Digital version

ENGS252 Signals and Systems - Homework 11

Mher Saribekyan A09210183

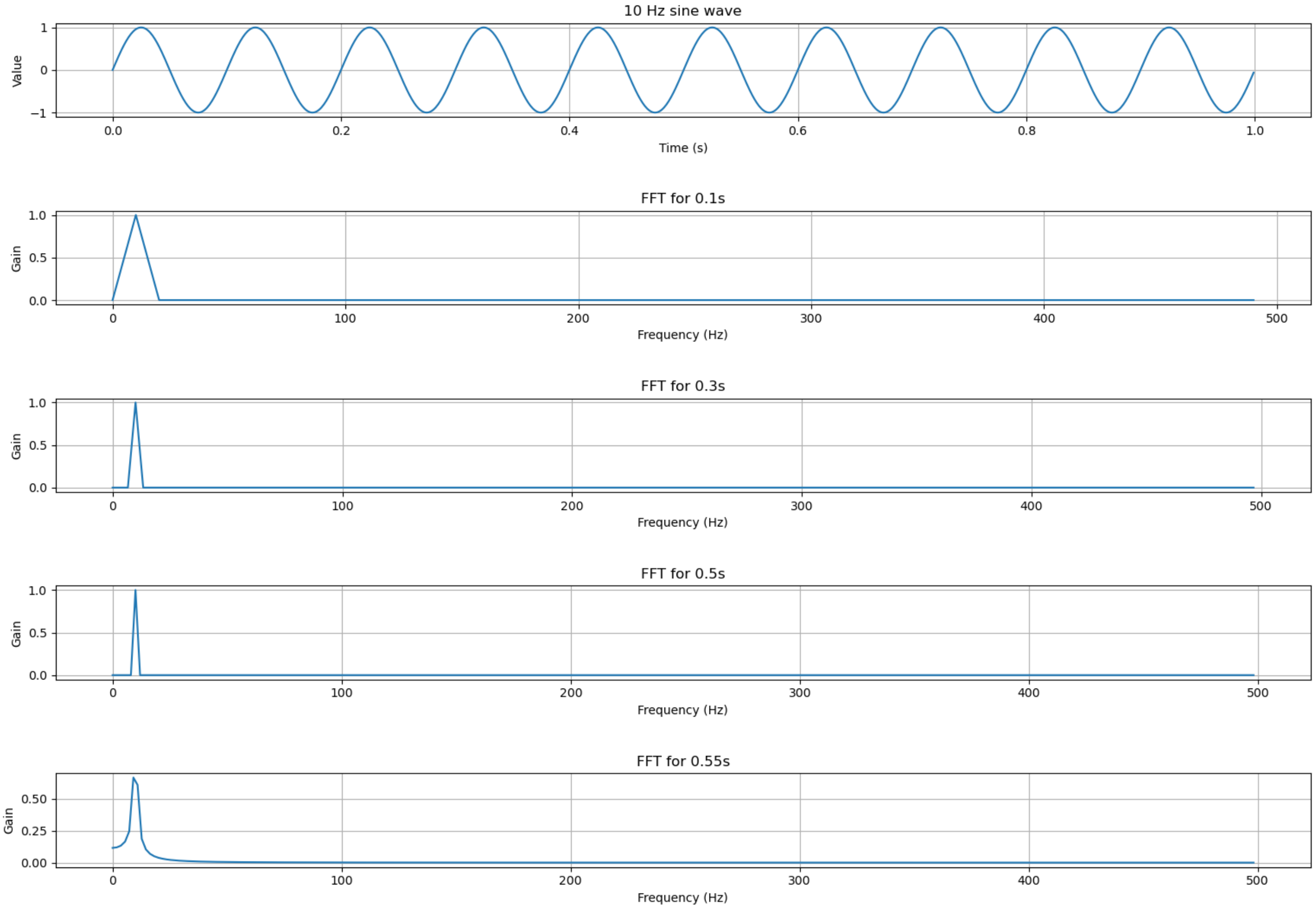
May 4, 2026

Problem 1

Generate a sinusoid signal with a sampling rate of 10^3 Hz and frequency 10 Hz. Compare the results of FFT of that signal with the duration of 0.1s=100ms; 0.3s=300ms; 0.5s=500ms, 0.55s=550ms. Pay attention to aliasing affects.

```
import numpy as np
import matplotlib.pyplot as plt

freq = 10 # change to 8 for b)
fs = 1000 # change to 32 for b)
ds = [0.1, 0.3, 0.5, 0.55]
ts = [np.arange(0, d, 1/fs) for d in ds]
ss = [np.sin(2 * np.pi * freq * t) for t in ts]

fig, axs = plt.subplots(len(ts) + 1, 1, figsize=(4, 8))

# Sine wave only for visualization
t0 = np.arange(0, 1, 1/fs)
s0 = np.sin(2 * np.pi * freq * t0)

axs[0].plot(t0, s0)
axs[0].set_title(f"{freq} Hz sine wave")
axs[0].set_xlabel("Time (s)")
axs[0].set_ylabel("Value")
axs[0].grid()

# FFT for each case
for i, t in enumerate(ts):
    fft_values = np.fft.fft(ss[i])
    N = len(ss[i])

    freq = np.fft.fftfreq(N, 1/fs)
    magnitude = 2 * np.abs(fft_values) / N

    axs[i + 1].plot(freq[:N//2], magnitude[:N//2])
    axs[i + 1].set_title(f"FFT for {ds[i]}s")
    axs[i + 1].set_xlabel("Frequency (Hz)")
    axs[i + 1].set_ylabel("Gain")
    axs[i + 1].grid()

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

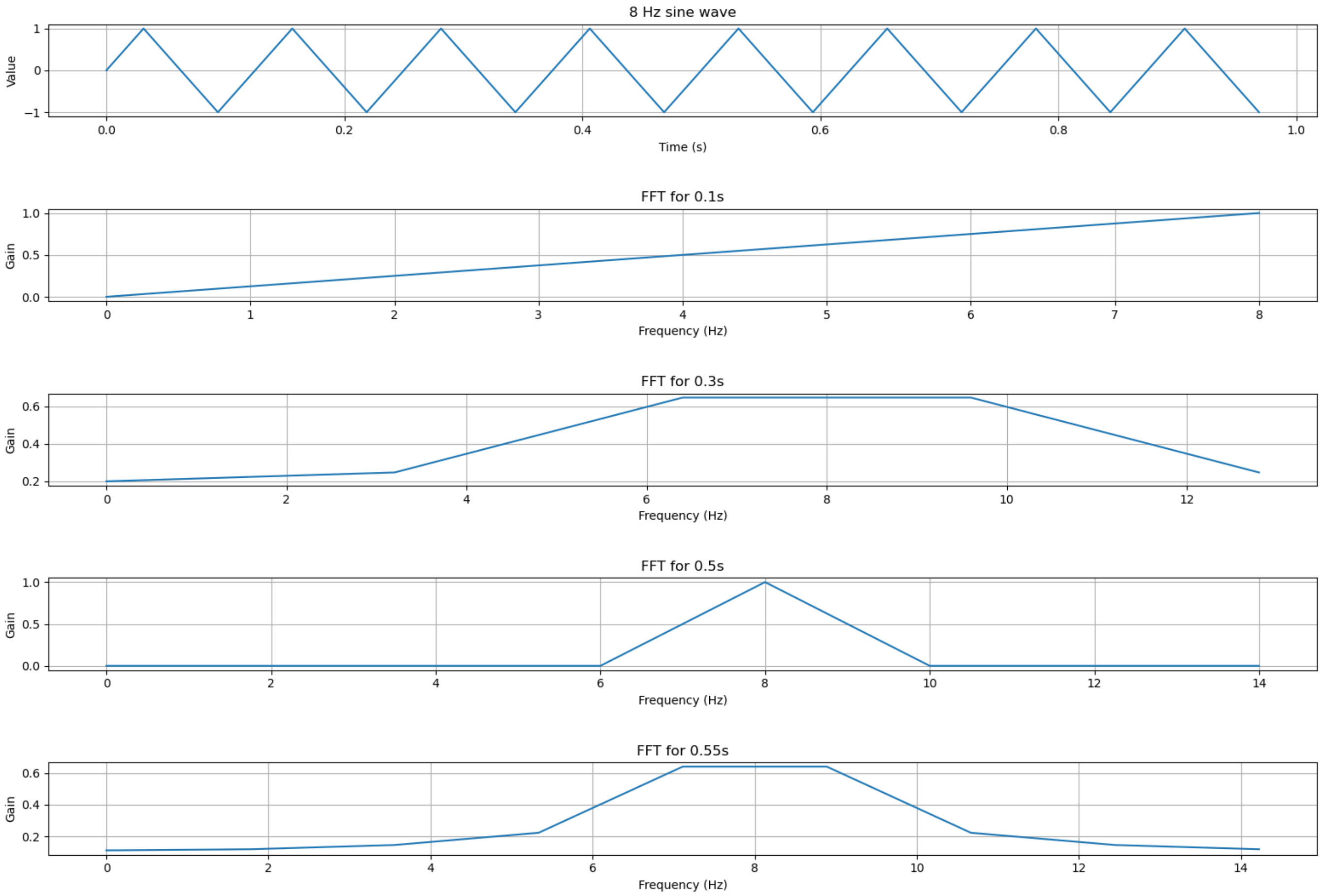
a) Give the frequency resolution in each case. Explain the difference in results.



$$\Delta f_1 = \frac{1}{0.1} = 10Hz,\ \Delta f_2 = \frac{1}{0.3} \approx 3.33Hz,\ \Delta f_3 = \frac{1}{0.5} = 2Hz,\ \Delta f_4 = \frac{1}{0.55} \approx 1.81Hz,$$

We see that 10 Hz is a multiple of the first three signal durations, while for 0.55s, it is not, hence we get good results for the first three cases, and aliasing for the 0.55s case.

b) How the result will be changed if the sampling rate would be 8 Hz?

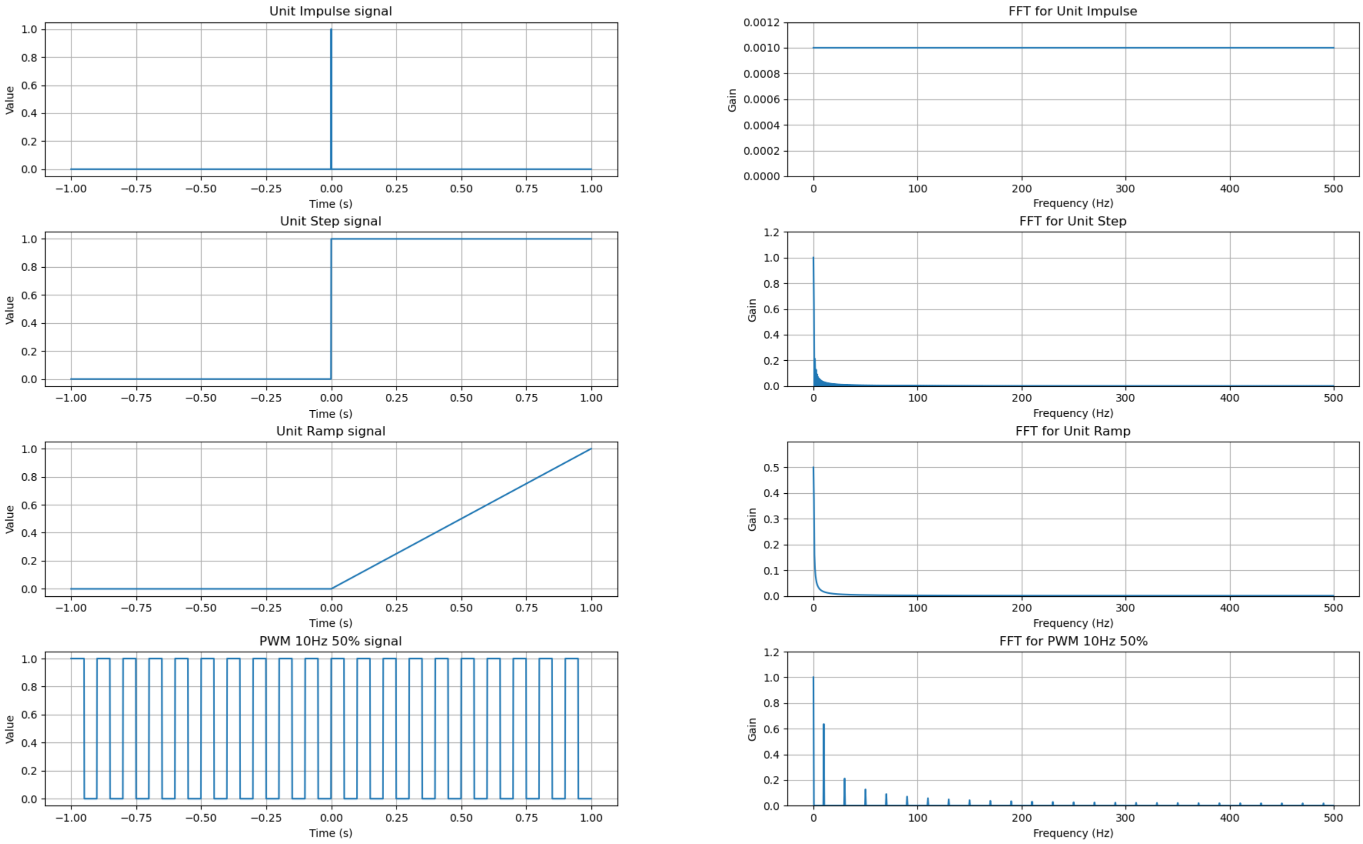


$$\Delta f_1 = \frac{1}{0.1} = 10Hz,\ \Delta f_2 = \frac{1}{0.3} \approx 3.33Hz,\ \Delta f_3 = \frac{1}{0.5} = 2Hz,\ \Delta f_4 = \frac{1}{0.55} \approx 1.81Hz,$$

We see that 8 Hz is only multiple of 2 Hz, and is not a multiple for the other frequency resolutions. Hence we get clean output for 0.5s, and aliasing for the other three cases. It is also noticable that for low sampling rates and short signal durations, we do not get the FFT we expect, because there is not enough data for the 0.1s case to make conclusions about the frequency.

Problem 2

Generate basic signals (unit impulse, unit step, unit ramp, PWM with any duty cycle) with a sampling rate of 10^3 Hz. Get the FFT of this signals using the fft tool. Draw the magnitude spectrum. Compare the results of FFT of the signals and give a comment about the difference.



For the unit impulse, we get all the frequencies on the FFT plot, with equal magnitude, because it is a very sharp signal with small timespan, which generates lots of frequencies. For unit step and ramp function, we also get many frequencies. Mostly the DC component, while higher frequencies have smaller magnitudes. For the PWM, we get odd harmonics of the PWM frequency (10Hz). This is because our function is periodic and also includes sharp edges.

```
import numpy as np
import matplotlib.pyplot as plt

fs = 1000
t = np.arange(-1, 1, 1/fs)
signals = [
    ("Unit Impulse", np.where(np.isclose(t, 0), 1, 0)),
    ("Unit Step", np.where(t >= 0, 1, 0)),
    ("Unit Ramp", np.where(t >= 0, t, 0)),
    ("PWM 10Hz 50%", np.where(
        (0 <= (10 * t - np.floor(10 * t))) &
            ((10 * t - np.floor(10 * t)) <= 0.5),
        1, 0)),
]

fig, axs = plt.subplots(len(signals), 2, squeeze=False, figsize=(8, 12))

for i, (name, s) in enumerate(signals):
    lax, rax = axs[i]

    lax.plot(t, s)
    lax.set_title(f"{name} signal")
```

```
    lax.set_xlabel("Time (s)")
    lax.set_ylabel("Value")
    lax.grid()

    fft_values = np.fft.fft(s)
    N = len(s)

    freq = np.fft.fftfreq(N, 1/fs)
    magnitude = 2 * np.abs(fft_values) / N

    rax.set_ylim(0, np.max(magnitude[:N//2]) * 1.2)
    rax.plot(freq[:N//2], magnitude[:N//2])
    rax.set_title(f"FFT for {name}")
    rax.set_xlabel("Frequency (Hz)")
    rax.set_ylabel("Gain")
    rax.grid()

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

Problem 3

We are given the discrete-time sinusoidal segment $\{6.2\cos(0.75\pi n + 1),\ n = 0, ..., (N-1)\}$. For what values of N will the spectrum computed using the DFT have no spectral leakage?

$$\omega = 0.75\pi$$

N should be chosen, such that the signal finishes its period perfectly, which means $\omega = 2\pi \cdot \dfrac{k}{N}$ should be an integer, for at least one integer value of k. Therefore $N = \dfrac{2}{0.75}k \implies 3N = 8k$, if we choose k to be a multiple of 3, we come to the conclusion that N should be a multiple of 8, that is $N \in \{8, 16, 24,\ ...\}$.