

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій

Звіт

про виконання лабораторної роботи №1
ДИСКРЕТНА ЗГОРТКА СИГНАЛІВ

Виконала
студентка групи ФеС-21
Янишин А.О

Перевірив
Вдовиченко В. М.

Львів 2024 р.

Лабораторна робота №1

Дискретна згортка сигналів

Мета роботи:

Ознайомитися з поняттям дискретних систем. Освоїти процес та алгоритм дискретної згортки сигналів.

Завдання до роботи:

Створити програму для знаходження дискретної згортки $\{f_m\}$ дискретних сигналів $\{x_k\}$ і $\{y_k\}$.

Теоретичні відомості

Узагальнена структурна схема дискретної системи представлена на рис. 1.

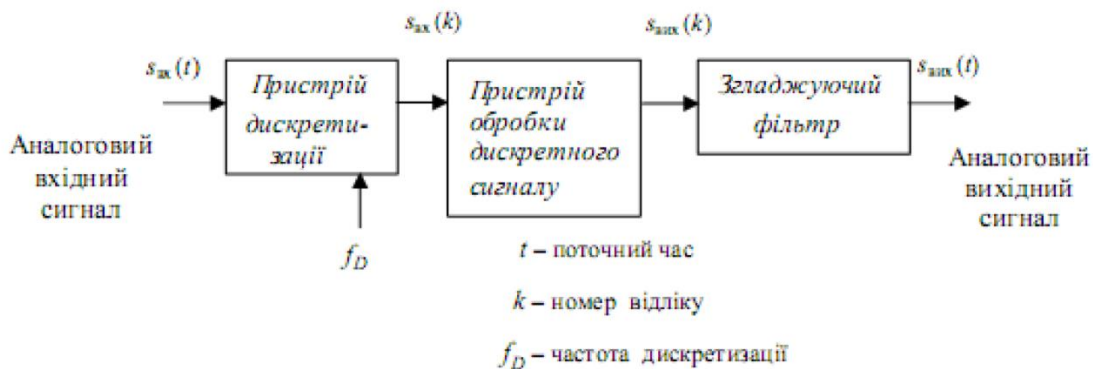


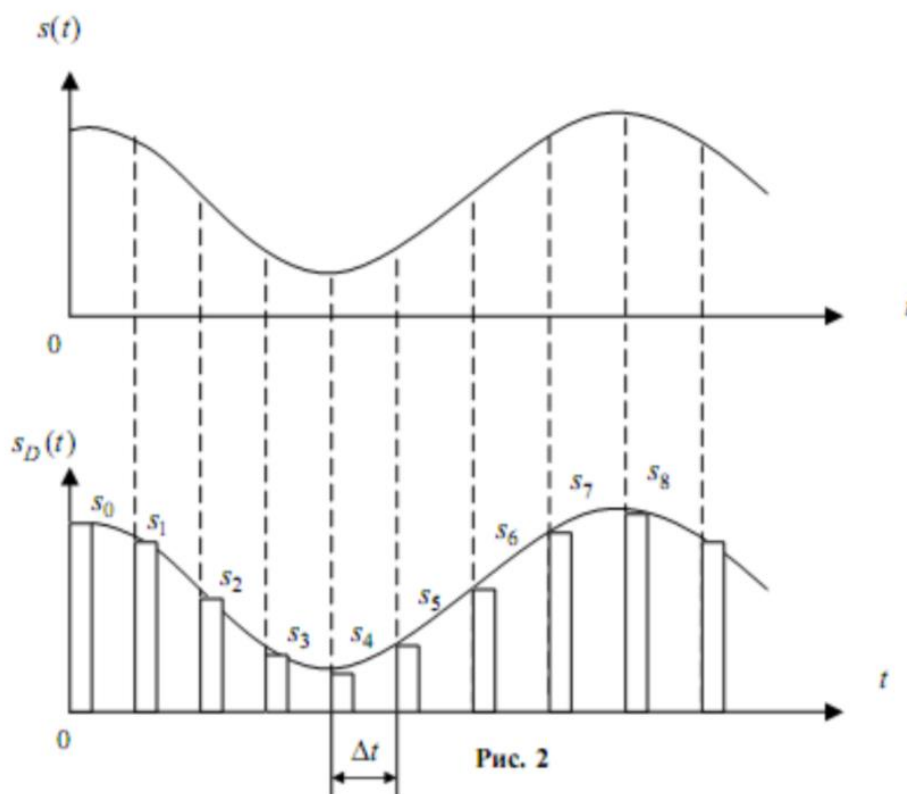
Рис. 1

Вхідний аналоговий сигнал переводиться в послідовність відліків $s_{\text{вх}}(k)$ і надходить на пристрій обробки, звідки знімається вихідна імпульсна послідовність $s_{\text{вих}}(k)$, яка потім згладжується фільтром.

Окремим випадком дискретної системи є система цифрової обробки сигналу (ЦОС), коли послідовність вхідних відліків $s_{\text{вх}}(k)$ оцифровується. У цьому

випадку, очевидно, пристрій обробки повинен мати аналогово-цифровий перетворювач (АЦП) на вході і цифро-аналоговий перетворювач (ЦАП) на виході.

Перехід від аналогового безперервного сигналу $s(t)$ до дискретного $s_D(t)$ здійснюється шляхом дискретизації по часу (рис. 2). З рисунків бачимо, що вихідний неперервний сигнал $s(t)$ представляється послідовністю відліків $\{s_k\}$, де $s_k = s(k\Delta t)$. Інтервал Δt називають кроком дискретизації, а $f_D = \frac{1}{\Delta t}$ – частотою дискретизації. Зрозуміло, що для уникнення втрат інформації крок дискретизації повинен бути досить малим. З іншого боку, занадто часті відліки ведуть до невинновданої надмірності інформації і ускладнення апаратури. Відповідь про правильний вибір Δt дає теорема Найквіста-Котельникова-Шенонна.



Теорема Найквіста-Котельникова-Шеннона: довільний сигнал $s(t)$, спектр якого обмежений частотою F_B , може бути повністю відтворений по послідовності своїх відліків, взятих з інтервалом $\Delta t \leq \frac{1}{2F_B}$.

При цьому відновлення здійснюється за допомогою ряду:

$$s(t) = \sum_{k=-\infty}^{\infty} S_k \frac{\sin\left[\frac{\pi}{\Delta t}(t-k\Delta t)\right]}{\frac{\pi}{\Delta t}(t-k\Delta t)}.$$

Згортку двох аналогових сигналів можна зобразити у вигляді:

$$x(t) * y(t) = f(t) = \int_{-\infty}^{\infty} x(\tau)y(t - \tau)d\tau = \int_{-\infty}^{\infty} y(\tau)x(t - \tau)d\tau$$

За аналогією зі згорткою неперервних сигналів в теорії дискретних систем вводять дискретну згортку – сигнал, відліки якого пов'язані з відліками дискретних сигналів $\{x_k\}$ і $\{y_k\}$ співвідношенням:

$$f_m = \sum_{k=0}^{\infty} x_k y_{m-k}, m = 0, 1, 2...$$

Варіант 15

Програма для знаходження дискретної згортки дискретних сигналів:

```
def discrete_convolution(x, y):
    len_x = len(x)
    len_y = len(y)
    result_len = len_x + len_y - 1
    result = [0] * result_len

    for m in range(result_len):
        sum_value = 0
        for k in range(len_x):
            if 0 <= m - k < len_y:
                sum_value += x[k] * y[len_y - 1 - (m - k)]
        result[m] = sum_value

    return result

x = [6, 5, 4]
y = [8, 1, 0]

x1 = [6, 4, 0, 1, 2, 3, 6, 4, 0, 3, 8, 7 ]
y1 = [4, 0, 8, 2, 7, 3]

x2 = [2, 6, 4, 0, 8, 2]
y2 = [6, 4, 0, 1, 2, 8, 6, 4, 0, 8, 3, 7 ]
```

```
print(f"Результат згортки: {discrete_convolution(x, y)}")
print(f"Результат згортки: {discrete_convolution(x1, y1)}")
print(f"Результат згортки: {discrete_convolution(x2, y2)}")
```

Вхідні дані:

15	6, 5, 4	8, 1, 0
	6, 4, 0, 1, 2, 3, 6, 4, 0, 3, 8, 7	4, 0, 8, 2, 7, 3,
	2, 6, 4, 0, 8, 2	6, 4, 0, 1, 2, 8, 6, 4, 0, 8, 3, 7

Скріншот роботи програми:

```
D:\Navchanya\2_kurs\DSP\Lab1\pythonProject\.venv\Scripts\python.exe D:\Navchanya\2_kurs\DSP\Lab1\pythonP
Результат згортки: [0, 6, 53, 44, 32]
Результат згортки: [18, 54, 40, 59, 45, 49, 67, 76, 68, 73, 89, 107, 105, 78, 68, 32, 28]
Результат згортки: [14, 48, 62, 60, 96, 74, 138, 92, 78, 70, 88, 68, 64, 26, 32, 56, 12]

Process finished with exit code 0
```