

Лабораторна робота №5

Візуалізація даних

Мета роботи: отримати поглиблені навички з візуалізації даних; ознайомитись з `matplotlib.widgets`, `scipy.signal.filters`, а також з `Plotly`, `Bokeh`, `Altair`; отримати навички зі створення інтерактивних застосунків для швидкого підбору параметрів і аналізу отриманих результатів

Постановка задачі:

Створіть програму, яка дозволить користувачам малювати графік функції гармоніки (функція виду $y(t) = A * \sin(\omega * t + \varphi)$) з накладеним шумом та надавати можливість змінювати параметри гармоніки та шуму за допомогою інтерактивного інтерфейсу, що включає в себе слайдери, кнопки та чекбокси. Зашумлену гармоніку відфільтруйте за допомогою фільтру на вибір, порівняйте результат.

Завдання 1 (3б):

1. Створіть програму, яка використовує бібліотеки `Matplotlib` для створення графічного інтерфейсу.
2. Реалізуйте функцію `harmonic_with_noise`, яка приймає наступні параметри:
 - `amplitude` - амплітуда гармоніки.
 - `frequency` - частота гармоніки.
 - `phase` – фазовий зсув гармоніки
 - `noise_mean` - амплітуда шуму.
 - `noise_covariance` – дисперсія шуму
 - `show_noise` - флаг, який вказує, чи слід показувати шум на графіку.
3. У програмі має бути створено головне вікно з такими елементами інтерфейсу:
 - Поле для графіку функції (`plot`)
 - Слайдери (`sliders`), які відповідають за амплітуду, частоту гармоніки, а також слайдери для параметрів шуму
 - Чекбокс для перемикання відображення шуму на гармоніці
 - Кнопка «Reset», яка відновлює початкові параметри
4. Програма повинна мати початкові значення кожного параметру, а також передавати параметри для відображення оновленого графіку.
5. Через чекбокс користувач може вмикати або вимикати відображення шуму на графіку. Якщо прапорець прибрано – відображати «чисту гармоніку», якщо ні – зашумлену.
6. Після оновлення параметрів програма повинна одразу оновлювати графік функції гармоніки з накладеним шумом згідно з виставленими параметрами.
 - *Зауваження:* якщо ви змінили параметри гармоніки, але не змінювали параметри шуму, то шум має залишитись таким як і був, а не генеруватись наново. Якщо ви змінили параметри шуму, змінюватись має лише шум – параметри гармоніки мають залишатись незмінними.
7. Після натискання кнопки «Reset», мають відновитись початкові параметри
8. Залиште коментарі та інструкції для користувача, які пояснюють, як користуватися програмою.
9. Завантажте файл зі скриптом до вашого репозиторію на `GitHub`
10. Надайте короткий звіт про ваш досвід та вивчені навички.

Зауваження: структура застосунку, назви функцій та змінних, навіть вибір бібліотек наведено з метою рекомендації. Ви вільні обирати це за вашим смаком, головна вимога – повне збереження функціоналу.

Завдання 2 (26):

1. Отриману гармоніку з накладеним на неї шумом відфільтруйте за допомогою фільтру на ваш вибір (наприклад [scipy.signal.iirfilter](https://docs.scipy.org/doc/scipy/reference/signal.html), повний список за посиланням: <https://docs.scipy.org/doc/scipy/reference/signal.html>). Відфільтрована гармоніка має бути максимально близька до «чистої»
2. Відобразіть відфільтровану «чисту» гармоніку поряд з початковою
3. Додайте відповідні інтерактивні елементи (чекбокс показу, параметри фільтру тощо) та оновіть існуючі: відфільтрована гармоніка має оновлюватись разом з початковою.

Завдання 3 (додаткове, 5б)

1. Реалізуйте завдання 1 за допомогою сучасних графічних бібліотек на ваш вибір: Plotly, Bokeh, Altair тощо. Додайте декілька вікон для візуалізації замість одного, спадне меню (drop-down menu) та інші інтерактивні елементи на власний розсуд.
2. Реалізуйте ваш власний фільтр, використовуючи виключно Python (а також numpy, але виключно для операцій з масивами numpy.ndarray). Застосуйте фільтр

Корисні посилання

1. <https://matplotlib.org/stable/gallery/widgets/index.html>
2. <https://docs.scipy.org/doc/scipy/reference/signal.html>
3. <https://plotly.com/python/#controls>
4. https://docs.bokeh.org/en/latest/docs/user_guide/interaction/widgets.html
5. https://altair-viz.github.io/user_guide/interactions.html