Digital version

ENGS252 Signals and Systems - Homework 4

Mher Saribekyan A09210183

March 1, 2026

Problem 1

a) Write a code that will generate a unit impulse signal. Your code should represent the signal graphically.

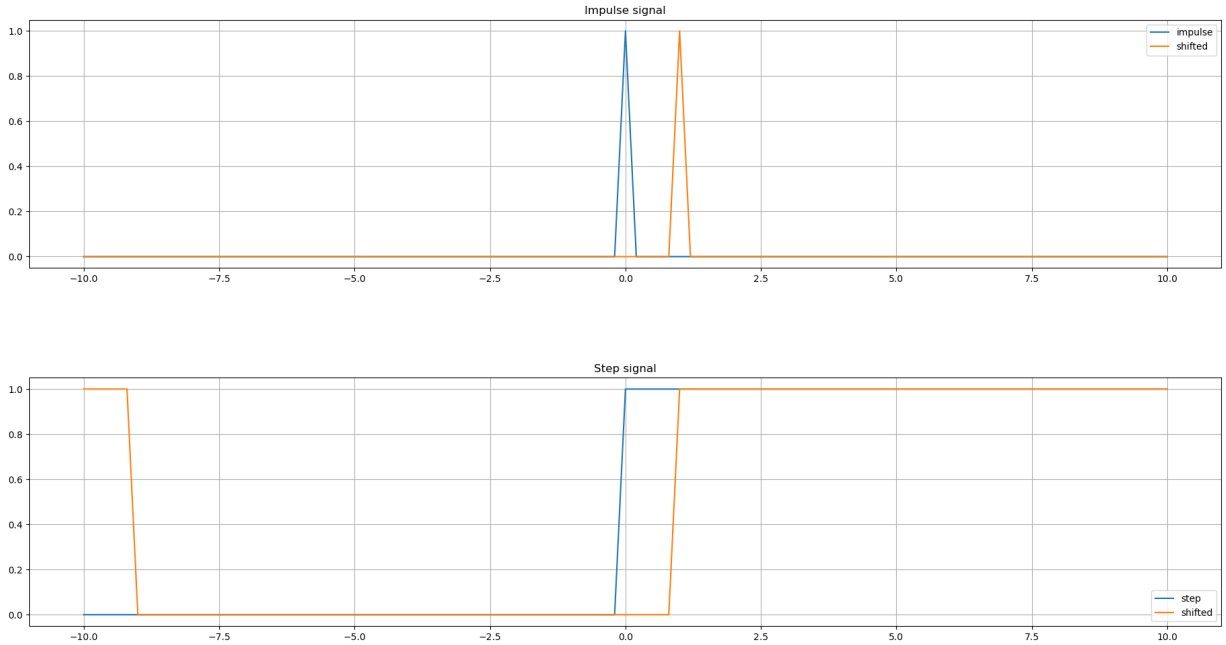
b) Write a code that will represent the unit impulse signal shifted by 5 steps.

c) Write a code that will generate a unit step signal. Your code should represent the signal graphically.

d) Write a code that will represent the unit step signal shifted by 5 steps.

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

t = np.linspace(-10, 10, 101)

def impulse(t):
    return np.where(t == 0, 1, 0)

def step(t):
    return np.where(t >= 0, 1, 0)

imp = impulse(t)                          # Unit impulse signal
imp_shifted = impulse(np.roll(t, 5))      # Shifted impulse
stp = step(t)                             # Unit step signal
stp_shifted = step(np.roll(t, 5))         # Shifted step


fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 4))

ax1.plot(t, imp, label="impulse")
ax1.plot(t, imp_shifted, label="shifted")
ax1.legend()
ax1.set_title("Impulse signal")
ax1.grid()

ax2.plot(t, stp, label="step")
ax2.plot(t, stp_shifted, label="shifted")
ax2.legend()
ax2.set_title("Step signal")
ax2.grid()
```

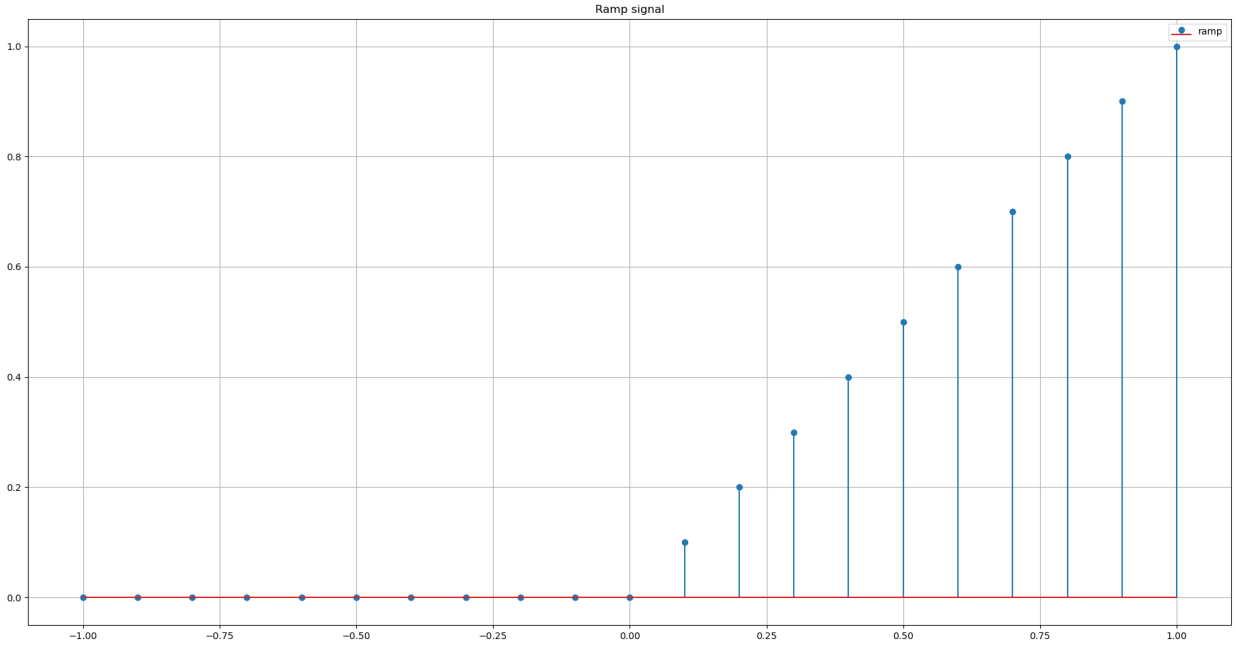
```
plt.tight_layout()
plt.show()
```



Problem 2

Write a code that will generate a unit ramp signal. Your code should represent the signal graphically. Take the sampling rate 10 Hz (this sampling rate must be seen on the time scale of the graph).

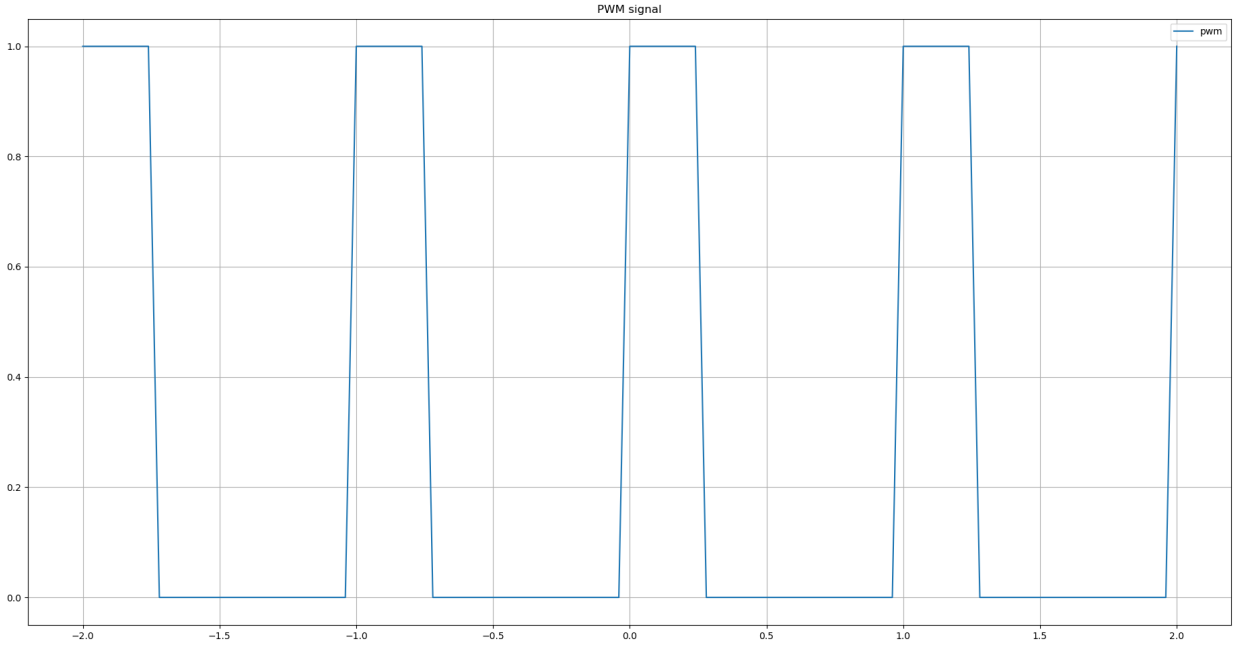
```
import matplotlib.pyplot as plt
import numpy as np

t = np.linspace(-1, 1, 21)

def ramp(t):
    return np.where(t >= 0, t, 0)

rmp = ramp(t)

fig, (ax1) = plt.subplots(1, 1, figsize=(10, 4))

ax1.stem(t, rmp, label="ramp")
ax1.legend()
ax1.set_title("Ramp signal")
ax1.grid()

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



Problem 3

Write a code that will generate a PWM signal. Mention the duty cycle of that signal.

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

t = np.linspace(-2, 2, 101)

def pwm_signal(t, duty_cycle):
    return np.where(t - np.floor(t) < duty_cycle, 1, 0)

pwm = pwm_signal(t, 0.25)

fig, (ax1) = plt.subplots(1, 1, figsize=(10, 4))

ax1.plot(t, pwm, label="pwm")
ax1.legend()
ax1.set_title("PWM signal")
ax1.grid()

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



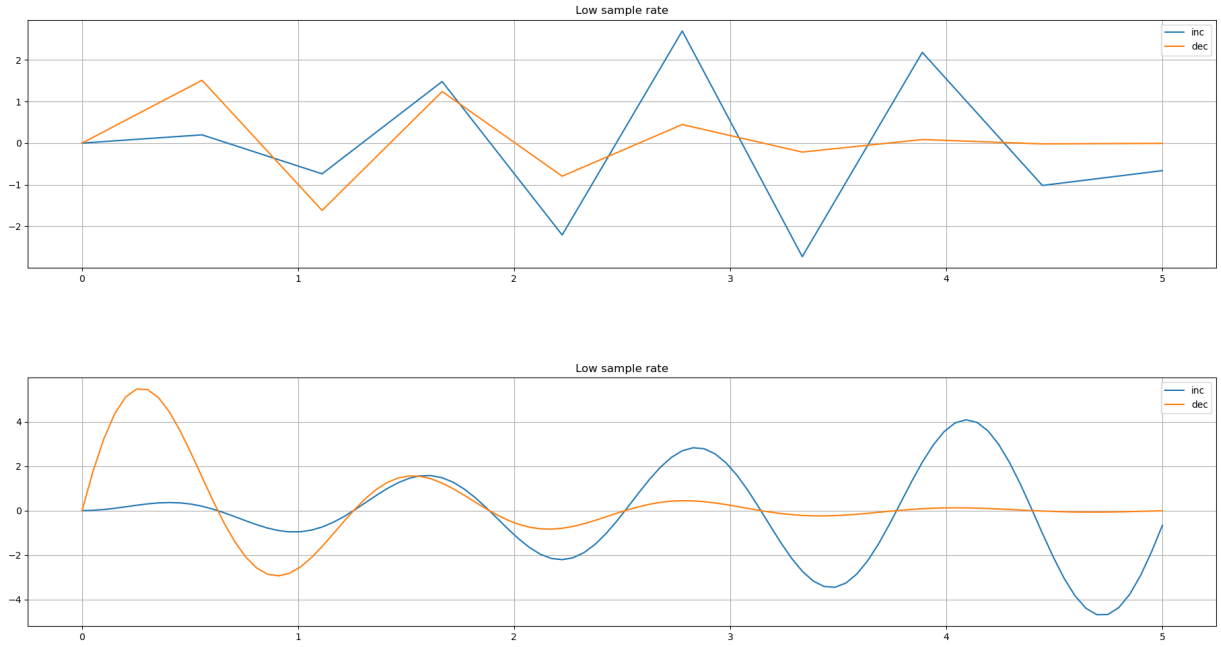
Duty cycle is 0.25 or 25%.

Problem 4

Write a code that will generate a sinusoid signal with increasing amplitude (and another code for decreasing amplitude. Your code should represent the signal graphically. Show that the precision does depend on the sampling rate.

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

t_low = np.linspace(0, 5, 10)
t_high = np.linspace(0, 5, 100)

def increasing_sine(t):
    return t * np.sin(5 * t)

def decreasing_sine(t):
    return (np.e ** (2-t)) * np.sin(5 * t)

inc_low = increasing_sine(t_low)
dec_low = decreasing_sine(t_low)
inc_high = increasing_sine(t_high)
dec_high = decreasing_sine(t_high)

fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 4))

ax1.plot(t_low, inc_low, label="inc")
ax1.plot(t_low, dec_low, label="dec")
ax1.legend()
ax1.set_title("Low sample rate")
```

```
ax1.grid()

ax2.plot(t_high, inc_high, label="inc")
ax2.plot(t_high, dec_high, label="dec")
ax2.legend()
ax2.set_title("Low sample rate")
ax2.grid()


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



Problem 5

Bring any function for PWM signal with a duty cycle equal to 40%. Such that the mean value of it is 3 V. The graph is enough.

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

t = np.linspace(-2, 2, 101)

def pwm_signal(t):
    return np.where(t - np.floor(t) < 0.6, 5, 0)

pwm = pwm_signal(t)

fig, (ax1) = plt.subplots(1, 1, figsize=(10, 4))

ax1.plot(t, pwm, label="pwm")
```

```
ax1.legend()
ax1.set_title("PWM signal")
ax1.grid()

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