DOI: https://doi.org/10.36910/6775-2524-0560-2021-42-06
УДК 534.44:004
Головін Микола Борисович, канд. фіз.-мат. наук, доцент,
https://orcid.org/0000-0003-4516-4677
Головіна Ніна Анатоліївна, канд. фіз.-мат. наук, доцент
ВНУ імені Лесі Українки, м. Луцьк

ФУР'Є ПЕРЕТВОРЕННЯ В ЯКОСТІ АПЛІКАЦІЇ СПЕКТРАЛЬНОГО АНАЛІЗУ ЗВУКІВ У КУРСАХ КОМП'ЮТЕРНОЇ ФІЗИКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

Головін М.Б., Головіна Н.А. Фур'є перетворення в якості аплікації спектрального аналізу звуків у курсах комп'ютерної фізики та захисту інформації. Розглянуто та апробовано прості приклади програмного коду в яких реалізуються Фур'є перетворення на мові Python засобами бібліотеки Numpy. Програмний код може бути використаний, як аплікація в багатьох розділах фізики для ілюстрації аналізу хвильових процесів, зокрема, в першу чергу, в акустиці та в оптиці, а також в курсі захисту інформації. Матеріал демонструє чисельні міжпредметні зв'язки математики та інформатики з різними розділами фізики та з курсом захисту інформації.
Ключові слова: Фур'є перетворення, спектральний аналіз, навчальне програмування, мова Python, бібліотека Numpy, міжпредметні зв'язки, комп'ютерна фізика, захист інформації.

Головин М.Б., Головина Н.А. Фурье преобразования в качестве аппликации спектрального анализа звуков в курсах компьютерной физики и защиты информации. Рассмотрены и апробированы простые примеры программного кода в которых реализуются Фурье преобразования на языке Python средствами библиотеки Numpy. Программный код может быть использован, как аппликация во многих разделах физики для иллюстрации анализа волновых процессов, в частности, в первую очередь, в акустике и в оптике, а также в курсе защиты информации. Материал демонстрирует многочисленные межпредметные связи математики и информатики с различными разделами физики и с курсом защиты информации.
Ключевые слова: Фурье преобразования, спектральный анализ, учебное программирование, язык Python, библиотека Numpy, межпредметные связи, компьютерная физика, защита информации.

Holovin M.B., Holovina N. A. Usage of Fourier transforms for spectral analysis of sounds in the courses of computer physics and information security. The simple examples of program code through which Fourier transform is realized on the Pyton language by the means of Numpy library is considered and tested. The program code may be used in different domains of physics for illustration the analysis of wave processes. In particular, it is especially relevant in the domains of acoustics, optics and the course of information security. The material demonstrates different inter-disciplinary connections between mathematics, informatics with different domains of physics and the course of information security.
Key words: Fourier transforms, spectral analysis, educational programming, Python language, Numpy library, interdisciplinary connections, computer physics, information security.

Постановка проблеми. Перетворення Фур'є в контексті вивчення фізики мають важливе світоглядне значення. На цьому прикладі можна проілюструвати нерозривність фізики, математики, а також інформатики. Відповідно міжпредметні зв'язки в цій темі максимально видимі. Особливо цікава в цьому сенсі акустика. Тут прості фізичні сигнали, можна отримувати без додаткових фізичних сенсорів та синтезаторів сигналу, адже пристрій, що здійснює математичну обробку (ноутбук) має на борту мікрофон і гучномовці. Ця робота є продовженням роботи [1] по підбору цікавих задач комп'ютерної фізики.

Іншим важливим моментом є те, що математичні перетворення Фур'є використовуються не тільки в акустиці. Існує багато практичних застосувань в інших різних областях знань, зокрема, в теорії чисел, комбінаториці, обробці сигналів, теорії ймовірностей, статистиці, криптографії, акустиці, океанології, оптиці, геометрії. Такий набір застосувань в різних сферах науки дозволяє створити нові міжпредметні зв'язки в широкому діапазоні напрямків.

З математичної точки зору обробка фізичних сигналів через перетворення Фур'є дозволяє реалізувати декомпозицію фізичного сигналу на сукупність відповідних синусоїдальних складових з їх відповідними параметрами: частотою, їх амплітудою та фазою. Також є можливість здійснювати і обернений перехід у модельний фізичний сигнал, який по суті є композицією різноамплітудних частот коливання матерії з відповідними фазовими зсувами. Перетворення Фур'є можна розглядати, як транзити в розгляді фізичного явища з часового простору (time domain) в частотний простір (frequency domain) та у зворотному напрямку.

Аналіз саме звукових коливань є цікавою темою з прикладної точки зору і сьогодні. Підсистема аналізу звукових коливань може входити як складова частина в системи розпізнавання голосу, які в свою чергу можуть входити в системи захисту банків, військових об'єктів, державних установ. Крім того, такі системи можуть використовуватись в криміналістичних системах ідентифікації конкретних

звуків. Аналіз акустичних коливань може стати у нагоді при автоматизованій діагностиці різноманітних систем від механічних (двигуни внутрішнього згорання) до біологічних, в тому числі тіла людини. Задача ідентифікації звукових коливань актуальна з точки зору інформатики саме тому, що входить у велику проблему розпізнавання образів.

Людина легко справляється з проблемою розпізнавання, розрізняючи голоси. Процеси сприйняття і розпізнавання звуку у людини майже не торкаються свідомості. З біологічної точки зору ці процеси важко піддаються дослідженню.

Метою роботи є розгляд та апробація простих прикладів програмного коду в яких реалізуються Фур'є перетворення на мові Python. Цей код є важливим, адже може бути використаний, як аплікація в різноманітних розділах комп'ютерної фізики та в курсі захисту інформації.

Виклад основного матеріалу дослідження. Звук це коливальний рух частинок пружного середовища, який поширюється у вигляді хвиль у газоподібному, рідкому чи твердому середовищах. Це фізичне явище суб'єктивно сприймається органами слуху людини чи тварини. Людина чує звук в діапазоні частот від 16 Герц до 20 кілоГерц.

Усяке тіло, що коливається зі звуковою частотою, будучи поміщеним у пружне середовище, стає джерелом звукових хвиль. У струнних інструментах джерело звуку — струна, з'єднана з резонатором - корпусом інструмента, а у гучномовцях звук випромінює пружна коливна поверхня тієї чи іншої форми. Спеціальним пристроєм для генерації каліброваних звуків є камертон.

Серед звуків, що ми чуємо, є такий сорт, що називається шумом. Йому відповідають нерегулярні короткі стохастичні коливання різної частоти, амплітуди і фази. Музичний звук має інший характер. Музика характеризується наявністю більш-менш тривалих тонів, чи музичних «нот».

Звук струни, що вібрує, породжується так званими стоячими хвилями струни. Ці хвилі при схильності до затухання все ж таки мають досить тривалий час дії. Звук, який при цьому виникає, триває поки струна коливається. Частотний спектр цього коливання є квазістатичним.

Аналіз звуку – це розклад складного звукового процесу на ряд гармонічних коливань через Фур'є перетворення. Частотний аналіз дозволяє отримати розподіл амплітуд складових за частотами, амплітудами та фазами. Це амплітудно - частотні спектри. Інколи – розподіл фаз частотних складових. Це – фазочастотні спектри [2].

Ідентифікація звуків базується на програмній реалізації математичного Фур'є перетворення, дозволяє швидко здійснювати дослідження індивідуальних характеристик звукових коливань, як сукупності гармонічних складових.

Модельний експеримент. Розглянемо Фур'є перетворення реалізовані на мові Python з використанням бібліотеки Numpy. Нижче представлений код програми, який спочатку синтезує звукові коливання, як сукупність гармонічних складових, а потім перетворенням Фур'є вилучає зі складного, щойно синтезованого амплітудно-часового спектру, амплітудно-частотні складові цього спектру.[3, 4]. Представлена програма діюча, її можна відновити, якщо зшити до купи представлені в тексті програмні фрагменти.

Перший фрагмент програми підключає потрібні бібліотеки та вводить початкові параметри експерименту.

```
from numpy import array, arange, abs as np_abs
from numpy.fft import rfft, rfftfreq
from math import sin, pi, exp
import matplotlib.pyplot as plt
import matplotlib as mpl
FD = 250000                    #частота дискретизації, відліків в секунду
N = 2500                       #довжина вхідного масиву, N/FD секунд
F=25.0                         #циклічна частота вхідного сигналу
w=(2.*pi*F/FD)                 #відлік кругової частоти
A = array([5.0, 10.0, 20.0])   # масив амплітуд гармонійних складових звуку
k = array([100, 200, 400])     # масив частот гармонійних складових звуку
```

Другий фрагмент програми реалізує синтез модельного звуку на основі вхідних амплітудно-частотних параметрів гармонічних складових цього звуку. Звук є амплітудно-часовими коливаннями середовища. З програмного фрагменту видно, що модельний звук формується двома циклами. Зовнішній цикл по параметру «t» - час. Внутрішній цикл по «i» перебирає синусоїдальні складові з притаманними їм амплітудами «A[]» та частотами «k[]». Параметр «i» відповідальний за частотний склад звуку. У циклі по «i» відбувається підстановка у відповідну формулу амплітуд гармонічних

складових та їх частотних коефіцієнтів «k». Видно, що в даному конкретному випадку, синтез звуку реалізується трьома частотними складовими, кожна з яких має свій амплітудний вклад.

Результуючий звук розташовується в масиві «Zvuk», який надалі і буде піддаватись перетворенням Фур'є. З формули, що відповідає за послідовне заповнення масиву у відповідності з розгортанням часу видна експоненціальна складова. Вона відповідає за затухання звукових коливань з часом. Результуючий графік представлений на рис.1.

```
# генерація звуку на основі частотно-амплітудних параметрів гармонійних складових
Zvuk = array([0 for t in range(N)])                    # ініціалізація масиву модельного звуку
for t in range(N):                                     # цикл по модельному часу
  for i in range(3):                                   # цикл по модельних гармонічних складових
    Zvuk[t]=Zvuk[t]+A[i]*exp(1-t*0.002)*sin(k[i]*w*t)  # формування модельного звуку
plt.plot(arange(N)/float(FD), Zvuk, 'r')               # підготовка даних для виводу їх в вигляді графіку
plt.xlabel('Час, сек.'); plt.ylabel(' Амплітуда сигналу ')  # вивід підписів на осях графіку
plt.title(' Вхідний сигнал '); plt.grid(True)          # вивід заголовку та координатної сітки
plt.show()                                             # вивід амплітудно-часового графіку звуку
```

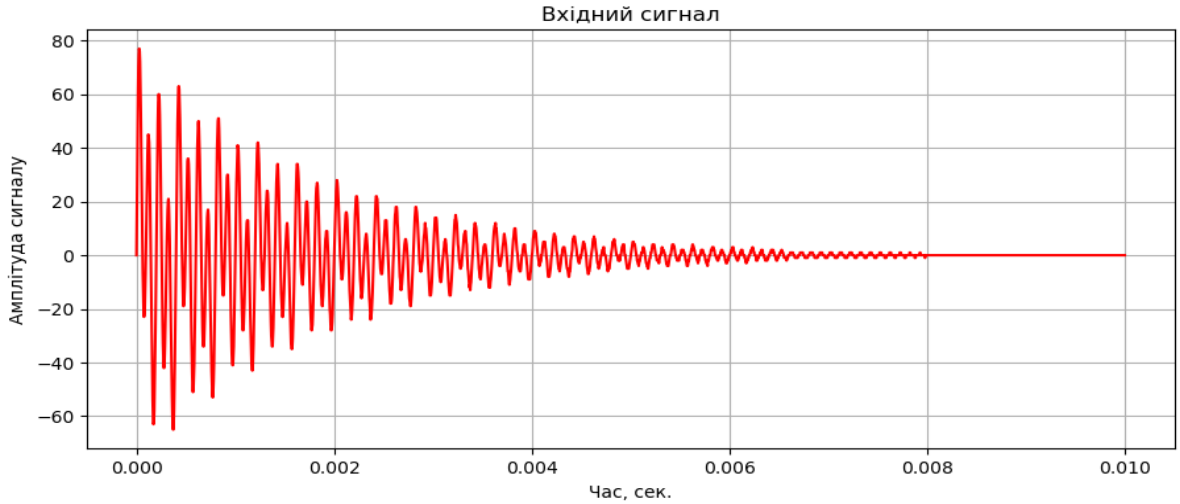

Рис.1 Графік синтезованого звуку

Третій фрагмент програми відповідає за Фур'є перетворення, що реалізує вилучення зі звуку, представленого амплітудно-часовою залежністю, амплітудно-частотних параметрів гармонічних складових звуку.

Функція rfft() з бібліотеки numpy обчислює одновимірне n-точкове дискретне перетворення Фур'є (DFT) дійсного масиву за допомогою ефективного алгоритму, який називається «швидке перетворення Фур'є» (ШПФ). Далі в програмному коді ми використали метод rfftfreq() для вилучення частотних даних, що відповідають методу rfft().

```
# перетворення Фур'є сигналу з амплітудно-часового в амплітудно-частотний
Spectr= rfft(Zvuk)                                     #обчислення дискретного дійсного rfft  перетворення Фур'є
plt.plot(rfftfreq(N, 1./FD), np_abs( Spectr )/N)       # підготовка даних для виводу їх в вигляді графіку
plt.xlabel('Частота, Гц'); plt.ylabel(' Амплітуда синусоїдальних складових ') # вивід підписів на осях графіку
plt.title('Спектр синусоїдальних складових звуку '); plt.grid(True)  # вивід заголовку та координатної сітки
plt.show()                                             #вивід амплітудно-частотного графіку гармонійних складових звуку
```

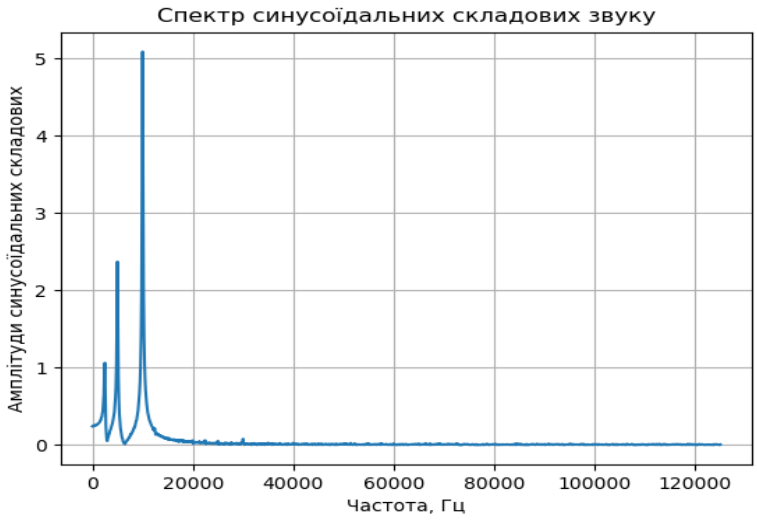

Рис.2 Амплітудно-частотний спектр звуку

З рис.2 видно, що Фур'є перетворення реалізоване в бібліотеці Numpy мови Python досить добре здійснює аналіз гармонічних складових звукових коливань.

Амплітуди гармонічних складових звуку, що представлений в модельному експерименті, співвідносяться, як 5: 10: 20, а частоти гармонічних складових співвідносяться, як 100: 200: 400.

В амплітудній і в частотній модальності вхідних параметрів діє один принцип: кожний наступний параметр перевищує попередній у два рази. Видно, що після перетворення, у вихідних параметрах (рис.2) ця тенденція теж приблизно відтворюється.

Експериментальні дослідження реального звукового сигналу. Були проведені також експериментальні дослідження реального звукового сигналу гітарної струни (рис.3) на предмет його спектрального аналізу за допомогою Фур'є перетворення (рис.4).

```
import numpy as np
import soundfile as sf
import matplotlib.pyplot as plt
data, fs = sf.read('zvuk_struni.wav')                    #завантаження звуку струни
N = len(data); T = 1.0/fs; x = np.linspace(0, (N*T), N);
plt.plot(x, data); plt.xlabel('Час, сек.'); plt.ylabel('Амплітуда сигналу')
plt.title('Вхідний сигнал'); plt.grid(True)
plt.show()                                               # вивід амплітудно-часового графіку звуку
yf = np.fft.rfft(data); xf = np.fft.rfftfreq(data.size, d=T)   # перетворення Фур'є
plt.plot(xf, yf); plt.xlabel('Частота, Гц'); plt.ylabel ('Амплітуда синусоїдальних складових ')
plt.title('Спектр синусоїдальних складових звуку '); plt.grid(True)
plt.show()                                               # вивід спектру гармонійних складових
```

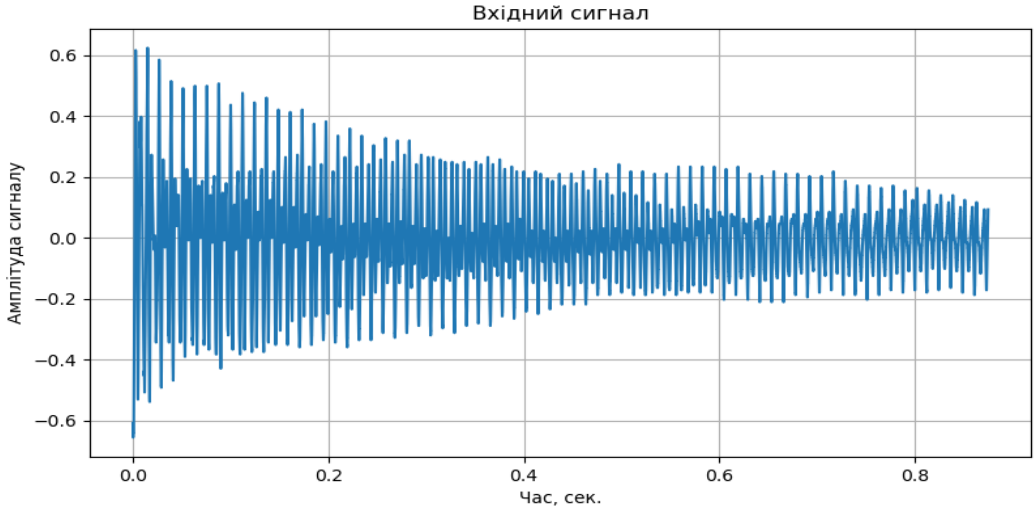

Рис.3 Графік реального звуку гітарної струни

Функція rfft() повертає масив відповідної розмірності, в якому записаний результат ДПФ. Там містяться спочатку додатні частоти від нуля до частоти Найквіста (Котельнікова), потім від'ємні в порядку зростання.

У разі дійсного сигналу від'ємні частоти повністю симетричні додатнім, і тоді немає потреби їх записувати: довжина вихідного масиву N/2 + 1, частоти від нуля до частоти Найквіста. Необхідно зазначити, що дискретний спектр може дати не тільки додатні, а і від'ємні частоти. Це можна побачити, якщо записати ряд Фур'є у комплексній (експоненційній) формі:

$$f(t) = \sum_{k=-\infty}^{\infty} A_k e^{ik\omega_0 t}$$

де $A_к$ є комплексним коефіцієнтом Фур'є, що дорівнює

$$A_k = A_k e^{-i\Phi_k} = \frac{1}{T}\int_{-T/2}^{T/2} f(t)\, e^{-ik\omega_0 t} dt$$

Оскільки, в даному випадку k⊂ (—∞, +∞), то спектральні характеристики $A_к$ та $Ф_к$ будуть утворювати дискретний спектр, який має як додатні, так і від'ємні частоти. На практиці функція f(t) завжди є дійсною функцією. Тому від'ємні частоти фактично не несуть ніякої додаткової інформації, що можна легко показати, використовуючи властивості симетрії модуля та фази фур'є-образу, з яких випливає, що внаслідок дійсності функції при заміні додатних частот на від'ємні, амплітуди гармонічних складових не змінюються, а початкові фази змінюються на протилежні. Тобто амплітудний спектр є парною функцією, а фазовий – непарною функцією. Через це, аналізуючи спектральні характеристики сигналів, від'ємними частотами зазвичай нехтують.

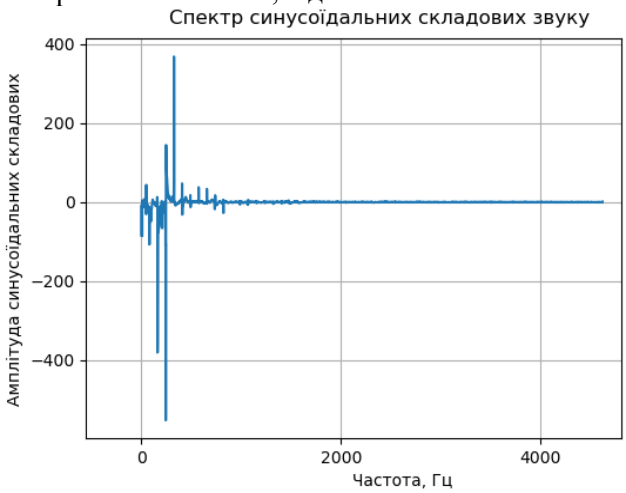

Рис.4 Амплітудно-частотний спектр звуку струни

Кожна складова спектра, записаного в комплексній формі, зображується на графіку двома лініями на частотах $\pm\omega_к$ і має амплітуду A/2, а постійна складова зображується такою ж, як і в спектрі, записаному у дійсній формі. Спектр сигналу, якщо його записувати через комплексні гармоніки, на відміну від спектра у дійсній формі, займе на осі частот не тільки область додатних, але й область від'ємних значень, але амплітуди усіх гармонічних складових, окрім складової з ω = 0, удвічі менші за амплітуди складових дійсного спектру.

Зрозуміло, від'ємні частоти не мають фізичного змісту, але, подання сигналу у комплексній формі робить його математичну обробку більш простою і зручною.

Зі спектру, представленого на рис.4 видно, два основних гармонічних коливання в спектрі струни з частотою 250 та 325 Гц . Коливання з частотою 325 Гц має амплітуду більше ніж в два рази більшу. Основна частота відповідає табличним значенням для найтовстішої (басової) гітарної струни.

Висновки

- Створено програмний код для дослідження індивідуальних характеристик звукових коливань, як сукупності гармонічних складових.
- Проведено первинну перевірку програми на предмет коректності роботи. Для цього синтезовано звукове коливання з трьох гармонічних складових та здійснено зворотні дії по аналізу звуку перетворенням Фур'є.
- Перевірка показала коректність у визначенні співвідношень між амплітудами гармонічних складових та відповідно співвідношень між частотами цих складових.

- Проведено Фур'є перетворення реального звуку, отриманий спектр звуку гітарної струни. Виявлено, що спектр містить спочатку додатні частоти від нуля до частоти Найквіста (Котельнікова), а потім і від'ємні. Виявлена також і симетрія додатних та від'ємних складових.
- Практична цінність роботи полягає в тому, що представлений у статті програмний код може бути застосований, як підсистема у прикладних програмах захисту, розпізнавання та керування системами різного характеру.
- Матеріал демонструє чисельні міжпредметні зв'язки математики та інформатики з різними розділами фізики та з курсом захисту інформації.

Список бібліографічного опису.

1. Головін М.Б. Аплікації з комп'ютерної фізики мовою Visual Python на прикладі моделювання силової взаємодії. / Н.А.Головіна, М.Б.Головін, А.А.Федонюк // Комп'ютерно-інтегровані технології: освіта, наука, виробництво" Луцьк, 2020. Випуск № 40 с.16-22. http://cit-journal.com.ua/index.php/cit/article/view/151
2. Нуссбаумер Г. Быстрое преобразование Фурье и алгоритмы вычисления сверток. — М.: Радио и связь, 1985.
3. Дж. Вандер Плас. Python для сложных задач. Наука о данных и машинное обучение = Python Data Science Handbook: Essential Tools for Working with Data. — Питер, 2017. — 576 с.
4. Тараненко Ю. Спектральный анализ сигналов нелинейных звеньев АСУ на Python https://habr.com/ru/post/325502/

References

1. Holovin M.B. Applications of computer physics via Visual Python language on the example of modeling of force interaction. / N.A.Holovina, M.B. Holovin, A.A.Fedoniuk // Computer integrated technologies: education, science, production" Lutsk, 2020. Issue № 40 c.16-22. http://cit-journal.com.ua/index.php/cit/article/view/151
2. Nussbaumer G. Fast Fourier transforms and algorithms for calculating convolutions. — M.: Radio and communication, 1985.
3. Gr.Vander Plas. Python for complex tasks. Data science and machine learning = Python Data Science Handbook: Essential Tools for Working with Data. — Petersburg, 2017. — 576 c.
4. Taranenko Yu. Spectral analysis of signals of non-linear links of the automated control system on Python. https://habr.com/ru/post/325502/