figure 1: Geographical distribution of the 100 seismic events.

This is the Python code that was run to generate the map and the set of 100 HDF data files: (You can, but *you don't need to* run this script yourself. You may also access the file listed below directly at `https://www.uio.no/studier/emner/matnat/ifi/IN3190/h24/weekly-tasks/oslo_seismic_station/run_data_preparation.py`.)

```
# Requires the obspy and obspyh5 packages
#
# Can be installed through Conda:
#
# conda config --add channels conda-forge
# conda create -n obsh5 numpy obspy h5py scienceplots cartopy
# conda activate obsh5
# pip install obspyh5

import obspy
from obspy.clients.fdsn import Client
from obspy import UTCDateTime
```

```python
import matplotlib.pyplot as plt
import scienceplots # For plot formatting

plt.style.use(['science', 'no-latex'])

# Define FDSN client for earthquake catalog loading
client = Client("IRIS")

# Define parameters for a significant earthquake search
min_magnitude = 6.8
t1 = UTCDateTime("2020-01-01T00:00:00")
t2 = UTCDateTime("2024-07-31T23:59:59")
Nevents = 100

# Load the catalog of events
event_catalog = client.get_events(starttime=t1, endtime=t2, minmagnitude=
    min_magnitude, limit=Nevents, orderby="time", includearrivals=True)

# Print event details
print(event_catalog.__str__(print_all=True))

event_catalog.plot(outfile='event_map.pdf')


# ---- Load data
#
#
data_client = Client("UIB-NORSAR") # A Norwegian data centre

# University of Oslo campus seismic station information
network = "NS"
station = "OSL"
location = "00"
channel = "HHZ"

# Loop over all events
for this_event in event_catalog:

    # Define time window around the event origin time
    origin_time = this_event.origins[0].time
    start_time = origin_time - 5 * 60  # 5 minutes before origin time
    end_time = origin_time + 2 * 3600  # 2 hours after origin time

    # Get waveform data, decimate, and put into a trace vector
    st = data_client.get_waveforms(network=network, station=station, location=
        location, channel=channel, starttime=start_time, endtime=end_time)
    st.decimate(2)

    tr = st[0]
    print('f_T = ' + str(tr.stats.sampling_rate) + ' s')

    # Detrend and frequency bandpass
    tr.detrend('linear')
    tr.filter('bandpass', freqmin=0.05, freqmax=8)

    # Trim start and end to remove prospective filtering artefacts
    trimlength = 5
    tr.trim(starttime=start_time+trimlength, endtime=end_time+trimlength)

    # Save to HDF format
    fn = this_event.origins[0].time.isoformat() + '.h5'
    print('Saving ' + fn)
```

```
tr.write(fn, 'H5')
```

Your tasks are to:

a) Download the data from one of these 100 events – you can choose yourself which one. The HDF formatted data are available in the following folder:
`https://www.uio.no/studier/emner/matnat/ifi/IN3190/h24/weekly-tasks/oslo_seismic_station/`
There is one file per event. Plot the full time series. Write n as the x axils label and $x(n)$ as your y axis label.

b) Choose some start index at a location you find interesting in the full timeseries. Plot 40000 of the $x(n)$ signal samples, starting from your chosen index.

c) Plot 40000 signal samples, but with $x(n)$ shifted by 1000 samples, in the same figure. That is: plot $x(n-10000)$ within your interval of interest.

The following Matlab code stub snippet might be useful when you elaborate your data reading and plotting. You are also free to use, e.g., Julia or Python if you prefer. (The file can be accessed here: `https://www.uio.no/studier/emner/matnat/ifi/IN3190/h24/weekly-tasks/oslo_seismic_station/read_into_matlab.m`)

```
% Filename for loading the HDF file
fn = 'XXXXXXXXXXX';     % Replace XXXXXXX by one of the data filenames, such as for
    example 2024-07-19T01:50:48.107000.h5

% Extract metadata info from the file
info = h5info(fn)
group_name = info.Groups.Name
ds_name = info.Groups.Datasets.Name

% Read time series data and put into a numerical variable
data = h5read(fn, [group_name '/' ds_name]);

% Plot the full time series in a separate figure
%
% XXXXXXX add code here


% Plot the selected samples of the time series
% and the selected samples but delayed
%
% XXXXXXX add code here
```