CENG 384 - Signals and Systems for Computer Engineers
Spring 2023
Homework 1

İşleyici, Osman Taylan
e2448496@ceng.metu.edu.tr

LastName2, FirstName2
exxxxxxx@ceng.metu.edu.tr

March 30, 2023

---

1. (a)
   (b)
   (c)
   (d)
2.

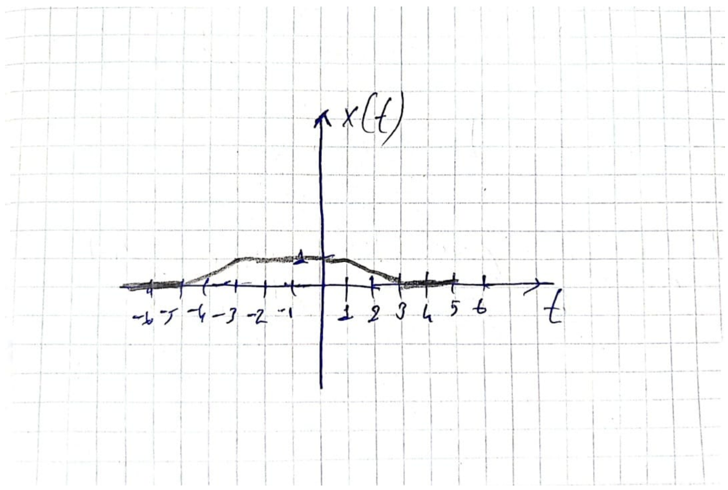

Figure 1: Signal $x(\frac{1}{2}t + 1)$

3. (a)
   (b)
4. (a) The signal is periodic and its fundamental period is $\frac{2\pi}{3}$ since the fundamental period of signal $k * cos(at + b)$ is determined by $\frac{2\pi}{a}$.
   (b) The fundamental period of discrete signal $mcos[an + b]$ is the smallest integer $\frac{2\pi k}{a}$ where k is any integer. Here the period of cosine part is 20, and the period of the sine part is 20 too. So the period of signal itself is least common multiple of these two periods, 20. The signal is periodic.
   (c) Since there isn't any integer k which makes $\frac{2\pi k}{7}$ an integer. The signal is not periodic.
5. (a)
   (b)
6. (a)
   - It has a memory since output depends on input $x(2t + 3)$.
   - It's not stable. We can prove this by counterexample, if $x(t) = 1$, $y(t) = t$ which is an unbounded output. So bounded inputs (1) yields unbounded outputs.
   - It's not causal since the output depends on inputs greater than t if $t > -3$.
   - It's linear since for two input signals x_1 and x_2 and their corresponding outputs y_1 and y_2; $ay_1(t) + by_2(t) = atx_1(2t + 3) + btx_2(2t + 3)$ is equal to $y_3(t) = tx_3(2t + 3)$ where x_3 is $ax_1 + b_x2$.
   - It's invertible since we can write a system $w(t) = \frac{2y(\frac{t-3}{2})}{t-3}$ whose output for the signal $y(t)$ is $x(t)$.

- The system is time varying because of two reasons, there is a time scaling on input $(2t+3)$, and there is also a time multiplier on signal. Both these reasons are enough for time variance.

(b)
- The system has a memory since the output depends not only on $x[n]$ but infinitely many inputs before n.
- The system is not stable, we can prove this by counterexample let's say out input is $x[n] = 1$ for every n. The output of the system would be $y[n] = \infty$ for every n. Since the output is unbounded while the input is bounded, system is not stable.
- It's linear, for two inputs $x_1[n]$ and $x_2[n]$ the response of system to $ax_1[n]+bx_2[n] = \sum_{k=1}^{\infty}(ax_1[n-k]+bx_2[n-k])$ is equal to $ay_1[n] + by_2[n]$.
- It's invertible, the output of system $w[n] = y[n+1] - y[n]$ for the input signal $y[n]$ is equal to $x[n]$.
- It's time invariant since for input $x'[n] = x[n-n_0]$ the response of system $y'[n]$ is equal to $\sum_{k=1}^{\infty} x'[n-k] = \sum_{k=1}^{\infty} x[n-n_0-k] = y[n-n_0]$.

7.
```
from brg.datastructures import Mesh

mesh = Mesh.from_obj('faces.obj')
mesh.draw()
```

(a)

(b)

Listing 1: Solution code of part b

```python
import matplotlib.pyplot as plt


def main():
    filePath = input() #The program takes the path of csv file from the user
    fileCSV = open(filePath)
    csvList = [i for i in fileCSV.read().split(",")]
    startingIndex = int(csvList[0])
    a = int(csvList[1])
    b = int(csvList[2])
    csvList = [float(i) for i in csvList[3:]]
    xVals = []
    yVals = []

    for iindex, i in enumerate(csvList):

        """In the following lines the code basically controls whether the current
        element  should be represented after the shift and scale operations. If it
        should, it finds the correct place for the element on the graph."""

        if (iindex + startingIndex - b)%a == 0: #If an + b = currentindex for some n
            xVals.append((iindex + startingIndex - b)//a) #Find n, add n to the x axis.
            yVals.append(i) #Add i to the y axis

    plt.stem(xVals, yVals)
    plt.show()


if __name__ == "__main__":
    main()
```

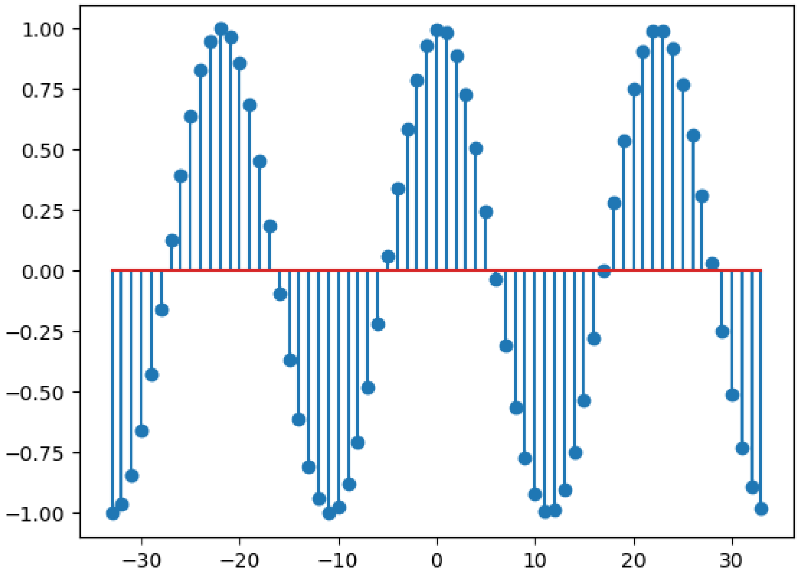

Figure 2: Output graphic for sine_part_b.csv

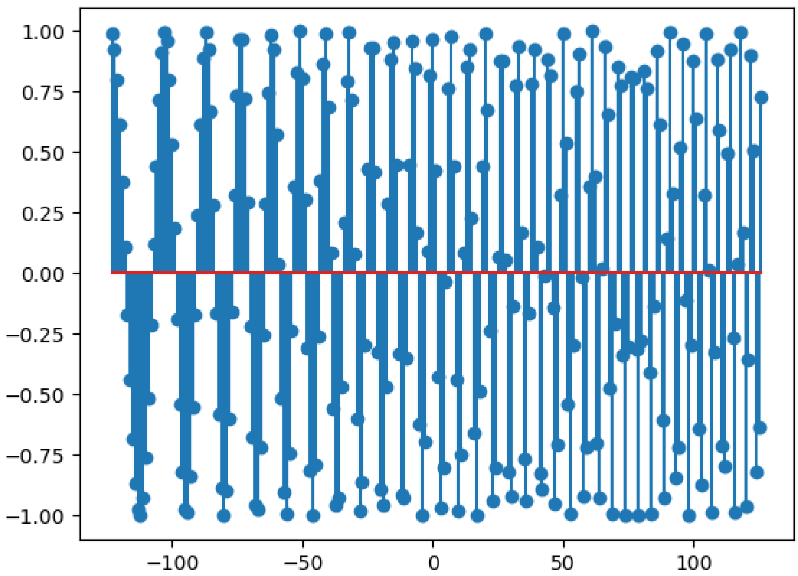

Figure 3: Output graphic for chirp_part_b.csv

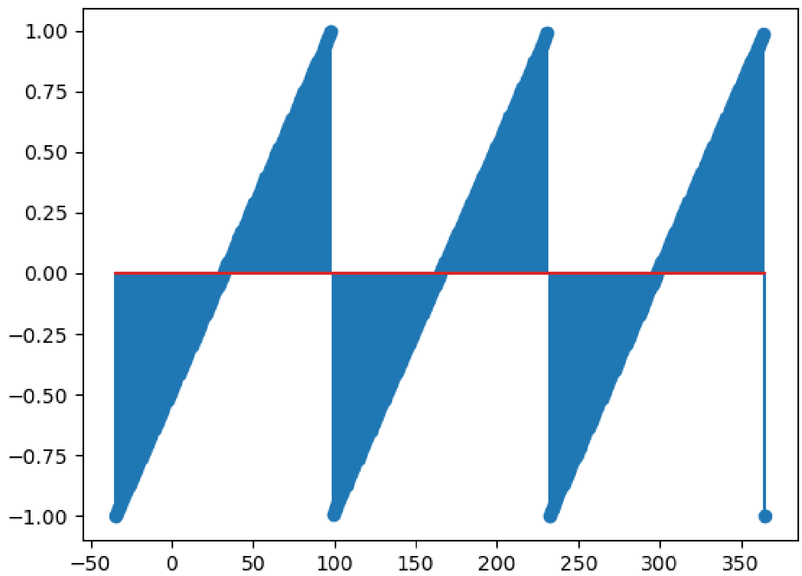

Figure 4: Output graphic for shifted_sawtooth_part_b.png