

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ТА ОБРОБЛЕННЯ ДАНИХ

Навчальний посібник

Харків «ХАІ» 2024

УДК 621.396.6
К63

Колектив авторів:

А. В. Попов, О. О. Каданер, Д. В. Колесніков, О. С. Інкарбаєва

Рецензенти: д-р техн. наук, проф. М. В. Руженцев,
канд. фіз.-мат. наук, доц. В. В. Мазуренко

Комп'ютерне моделювання та оброблення даних [Текст] : навч.
К63 посіб. / А. В. Попов, О. О. Каданер, Д. В. Колесніков, О. С. Інкарбаєва.
– Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харків. авіац.
ін-т», 2024. – 72 с.

ISBN 978-966-662-939-8

Наведено теоретичні основи, методи й алгоритми комп'ютерного моделювання й оброблення даних, що використовуються під час проектування електронної апаратури, які входять до програми підготовки бакалаврів за спеціальностями 172 «Телекомунікації та радіотехніка» і 163 «Біомедична інженерія», а також необхідні приклади програмної реалізації основних алгоритмів цифрового моделювання й оброблення даних з використанням мови програмування *Python*. Розглянуто загальні принципи моделювання, методи моделювання сигналів, впливів і систем на цифрових обчислювальних машинах. Особливу увагу приділено практичним питанням реалізації алгоритмів комп'ютерного моделювання.

Для студентів факультету радіоелектроніки, комп'ютерних систем та інфокомунікацій.

Іл. 65. Табл. 1. Бібліогр.: 10 назв

УДК 621.396.6

ISBN 978-966-662-939-8

© Колектив авторів, 2024
© Національний аерокосмічний
університет ім. М. Є. Жуковського
«Харківський авіаційний інститут», 2024

ВСТУП

Проектування радіотехнічних, радіоелектронних, телекомунікаційних, біотехнічних пристроїв та систем є складним науково-технічним процесом, методологічною основою якого є математичне моделювання з використанням системного підходу. Комп'ютерне моделювання є практичною реалізацією математичного моделювання з використанням засобів обчислювальної техніки. Комп'ютерне моделювання значно зменшує терміни й вартість проектування завдяки аналізу великої кількості варіантів і можливості досліджувати поведінку системи, яка проектується, при різних поєднаннях зовнішніх чинників, що не завжди вдається реалізувати під час натурних випробувань.

Згідно із системним підходом до проектування розробнику необхідно виявити всі зовнішні зв'язки проектного виробу, усі зовнішні чинники, що впливають на функціонування системи, і на основі аналізу фізики процесів, які відбуваються в системі, вибрати або розробити математичні моделі, що описують процеси як всередині системи, так і поза нею. Різноманітність фізичних процесів, що відбуваються в модельованих системах, потребує залучення для вирішення завдань комп'ютерного моделювання методів з різних областей математичної теорії.

Для застосування математичних моделей на практиці розробник має вибрати або розробити відповідні розрахункові методи, які стануть основою алгоритмів комп'ютерних обчислень. Велика кількість наявних методів обчислювальної математики потребує вибору найбільш адекватних для розв'язуваної задачі з урахуванням вимог як до точності обчислень, так і до обчислювальних витрат, зокрема місткості пам'яті й очікуваного часу обчислень для досягнення результату з необхідною точністю.

Реалізація алгоритмів у вигляді програмного забезпечення комп'ютерного моделювання потребує використання деякої мови програмування, яка надає відповідні програмні засоби як реалізації алгоритмів обчислень, так і візуалізації отриманих результатів. Вибір мови програмування – це завжди компроміс між складністю реалізації алгоритмів обчислень щодо витрат робочого часу програміста й по-тужності обчислювальних засобів і часом виконання програми моделювання. Застосування найсучасніших програмних комплексів комп'ютерного моделювання не завжди забезпечує успішне моделювання складних пристроїв і систем через різноманітність фізичних процесів, що відбуваються в модельованих системах і навколишньому середовищі. Тому в навчальному посібнику розглянуто методологію комп'ютерного моделювання й оброблення його результатів від простого до складного, принципи будування алгоритмів моделювання та їх програмної реалізації на прикладі типових завдань, що постають під час проектування радіоелектронних засобів.

1. МЕТОДОЛОГІЯ МОДЕЛЮВАННЯ РАДІОЕЛЕКТРОННИХ АПАРАТІВ І СИСТЕМ

1.1. Класифікація систем

Будь-яку технічну або біотехнічну систему можна подати у вигляді «чорного ящика», що виконує деяку функцію перетворення інформації в умовах зовнішніх впливів (рис. 1.1).

При такому поданні системи:

- $\vec{x}$ – множина вхідних впливів (сигналів), що надходять у систему;
- $\vec{z}$ – множина зовнішніх, пов'язаних з навколишнім середовищем, чинників, що впливають на систему і можуть бути контрольованими або неконтрольованими, керованими або некерованими, детермінованими або випадковими;

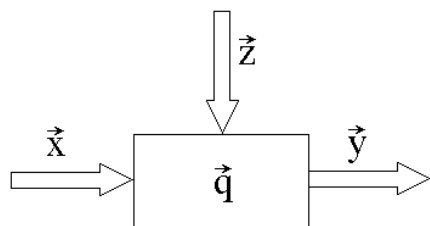


Рис. 1.1. Системотехнічна модель об'єкта проектування

- $\vec{q}$ – множина внутрішніх параметрів системи, у тому числі стану системи в попередні моменти часу;
- $\vec{y}$ – вектор вихідних параметрів – реакція системи на вхідні дії й зовнішні чинники:

$$\vec{y}(\rho) = A\{\vec{x}(\rho), \vec{q}(\rho, \vec{z})\}, \quad (1.1)$$

де $A(\bullet)$ – оператор системи (функція перетворення інформації, яка виконується системою); ρ – деякий функціонал, наприклад час t , частота f (при частотному поданні використовують також кутову частоту $\omega = 2\pi f$, оператор Лапласа $p = a + j\omega$), координата d (відстань, просторове положення) тощо.

Залежно від наявності даних про оператор $A(\bullet)$ системи поділяють на кілька типів:

а) система типу «чорний ящик» – оператор $A(\bullet)$ є повністю невідомим і відновлюється під час проєктування експериментальним шляхом унаслідок оброблення векторів $\vec{x}$, $\vec{y}$, $\vec{z}$;

б) система типу «сірий ящик» – оператор $A(\bullet)$ є відомим частково, наприклад, з точністю до параметрів $\vec{q}$, тобто відомим є аналітичний вираз $A = f(\vec{q})$, а параметри $\vec{q}$ є невідомими;

в) система типу «білий ящик» – відомими є вигляд оператора $A(\bullet)$ і значення параметрів $\vec{q}$.

За типом оператора $A(\bullet)$ системи поділяють залежно від сукупності їх властивостей, а саме:

- фізична реалізованість;
- стохастичність;
- векторність;
- лінійність;
- інерційність;
- стаціонарність;
- розподіленість;
- автономність та ін.

У фізично реалізованих системах вихідна реакція системи формується за попереднім і поточним значенням вхідного впливу (рис. 1.2, а). У фізично нереалізованих системах для визначення вихідної реакції системи в поточний момент часу необхідним є знання майбутніх значень вхідного сигналу (рис. 1.2, б).

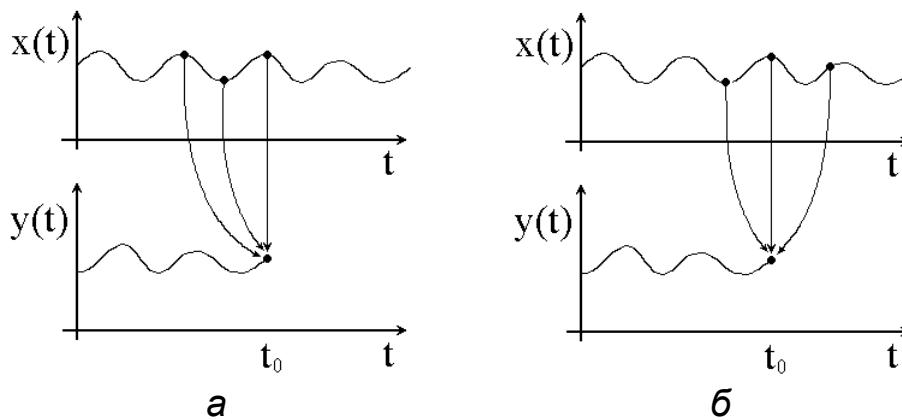


Рис. 1.2. Формування вихідного сигналу в системах

Для детермінованих систем існує однозначна відповідність між вхідним і вихідним сигналами $y = A(\bar{x})$. В імовірнісних (стохастичних) системах значення вихідного сигналу можна передбачити за значенням вхідного тільки з деякою певною ймовірністю $P(\bar{y}/\bar{x}) = f(A(\bar{x}))$. Причинами цієї невизначеності можуть бути:

- змінення внутрішніх параметрів $\bar{q}$ за випадковим законом;
- випадкове змінення структури системи (наприклад, раптова відмова), що є еквівалентним випадковому зміненню оператора системи $A(\cdot)$;
- вплив зовнішніх чинників випадкового характеру $\bar{z}$.

Розрізняють також одновимірні (скалярні) і багатовимірні (векторні) системи. В одновимірних системах вхідний $x(t)$ і вихідний $y(t)$ процеси – скалярні величини: $y = A(x)$. Систему вважають багатовимірною, якщо хоча б один з процесів $\bar{x}(t)$, $\bar{y}(t)$ є векторним. Система може бути векторною за входом, наприклад, $y(t) = x_1(t) + x_2(t)$, за виходом,

наприклад, $y_1(t) = ax(t)$, $y_2(t) = bx(t)$, або мати багато входів і виходів, будь-яким чином пов'язаних один з одним (рис. 1.3).

Лінійні системи на відміну від нелінійних підпорядковуються принципу суперпозиції:

$$y = A \left\{ \sum_k (a_k x_k) \right\} = \sum_k (a_k A \{ x_k \}). \quad (1.2)$$

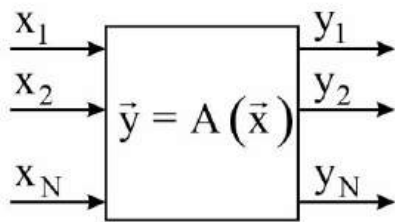


Рис. 1.3. Багатовимірна система

У безінерційних (статичних) системах вихідний сигнал у певний момент часу залежить тільки від значення вхідних сигналів у той же самий момент часу. Інерційні (динамічні) системи мають «пам'ять про попереднє»: значення вихідного сигналу в певний момент часу залежить від значень вхідного сигналу в той же самий момент і від значень вхідного й

вихідного сигналів у попередні моменти часу.

У стаціонарних (інваріантних) системах зсув вхідного сигналу в часі призводить до такого ж зсуву вихідного сигналу $x(t + \tau) \rightarrow y(t + \tau)$, тоді як у нестаціонарних системах вихідний сигнал залежить від моменту подання вхідного сигналу, тобто $A = A(\cdot, t)$.

У системах з розподіленими параметрами реакція системи залежить не тільки від часу, але й від координат. Така система описується диференціальними рівняннями в часткових похідних за часом і простором. Системи із зосередженими параметрами описуються звичайними диференціальними рівняннями, і їх реакція залежить тільки від часу.

Автономними називають системи, які не зазнають вхідного впливу: $x(t) \equiv 0$, $y(t) \equiv A(\vec{q}, \vec{z})$. Прикладом такої системи може бути генератор сигналів. У неавтономних системах їх реакція визначається не тільки внутрішнім станом, а й вхідним сигналом: $y(t) = A(\vec{q}, x(t))$.

1.2. Розрахункові моделі й комп'ютерне моделювання

Моделювання – це процес пізнання об'єктивної дійсності, при якому досліджуваний об'єкт (оригінал) свідомо замінюється іншим об'єктом (моделлю), який є подібним до оригіналу. Ступінь відповідності моделі реальним процесам, що відбуваються в системі, яка моделюється, називають адекватністю моделі.

Процес проєктування системи з використанням моделі полягає в дослідженні моделі з подальшим перенесенням отриманих даних на модельований об'єкт.

Під час проєктування застосовують як матеріальні моделі, наприклад просторові (використовують геометричну подібність моделі оригіналу) або фізичні (що базуються на подібності фізичних принципів моделі й оригіналу), так і абстрактні – знакові, аналітичні, імітаційні, імовірнісні тощо.

Знакові моделі зазвичай являють собою креслення і схеми проєктивного виробу, наприклад схеми електричних принципів.

Аналітичні моделі будуються на основі деякої теорії з використанням математичних конструкцій (функцій) і є основою для розрахункових моделей.

Кінцевий результат математичного моделювання – формальні співвідношення для кількісного або якісного аналізу або комп'ютерного моделювання.

Імітаційні моделі в основі мають алгоритм, що імітує поведінку об'єкта. Алгоритм можна побудувати на базі або аналітичних моделей, або знакових (структурних схем, графів, топологічних моделей).

Імовірнісні моделі застосовують:

- якщо зовнішні впливи є незакономірними;
- якщо є суб'єктивні чинники, пов'язані з участю людини;
- якщо моделюються чинники технологічних процесів, що мають розкид параметрів;
- якщо є незворотні (старіння, зношення) або зворотні процеси, такі як розрегулювання, наведення, перешкоди.

Одним з головних елементів проєктування радіоелектронних засобів є проведення розрахунків, при виконанні яких використовуються математичні моделі фізичних процесів, що відбуваються в проєктованому виробі. Такі моделі називають розрахунковими. Класифікацію розрахункових моделей зображено на рис. 1.4.

Розрахункові електричні моделі дають змогу:

- повністю або частково описати електричні процеси, що відбуваються в схемі під час її функціонування;
- отримати із заданою точністю як режимні, так і функціональні електричні характеристики об'єкта, що моделюється.

У цих моделях має бути враховано:

- паразитні провідності, ємності, індуктивності й інші параметри, що відображають вплив конструкції на електричні процеси;
- електричні наведення від інших електронних засобів, навколишніх об'єктів і середовища.

Розрахункові моделі механічних систем:

- дають змогу повністю або частково відображати механічні процеси в конструкції, пов'язані з виникненням механічних деформацій при зовнішніх впливах;

– забезпечують отримання із заданою точністю необхідних механічних характеристик апаратури (частотних – для вібрацій, часових – для ударних впливів, статичних – для лінійних прискорень і т. д.);

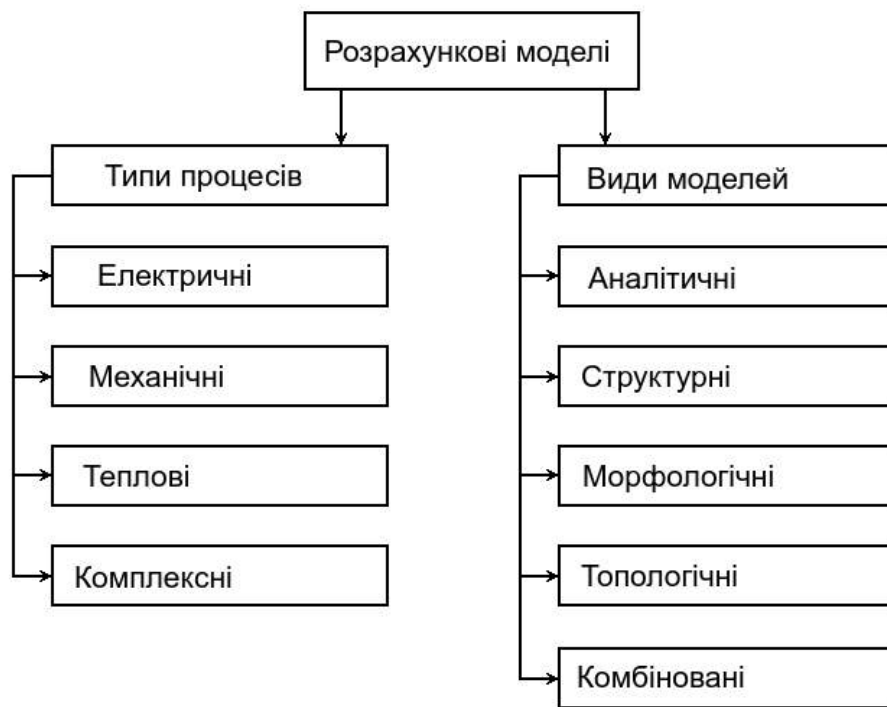


Рис. 1.4. Класифікація розрахункових моделей

У цих моделях має бути враховано:

- розподіленість маси несної конструкції й анізотропність механічних властивостей деталей конструкції, ефекти внутрішнього тертя в деталях при їх деформаціях;
- реальну кінцеву жорсткість кріплення радіоелементів до несних конструкцій (друкованих плат).

Теплові розрахункові моделі:

- описують процеси перенесення тепла, пов'язані з теплообміном в конструкції під впливом зовнішнього середовища;
- дають змогу розраховувати процеси тепловиділення в радіоелементи, тепловий опір під час передання тепла в навколишнє середовище, дію систем охолодження й термостатування;
- ураховують різні механізми теплоперенесення, такі як кондукція, конвекція і випромінювання;
- мають відображати розподіленість маси конструкції як теплової ємності і анізотропію теплових властивостей радіодеталей.

Згідно з вимогами системного підходу до проектування математичні моделі мають описувати повний комплекс фізичних процесів, що відбуваються як усередині апаратури, так і поза нею, як електричних, так і механічних, теплових і т. д. При цьому вхідними впливами $\vec{x}$ (див. рис. 1.1) можуть бути електричні сигнали, електромагнітні завади, механічні вібрації тощо, які безпосередньо визначають реакцію системи $\vec{y}$. Зовнішніми впливами $\vec{z}$ описують множину зовнішніх чинників, пов'язаних з навколишнім середовищем, які так чи інакше впливають на роботоздатність приладу (наприклад, температура навколишнього

середовища, механічні вібрації й удари, зовнішні електромагнітні поля тощо).

Стан системи описується при моделюванні вектором параметрів системи $\vec{q}$, який може містити як електричні параметри (опір, коефіцієнт підсилення, напруга і струм), так і параметри системи, які мають іншу фізичну природу (температура елементів системи, механічні навантаження на конструкцію тощо).

У зв'язку з різноманітністю фізичних явищ, що відбуваються в електронних системах, залежністю їх від часу, просторового положення тощо, у межах цього курсу всі сигнали, впливи, зовнішні чинники, а також реакція системи на них будуть розглядатися як деякі фізичні процеси, що описуються єдиними математичними моделями незалежно від їх фізичної природи.

Таким чином, комп'ютерне моделювання ґрунтується на математичних моделях процесів і систем. Для реалізації процесу моделювання з допомогою засобів обчислювальної техніки необхідно вирішити кілька завдань:

1. Виконати аналіз фізичних процесів у системі, що моделюється.
2. Підібрати адекватні розрахункові моделі для опису як процесів, що відбуваються всередині модельованої системи, так і зовнішніх впливів і чинників, що впливають на об'єкт моделювання.
3. Розробити алгоритми моделювання, які відповідають вибраним розрахунковим моделям, визначити параметри моделювання, що забезпечують адекватність моделювання й необхідну точність.
4. Розробити програму (або комплекс програм), яка реалізує розроблені алгоритми однією із мов програмування високого рівня (C++, C#, Python, Pascal, Fortran, Java тощо) або з використанням будь-якого програмного комплексу автоматизованого проєктування (CAD), наприклад MathCAD, MatLab, Simulink тощо.
5. Змоделювати процеси, що впливають на об'єкт моделювання.
6. Змоделювати реакції об'єкта (системи) на зовнішні впливи.
7. Обробити дані моделювання, за результатами якого (графічними залежностями, зображеннями, таблицями тощо), розробник системи може зробити однозначні висновки.

У межах цієї навчальної дисципліни буде розглянуто типові моделі, методи, методики і їх програмну реалізацію для вирішення перелічених вище завдань комп'ютерного моделювання радіотехнічних, радіоелектронних, телекомунікаційних, біотехнічних апаратів і систем.

1.3. Програмні засоби комп'ютерного моделювання

Для вирішення завдань комп'ютерного моделювання в цій навчальній дисципліні було вибрано мову програмування *Python*, що є досить простою за синтаксисом і має потужні засоби оброблення й візуалізації даних. *Python* – інтерпретована скриптова мова програмування, тобто текст програми перекладається на послідовність команд процесора під час виконання програми. У цьому курсі використовується пакет *Anaconda*, який містить інтерпретатор мови *Python3*, набір найбільш часто використовуваних бібліотек і середовище розроблення й виконання програм, що запускається в браузері.

Особливістю мови програмування *Python* є те, що в пам'яті комп'ютера зберігаються дані, а так звані змінні (імена даних) лише вказують на область зберігання даних. Назва змінної має починатися з алфавітного символу або знака підкреслення і може містити алфавітно-цифрові символи і знак підкреслення. Назва змінної не має збігатися з назвою ключових слів мови *Python*, таких як `and`, `as`, `assert`, `break`, `class`, `continue`, `def`, `del`, `elif`, `else`, `except`, `False`, `finally`, `for`, `from`, `global`, `if`, `import`, `in`, `is`, `lambda`, `None`, `nonlocal`, `not`, `or`, `pass`, `raise`, `return`, `True`, `try`, `while`, `with`, `yield`.

Імена змінних пов'язуються з даними з допомогою символу «=», наприклад:

```
a = 5; name = "Jack".
```

Типи даних, які використовуються в курсі:

1) *int* – ціле число або 0, наприклад: 3, -7, 123;

2) *float* – число з рухомою точкою, тобто дійсне, наприклад: 314.15; часто використовується експоненціальний запис: 3.1415E2;

3) *complex* – комплексні числа, записуються, наприклад, як $1 + 2j$, де $j = \sqrt{-1}$, і можуть створюватися як `x = complex(a,b)`, де *a* – дійсна частина, *b* – уявна; з допомогою методів `real` та `imag` можна отримати відповідні частини, наприклад:

```
x = 5.0 - 3.0j.
```

`x.imag` дасть результат -3.0, `x.real` – 5.0;

4) *string* – текстові рядки, наприклад: `name = "Jack"`.

Python – це мова з неявною сильною динамічною типізацією. Неявна означає, що при оголошенні змінної необов'язково вказувати її тип; сильна – не можна здійснювати операції над даними різних типів, необхідно відстежувати в програмі типи даних, що використовуються в операціях; динамічна – тип змінної визначається під час виконання програми.

Арифметичні операції з числами, які використовуються:

1) додавання: $a = b + c$; додавати можна як числа, так і змінні, присвоювати як новій змінній, так і тій, що використовується в операції, наприклад: $a = a + 5$, аналог: $a += 5$;

2) віднімання: $a = b - c$;

3) множення: $a = b * c$;

4) ділення: $a = b / c$;

5) отримання цілої частини від ділення: $a = b // c$;

6) отримання дрібної частини від ділення: $a = b \% c$;

7) піднесення до степеня: $a = b ** c$.

Не всі необхідні операції реалізуються через убудовані оператори. Для деяких (наприклад, для добування кореня) необхідно імпортувати бібліотеку, яка містить алгоритм обчислення необхідних функцій. Основні математичні функції знаходяться в бібліотеці *math*.

Підключення бібліотеки виконується в такий спосіб:

```
import math.
```

Цей рядок пишеться на самому початку скрипту. Якщо не потрібна вся бібліотека цілком, а тільки окремі функції з неї, то можна імпортувати тільки їх, наприклад:

```
from math import sqrt, pi, sin, cos, atan.
```

Різні бібліотеки, їх склад і необхідні в цьому курсі функції буде розглянуто в міру виникнення необхідності їх використання.

Розглянутих інструментів *Python* достатньо для моделювання електричних схем при поданні на їх вхід гармонійної напруги. Як приклад вирішимо класичну задачу теорії кіл – розрахунок напруги на виході електричного кола, заданого схемою електричною принциповою (рис. 1.5).

З огляду на теорію моделювання схема на рис. 1.5 є знаковою моделлю об'єкта «електричний пристрій», що має вхід (вузол зліва) і вихід (вузол справа) згідно з правилами зображення схем електричних принципових. Знаковій моделі ставляться у відповідність розрахункові моделі комплексних опорів елементів схеми, що вивчалися в курсі «Теорія кіл»: $Z_R = R$, $Z_L = j\omega L$, $Z_C = 1 / j\omega C$, де ω – кутова частота $\omega = 2\pi f$, яка відповідає частоті f вхідного гармонічного сигналу вигляду

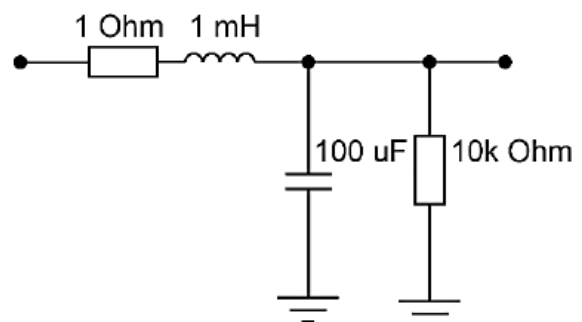


Рис. 1.5. Схема електричного кола

$$s(t) = A \cos(\omega t + \varphi),$$

з амплітудою A , початковою фазою φ , що змінюється в часі t .

Для прикладу задамо амплітуду $A = 2$ В, початкову фазу $\varphi = 0$, частоту $f = 500$ Гц. Перетворимо схему (див. рис. 1.5) на еквівалентну (рис. 1.6) за правилами теорії кіл:

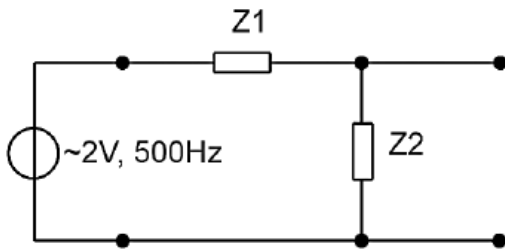


Рис. 1.6. Еквівалентна схема

– опір послідовного з'єднання елементів

$$Z_1 = Z_{R1} + Z_L;$$

– опір паралельного з'єднання

$$Z_2 = \frac{Z_{R2}Z_C}{Z_{R2} + Z_C};$$

– до входу під'єднаємо джерело напруги амплітудою 2 В і частотою 500 Гц.

Тоді за законом Ома через коло послідовного з'єднання опорів Z_1 і Z_2 буде проходити струм $I = U / (Z_1 + Z_2)$, що створить на опорі Z_2 спад напруги $U = I Z$, який і буде шуканою вихідною напругою електричного кола.



Рис. 1.7. Іконка Jupyter Notebook

Для виконання комп'ютерного моделювання запусимо інтерактивну середу *Jupyter Notebook* з пакета програм *Anaconda* шляхом запуску програми з іконкою (рис. 1.7) і створимо новий процес *Notebook Python3*, як показано на рис. 1.8. У новому вікні браузера (рис. 1.9) у поле введення введемо текст програми мовою *Python*, запуск якої здійснюється натисканням кнопки «►».

Результат виконання програми показано на рис. 1.10. Таким чином, отримано напругу на виході пристрою $U = (0.2 - 6.34j)$. Для правильної інтерпретації отриманих даних слід обробити дані моделювання. Для комплексної напруги U необхідно визначити амплітуду і фазу, як показано на рис. 1.11. Результатом моделювання є амплітуда вихідної напруги 6,35 В і її фаза $-87,6^\circ$, що дає змогу зробити такий висновок за результатами моделювання: сигнал на виході пристрою

$$s(t) = 6,35 \cos(2\pi 500t - 87,6^\circ).$$

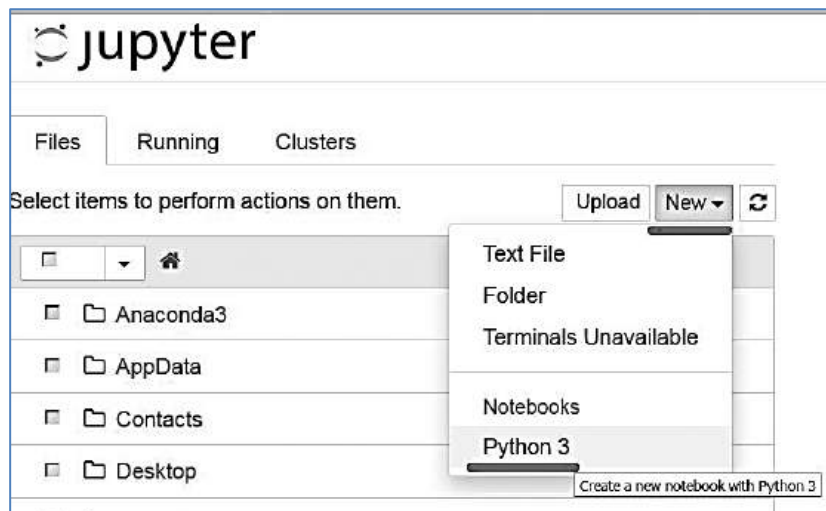


Рис. 1.8. Створення нового *Notebook Python3*

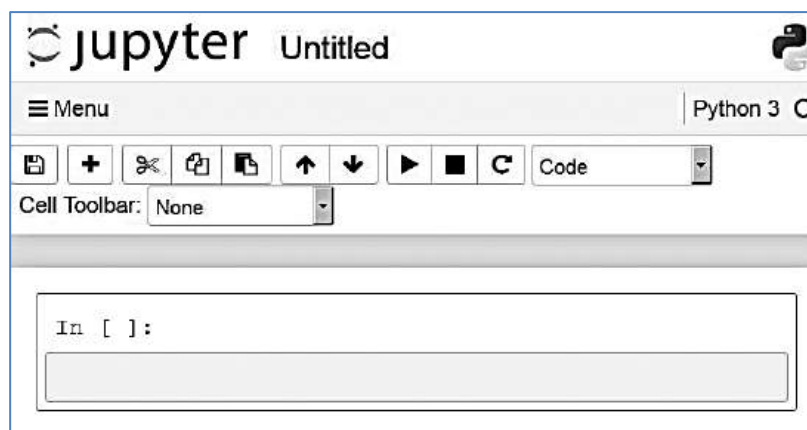


Рис. 1.9. Інтерактивне середовище програмування *juPyter Notebook*

```
from math import sqrt, pi, atan
R1=1.0; L=1.0e-3; C=100.0e-6; R2=10.0e3
f=500.0; w=2.0*pi*f; A=2.0
ZR1=R1+0.0j; ZR2=R2+0.0j; ZL=1j*w*L; ZC=1.0/(1j*w*C)
Z1=ZR1+ZL; Z2=ZR2*ZC/(ZR2+ZC); I=A/(Z1+Z2); U=I*Z2
print(U)
```

(0.26526831893245956-6.348754249870358j)

Рис. 1.10. Текст програми й результат її виконання

abs (U)
6.354293651186914
atan (U.imag/U.real) *180/pi
-87.60741727876159

Рис. 1.11. Результат оброблення даних моделювання

1.4. Комп'ютерне моделювання як метод дослідження об'єктів

Як зазначалося в підрозд. 1.1, під час проектування об'єкт розроблення подають у вигляді «чорного ящика» (див. рис. 1.1), що має вхід і вихід і описується оператором передання $A = f(\vec{q})$, де $\vec{q}$ – вектор параметрів об'єкта. Для радіотехнічних, радіоелектронних, телекомунікаційних, біотехнічних апаратів і систем, а також для опису механічних, біологічних, теплових та інших систем як оператор передання використовується комплексна передатна функція пристрою, об'єкта, системи.

За означенням комплексна передатна функція пов'язує вхідний вплив з вихідною реакцією об'єкта:

$$K(j\omega) = \frac{U_{ВИХ}(j\omega)}{U_{ВХ}(j\omega)}.$$

Дослідження комплексної передатної функції і її залежності від параметрів $\vec{q}$ об'єкта створює основу для комп'ютерного проектування радіоелектронних пристроїв.

Розглянемо конкретний приклад дослідження об'єкта (електричного кола) схему якого зображено на рис. 1.5, методом комп'ютерного моделювання. Комплексну передатну функцію можна визначити як відношення вихідної комплексної напруги до вхідної, тому застосуємо методи теорії кіл для її визначення. Для цього скористаємося програмою, показаною на рис. 1.10.

Для визначення залежності $K(j\omega)$ від частоти необхідно створити таблицю частот f з рівномірним кроком Δf , наприклад, від 0 до 1000 Гц з кроком 0,5 Гц. Для цього необхідно створити список *Python*:

```
f = [i*df for i in range(n)],
```

де $n = 1000/0,5 = 2000$ – кількість точок за частотою для розрахунку $K(j\omega)$;

i – номер точки.

Тоді програму розрахунку електричного кола необхідно повторити 2000 разів з допомогою оператора циклу *for* :

```
for i in range(n):  
    # Розрахунок напруги на виході пристрою
```

Для кожної точки за частотою будемо обчислювати значення кутової частоти:

```
w = 2*pi*f[i].
```

Оскільки на частоті, що дорівнює нулю, при обчисленні опору ємності відбувається ділення на нуль, унесемо корекцію в алгоритм: замість 0 у формулу підставимо дуже маленьке значення частоти, наприклад у 1000 разів менше від кроку за частотою. Це реалізується з допомогою умовного оператора **if**.

Для нагромадження даних моделювання використовуємо «порожній» список **K = []**, який будемо заповнювати у міру обчислення відношення вихідного сигналу до вхідного методом **append**.

Після виконання програми отримаємо дані моделювання у вигляді списку комплексних чисел (2000 шт.), як показано на рис. 1.12.

```
from math import sqrt, pi, atan
R1=1.0; L=1.0e-3; C=100.0e-6; R2=10.0e3
f=500.0; A=2.0; n=2000; df=0.5; f=[]; K=[]
for i in range(n):
    f.append(i*df); w=2.0*pi*f[i]
    if(w==0): w=df*2*pi/1000.0
# Розрахунок напруги на виході пристрою
ZR1=R1+0.0j; ZR2=R2+0.0j; ZL=1j*w*L; ZC=1.0/(1j*w*C)
Z1=ZR1+ZL; Z2=ZR2*ZC/(ZR2+ZC); I=A/(Z1+Z2); U=I*Z2
# Розрахунок комплексної передатної функції
K.append(U/A)
print(K)

[(0.999900009999888-3.144110539372853e-07j),
(0.999900897898896-0.0003144111288456708j),
(0.999903561
```

Рис.1.12. Програма і результат розрахунку
комплексної передатної функції

Оброблення даних моделювання в цьому завданні полягає в обчисленні амплітудно-частотної (АЧХ) і фазо-частотної (ФЧХ) характеристик електричного кола й поданні їх у графічному вигляді, зручному для аналізу. АЧХ є модулем $K(j\omega)$, а ФЧХ – аргументом $K(j\omega)$, таблиці відповідних чисел можуть нагромаджуватися під час обчислень у списках A4X і F4X.

Для будування графіків отриманих даних скористаємося функцією **plot** з бібліотеки **matplotlib** (модуль **pylab**). Будування «сітки» на графіку забезпечує функція **grid**.

Нове «полотно» для графіка відкриває функція **figure**. Інструкція **%matplotlib inline** забезпечує включення графіків у поле документа *Python*

Notebook. Повний текст програми й результати оброблення даних моделювання показано на рис.1.13–1.15.

```
from math import sqrt, pi, atan
from matplotlib.pyplot import plot, grid, figure
%matplotlib inline
R1=1.0; L=1.0e-3; C=100.0e-6; R2=10.0e3
f=500.0; A=2.0; n=2000; df=0.5; f=[]; A4X=[]; P4X=[]
for i in range(n):
    f.append(i*df); w=2.0*pi*f[i]
    if(w==0): w=df*2*pi/1000.0
    ZR1=R1+0.0j; ZR2=R2+0.0j; ZL=1j*w*L; ZC=1.0/(1j*w*C)
    Z1=ZR1+ZL; Z2=ZR2*ZC/(ZR2+ZC); I=A/(Z1+Z2); U=I*Z2
    K=(U/A); A4X.append(abs(K))
    P4X.append(180*atan(K.imag/K.real)/pi)
fig_1=figure()
fig_1.suptitle('A4X', fontsize=12) # назва вікна
plot(f,A4X); grid()
fig_2=figure(2)
fig_2.suptitle('P4X', fontsize=12) # назва вікна
plot(f,P4X); grid()
```

Рис.1.13. Аналіз АЧХ і ФЧХ електронної схеми

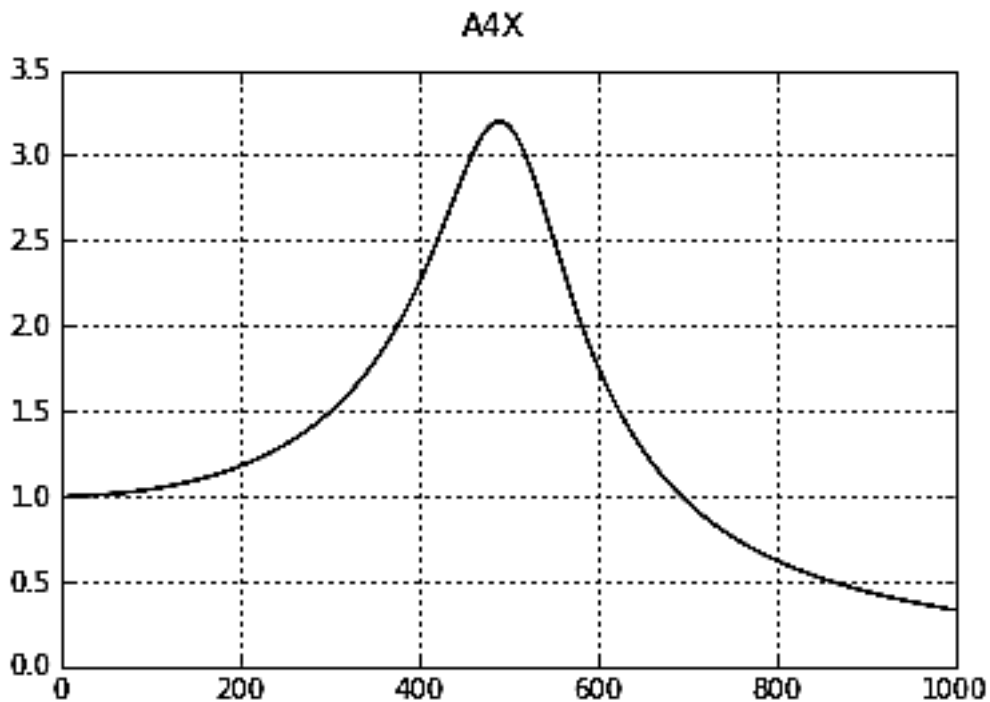


Рис.1.14. АЧХ електронної схеми

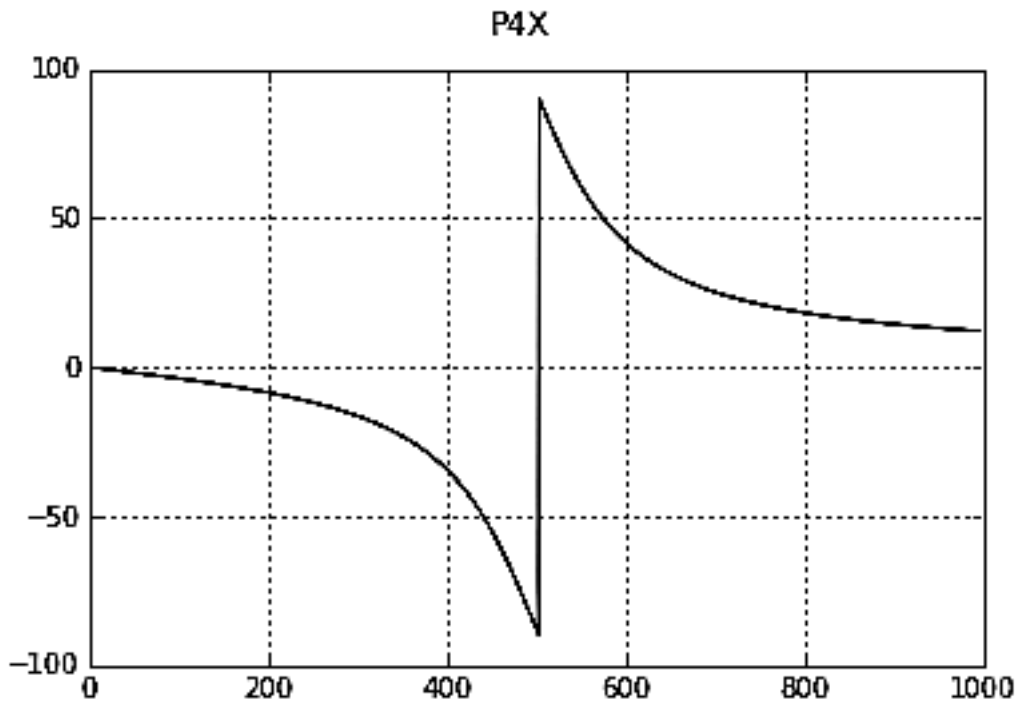


Рис. 1.15. ФЧХ електронної схеми

Контрольні запитання

1. Що таке комп'ютерне моделювання? У чому полягає різниця між комп'ютерним моделюванням і математичним?
2. Класифікація систем з огляду на комп'ютерне моделювання.
3. Які види моделей використовуються при комп'ютерному моделюванні?
4. За яким типом моделі будується схема електрична принципова: «чорний ящик» чи «білий ящик»?
5. До якого типу моделей належить схема електрична принципова?
6. Чим відрізняються детерміновані моделі від імовірнісних?
7. Які типи даних використовуються при комп'ютерному моделюванні?
8. Якими є особливості використання мови програмування *Python* при комп'ютерному моделюванні?
9. Що таке комплексні числа і як вони зберігаються в пам'яті комп'ютера?
10. Отримано результат обчислення змінної напруги з частотою 100 Гц на виході пристрою: $2 - 4j$. Як його інтерпретувати?

2. КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ СИГНАЛІВ І ПРОЦЕСІВ

Процесом називають послідовне змінення станів у розвитку будь-якого явища. У технічній літературі під процесом у більшості випадків розуміють змінення в часі або просторі деякої фізичної величини й позначають її як функцію часу t або іншого безперервного параметра (наприклад, координат об'єкта $\{x, y\}$): $s(t)$, $b(x, y)$ тощо).

Усі спостережувані процеси, що характеризують фізичні явища, можна поділити на детерміновані й випадкові. Детермінованими називають процеси (сигнали, впливи, реакції систем тощо), які можна описати математичними співвідношеннями. Значення детермінованого сигналу в будь-який момент часу можна визначити точно. Процеси, що описують детерміновані явища, можуть бути періодичними й перехідними. На проміжному становищі перебувають так звані майже періодичні процеси, що мають властивості як періодичних, так і перехідних процесів (рис. 2.1).

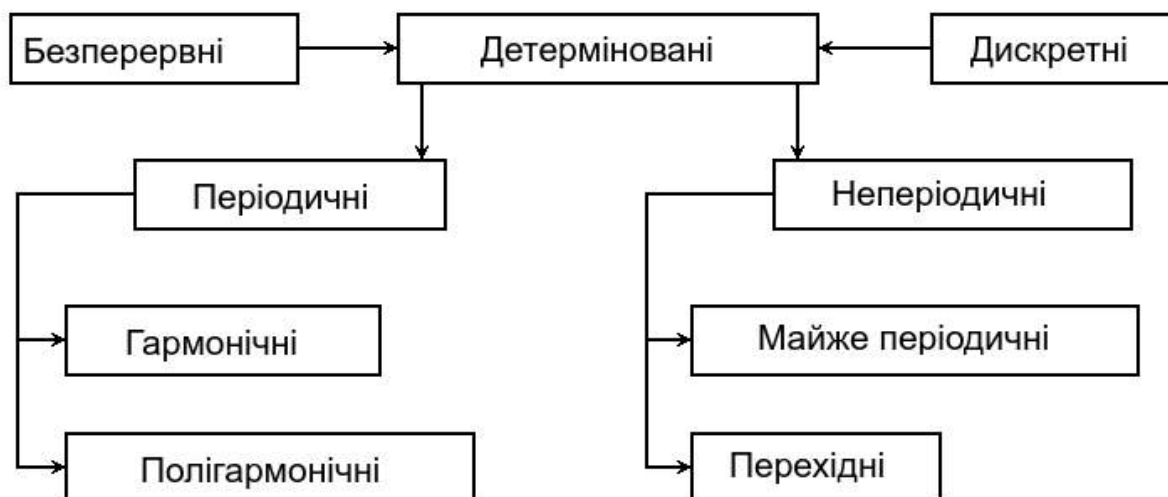


Рис. 2.1. Класифікація детермінованих процесів

2.1. Математичні моделі детермінованих процесів

Періодичними називають процеси, які можна описати функцією часу (або координат). Ці процеси точно повторюють свої значення через однакові проміжки часу T_1 :

$$x(t) = x(t \pm kT_1), \quad k = 1, 2, 3, \dots$$

Інтервал часу, за який відбувається одне повне коливання, називають періодом процесу $f_1 = 1/T_1$. Кількість повторень процесу (кількість циклів) називають основною частотою процесу $f_1 = 1/T_1$.

Гармонічними називають періодичні процеси, які можна описати функцією часу

$$x(t) = X \cos(\omega_1 t + \varphi), \quad \omega_1 = 2\pi f_1, \quad (2.1)$$

де X – амплітуда; f_1 – частота; φ – початкова фаза (у радіанах); $x(t)$ – значення функції в момент часу t .

Гармонічний процес можна подати функцією часу, як показано на рис. 2.2, або функцією частоти (так званим спектральним поданням). За частотним спектром (рис. 2.3) можна визначити амплітуду X і частоту f_1 періодичного сигналу. Частотне подання сигналу є значно зручнішим від його часового графіка, однак не містить інформації про початкову фазу колювання φ .

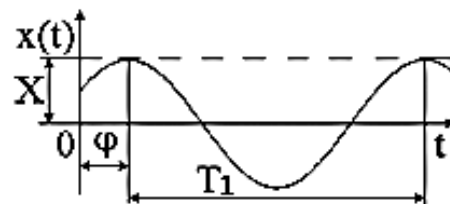


Рис. 2.2. Гармонічний процес

Відомо, що будь-яку синусоїду можна розкласти на суму двох так званих квадратурних компонент:

$$x(t) = X \cos(\omega_1 t + \varphi) = X \cos(\varphi) \cos(\omega_1 t) + X \sin(\varphi) \sin(\omega_1 t), \quad (2.2)$$

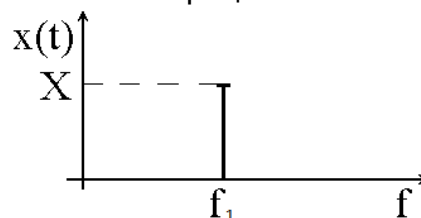


Рис. 2.3. Спектральне подання гармонічного процесу

де $\omega_1 = 2\pi f_1$.

У цьому випадку гармонічний процес зображується двома графіками (рис. 2.4, а) або двома спектральними діаграмами (рис. 2.4, б), які повністю характеризують усі параметри процесу, крім амплітуди й фази процесу в явному вигляді. Таке відображення процесів (рис. 2.4, б) широко використовується при комп'ютерному моделюванні детермінованих сигналів і впливів.

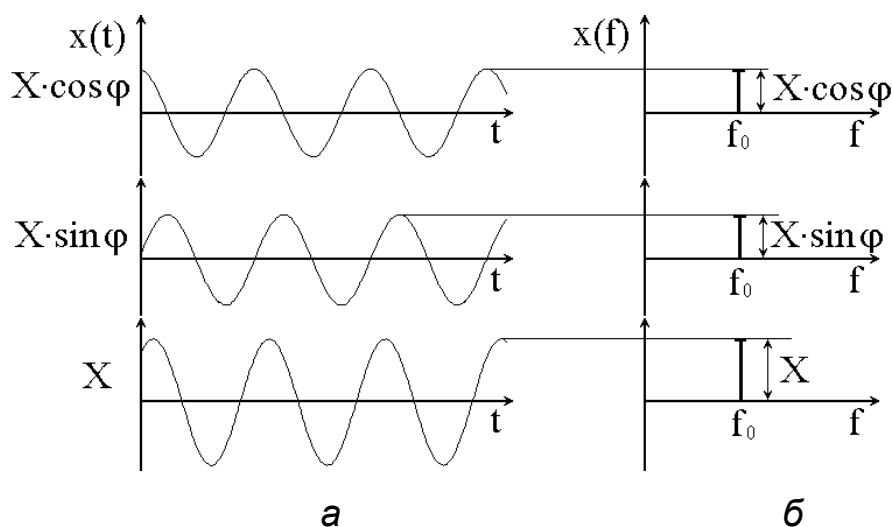


Рис. 2.4. Подання гармонічного процесу у вигляді суми двох компонент:
а – часове; б – частотне

Більш конструктивним є подання гармонічного процесу з допомогою комплексних векторів. Згідно з теоремою Ейлера $\cos \alpha + j \sin \alpha = e^{j\alpha}$, де $j = \sqrt{-1}$. Будь-який гармонічний процес можна подати в комплексній формі:

$$\dot{x}(t) = X e^{j(\omega t + \varphi)} = X_R \cos(\omega t) + j X_I \sin(\omega t)$$

або

$$\dot{x}(t) = X_R(t) + j X_I(t), \quad (2.3)$$

де $X_R = X \cos \varphi$ – реальна частина комплексного числа (2.3), $X_I = X \sin \varphi$ – уявна його частина, причому

$$X = \sqrt{X_R^2 + X_I^2}, \quad \varphi = \arctg(X_I / X_R). \quad (2.4)$$

Подання гармонічного сигналу у вигляді комплексного числа принципово нічим не відрізняється від розкладання формули (2.2) (див. рис. 2.4), але дає змогу визначити поняття комплексного спектра, який складається з двох діаграм – амплітудного спектра, що характеризує амплітуду процесу на певній частоті, і фазового (рис. 2.5).

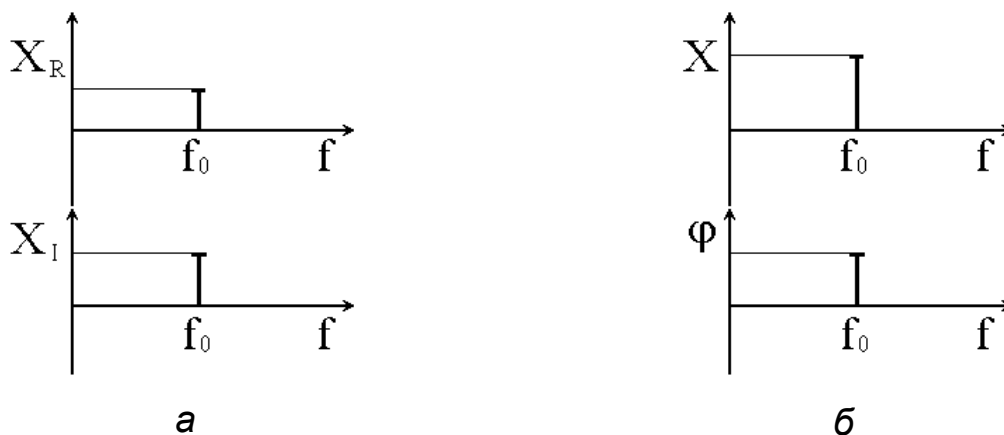


Рис. 2.5. Подання процесу у вигляді комплексного спектра:
 а – реальна й уявна частини процесу;
 б – амплітудний і фазовий спектри

Таким чином, будь-який гармонічний сигнал можна подати у вигляді суми двох гармонічних складових $\cos(\omega t)$ і $\sin(\omega t)$ з однаковою частотою ω і амплітудами X_R і X_I , які точно визначають амплітуду й початкову фазу гармонічного процесу згідно з формулою (2.4).

Полігармонічний процес (рис. 2.6) можна подати у вигляді

$$x(t) = \sum_{k=0}^n X_k \cos(k\omega_1 t + \varphi_k), \quad (2.5)$$

де $\omega_1 = 2\pi f_1$, f_1 – основна частота. У загальному випадку $n \rightarrow \infty$.

Очевидно, що гармонічний процес є окремим випадком полігармонічного при $n=1$.

Процес $x(t)$ можна подати у вигляді суми квадратурних компонент:

$$x(t) = \frac{X_0}{2} + \sum_{k=1}^n [X_{Rk} \cos(k\omega_1 t) + X_{Ik} \sin(k\omega_1 t)]. \quad (2.6)$$

Вираз (2.6) називають рядом Фур'є, причому

$$X_{Rk} = \frac{2}{T_1} \int_0^{T_1} x(t) \cos(k\omega_1 t) dt, \quad k = 0 \dots \infty,$$

$$X_{Ik} = \frac{2}{T_1} \int_0^{T_1} x(t) \sin(k\omega_1 t) dt, \quad k = 0 \dots \infty.$$

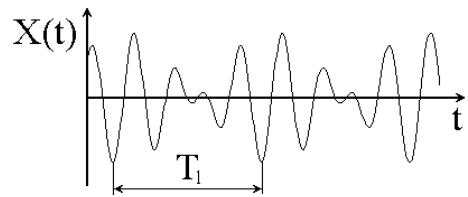


Рис. 2.6. Полігармонічний процес

У комплексній формі полігармонічний процес можна подати як

$$X(t) = X_0 + \sum_{k=1}^{\infty} X_k e^{j(k\omega_1 t - \varphi_k)}. \quad (2.7)$$

Можливим є й інший спосіб запису ряду Фур'є для полігармонічного процесу:

$$x(t) = X_0 + \sum_{k=1}^{\infty} X_k \cos(k\omega_1 t - \varphi_k). \quad (2.8)$$

Очевидно, що полігармонічний процес складається з постійної компоненти X_0 і в загальному випадку нескінченної кількості синусоїдальних компонент, які називають гармоніками (або гармонічними складовими) з амплітудами X_k і початковими фазами φ_k , причому частоти всіх гармонік є кратними основній частоті процесу f_1 (рис. 2.7).

Іноді компоненти з основною частотою в спектрі полігармонічного процесу може й не бути. Наприклад, періодичний процес формується внаслідок змішування трьох синусоїд з частотами 60, 75 і 100 Гц. Найбільший спільний дільник цих трьох чисел дорівнює 5 Гц, тому основна частота полігармонічного процесу становитиме 5 Гц, а період результтивного процесу

$$T_1 = 1/f_1 = 1/5 \text{ Гц} = 0,2 \text{ с.}$$

Отже, при розкладанні в ряд Фур'є всі значення x_k дорівнюватимуть нулю, крім $k = 12, 15, 20$ (рис. 2.8).



Рис. 2.7. Комплексний спектр полігармонічного процесу

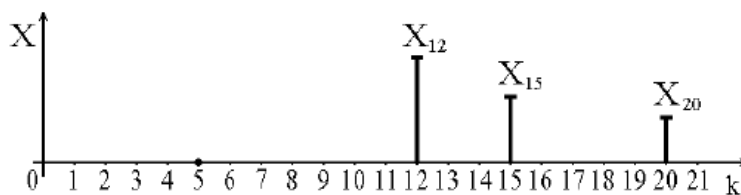


Рис. 2.8. Спектр полігармонічного процесу без основної частоти

Як було розглянуто вище, процес, утворений підсумовуванням двох або більше синусоїд з кратними частотами, є періодичним. Однак якщо процес утворюється підсумовуванням двох і більше синусоїд з довільними частотами, то він буде періодичним тільки в тому випадку, якщо відношення всіх можливих пар частот являють собою раціональні числа. В іншому випадку основний період процесу дорівнює нескінченності, такі процеси називають **майже періодичними** і описують функцією часу

$$x(t) = X_0 + \sum_{k=1}^{\infty} X_k \cos(\omega_k t - \varphi_k), \quad (2.9)$$

де не всі відношення f_k/f_j є раціональними числами.

Фізичні явища, яким відповідають майже періодичні процеси, утворюються здебільшого при підсумовуванні двох або більше незалежних гармонічних процесів. Прикладом майже періодичних процесів є вібрації літака з кількома моторами, що працюють на своїх частотах.

Для опису майже періодичних процесів використовуються ті ж математичні моделі, що й для опису полігармонічних, однак відсутність періодичності створює деякі проблеми при цифровому обробленні такого типу процесів, оскільки більшість методів оброблення розроблялася для періодичних сигналів.

До **перехідних** належать усі неперіодичні процеси, які не є майже періодичними. Важлива відмінність перехідних процесів від періодичних і

майже періодичних полягає в тому, що їх неможливо подати з допомогою спектра, який складається з дискретних компонент.

Будь-який безперервний сигнал $x_1(t)$, заданий на інтервалі $t \in [-\infty, \infty]$, можна подати у вигляді (підсумовування у виразі (2.7) замінено інтегруванням):

$$x_1(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X_1(\omega) e^{j\omega t} d\omega, \quad (2.10)$$

де

$$X_1(\omega) = \int_{-\infty}^{\infty} x_1(t) e^{-j\omega t} dt.$$

Функція $X(\omega)$ характеризує спектральний склад сигналу $x(t)$ і має назву спектральної густини сигналу $x(t)$. Така назва зумовлена тим, що для неперіодичного сигналу $x(t)$ (при $T \rightarrow \infty$) частотний інтервал між сусідніми гармонічними складовими прямує до нуля, тобто на відміну від періодичних сигналів неперіодичні мають суцільний спектр.

Приклади типових перехідних процесів і їх спектральних густин показано на рис. 2.9–2.11.

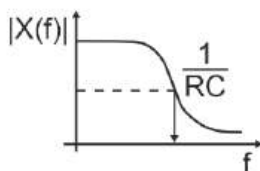
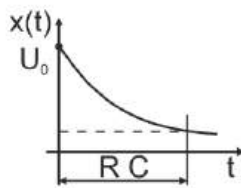
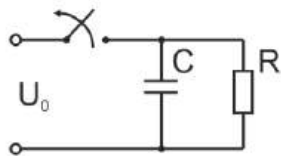


Рис. 2.9. Перехідний процес розряду конденсатора при розмиканні ключа

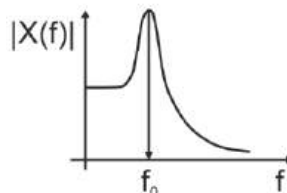
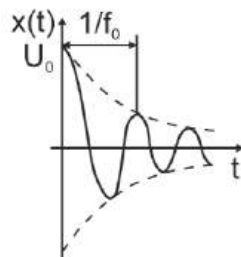
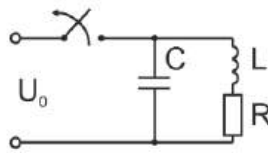


Рис. 2.10. Перехідний процес в коливальному контурі при розмиканні ключа

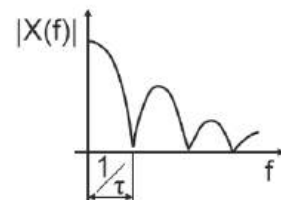
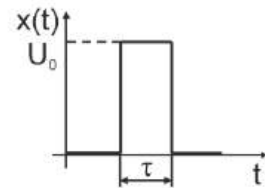
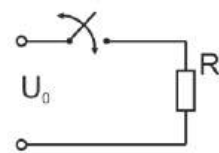


Рис. 2.11. Перехідний процес на активному навантаженні при короткочасному замиканні ключа

2.2. Дискретизація детермінованих процесів

Особливістю комп'ютерного моделювання є те, що всі безперервні функції (часу, частоти, температури тощо) $x(t)$ подають послідовністю їх відліків x_i (рис. 2.12, а), узятих через однакові проміжки часу Δt (або іншого параметра) і визначають згідно з теоремою Котельникова граничну частоту спектра сигналу, щоб однозначно відновити функцію $x(t)$ за відліками x_i (рис. 2.12, б):

$$F_{\max} = \frac{1}{2\Delta t}. \quad (2.11)$$

Частота дискретизації при цьому

$$f_D = 2F_{\max}. \quad (2.12)$$

Кількість відліків N (обсяг вибірки) пов'язана з необхідним часом спостереження T за процесом

$$N = \frac{T}{\Delta t}. \quad (2.13)$$

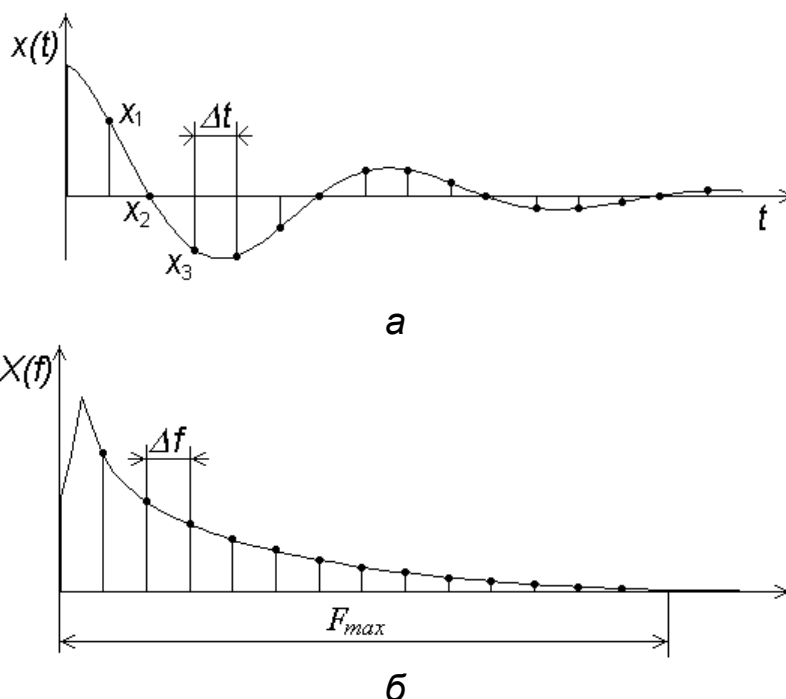


Рис. 2.12. Дискретне подання безперервного процесу (а) і його спектра (б)

Для аналізу спектрального складу процесів використовують зазвичай процедуру швидкого перетворення Фур'є (ШПФ) для дискретних відліків процесу x_i :

$$X(f) = \int_0^T x(t) \exp(-j2\pi ft) dt.$$

Після застосування процедури ШПФ отримують послідовність відліків X_k , що є дискретним поданням спектра сигналу з рівномірним дискретним кроком за частотою Δf (див. рис. 2.12, б), який однозначно пов'язаний з часом спостереження за процесом T як

$$\Delta f = \frac{1}{T}. \quad (2.14)$$

Особливістю процедури ШПФ є вимога до обсягу вибірки (2.13) $N = 2^M$, де M – ціле число.

2.3. Комп'ютерне моделювання детермінованих процесів

При моделюванні сигналу $x(t)$ принципово важливо визначити час моделювання T , оскільки роздільна здатність за частотою процедури ШПФ (рис. 2.13) залежить від часу спостереження сигналу:

$$\Delta f = 1/T.$$

При цьому необхідно мати на увазі, що процедура ШПФ є орієнтованою на оброблення періодичних сигналів, тобто обчислений таким способом спектр сигналу може відрізнятися від справжнього внаслідок періодичного продовження сигналу при обчисленні спектра. Цей ефект показано на рис. 2.13.

Максимально можлива частота в спектрі сигналу F визначає мінімальну частоту дискретизації $f'_D = 2F$.

Однак на практиці слід брати частоту дискретизації в 5–10 разів більшою, щоб отримати 10–20 відліків сигналу на період. Це не тільки полегшує візуалізацію, але й зменшує похибку моделювання.

Інтервал дискретизації (інтервал часу між відліками (див. рис. 2.12, а)) однозначно пов'язаний з частотою дискретизації:

$$\Delta t' = 1/f'_D.$$

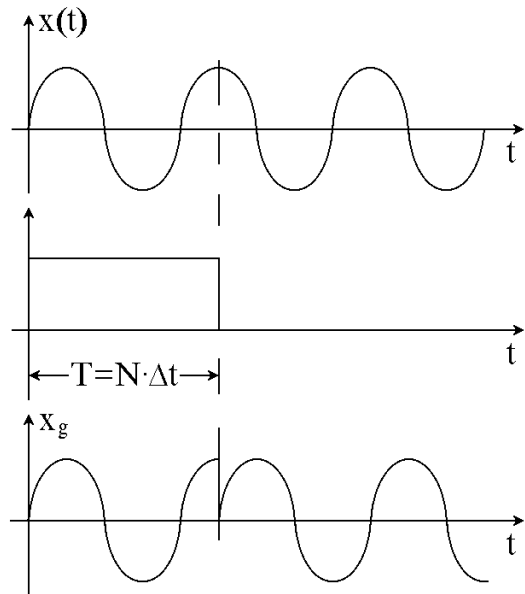


Рис. 2.13. Періодичне продовження $x_g(t)$ сигналу $x(t)$

Тоді необхідна кількість відліків (обсяг вибірки)

$$N' = T\Delta t'.$$

Для подальшого використання ШПФ отримане значення N' слід «округлити вверх» до найближчого значення, кратного 2^m , наприклад:

$$N' = 900 \Rightarrow N = 1024.$$

Далі уточнюємо інтервал дискретизації

$$\Delta t = T/N,$$

частоту дискретизації

$$f_D = 1/\Delta t,$$

і формуємо послідовність відліків дискретного часу

$$t_i = i\Delta t,$$

де $i = 0 \dots N-1$ або $i = 1 \dots N$ залежно від використовуваних програмних засобів.

Тепер можна виконувати власне моделювання сигналу:

$$x_i = \phi(t_i),$$

де $\phi(\bullet)$ – математична модель сигналу.

При спектральному методі формування сигналу після визначення часу спостереження сигналу T й обсягу вибірки N необхідно визначити кількість відліків у дискретному спектрі сигналу за формулою

$$M = N/2,$$

оскільки при використанні ШПФ тільки половина відліків спектра містить корисну інформацію. Другу половину масиву з N відліків деякі процедури ШПФ не повертають зовсім, а деякі повертають комплексно-зв'язані та дзеркально розташовані значення спектра сигналу (рис. 2.14).

Якщо як вихідні дані задано роздільну здатність за частотою Δf (наприклад, при моделюванні сигналів зі складними видами модуляції), то необхідний час моделювання можна визначити як

$$T = 1/\Delta f,$$

а обсяг вибірки буде залежати від ширини спектра сигналу $F_{\max}$:

$$M = F_{\max}/\Delta f.$$

Сітку частот визначають за часом спостереження за процесом, тобто

$$f_j = j\Delta f = j/T,$$

де $j=0\dots M-1$ або $j=1\dots M$ залежно від використовуваного програмного забезпечення. Наприклад, при використанні процедури $fft(\cdot)$ мови програмування *Python* змінна циклу за частотою $j=0\dots M$.

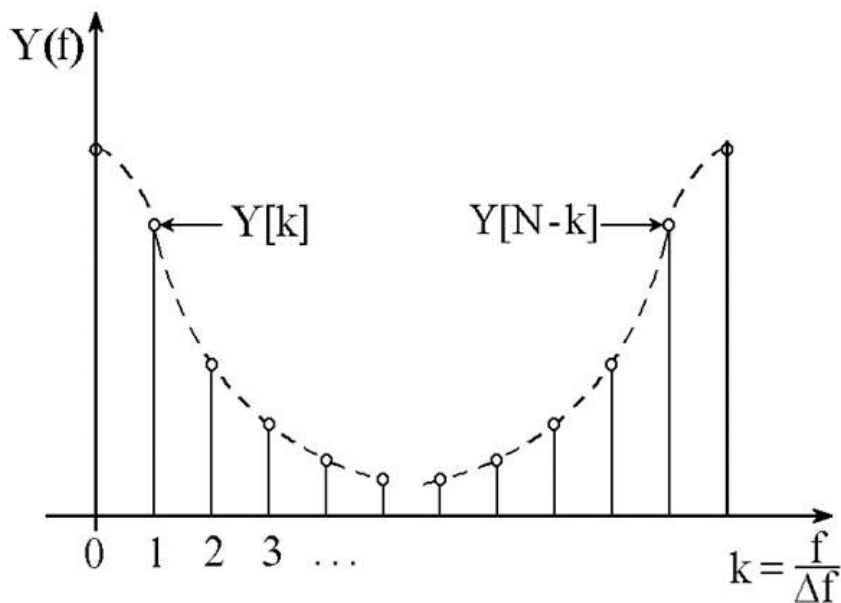


Рис. 2.14. Дзеркальне розташування копії спектра

Тепер можна виконати формування спектра сигналу

$$X_j = \phi(f_j),$$

де $\phi(\cdot)$ – математична модель спектра сигналу.

Після оберненого перетворення Фур'є

$$x(t) = F^{-1}\{X(f)\},$$

буде одержано необхідний сигнал $x(t)$.

Моменти часу t_i , для яких отримано відліки сигналу $x_i = x(t_i)$, визначаються обсягом вибірки й часом спостереження:

$$N = 2M, \quad \Delta t = T/N, \quad t_i = i\Delta t,$$

де $i=0\dots N-1$ або $i=1\dots N$ залежно від використовуваних програмних засобів.

2.4. Приклад моделювання гармонічного процесу

Розглянемо приклад комп'ютерного моделювання гармонічного сигналу $s(t) = A \cos(2\pi ft + \varphi)$ (рис. 2.15) з такими параметрами:

- амплітуда $A = 3$ В;
- частота $f = 2$ Гц;
- початкова фаза $\varphi = -30^\circ$.

```
from math import cos, pi
import matplotlib.pyplot as plt
%matplotlib inline
A=3.0; f=2.0; fi=-30.0*pi/180.0;
F=2.0; k=10; fd=2*F*k; dt=1.0/fd
T=2/f; n=int(fd*T)+1
t=[]; s=[]
for i in range(n):
    t.append(i*dt)
    s.append(A*cos(2*pi*f*t[i]+fi))
plt.plot(t, s, color='green', label='s(t)')
plt.grid(); plt.legend()
```

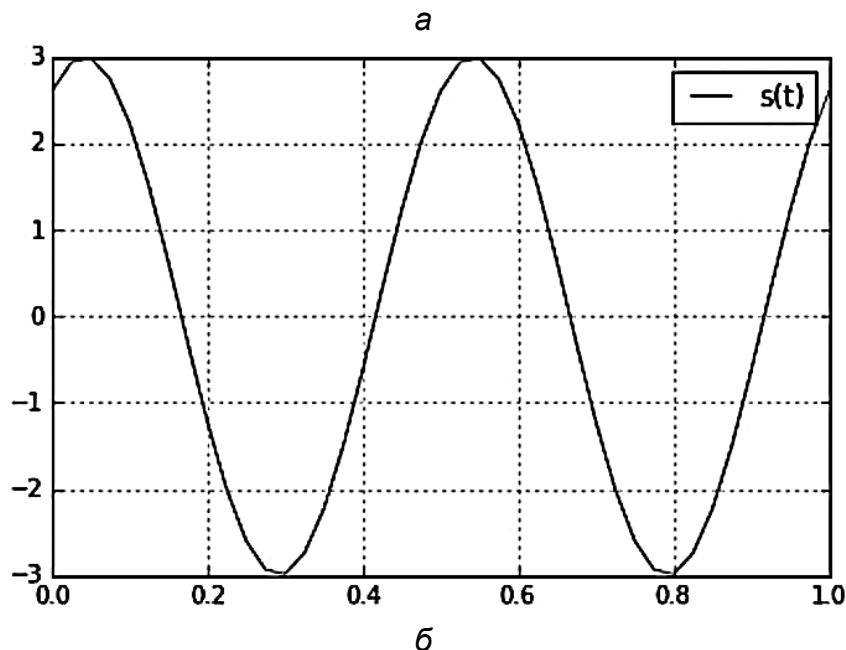


Рис. 2.15. Приклад програми моделювання гармонічного процесу (а)
і результати моделювання (б)

Визначимо параметри моделювання.

1. Максимальна (верхня) частота сигналу $F = 2$ Гц.
2. Згідно з теоремою Котельникова частота дискретизації

$$f_d = 2Fk,$$

коефіцієнт «запасу», що визначає точність моделювання, $k = 10$.
Тоді $f_d = 2 \cdot 2 \text{ Гц} \cdot 10 = 40$.

3. Період повторення сигналу $T_S = 1 / f = 0,5 \text{ с}$. Візьмемо час моделювання $T = 2T_S$ (два періоди синусоїди): $T = 2 \cdot 0,5 = 1 \text{ с}$.

4. Кількість відліків сигналу для моделювання

$$N = f_d T = 40 \cdot 1 = 40.$$

5. Інтервал дискретизації (крок моделювання за часом)

$$\Delta t = 1 / f_d.$$

6. Послідовність моментів часу моделювання

$$t_i = i\Delta t, \quad i = 0 \dots N.$$

Власне моделювання сигналу полягає в обчисленні заданої функції $s(t) = A \cos(2\pi f t + \varphi)$ у дискретні моменти часу t_i , $i = 0 \dots N$:

$$s_i = A \cos(2\pi f t_i + \varphi),$$

а оброблення даних моделювання – у будуванні графіка отриманих даних.

2.5. Приклад моделювання й спектрального аналізу полігармонічного процесу

Виконаємо комп'ютерне моделювання й аналіз спектра полігармонічного сигналу

$$s(t) = A_1 \cos(2\pi f_1 t + \varphi_1) + A_2 \cos(2\pi f_2 t + \varphi_2),$$

з такими параметрами:

- амплітуда $A_1 = 2$, $A_2 = 0,5$;
- частота $f_1 = 2 \text{ Гц}$, $f_2 = 6 \text{ Гц}$;
- початкові фази $\varphi_1 = 60^\circ$, $\varphi_2 = 30^\circ$.

Визначимо параметри моделювання.

1. Максимальна (верхня) частота сигналу $F_B = 6 \text{ Гц}$.

2. Згідно з теоремою Котельникова частота дискретизації

$$f_d = 2Fk$$

з коефіцієнтом «запасу» $k = 5$.

3. Оскільки мінімальна частота $f_1 = 2$ Гц, крок за частотою Δf візьмемо 1 Гц. Тоді час моделювання $T = 1 / \Delta f = 1$ с.
4. Кількість відліків сигналу для моделювання $N = f_d T$.
5. Інтервал дискретизації (крок моделювання за часом)

$$\Delta t = \frac{1}{f_d}.$$

6. Послідовність моментів часу моделювання

$$t_i = i \Delta t, \quad i = 0 \dots N.$$

Власне моделювання сигналу полягає в обчисленні заданої функції $s(t)$ у дискретні моменти часу t_i , $i = 0 \dots N$.

Оброблення даних моделювання полягає в обчисленні комплексного спектра сигналу $S(j\omega)$, будуванні графіків $s(t)$, реальної та уявної частин комплексного спектра, його модуля (амплітудного спектра) і фази (фазового спектра).

Розроблення алгоритму та програми моделювання. Для розв'язання задачі спектрального аналізу даних моделювання необхідно обчислити перетворення Фур'є, для чого скористаємося модулем *numpy.fft* з бібліотеки *numpy*.

NumPy (Numeric Python) – це бібліотека мови *Python*, що містить безліч математичних функцій і має підтримку великих багатовимірних масивів і матриць. Бібліотека *NumPy* «працює» з однорідними структурами даних – масивами, які є окремим випадком списків *Python*. Масиви на відміну від списків мають містити дані одного типу.

Список *Python* можна перетворити на масив функцією *array*. Порожній масив створюється функцією *empty*, заповнений нулями – функцією *zeros*. При цьому може вказуватися потрібний тип даних. Наприклад, **`x = numpy.zeros(m, dtype = float)`** – масив з m дійсних чисел, що дорівнюють нулю (0,0), а інструкція **`y = np.empty(10, dtype = complex)`** створить порожній масив для зберігання десяти комплексних чисел.

Другою особливістю бібліотеки *NumPy* є можливість роботи з масивами не тільки як зі списками (з поелементним доступом, наприклад **`A = X[10]`** – змінній A присвоїти значення 10-го елемента масиву X), а й виконувати операції з масивами цілком без використання циклів. Наприклад, якщо X – масив, що містить 1000 елементів, то інструкція **`Y = sin(X)`** заповнить масив Y тисячею значень функції *sin* від відповідних елементів масиву X .

Для обчислення комплексного спектра сигналу скористаємося функцією *rfft* з модуля *fft* бібліотеки *NumPy*. Ця функція виконує пряме швидке перетворення Фур'є від масиву розміром N реальних даних, повертаючи $N / 2 + 1$ значення комплексного спектра.

Приклад програми моделювання мовою *Python* і результати моделювання показано на рис. 2.16–2.18.

```
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
from numpy import cos, pi, arctan2, degrees
from numpy.fft import fft
Fb=12.0; k1=5; fd=2*Fb*k1; dt=1.0/fd
df=1.0; k2=1; T=k2/df; n=int(fd*T)
t=np.empty(n,dtype=float); s1=np.empty(n,dtype=float)
s2=np.empty(n,dtype=float); ss=np.empty(n,dtype=float)
for i in range(n):
    t[i]=i*dt
s1=2.0*cos(2*pi*2.0*t-60.0*pi/180.0)
s2=0.5*cos(2*pi*6.0*t+30.0*pi/180.0)
ss=s1+s2
fig1=plt.figure()
plt.plot(t,s1,color='red', label='s1')
plt.plot(t,s2,color='blue', label='s2')
plt.plot(t,ss,color='green', label='ss')
plt.grid(); plt.legend()
```

Рис. 2.16. Програма моделювання та спектрального аналізу полігармонічного сигналу

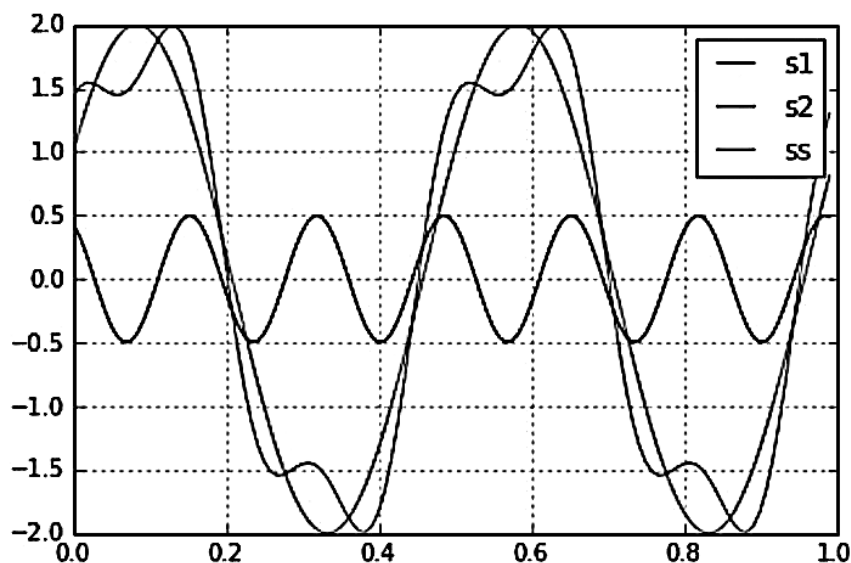


Рис. 2.17. Результат моделювання полігармонічного сигналу

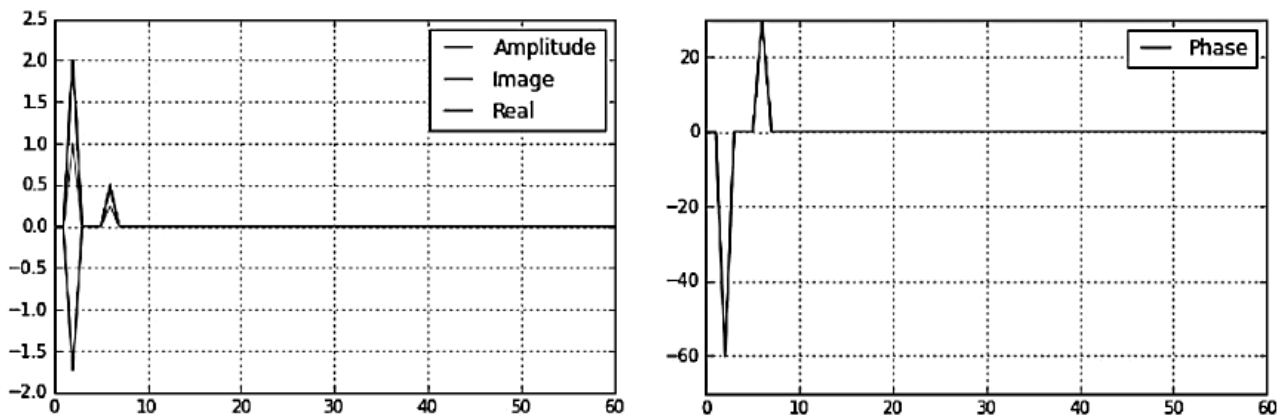


Рис. 2.18. Результати оброблення даних моделювання полігармонічного сигналу

2.6. Приклад моделювання періодичної послідовності імпульсів

Періодичну послідовність прямокутних імпульсів p із заданим періодом T_1 і шпаруватістю q (відношенням тривалості імпульсу до його періоду) можна отримати шляхом перетворення гармонійного процесу з такою ж частотою $f_1 = 1/T_1$ за таким алгоритмом:

1. Формуємо початковий процес $c(t) = 1\cos(2\pi f_1 t)$.

2. Оскільки функція $\cos(\cdot)$ є симетричною відносно нуля, а її період становить 360° (рис. 2.19, а), за заданою шпаруватістю q визначаємо поріг перетворення синусоїди на імпульси $p = \cos\left(q \frac{360}{2}\right)$.

3. Якщо значення $c(t) > p$, то значенню імпульсного процесу $s(t)$ присвоюємо «1», інакше – «0» (рис. 2.19, б).

4. Для зручності аналізу отриманий процес можна зсунути в часі, як показано на рис. 2.19, в. Для цього вихідний процес $c(t)$ необхідно (до початку моделювання) зсунути за фазою, наприклад, на 180° :

$$c(t) = \cos(2\pi f_1 t - 180^\circ).$$

Для розрахунку параметрів моделювання необхідно визначити максимальну (верхню) частоту сигналу F_B . Теоретично максимальна частота спектра прямокутного імпульсу є нескінченною, тому F_B беруть, ураховуючи структуру спектра (рис. 2.20) і необхідну точність моделювання. Для послідовності імпульсів з періодом T і їх тривалістю τ на практиці беруть $F_B = (3 \dots 10) / \tau$. Тоді частоту дискретизації візьмемо $f_d = 2F_B k$, де $k = 5 \dots 10$ – коефіцієнт «запасу», що забезпечує точність моделювання.

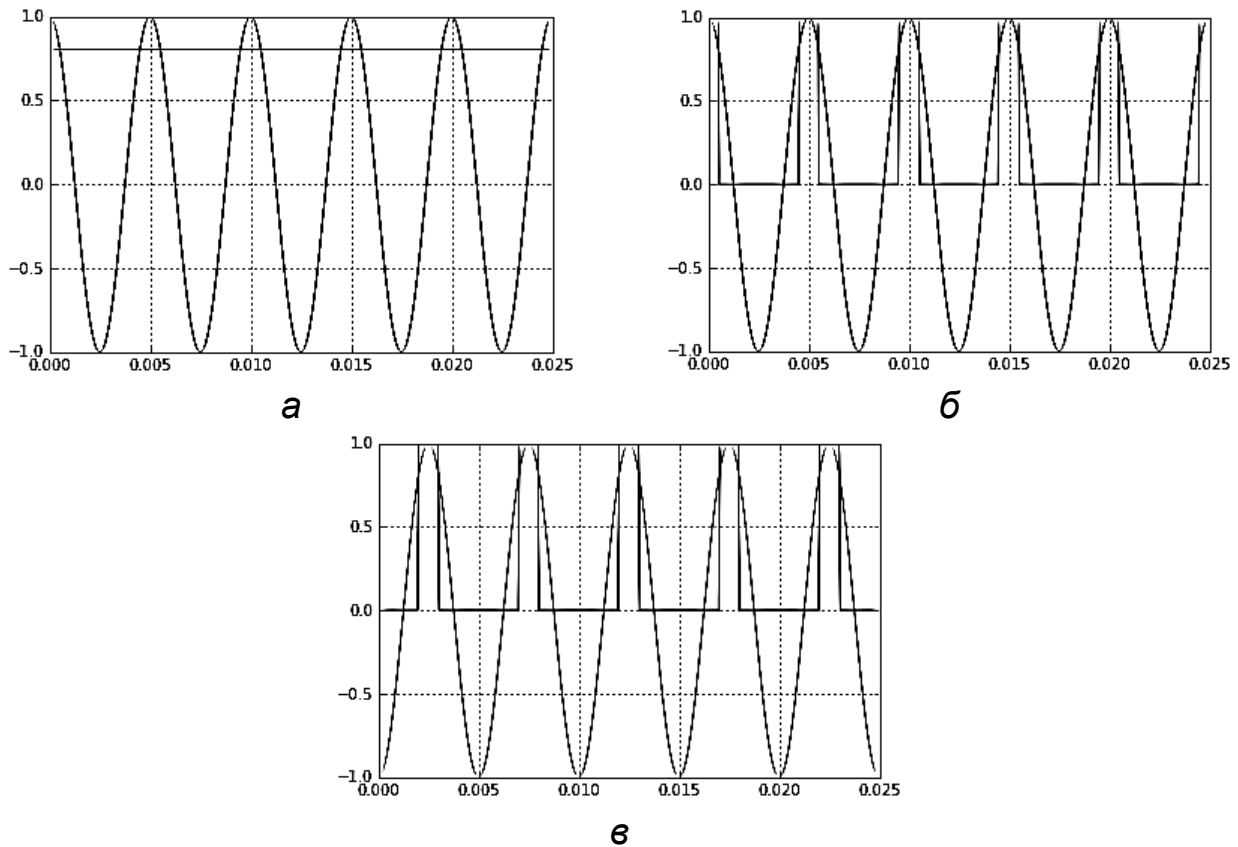


Рис. 2.19. Моделювання періодичної послідовності імпульсів

Мінімальна частота в спектрі $f_1 = 1/T$, крок за частотою Δf беруть не більше $\Delta f = f_1/2$. Тоді час моделювання $T_M = 1/\Delta f$, кількість відліків сигналу для моделювання $N = f_d T_M$, інтервал дискретизації (крок моделювання за часом) $\Delta t = 1/f_d$. Послідовність моментів часу моделювання $t_i = i\Delta t$, $i = 0 \dots N$.

Власне, моделювання сигналу полягає в обчисленні функції $c(t)$ у дискретні моменти часу t_i , $i = 0 \dots N$, і перетворенні її на послідовність імпульсів $s(t)$ за алгоритмом, розглянутим вище. Оброблення даних моделювання полягає в будіванні графіків $s(t)$, реальної та уявної частини комплексного спектра, його модуля (амплітудного спектра) і фази (фазового спектра), обчисленні комплексного спектра сигналу $S(j\omega)$.

Приклад програми моделювання й спектрального аналізу послідовності імпульсів показано на рис. 2.21, 2.22.

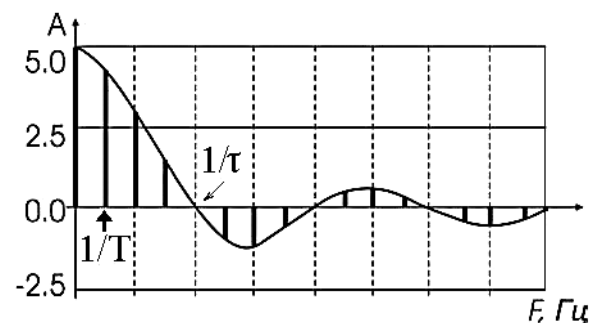


Рис. 2.20. Структура спектра періодичної послідовності прямокутних імпульсів

```

from numpy.fft import rfft
import numpy as np
from numpy import pi, cos, sin, radians
from matplotlib.pyplot import plot, grid, figure
%matplotlib inline
T=5.0e-3; q=0.1 #період T, шпаруватість q
ti=T*q; #довжина імпульсу ti
Fb=5.0/ti; fd=4*Fb #частота дискретизації
f1=1/T; df=f1/4.0; Tm=1/df #час моделювання
n=int(fd*Tm) #кількість відліків
dt=Tm/n; t=[i*dt for i in range(n)]; t=np.array(t)
c=np.empty(n, dtype=float)
c=1.0*cos(2*pi*f1*t+radians(-90.0))
s=np.empty(n, dtype=float)
p=cos(q*180/pi)
for i in range(n):
    if c[i]>p:
        s[i]=1.0
    else:
        s[i]=0.0
fig_1 = figure()
plot(t,s,linewidth=1, color="b"); grid(True)
S = rfft(s)/n;
f=[i*df for i in range(n//2+1)]
A=abs(S)
fig_2 = figure()
plot(f,A,linewidth=1, color="b"); grid(True)

```

Рис. 2.21. Програма моделювання й спектрального аналізу
послідовності імпульсів

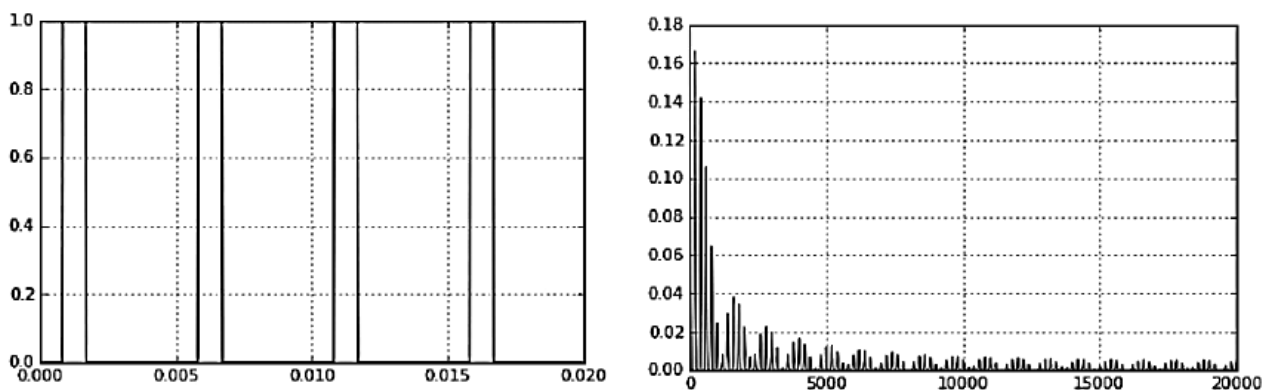


Рис. 2.22. Результати моделювання й спектрального аналізу послідовності
імпульсів

Контрольні запитання

1. Наведіть класифікацію процесів з огляду на комп'ютерне моделювання.
2. Який процес називають періодичним? Якою математичною моделлю його описують?
3. Що таке комплексний спектр періодичного процесу?
4. Що таке амплітудний і фазовий спектри періодичного процесу?
5. Який процес називають полігармонійним? Якою математичною моделлю його описують?
6. Який процес називають перехідним? Якою математичною моделлю його описують?
7. Що таке дискретизація сигналу?
8. Що таке інтервал дискретизації сигналу? Як цей інтервал пов'язаний з частотою дискретизації?
9. Як визначити частоту дискретизації для гармонічного процесу?
10. Як визначити частоту дискретизації для полігармонічного процесу?
11. Як визначити частоту дискретизації для періодичної послідовності імпульсів?
12. Що таке швидке перетворення Фур'є (ШПФ)? Для чого його застосовують при комп'ютерному моделюванні?
13. Що таке роздільна здатність ШПФ за частотою («сітка частот»)?
14. Що таке час моделювання процесу? Як час моделювання процесу пов'язаний з роздільною здатністю ШПФ за частотою?
15. Як визначити мінімально достатній час моделювання для гармонічного процесу?
16. Як визначити мінімально достатній час моделювання для полігармонічного процесу?
17. Як визначити мінімально достатній час моделювання періодичної послідовності імпульсів?
18. Як визначити обсяг вибірки (кількість відліків, розмір масиву даних) для моделювання періодичних процесів?

3. КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ЛІНІЙНИХ СИСТЕМ

3.1. Математичні моделі лінійних систем

Лінійними називають системи, що підпорядковуються принципу суперпозиції: якщо на систему з оператором A впливає сукупність зовнішніх впливів $x = \sum_{k=1}^n a_k x_k$, то реакцію системи можна подати як суму реакцій на окремі дії:

$$A\left\{\sum_{k=1}^n a_k x_k\right\} = y = \sum_{k=1}^n a_k A(x_k). \quad (3.1)$$

У принципі суперпозиції використовуються властивості адитивності $A(x_1 + x_2) = A(x_1) + A(x_2)$ і однорідності $cA(cx_1) = A(cx_1)$.

Лінійну систему називають статичною, якщо вихідна реакція системи залежить тільки від значень вхідних параметрів у певний момент часу і не залежить від значень вхідних впливів і станів системи в попередні моменти часу. Наприклад, $y = kx$ – модель одновимірної лінійної статичної системи.

Систему називають динамічною (інерційною), якщо значення її вихідного сигналу залежать від значень вхідного сигналу в один той самий момент часу і від попереднього стану системи.

З математичного погляду лінійна динамічна система (ЛДС) описується диференціальним рівнянням

$$\sum_{i=0}^n a_i \frac{\partial^i y(t)}{\partial t^i} = \sum_{j=0}^m b_j \frac{\partial^j x(t)}{\partial t^j}, \quad (3.2)$$

наприклад:

$$a_2 \frac{\partial^2 y(t)}{\partial t^2} + a_1 \frac{\partial y(t)}{\partial t} + a_0 y(t) = b_1 \frac{\partial x(t)}{\partial t} + b_0 x(t)$$

або

$$a_2 y''(t) + a_1 y'(t) + a_0 y(t) = b_1 x'(t) + b_0 x(t).$$

Якщо ЛДС має постійні параметри, фізично реалізується і є стійкою, то її динамічні властивості можна описати з допомогою комплексної передатної функції системи (або комплексної частотної характеристики) $\dot{K}(j\omega)$.

Методи знаходження комплексної передатної функції розглядалися в курсі «Теорія електричних кіл».

Частотна характеристика в загальному випадку є комплексною величиною, яку зручно подавати у вигляді амплітудно-частотної $A(f)$ (АЧХ) і фазо-частотної $\phi(f)$ (ФЧХ) характеристик:

$$\dot{K}(j\omega)\big|_{\omega=2\pi f} = |\dot{K}(f)|e^{-j\phi(f)} = A(f)e^{-j\phi(f)}. \quad (3.3)$$

Використовуючи поняття АЧХ і ФЧХ, частотній характеристиці можна надати простої фізичної інтерпретації. Нехай на вхід системи з частотною характеристикою $\dot{K}(f)$ надходить гармонійний детермінований сигнал $x(t) = X \cos(\omega t + \phi_x)$, який існує на нескінченному інтервалі часу, тобто всі перехідні процеси в ЛДС є завершеними і система перебуває в сталому режимі. Тоді процес на виході з системи також буде гармонійним коливанням $y(t) = Y \cos(\omega t + \phi_y)$.

Відношення амплітуд вихідного і вхідного процесів буде визначатися значенням АЧХ на певній частоті $f = \omega/2\pi$, тобто $A(f) = Y/X$, а різниця фаз цих процесів – ФЧХ системи:

$$\phi_y - \phi_x = \phi(f).$$

Таким чином, гармонічні функції типу $e^{j\omega t}$ під час проходження через ЛДС не змінюють свого вигляду (в усталеному режимі), а змінюють лише амплітуду й фазу, що фіксується співвідношенням між входом і виходом ЛДС:

$$Y(f) = \dot{K}(f)X(f). \quad (3.4)$$

На основі (3.4) алгоритм моделювання сигналу на виході ЛДС можна подати в такому вигляді.

1. Початковий сигнал $x(t)$ дискретизують з інтервалом часу Δt (див. розд. 2 і рис. 3.1, а).

2. Для дискретної послідовності вхідного сигналу $x[i]$ завдяки застосуванню ШПФ визначають її дискретний спектр $X[k] = F\{x[i]\}$, що являє собою дискретизацію безперервного спектра $X(f) = F\{x(t)\}$ з кроком за частотою Δf (рис. 3.1, б).

3. Якщо відомо передатну функцію системи $\dot{K}(j\omega)$ у вигляді аналітичного виразу або алгоритм її обчислення, то її можна подати в дискретному вигляді як масив комплексних коефіцієнтів передання для частот f з номером k (рис. 3.1, в):

$$W[k] = K(k\Delta f).$$

4. Спектр вихідного сигналу визначають як добуток елементів масивів: $Y[k] = X[k]W[k]$ (рис. 3.1, в).

5. Дискретну послідовність $y[i]$ (сигнал на виході системи) обчислюють як обернене ШПФ спектра $Y[k]$ (рис. 3.1, д):

$$y[i] = F^{-1}[Y[k]].$$

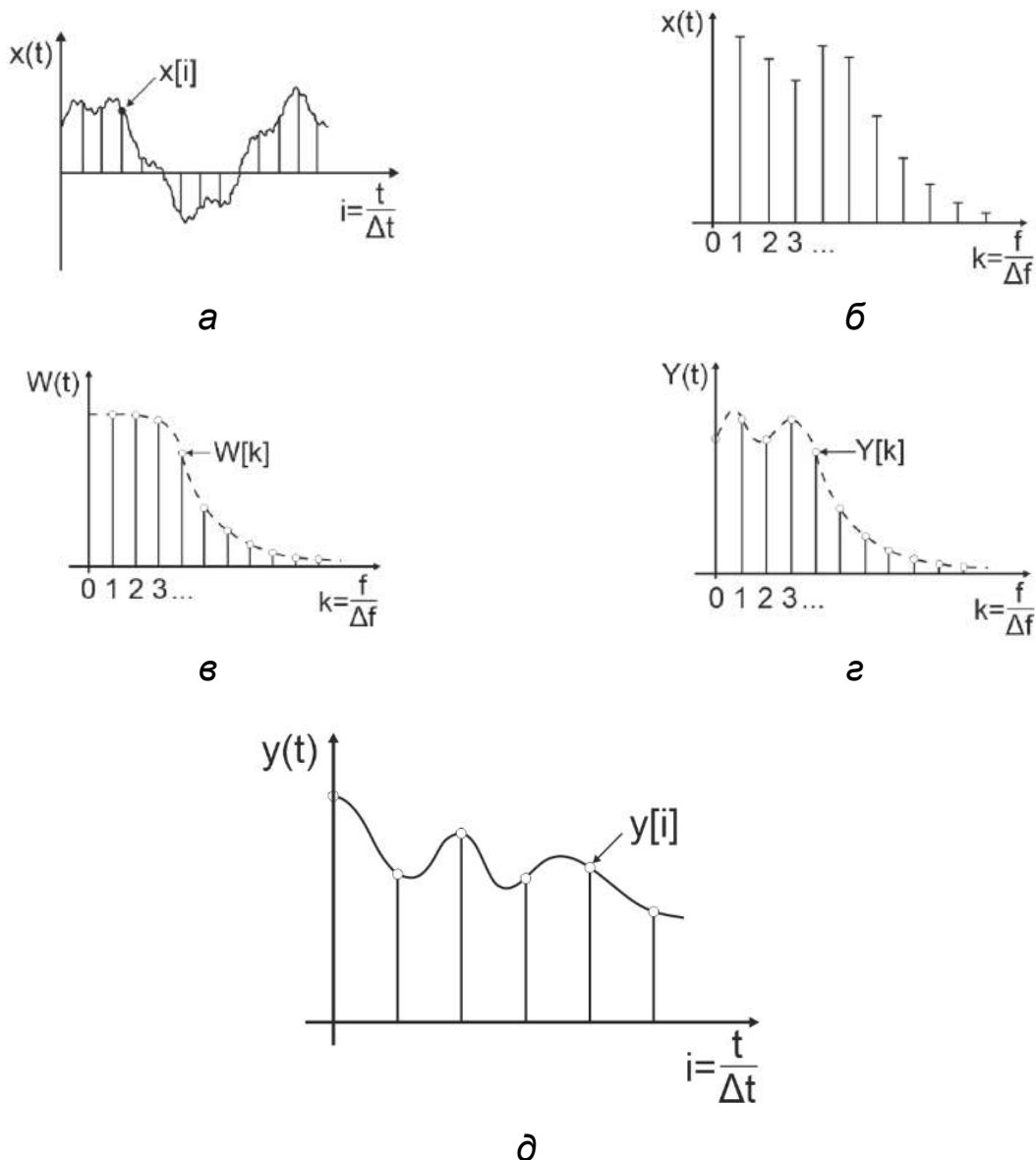


Рис. 3.1. Моделювання проходження сигналу через систему із застосуванням спектрального методу

Через обмеженість довжини послідовності $y[i]$ відновлення відбувається з похибкою, величина якої є обернено пропорційною обсягу вибірки як $\sqrt{N}$.

3.2. Приклад моделювання проходження періодичної послідовності імпульсів крізь коливальний контур

Припустимо, на вхід пристрою, схему електричну принципову якого показано на рис. 3.2, надходить періодична послідовність прямокутних імпульсів тривалістю $\tau = 0,004$ с з амплітудою $A = 2$ В і періодом повторення $T = 0,02$ с. Під час моделювання роботи пристрою необхідно визначити форму вихідного сигналу.

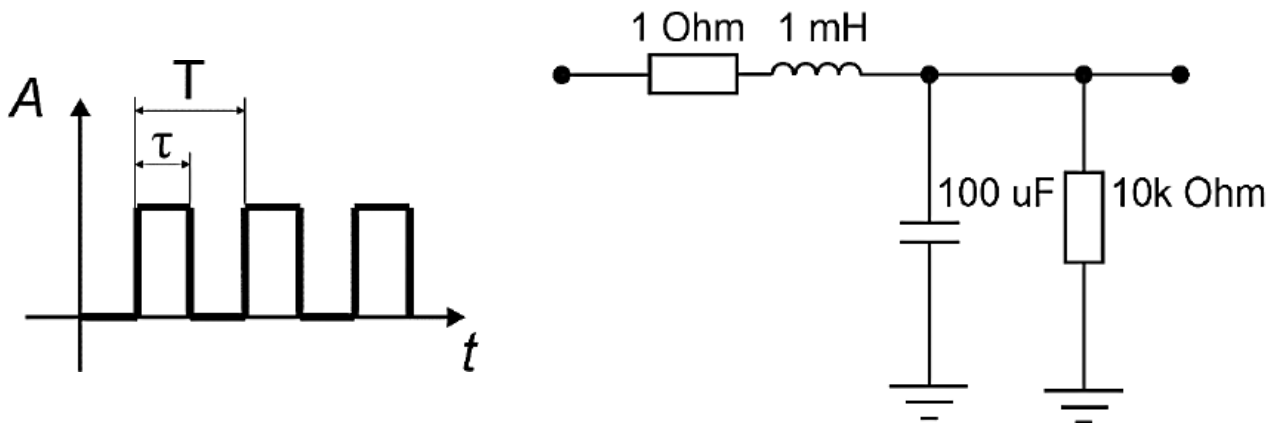


Рис. 3.2. Вхідний сигнал і схема електричного кола

Для вирішення цього завдання необхідним є виконання наступних дій:

1. Параметри моделювання – частоту дискретизації, час моделювання й обсяг вибірки – визначаємо за параметрами спектра вхідного сигналу за методикою підрозд. 2.5. Ширину спектра сигналу визначимо як $F_{\max} = k/\tau = 1250$ Гц (k візьмемо таким, що дорівнює 5). Тоді частоту дискретизації можна визначити як $f_d = 2kF_{\max} = 2 \cdot 2 \cdot 1250 = 5000$ Гц (k візьмемо таким, що дорівнює 2). Інтервал дискретизації $\Delta t = 1/f_d = 0,2$ мс. Частота повторення імпульсів $f_1 = 1/T = 50$ Гц. Відповідно до рис. 2.20, крок дискретизації за частотою (роздільну здатність за частотою алгоритму ШПФ) задамо як $\Delta f = f_1/k$, візьмемо $k = 5$, $\Delta f = 10$ Гц. Тоді час моделювання $T_m = 1/\Delta f = 0,1$ с, а кількість відліків, необхідних для моделювання, $N = f_d T_m = 500$.

2. Формуємо масив відліків часу: $t_i = i \Delta t$,

3. Моделюємо вхідний сигнал s_i за алгоритмом розглянутим у підрозд. 2.6:

$$\text{якщо } \cos(2\pi f_1 t_i) > \pi\tau/T, \text{ то } s_i = A, \text{ інакше } - s_i = 0.$$

4. Комплексний спектр вхідного сигналу S визначаємо з допомогою функції `rfft`: $S = \text{rfft}(s)$. Для будування графіка виокремлюємо амплітудний спектр сигналу: $As = \text{abs}(S)$.

5. Формуємо масив відліків частоти: $f_i = i\Delta f$, $i = 0 \dots N/2 + 1$.

6. Для моделювання роботи пристрою обчислюємо значення комплексної передатної функції пристрою одним із методів теорії кіл. У цьому прикладі скористаємося методом еквівалентних перетворень. Для схеми пристрою, зображеної на рис. 3.2 запишемо:

– комплексні опори реактивних елементів схеми:

$$Z_L = j\omega L, \quad Z_C = \frac{1}{j\omega C};$$

– опір послідовно з'єднаних елементів R_1 і L :

$$Z_1 = R_1 + Z_L;$$

– опір паралельно з'єднаних елементів R_2 і C :

$$Z_2 = \frac{R_2 Z_C}{R_2 + Z_C};$$

– комплексний коефіцієнт передавання:

$$H = \frac{Z_2}{Z_1 + Z_2}.$$

7. Значення комплексної передатної функції пристрою обчислюємо для всіх частот «сітки частот» f_i , $i = 0 \dots N/2 + 1$ і зберігаємо в масиві $H[]$. АЧХ пристрою визначаємо як модуль $|H|$.

8. Масив комплексного спектра вихідного сигналу визначаємо як добуток масиву спектра вхідного сигналу і масиву комплексної передатної функції $U = H * S$.

9. Для відновлення форми вихідного сигналу обчислюємо обернене перетворення Фур'є від комплексного спектра вихідного сигналу, для чого використаємо функцію `irfft`: $u = \text{irfft}(U)$.

Повний текст програми моделювання й результати обчислень показано на рис. 3.3, 3.4.

```

import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
from numpy import pi, cos
from numpy.fft import rfft, irfft
T1=0.02; q=0.2; tau=q*T1; A=2.0; f1=1/T1
Fm=5.0/tau; fd=2*2*Fm; dt=1.0/fd
df=f1/5.0; T=1/df; n=int(fd*T)
t=np.empty(n, dtype=float)
s=np.zeros(n, dtype=float)
p=cos(q*pi)
for i in range(n):
    t[i]=dt*i
    if(cos(2*pi*f1*t[i])>p): s[i]=A
fig1=plt.figure(); plt.plot(t,s)
S=rfft(s); As=abs(S)
m=int(n/2+1)
f=np.empty(m, dtype=float)
for i in range(m):
    f[i]=df*i
fig2=plt.figure(); plt.plot(f,As)
H=np.empty(m, dtype=complex)
R1=1.0; L=1.0e-3; C=100.E-6; R2=10.0e3
for i in range(m):
    f[i]=df*i; w=2*pi*f[i]
    if (w==0): w=df/1000.0
    ZL=1j*w*L; ZC=1.0/(1j*w*C)
    Z1=R1+ZL; Z2=ZC*R2/(ZC+R2)
    H[i]=Z2/(Z1+Z2)
A4X=abs(H)
fig3=plt.figure(); plt.plot(f,A4X)
U=H*S #спектр вихідного сигналу
Au=abs(U)
fig4=plt.figure(); plt.plot(f,Au)
u=irfft(U);
# Графіки
fig1=plt.figure()
plt.plot(t,s)
plt.plot(t,u)

```

Рис. 3.3. Програма моделювання електронного пристрою

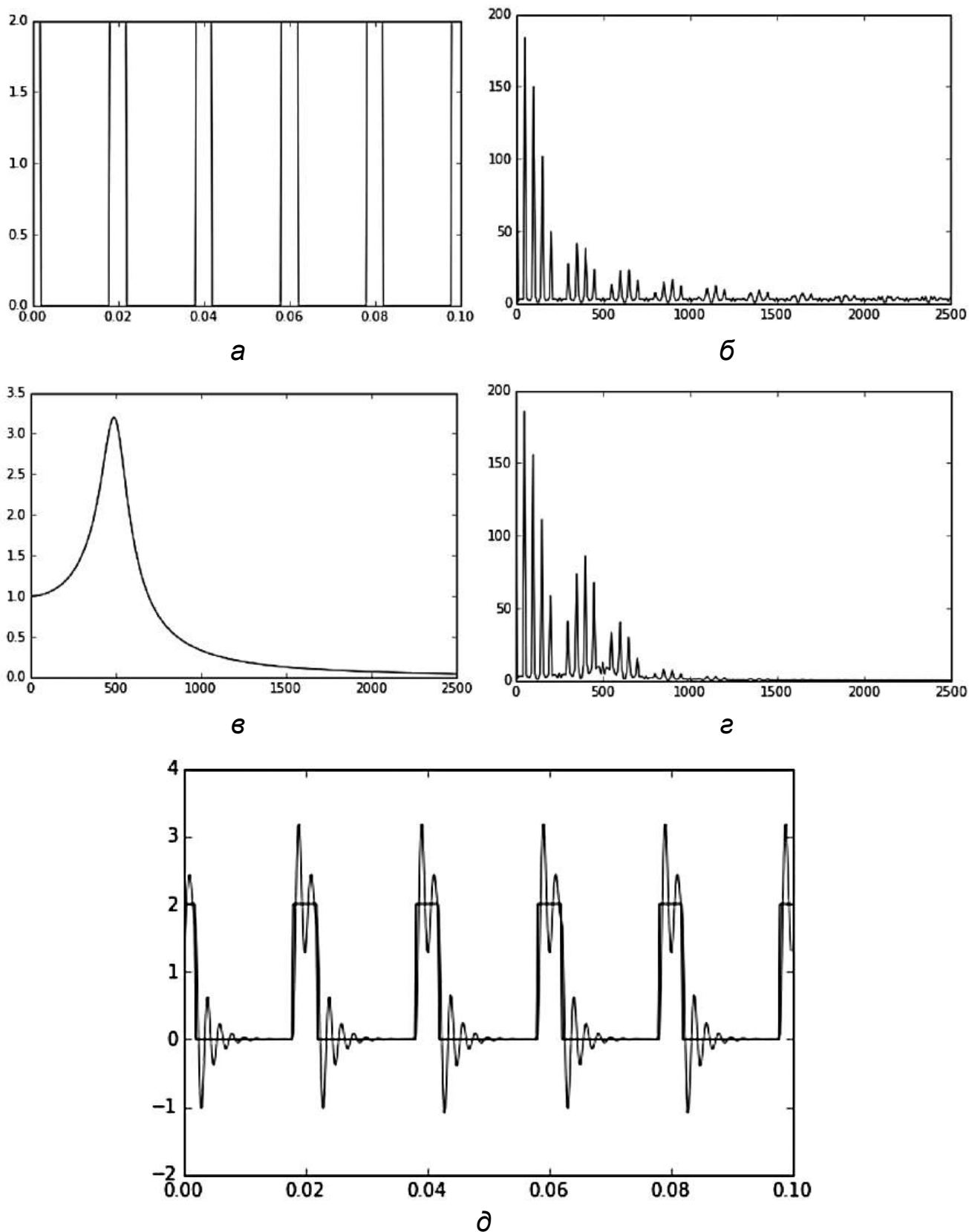


Рис. 3.4. Результати моделювання електронного пристрою:
 а – вхідний сигнал; б – спектр вхідного сигналу;
 в – АЧХ пристрою; г – спектр вихідного сигналу;
 д – сигнал на виході пристрою і його порівняння із вхідним сигналом

Контрольні запитання

1. Що таке комплексна передатна функція об'єкта (пристрою, електричної схеми)?
2. Що таке АЧХ і ФЧХ об'єкта (пристрою, електричної схеми)?
3. Як побудувати АЧХ- і ФЧХ-пристрої засобами мови програмування *Python*?
4. Які функції графічної бібліотеки мови програмування *Python* є необхідними для будування графіків функцій і сигналів?
5. Як за заданою математичною моделлю сигналу на вході пристрою обчислити параметри моделювання?
6. Які бібліотеки мови програмування *Python* є необхідними для моделювання роботи пристрою?
7. Як визначити частоту дискретизації вхідного сигналу при моделюванні електронного пристрою?
8. Як визначити розмір масиву для дискретної моделі вхідного сигналу?
9. Як розрахувати спектр вхідного сигналу?
10. Як визначити розмір масиву для зберігання частотної характеристики пристрою?
11. Як визначити частоти для обчислення передатної функції пристрою?
12. Як розрахувати спектр вихідного сигналу?
13. Як розрахувати частоти, на яких обчислено спектр сигналу?
14. Які функції бібліотеки *numpy* мови програмування *Python* є необхідними для обчислення перетворення Фур'є?
15. Якими функціями мови програмування *Python* можна створити нові масиви?
16. У яких випадках у мові програмування *Python* можна перемножувати масиви безпосередньо без використання операторів циклу?
17. Як побудувати графік комплексної частотної характеристики радіоелектронного пристрою?
18. Як побудувати графіки спектра сигналу й частотної характеристики пристрою в одній системі координат з використанням графічної бібліотеки мови програмування *Python*?
19. Як розрахувати форму вихідного сигналу?

4. КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ВИПАДКОВИХ ВЕЛИЧИН, ДІЙ І ЧИННИКІВ

Під час розроблення й проектування електронних приладів і систем, а також технологічних процесів їх виробництва доводиться враховувати випадковий характер параметрів комплектувальних електрорадіоелементів, конструктивних розмірів, вхідних сигналів, зовнішніх впливів (кліматичних, біологічних, радіаційних і т. д.), суб'єктивні чинники, пов'язані з участю людини-оператора (час реакції, неточність рухів), наявність наведень і завад, а також невідомі процеси в пристроях (застарівання, зношення, відмови). Причому в реальних умовах має місце спільна дія безлічі чинників і їх взаємний вплив (наприклад, як параметри, так і швидкість застарівання радіоелементів залежать від температури навколишнього середовища).

Моделювання випадкових чинників, сигналів і впливів базується на теорії ймовірностей, теорії випадкових функцій і випадкових процесів. При цьому ймовірні моделі поділяють на такі:

- дискретні й безперервні;
- залежні й незалежні;
- випадкові величини й процеси;
- одновимірні й багатовимірні (векторні).

4.1. Імовірнісні моделі дискретних випадкових величин

Припустимо, випробування проводиться N разів, результатом якого можуть бути M подій a_k , $k=1, M$ ($N \gg M$). Для кожної події a_k підраховується, скільки разів сталася ця подія: $a_k \rightarrow n_k$, $\sum_{k=1}^M n_k = N$.

Очевидно, що n_k є функцією N . Відношення n_k/N називають відносною частотою появи події a_k . При $N \rightarrow \infty$ відношення $n_k/N \rightarrow P(a_k)$,

$\lim_{N \rightarrow \infty} \frac{n_k}{N} = P(a_k)$, де $P(a_k)$ – імовірність події a_k .

Імовірність події – це міра об'єктивної можливості його появи. Подію, яка обов'язково має відбутися (або вже сталася), називають достовірною. Імовірність цієї події вважають такою, що дорівнює одиниці. Імовірність неможливої події беруть такою, що дорівнюють нулю. Таким чином, $0 \leq P(A) \leq 1$.

Дискретною випадковою величиною (ДВВ) A називають випадкову величину, єдино можливі значення якої є дискретними величинами a_k , $k=1, M$ (наприклад, пульс – кількість ударів серця за хвилину).

Вичерпною характеристикою ДВВ є її закон розподілу, який задається:

- 1) у вигляді таблиці;
- 2) у вигляді діаграм розподілу (рис. 4.1);
- 3) аналітично: $P_k = P\{A = a_k\}$.

Слід зазначити, що обов'язковою є умова

$$\sum_{k=1}^M P_k = 1.$$

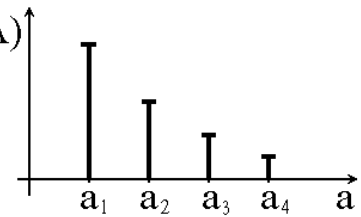


Рис. 4.1. Діаграма закону розподілу

Важливими для практики аналітичними моделями ДВВ є біноміальний закон розподілу і закон розподілу Пуассона.

4.1.1. Біноміальний закон розподілу

Припустимо, що при N незалежних випробуваннях подія A настає k разів. Подія A в кожному експерименті може виникнути не більше одного разу і з ймовірністю Q . Таким чином, випадкова $k = 1, 2, 3, \dots, N$. Ймовірність того, що при N випробуваннях подія A настане точно k разів, визначається формулою

$$p_k = C_N^k Q^k (1-Q)^{N-k},$$

де $C_N^k = \frac{N!}{k!(N-k)!}$.

Для цього закону середнє значення $m_k = \sum_{k=0}^N k p_k = NQ$, дисперсія

$$\sigma_k^2 = NQ(1-Q) = \sum_{k=0}^N k^2 p_k.$$

Біноміальним законом описується, наприклад, кількість відмов під час випробування N виробів, які відмовляють з ймовірністю Q . Форма цього закону розподілу істотно залежить від ймовірності Q (рис. 4.2).

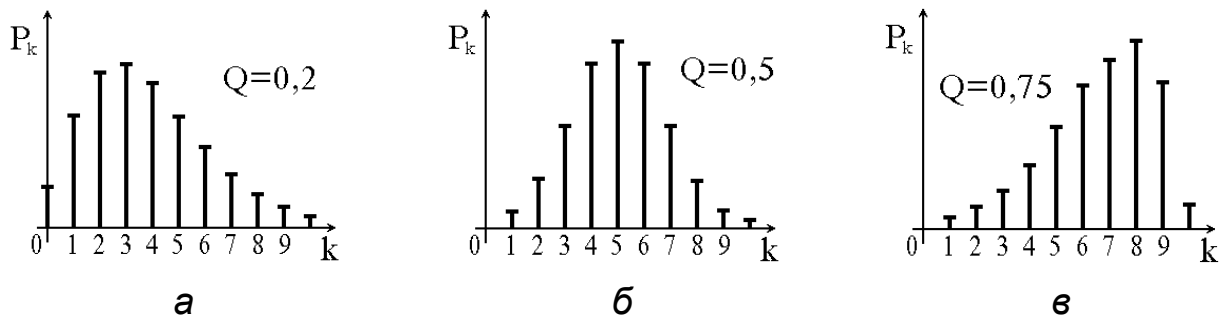


Рис. 4.2. Біноміальний закон розподілу

4.1.2. Закон Пуассона

При $N \rightarrow \infty$ біноміальний закон розподілу перетворюється на закон Пуассона. Якщо взяти $NQ = a$, то $p_k = \frac{a^k}{k!} e^{-a}$, де a – параметр закону (середнє значення кількості подій), Q – імовірність одиничної події при N незалежних випробуваннях, коли $N \rightarrow \infty$. Законом Пуассона описується багато фізико-технічних явищ, що мають імпульсний характер, наприклад:

- 1) кількість розрядів блискавок під час грози (створюють електромагнітні імпульсні завади роботі ЕА);
- 2) кількість відмов апаратури за деякий проміжок часу;
- 3) ударні навантаження під час транспортування апаратури;
- 4) кількість телефонних дзвінків (у системах зв'язку) тощо.

Середнє значення m_k і дисперсія σ_k^2 для закону Пуассона дорівнюють a .

4.2. Імовірнісні моделі безперервних випадкових величин

Випадкові величини, можливі значення яких відрізняються один від одного на нескінченно малу величину, називають безперервними випадковими величинами (БВВ).

Моделі чинників, сигналів, впливів у вигляді безперервних випадкових величин використовують, наприклад, для опису:

- розкиду параметрів комплектувальних радіоелементів;
- вхідних сигналів і зовнішніх впливів;
- вихідних реакцій систем тощо.

Якщо для опису ДВВ використовуються розподіли ймовірностей, то для опису БВВ – щільності розподілу ймовірностей (рис. 4.3).

Для БВВ x інтервал між сусідніми можливими значеннями прямує до нуля й імовірність значення x описується щільністю розподілу:

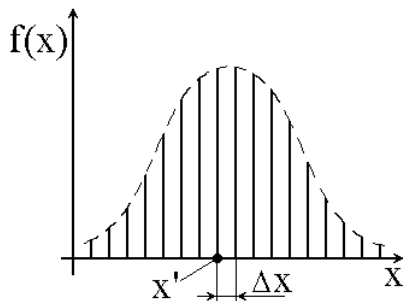


Рис. 4.3. Щільність розподілу

$$f(x) = P \frac{[x' \leq x \leq x' + \Delta x]}{\Delta x}. \quad (4.1)$$

Щільність розподілу $f(x)$ завжди є дійсною невід'ємною функцією. Оскільки в загальному випадку повна множина значень

$$x \in [-\infty, \infty], \text{ то } \int_{-\infty}^{\infty} f(x) dx = 1.$$

Імовірність того, що випадкове значення x буде знаходитися в інтервалі Δx , визначають за формулою

$$P_{\Delta x}(x \in [x', x' + \Delta x]) = \int_{x'}^{x' + \Delta x} f(x) dx.$$

Імовірність того, що значення x не перевищує деякого значення X , тобто

$$F(X) = P(x \leq X) = \int_{-\infty}^X f(x) dx, \quad (4.2)$$

залежить від цього значення і має назву функції розподілу.

Для опису характеру групування БВВ широко використовуються такі числові характеристики, як моменти розподілу k -го порядку – початкові m_k і центральні μ_k :

$$m_k = M_k(x) = \int_{-\infty}^{\infty} x^k f(x) dx, \quad \mu_k = \int_{-\infty}^{\infty} (x - m_1)^k f(x) dx,$$

де m_1 – перший початковий момент ($k=1$) – математичне сподівання БВВ x ,

$$m_1 = m_x = \int_{-\infty}^{\infty} x f(x) dx. \quad (4.3)$$

Другий початковий момент $m_2 = M_2(x) = \int_{-\infty}^{\infty} x^2 f(x) dx$ – це середнє значення квадрата БВВ. Другий центральний момент – дисперсія БВВ

$$\mu_2 = \int_{-\infty}^{\infty} (x - m_x)^2 f(x) dx, \quad (4.4)$$

– характеризує розкид випадкової величини відносно середнього значення. Для порівняння за одиницями виміру з математичним сподіванням замість дисперсії часто використовується так зване середньоквадратичне відхилення

$$\sigma_x = \sqrt{\mu_2}. \quad (4.5)$$

Форму закону розподілу характеризують третій і четвертий центральні моменти. На практиці використовують коефіцієнти асиметрії та ексцесу:

$$\beta_1 = \frac{M[(x - m_x)^3]}{\sigma_x^3}, \quad (4.6)$$

$$\beta_2 = \frac{M[(x - m_x)^4]}{\sigma_x^4} - 3. \quad (4.7)$$

4.2.1. Рівномірний закон розподілу

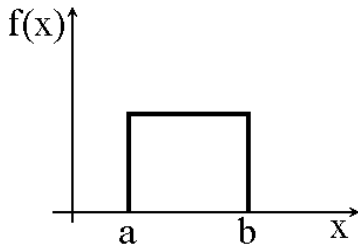


Рис. 4.4. Щільність
рівномірного розподілу

Рівномірним (рівноймовірним) називають закон розподілу такої БВВ, яка з однаковою ймовірністю може набувати будь-якого значення в заданих межах (рис. 4.4). Такому закону розподілу підпорядковується, наприклад, помилка при наближених обчисленнях, що виникає через округлення чисел до будь-якої значущої цифри. При цьому межі помилки $[a, b]$ будуть відповідати

половині значущої цифри, до якої здійснюють округлення. Щільність рівномірного розподілу та функція розподілу

$$f(x) = \begin{cases} 0, & x < a, \\ \frac{1}{b-a}, & a < x < b, \\ 0, & x > b. \end{cases}$$

Функція розподілу має вигляд

$$F(x) = \begin{cases} 0, & x \leq a, \\ \frac{x-a}{b-a}, & a < x < b, \\ 1, & x > b, \end{cases}$$

математичне сподівання

$$m_x = \frac{1}{b-a} \int_a^b x dx = \frac{a+b}{2},$$

дисперсія

$$\sigma^2 = \frac{1}{b-a} \int_a^b (x - m_x)^2 dx = \frac{1}{12} (b-a)^2.$$

Асиметрія $\beta_1 = 0$, а ексцес $\beta_2 = -1,2$.

4.2.2. Експоненційний розподіл

Експоненційний розподіл (рис. 4.5) широко використовують у теорії надійності й теорії масового обслуговування. Щільність експоненційного розподілу $f(x) = ae^{-ax}$ і функція розподілу $F(x) = 1 - e^{-ax}$, математичне сподівання $m_x = \frac{1}{a}$, дисперсія $\sigma_x^2 = \frac{1}{a^2}$, асиметрія $\beta_1 = 2$, ексцес $\beta_2 = 6$.

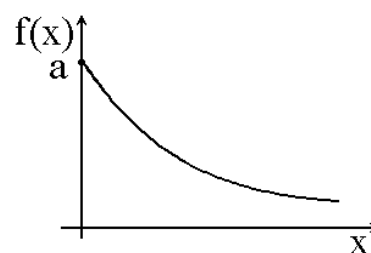


Рис. 4.5. Щільність експоненціального розподілу

4.2.3. Закон розподілу Сімпсона

Закону розподілу Сімпсона підпорядковується розподіл суми двох незалежних рівномірно розподілених СВ.

Щільність розподілу Сімпсона (рис. 4.6)

$$f(x) = \begin{cases} 0, & -\infty < x < a, \\ \frac{4(x-a)}{(b-a)^2}, & a < x < \frac{a+b}{2}, \\ \frac{4(b-x)}{(b-a)^2}, & \frac{a+b}{2} < x < b, \\ 0, & b < x < \infty, \end{cases}$$

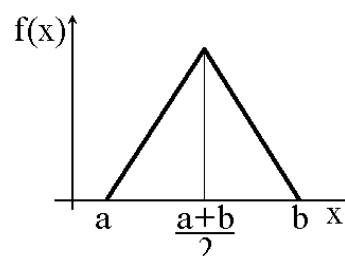


Рис. 4.6. Щільність розподілу Сімпсона

функція розподілу

$$F(x) = \begin{cases} 0, & x < a, \\ \frac{2(x-a)^2}{(b-a)^2}, & a < x < \frac{a+b}{2}, \\ 1 - \frac{2(b-x)^2}{(b-a)^2}, & \frac{a+b}{2} < x < b, \\ 1, & b < x. \end{cases}$$

Середнє значення $m_x = \frac{1}{2}(a+b)$, дисперсія $\sigma_x^2 = \frac{1}{24}(b-a)^2$, асиметрія $\beta_1 = 0$, ексцес $\beta_2 = 0,6$.

4.2.4. Закон розподілу арксинуса

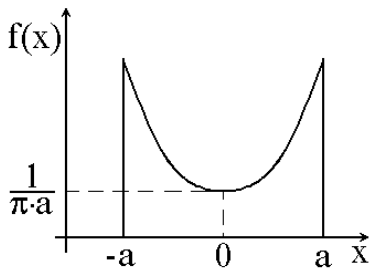


Рис. 4.7. Щільність розподілу арксинуса

Закону розподілу арксинуса відповідає розподіл миттєвих значень синусоїди з випадковою початковою фазою. Щільність розподілу арксинуса (рис. 4.7)

$$f(x) = \begin{cases} 0, & x < -a, \\ \frac{1}{\pi} \frac{1}{\sqrt{a^2 - x^2}}, & -a < x < a, \\ 0, & x > a, \end{cases}$$

функція розподілу

$$F(x) = \begin{cases} \frac{1}{2} + \frac{1}{\pi} \arcsin \frac{x}{a}, & x \in [-a, a], \\ 1, & x \notin [-a, a]. \end{cases}$$

Середнє значення $m_x = 0$, дисперсія $\sigma^2 = \frac{a^2}{2}$, асиметрія $\beta_1 = 0$, ексцес $\beta_2 = -1,5$.

4.2.5. Закон розподілу Релея

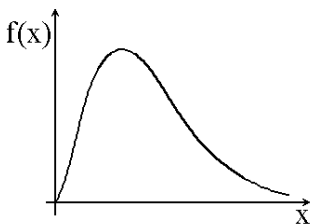


Рис. 4.8. Щільність розподілу Релея

Цей закон широко використовують у радіотехніці для моделювання обвідної суміші високочастотного сигналу й шуму. Щільність розподілу Релея (рис. 4.8)

$$f(x) = \begin{cases} 0, & x < 0, \\ \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}, & x > 0, \end{cases}$$

функція розподілу

$$F(x) = \begin{cases} 0, & x < 0, \\ 1 - e^{-\frac{x^2}{2\sigma^2}}, & x > 0. \end{cases}$$

Середнє значення $m_x = \frac{\sigma\sqrt{\pi}}{\sqrt{2}} \approx 1,25\sigma$, дисперсія $D_x = \frac{4-\pi}{2}\sigma^2 \approx 0,43\sigma^2$, асиметрія $\beta_1 = 0,63$, ексцес $\beta_2 = 0,3$.

4.2.6. Нормальний закон розподілу

Згідно з центральною граничною теоремою сума великої кількості незалежних випадкових величин має нормальний розподіл. За нормальним законом описується безліч різних за своєю природою випадкових величин. Його використовують у різних областях науки й техніки як імовірнісну модель шуму, яка є однією з основних імовірнісних моделей у статистичній радіотехніці.

Щільність нормального розподілу (рис. 4.9)

$$f(x) = N(x; a; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-a)^2}{2\sigma^2}}.$$

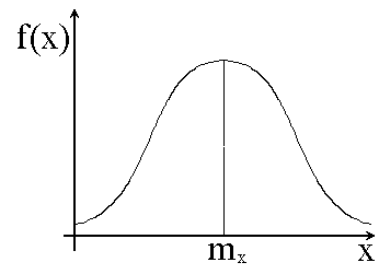


Рис. 4.9. Щільність нормального розподілу

Функція розподілу нормального закону не виявляється в аналітичних функціях:

$$F(x) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^x e^{-\frac{(x-a)^2}{2\sigma^2}} dx.$$

Середнє значення $m_x = a$, дисперсія $D_x = \sigma^2$, асиметрія й ексцес дорівнюють нулю.

4.3. Моделювання випадкових сигналів і впливів

При проведенні статистичного імітаційного експерименту машинним шляхом імітуються реалізації x_{ik} випадкових первинних параметрів x_i , за якими відповідно до моделі системи, що моделюється, обчислюють реалізації y_{ik} вихідних параметрів y_i . Статистичне оброблення отриманих масивів y_{ik} дає змогу визначити необхідні статистичні характеристики. Таким чином, основні труднощі при статистичному моделюванні полягають у формуванні реалізацій випадкових значень первинних параметрів.

4.3.1. Імітація значень рівномірно розподіленої величини

Більшість алгоритмів моделювання випадкових величин базується на функціональному перетворенні реалізацій випадкової величини, рівномірно розподіленої в інтервалі $[0, 1]$. Таку величину, яка підпорядковується закону розподілу зі щільністю розподілу

$$f(x) = 1, \quad x \in [0, 1], \quad (4.8)$$

називають стандартною.

Відомо, що закону розподілу (4.8) підпорядковується помилка округлення при наближених обчисленнях. наприклад, $5,7 \cdot 6,2 = 35,34$. Якщо результат округлити до 35, тобто виокремити цілу частину результату, то помилка округлення 0,34 буде знаходитися в інтервалі від 0 до 1. На основі цього будують алгоритми генерації випадкових чисел з рівномірним законом розподілу.

Для отримання реалізацій рівномірно розподілених в інтервалі $[0, 1]$ випадкових чисел застосовують мультиплікативний алгоритм, який полягає в такому:

- 1) задають початкове випадкове непарне число x_0 , що належить до інтервалу $[0, 1]$ і не є кратним 5, наприклад, $x_0 = 0,673783$;
- 2) множать x_0 на коефіцієнт k , значення якого може бути довільним, проте, як впливає з досліджень, серед безлічі двозначних чисел тільки два з них (37 і 73) забезпечують стійку роботу алгоритму;
- 3) в отриманому значенні виокремлюють дробову частину, яка і є новим випадковим числом:

$$x_i = x_{i-1} k - \text{int}(x_{i-1} k),$$

де $\text{int}(\)$ – відповідна функція виокремлення цілої частини числа;

- 4) дії, викладені в пп. 2 і 3, повторюють N разів, де N – необхідна кількість реалізацій рівномірно розподіленого числа.

Приклад «генерації» випадкових чисел за наведеним алгоритмом із використанням мови програмування *Python* показано на рис. 4.10.

У різних системах програмування є стандартні процедури (підпрограми) генерації реалізацій таких чисел. Зокрема, у мові програмування *Python* для вирішення завдань моделювання випадкових чисел використовується модуль **numpy.random**. Для отримання випадкових чисел з рівномірним законом розподілу можна використати функцію *random*, як показано на рис. 4.11.

```

import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
n=1000 #кількість відліків випадкової величини
x=np.empty(n); I=np.empty(n)
x[0]=0.63764321
for i in range(1,n):
    I[i]=i #номер відліку для побудови графіка
    xx=x[i-1]*37
    x[i]=xx-int(xx) #наступні 1...n «випадкові» числа
fig1=plt.figure()
plt.plot(I[0:100], x[0:100], 'o') #перших 100 «випадкових» чисел

```

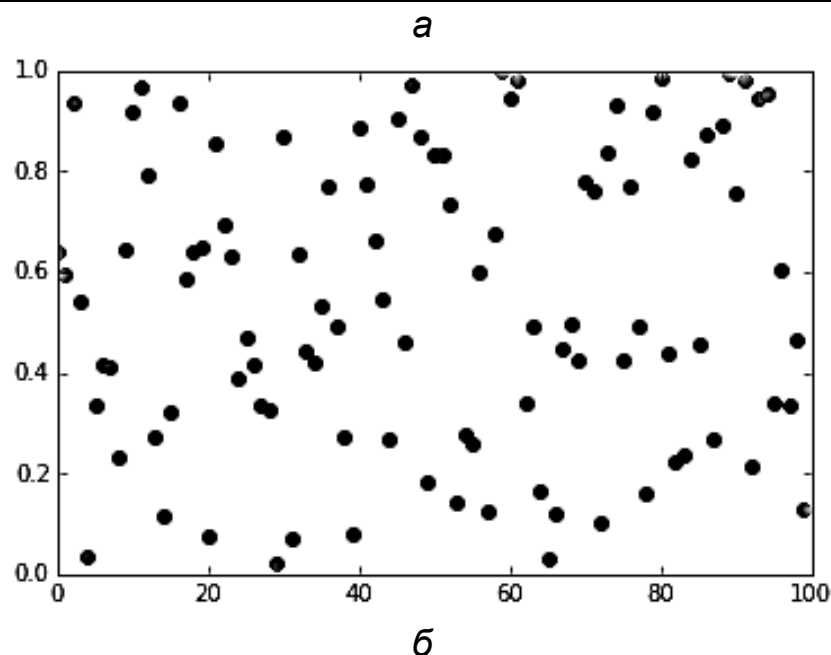


Рис. 4.10. Приклад програми (а) та результати моделювання (б) випадкових чисел з рівномірним законом розподілу

```

import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
import numpy.random as nr
n=100 #кількість відліків випадкової величини
xr=np.empty(n); I=np.empty(n)
for i in range(1,n):
    I[i]=i
xr=nr.random(n)
fig1=plt.figure()
plt.plot(I[0:100],xr[0:100], 'o') #перших 100 «випадкових» чисел

```

Рис. 4.11. Приклад використання стандартної бібліотечної функції *random*

4.3.2. Моделювання випадкових подій

Моделювання простих випадкових подій. Припустимо, випадкова подія A може відбутися з імовірністю $P(A)=P$. Необхідно побудувати алгоритм формування події A . Сформуємо рівноймовірно розподілену випадкову величину x_i , $i=1...N$, де N – інтервал ($N\Delta t=T$ – час) очікування події A . Наприклад, $P=0,8$, тоді з урахуванням (4.8) імовірність того, що x_i буде меншим від значення 0,8 (рис. 4.12),

$$P(x_i < 0,8) = \int_0^{0,8} f(x) dx = 0,8,$$

звідки впливає алгоритм моделювання випадкових подій, зображений на рис. 4.13, де r_i – випадкова величина з рівноймовірним законом розподілу, P – імовірність події A .

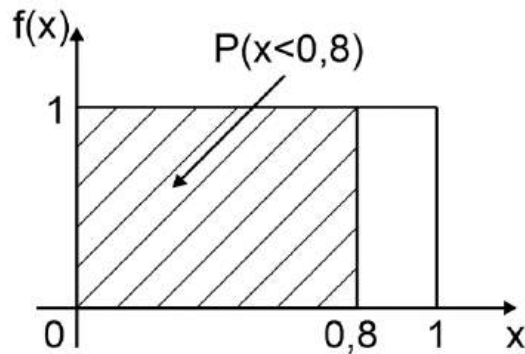


Рис. 4.12. Імовірність події

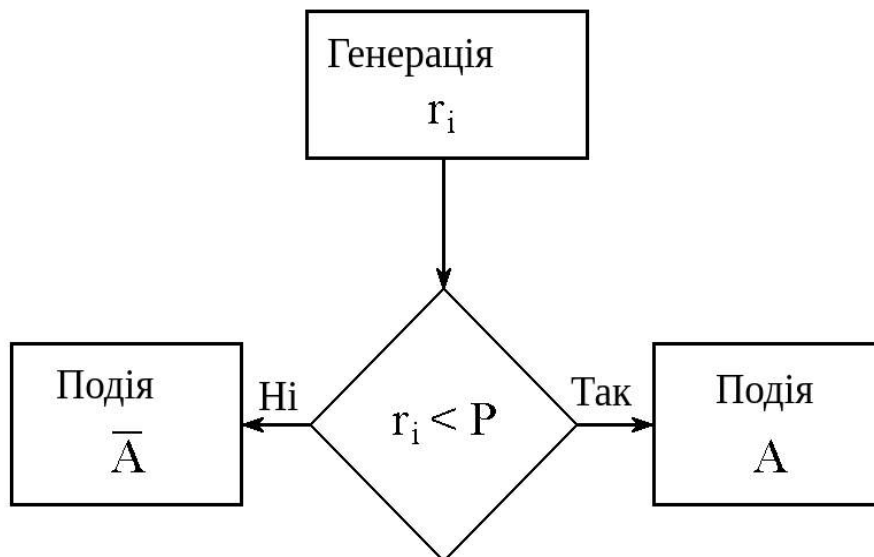


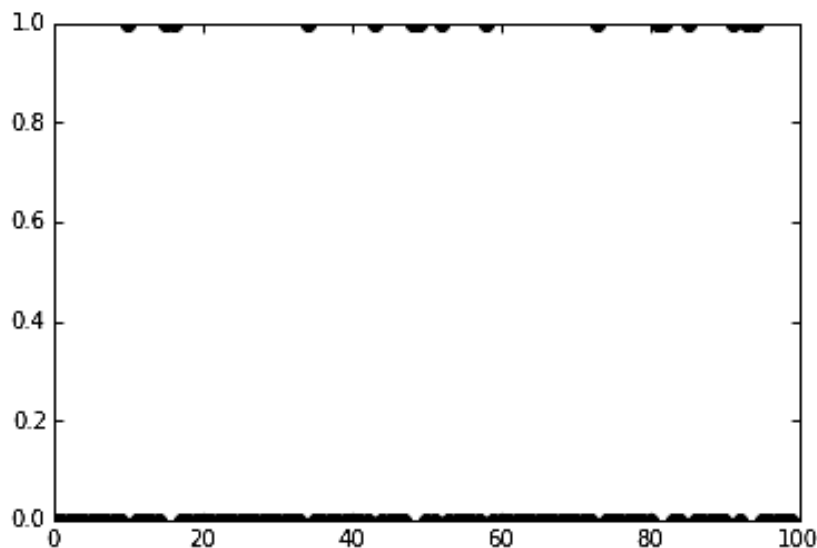
Рис. 4.13. Алгоритм моделювання випадкової події

При N зверненнях до генератора випадкових чисел ($i=1..N$) буде імітовано M подій. При цьому $\lim_{N \rightarrow \infty} \frac{M}{N} = P(A)$.

Приклад програми й результатів моделювання випадкової події A з ймовірністю появи P показано на рис. 4.14 (виникнення події марковано як нуль, відсутність – як одиниця). Як видно з рис. 4.14, б, випадкова подія не виникла 16 разів зі 100. Теоретично при ймовірності 0,8 подія не мала виникнути $(1 - 0,8) \cdot 100 = 20$ разів. Така розбіжність і є особливістю ймовірнісного моделювання. При наступному запуску програми відбувається випадкове змінення кількості подій, і тільки при дуже великому обсязі експериментів «експериментальна» ймовірність буде наближатися до теоретичної (заданої).

```
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
import numpy.random as nr
n=100 #кількість відліків
xr=np.empty(n); I=np.empty(n);
A=np.zeros(n)
xr=nr.random(n); P=0.8
for i in range(1,n):
    I[i]=i
    if xr[i]>P: A[i]=1
fig1=plt.figure()
plt.plot(I[0:100], A[0:100], 'o')
```

а



б

Рис. 4.14. Приклад програми (а) і результати моделювання (б) випадкових подій із заданою ймовірністю

Моделювання складних подій. Припустимо, C – складна подія, яка полягає в тому, що одночасно мають відбутися дві прості незалежні події: A з імовірністю $P(A)$ і B з імовірністю $P(B)$. При цьому можливими є чотири події: $C = A \cap B$, $\bar{A} \cap B$, $A \cap \bar{B}$, $\bar{A} \cap \bar{B}$.

Для моделювання таких подій необхідно послідовно отримати два рівноймовірно розподілених числа r_1 і r_2 , за якими буде визначатися необхідна подія згідно з алгоритмом, показаним на рис. 4.15.

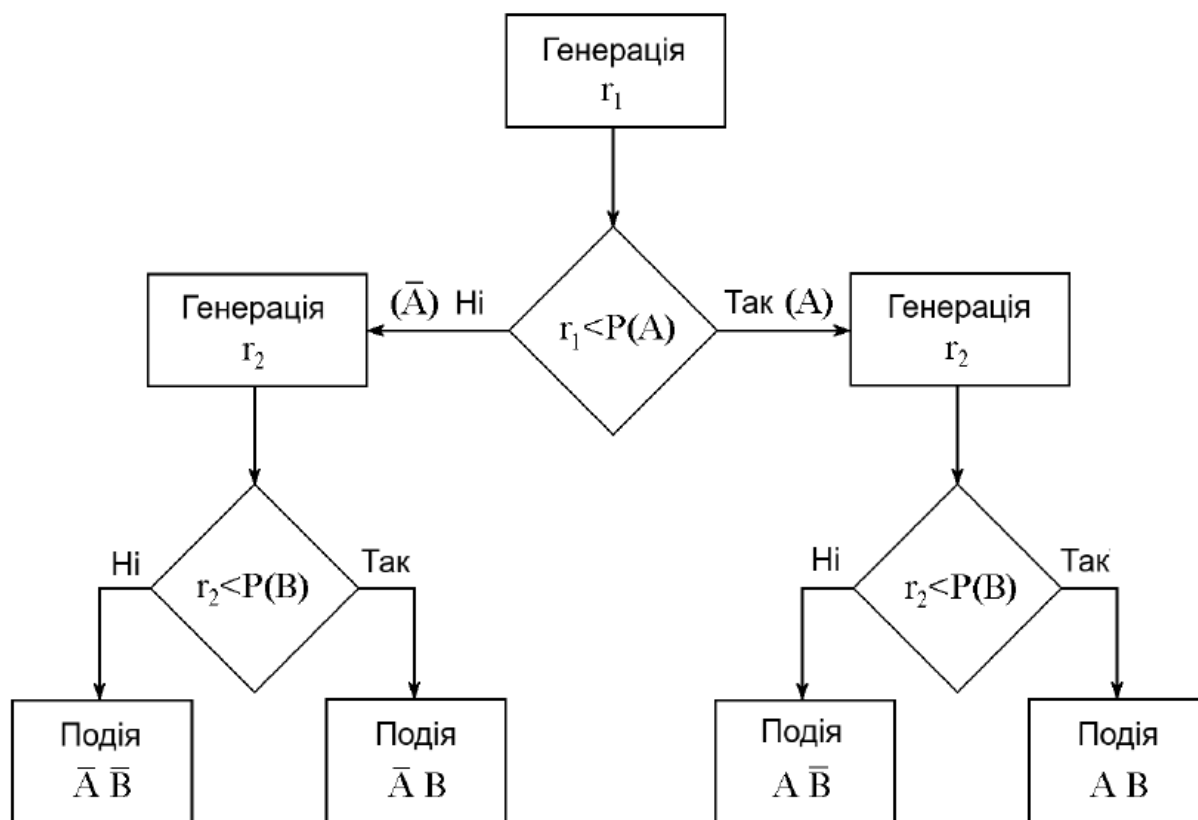


Рис. 4.15. Алгоритм моделювання складних подій

4.3.3. Моделювання неперервних випадкових величин

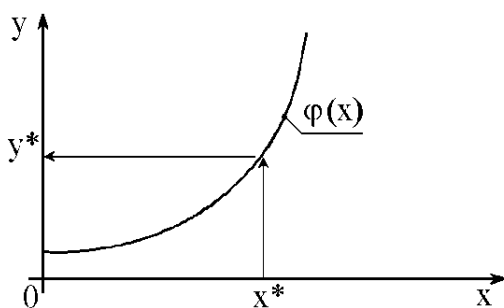


Рис. 4.16. Функціональне перетворення випадкової величини

Припустимо, що є випадкова величина зі щільністю розподілу $f(x)$. Випадкову величину x^* можна трансформувати у величину y^* з допомогою безперервного перетворення $y = \varphi(x)$, як показано на рис. 4.16.

Випадкова величина y має закон розподілу $f(y)$. Зважаючи на безперервність перетворення $y = \varphi(x)$ має місце збереження ймовірності:

$$P\{x_i < x\} = P\{y_i < y\} \quad \text{або} \quad F(x) = F(y),$$

де x_i, y_i – реалізації величин x і y , при цьому $y_i = \varphi(x_i)$, або

$$\int_{-\infty}^x f(x) dx = \int_{-\infty}^y f(y) dy. \quad (4.9)$$

Нехай x – стандартна величина r , рівномірно розподілена на інтервалі $[0...1]$. Тоді зі співвідношення (4.9) отримаємо

$$r = \int_{-\infty}^y f(y) dy. \quad (4.10)$$

Розв'язання рівняння (4.10) відносно y приводить до необхідної функції перетворення $y = \varphi(r)$ для отримання за реалізацією r_i стандартної випадкової величини r , а за реалізацією y_i – випадкової величини y з необхідним законом розподілу $f(y)$. Це рівняння можна записати так:

$$r = F(y), \quad (4.11)$$

а його розв'язок –

$$y = F^{-1}(r), \quad (4.12)$$

де F^{-1} – функція, обернена до F .

Таким чином, алгоритм моделювання випадкової величини y із заданим законом розподілу (щільністю розподілу $f(y)$ або функцією розподілу $F(y)$) можна отримати шляхом знаходження функції $F^{-1}(\cdot)$, оберненої до функції розподілу $F(y)$.

Якщо як вихідні дані задано щільність розподілу $f(y)$, то необхідно аналітично отримати вирази

$$F(y) = \int_{-\infty}^y f(y) dy \quad \text{і} \quad y = F^{-1}(r),$$

після чого за реалізацією стандартних випадкових чисел r_i , рівномірно розподілених на інтервалі $[0...1]$, можна отримати реалізацію випадкової величини $y_i = F^{-1}(r_i)$, як показано на рис. 4.17.

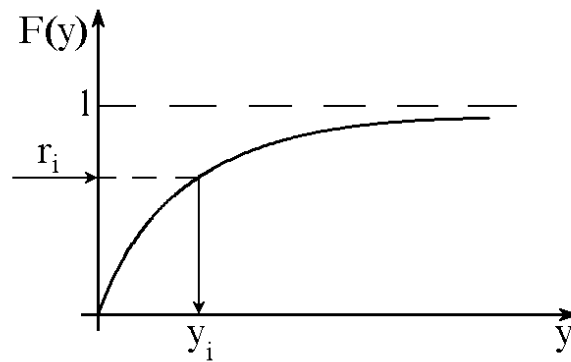


Рис. 4.17. Моделювання неперервних випадкових величин методом оберненої функції

Застосування методу оберненої функції (4.12) покажемо на прикладі експоненціального розподілу $f(y) = \lambda e^{-\lambda y}$.

Функція розподілу, яка може набувати значень в інтервалі $[0 \dots 1]$,

$$F(y) = \int_{-\infty}^y f(y) dy = \int_0^y \lambda e^{-\lambda y} dy = 1 - e^{-\lambda y}.$$

Дорівнюємо її значення до випадкової величини r з рівноймовірним розподілом в інтервалі $[0 \dots 1]$:

$$1 - e^{-\lambda y} = r,$$

звідки обернена функція

$$y = -\frac{1}{\lambda} \ln(1 - r).$$

Тоді алгоритм отримання реалізації випадкової величини y з експоненціальним законом розподілу з урахуванням симетричності рівномірного закону розподілу матиме вигляд

$$y_i = -\frac{1}{\lambda} \ln(r_i),$$

а його програмну реалізацію показано на рис. 4.18.

Методом оберненої функції можна отримати випадкові величини з будь-яким аналітично інтегрованим законом розподілу.

На жаль, не для всіх законів розподілу існує аналітична обернена функція $F^{-1}(\bullet)$. Зокрема, це спостерігається при широко використовуваному нормальному закону розподілу

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-m)^2}{2\sigma^2}\right\} = N(m, \sigma).$$

```
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
import numpy.random as nr
n=100 #кількість відліків
xr=np.empty(n)
I=np.empty(n)
E=np.empty(n)
L=1.0
xr=nr.random(n)
for i in range(1,n):
    I[i]=i
    E[i]=-np.log(xr[i]/L)
fig1=plt.figure()
plt.plot(I[1:100], E[1:100], 'o')
```

a

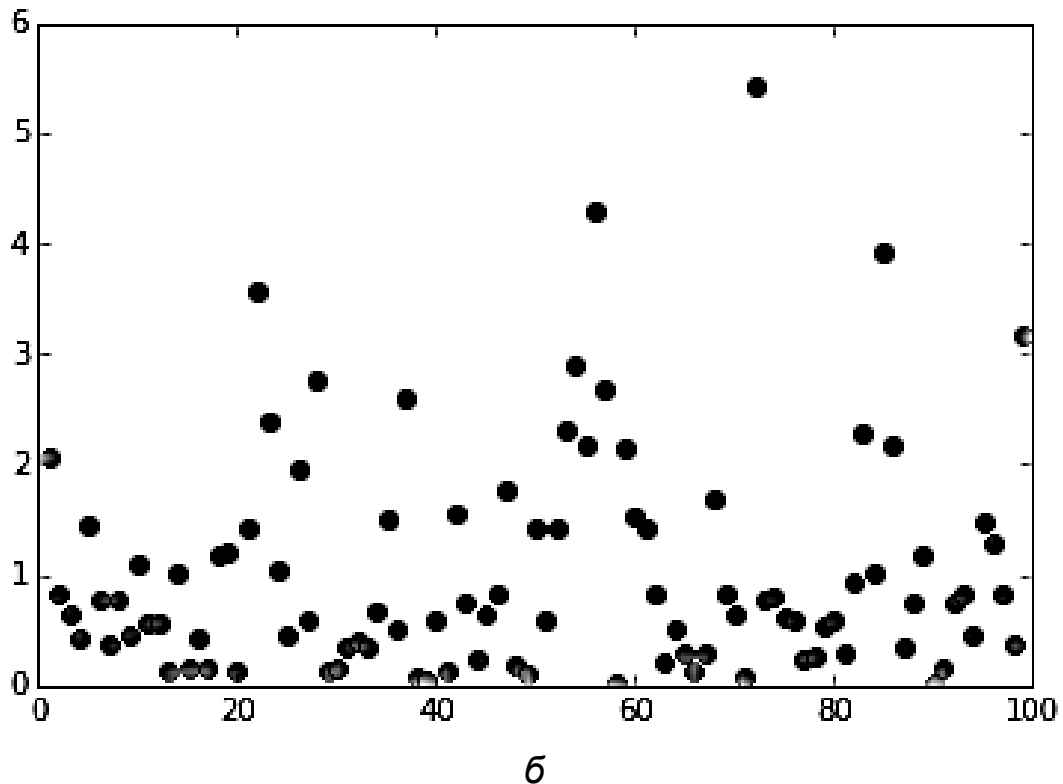


Рис. 4.18. Приклад програми (а) і результати моделювання (б) випадкової величини методом оберненої функції

У таких випадках застосовують інші методи формування реалізацій, що базуються на окремих властивостях розподілу. Відомо, що закон розподілу суми досить великої кількості довільно розподілених доданків (центральна гранична теорема) наближається до нормального. З урахуванням цієї властивості алгоритм формування реалізацій y_i з нормальним розподілом з реалізацій r_i стандартної величини з рівномірним розподілом будують таким чином:

1) беруть n незалежних реалізацій величини r_i , рівномірно розподілених в інтервалі $[0...1]$:

$$\{r_j\}, \quad j=1...n;$$

2) формують суму

$$z = \sum_{j=1}^n r_j,$$

яка є розподіленою за нормальним законом і має параметри

$$m_z = \frac{n}{2}, \quad \sigma_z = \sqrt{\frac{n}{12}};$$

3) центрують і нормують величину z , зводячи її до стандартної нормальної величини z_0 з розподілом $N(0,1)$:

$$z_0 = \frac{z - m_z}{\sigma_z};$$

4) величину z_0 перетворюють на нормально розподілену випадкову величину y з розподілом $N(m, \sigma)$:

$$y = m + \sigma z_0;$$

5) остаточно записують алгоритм формування реалізацій:

$$y_i = m + \sigma \sqrt{\frac{12}{n}} \left\{ \sum_{j=1}^n r_j - \frac{n}{2} \right\}. \quad (4.13)$$

Цей алгоритм дає змогу отримати хороше наближення до нормального закону вже при $n > 10$. Найбільш зручною є така форма алгоритму (4.13) при $n=12$ і $n=48$:

$$y_i = m + \sigma \left\{ \sum_{j=1}^{12} r_j - 6 \right\}, \quad y_i = m + 2\sigma \left\{ \sum_{j=1}^{48} r_j - 24 \right\}.$$

У першому випадку отримаємо швидкий алгоритм із середньою точністю формування випадкових чисел з нормальним законом розподілу, у другому – більш повільний при більш високій точності. Слід зазначити, що збільшення обсягу підсумовування вибірки в (4.13) $n > 50$ не призводить до помітного поліпшення якості генератора випадкових чисел. Приклад програмної реалізації алгоритму (4.13) показано на рис. 4.19.

На основі випадкових величин з нормальним законом розподілу можна отримати випадкові величини з іншими законами розподілу шляхом застосування нелінійних перетворень. Наприклад, випадкові величини із законом розподілу Релея

$$f(y) = \frac{y}{\sigma^2} e^{\frac{-y^2}{2\sigma^2}}, \quad y > 0, \quad \sigma = 1,$$

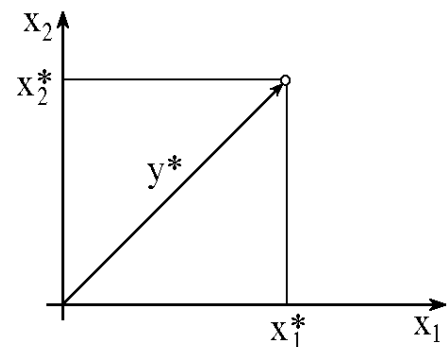


Рис. 4.20. Довжина випадкового вектора

можна отримати з двох нормально розподілених чисел x_1 і x_2 шляхом перетворення $y_i = \sqrt{x_{1i}^2 + x_{2i}^2}$, оскільки під час перетворення нормально розподіленої двовимірної випадкової величини $N(0,1)$ (рис. 4.20, точка на площині) з декартової системи координат на полярну довжина радіуса-вектора підпорядковується закону розподілу Релея.

Слід зазначити, що в бібліотеці **numpy.random** є процедури отримання випадкових чисел з найбільш поширеними в радіотехніці законами розподілу ймовірностей, зокрема:

- з рівноймовірним *rand*;
- нормальним *standard_normal*;
- експоненційним *exponential*;
- Релея *Rayleigh*.

Крім того бібліотека **numpy.random** містить кілька процедур формування масивів випадкових чисел з менш уживаними законами розподілу, зокрема з логарифмічно нормальним, бета- і гама-розподілами, розподілами Лапласа, Парето, Пуассона тощо.

```

import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
n=12 #кількість відліків рівномірних чисел
N=100 #кількість відліків нормальних чисел
I=np.empty(N)
X=np.empty(N)
mX=1.0
sigmaX=2.0
x0=0.63764321
for i in range(N):
    I[i]=i
    s=-n/2.0
    for j in range(1,n):
        x0=x0*37
        x0=x0-int(x0)
        s=s+x0
    X[i]=mX+sigmaX*s
# Графіки
fig1=plt.figure()
plt.plot(I[1:100], X[1:100], 'o')

```

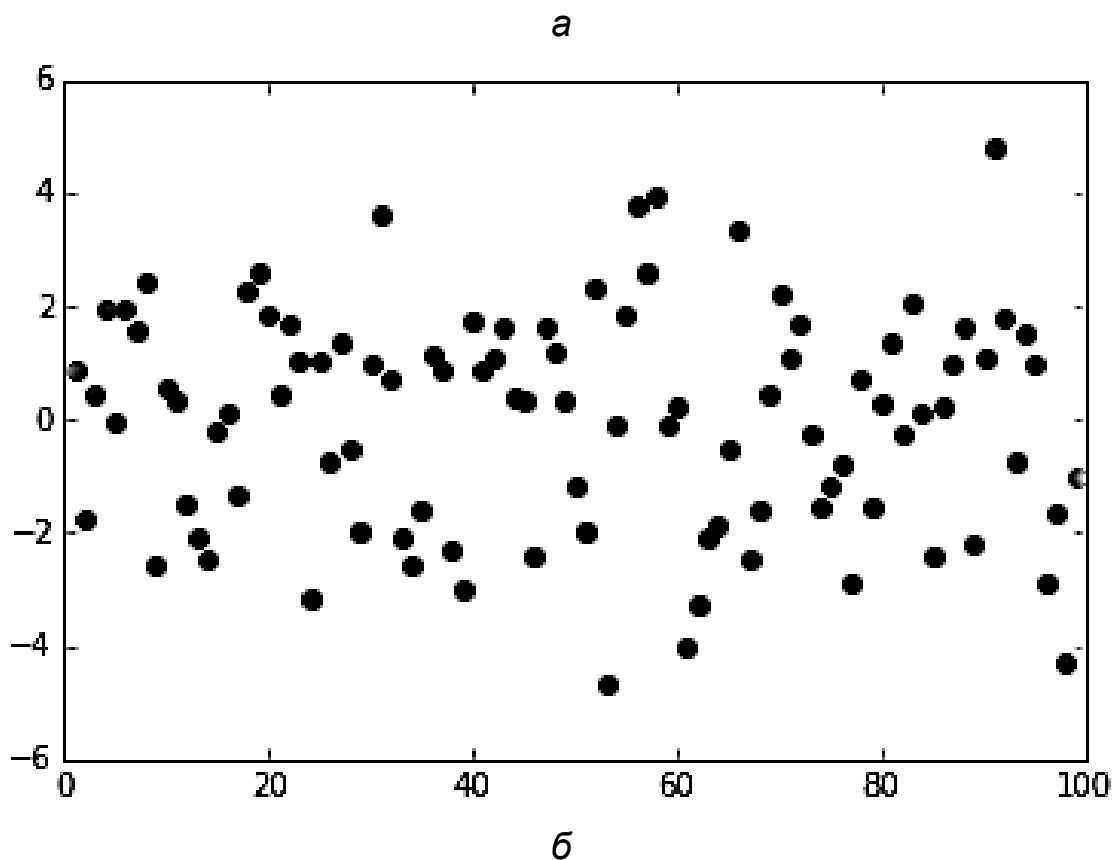


Рис. 4.19. Приклад програми (а) і результати моделювання (б) випадкової величини з нормальним законом розподілу

Контрольні запитання

1. Що таке ймовірнісна модель?
2. Які ймовірнісні моделі використовуються при комп'ютерному моделюванні дискретних випадкових величин?
3. Які моделі використовуються при комп'ютерному моделюванні безперервних випадкових величин?
4. Як моделювати випадкові величини з рівномірним законом розподілу?
5. Як змоделювати випадкові події?
6. Як змоделювати дискретну випадкову величину?
7. Як змоделювати випадкову величину методом оберненої функції?
8. Як отримати випадкову величину з нормальним законом розподілу?
9. Що таке центрована випадкова величина?
10. Як з масиву центрованих випадкових величин отримати масив випадкових величин із заданим математичним сподіванням?
11. Чим відрізняється щільність розподілу випадкової величини від функції розподілу?
12. Що таке моменти розподілу випадкової величини? Який вони мають фізичний зміст?
13. Чи зміниться форма закону розподілу ймовірностей, якщо випадкові числа помножити на фіксований коефіцієнт? Які параметри закону розподілу при цьому зміняться?
14. Чи зміниться форма закону розподілу ймовірностей, якщо випадкові числа підставити у деяку нелінійну функцію, наприклад логарифм?
15. Яку бібліотеку мови програмування *Python* можна використовувати для отримання випадкових чисел?

5. ОБРОБЛЕННЯ ДАНИХ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ ВИПАДКОВИХ ВЕЛИЧИН

5.1. Методика оброблення даних імітаційного експерименту

Для статистичного оброблення даних імітаційного експерименту застосовують типові алгоритми статистичного аналізу спостережень, основними з яких є такі:

1. *Знаходження вибіркового середнього значення* (оцінка математичного сподівання)

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i,$$

де N – кількість відліків; x_i – значення відліків. Розрахована в такий спосіб величина є незміщеною оцінкою істинного середнього значення.

2. *Визначення вибіркового середньоквадратичного відхилення*

$$\hat{\sigma} = \sqrt{\sum_{i=1}^N \frac{(x_i - \bar{x})^2}{N-1}}.$$

Величина σ^2 є незміщеною оцінкою дисперсії випадкового процесу.

3. *Будування гістограми щільності розподілу*. При обсязі вибірки N щільність можна оцінити як

$$\hat{f}(x) = \frac{N_x}{N\Delta x},$$

де Δx – вузький інтервал, симетричний відносно точки x ; N_x – кількість потраплянь в інтервал Δx .

При рівномірному кроці Δx одним із найбільш швидких є адресний алгоритм будування гістограми, що складається з таких етапів:

- для заданої вибірки x_i ($i=1..N$) знаходять мінімальне (MIN) і максимальне (MAX) значення;
- визначають крок ΔX для заданої кількості комірок розбиття M гістограми $\Delta X = (MAX - MIN)/M$;
- для кожного значення x_i ($i=1..N$) визначають його положення в гістограмі $H: K = (x_i - MIN)/\Delta X$ (K знаходиться в інтервалі $[0, M]$);
- виконують нагромадження гістограми $H_K = H_K + 1$ (масив гістограми H має бути попередньо обнуленим);
- оцінюють щільність розподілу:

$$\hat{f}_k = \frac{H_k}{N \Delta x}, \quad k = 0 \dots M.$$

Слід зазначити, що в більшості програмних засобів є вбудовані процедури будування гістограм. Наприклад, у програмі MathCAD для цих цілей призначено функцію *hist(•)*. Аналогічну підпрограму мовою *Python* розміщено в бібліотеці *matplotlib*. Приклад її застосування буде наведено нижче.

4. Визначення функції розподілу

$$F(x) = \int_{-\infty}^x f(z) dz$$

за результатами оцінювання щільності розподілу f_K як

$$\hat{F}_k = \sum_{i=0}^k \hat{f}_i, \quad k = 0 \dots M.$$

5. *Оцінювання форми закону розподілу.* Форму густини ймовірності характеризують нормовані центральні моменти 3-го й 4-го порядків – коефіцієнти асиметрії та ексцесу:

$$\hat{\alpha} = \frac{\sum_{i=1}^N (x_i - \bar{x})^3}{N \hat{\sigma}^3}, \quad \hat{\varepsilon} = \frac{\sum_{i=1}^N (x_i - \bar{x})^4}{N \hat{\sigma}^4} - 3. \quad (5.1)$$

Коефіцієнт асиметрії характеризує симетричність закону розподілу. Наприклад, для нормального закону $\alpha = 0$, для експоненціального – $\alpha = 2$. Коефіцієнт ексцесу визначає ступінь «гостровершинності» закону розподілу. За еталон узято нормальний закон: $\varepsilon = 0$. Для рівномірного закону $\varepsilon = -1,2$, трикутного (Сімпсона) – $\varepsilon = -0,6$, експоненціального – $\varepsilon = 6$.

Якщо теоретичні значення асиметрії та ексцесу для різних законів розподілу нанести на площину в системі координат $\{\alpha, \varepsilon\}$, як показано на рис. 5.1, то отримаємо діаграму, що має назву площини моментів.

6. *Вибір статистичної моделі* (підбір закону розподілу за експериментальними даними). Якщо оцінки коефіцієнтів асиметрії $\hat{\alpha}$ і ексцесу $\hat{\varepsilon}$, отримані згідно з (5.1), нанести точками на площину моментів (див. рис. 5.3), то як статистичну модель досліджуваної випадкової величини або випадкового процесу можна взяти найближчий з відомих законів розподілу. Якщо на площині моментів поруч з точкою, що відповідає експериментальним даним, виявилось кілька відомих законів розподілу, то слід перевірити всі варіанти статистичної моделі на адекватність.

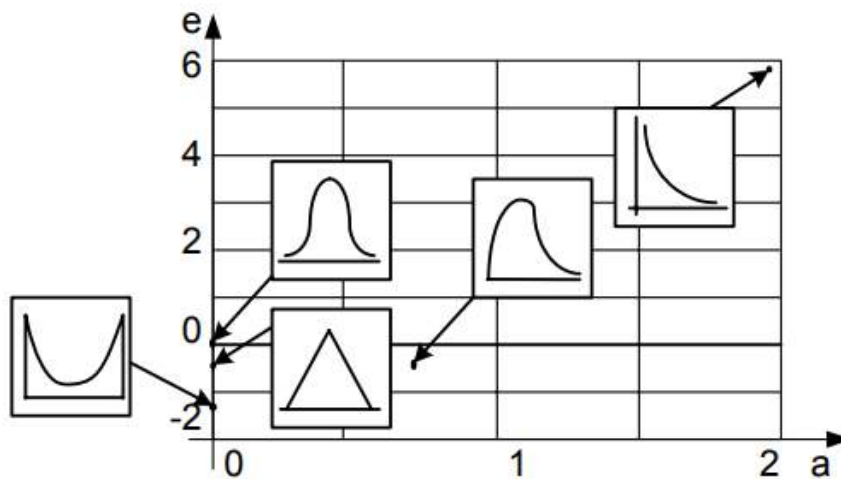


Рис. 5.1. Фрагмент площини моментів з нанесеними на неї положеннями типових законів розподілу

7. Перевірка узгодженості теоретичного й експериментального розподілів можна здійснити за критерієм Пірсона (критерієм χ^2), для якого необхідно обчислити гістограму $H(x)$ експериментальної вибірки $\{x_i\}$ і теоретичні значення щільності для передбачуваного закону розподілу $f(x)$, після чого знайти теоретичні значення ймовірностей потрапляння в інтервали гістограми:

$$P_k = \int_{x_k}^{x_{k+1}} f(z) dz, \quad k=1..M,$$

де M – кількість інтервалів у гістограмі. Відхилення χ^2 можна визначити за формулою

$$\chi^2 = \sum_{k=1}^M \frac{(NP_k - H_k)^2}{NP_k}, \quad (5.2)$$

де N – кількість відліків у вибірці $\{x_i\}$.

Висновок про непротиворіччя емпіричного розподілу (гістограми $H(x)$) заданому теоретичному розподілу $f(x)$ роблять на основі порівняння розрахованого значення χ^2 з квантилем цього розподілу $\chi_{табл}^2(\tau, P)$, значення якого наведено в табл. 5.1. Квантиль розподілу визначається заданою ймовірністю P прийняття правильного рішення (про непротиворіччя розподілів) і кількістю ступенів свободи τ , яка дорівнює кількості розрядів гістограми за відрахунком кількості незалежних умов, накладених на експериментальні ймовірності $f(x)$:

1) $\sum f_k = 1$ (ця умова накладається завжди);

2) $\sum x_k f_k = m_x$ (якщо ставлять вимогу відповідності середнього значення);

3) $\sum x_k^2 f_k = \sigma_x^2$ (якщо ставлять вимогу відповідності дисперсії).

Якщо $\chi^2 < \chi_{табл}^2$, то теоретичний та емпіричний розподіли з імовірністю P не суперечать один одному.

Таблиця 5.1

Значення квантилей χ^2 розподілу

Кількість ступенів свободи τ	Значення χ^2 для ймовірності P							
	0,90	0,80	0,70	0,50	0,30	0,20	0,10	0,05
1	0,016	0,064	0,148	0,455	1,074	1,642	2,71	3,84
2	0,211	0,446	0,713	1,386	2,41	3,22	4,60	5,99
3	0,584	1,005	1,424	2,37	3,66	4,64	6,25	7,82
4	1,064	1,649	2,20	3,36	4,88	5,99	7,78	9,49
5	1,610	2,34	3,00	4,35	6,06	7,29	9,24	11,07
6	2,20	3,07	3,83	5,35	7,23	8,56	10,64	12,59
7	2,83	3,82	4,67	6,35	8,38	9,80	12,02	14,07
8	3,49	4,59	5,53	7,34	9,52	11,03	13,36	15,51
9	4,17	5,38	6,39	8,34	10,66	12,24	14,68	16,92
10	4,86	6,18	7,27	9,34	11,78	13,44	15,99	18,31
11	5,58	6,99	8,15	10,34	12,90	14,63	17,28	19,68
12	6,30	7,81	9,03	11,34	14,01	15,81	18,55	21,0
13	7,04	8,63	9,93	12,34	15,12	16,98	19,81	22,4
14	7,79	9,47	10,82	13,34	16,22	18,15	21,1	23,7
15	8,55	10,31	11,72	14,34	17,32	19,31	22,3	25,0

5.2. Приклад оброблення даних імітаційного експерименту

Як дані імітаційного експерименту в прикладі використаємо дані процедури імітації випадкових чисел з нормальним законом розподілу з бібліотеки **numpy**. Обсяг вибірки становить 100 чисел. Вихідні дані показано на рис. 5.2.

Для обчислення мінімального й максимального значень отриманих даних можна використати вбудовані функції *Python*: *min(•)*, *max(•)*.

Математичне сподівання отриманої вибірки обчислюється функцією *mean(•)*, середньоквадратичне відхилення – функцією *std(•)*.

Далі для заданої кількості інтервалів *Nint* обчислюється ширина інтервалу гістограми *dx*. Побудова гістограми та її графічне відображення реалізується процедурою *hist* з бібліотеки **matplotlib.pyplot**. Результат будування гістограми показано на рис. 5.3.

Слід зазначити, що процедура *hist* повертає структуру даних, яка містить список власне гістограми (*h2[0]* у прикладі) і список меж інтервалів гістограми (*h2[1]* у прикладі). Для подальшої роботи ці списки бажано розділити й зберегти в окремих масивах.

Для оцінювання якості ступеня відповідності даних нормальному закону розподілу за критерієм Пірсона необхідно обчислити «теоретичну» гістограму – теоретичні ймовірності потрапляння випадкових чисел в інтервали, для чого необхідно для кожного інтервалу обчислити значення теоретичної щільності розподілу, помноженої на ширину інтервалу й кількість відліків, що використовувалися в експерименті. За цими даними обчислюється значення критерію Пірсона (5.2).

Повний текст програми оброблення даних показано на рис. 5.4.

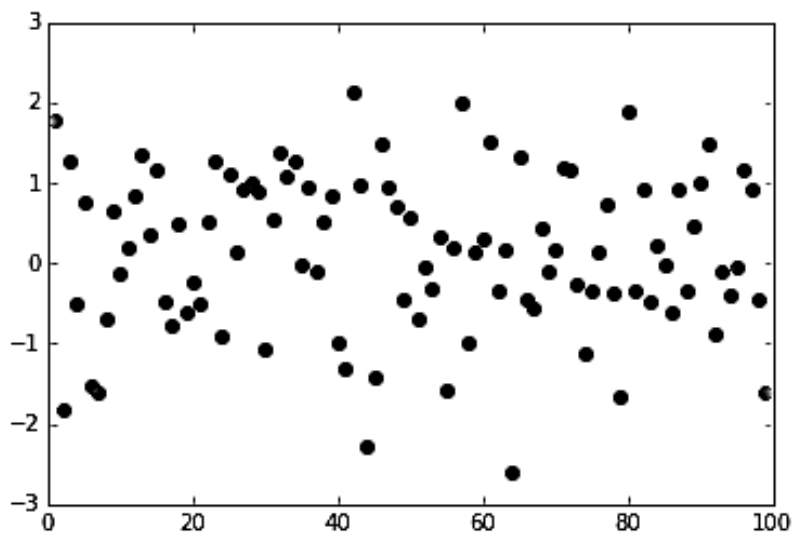


Рис. 5.2. Вихідні дані імітаційного експерименту

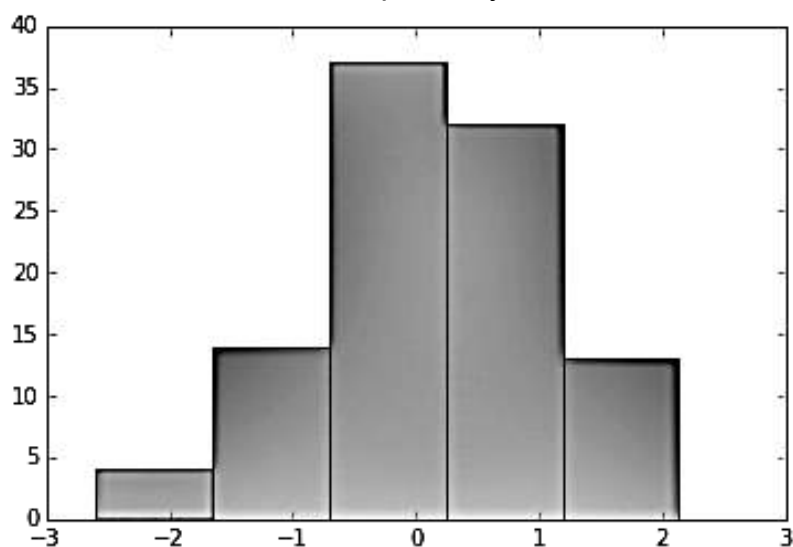


Рис. 5.3. Результат будування гістограми

```

import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
import numpy.random as nr
N=100; I=np.zeros(N); x=np.array(N)
x=nr.standard_normal(N)
for i in range(N):
    I[i]=i
fig1=plt.figure(); plt.plot(I[1:100],x[1:100], 'o')
# Оброблення
mx=np.mean(x); sigm=np.std(x)
minx=min(x); maxx=max(x)
print("min=", minx, " max=", maxx)
print("mean=", mx, " sko=", sigm)
Nint=5; dx=(maxx-minx)/Nint
fig2=plt.figure()
h2=plt.hist(x,Nint)
h=np.array(h2[0]); gi=np.array(h2[1])
t=np.empty(Nint); d=np.empty(Nint)
for i in range(Nint):
    t[i]=N*dx*np.exp(-0.5*((gi[i]-mx)**2)/sigm**2)
    )/np.sqrt(2*3.1415*sigm**2)
    d[i]=((t[i]-h[i])**2)/t[i]
x2=sum(d); print(x2)

```

Рис. 5.4. Програма оброблення даних імітаційного експерименту

Контрольні запитання

1. Чим вибіркове середнє значення відрізняється від математичного сподівання?
2. Що таке гістограма?
3. Як за значеннями гістограм побудувати графік щільності розподілу?
4. Як вибрати статистичну модель для даних імітаційного експерименту?
5. Як обчислити теоретичну ймовірність потрапляння випадкового числа в заданий інтервал?
6. Як визначити кількість ступенів свободи для застосування критерію Пірсона?
7. Як перевірити адекватність вибраної статистичної моделі?

БІБЛІОГРАФІЧНИЙ СПИСОК

Інструментальні засоби Python для моделювання та системного аналізу часових рядів при вирішенні задач кіберзахисту інформаційно-комунікаційних систем : навч. посіб. / Д. М. Шарадкін, І. Ю. Субач, А. В. Микитюк. – Київ : КПІ ім. Ігоря Сікорського, 2023. – 139 с.

Комп'ютерне моделювання систем та процесів. Методи обчислень : навч. посіб. Ч. 1 / Р. Н. Кветний, І. В. Богач, О. Р. Бойко, О. Ю. Софіна, О. М. Шушура. – Вінниця : ВНТУ, 2012. – 193 с.

Копей, В. Б. Мова програмування Python для інженерів і науковців : навч. посіб. / В. Б. Копей. – Івано-Франківськ : ІФНТУНГ, 2019. – 274 с.

Наконечний, А. Й. Цифрова обробка сигналів : навч. посіб. / А. Й. Наконечний, Р. А. Наконечний, В. А. Павлиш. – Львів : Вид-во Львів. політехніки, 2010. – 368 с.

Томашевський, В. М. Моделювання систем / В. М. Томашевський. – Київ : Вид. група ВНУ, 2005. – 352 с.

Haslwanter, T. An Introduction to Statistics with Python / T. Haslwanter. – Switzerland : Springer International Publishing AG, 2016. – 285 p.

Johansson R. Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib / R. Johansson. – New York, USA : Apress Media LLC, 2018. – 700 p.

Python 3.12 documentation [Електронний ресурс]. – Режим доступу : <https://docs.python.org/3/>

NumPy Documentation [Електронний ресурс]. – Режим доступу : <https://numpy.org/doc/>

Matplotlib: Visualization with Python [Електронний ресурс]. – Режим доступу: <https://matplotlib.org/stable/index.html>

ЗМІСТ

Вступ	3
1. Методологія моделювання радіоелектронних апаратів і систем	4
1.1. Класифікація систем	4
1.2. Розрахункові моделі й комп'ютерне моделювання	6
1.3. Програмні засоби комп'ютерного моделювання	10
1.4. Комп'ютерне моделювання як метод дослідження об'єктів	14
Контрольні запитання	17
2. Комп'ютерне моделювання сигналів і процесів	18
2.1. Математичні моделі детермінованих процесів	18
2.2. Дискретизація детермінованих процесів	24
2.3. Комп'ютерне моделювання детермінованих процесів	25
2.4. Приклад моделювання гармонічного процесу	28
2.5. Приклад моделювання й спектрального аналізу полігармонічного процесу	29
2.6. Приклад моделювання періодичної послідовності імпульсів	32
Контрольні запитання	35
3. Комп'ютерне моделювання лінійних систем	36
3.1. Математичні моделі лінійних систем	36
3.2. Приклад моделювання проходження періодичної послідовності імпульсів крізь коливальний контур	39
Контрольні запитання	43
4. Комп'ютерне моделювання випадкових величин, дій і чинників	44
4.1. Імовірнісні моделі дискретних випадкових величин	44
4.1.1. Біноміальний закон розподілу	45
4.1.2. Закон Пуассона	46
4.2. Імовірнісні моделі безперервних випадкових величин	46
4.2.1. Рівномірний закон розподілу	48
4.2.2. Експоненційний розподіл	49
4.2.3. Закон розподілу Сімпсона	49
4.2.4. Закон розподілу арксинуса	50
4.2.5. Закон розподілу Релея	50
4.2.6. Нормальний закон розподілу	51
4.3. Моделювання випадкових сигналів і впливів	51
4.3.1. Імітація значень рівномірно розподіленої величини	52
4.3.2. Моделювання випадкових подій	54
4.3.3. Моделювання неперервних випадкових величин	56
Контрольні запитання	63
5. Оброблення даних комп'ютерного моделювання випадкових величин	64
5.1. Методика оброблення даних імітаційного експерименту	64
5.2. Приклад оброблення даних імітаційного експерименту	67
Контрольні запитання	69
Бібліографічний список	70

Навчальне видання

**Попов Анатолій Владиславович
Каданер Олександр Олександрович
Колесніков Денис Вікторович
Інкарбаєва Ольга Сергіївна**

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ТА ОБРОБЛЕННЯ ДАНИХ

Редактор О. Ф. Серьожкіна

Зв. план, 2024

Підписано до друку 21.11.2024

Формат 60x84 1/16. Папір офс. Офс. друк

Ум. друк. арк. 4. Обл.-вид. арк. 4,5. Наклад 100 пр.

Замовлення 137-24. Ціна вільна

Видавець і виготовлювач

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

61070, Харків-70, вул. Чкалова, 17

<https://khai.edu>

Видавничий центр «ХАІ»

61070, Харків-70, вул. Чкалова, 17

izdat@khai.edu

Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру видавців, виготовлювачів і розповсюджувачів
видавничої продукції сер. ДК № 391 від 30.03.2001