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

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

Deveci, Cengizhan
e2448322@ceng.metu.edu.tr

April 2, 2023

---

1. (a) If z = x + yj then $\overline{z}$ = x - yj.

When we put the this equation to second equation we get $2(x+yj)+5 = j-(x-yj) => 2x+2yj+5 = j-x+yj$.

$=> 3x + yj = -5 + j$

Therefore we will get 3x = -5 so x = -5/3. And y = 1.

So z = -5/3 + j.

$|z|$ will become $\sqrt{(\frac{-5}{3})^2 + 1^2} = \sqrt{\frac{25}{9} + 1} = \frac{\sqrt{34}}{3}$.

So, $|z|^2 = \frac{34}{9}$.

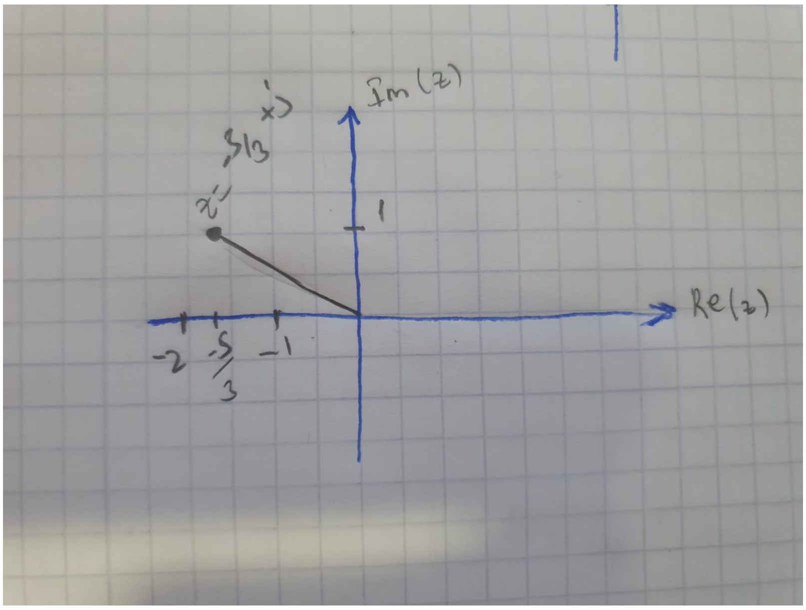


Figure 1: z on the complex plane

(b) $z = re^{j\theta}$ so $z^5 = (re^{j\theta})^5 = r^5 e^{5j\theta} = 32j$. and $j = e^{j\frac{\pi}{2}}$.

So, $r^5 e^{5j\theta} = 32e^{j\frac{\pi}{2}}$. Therefore $5\theta = \frac{\pi}{2} => \theta = \frac{\pi}{10}$ and $r^5 = 32 => r = 2$.

As a result, $z = 2e^{\frac{\pi}{10}j}$.

(c) Firstly, multiplying with the conjugation of j - 1 both side.

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

And magnitude of z is $|z| = \sqrt{(\frac{\sqrt{3}}{2})^2 + (\frac{-1}{2})^2} = \sqrt{\frac{3}{4} + \frac{1}{4}} = 1.$

And the angle of z is $2\pi - arctan(\frac{\frac{\sqrt{3}}{2}}{\frac{1}{2}}) = \frac{5\pi}{3}.$

(d) $j = e^{j\frac{\pi}{2}}$ and z $= je^{-j\frac{\pi}{2}}$. When we put the j to the equation, we get

$z = e^{j\frac{\pi}{2}}e^{-j\frac{\pi}{2}} = e^{j\frac{\pi}{2}-j\frac{pi}{2}} = e^0 = 1.$

z = 1 + 0j.

2.

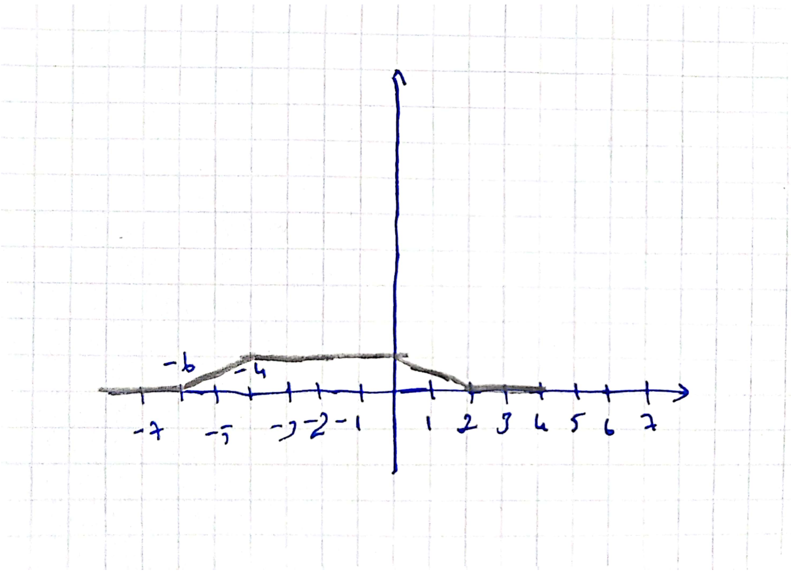
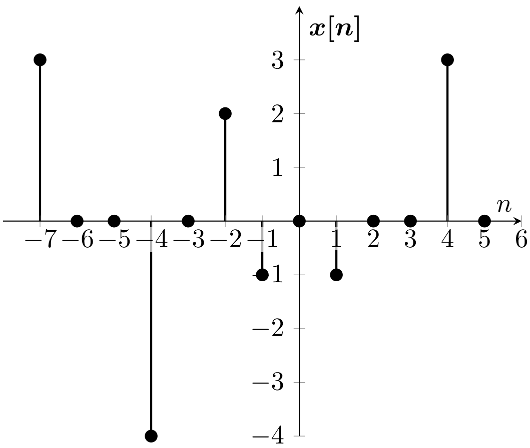


Figure 2: Signal $x(\frac{1}{2}t + 1)$

3. (a)



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

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 for k's 13 and 7. 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) $x(4) = u(t-1) - 3u(t-3) + u(t-4)$

(b)

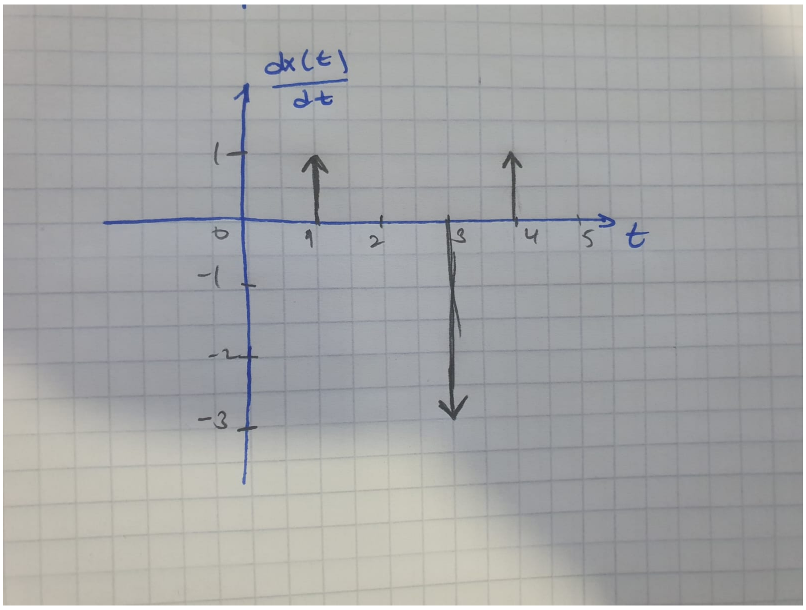


Figure 3: graph of $\frac{dx(t)}{dt}$

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. (a)

Listing 1: Solution code of part a

```
import matplotlib.pyplot as plt

def main():
    filePath = input()
    fileCSV = open(filePath)
    csvList = [i for i in fileCSV.read().split(",")]
    startingIndex = int(csvList[0])
    csvList = [float(i) for i in csvList[1:]]
    even_results = dict()
    odd_results = dict()

    for iindex, i in enumerate(csvList):
        if((iindex + startingIndex) not in even_results.keys()):
            even_results[iindex + startingIndex] = i
        else:
            even_results[iindex + startingIndex] += i

        if(-(iindex + startingIndex) not in even_results.keys()):
            even_results[-(iindex + startingIndex)] = i
        else:
            even_results[-(iindex + startingIndex)] += i
```

```
        if((iindex + startingIndex) not in odd_results.keys()):
            odd_results[iindex + startingIndex] = i
        else:
            odd_results[iindex + startingIndex] += i

        if(-(iindex + startingIndex) not in odd_results.keys()):
            odd_results[-(iindex + startingIndex)] = -i
        else:
            odd_results[-(iindex + startingIndex)] -= i

    xVals_even = []
    yVals_even = []
    xVals_odd = []
    yVals_odd = []

    for i in even_results.items():
        xVals_even.append(i[0])
        yVals_even.append(i[1]/2)

    for i in odd_results.items():
        xVals_odd.append(i[0])
        yVals_odd.append(i[1]/2)


    plt.stem(xVals_even, yVals_even)
    plt.show()

    plt.stem(xVals_odd, yVals_odd)
    plt.show()

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

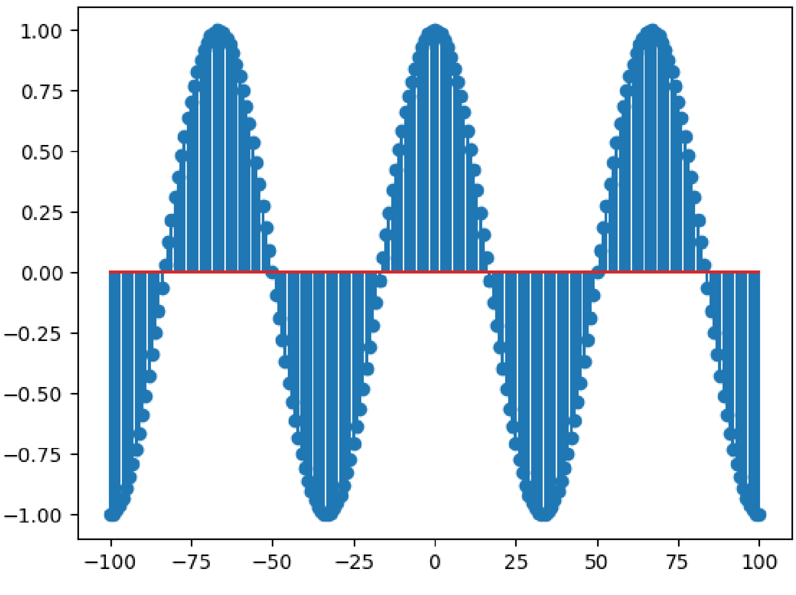


Figure 4: Even part for sine_part_a.csv

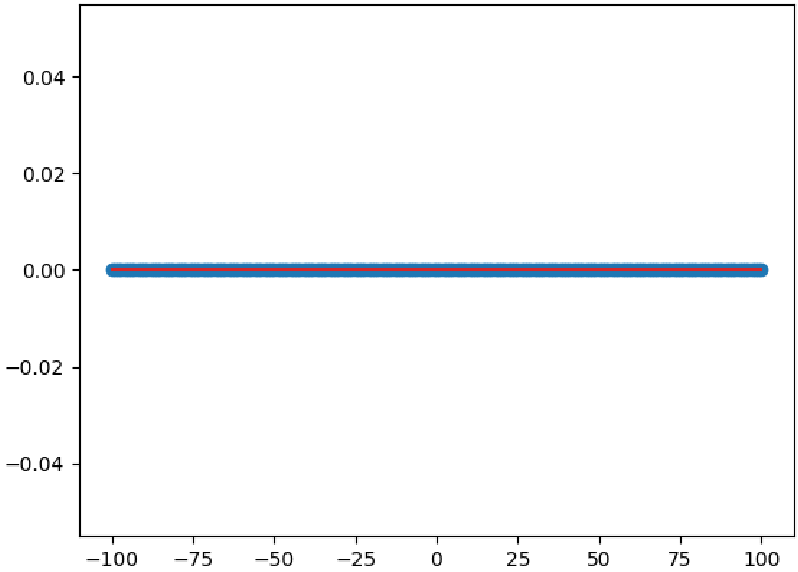


Figure 5: Odd part for sine_part_a.csv

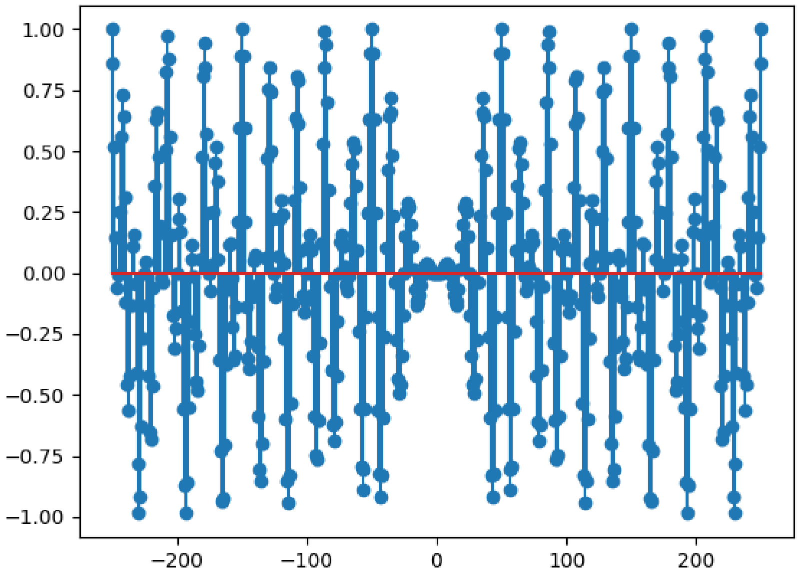


Figure 6: Even part for chirp_part_a.csv

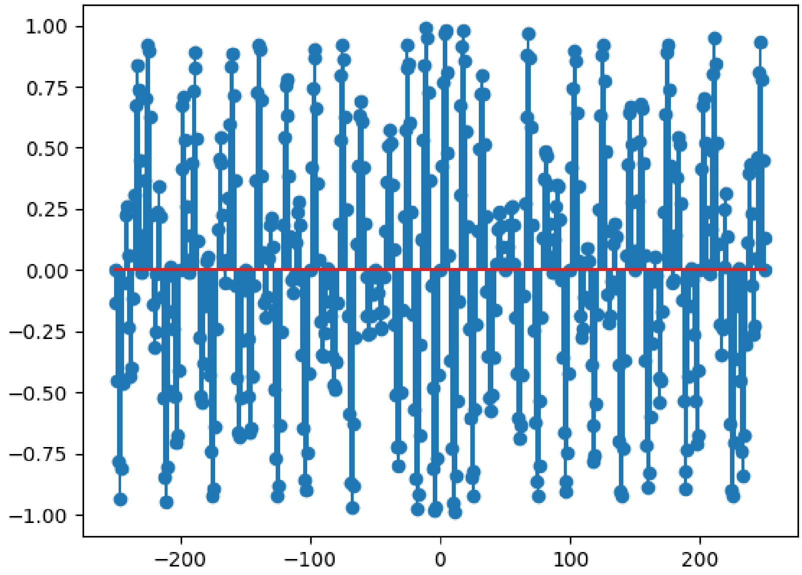


Figure 7: Odd part for chirp_part_a.csv

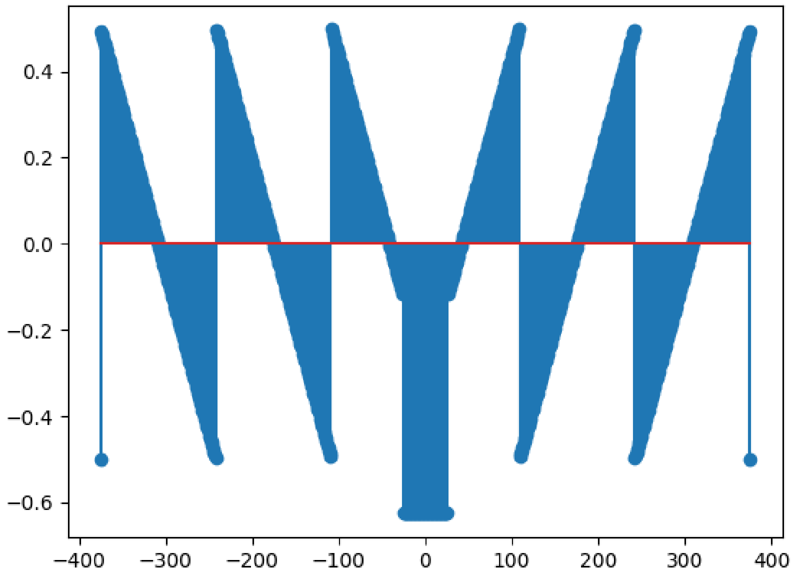


Figure 8: Even part for shifted_sawtooth_part_a.csv

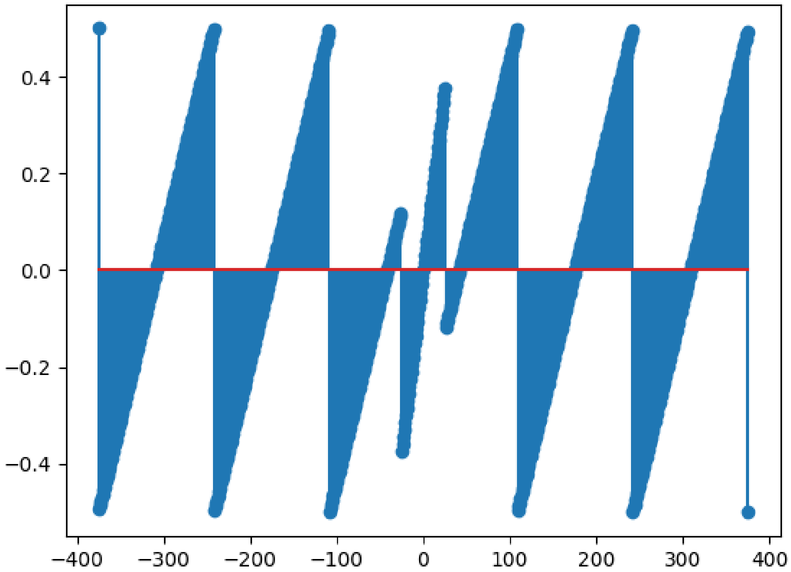


Figure 9: Odd part for shifted_sawtooth_part_a.csv

(b)

Listing 2: Solution code of part b

```
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()
```

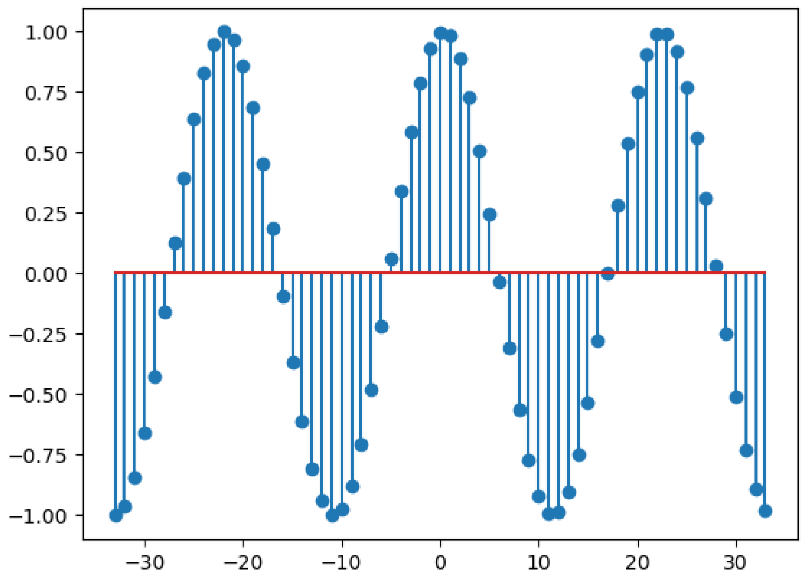


Figure 10: Output graphic for sine_part_b.csv

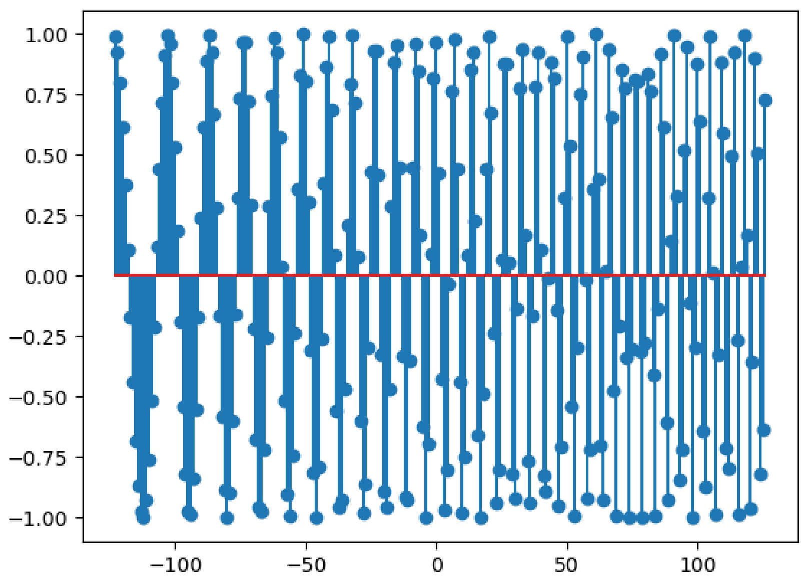


Figure 11: Output graphic for chirp_part_b.csv

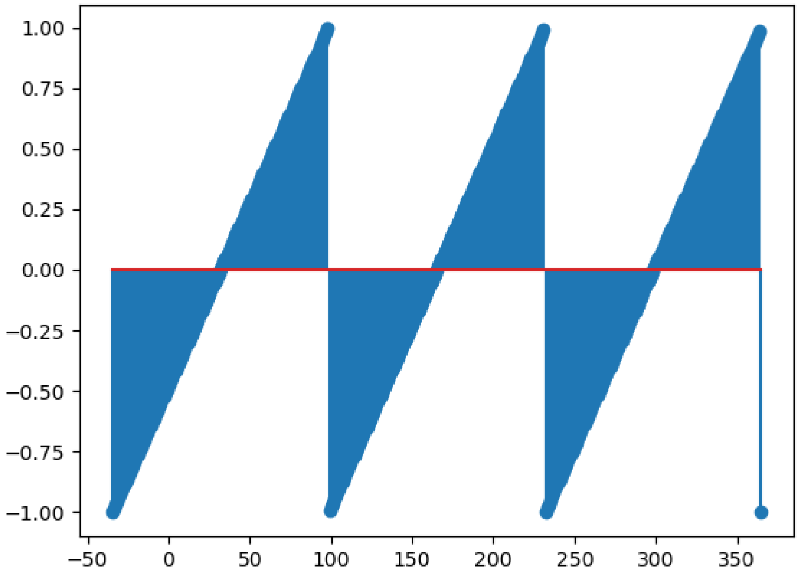


Figure 12: Output graphic for shifted_sawtooth_part_b.csv