

Prashant Kumar Aleshwaram<sup>†</sup>, Mohammed Zafar Ali Khan\* and G V V Sharma\*

Problems 1-5 are related.

**Problem 1.** Let

$$x(n) = \left\{ 0, \underset{\uparrow}{2}, 3, 1 \right\} \quad (1.1)$$

Sketch  $x(n)$

**Solution:** The following code yields Fig. 1.

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

n=np.linspace(-3,7,11)
x=np.array
    ([0,0,0,0,2,3,1,0,0,0,0])
plt.stem(n,x)
plt.xlabel('$n$')
plt.ylabel('$x(n)$')
plt.ylim((0,3))
#Ignore the following command
plt.savefig(' ../ figs/3 a. eps ')
plt.show()
```

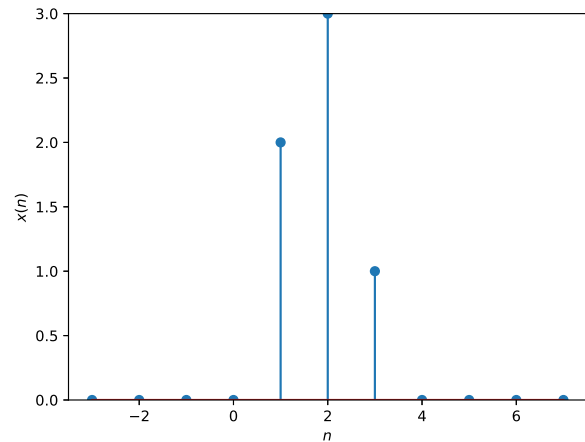


Fig. 1

**Problem 2.** Is  $x(n)$

- 1) An energy signal?
- 2) A power signal?
- 3) A causal signal?
- 4) A periodic signal?

**Problem 3.** Sketch  $3x(n)$ .

**Solution:** The following code yields Fig. 3.

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

\*The authors are with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: {zafar,gadepall}@iith.ac.in.† The author is an intern with the Teaching Learning Centre, IIT Hyderabad email: akumar10@student.nitw.ac.in. All content in the manuscript is released under GNU GPL. Free to use for anything.

```
n=np.linspace(-3,7,11)
x=np.array
    ([0,0,0,0,2,3,1,0,0,0,0])
plt.stem(n,3*x)
plt.xlabel('$n$')
plt.ylabel('$3x(n)$')
plt.ylim((0,10))
#Ignore the following command
plt.savefig(' ../ figs/3 b. eps ')
plt.show()
```

**Problem 4.** Sketch  $x(n-2)$ .

**Solution:** The following code yields Fig. 4.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.ndimage.interpolation
    import shift
```

```
n=np.linspace(-3,7,11)
#x(n)
```

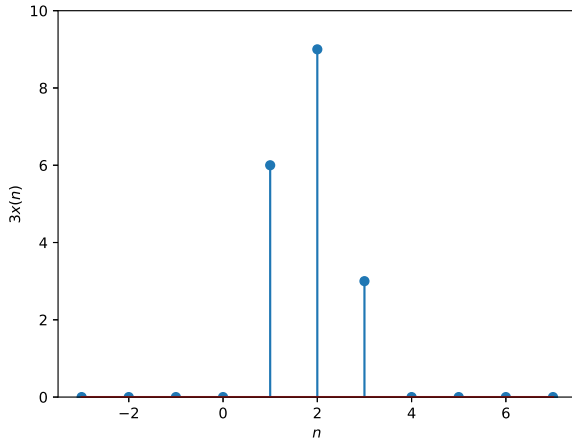


Fig. 3

```
xn=np.array
    ([0,0,0,0,2,3,1,0,0,0,0])
#x(n-2)
xn_right=shift(xn, 2, cval=0)

plt.stem(n,xn_right)
plt.xlabel('$n$')
plt.ylabel('$x(n-2)$')
plt.ylim((0,4))
#Ignore the following command
plt.savefig('../figs/3c.eps')
plt.show()
```

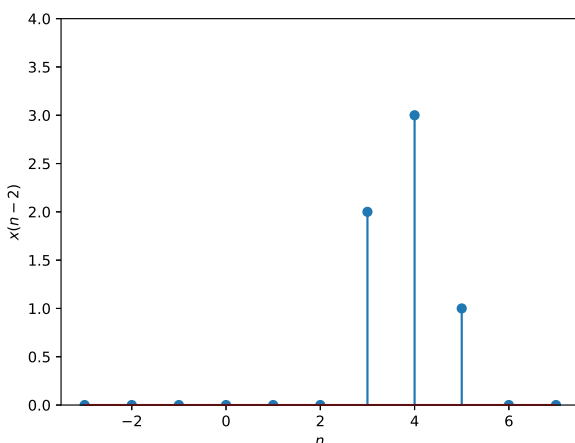


Fig. 4

**Problem 5.** Sketch  $x(3-n)$ .

**Solution:** The following code yields Fig. 5.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.ndimage.interpolation
    import shift

#x(n)
xn=np.array([0,0,0,0,2,3,1])
#x(-n)
xflip = np.flip(xn,0)
#x(3-n)
xflip_right=shift(xflip, 3, cval
    =0)
#plotting
n=np.linspace(-3,3,7)
plt.stem(n,xflip_right)
plt.xlabel('$n$')
plt.ylabel('$x(3-n)$')
plt.ylim((0,4))
##Ignore the following command
plt.savefig('../figs/3d.eps')
plt.show()
```

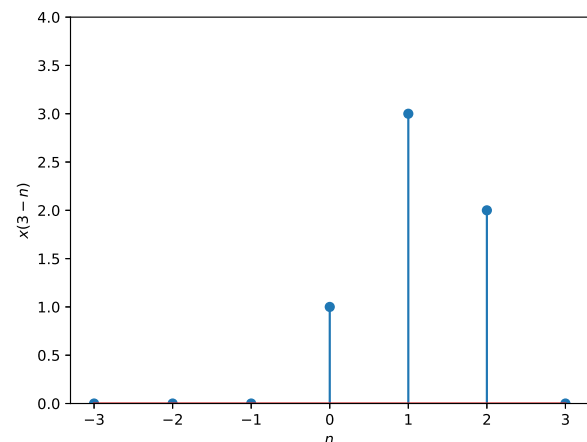


Fig. 5

**Problem 6.** Sketch

$$x(t) = \begin{cases} 4 - |t| & 1 \leq |t| \leq 4 \\ 2 + |t| & |t| < 1 \end{cases} \quad (6.1)$$

and find out if its even or odd.

**Solution:** The following code yields Fig. 6. As we can see,  $x(t) = x(-t)$  and the function is even.

```

import numpy as np
import matplotlib.pyplot as plt

# |t| < 1
t1=np.linspace(-1,1,25)
x1=2+abs(t1)

# 1 < |t| < 4
t2=np.linspace(1,4,25)
x2=-abs(t2)+4
t3=np.linspace(-4,-1,25)
x3=(t3)+4

#x(t)
x=np.concatenate((x3,x1,x2), axis
= 0)

t = np.concatenate((t3,t1,t2),
axis = 0)

#x(-t)
xflip = np.fliplr([x])[0]

#Plotting
plt.plot(t,x, label='$x(t)$')
plt.plot(t,xflip,'o', mfc='none',
label='$x(-t)$')

plt.grid()
plt.xlabel('$t$')
plt.ylabel('$x(t)$')
plt.legend()
##Ignore the following command
plt.savefig('../figs/4a.eps')
plt.show()

```

### Problem 7. Sketch

$$x(t) = \begin{cases} t+4 & -4 \leq t \leq -1 \\ -5t-2 & -1 < t < 0 \\ \frac{2}{3}(t-3) & 0 \leq t \leq 3 \\ 0 & \text{otherwise} \end{cases} \quad (7.1)$$

and find out if its even or odd.

**Solution:** The following code yields Fig. 7. As we can see, the function is neither even nor odd.

```

import numpy as np

```

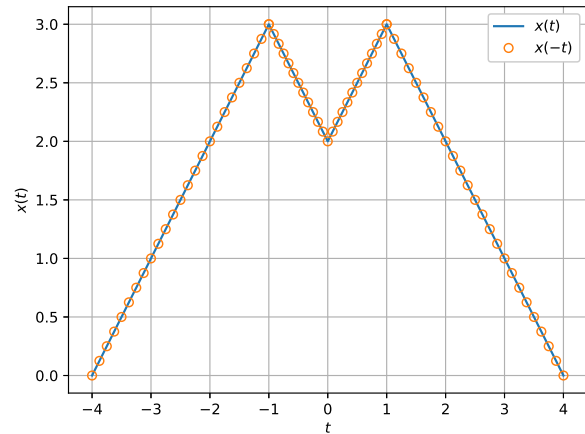


Fig. 6

```

import matplotlib.pyplot as plt

#-4 < t < -1
t3=np.linspace(-4,-1,60)
x3=t3+4

#-1 < t < 0
t1=np.linspace(-1,0,20)
x1=-5*t1-2

#0 < t < 3
t2=np.linspace(0,3,60)
x2=(2.0/3)*(t2-3)

#3 < t < 4
t4=np.linspace(3,4,20)
z=np.zeros(20)

#x(t)
x=np.concatenate((x3,x1,x2,z),
axis = 0)

t = np.concatenate((t3,t1,t2,t4),
axis = 0)

#x(-t)
xflip = np.fliplr([x])[0]

#Plotting
plt.plot(t,x, label='$x(t)$')
plt.plot(t,xflip, label='$x(-t)$')

```

```
plt.grid()
plt.xlabel('$t$')
plt.ylabel('$x(t)$')
plt.legend()
plt.axis('equal')
##Ignore the following command
plt.savefig('../figs/4b.eps')
plt.show()
```

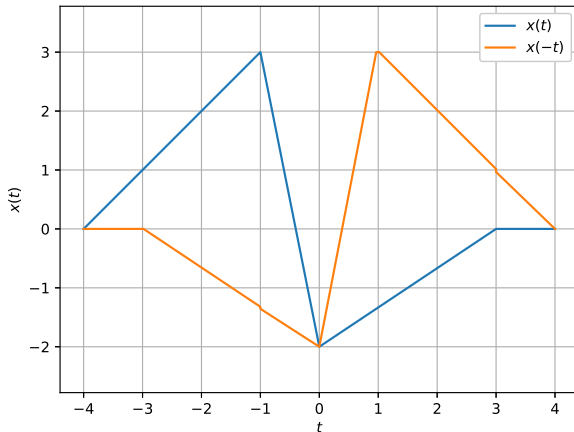


Fig. 7

**Problem 8.** Sketch

$$x(t) = \begin{cases} 3(t+4) & -4 \leq t \leq -3 \\ -5t - 12 & -3 < t < -2 \\ 3t + 4 & -2 \leq t \leq -1 \\ -t & -1 < t < 1 \\ 3t - 4 & 1 \leq t \leq 2 \\ -5t + 12 & 2 < t < 3 \\ 3(t-4) & 3 \leq t \leq 4 \\ 0 & \text{otherwise} \end{cases} \quad (8.1)$$

and find out if its even or odd.

**Solution:** The following code yields Fig. 8. As we can see,  $x(t) = -x(-t)$  and the function is odd.

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

```
#-4 < t < -3
t7=np.linspace(-4,-3,25)
x7=(t7+4)*3
```

```
#-3 < t < -2
```

```
t5=np.linspace(-3,-2,25)
x5=(-5*t5)-12
```

```
# -2 < t < -1
t4=np.linspace(-2,-1,25)
x4=3*(t4)+4
```

```
#-1 < t < 1
t1=np.linspace(-1,1,25)
x1=-t1
```

```
#1 < t < 2
t2=np.linspace(1,2,25)
x2=3*(t2)-4
```

```
#2 < t < 3
t3=np.linspace(2,3,25)
x3=(-5*t3)+12
```

```
#3 < t < 4
t6=np.linspace(3,4,25)
x6=(t6-4)*3
```

```
#x(t)
x=np.concatenate((x7,x5,x4,x1,x2,
x3,x6), axis = 0)
```

```
t = np.concatenate((t7,t5,t4,t1,t2,
t3,t6), axis = 0)
```

```
#x(-t)
xflip =np.fliplr([x])[0]
```

```
#Plotting
plt.plot(t,x, label='$x(t)$')
plt.plot(t,xflip,'o', mfc='none',
label='$x(-t)$')
```

```
plt.grid()
plt.xlabel('$t$')
plt.ylabel('$x(t)$ and $x(-t)$')
plt.legend()
##Ignore the following command
plt.savefig('../figs/4c.eps')
plt.show()
```

Problems 9-13 are related.

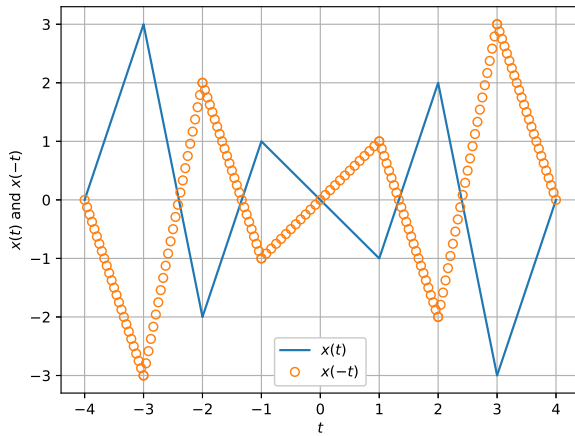


Fig. 8

**Problem 9. Sketch**

$$x(t) = \begin{cases} t+4 & -4 \leq t \leq -2 \\ t-4 & 2 \leq t \leq 4 \\ 0 & \text{otherwise} \end{cases} \quad (9.1)$$

**Solution:** The following code yields Fig. 9

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

t1=np.linspace(-6,-4,10)
t2=np.linspace(-4,-2,10)
t3=np.linspace(-2,2,10)
t4=np.linspace(2,4,10)
t5=np.linspace(4,6,10)

x1=t1*0
x2=t2+4
x3=t3*0
x4=t4-4
x5=t5*0

#x(t)
x=np.concatenate((x1,x2,x3,x4,x5),
                 axis=0)

t=np.concatenate((t1,t2,t3,t4,t5),
                 axis=0)

plt.plot(t,x)
plt.xlabel('t')
```

```
plt.ylabel('x(t)')
plt.grid()
#Ignore the following command
plt.savefig('../figs/8.eps')
plt.show()
```

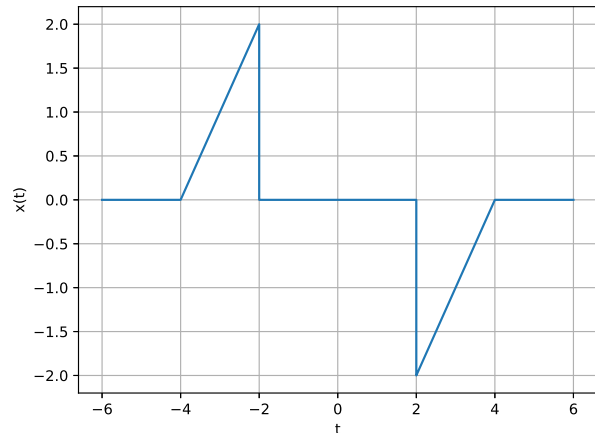


Fig. 9

**Problem 10. Sketch  $x(t+1)$** 

**Solution:** The following code yields Fig. 10

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.ndimage.interpolation
import shift

t1=np.linspace(-6,-4,40)
t2=np.linspace(-4,-2,40)
t3=np.linspace(-2,2,80)
t4=np.linspace(2,4,40)
t5=np.linspace(4,6,40)

x2=t2+4
x3=t3*0
x4=t4-4
x1=t1*0

x=np.concatenate((x1,x2,x3,x4,x1),
                 axis=0)

t=np.concatenate((t1,t2,t3,t4,t5),
                 axis=0)

# 0-1 x(t) has 10 samples i.e., x(
t+1)=x(n-20)
```

```

andig = 20
anshift = -1

x_t=shift(x, anshift*andig, cval=0)

plt.plot(t,x,label='x(t)')
plt.plot(t,x_t,label='x(t+1)')
plt.xlabel('t')
plt.ylabel('$x(t)$ and $x(t+1)$')
plt.axis([-6,6,-3,3])
plt.grid()
plt.legend()
#Ignore the following command
plt.savefig('../figs/8a.eps')
plt.show()

```

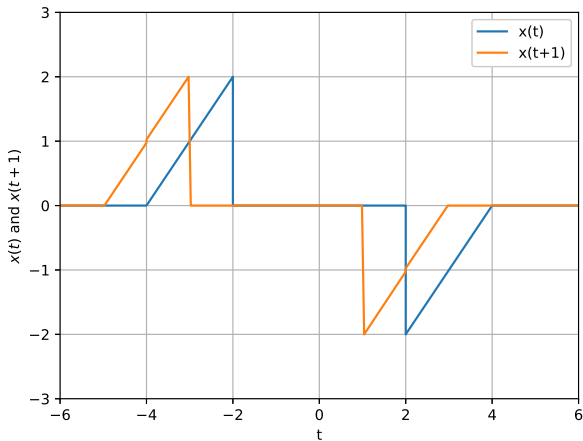


Fig. 10

**Problem 11.** Sketch  $5x\left(\frac{t}{3}\right)$

**Problem 12.** Sketch  $2x(t-2)$

**Problem 13.** Sketch  $x(t) - x(-t)$ .

Problems 14-16 are related.

**Problem 14.** Let  $x(t) = e^{-t}$  and  $y(t) = x(t)u(t)$ . Sketch  $x(t)$  and  $y(t)$ .

**Problem 15.** Sketch  $x_1(t) = x(t-2)$  and  $y_1(t)$ , where  $y_1(t)$  is the output when  $x_1(t)$  is the input to the system.

**Problem 16.** Sketch  $y(t-2)$  and  $y_1(t)$  in the same graph. Is the system time invariant?

Problems 17-18 are related.

**Problem 17.** If  $x(t) = u(t)$ , sketch

$$y(t) = \sum_{m=-1}^2 mx(t-m) \quad (17.1)$$

**Problem 18.** Sketch  $h(t)$  for the previous problem. Is the system causal?

Problems 19-23 are related.

**Problem 19.** Sketch

$$x(k) = \left\{ -1, -1, -1, 1, -1, 1, 1, 1 \right\} \quad (19.1)$$

and

$$h(k) = \left\{ 0, -1, 1 \right\} \quad (19.2)$$

**Problem 20.** Sketch  $h(-k)$  and  $x(k)h(n-k)$  for  $n = -2, -1, \dots, 7$ .

**Problem 21.** Sketch  $y(n) = \sum_k x(k)h(n-k)$  for  $n = -2, -1, \dots, 7$ .

**Problem 22.** Run the following code and compare with the result of the previous problem.

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.ndimage.interpolation
import shift

```

```

#x(n-2)
x=[-1,-1,-1,1,-1,1,1,1]
#h(n)
h=[0,-1,1]
#y(n-2)
y= np.convolve(x,h)
n = np.linspace(-2,7,10)
plt.stem(n,y)
#plt.stem(y)
plt.xlabel('n')
plt.ylabel('y(n)=x(n)*h(n)')
plt.grid()

plt.show()

```

**Problem 23.** Sketch  $h(n) * h(-n)$ .