BLG354E
Homework-1

03.04.2023
Res. Asst. Yusuf Huseyin Sahin
sahinyu@itu.edu.tr

- You should write all your code in Python language.
- For the mathematical questions, it is not necessary to use LaTeX etc. You can clearly write your answers on papers and scan them.
- Cheating is highly discouraged. If you are planning to use different libraries or functions, please ask me about it.

1 (20 pts) - Analysing Signals

Using **matplotlib** and **numpy** libraries,
A) Plot the following continuous and discrete signals along t=-100 ... 100 or n=-100 ... 100.

- $x[n] = 5e^{j(\frac{7\pi n}{8}+\frac{\pi}{3})}$
- $x[n] = (0.2n)u[n+2] - (0.2n)u[2-n] + u[5]$
- $x[n] = \sum_{m=0}^{\infty} \delta[n-m]$
- $x(t) = \begin{cases} t+1, & -1 \leq t \leq 0 \\ 1, & 0 \leq t \leq 2 \\ -t+3, & 2 \leq t \leq 3 \\ 0, & otherwise \end{cases}$
- $x(t) = 5sin(2\pi t)cos(\pi t - 8)$

B) For each continuous signal given in A, plot the outputs for

$$y(t) = tx(t/2) \tag{0.1}$$

Consider unknown parts of the signals as zero.
C) For each discrete signal given in A, plot the outputs for

$$y[n] = \sum_{m=-\infty}^{n} x[m] \tag{0.2}$$

Consider unknown parts of the signals as zero.

2 (30 pts) - Part 2: System Properties - I

For the systems given below, determine whether each system is

- memoryless
- invertible
- causal
- stable.

A) $y(t) = t^2 x(t+10)$
B) $y(t) = 7x^2(t) + 5x(t) + 3$
C) $y[n] = \sum_{m=-\infty}^{n} x[m]$
D) $y[n] = 0.5x[6n-2] + 0.5x[6n+2]$

3 (30 pts) - Part 3: System Properties - II

For the system

$$y[n] = a_0 x[n] + a_1 x[n-1] + a_2 x[n-2] + a_3 x[n-3] \tag{0.3}$$

A) Draw a block didagram for the system.
B) Show whether the system is BIBO stable for all a_0, a_1, a_2, a_3.
C) Solve **A** and **B** for $y[n] = a_0 x[n] + a_1 x[n-1] + a_2 x[n-2] + a_3 x[n-3] + a_4 x[n-4] + ... = \sum_{i=0}^{\infty} a_i x[n-i]$.

4 (20 pts) - Part 4: SignalTime Manipulation

For the continuous system

$$x_1(t) = u(t+4) - u(t-4) + u(t+3) - u(t-3) + u(t+1) - u(t-1) \tag{0.4}$$

and its discrete variant

$$x_2[n] = u[n+4] - u[n-4] + u[n+3] - u[n-3] + u[n+1] - u[n-1] \tag{0.5}$$

Plot the following graphs in Python.

- $y(t) = x_1(t/2) + x_1(2t)$
- $y(t) = \sum_{k=1}^{20} x_1(t/k)$
- $y[n] = x_2[n/2] + x_1[2t]$
- $y[n] = \sum_{k=1}^{20} x_2[n/k]$

5 (10 pts) - Bonus Part: Listen To Your Heart, Closely

Work of the eyes is done, now
go and do heart-work
on all the images imprisoned within you; for you

Turning Point (Wendung), Rainer Maria Rilke

In this part, we will work on a maze game. A screenshot from the game is given in Figure 2.

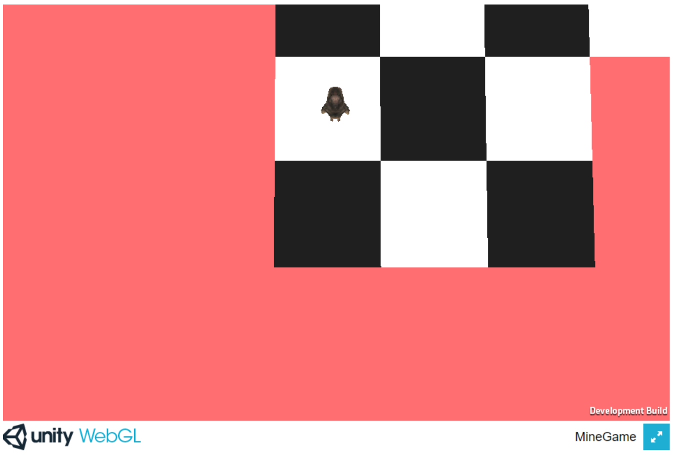

Figure 1: A screen from the game

In this game, we should move a monster along an L-shaped platform. However, some tiles on this platform are mined. To reach our goal, we should listen to the monster's heart. The monster's heartbeat is shaped as a triangle wave if it is in a safe spot. The heartbeat changes into a square wave if it is close to a mined tile. To be specific, being close to a mined tile by the 1/6 of the tile distance is necessary to change the beat. Example heartbeat signals are given in Figure 2. These signals are hard to hear. However, by recording them using Python, you can examine them.

The first library we will benefit in this part is **soundcard**. Using soundcard library, we can capture the computer's system audio. Thus, we can decide on the monster's path according to the captured signal. An example usage of the library is given below.

```python
import soundcard as sc

mics = sc.all_microphones(include_loopback=True)
default_mic = mics[1]    #this value could be changed according to your default audio output.

```

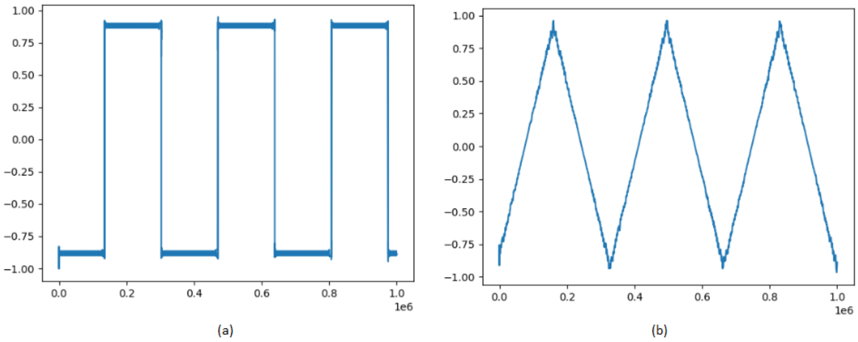

Figure 2: Heartbeats of the monster. (a) Safe, (b) In danger.

```
with default_mic.recorder(samplerate=148000) as mic:
    data = mic.record(numframes=1000000)
```

Another useful library for this homework is **pyautogui**, which is used to simulate mouse and keyboard interactions with Python. The example script given below clicks random keyboard buttons.

```
import pyautogui

pyautogui.keyDown('shift')

pyautogui.keyDown('w')
time.sleep(1)
pyautogui.keyUp('w')

pyautogui.keyUp('shift')

#pyautogui.keyUp and pyautogui.keyDown functions are used to simulate
    holding a button. For simple presses, pyautogui.press can be used.
```

You can visit my website[1] for the game. Write a Python script using pyautogui and soundcard libraries to make the monster reach the end of the platform. (It may fall down after winning the game, it's OK.)

[1] https://web.itu.edu.tr/sahinyu/mine_project_web/