

РОБОТА із ЗОБРАЖЕННЯМИ

Файл: Image_07_002

Інтеграл Фур'є. Спектр 1D дискретизованого сигналу.

Використуємо бібліотеку [SCIPY](#)

```
from scipy import fft # модуль для перетворення Фур'є
import matplotlib.pyplot as plt
import numpy as np
from numpy.random import Generator, MT19937
rg = Generator(MT19937(12345))
```

Визначення функції відліків

```
tau = .02
omega = np.linspace(-20/tau ,20/tau ,201)

sinc_arg = omega*tau/2

snc = np.sinc(sinc_arg)

plt.figure()
plt.title("Функція відліків")
plt.plot(omega , snc)
plt.xlabel("Rad-1")
plt.ylabel("Sinc()")
plt.grid()
plt.show()
```



Відновлення сигналу ПРЯМОКУТНИЙ ІМПУЛЬС

```
# Number of sample points
t_max = 10.0 # full interval in seconds
N = 20      # Number of sample points
S = t_max / N # sample spacing
f_main = 1.0 # ЧАСТОТА ОСНОВНОГО СИГНАЛА ГЕРЦ
tau_main = 1.0/f_main # ПЕРИОД ОСНОВНОГО СИГНАЛА сек
print ('Інтервал ', t_max , 'секунд. Кількість відліків', N, 'Довжина відліку', S)
print ('Основан частота сигналу', f_main, 'Hz, Основний період', tau_main )

t = np.linspace(0, t_max, N+1) # start, stop, num

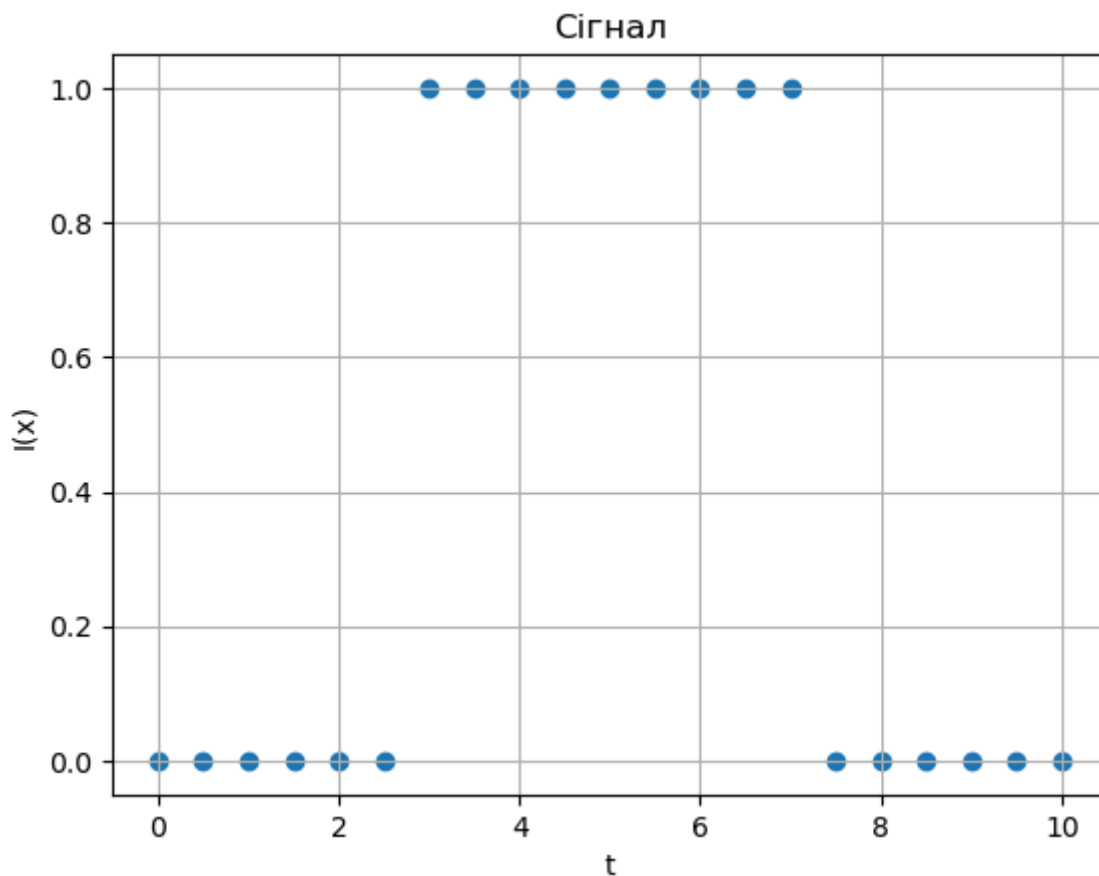
# Визначення вузьких прямокутних імпульсів
def func(x, t_mx):
    if x < 0.3 * t_mx:
        p = 0.0
    elif x > 0.7 * t_mx:
        p = 0.0
    else : p = 1.0
    return p
```

Інтервал 10.0 секунд. Кількість відліків 20 довжина відліку 0.5
 Основан частота сигналу 1.0 Hz, Основний період 1.0

```

y = [func(x, t_max) for x in t]
plt.title("Сигнал")
plt.scatter(t, y)
plt.xlabel("t")
plt.ylabel("I(x)")
plt.grid()
plt.show()

```



Зрушені функції відліків

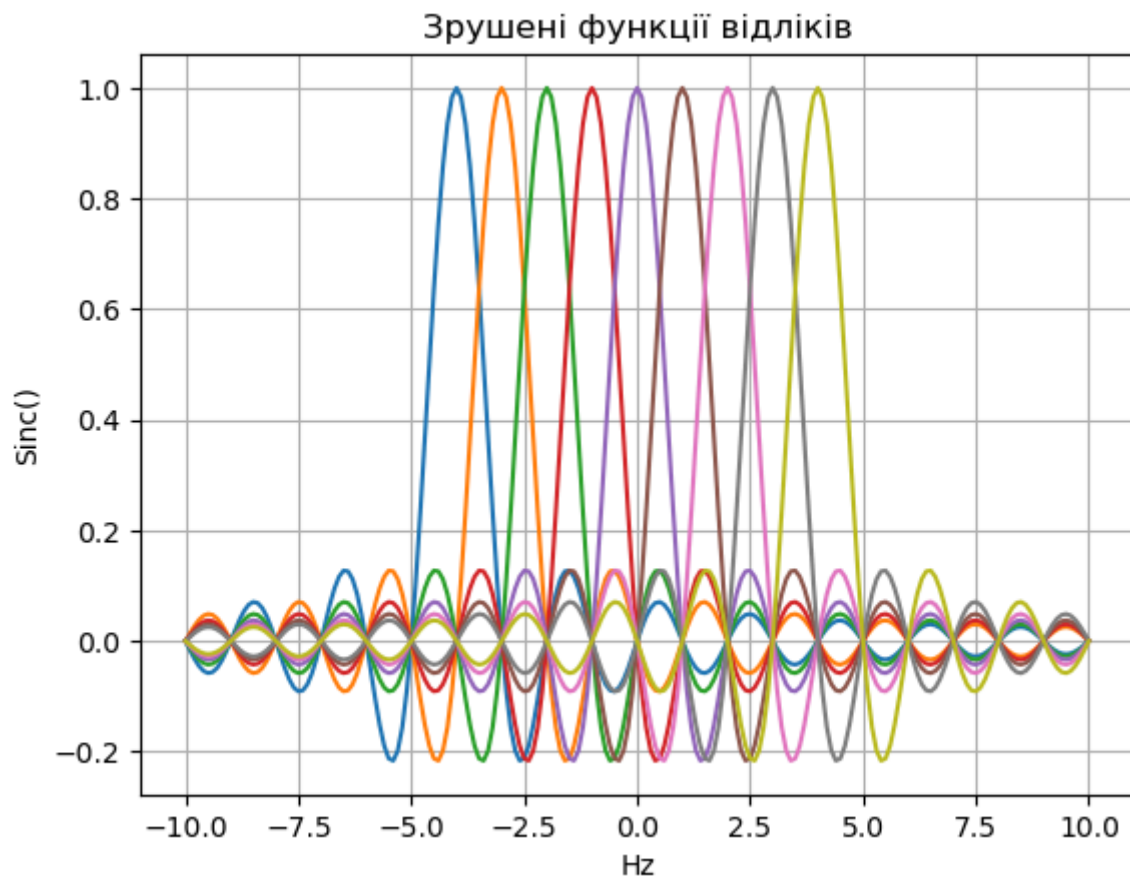
```

omega = np.linspace(-10,10,201)
snc = np.sinc(omega)
snc_4 = np.sinc(omega+4)
snc_3 = np.sinc(omega+3)
snc_2 = np.sinc(omega+2)
snc_1 = np.sinc(omega+1)
snc0 = np.sinc(omega)
snc1 = np.sinc(omega-1)
snc2 = np.sinc(omega-2)
snc3 = np.sinc(omega-3)
snc4 = np.sinc(omega-4)

plt.figure()
plt.title("Зрушені функції відліків")
plt.plot(omega, snc_4)
plt.plot(omega, snc_3)
plt.plot(omega, snc_2)

```

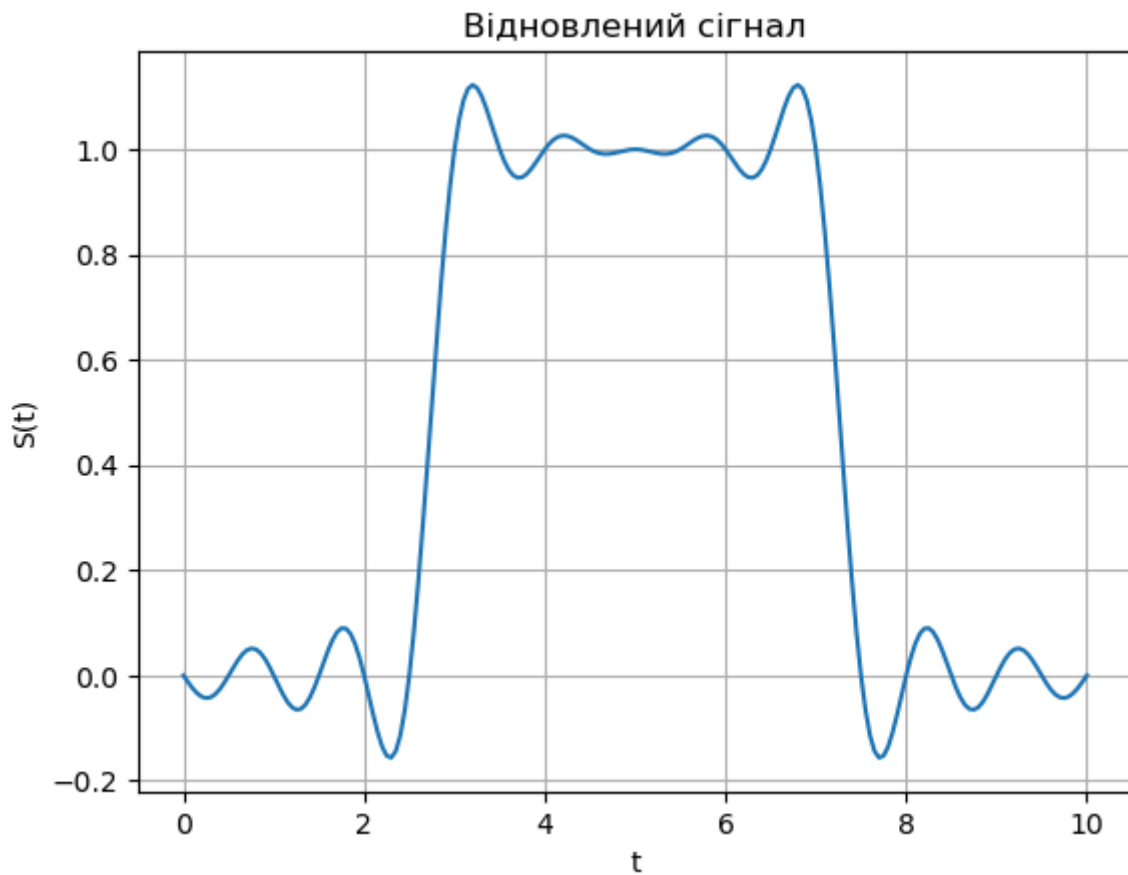
```
plt.plot(omega, snc_1)
plt.plot(omega, snc0)
plt.plot(omega, snc1)
plt.plot(omega, snc2)
plt.plot(omega, snc3)
plt.plot(omega, snc4)
plt.xlabel("Hz")
plt.ylabel("Sinc()")
plt.grid()
plt.show()
```



Відновлення сигналу

```
signal = np.zeros(201, dtype = np.float32)
for i in range (201):
    signal[i] =
snc_4[i]+snc_3[i]+snc_2[i]+snc_1[i]+snc0[i]+snc1[i]+snc2[i]+snc3[i]+snc4[i]

ts = np.linspace(0, t_max, 201) # start, stop, num
plt.title("Відновлений сигнал")
plt.plot(ts,signal)
plt.xlabel("t")
plt.ylabel("S(t)")
plt.grid()
plt.show()
```



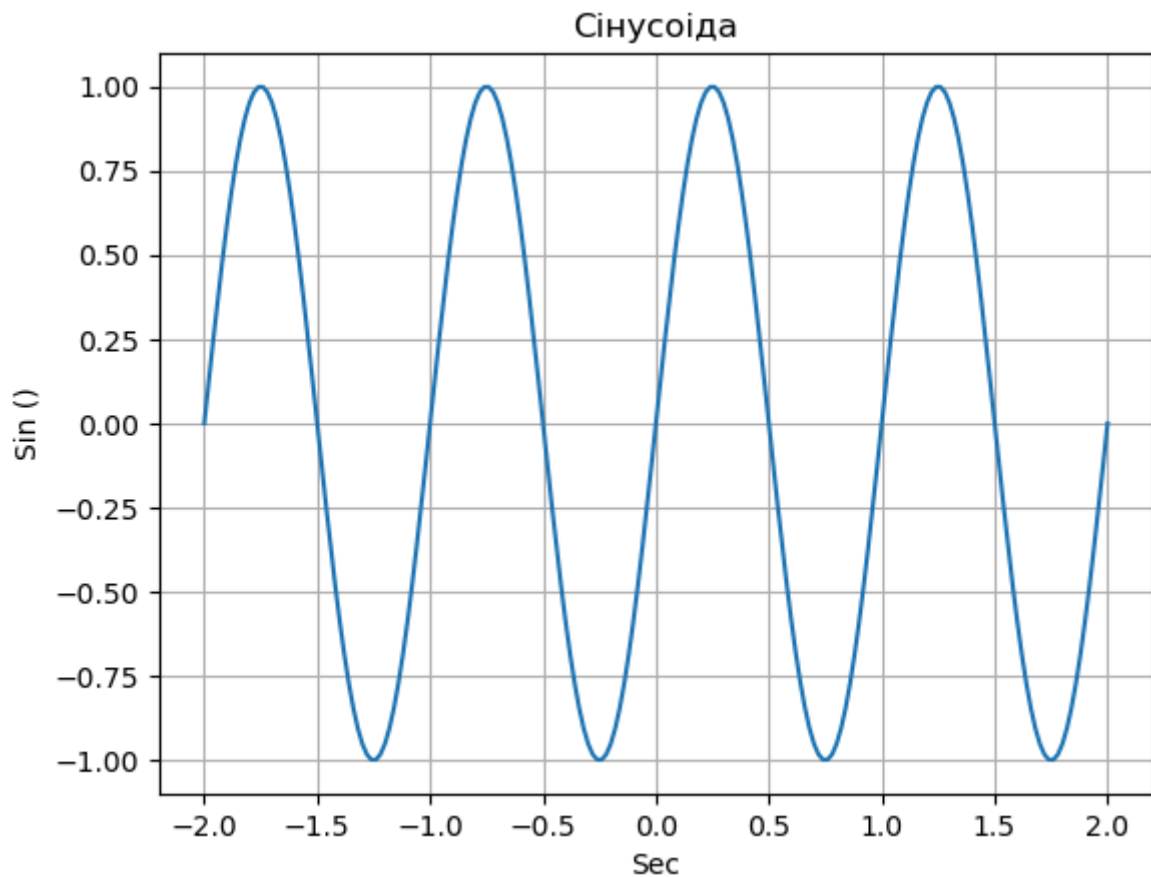
Відновлення синусоїдального сигналу

```
period = 1.
frequency = 1 / period
omega = 2*np.pi*frequency
print (period, frequency, omega)

time = np.linspace(-2*period , 2*period ,201)
signal = np.sin(omega*time)
```

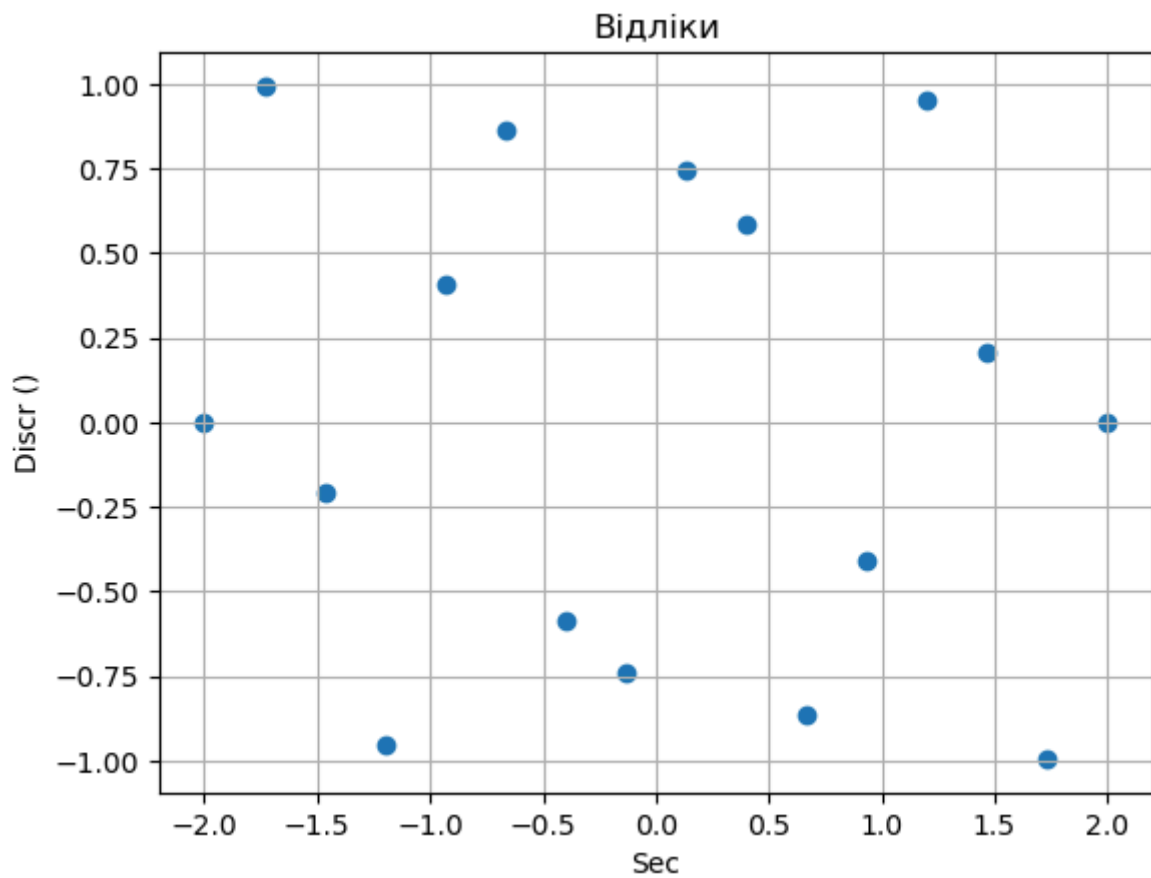
```
1.0 1.0 6.283185307179586
```

```
plt.figure()
plt.title("Сінусоїда")
plt.plot(time, signal)
plt.xlabel("Sec")
plt.ylabel("Sin ()")
plt.grid()
plt.show()
```



```
# дискретизація сінусоїдального сигналу
sampling_period = .25 # період дискретизації
rate = period/sampling_period
start_time = 0.0
print (sampling_period, rate)
discret_time = np.linspace((-2*period+start_time), 2*period,
np.int32(4*period/sampling_period))
discr_signal = np.sin(omega*discret_time)
print ('Samples', len(discr_signal))
plt.figure()
plt.title("Відліки")
plt.scatter(discret_time, discr_signal)
plt.xlabel("Sec")
plt.ylabel("Discr ()")
plt.grid()
plt.show()
```

```
0.25 4.0
Samples 16
```



```

omega = np.linspace(-10,10,201)
snc = np.sinc(omega)
snc_3 = np.sinc(omega+3)
snc_2 = np.sinc(omega+2)
snc_1 = np.sinc(omega+1)
snc0 = np.sinc(omega)
snc1 = np.sinc(omega-1)
snc2 = np.sinc(omega-2)
snc3 = np.sinc(omega-3)

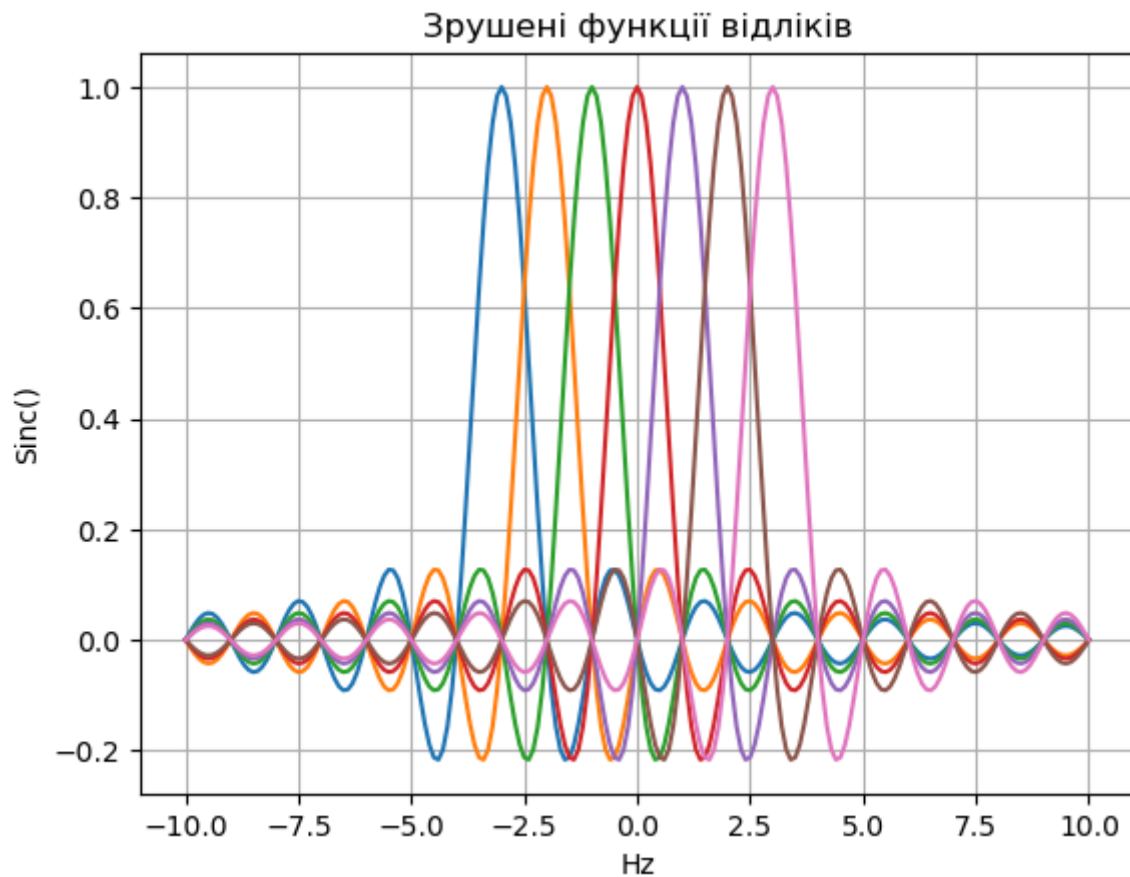
```

```

omega = np.linspace(-10,10,201)

plt.figure()
plt.title("Зрушені функції відліків")
plt.plot(omega, snc_3)
plt.plot(omega, snc_2)
plt.plot(omega, snc_1)
plt.plot(omega, snc0)
plt.plot(omega, snc1)
plt.plot(omega, snc2)
plt.plot(omega, snc3)
plt.xlabel("Hz")
plt.ylabel("Sinc()")
plt.grid()
plt.show()

```



```

signal = np.zeros(201, dtype = np.float32)
for i in range (201):
    signal[i] = discr_signal[5]*snc_3[i]+ \
                discr_signal[6]*snc_2[i]+ \
                discr_signal[7]*snc_1[i]+ \
                discr_signal[8]*snc0[i]+ \
                discr_signal[9]*snc1[i]+ \
                discr_signal[10]*snc2[i]+ \
                discr_signal[11]*snc3[i]

ts = np.linspace(0, t_max, 201) # start, stop, num
plt.title("Відновлений сигнал")
plt.plot(ts,signal)
plt.xlabel("t")
plt.ylabel("S(t)")
plt.grid()
plt.show()

```