CSE220 Signals and Linear Systems
Online on Signal Basics

14 September 2024

Time: 30 min
Read the entire instruction carefully before starting to code.

1 Introduction

In this online, your task is to implement the following functions for **discrete** signals

1. `time_reverse_signal`
2. `odd_even_decomposition`

2 Representing Discrete Signal

We will represent a discrete signal as a numpy array. We will assume that all signals extend from $-\infty$ to ∞, but the numpy array will only contain the signal values from -8 to 8. Signal values outside this range is considered to be 0.
For example, the array $x = [0, 0, 0, 0, 0, 0, 0.5, 2, 1, 0.5, 1, 0, 0, 0, 0, 0, 0]$ represents the signal $x[n]$ in figure 1.

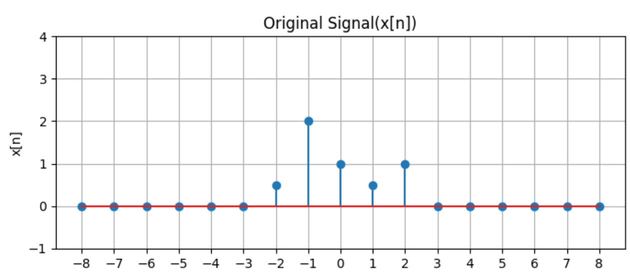

Figure 1: Signal corresponding to $[0, 0, 0, 0, 0, 0, 0.5, 2, 1, 0.5, 1, 0, 0, 0, 0, 0, 0]$

3 Tasks

You are **provided with a python file**. In this file, you have to implement 2 functions.

3.1 Task 1

Implement: `time_reverse_signal(x)`
Input Parameter:

- x: A numpy array representing a **discrete** signal.

The function should return a numpy array representing the time reversed signal `x[-n]`.
Example: see figure 2

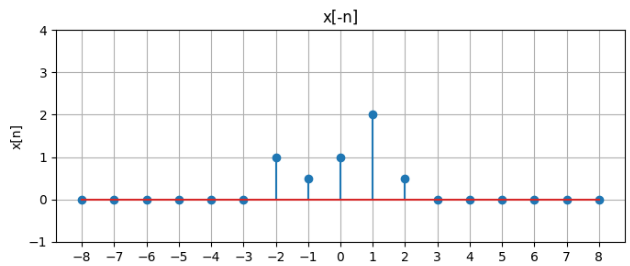

Figure 2: $x[-n]$

3.2 Task 2

Function to be implemented: `odd_even_decomposition(x)`
Input Parameters:

- x: A numpy array representing a **discrete** signal.

The function should return 2 numpy arrays. First array should be the odd component and second array should be the even component. Example: see figure 3 and 4.

3.3 Bonus Task

You will get bonus marks if you can complete the tasks using numpy functions rather than using explicit python loops.

4 Mark Distribution

See table 1.

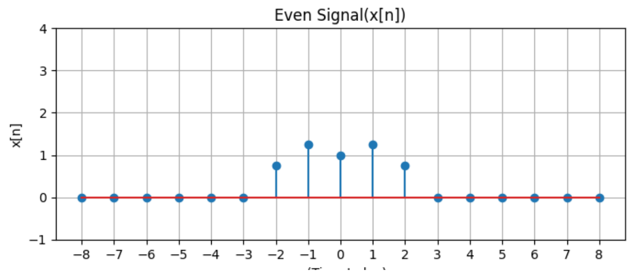

Figure 3: Even component of $x[n]$

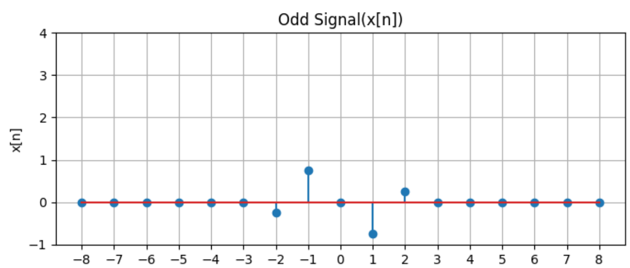

Figure 4: Odd component of $x[n]$

Task 1	6
Task 2	4
Bonus	2

Table 1: Mark Distribution