Homework Signal 1 (for assignment)

Week 1

6733172621 Patthadon Phengpinij

Collaborators. ChatGPT (for LaTeX styling and grammar checking)

1 Representing Signals

Problem 1. Sketch the following signals

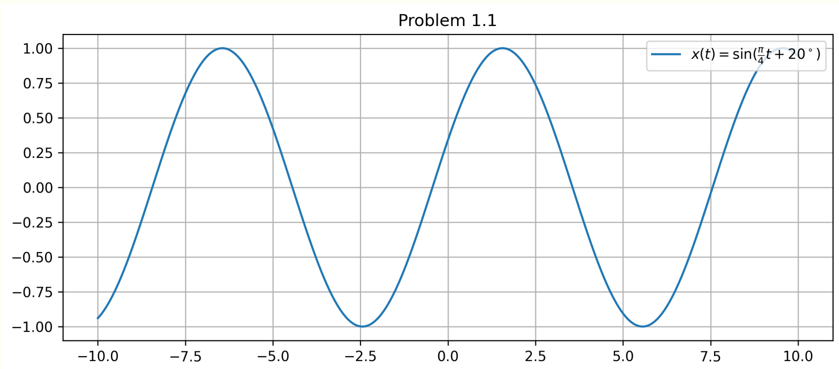
TO SUBMIT

1.1 $x(t) = \sin\left(\frac{\pi}{4}t + 20°\right)$

Solution. Using Python and Matplotlib to plot the signal $x(t) = \sin\left(\frac{\pi}{4}t + 20°\right)$:

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

fig = plt.figure()

t = np.arange(-10, 10, 0.01)
x = np.sin(np.pi/4 * t + np.pi/9)

plt.title("Problem 1.1")
plt.plot(t, x, label=r"$x(t) = \sin(\frac{\pi}{4}t + 20^\
    circ)$")
plt.legend(loc="upper right")
plt.grid(True)
plt.show()
```

The plot of the signal is shown below:



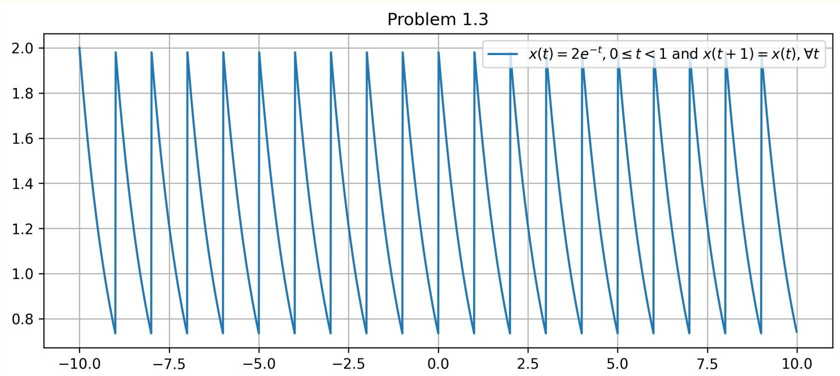
TO SUBMIT

1.3 $x(t) = 2e^{-t}, 0 \leq t < 1$ and $x(t+1) = x(t), \forall t$

Solution. Using Python and Matplotlib to plot the piecewise signal $x(t) = 2e^{-t}, 0 \leq t < 1$ and $x(t+1) = x(t), \forall t$:

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

def x3(t):
    if t >= 1:
        return x3(t - 1)
    if t < 0:
        return x3(t + 1)
    return 2 * (np.e ** (-t))

fig = plt.figure(figsize=(10, 4))

t = np.arange(-10, 10, 0.01)
x3_vectorize = np.vectorize(x3)
x = x3_vectorize(t)

plt.title("Problem 1.3")
plt.plot(t, x, label=r"$x(t) = 2e^{-t}, 0 \leq t < 1 \text{ and } x(t + 1) = x(t), \forall t$")
plt.legend(loc="upper right")
plt.grid(True)
plt.show()
```

The plot of the signal is shown below:



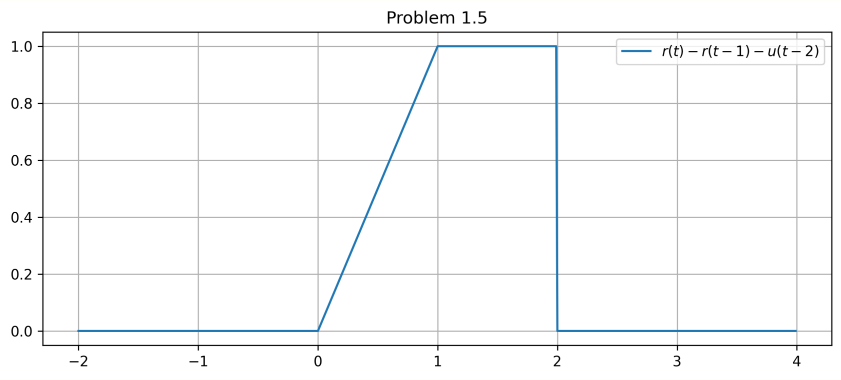
TO SUBMIT

1.5 $x(t) = r(t) - r(t-1) - u(t-2)$

Solution. Using Python and Matplotlib to plot the piecewise signal $x(t) = r(t) - r(t-1) - u(t-2)$:

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

def unit_signal(t):
    return 1.0 if t >= 0 else 0.0

def ramp_signal(t):
    return t * unit_signal(t)

unit_signal_vectorize = np.vectorize(unit_signal)
ramp_signal_vectorize = np.vectorize(ramp_signal)

fig = plt.figure(figsize=(10, 4))

t = np.arange(-2, 4, 0.01)

r1 = ramp_signal_vectorize(t)
r2 = ramp_signal_vectorize(t - 1)
u1 = unit_signal_vectorize(t - 2)

x = r1 - r2 - u1

plt.title("Problem 1.5")
plt.plot(t, x, label=r"$r(t) - r(t-1) - u(t-2)$")
plt.legend(loc="upper right")
plt.grid(True)
plt.show()
```

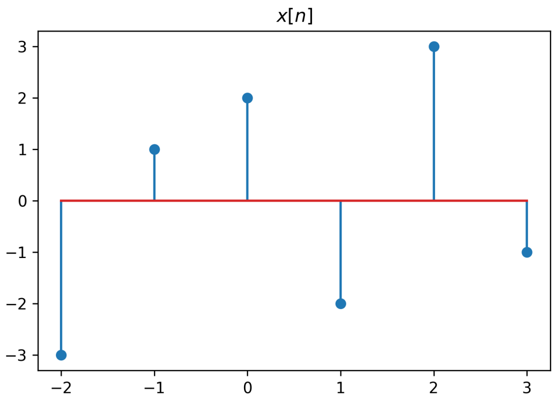
The plot of the signal is shown below:



Problem 2. For the discrete time signal $x[n]$ shown in Figure below, sketch each of the following

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

fig = plt.figure(figsize=(6, 4))

t = np.arange(-2, 4)
x_t = np.array([-3, 1, 2, -2, 3, -1])

plt.stem(t, x_t)
plt.title(r"$x[n]$")
plt.show()
```

With the resulting plot shown below:



Solution. By using Python, we can create a function to transform the signal based on the given transformation function:

```python
def transform_signal(x, n, f):
    """Return x[f(n)] for any discrete-time signal x[n].
    """
    f_n = f(n)
    f_n_int = f_n[np.floor(f_n) == f_n]

    x_new = np.zeros(f_n_int.shape[0], dtype=float)

    idx = 0
    for i, val in enumerate(f_n):
        if int(val) == val:
            x_new[idx] = x[i]
            idx += 1

    return x_new, f_n_int
```

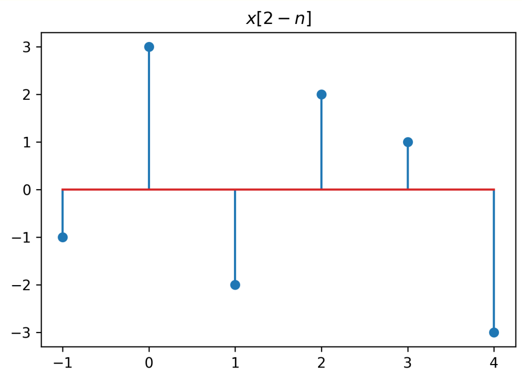
TO SUBMIT

2.1 $x[2-n]$

Solution. Using Python and Matplotlib to plot the signal $x[2-n]$:

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

fig = plt.figure(figsize=(6, 4))

t = np.arange(-2, 4)
x_t = np.array([-3, 1, 2, -2, 3, -1])

x_t, t = transform_signal(x_t, t, lambda x: 2 - x)

plt.stem(t, x_t)
plt.title(r"$x[2 - n]$")
plt.show()
```

With the resulting plot shown below:



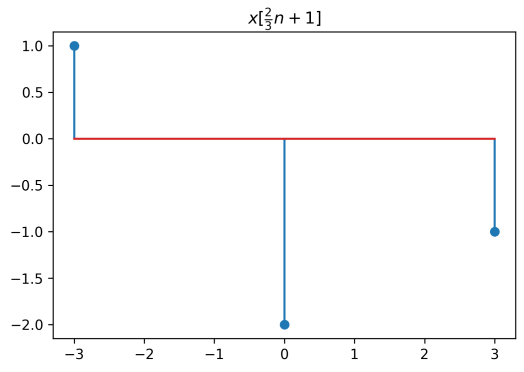
TO SUBMIT

2.3 $x\left[\frac{2}{3}n+1\right]$

Solution. Using Python and Matplotlib to plot the signal $x[\frac{2}{3}n+1]$:

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

fig = plt.figure(figsize=(6, 4))

t = np.arange(-2, 4)
x_t = np.array([-3, 1, 2, -2, 3, -1])

x_t, t = transform_signal(x_t, t, lambda x: (x - 1) * 3 /
    2)

plt.stem(t, x_t)
plt.title(r"$x[\frac{2}{3}n + 1]$")
plt.show()
```

With the resulting plot shown below:



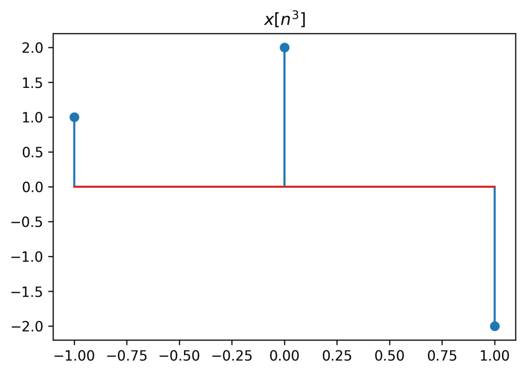
TO SUBMIT

2.5 $x[n^3]$

Solution. Using Python and Matplotlib to plot the signal $x[n^3]$:

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

fig = plt.figure(figsize=(6, 4))

t = np.arange(-2, 4)
x_t = np.array([-3, 1, 2, -2, 3, -1])

x_t, t = transform_signal(x_t, t, lambda x: np.cbrt(x))

plt.stem(t, x_t)
plt.title(r"$x[n^3]$")
plt.show()
```

With the resulting plot shown below:



Problem 9. Evaluate the following integrals

TO SUBMIT

9.1 $\int_{-\infty}^{\infty} \left(\frac{2}{3}t - \frac{3}{2}\right) \delta(t-1)\, dt$

Solution. Using the sifting property of the delta function, we have:

$$\begin{aligned}\int_{-\infty}^{\infty} \left(\frac{2}{3}t - \frac{3}{2}\right) \delta(t-1)\, dt &= \left(\frac{2}{3}(1) - \frac{3}{2}\right) \\ &= \frac{2}{3} - \frac{3}{2} \\ \int_{-\infty}^{\infty} \left(\frac{2}{3}t - \frac{3}{2}\right) \delta(t-1)\, dt &= \boxed{-\frac{5}{6}}\end{aligned}$$

TO SUBMIT

9.3 $\int_{-3}^{-2} \left[e^{(-t+1)} + \sin\left(\frac{2\pi t}{3}\right)\right] \delta\left(t - \frac{3}{2}\right)\, dt$

Solution. Because the argument of the delta function $t - \frac{3}{2}$ has its root at $t = \frac{3}{2}$, which is outside the integration limits of -3 to -2, the integral evaluates to zero:

$$\int_{-3}^{-2} \left[e^{(-t+1)} + \sin\left(\frac{2\pi t}{3}\right)\right] \delta\left(t - \frac{3}{2}\right)\, dt = \boxed{0}$$