

АНАЛІЗ МОЖЛИВОСТЕЙ ІНСТРУМЕНТІВ PYTHON ДЛЯ ЦИФРОВОЇ ОБРОБКИ СИГНАЛІВ

Останнім часом мова програмування Python досить активно розвивається і все частіше використовується при створенні найрізноманітніших проєктів. Дана мова є високорівневою мовою програмування загального призначення і орієнтована на підвищення продуктивності розробника та читабельності коду. Такі інструменти мови Python як SciPy і NumPy розкривають досить широкі можливості для цифрового оброблення сигналів [1–4].

Бібліотека SciPy [5] має відкритий вихідний код і призначена для виконання наукових і інженерних розрахунків. Основною структурою даних в SciPy є багатовимірний, реалізований модулем NumPy масив.

Для оброблення сигналів можуть бути використані такі пакети:

1. `scipy.signal`, що забезпечує виконання таких операцій: згортка одномірних і двомірних послідовностей; робота зі сплайнами; проектування фільтрів; фільтрація сигналів, включаючи зміну частоти дискретизації і перетворення Гільберта; моделювання лінійних безперервних і дискретних систем; формування періодичних і неперіодичних сигналів; формування вікон для спектрального аналізу; робота з вейвлетами; пошук пікових значень; спектральний аналіз.

2. `scipy.stats` дає змогу виконувати: моделювання безперервних і дискретних випадкових величин; розрахунок різнотипних статистик, відстаней та кореляційних функцій; проведення статистичних тестів; перетворення випадкових величин та ін.

Пакет NumPy [6] забезпечує підтримку великих багатовимірних масивів і матриць разом з великою бібліотекою високорівневих і дуже швидких математичних функцій для операцій з цими масивами. Із використанням таких матриць можна створювати власні високоефективні функції для оброблення сигналів.

Для візуалізації процесу оброблення даних на дво- та тривимірних графіках використовується бібліотека Matplotlib [6]. Бібліотека для машинного навчання [Scikit-learn](#) дозволяє реалізовувати різні алгоритми класифікації, регресії і кластеризації, які побудовані на взаємодії бібліотек NumPy і SciPy з Python.

Крім розглянутих основних інструментів існує величезна кількість створених спільнотою Python пакетів для аналізу та оброблення сигналів. Наприклад, на веб-сервісі GitHub можна знайти пакети `spectrum` для реалізації різних методів спектрального аналізу, `pyftb` для частотно-часового аналізу.

У науковому середовищі для розробки, обробки даних і інтерактивного виконання застосунків, пов'язаних з бібліотеками NumPy, Matplotlib, SymPy і SciPy активно використовується інтерактивна оболонка мови програмування Python – IPython, яка поєднує можливості інтерактивної консолі Python і командної оболонки Unix.

Оскільки Python є інтерпретованою мовою, то швидкість виконання коду менша, ніж для мов, що використовують компілятор. Якщо швидкодія написаного на Python коду не задовольняє поставленим вимогам, то він може бути легко модифікований для скорочення часу його виконання із використанням таких технологій як Cython, Numba і PyPy [7].

Список використаних джерел

1. Downey A. B. Think DSP: Digital Signal Processing in Python. Green Tea Press, 2014. 153 p.
2. Charbit M. Digital Signal Processing with Python Programming. Wiley, 2017. 284 p.
3. Unpingco J. Python for Signal Processing. Featuring IPython Notebooks. Springer, 2014. 133 p.
4. Idris I. Python Data Analysis. Learn how to apply powerful data analysis techniques with popular open source Python modules. Packt Publishing, 2014. 348 p.
5. Blanco-Silva F. J. Mastering SciPy. Implement state-of-the-art techniques to visualize solutions to challenging problems in scientific computing, with the use of the SciPy stack. Packt Publishing, 2015. 404 p.
6. Johansson R. Numerical Python. Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib. 2nd Edition. Apress, 2019. 709 p.
7. Gorelick M., Ozsvald I. High Performance Python. O'Reilly, 2014. 370 p.