

رسم سی > 13.py > ...

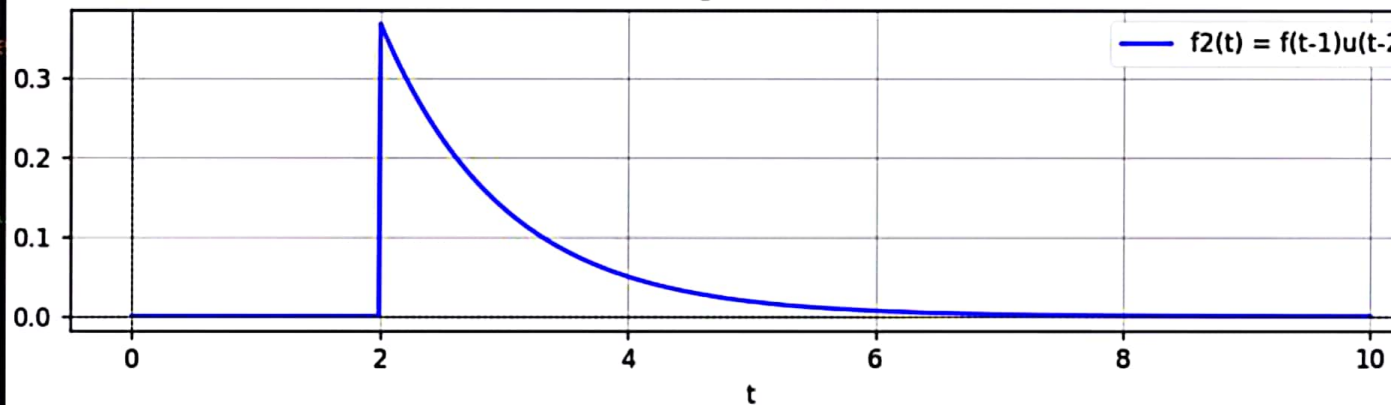
```
plt.plot(t, f2_t, label='f2(t) = f(t-1)u(t-2)', color='blue')
plt.xlabel('t')
plt.ylabel('f2(t)')
plt.title('Signal f2(t)')
plt.axhline(0, color='black', lw=0.5, ls='--') # افقی صفر
plt.axvline(0, color='black', lw=0.5, ls='--') # عمودی صفر
plt.grid(True)
plt.legend()

# رسم مشتق f2(t)
plt.subplot(3, 1, 2)
plt.plot(t, f2_derivative, label="f2'(t)", color='orange')
plt.xlabel('t')
plt.ylabel("f2'(t)")
plt.title('Derivative of f2(t)')
plt.axhline(0, color='black', lw=0.5, ls='--') # افقی صفر
plt.axvline(0, color='black', lw=0.5, ls='--') # عمودی صفر
plt.grid(True)
plt.legend()

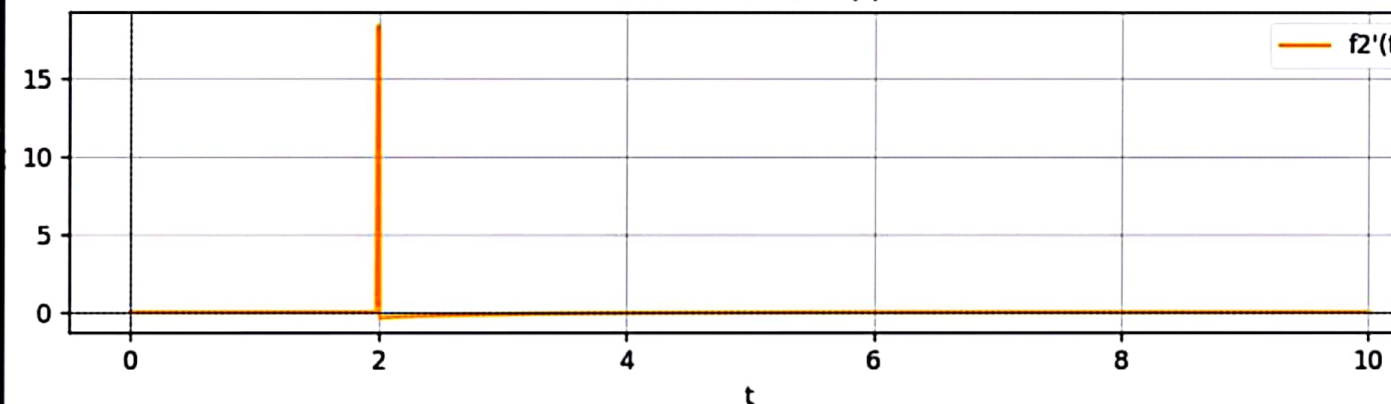
# رسم انتگرال f2(t)
plt.subplot(3, 1, 3)
plt.plot(t, f2_integral, label='∫f2(t) dt', color='green')
plt.xlabel('t')
plt.ylabel('∫f2(t) dt')
plt.title('Integral of f2(t)')
plt.axhline(0, color='black', lw=0.5, ls='--') # افقی صفر
plt.axvline(0, color='black', lw=0.5, ls='--') # عمودی صفر
plt.grid(True)
plt.legend()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

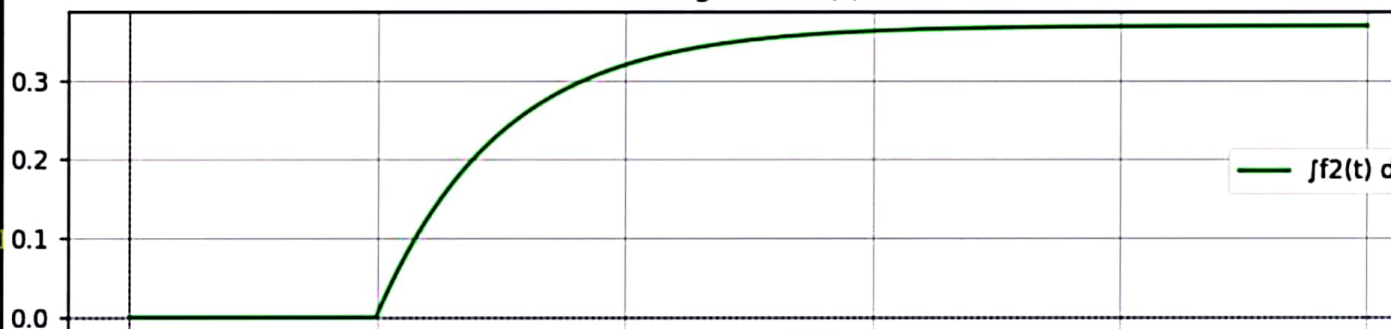
E:\university\انگیزش ل\موسیقی & C:/Users/Mr.Sani/AppData/Local

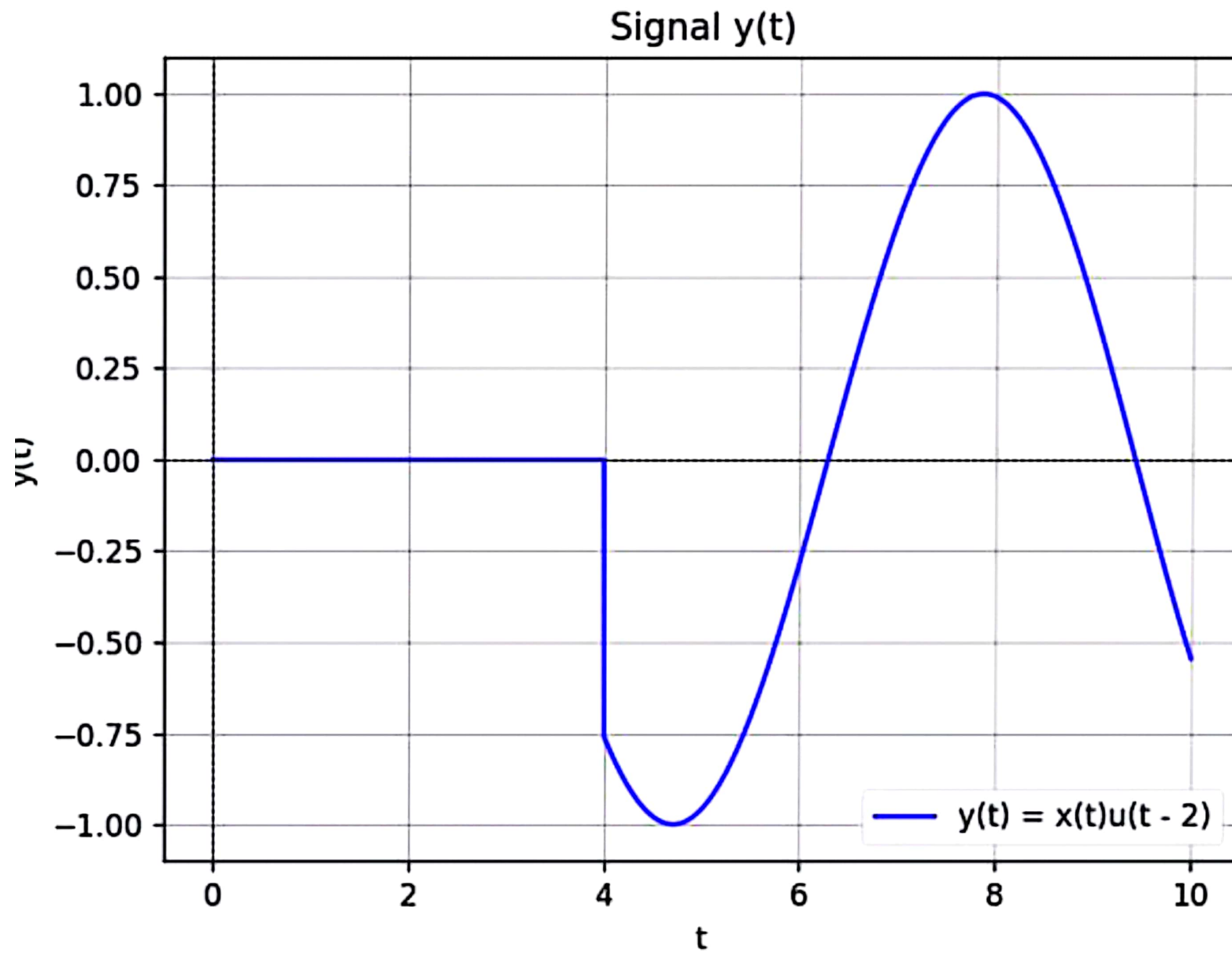


Derivative of f2(t)



Integral of f2(t)



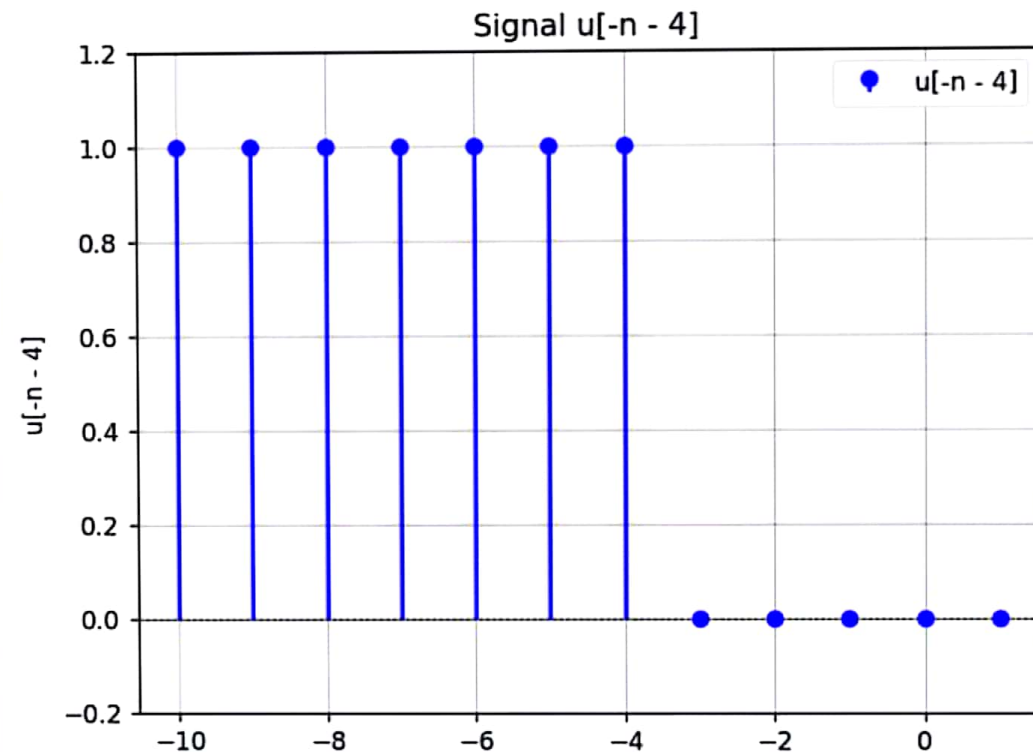


```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # مجور گسسته n تعریف دامنه
5 n = np.arange(-10, 2, 1) # 1 تا 10- از
6
7 # تعریف تابع واحد گسسته u[n]
8 def u(n):
9     return np.where(n >= 0, 1, 0)
10
11 # تعریف سیگنال u[-n - 4]
12 x_n = u(-n - 4)
13
14 # رسم سیگنال
15 plt.figure()
16 plt.stem(n, x_n, basefmt=" ", linefmt='b-', markerfmt='bo', label='u[-n - 4]')
17 plt.xlabel('n')
18 plt.ylabel('u[-n - 4]')
19 plt.title('Signal u[-n - 4]')
20 plt.axhline(0, color='black', lw=0.5, ls='--') # خط افقی صفر
21 plt.grid(True)
22 plt.ylim(-0.2, 1.2) # تنظیم محدوده مجور عمودی
23 plt.legend()
24 plt.show()
25

```

Figure 1



```

# محور زمان t تعریف دامنه
t = np.linspace(-1, 3, 1000) # از -1 تا 3 با 1000 نقطه

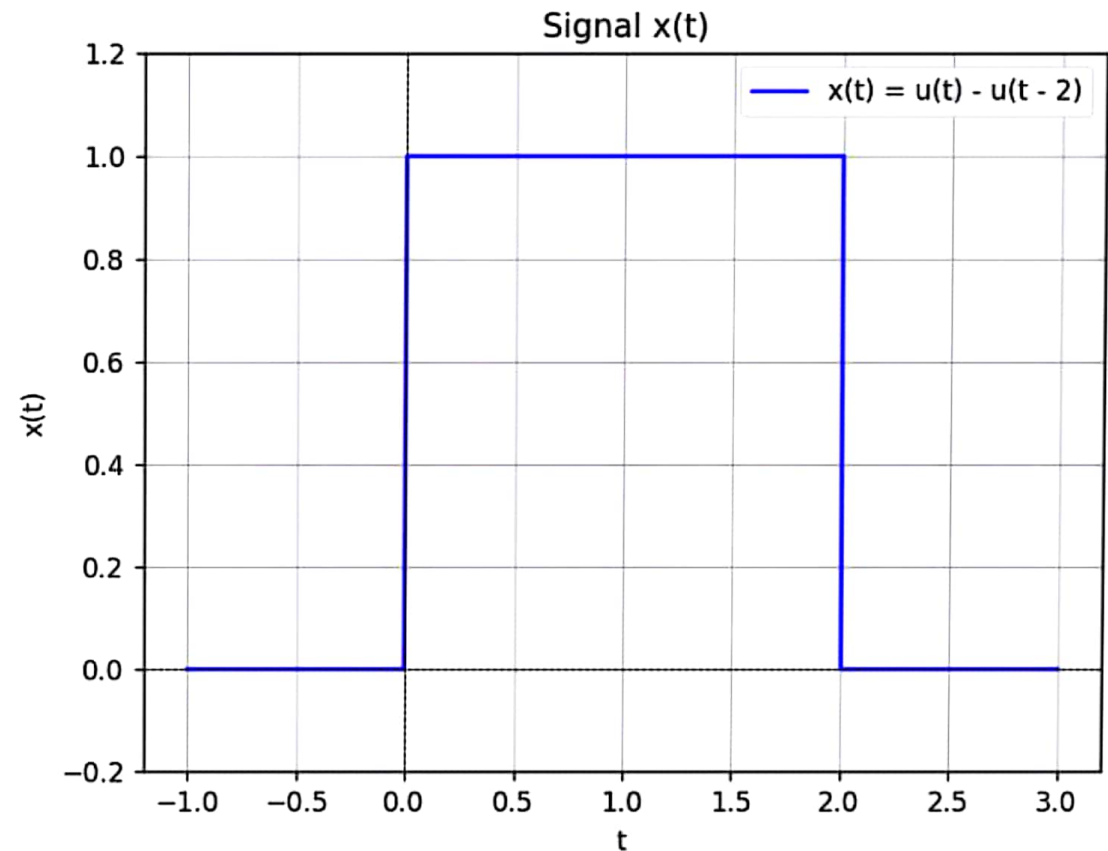
# تعریف تابع واحد u(t)
def u(t):
    return np.where(t >= 0, 1, 0)

# تعریف سیگنال x(t) = u(t) - u(t - 2)
x_t = u(t) - u(t - 2)

# رسم سیگنال
plt.figure()
plt.plot(t, x_t, label='x(t) = u(t) - u(t - 2)', color='blue')
plt.xlabel('t')
plt.ylabel('x(t)')
plt.title('Signal x(t)')
plt.axhline(0, color='black', lw=0.5, ls='--') # خط افقی صفر
plt.axvline(0, color='black', lw=0.5, ls='--') # خط عمودی صفر
plt.grid(True)
plt.ylim(-0.2, 1.2) # تنظیم محدوده محور عمودی
plt.legend()
plt.show()

```

Figure 1



(x, y) = (-0.973, -0.0



```

import matplotlib.pyplot as plt

# محور گسسته n تعریف دامنه
n = np.arange(0, 6, 1) # 0 تا 5

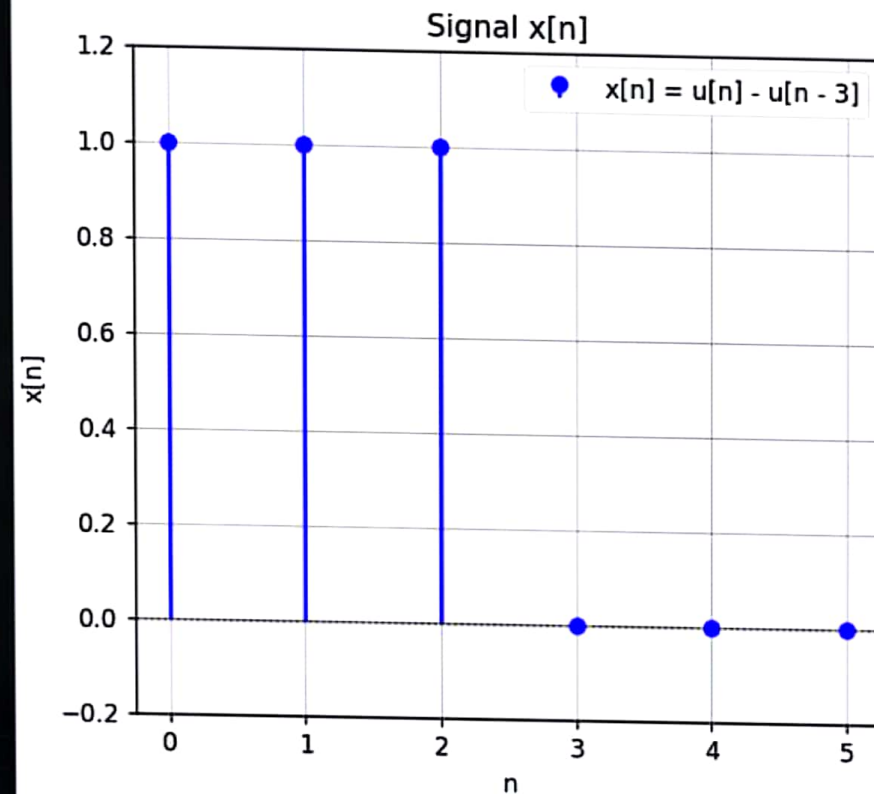
# u[n] تعریف تابع واحد گسسته
def u(n):
    return np.where(n >= 0, 1, 0)

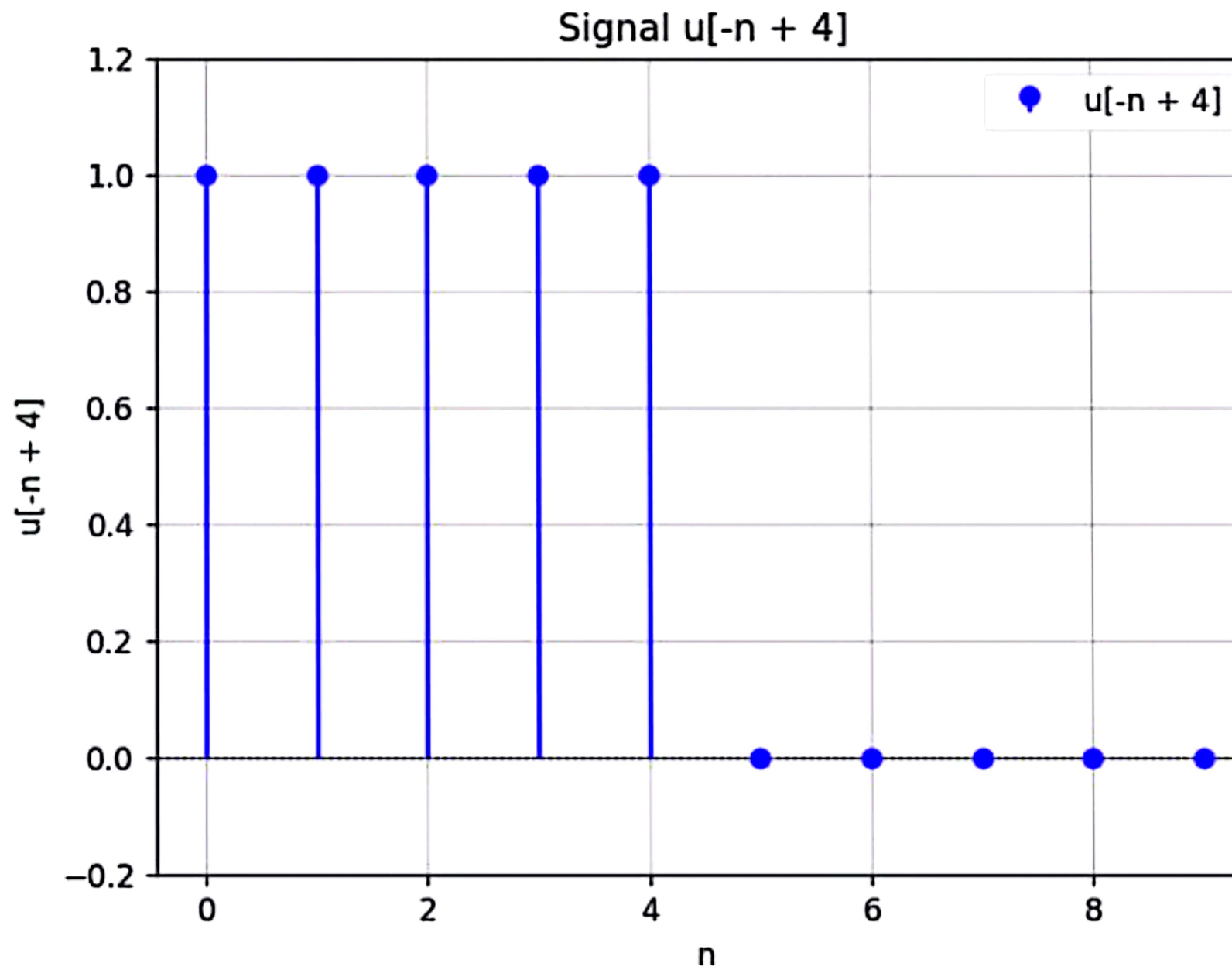
# x[n] تعریف سیگنال x[n] = u[n] - u[n - 3]
x_n = u(n) - u(n - 3)

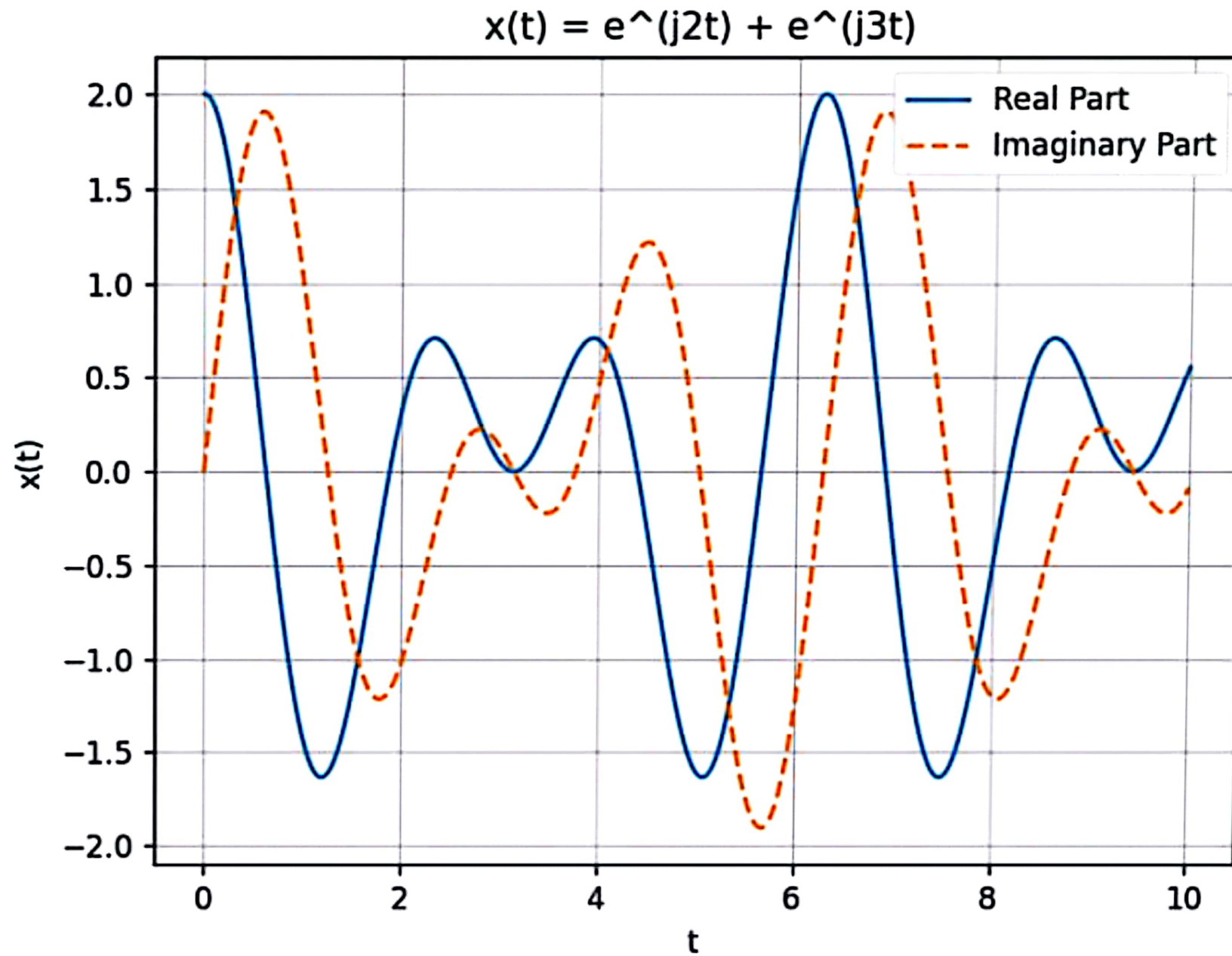
# رسم سیگنال
plt.figure()
plt.stem(n, x_n, basefmt=" ", linefmt='b-', markerfmt='bo', label='x[n] = u[n] - u[n - 3]')
plt.xlabel('n')
plt.ylabel('x[n]')
plt.title('Signal x[n]')
plt.axhline(0, color='black', lw=0.5, ls='--') # خط افقی صفر
plt.grid(True)
plt.ylim(-0.2, 1.2) # تنظیم محدوده محور عمودی
plt.legend()
plt.show()

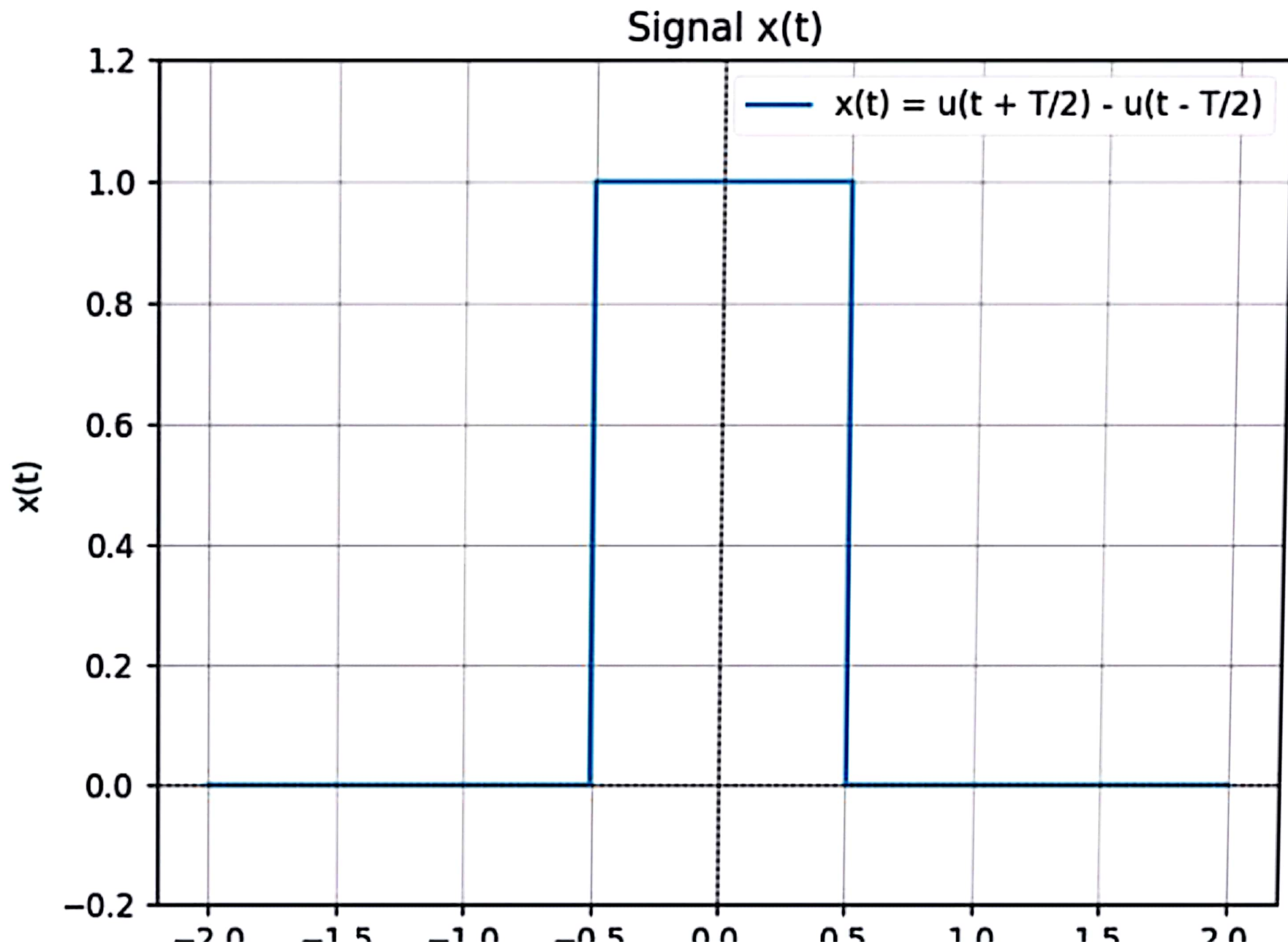
```

Figure 1











```

> 6.py > ...
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # محور گسسته n تعریف دامنه
5 n = np.arange(0, 50, 1) # (اعداد صحیح) 1 با گام 50 تا 0 از
6
7 # تعریف سیگنال مختلط x[n] = e^(j2n) + e^(j3n)
8 x_n = np.exp(1j * 2 * n) + np.exp(1j * 3 * n)
9
10 # استخراج قسمت حقیقی و موهومی
11 real_part = np.real(x_n)
12 imaginary_part = np.imag(x_n)
13
14 # رسم قسمت حقیقی سیگنال
15 plt.figure()
16 plt.stem(n, real_part, label='Real Part', basefmt=" ")
17 plt.stem(n, imaginary_part, label='Imaginary Part', basefmt=" ", linefmt='--', markerfmt='o')
18 plt.xlabel('n')
19 plt.ylabel('x[n]')
20 plt.title('x[n] = e^(j2n) + e^(j3n)')
21 plt.legend()
22 plt.grid(True)
23 plt.show()

```

