

# Homework Signal 1

Week 1

6733172621 Pathadon Phengpinij

*Collaborators.* ChatGPT (for L<sup>A</sup>T<sub>E</sub>X styling and grammar checking)

## 1 Representing Signals

**Problem 1.** Sketch the following signals

TO SUBMIT

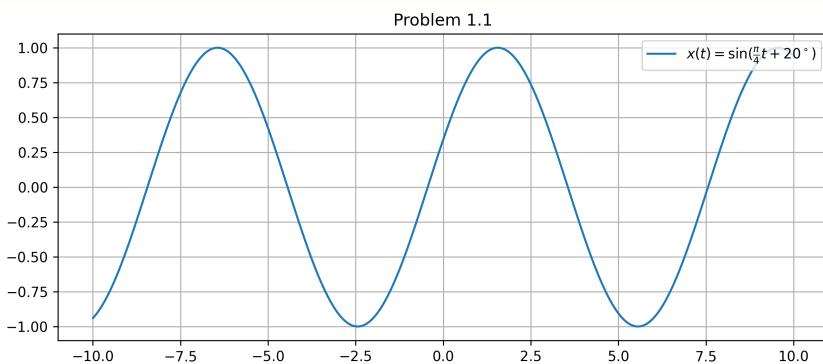
a)  $x(t) = \sin\left(\frac{\pi}{4}t + 20^\circ\right)$

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

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure()
5
6 t = np.arange(-10, 10, 0.01)
7 x = np.sin(np.pi/4 * t + np.pi/9)
8
9 plt.title("Problem 1.1")
10 plt.plot(t, x, label=r"$x(t) = \sin(\frac{\pi}{4}t + 20^\circ)$")
11 plt.legend(loc="upper right")
12 plt.grid(True)
13 plt.show()
```

The plot of the signal is shown below:



$$\text{b)} \quad x(t) = \begin{cases} t+2, & t \leq 2 \\ 0, & -2 \leq t \leq 2 \\ t-2, & t \geq 2 \end{cases}$$

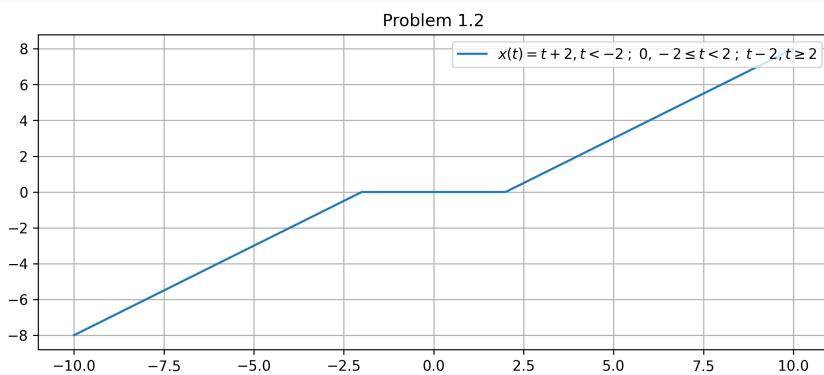
**Solution.** Using Python and Matplotlib to plot the piecewise signal  $x(t)$ :

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(10, 4))
5
6 t = np.arange(-10, 10, 0.01)
7 x = np.piecewise(t, [t < -2, (t >= -2) & (t < 2), t >=
8     2], [lambda t: t + 2, 0, lambda t: t - 2])
9
10 plt.title("Problem 1.2")
11 plt.plot(t, x, label=r"$x(t) = t+2, t < -2 ; 0, -2 \leq t < 2 ; t-2, t \geq 2$")
12 plt.legend(loc="upper right")
13 plt.grid(True)
14 plt.show()

```

The plot of the signal is shown below:



## TO SUBMIT

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

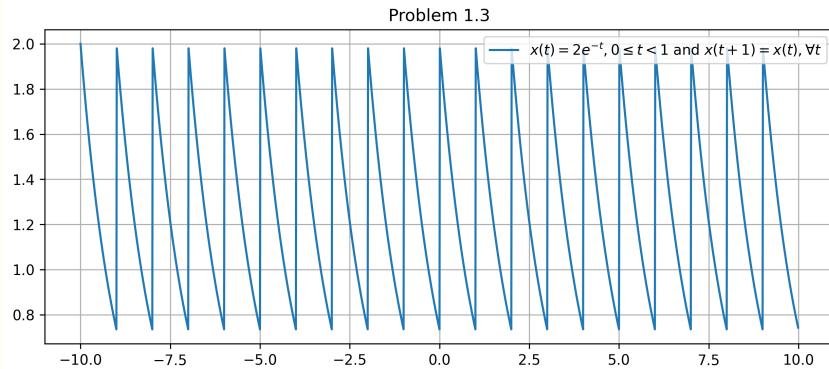
**Solution.** Using Python and Matplotlib to plot the piecewise signal  $x(t) = 2e^{-t}$ ,  $0 \leq t < 1$  and  $x(t+1) = x(t)$ ,  $\forall t$ :

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 def x3(t):
5     if t >= 1:
6         return x3(t - 1)
7     if t < 0:
8         return x3(t + 1)
9     return 2 * (np.e ** (-t))
10
11 fig = plt.figure(figsize=(10, 4))
12
13 t = np.arange(-10, 10, 0.01)
14 x3_vectorize = np.vectorize(x3)
15 x = x3_vectorize(t)
16
17 plt.title("Problem 1.3")
18 plt.plot(t, x, label=r"$x(t) = 2e^{-t}$, $0 \leq t < 1$ \
19           and $x(t+1) = x(t)$, $\forall t$")
20 plt.legend(loc="upper right")
21 plt.grid(True)
22 plt.show()

```

The plot of the signal is shown below:



d)  $x(t) = u(t) + 5u(t - 1) + 2u(t - 2)$

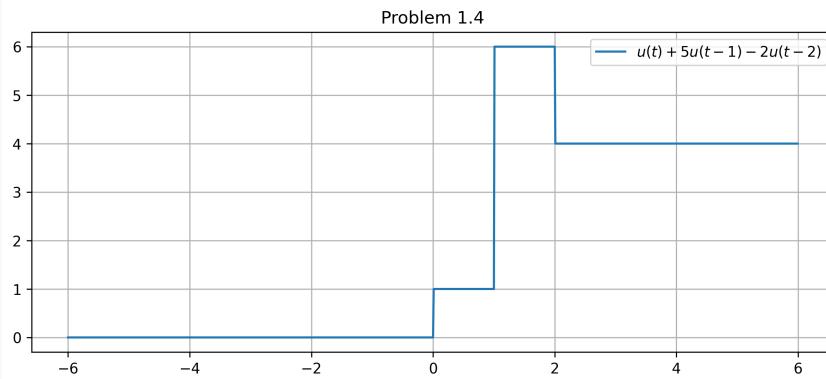
**Solution.** Using Python and Matplotlib to plot the piecewise signal  $x(t) = u(t) + 5u(t - 1) + 2u(t - 2)$ :

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 def unit_signal(t):
5     return 1.0 if t >= 0 else 0.0
6
7 unit_signal_vectorize = np.vectorize(unit_signal)
8
9 fig = plt.figure(figsize=(10, 4))
10
11 t = np.arange(-6, 6, 0.01)
12
13 u1 = unit_signal_vectorize(t)
14 u2 = unit_signal_vectorize(t - 1)
15 u3 = unit_signal_vectorize(t - 2)
16
17 x = u1 + 5 * u2 - 2 * u3
18
19 plt.title("Problem 1.4")
20 plt.plot(t, x, label=r"$u(t) + 5u(t-1) - 2u(t-2)$")
21 plt.legend(loc="upper right")
22 plt.grid(True)
23 plt.show()

```

The plot of the signal is shown below:



## TO SUBMIT

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

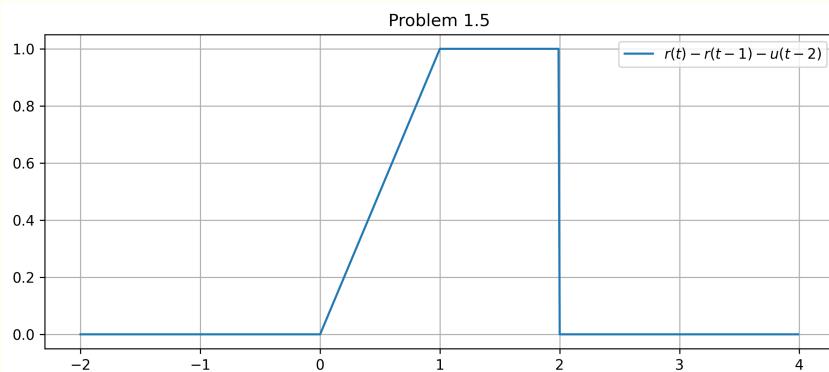
**Solution.** Using Python and Matplotlib to plot the piecewise signal  $x(t) = r(t) - r(t - 1) - u(t - 2)$ :

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 def unit_signal(t):
5     return 1.0 if t >= 0 else 0.0
6
7 def ramp_signal(t):
8     return t * unit_signal(t)
9
10 unit_signal_vectorize = np.vectorize(unit_signal)
11 ramp_signal_vectorize = np.vectorize(ramp_signal)
12
13 fig = plt.figure(figsize=(10, 4))
14
15 t = np.arange(-2, 4, 0.01)
16
17 r1 = ramp_signal_vectorize(t)
18 r2 = ramp_signal_vectorize(t - 1)
19 u1 = unit_signal_vectorize(t - 2)
20
21 x = r1 - r2 - u1
22
23 plt.title("Problem 1.5")
24 plt.plot(t, x, label=r"$r(t) - r(t-1) - u(t-2)$")
25 plt.legend(loc="upper right")
26 plt.grid(True)
27 plt.show()

```

The plot of the signal is shown below:



**Problem 2.** Determine whether each of following signals is periodic, and if so, find its period.

a)  $x(t) = \sin\left(\frac{\pi}{3}t\right) + \cos\left(\frac{8\pi}{3}t\right)$

**Solution.** Consider each part of the signal separately:

$$\sin\left(\frac{\pi}{3}t\right) \text{ has a period of } T_1 = \frac{2\pi}{\frac{\pi}{3}} = 6$$

$$\cos\left(\frac{8\pi}{3}t\right) \text{ has a period of } T_2 = \frac{2\pi}{\frac{8\pi}{3}} = \frac{3}{4}$$

Considering the least common multiple of the two periods:

$$T = \text{lcm}(T_1, T_2) = \text{lcm}(6, \frac{3}{4}) = 6$$

Thus, the signal  $x(t) = \sin\left(\frac{\pi}{3}t\right) + \cos\left(\frac{8\pi}{3}t\right)$  **is periodic** with a period of  $[T = 6]$ .

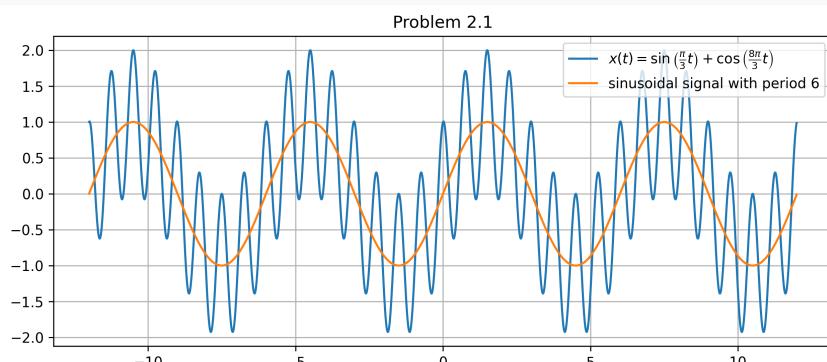
By using Python and Matplotlib, we can visualize the periodicity of the signal:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(10, 4))
5
6 t = np.arange(-12, 12, 0.01)
7
8 x1 = np.sin(np.pi/3 * t)
9 x2 = np.cos(8*np.pi/3 * t)
10
11 x = np.sin(np.pi/3 * t)
12
13 plt.title("Problem 2.1")
14 plt.plot(t, x1 + x2, label=r"$x(t) = \sin\left(\frac{\pi}{3}t\right) + \cos\left(\frac{8\pi}{3}t\right)$")
15 plt.plot(t, x, label="sinusoidal signal with period 6")
16 plt.grid(True)
17 plt.legend(loc="upper right")
18 plt.show()

```

The plot of the signal is shown below:



b)  $x(t) = \exp(j\frac{7\pi}{6}t) + \exp(j\frac{5\pi}{6}t)$

**Solution.** Consider each part of the signal separately:

$$\exp\left(j\frac{7\pi}{6}t\right) \text{ has a period of } T_1 = \frac{2\pi}{\frac{7\pi}{6}} = \frac{12}{7}$$

$$\exp\left(j\frac{5\pi}{6}t\right) \text{ has a period of } T_2 = \frac{2\pi}{\frac{5\pi}{6}} = \frac{12}{5}$$

Considering the least common multiple of the two periods:

$$T = \text{lcm}(T_1, T_2) = \text{lcm}\left(\frac{12}{7}, \frac{12}{5}\right) = \frac{12}{1} = 12$$

Thus, the signal  $x(t) = \exp(j\frac{7\pi}{6}t) + \exp(j\frac{5\pi}{6}t)$  **is periodic** with a period of  $T = 12$ .

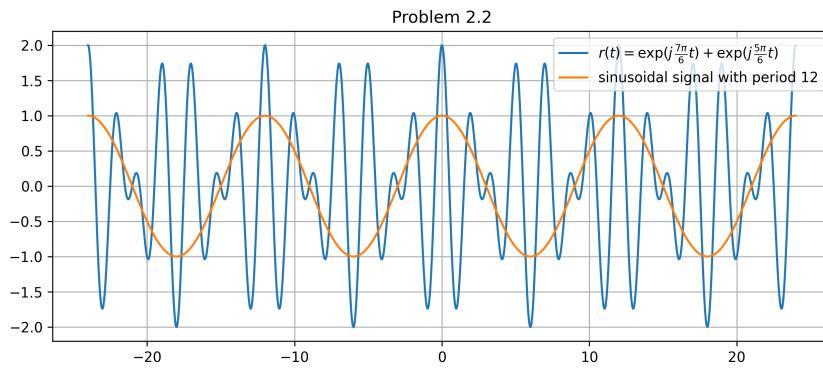
By using Python and Matplotlib, we can visualize the periodicity of the signal:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(10, 4))
5
6 t = np.arange(-24, 24, 0.01)
7
8 x1 = np.exp(1j * 7*np.pi/6 * t)
9 x2 = np.exp(1j * 5*np.pi/6 * t)
10
11 x = np.exp(1j * np.pi/6 * t)
12
13 plt.title("Problem 2.2")
14 plt.plot(t, x1 + x2, label=r"$r(t) = \exp(j\frac{7\pi}{6}t) + \exp(j\frac{5\pi}{6}t)$")
15 plt.plot(t, x, label="sinusoidal signal with period 12")
16 plt.grid(True)
17 plt.legend(loc="upper right")
18 plt.show()

```

The plot of the signal is shown below:



c)  $x(t) = \exp(j\frac{7\pi}{6}t) + \exp(\frac{5\pi}{6}t)$

**Solution.** Consider each part of the signal separately:

$$\exp\left(j\frac{7\pi}{6}t\right) \text{ has a period of } T_1 = \frac{2\pi}{\frac{7\pi}{6}} = \frac{12}{7}$$

$\exp\left(\frac{5\pi}{6}t\right)$  has no period since it is not a sinusoidal function. (non-periodic signal)

Thus, the signal  $x(t) = \exp(j\frac{7\pi}{6}t) + \exp(\frac{5\pi}{6}t)$  **is non-periodic** since one part of the signal is non-periodic.

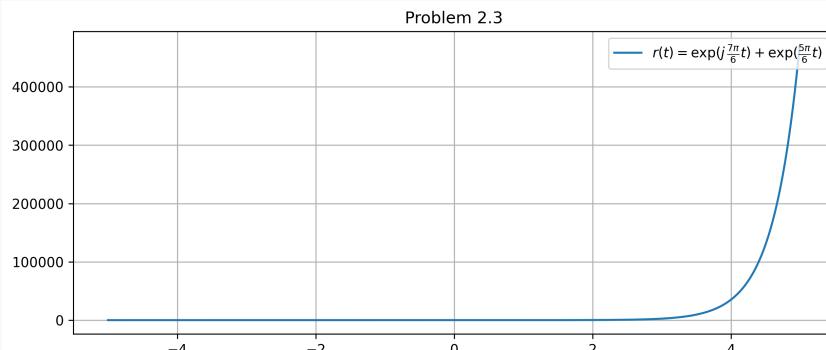
By using Python and Matplotlib, we can visualize the signal:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(10, 4))
5
6 t = np.arange(-5, 5, 0.01)
7
8 x1 = np.exp(1j * 7*np.pi/6 * t)
9 x2 = np.exp(5*np.pi/6 * t)
10
11 plt.title("Problem 2.3")
12 plt.plot(t, x1 + x2, label=r"$r(t) = \exp(j\frac{7\pi}{6}t) + \exp(\frac{5\pi}{6}t)$")
13 plt.grid(True)
14 plt.legend(loc="upper right")
15 plt.show()

```

The plot of the signal is shown below:



**Problem 3.** Determine whether the following signals are power or energy signals or neither. Justify your answers

a)  $x(t) = A \sin(t), -\infty < t < \infty$

**Solution.** Consider the energy of the signal:

$$\begin{aligned} E &= \lim_{N \rightarrow \infty} \int_{-N}^N |x(t)|^2 dt \\ &= \lim_{N \rightarrow \infty} \int_{-N}^N |A \sin(t)|^2 dt \\ &= \lim_{N \rightarrow \infty} A^2 \int_{-N}^N \sin^2(t) dt \\ &= \lim_{N \rightarrow \infty} A^2 \int_{-N}^N \frac{1 - \cos(2t)}{2} dt \\ &= \lim_{N \rightarrow \infty} \frac{A^2}{2} \left[ t - \frac{\sin(2t)}{2} \right]_{-N}^N \\ &= \lim_{N \rightarrow \infty} \frac{A^2}{2} (N - (-N)) \\ &= \lim_{N \rightarrow \infty} A^2 N \\ E &= \infty \end{aligned}$$

The integral diverges, so the energy is infinite.

Now, consider the power of the signal:

$$\begin{aligned} P &= \lim_{N \rightarrow \infty} \frac{1}{2N} \int_{-N}^N |x(t)|^2 dt \\ &= \lim_{N \rightarrow \infty} \frac{1}{2N} A^2 N \\ P &= \frac{A^2}{2} \end{aligned}$$

The integral converges to a finite value, so the power is finite.

Thus, the signal  $x(t) = A \sin(t)$  is **a power signal** with power  $P = \frac{A^2}{2}$ .

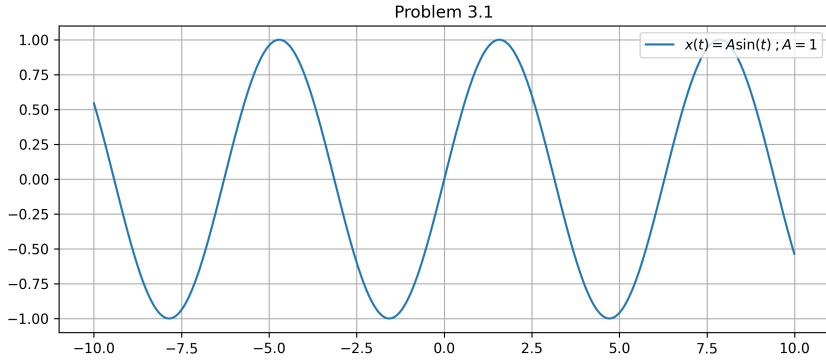
By using Python and Matplotlib, we can visualize the signal:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(10, 4))
5
6 t = np.arange(-10, 10, 0.01)
7 x = np.sin(t)
8
9 plt.title("Problem 3.1")
10 plt.plot(t, x, label=r"$x(t) = A \sin(t)$; $A = 1$")
11 plt.grid(True)
12 plt.legend(loc="upper right")
13 plt.show()

```

The plot of the signal is shown below:



b)  $x(t) = A(u(t - a) - u(t + a))$ ,  $a > 0$

**Solution.** Consider the energy of the signal:

$$\begin{aligned}
 E &= \lim_{N \rightarrow \infty} \int_{-N}^N |x(t)|^2 dt \\
 &= \lim_{N \rightarrow \infty} \int_{-N}^N |A(u(t - a) - u(t + a))|^2 dt \\
 &= \lim_{N \rightarrow \infty} A^2 \int_{-N}^N |u(t - a) - u(t + a)|^2 dt \\
 &= \lim_{N \rightarrow \infty} A^2 \int_{-N}^N (u(t - a) - u(t + a))^2 dt \\
 &= \lim_{N \rightarrow \infty} A^2 \int_{-N}^N (u(t + a) - u(t - a)) dt \\
 &= \lim_{N \rightarrow \infty} A^2 \int_{-a}^a 1 dt \\
 &= \lim_{N \rightarrow \infty} A^2(a - (-a)) \\
 E &= 2aA^2
 \end{aligned}$$

The integral converges to a finite value, so the energy is finite.

Now, consider the power of the signal:

$$\begin{aligned}
 P &= \lim_{N \rightarrow \infty} \frac{1}{2N} \int_{-N}^N |x(t)|^2 dt \\
 &= \lim_{N \rightarrow \infty} \frac{1}{2N} 2aA^2 \\
 P &= 0
 \end{aligned}$$

The integral converges to 0, so the power is 0.

Thus, the signal  $x(t) = A(u(t - a) - u(t + a))$  is **a energy signal** with energy  $E = 2aA^2$ .

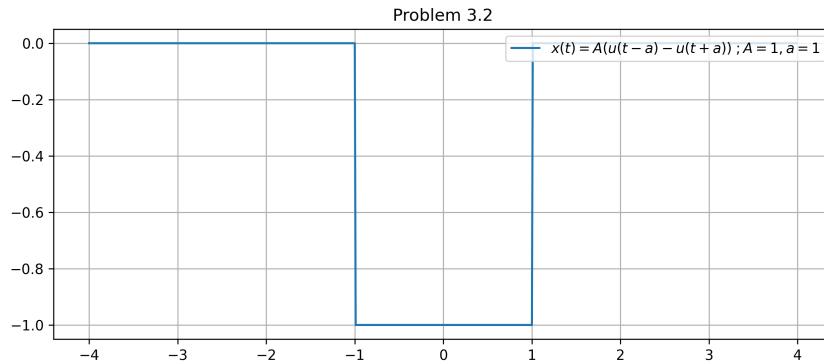
By using Python and Matplotlib, we can visualize the signal:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 def unit_signal(t):
5     return 1.0 if t >= 0 else 0.0
6
7 unit_signal_vectorize = np.vectorize(unit_signal)
8
9 fig = plt.figure(figsize=(10, 4))
10
11 t = np.arange(-4, 4, 0.01)
12 x = unit_signal_vectorize(t - 1) - unit_signal_vectorize(
13     t + 1)
14
15 plt.title("Problem 3.2")
16 plt.plot(t, x, label=r"$x(t) = A(u(t-a) - u(t+a))$")
17 plt.grid(True)
18 plt.legend(loc="upper right")
19 plt.show()

```

The plot of the signal is shown below:



c)  $x(t) = \exp(-at)u(t)$ ,  $a > 0$

**Solution.** Consider the energy of the signal:

$$\begin{aligned}
E &= \lim_{N \rightarrow \infty} \int_{-N}^N |x(t)|^2 dt \\
&= \lim_{N \rightarrow \infty} \int_{-N}^N |\exp(-at)u(t)|^2 dt \\
&= \lim_{N \rightarrow \infty} \int_0^N \exp(-2at) (u(t))^2 dt \\
&= \lim_{N \rightarrow \infty} \left[ -\frac{1}{2a} \exp(-2at) \right]_0^N \\
E &= \lim_{N \rightarrow \infty} \left( -\frac{1}{2a} \exp(-2aN) + \frac{1}{2a} \right) = \frac{1}{2a}
\end{aligned}$$

The integral converges to a finite value, so the energy is finite.

Now, consider the power of the signal:

$$\begin{aligned} P &= \lim_{N \rightarrow \infty} \frac{1}{2N} \int_{-N}^N |x(t)|^2 dt \\ &= \lim_{N \rightarrow \infty} \frac{1}{2N} \frac{1}{2a} \\ P &= 0 \end{aligned}$$

The integral converges to 0, so the power is 0.

Thus, the signal  $x(t) = \exp(-at)u(t)$ ,  $a > 0$  is **a energy signal** with energy  $E = \frac{1}{2a}$ .

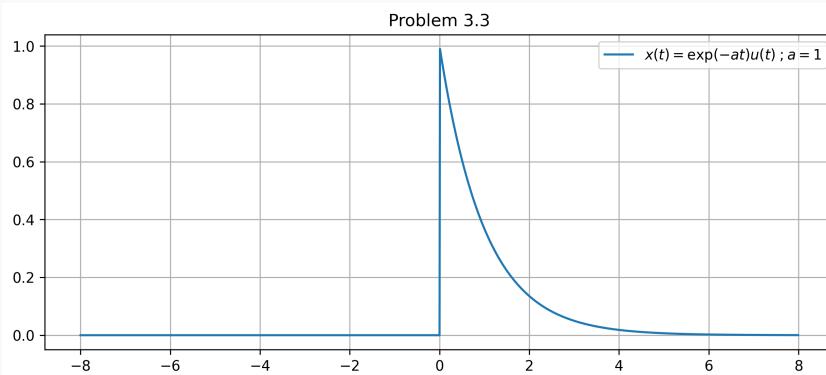
By using Python and Matplotlib, we can visualize the signal:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 def unit_signal(t):
5     return 1.0 if t >= 0 else 0.0
6
7 unit_signal_vectorize = np.vectorize(unit_signal)
8
9 fig = plt.figure(figsize=(10, 4))
10
11 t = np.arange(-8, 8, 0.01)
12 x = np.exp(-t) * unit_signal_vectorize(t)
13
14 plt.title("Problem 3.3")
15 plt.plot(t, x, label=r"$x(t) = \exp(-at)u(t)$; $a = 1$")
16 plt.grid(True)
17 plt.legend(loc="upper right")
18 plt.show()

```

The plot of the signal is shown below:



d)  $x(t) = A \exp(bt)u(t)$ ,  $b > 0$

**Solution.** Consider the energy of the signal:

$$\begin{aligned} E &= \lim_{N \rightarrow \infty} \int_{-N}^N |x(t)|^2 dt \\ &= \lim_{N \rightarrow \infty} \int_{-N}^N |A \exp(bt)u(t)|^2 dt \\ &= \lim_{N \rightarrow \infty} \int_0^N |A \exp(bt)|^2 dt \\ &= \lim_{N \rightarrow \infty} \int_0^N A^2 \exp(2bt) dt \\ &= \lim_{N \rightarrow \infty} A^2 \left[ \frac{1}{2b} \exp(2bt) \right]_0^N \\ &= \lim_{N \rightarrow \infty} A^2 \left( \frac{1}{2b} \exp(2bN) - \frac{1}{2b} \right) \\ E &= \infty \end{aligned}$$

The integral diverges, so the energy is infinite.

Now, consider the power of the signal:

$$\begin{aligned} P &= \lim_{N \rightarrow \infty} \frac{1}{2N} \int_{-N}^N |x(t)|^2 dt \\ &= \lim_{N \rightarrow \infty} \frac{1}{2N} A^2 \left( \frac{1}{2b} \exp(2bN) - \frac{1}{2b} \right) \\ P &= \infty \end{aligned}$$

The integral diverges, so the power is infinite.

Thus, the signal  $x(t) = A \exp(bt)u(t)$ ,  $b > 0$  is **neither a energy nor a power signal**.

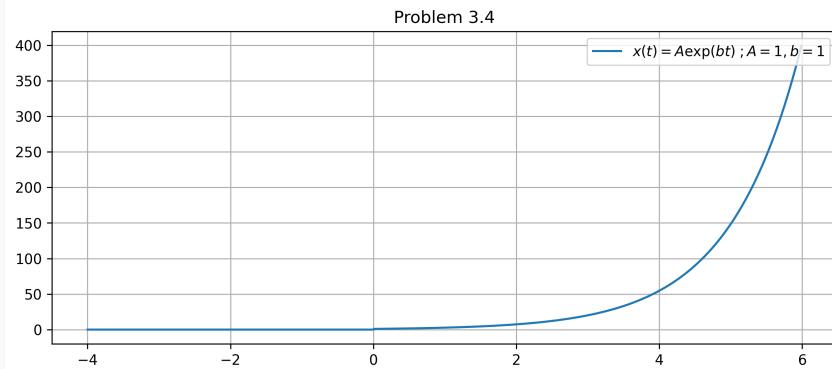
By using Python and Matplotlib, we can visualize the signal:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 def unit_signal(t):
5     return 1.0 if t >= 0 else 0.0
6
7 unit_signal_vectorize = np.vectorize(unit_signal)
8
9 fig = plt.figure(figsize=(10, 4))
10
11 t = np.arange(-4, 6, 0.01)
12 x = np.exp(t) * unit_signal_vectorize(t)
13
14 plt.title("Problem 3.4")
15 plt.plot(t, x, label=r"$x(t) = A \exp(bt) ; A = 1, b = 1$")
16 plt.grid(True)
17 plt.legend(loc="upper right")
18 plt.show()

```

The plot of the signal is shown below:



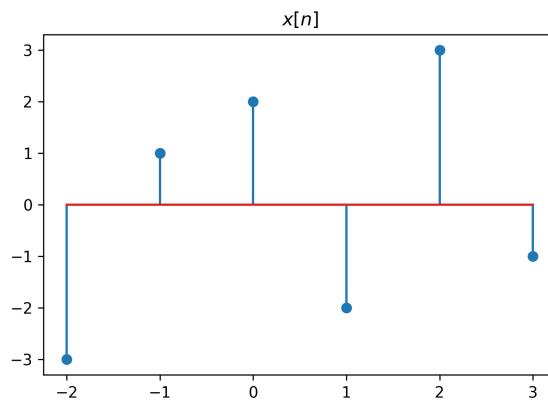
**Problem 4.** For the discrete time signal  $x[n]$  shown in Figure below, sketch each of the following

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(6, 4))
5
6 t = np.arange(-2, 4)
7 x_t = np.array([-3, 1, 2, -2, 3, -1])
8
9 plt.stem(t, x_t)
10 plt.title(r"$x[n]$")
11 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:

```

1 def transform_signal(x, n, f):
2     """Return x[f(n)] for any discrete-time signal x[n]."""
3
4     f_n = f(n)
5     f_n_int = f_n[int(f_n) == f_n]
6
7     x_new = np.zeros(f_n_int.shape[0], dtype=float)
8
9     idx = 0
10    for i, val in enumerate(f_n):
11        if int(val) == val:
12            x_new[idx] = x[i]
13            idx += 1
14
15    return x_new, f_n_int

```

### TO SUBMIT

a)  $x[2 - n]$

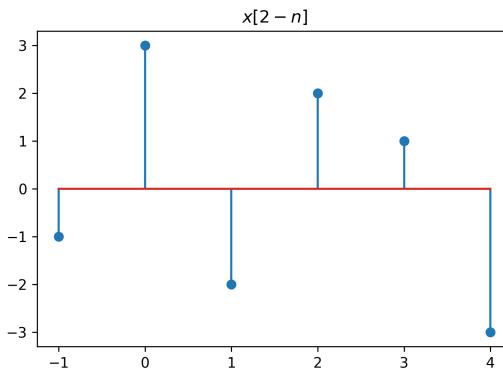
**Solution.** Using Python and Matplotlib to plot the signal  $x[2 - n]$ :

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(6, 4))
5
6 t = np.arange(-2, 4)
7 x_t = np.array([-3, 1, 2, -2, 3, -1])
8
9 x_t, t = transform_signal(x_t, t, lambda x: 2 - x)
10
11 plt.stem(t, x_t)
12 plt.title(r"$x[2 - n]$")
13 plt.show()

```

With the resulting plot shown below:



b)  $x[3n - 4]$

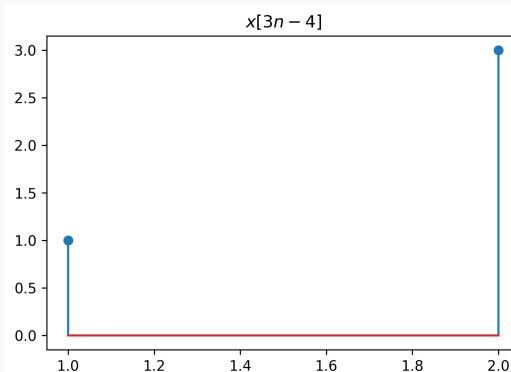
**Solution.** Using Python and Matplotlib to plot the signal  $x[3n - 4]$ :

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(6, 4))
5
6 t = np.arange(-2, 4)
7 x_t = np.array([-3, 1, 2, -2, 3, -1])
8
9 x_t, t = transform_signal(x_t, t, lambda x: (x + 4) / 3)
10
11 plt.stem(t, x_t)
12 plt.title(r"x[3n - 4]")
13 plt.show()

```

With the resulting plot shown below:



### TO SUBMIT

c)  $x[\frac{2}{3}n + 1]$

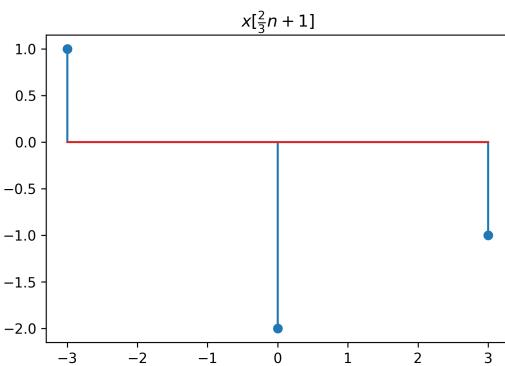
**Solution.** Using Python and Matplotlib to plot the signal  $x[\frac{2}{3}n + 1]$ :

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(6, 4))
5
6 t = np.arange(-2, 4)
7 x_t = np.array([-3, 1, 2, -2, 3, -1])
8
9 x_t, t = transform_signal(x_t, t, lambda x: (x - 1) * 3 / 2)
10
11 plt.stem(t, x_t)
12 plt.title(r"x[\frac{2}{3}n + 1]")
13 plt.show()

```

With the resulting plot shown below:



d)  $x[-\frac{n+8}{4}]$

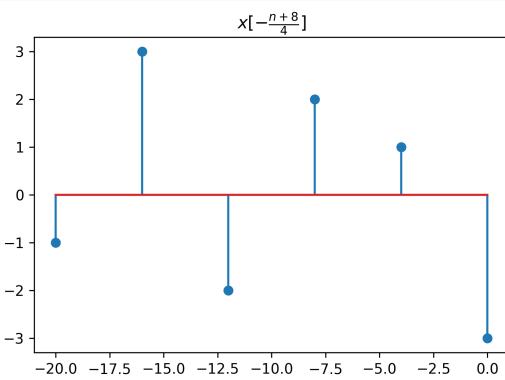
Solution. Using Python and Matplotlib to plot the signal  $x[-\frac{n+8}{4}]$ :

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(6, 4))
5
6 t = np.arange(-2, 4)
7 x_t = np.array([-3, 1, 2, -2, 3, -1])
8
9 x_t, t = transform_signal(x_t, t, lambda x: (-4 * x) - 8)
10
11 plt.stem(t, x_t)
12 plt.title(r"x[-\frac{n+8}{4}]")
13 plt.show()

```

With the resulting plot shown below:



**TO SUBMIT**e)  $x[n^3]$ 

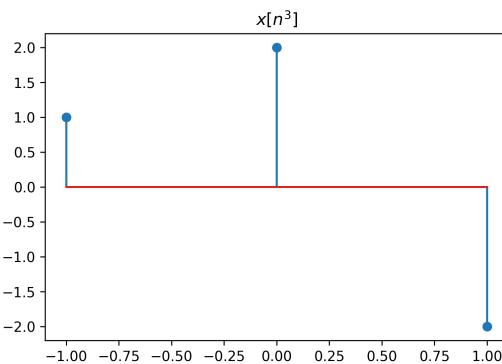
**Solution.** Using Python and Matplotlib to plot the signal  $x[n^3]$ :

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(6, 4))
5
6 t = np.arange(-2, 4)
7 x_t = np.array([-3, 1, 2, -2, 3, -1])
8
9 x_t, t = transform_signal(x_t, t, lambda x: np.cbrt(x))
10
11 plt.stem(t, x_t)
12 plt.title(r"$x[n^3]$")
13 plt.show()

```

With the resulting plot shown below:

f)  $x[2 - n] + x[3n - 4]$ 

**Solution.** Introduce a helper function to add two signals:

```

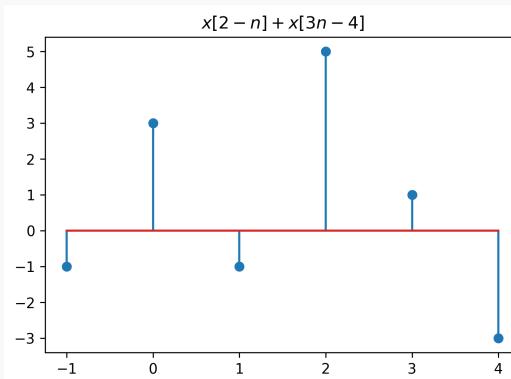
1 def add_discrete_signals(x1, t1, x2, t2):
2     """Return x1[n] + x2[n] for any discrete-time signals
3     x1[n] and x2[n]."""
4     t = np.union1d(t1, t2)
5     x = np.zeros(t.shape[0], dtype=float)
6
7     for i, val in enumerate(t):
8         if val in t1:
9             x[i] += x1[np.where(t1 == val)[0][0]]
10            if val in t2:
11                x[i] += x2[np.where(t2 == val)[0][0]]
12
13    return x, t

```

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

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(6, 4))
5
6 t = np.arange(-2, 4)
7 x_t = np.array([-3, 1, 2, -2, 3, -1])
8
9 x_t_1, t_1 = transform_signal(x_t, t, lambda x: 2 - x)
10 x_t_2, t_2 = transform_signal(x_t, t, lambda x: (x + 4) /
11     3)
12
13 x_t, t = add_discrete_signals(x_t_1, t_1, x_t_2, t_2)
14
15 plt.stem(t, x_t)
16 plt.title("x[2-n] + x[3n-4]")
17 plt.show()
```

With the resulting plot shown below:



**Problem 5.** Determine whether each of following signals is periodic, and if so, find its period.

a)  $x[n] = \sin\left(\frac{\pi n}{4} + \frac{\pi}{8}\right)$

**Solution.** Consider the signal:

$$\sin\left(\frac{\pi n}{4} + \frac{\pi}{8}\right) \text{ has a period of } N = \frac{2\pi}{\frac{\pi}{4}} = 8$$

Thus, the signal  $x[n] = \sin\left(\frac{\pi n}{4} + \frac{\pi}{8}\right)$  **is periodic** with a period of  $N = 8$ .

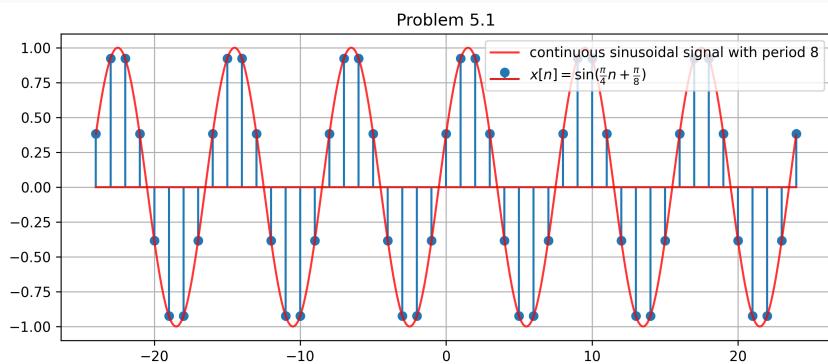
By using Python and Matplotlib, we can visualize the periodicity of the signal:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(10, 4))
5
6 t = np.arange(-24, 25, 1)
7 x = np.sin(np.pi/4 * t + np.pi/8)
8
9 tc = np.arange(-24, 24, 0.01)
10 xc = np.sin(np.pi/4 * tc + np.pi/8)
11
12 plt.title("Problem 5.1")
13 plt.stem(t, x, label=r"$x[n] = \sin(\frac{\pi}{4} n + \frac{\pi}{8})$")
14 plt.plot(tc, xc, "r", alpha=0.8, label="continuous sinusoidal signal with period 8")
15 plt.grid(True)
16 plt.legend(loc="upper right")
17 plt.show()

```

The plot of the signal is shown below:



b)  $x[n] = \sin\left(\frac{3\pi n}{4}\right) + \sin\left(\frac{\pi}{3}n\right)$

**Solution.** Consider the signal:

$$\sin\left(\frac{3\pi n}{4}\right) \text{ has a period of } N_1 = \frac{2\pi}{\frac{3\pi}{4}} = \frac{8}{3}$$

$$\sin\left(\frac{\pi n}{3}\right) \text{ has a period of } N_2 = \frac{2\pi}{\frac{\pi}{3}} = 6$$

Considering the least common multiple of the two periods:

$$N = \text{lcm}(N_1, N_2) = \text{lcm}\left(\frac{8}{3}, 6\right) = \text{lcm}\left(\frac{8}{3}, \frac{18}{3}\right) = \frac{72}{3} = 24$$

Thus, the signal  $x[n] = \sin\left(\frac{3\pi n}{4}\right) + \sin\left(\frac{\pi}{3}n\right)$  **is periodic** with a period of  $N = 24$ .

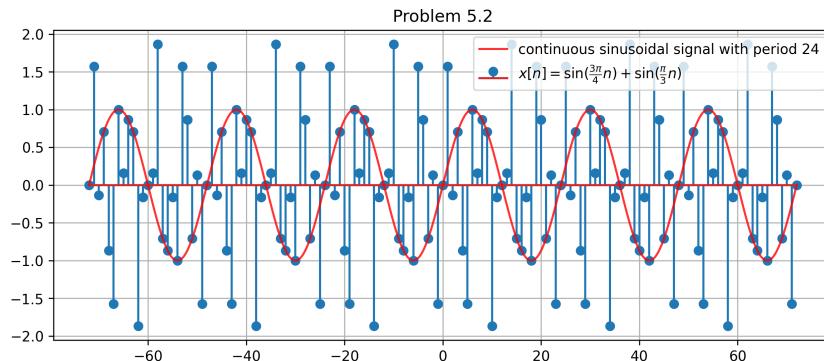
By using Python and Matplotlib, we can visualize the periodicity of the signal:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(10, 4))
5
6 t = np.arange(-72, 73, 1)
7 x = np.sin(3 * np.pi/4 * t) + np.sin(np.pi/3 * t)
8
9 tc = np.arange(-72, 72, 0.01)
10 xc = np.sin(np.pi/12 * tc)
11
12 plt.title("Problem 5.2")
13 plt.stem(t, x, label=r"$x[n] = \sin(\frac{3\pi}{4}n) + \sin(\frac{\pi}{3}n)$")
14 plt.plot(tc, xc, "r", alpha=0.8, label="continuous sinusoidal signal with period 24")
15 plt.grid(True)
16 plt.legend(loc="upper right")
17 plt.show()

```

The plot of the signal is shown below:



c)  $x[n] = \sin\left(\frac{3\pi n}{4}\right) \sin\left(\frac{\pi}{3}n\right)$

**Solution.** Using the product-to-sum identities, we can rewrite the signal as:

$$x[n] = \sin\left(\frac{3\pi n}{4}\right) \sin\left(\frac{\pi}{3}n\right) = \frac{1}{2} \left[ \cos\left(\frac{3\pi n}{4} - \frac{\pi n}{3}\right) - \cos\left(\frac{3\pi n}{4} + \frac{\pi n}{3}\right) \right]$$

Consider the signal:

$$\cos\left(\frac{3\pi n}{4} - \frac{\pi n}{3}\right) = \cos\left(\frac{5\pi n}{12}\right) \text{ has a period of } N_1 = \frac{2\pi}{\frac{5\pi}{12}} = \frac{24}{5}$$

$$\cos\left(\frac{3\pi n}{4} + \frac{\pi n}{3}\right) = \cos\left(\frac{13\pi n}{12}\right) \text{ has a period of } N_2 = \frac{2\pi}{\frac{13\pi}{12}} = \frac{24}{13}$$

Considering the least common multiple of the two periods:

$$N = \text{lcm}(N_1, N_2) = \text{lcm}\left(\frac{24}{5}, \frac{24}{13}\right) = \frac{24}{\text{gcd}(5, 13)} = 24$$

Thus, the signal  $x[n] = \sin\left(\frac{3\pi n}{4}\right) \sin\left(\frac{\pi}{3}n\right)$  **is periodic** with a period of  $N = 24$ .

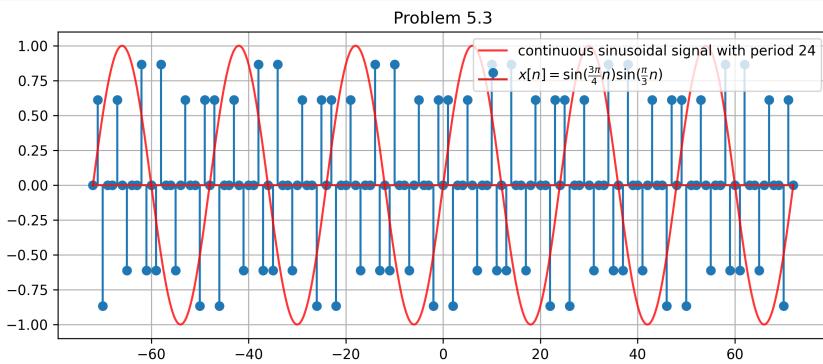
By using Python and Matplotlib, we can visualize the periodicity of the signal:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(10, 4))
5
6 t = np.arange(-72, 73, 1)
7 x = np.sin(3 * np.pi/4 * t) * np.sin(np.pi/3 * t)
8
9 tc = np.arange(-72, 72, 0.01)
10 xc = np.sin(np.pi/12 * tc)
11
12 plt.title("Problem 5.3")
13 plt.stem(t, x, label=r"$x[n] = \sin(\frac{3\pi}{4}n) \sin(\frac{\pi}{3}n)$")
14 plt.plot(tc, xc, "r", alpha=0.8, label="continuous sinusoidal signal with period 24")
15 plt.grid(True)
16 plt.legend(loc="upper right")
17 plt.show()

```

The plot of the signal is shown below:



d)  $x[n] = \exp\left(\frac{6\pi}{5}n\right)$

**Solution.**

Consider the signal, this is a exponential signal, which have not imaginary part, thus it **is not periodic**.

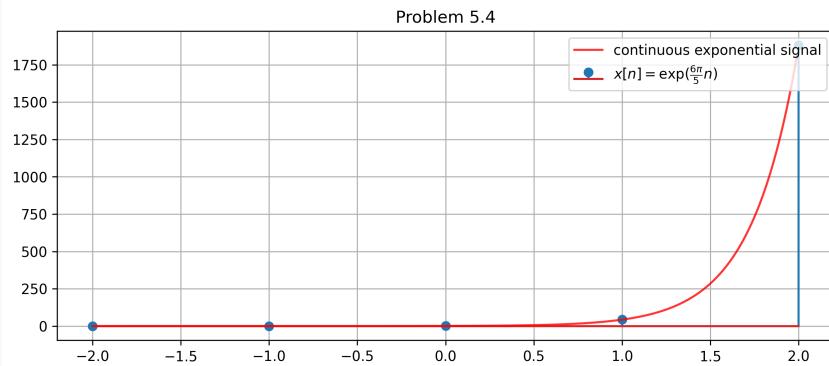
By using Python and Matplotlib, we can visualize the periodicity of the signal:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(10, 4))
5
6 t = np.arange(-2, 3, 1)
7 x = np.exp(6 * np.pi/5 * t)
8
9 tc = np.arange(-2, 2, 0.01)
10 xc = np.exp(6 * np.pi/5 * tc)
11
12 plt.title("Problem 5.4")
13 plt.stem(t, x, label=r"$x[n] = \exp(\frac{6\pi}{5}n)$")
14 plt.plot(tc, xc, "r", alpha=0.8, label="continuous
15 exponential signal")
16 plt.grid(True)
17 plt.legend(loc="upper right")
18 plt.show()

```

The plot of the signal is shown below:



e)  $x[n] = \exp(j\frac{5\pi}{6}n)$

**Solution.** Consider the signal:

$$\exp\left(j\frac{5\pi}{6}n\right) \text{ has a period of } N = \frac{2\pi}{\frac{5\pi}{6}} = \frac{12}{5}$$

We need to make the period an integer, thus we can multiply the period by 5:

$$N = 5 \cdot \frac{12}{5} = 12$$

Thus, the signal  $x[n] = \exp(j\frac{5\pi}{6}n)$  is periodic with a period of  $N = 12$ .

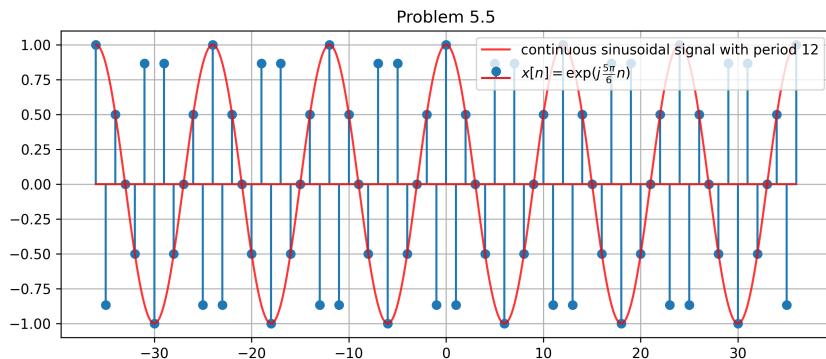
By using Python and Matplotlib, we can visualize the periodicity of the signal:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(10, 4))
5
6 t = np.arange(-36, 37, 1)
7 x = np.exp(1j * 5 * np.pi/6 * t)
8
9 tc = np.arange(-36, 36, 0.01)
10 xc = np.exp(1j * np.pi/6 * tc)
11
12 plt.title("Problem 5.5")
13 plt.stem(t, x, label=r"$x[n] = \exp(j\frac{5\pi}{6}n)$")
14 plt.plot(tc, xc, "r", alpha=0.8, label="continuous
15 sinusoidal signal with period 12")
16 plt.grid(True)
17 plt.legend(loc="upper right")
18 plt.show()

```

The plot of the signal is shown below:



f)  $x[n] = \sum_{m=-\infty}^{\infty} [\delta[n - 2m] + 2\delta[n - 3m]]$

**Solution.** Consider the signal, using the properties of the delta function:

1. The first term  $\sum_{m=-\infty}^{\infty} \delta[n - 2m]$  is a periodic signal with period of 2, since:

$$\sum_{m=-\infty}^{\infty} \delta[n - 2m] = \sum_{m=-\infty}^{\infty} \delta[n + 2 - 2m]$$

2. The second term  $2 \sum_{m=-\infty}^{\infty} \delta[n - 3m]$  is a periodic signal with period of 3, since:

$$2 \sum_{m=-\infty}^{\infty} \delta[n - 3m] = 2 \sum_{m=-\infty}^{\infty} \delta[n + 3 - 3m]$$

Thus, the overall signal  $x[n]$  **is periodic** with a fundamental period of  $\text{lcm}(2, 3) = 6$ .

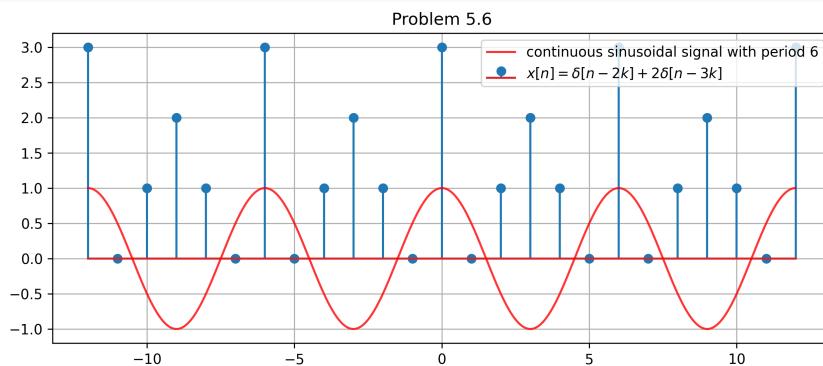
By using Python and Matplotlib, we can visualize the periodicity of the signal:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(10, 4))
5
6 t = np.arange(-12, 13, 1)
7 x1 = np.isin(np.mod(t, 2), [0]).astype(float) # delta[n-2k]
8 x2 = np.isin(np.mod(t, 3), [0]).astype(float) # delta[n-3k]
9
10 x = x1 + 2 * x2
11
12 tc = np.arange(-12, 12, 0.01)
13 xc = np.exp(1j * np.pi/3 * tc)
14
15 plt.title("Problem 5.6")
16 plt.stem(t, x, label=r"$x[n] = \delta[n-2k] + 2\delta[n-3k]$")
17 plt.plot(tc, xc, "r", alpha=0.8, label="continuous sinusoidal signal with period 6")
18 plt.grid(True)
19 plt.legend(loc="upper right")
20 plt.show()

```

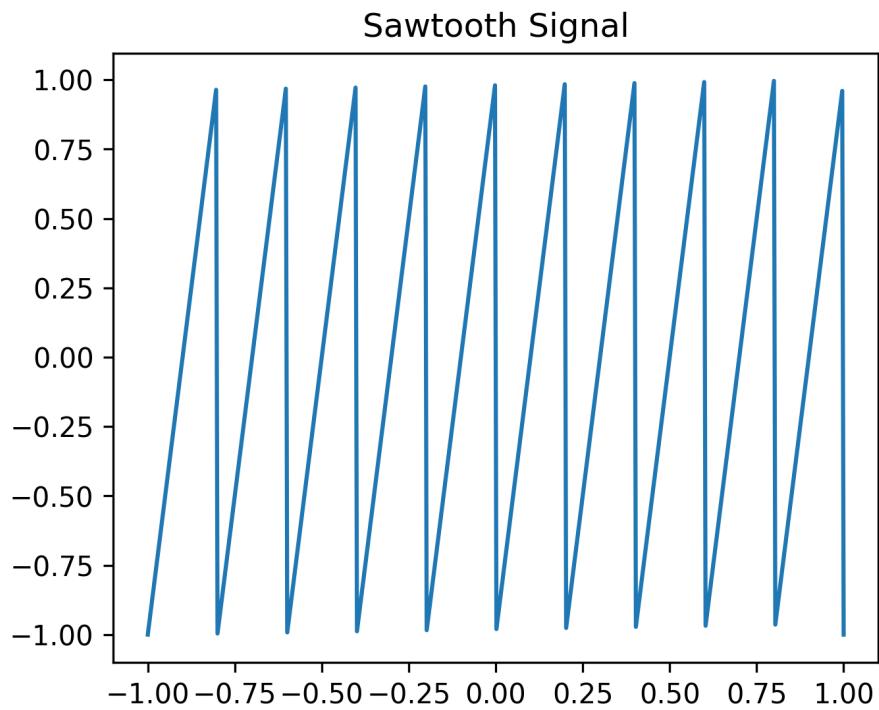
The plot of the signal is shown below:



**Problem 6. Signal transformations:** Study the sawtooth function in the figure below. Apply reflection, scaling, shifting operations to the signal and plot the transformed signals compared with the original sawtooth signal.

```
1 import numpy as np
2 from scipy import signal
3
4 fig = plt.figure(figsize=(5, 4))
5
6 t = np.linspace(-1, 1, 500)
7 sawtooth = signal.sawtooth(2 * np.pi * 5 * t)
8
9 plt.title("Sawtooth Signal")
10 plt.plot(t, sawtooth)
11 plt.show()
```

The plot of the signal is shown below:



- a) time scaling: scaling factor = 3 and  $1/3$

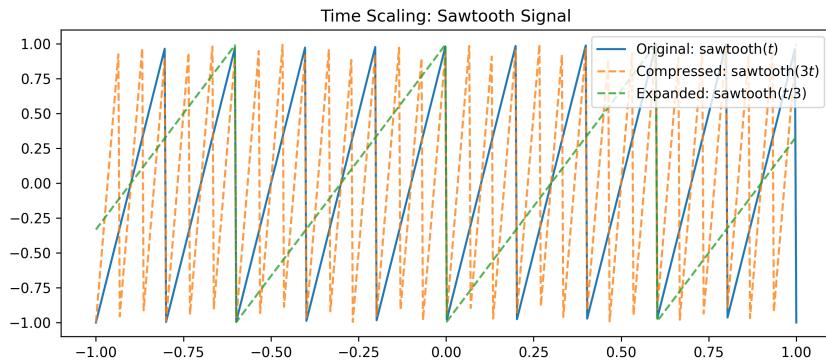
Solution. Using Python and Matplotlib to plot the time-scaled signals:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3 from scipy import signal
4
5 fig = plt.figure(figsize=(10, 4))
6
7 t = np.linspace(-1, 1, 500)
8 sawtooth = signal.sawtooth(2 * np.pi * 5 * t)
9
10 # scaling factor = 3 and 1/3
11 ## TODO : writing code for time scaling
12
13 t1 = t * 3
14 sawtooth1 = signal.sawtooth(2 * np.pi * 5 * t1)
15
16 t2 = t * (1/3)
17 sawtooth2 = signal.sawtooth(2 * np.pi * 5 * t2)
18
19 plt.title("Time Scaling: Sawtooth Signal")
20 plt.plot(t, sawtooth, label=r"Original: sawtooth$(t)$",
21           alpha=1)
21 plt.plot(t, sawtooth1, label=r"Compressed: sawtooth$(3t)$",
22           linestyle="--", alpha=0.8)
22 plt.plot(t, sawtooth2, label=r"Expanded: sawtooth$(t/3)$",
23           linestyle="--", alpha=0.8)
23 plt.legend(loc="upper right")
24 plt.show()

```

With the resulting plot shown below:



b) time shifting: shifting amount =  $\pm 0.05$

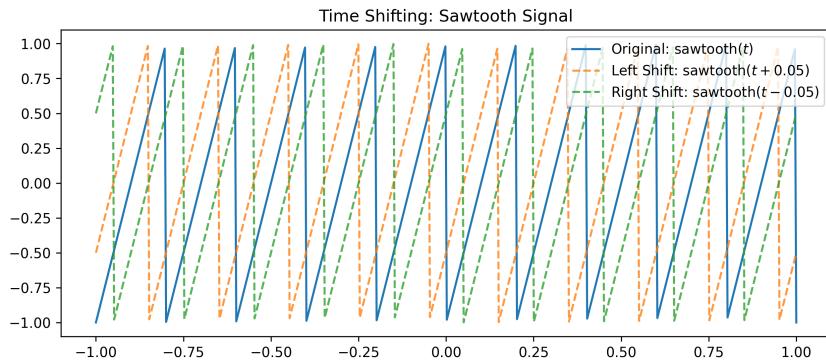
Solution. Using Python and Matplotlib to plot the time-shifted signals:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3 from scipy import signal
4
5 fig = plt.figure(figsize=(10, 4))
6
7 t = np.linspace(-1, 1, 500)
8 sawtooth = signal.sawtooth(2 * np.pi * 5 * t)
9
10 ## TODO : writing code for time shifting
11 # shifting t to the left and right 0.05 units
12
13 t1 = t + 0.05
14 sawtooth1 = signal.sawtooth(2 * np.pi * 5 * t1)
15
16 t2 = t - 0.05
17 sawtooth2 = signal.sawtooth(2 * np.pi * 5 * t2)
18
19 plt.title("Time Shifting: Sawtooth Signal")
20 plt.plot(t, sawtooth, label=r"Original: sawtooth$(t)$",
21           alpha=1)
21 plt.plot(t, sawtooth1, label=r"Left Shift: sawtooth$(t + 0.05)$",
22           linestyle="--", alpha=0.8)
22 plt.plot(t, sawtooth2, label=r"Right Shift: sawtooth$(t - 0.05)$",
23           linestyle="--", alpha=0.8)
23 plt.legend(loc="upper right")
24 plt.show()

```

With the resulting plot shown below:



c) **time reflection:** reflecting over the y-axis

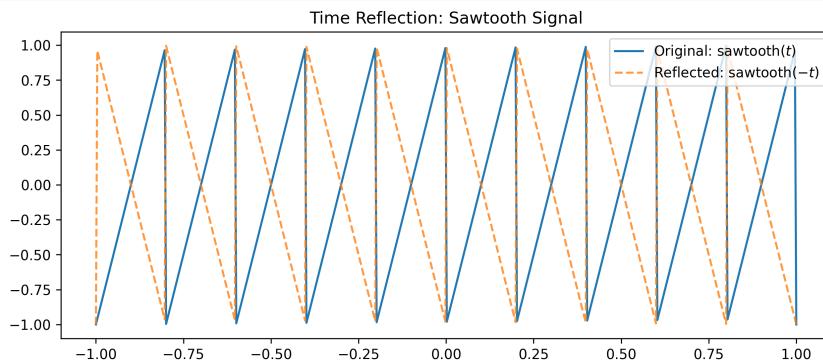
**Solution.** Using Python and Matplotlib to plot the time-reflected signals:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3 from scipy import signal
4
5 fig = plt.figure(figsize=(10, 4))
6
7 t = np.linspace(-1, 1, 500)
8 sawtooth = signal.sawtooth(2 * np.pi * 5 * t)
9
10 ## TODO : writing code for time Reflection
11
12 t1 = t
13 sawtooth1 = signal.sawtooth(2 * np.pi * 5 * (-1 * t))
14
15 plt.title("Time Reflection: Sawtooth Signal")
16 plt.plot(t, sawtooth, label=r"Original: sawtooth$(t)$",
17           alpha=1)
18 plt.plot(t, sawtooth1, label=r"Reflected: sawtooth$(-t)$",
19           linestyle="--", alpha=0.8)
20 plt.legend(loc="upper right")
21
22 plt.savefig("../images/problem_6_3.png", dpi=300,
23             bbox_inches="tight")
24
25 plt.show()

```

With the resulting plot shown below:



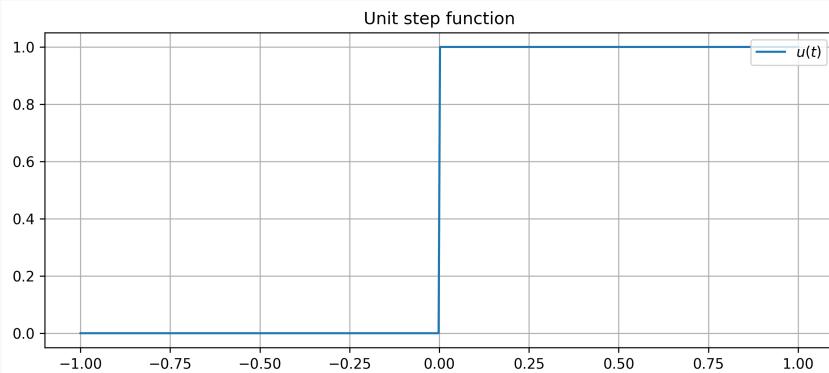
**Problem 7. Elementary signals:** study the ramp signal plotted in the example below. Plot these signals.

a) Writing code for plotting **Unit step function**

Solution. Using Python and Matplotlib to plot the unit step function:

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(10, 4))
5
6 t = np.linspace(-1, 1, 500)
7
8 ## TODO : writing code for plotting unit step function
9
10 def unit_signal(t):
11     return 1.0 if t >= 0 else 0.0
12
13 unit_signal_vectorize = np.vectorize(unit_signal)
14
15 x = unit_signal_vectorize(t)
16
17 plt.title("Unit step function")
18 plt.plot(t, x, label=r"$u(t)$", alpha=1)
19 plt.grid(True)
20 plt.legend(loc="upper right")
21 plt.show()
```

With the resulting plot shown below:

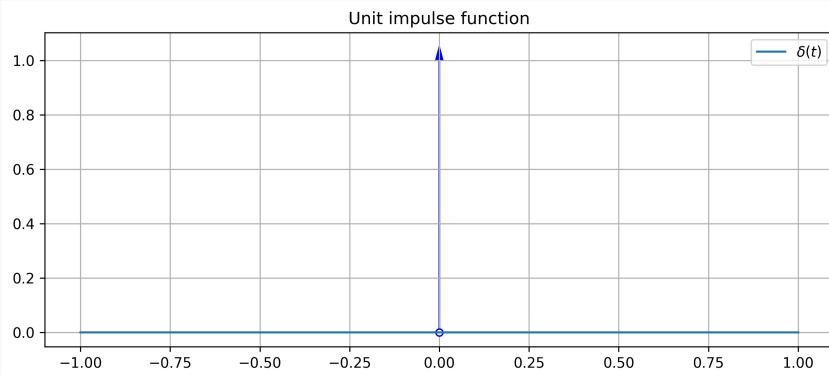


**b) Writing code for plotting Unit impulse function**

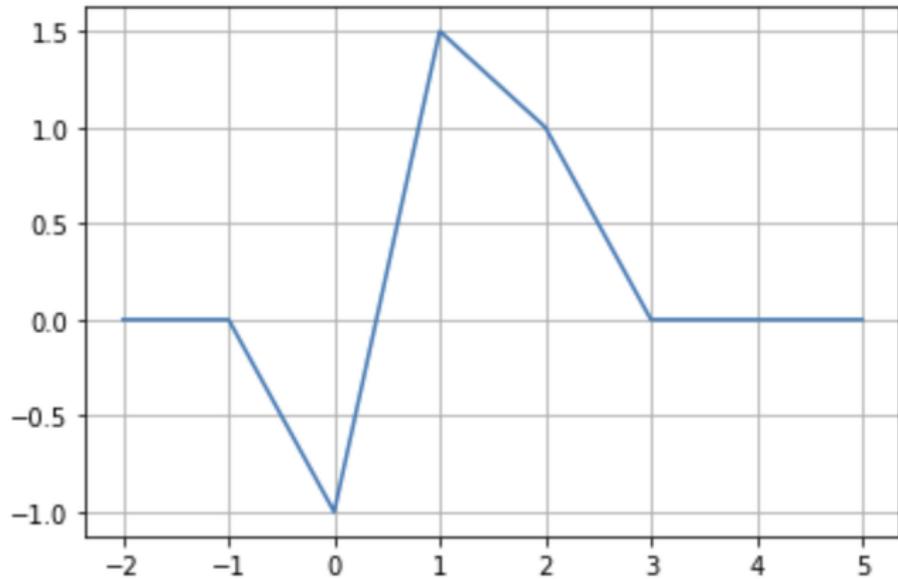
Solution. Using Python and Matplotlib to plot the unit impulse function:

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(10, 4))
5
6 t = np.linspace(-1, 1, 500)
7
8 ## TODO : writing code for plotting unit impulse function
9
10 x = np.zeros(t.shape[0])
11 x[t == 0] = 1
12
13 arrow = plt.arrow(0, 0, 0, 1, head_width=0.02,
14     head_length=0.05, fc="blue", ec="blue")
15 origin = plt.plot(0, 0, color="blue", marker = "o",
16     markerfacecolor="white", markersize=5)
17
18 plt.title("Unit impulse function")
19 plt.plot(t, x, label=r"$\delta(t)$", alpha=1)
20 plt.grid(True)
21 plt.legend(loc="upper right")
22 plt.show()
```

With the resulting plot shown below:



**Problem 8.** Express the signal that shown in Figure below using Unit-ramp functions



**Solution.** The signal can be expressed using unit-ramp functions as follows:

$$-r(t+1) + 3.5r(t) - 3r(t-1) - 0.5r(t-2) + r(t-3)$$

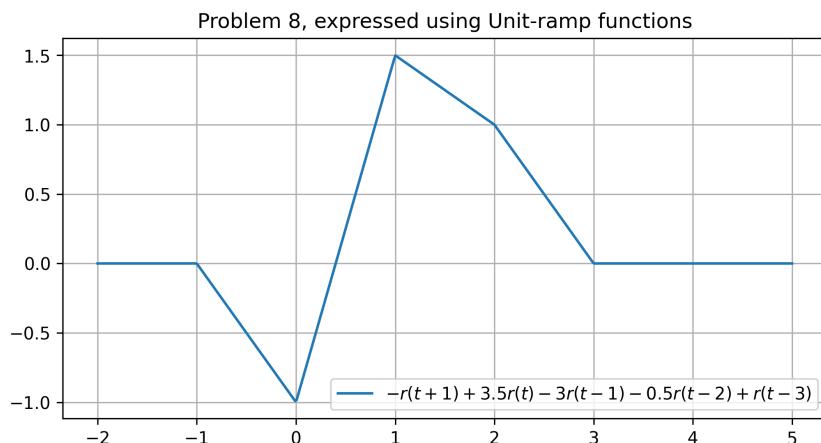
Using Python and Matplotlib to plot this unit-ramp function:

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 fig = plt.figure(figsize=(5, 4))
5
6 t = np.linspace(-2, 5, 500)
7
8 x = np.zeros(t.shape[0])
9
10 x += ramp_signal_vectorize(t + 1) * (-1)           # -r(
11     t+1)
12 x += ramp_signal_vectorize(t - 0) * (1 + 2.5)       # 3.5
13     r(t)
14 x += ramp_signal_vectorize(t - 1) * (-2.5 + -0.5)   # -3r
15     (t-1)
16 x += ramp_signal_vectorize(t - 2) * (0.5 + -1)      # -
17     -0.5r(t-2)
18 x += ramp_signal_vectorize(t - 3) * (1)              # r(t
19     -3)
20
21 plt.title("Problem 8, expressed using Unit-ramp functions")
22 plt.plot(t, x, label=r"$-r(t+1)+3.5r(t)-3r(t-1)-0.5r(t-2)$
23           +r(t-3)$")
24 plt.legend(loc="lower right")
25 plt.grid(True)
26 plt.show()

```

With the resulting plot shown below:



**Problem 9.** Evaluate the following integrals

TO SUBMIT

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

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

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

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

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

$$\begin{aligned} \int_{-\infty}^{\infty} (t-1) \delta\left(\frac{2}{3}t - \frac{3}{2}\right) dt &= \int_{-\infty}^{\infty} (t-1) \delta\left(t - \frac{9}{4}\right) \cdot \left| \frac{d}{dt} \left( \frac{2}{3}t - \frac{3}{2} \right) \right|^{-1} dt \\ &= \int_{-\infty}^{\infty} (t-1) \delta\left(t - \frac{9}{4}\right) \cdot \frac{3}{2} dt \\ &= \left(\frac{9}{4} - 1\right) \cdot \frac{3}{2} \\ \int_{-\infty}^{\infty} (t-1) \delta\left(\frac{2}{3}t - \frac{3}{2}\right) dt &= \frac{5}{4} \cdot \frac{3}{2} = \boxed{\frac{15}{8}} \end{aligned}$$

## TO SUBMIT

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

**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:

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

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

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

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