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

Şamlıoğlu, Erdem
e2448843@ceng.metu.edu.tr

Uludoğan, Mert
e2380996@ceng.metu.edu.tr

April 2, 2023

---

1. (a)

$2z + 5 = j - \overline{z}$, substitution with $z = x + y \cdot j$
$2x + 2y \cdot j + 5 = j - x + y \cdot j$
$(2x + 5) + 2y \cdot j = -x + (y + 1) \cdot j$
$2x + 5 = -x$, $3x = -5$
$x = -\frac{5}{3}$
$2y = y + 1$
$y = 1$
so, $z = -\frac{5}{3} + j$

$|z| = \sqrt{x^2 + y^2}$, $|z|^2 = x^2 + y^2$

$|z|^2 = (-\frac{5}{3})^2 + 1^2 = \frac{34}{9}$

(b)

$z = r \cdot e^{j\theta}$
$z^5 = r^5 \cdot e^{5j\theta}$
$z^5 = r^5 \cdot e^{5j\theta}$
$r^5 \cdot e^{5j\theta} = 32j$
$r^5 \cdot \big(cos(5\theta) + j \cdot sin(5\theta)\big) = 32j$

$r^5 \cdot cos(5\theta) = 0$, $r^5 \cdot j \cdot sin(5\theta) = 32j$

$cos(5\theta) = 0$, $5\theta = \frac{\pi}{2}$

$r^5 \cdot j \cdot sin(5\theta) = 32j$

$r^5 \cdot sin(5\theta) \cdot j = 32j$, $sin(5\theta) = sin(\frac{\pi}{2}) = 1$

$r^5 = 32$, $r = 2$

$z = r \cdot e^{j\theta} = r \cdot (cos\theta + j \cdot sin\theta) = 2 \cdot (cos\frac{\pi}{10} + j \cdot sin\frac{\pi}{2})$

(c)

$$z = \frac{(1 + j) \cdot (\frac{1}{2} + \frac{\sqrt{3}}{2} \cdot j)}{(j - 1)}$$

multiplying both nominator and denominator with (1+j), $z = \dfrac{2j \cdot (\frac{1}{2} + \frac{\sqrt{3}}{2} \cdot j)}{-2}$

$z = -j \cdot (\frac{1}{2} + \frac{\sqrt{3}}{2} \cdot j) = -j \cdot \frac{1}{2} + (-\frac{\sqrt{3}}{2} \cdot j^2)$

$z = \frac{\sqrt{3}}{2} - \frac{1}{2} \cdot j$

Magnitude: $\sqrt{(\frac{\sqrt{3}}{2})^2 + (\frac{1}{2})^2} = 1$

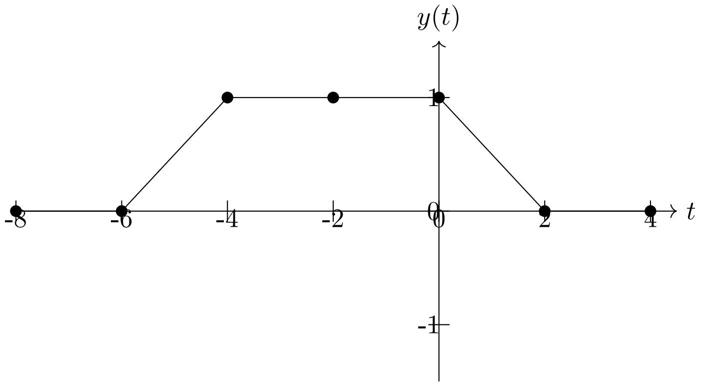
Angle: $cos\theta = \frac{\sqrt{3}}{2}$, $sin\theta = -\frac{1}{2}$, $\theta = -\frac{\pi}{6}$

(d)

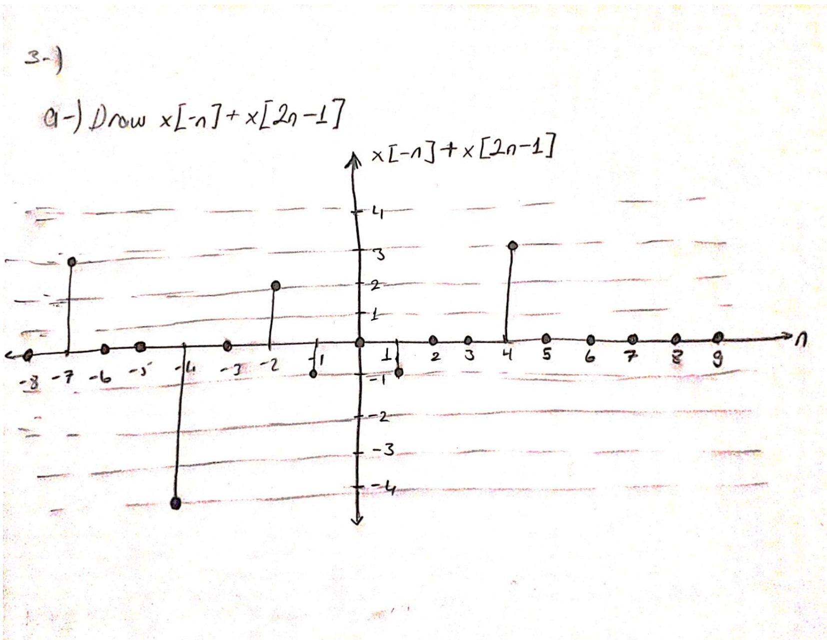
$z = j \cdot e^{-j\frac{\pi}{2}} = j \cdot (cos\frac{\pi}{2} - j \cdot sin\frac{\pi}{2})$

$z = j \cdot cos\frac{\pi}{2} - j^2 \cdot sin\frac{\pi}{2} = sin\frac{\pi}{2} + j \cdot cos\frac{\pi}{2}$

2. Here is the signal for $y(t) = x((1/2)t + 1)$



3. (a)



(b)

$x[-n] + x[2n-1] = 3 \cdot \delta(n+7) + 2 \cdot \delta(n+2) + (-1) \cdot \delta(n-1) + 3 \cdot \delta(n-4)$

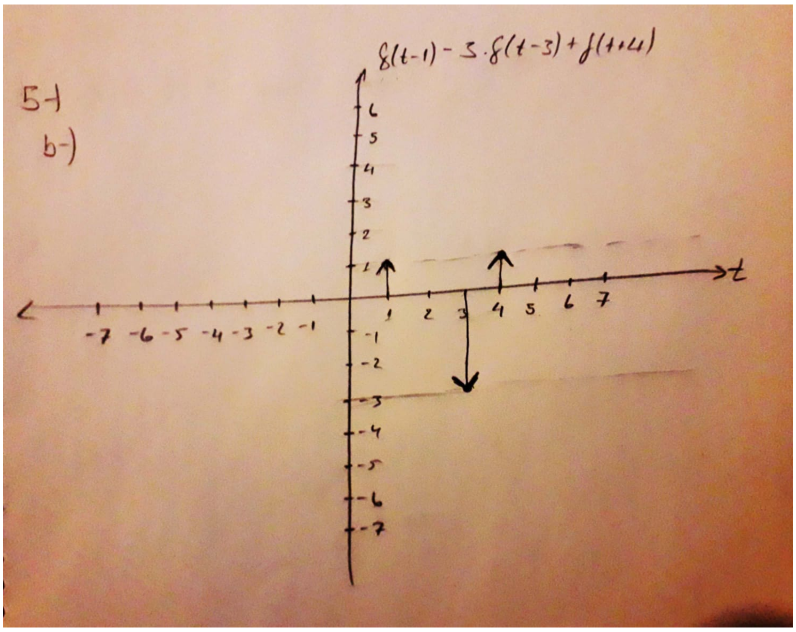
4. (a) $x(t) = 5sin(3t - \pi/4)$ is periodic. For continuous periodic signals, $x(t) = x(t+T)$.
For continuous signals, fundamental period can be find with $T_0 = 2\pi/\omega$. So with the formula, $T_0 = 2\pi/3$, which is the period of this continuous signal.

(b) $x[n] = cos[(13\pi/10)n] + sin[(7\pi/10)n]$ is periodic. For discrete periodic signals , x[n]=x[n+N] for N being an integer. For discrete signals, fundamental period can be find with $N_0 = 2\pi/\Omega$.
For the first term,$cos[(13\pi/10)n]$, fundamental period is $N_1 = (2\pi/\Omega)k = (2\pi/(13\pi/10))k = (20/13)k$. $k = 13$ and $N_1 = 20$.
For the second term, $sin[(7\pi/10)n]$, fundamental period is $N_2 = (2\pi/\Omega)k = (2\pi/(7\pi/10))k = (20/7)k$. $k = 7$ and $N_2 = 20$.
To find the fundamental period for the signal x[n], need less common multiple for both. $lcm(20, 20) = 20$. So the period of x[n] is 20.

(c) $x[n] = (1/2)cos[(7n - 5]$ is not periodic. For discrete periodic functions, x[n]=x[n+k] for integer k. For discrete signals, fundamental period can be find with $N_0 = 2\pi/\Omega$. For this signal $N_0 = (2\pi/\Omega)m = (2\pi/7)m$. There is no m to make it an integer. So x[n] is not periodic.

5. (a)
$x(t) = u(t-1) - 3 \cdot u(t-3) + u(t-4)$

(b)
$\frac{dx(t)}{dt} = \delta(t-1) - 3 \cdot \delta(t-3) + \delta(t-4)$



6. (a) 1)System has a memory since the present value of outputs depends on future values of inputs.
2)System is not stable. $y(t)$ depends on t, which is unbounded.
3)System is not causal, system output depends on the future input.
4)For linearity, $y(x_1 + x_2) = y(x_1) + y(x_2)$. We can say that $tx_1(2t+3) + tx_2(2t+3) = t(x_1+x_2)(2t+3)$. So system is linear.
5)System is not invertible. We can say $x((t-3)/2) = (2/(t-3))y((t-3)/2)$ Which makes this system for t=3, not invertible.
6)For time invariance $y(t-t_0) = (t-t_0)x(2t+3-2t_0) \neq tx(2t+3-2t_0)$. So system is not time invariant.

(b) 1)System has a memory since the present value of output depends on past values of inputs(sum of them).
2)System is not stable. Since it's output is summation of inputs, y[n] is unbounded, can go to infinity
3)System is causal. It does not depend on future inputs and depends on past inputs.
4)For linearity $y(x_1 + x_2) = y(x_1) + y(x_2)$. In this system, $y_1[n] = \sum_{k=1}^{\infty} x_1[n-k]$, $y_2[n] = \sum_{k=1}^{\infty} x_2[n-k]$. $y_3[n] = a_1 y_1[n] + a_2 y_2[n]$ for some constants a_1 and a_2.
Also for $x_3[n] = a_1 x_1[n] + a_2 x_2[n]$. Now comparing what we have found now and before, $y_{3.1} = \sum_{k=1}^{\infty} x_2[n-k] = a_1 \sum_{k=1}^{\infty} x_1[n-k] + a_2 \sum_{k=1}^{\infty} x_2[n-k]$. So this system is linear.
5)System is invertible. $x[n] = y[n+1] - y[n]$ The right side of the equation is equal to $(x[n] + x[n-1] + x[n-2] + ...) - (x[n-1] + x[n-2] + x[n-3] + ...)$.
6)For time invariance, we can check: $y[n] = x_1[n] = x[n-n_0]$. So we can say $y[n] = \sum_{k=1}^{\infty} x[n-n_0-k]$. Again $y_1[n] = y[n-n_0]$ which is equal to $\sum_{k=1}^{\infty} x[n-n_0-k]$. They are equal, which makes this system time invariant.

7. (a) The graphs are listed:

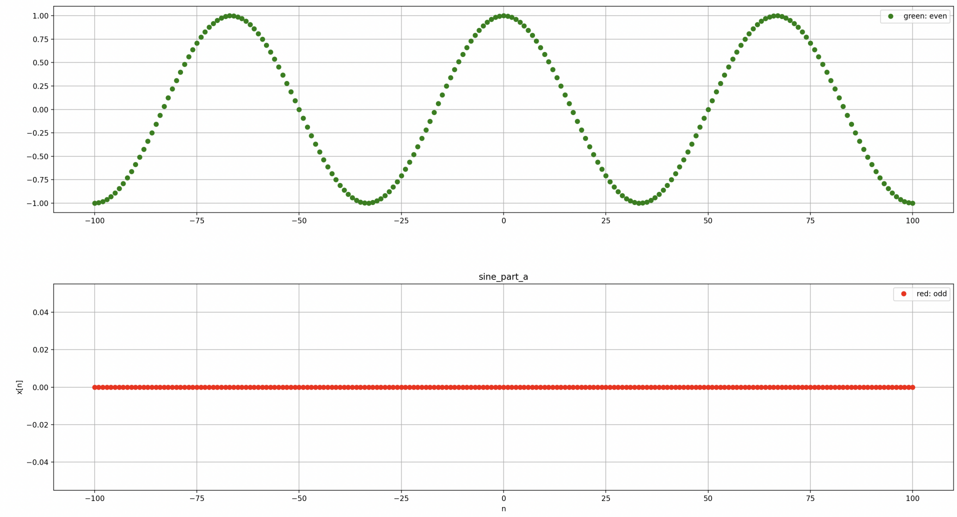


Figure 1: sine

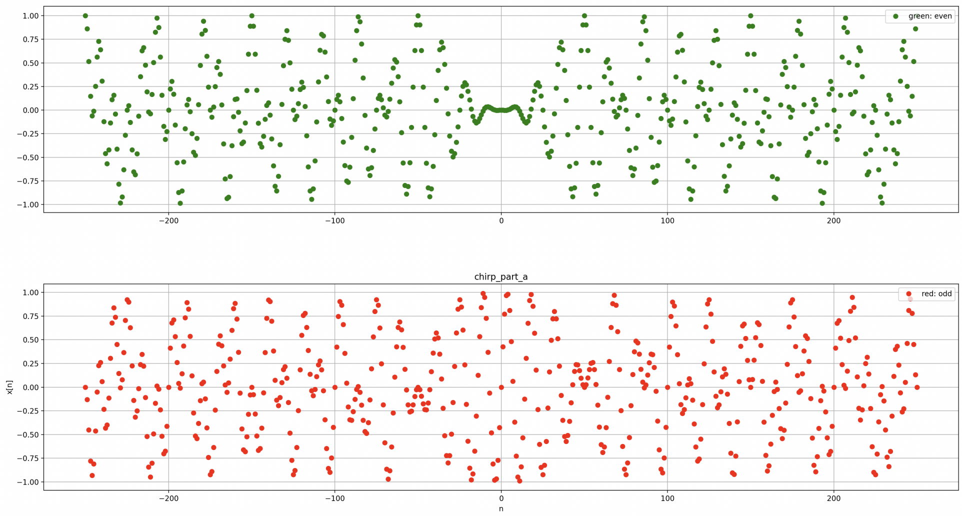


Figure 2: chirp

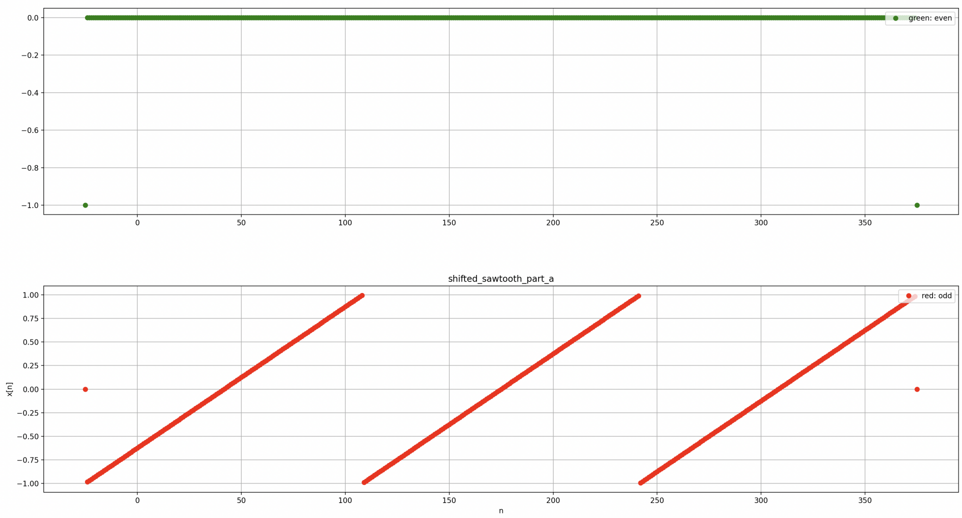


Figure 3: shifted

```python
import matplotlib.pyplot as plt

files_a = ("sine_part_a.csv", "shifted_sawtooth_part_a.csv", "chirp_part_a.csv")

def calc_a(file):
    with open(file, 'r') as csv_file:
        x = csv_file.readline().strip()
        _x = x.split(',')
        data = [float(x) for x in _x]
        almost_data = list(map(float, data))
        starting_index = int(almost_data[0])
        data = almost_data[1:]
        reverse_data = data[::-1]

        even = [(x + y) / 2 for x, y in zip(data, reverse_data)]
        odd = [(x - y) / 2 for x, y in zip(data, reverse_data)]

        x_values = list(range(starting_index,starting_index + len(data)))

        fig, axs = plt.subplots(2,1)

        axs[0].plot(
            x_values,
            even,
            'go'
        )
        axs[0].legend(["green: even"], loc='upper right')
        axs[1].plot(
            x_values,
            odd,
            'ro'
        )
        axs[1].legend(["red: odd"], loc='upper right')


        plt.xlabel("n")
        plt.ylabel("x[n]")

        #plt.xticks(np.arange(starting_index+4, starting_index + len(data)+4, 1))

        title = plt.title(file[:-4])
        plt.tight_layout()
        axs[0].grid(True)
        axs[1].grid(True)

        plt.show()

for file in files_a:
    result = calc_a(file)
```

(b) The graphs are as listed:

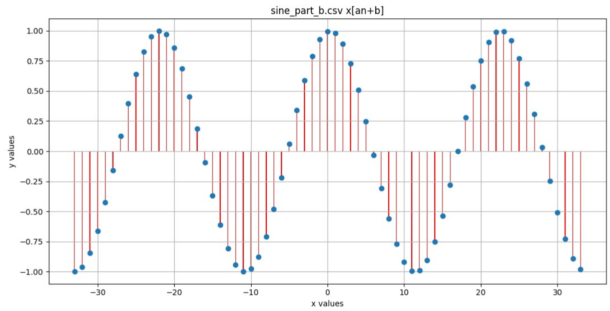


Figure 4: sine

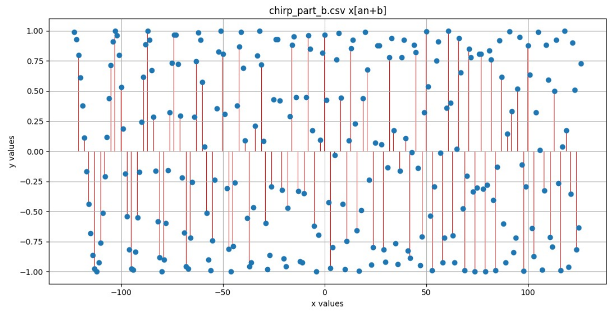


Figure 5: chirp

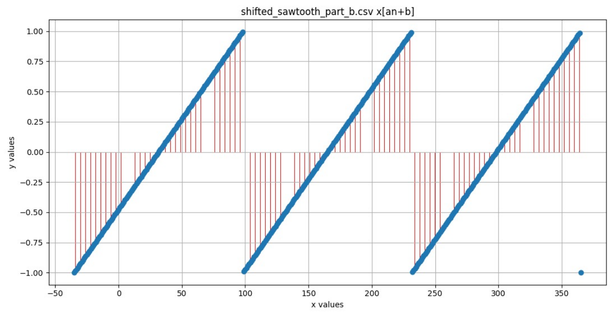


Figure 6: shifted

```
 from matplotlib import pyplot as plt

inputFiles = ["sine_part_b.csv", "shifted_sawtooth_part_b.csv",
"chirp_part_b.csv"]
funcDict = {}
checkAfterDict = {}

def store_in_dict(x_vals, y_vals, dict):
    for i in range(len(x_vals)):
        dict[x_vals[i]] = y_vals[i]
    #print("DICT", dict)

def return_new_y_array(new_x_vals,a,b):
    new_y_vals = []
    for el in new_x_vals:
        old_n = a*el + b
        y_val = funcDict[old_n]
        new_y_vals.append(y_val)
    return new_y_vals

for file in inputFiles:
```

```
with open(file, 'r') as f:
    for line in f:
        cells = line.split(',')
        floatCells = [float(i) for i in cells]
        start = int(floatCells[0])
        a = floatCells[1]
        b = floatCells[2]
        original_y_vals = floatCells[3:]
    #print(len(original_y_vals))

    original_x_vals = [i for i in range(start, start+ len(original_y_vals))]
    store_in_dict(original_x_vals, original_y_vals, funcDict)
    new_x_vals_temp = [(i-b)/a for i in original_x_vals]
    new_x_vals = [i for i in new_x_vals_temp if i - int(i) == 0]
    #print("new_x_vals_temp", len(new_x_vals_temp), new_x_vals_temp)
    #print("new_x_vals", len(new_x_vals), new_x_vals)
    new_y_vals = return_new_y_array(new_x_vals, a, b)
    #store_in_dict(new_x_vals, new_y_vals, checkAfterDict)

    plt.plot(new_x_vals, new_y_vals, 'o')
    plt.bar(new_x_vals, new_y_vals, color ='red', width = 0.1)
    plt.xlabel('x values')
    plt.ylabel('y values')
    plt.title(file + ' x[an+b]')
    plt.grid(True)
    plt.show()
```