EE 242 Lab 2a – Convolution - Convolving Signals

Grace Hwang, Anna Petrbokova, Henry Adams, Leonard Paya

This short lab only has one exercise to be completed as a team. The exercise should be in its own code cell followed by a markdown cell with report discussion. Your notebook should start with a markdown title and overview cell, which should be followed by an import cell that has the import statements for all assignments. For this assignment, you will need to import: numpy, the wavfile package from scipy.io, and matplotlib.pyplot.

In [9]:
```
# We'll refer to this as the "import cell." Every module you import should be imported here.
%matplotlib inline
import numpy as np
import matplotlib
import scipy.signal as sig
import matplotlib.pyplot as plt
# import whatever other modules you use in this lab -- there are more that you need than we've in
```

Summary

In labs 2A and 2B, you will work through a series of exercises to introduce you to working with audio signals and explore the impact of convolution and smoothing on the sound of the signals. Lab 2A should be done in section.

Lab 2a turn in checklist

• Lab 2a Jupyter notebook with code for the first assignment. Each assignment cell should contain markdown cells (same as lab overview cells) for the responses to lab report questions. Include your lab members' names at the top of the notebook.

Please submit the report as PDF or html

Assignment 1 -- Convolving Simple Signals

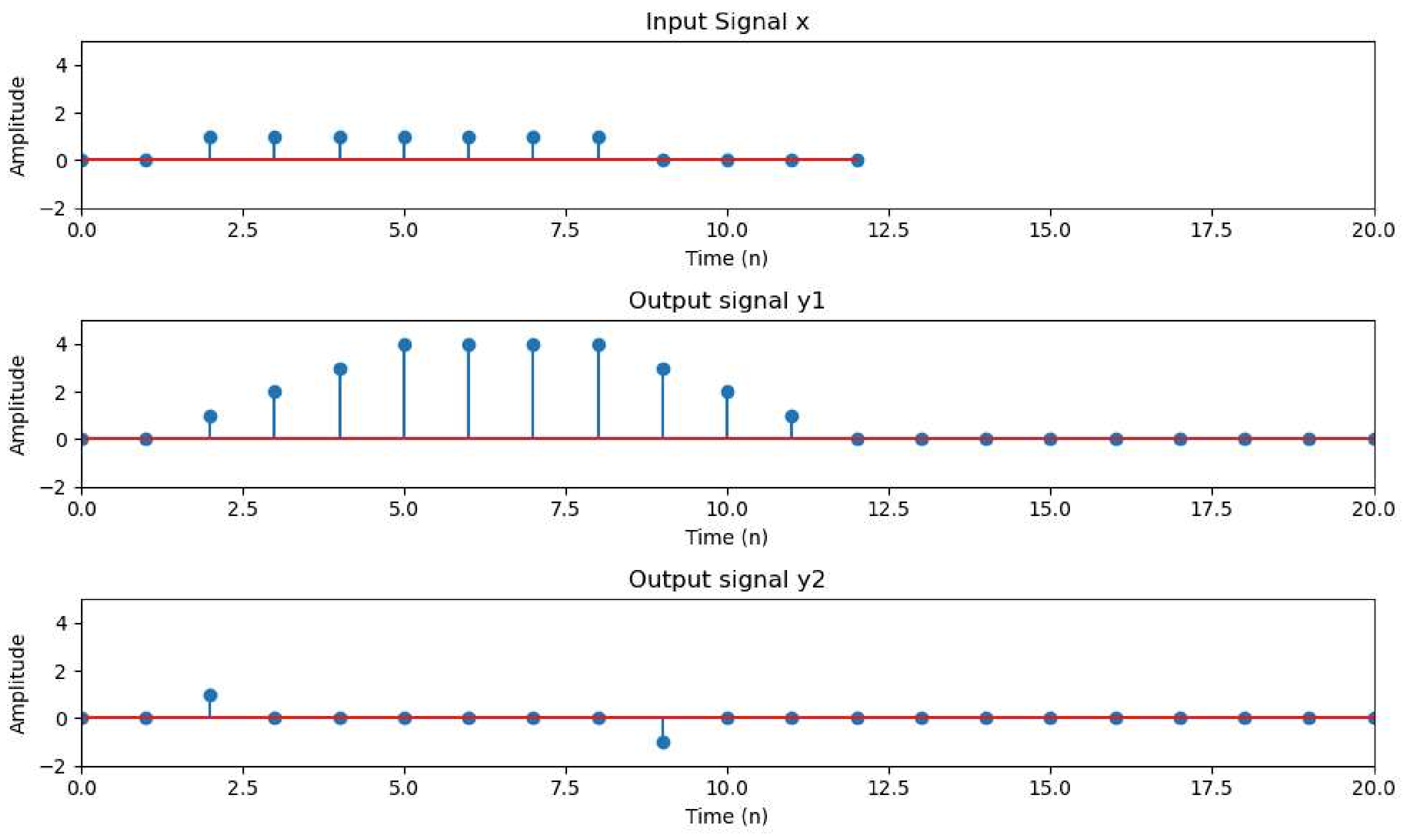
We will start by doing some simple convolutions similar to what you saw in class. There is a skeleton for Assignment 1 below, but you can create more cells as needed. This assignment will have three parts, A-C.

A. Create the following three discrete-time signals, assuming a time range of [0,12].
x = a box of height 1 starting at time t=2 and ending at t=8
h1 = a pulse of length 4 & height 1 starting at time t=0
h2 = 1 at first sample, -1 at at second sample, and 0 otherwise

B. Use numpy.convolve() function to find y1=$x * h1$ and y2=$x * h2$ (where $*$ indicates convolution, not multiplication) (See https://numpy.org/doc/stable/reference/generated/numpy.convolve.html). Notice the length of the output.

C. Create a time vectors nx, ny1 and ny2 for plotting x, y1 and y2. Use the stem plotting function to plot the signals on a 3x1 subplot, using a y-axis between -2 and 5 and a time axis from 0 to 20. Label and title the graphs. Verify that the signals for (y1) and (y2) match what you would expect.

In [10]:
```
# Assignment 1 - Convolving Simple Signals

# Part A

# Create three discrete-time signals, assuming a time range of [0,12]
# x: input signal vector, a box of height 1 starting at time n=2 and ending at n=8
# h1: a pulse of length 4 & height 1 starting at time 0
# h2 = 1 at n=0, -1 at n=1, and 0 otherwise
# TODO: Code that solves A

rangeTime = np.arange(13)
x = np.zeros(13)
x[2:9] = 1

h1 = np.zeros(13)
h1[:4] = 1

h2 = np.zeros(13)
h2[0] = 1
h2[1] = -1

# Part B
# Use the np.convolve() function to convolve the signals. Try plotting and observing the results
# Find the responses y1=x*h1 and y2=x*h2
# TODO: Code that solves B
y1 = np.convolve(x, h1)
y2 = np.convolve(x, h2)

# Part C
# Plot x, y1 and y2 in a 3x1 plot
# TODO: Code that solves C
#time for plotting

nx = np.arange(len(x))
ny1 = np.arange(len(y1))
ny2 = np.arange(len(y2))

plt.figure(figsize=(10, 6))
plt.subplot(3, 1, 1)
plt.stem(nx, x)
plt.title('Input Signal x')
plt.xlabel('Time (n)')
plt.ylabel('Amplitude')
plt.xlim(0, 20)
plt.ylim(-2, 5)

plt.subplot(3, 1, 2)
plt.stem(ny1, y1)
plt.title('Output signal y1')
plt.xlabel('Time (n)')
plt.ylabel('Amplitude')
plt.xlim(0, 20)
plt.ylim(-2, 5)

plt.subplot(3, 1, 3)
plt.stem(ny2, y2)
plt.title('Output signal y2')
plt.xlabel('Time (n)')
plt.ylabel('Amplitude')
plt.xlim(0, 20)
plt.ylim(-2, 5)

plt.tight_layout()
plt.show()
```



Discussion

The systems corresponding to impulse responses h_1[n] and h_2[n] capture different information about a signal. Comment on what aspects of the input signal correspond to the largest values of y_1[n] and y_2[n].

The largest values of y1[n] and y2[n], come from the sequence of non zero values that are next to each other, in other words the area under the function. The aspects of the input signal that corespond to the largest values are the number of non-zero simmilar values that are next to each other in a sequence, the more there is, the values more overlap there can be with the system.