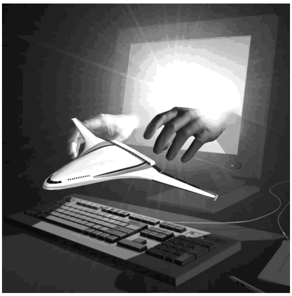
М.К. Філяшкін, В.В. Калініченко,
Ю.М. Кеменяш, М.Ф. Тупіцин

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МОДЕЛЮВАННЯ СИСТЕМ ЦИВІЛЬНОЇ АВІАЦІЇ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

М. К. ФІЛЯШКІН
В. В. КАЛІНІЧЕНКО
Ю.М. КЕМЕНЯШ
М. Ф. ТУПІЦИН

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МОДЕЛЮВАННЯ СИСТЕМ ЦИВІЛЬНОЇ АВІАЦІЇ



Рекомендовано
Вченою радою Національного авіаційного університету
як навчальний посібник для студентів напряму підготовки
«Автоматизація та комп'ютерно-інтегровані технології»
вищих навчальних закладів

Київ 2017

УДК 629. 7:004. 4(075 .8)
ББК О571. 5 -5я7
П 784 050с5/я7

Рецензенти:
О.А.Машков - д-р. техн. наук, проф., проректор з наукової роботи.
(Державна екологічна академія післядипломної освіти).

К.Л. Шевченко - д-р. техн. наук, проф.
(Національний технічний університет України «КПІ»).

П.О. Яковенко - лауреат Державної премії України в галузі науки і техніки, головний конструктор-керівник проектного відділу.
(Державне Київське конструкторське бюро «Луч»).

Рекомендовано Вченою радою Національного авіаційного університету
(*протокол №8/16 від 17.11.16*)

Філяшкін М.К., Калініченко В.В., Кеменяш Ю.М., Тупіцин М.Ф.
П 784 Програмне забезпечення моделювання систем цивільної авіації: Навчальний посібник – К.: «Принт-центр», 2017. – 256 с.
ISBN 978-966-97410-4-2

У навчальному посібнику розглянуто основні поняття, визначення, положення та принципи математичного моделювання, надано класифікацію математичних моделей і методів моделювання. Описані основні етапи математичного моделювання.

Для студентів вищих навчальних закладів, може бути корисним науково-технічним працівникам при розробці алгоритмічного та програмного забезпечення пілотажно-навігаційних комплексів, а також фахівцям у галузі розробки та досліджень складних систем цивільної авіації.

УДК 629.735.054.07(075.8)
ББК О571.5-5я7

ISBN 978-966-97410-4-2

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

АСНД – автоматизована система наукових досліджень
АСУ – автоматизована система управління
БДА – багатофакторний дисперсійний аналіз
БД – база даних
БЗ – база знань
ВП – вирішувач проблем
ВЧ – випадкові числа
ГВЧ – генератор випадкових чисел
ЕОМ – електрона обчислювальна машина
ЗВ – зовнішній вплив
ЗДУ – звичайне диференціальне рівняння
ІД – індукційний датчик
ІМ – імітаційна модель
ІНС – інерціальна навігаційна система
ЛП – логічний перетворювач
МНК – метод найменших квадратів
ОА – обслуговуючий апарат
ПВВ – проста випадкова вибірка
ПВЧ – псевдовипадкові числа
ПНК – пілотажно-навігаційний комплекс
ПФЕ – повний факторний експеримент
САУ – система автоматичного управління
СДУ – стохастичні диференціальні рівняння
СЗ – список заявок
СЛАР – система лінійних алгебраїчних рівнянь
СМО – система масового обслуговування
СМП – список майбутніх подій
СПП – список поточних подій
СПС – система повітряних сигналів
ЦА – цивільна авіація
ЦОМ – цифрова обчислювальна машина
ЧФЕ – частковий факторний експеримент

ВСТУП

Моделювання є невід'ємною частиною досліджень об'єктів, процесів або явищ, тобто систем. Під моделюванням розуміють дослідження систем на їх моделях з метою отримання пояснень, які цікавлять дослідника.

Удосконалення обчислювальної техніки відкрило перед моделюванням величезні перспективи для дослідження процесів і явищ навколишнього середовища. Комп'ютер як інструмент інформаційних технологій відіграє роль підсилювача технології моделювання. Комп'ютерні моделі прискорюють процеси дослідження, роблять їх більш точним.

В авіації моделювання є ефективним способом випробування складних авіаційних систем, що доповнює льотні випробування. Воно полягає у відтворенні засобами обчислювальної техніки процесів, ідентичних тим, що мають місце в реальному польоті. У порівнянні з льотними випробуваннями перевага моделювання зумовлено тим, що воно вільно від ряду обмежень і труднощів, властивих льотному експерименту, таких, наприклад, як необхідність дотримання умов безпеки польотів, проблематичність повторного відтворення ситуації, а також меншими матеріальними і часовими витратами, які припадають на одиничний експеримент.

При проведенні такого моделювання широко використовуються моделі динаміки польоту літальних апаратів (ЛА), а також моделі основних функціональними підсистемами ЛА: планера, системи управління, пілотажно-навігаційного обладнання, силової установки, тощо.

Іншими прикладами застосування математичного моделювання в цивільній авіації можуть служити дослідження і оптимізація технологічних процесів, наприклад, в авіаремонтних майстернях, процесів підготовки до рейсів повітряних суден в аеропорту, процесів обслуговування пасажиропотоків, процесів управління повітряним рухом.

При проведенні таких досліджень, як правило, використовують різного роду імітаційні моделі систем з дискретними подіями (мережеві моделі, моделі систем масового обслуговування і т.п.).

Отже, при дослідженні будь-якої системи основним етапом є створення її математичної моделі. При побудові математичної моделі, досліджуваного об'єкта чи явища виділяють ті його особливості, риси і деталі, які з одного боку у повній мірі відображають оригінал з точки зору заданої цілі дослідження, а з іншого допускають його математичну формалізацію. Від якості моделі залежить весь подальший аналіз об'єкта. Модель повинна бути достатньо точною, адекватною і повинна бути зручна для використання.

Зазвичай над створенням моделі працюють фахівці з різних областей, тому що в моделюванні досить велика роль між предметних зв'язків. Природно, що одним з фахівців повинен бути програміст. Найкращим варіантом є варіант коли фахівець з певної предметної галузі одночасно володіє навичками програмування.

При викладенні матеріалу посібника вважається, що математичне формулювання задачі вже існує, потрібно тільки вміти її розв'язувати на комп'ютері з використанням обраних чисельних методів, складаючи для цього алгоритми та програми розв’язку задачі.

У ході розв'язання конкретної практичної задачі спеціаліст повинен, перш за все, визначити, до якого типу математичної задачі належить ця практична задача, вибрати чисельний метод для її розв'язання та розробити його програмну реалізацію або вміти застосовувати для розв'язання задачі один із відомих комп'ютерних математичних пакетів прикладних програм та інтегрованих середовищ. Пріоритетнім е здатність формулювати моделюючий алгоритм на одному з вхідних машинних мов. Як вхідні мови для вирішення завдань моделювання можуть бути з успіхом використані універсальні алгоритмічні мови високого рівня.

Розробка програмної реалізації моделюючого алгоритму включає в себе складання деталізованої схеми алгоритму та складання комп'ютерної програми його реалізації. Програмна реалізація передбачає вибір мови програмування і інструментів розробки програмного забезпечення, яки підтримують цю мову. Причому будь-який алгоритм допускає певну кількість його програмних реалізацій.

У навчальному посібнику розглядаються саме питання розробки та програмної реалізації моделюючих алгоритмів з дослідження складних систем цивільної авіації. Порушуються також питання застосовування для розв'язання цих задач комп'ютерних математичних пакетів прикладних програм.

Глава 1. СИСТЕМИ ТА ЇХ ДОСЛІДЖЕННЯ

1.1. Система як предмет дослідження

При математичному моделюванні системи необхідно визначитися з базовим поняттям самої системи. Система в широкому змісті – еквівалент поняття математичної моделі й задається парою множин **X**, **Y** (**X** – множина входів, **Y** – множина виходів) і відношеннями, що формалізують зв'язок (залежність) між входами й виходами. Таким чином, у поняття «система» вкладається й поняття процеси, тобто термін *модель системи* варто сприймати як *модель об'єкта, явища або процесу*. ***Система*** – це задана сукупність елементів і відносин (предикатів), що діють як єдине ціле при досягненні заданої цілі.

Елемент – це частина системи не підлягаюча подальшому розчленовуванню на даному рівні дослідження. Окремі сукупності елементів і відносин між ними можуть утворювати ***підсистеми***, які формують систему, але окремо не мають властивості самої системи.

Система як сукупність елементів характеризується ***зв'язками*** - обмінами між елементами системи. Зв'язки дозволяють за допомогою переходів по них від елемента до елемента з'єднати два будь-яких елементи сукупності. Найпростішими типами зв'язків є послідовне, паралельне з'єднання й зворотний зв'язок.

Використовуючи поняття структури системи, можна представити систему як групу елементів із зазначенням зв'язків між ними, що дає уявлення про систему в цілому. Такий поділ системи на елементи може мати функціональну, алгоритмічну або іншу основу. Приклад функціональної структури - поділ електричного виконавчого пристрою системи автоматичного управління на двигун, на редуктор, на вузол кінцевих вимикачів, на вузол пристроїв зворотного зв'язку, на муфту включення. Приклад алгоритмічної структури - інструкція (технологічна карта) пошуку несправності технічного пристрою.

Структуру системи зображують у вигляді графічної схеми, що складається з підсистем або елементів і ліній зв'язку, які їх з'єднують. Такі схеми називаються структурними.

З'єднання систем і підсистем є системою й може бути формалізовано. Наприклад, послідовне з'єднання систем $\mathbf{S_1}$ і $\mathbf{S_2}$, зі входами $\mathbf{X_1}$, $\mathbf{X_2}$ і виходами $\mathbf{Y_1}$, $\mathbf{Y_2}$ – ($\mathbf{S_1} \subset \mathbf{X_1} \times \mathbf{Y_1}$, $\mathbf{S}_2 \subset \mathbf{X}_2 \times \mathbf{Y}_2$) є відношення $\mathbf{S} \subset \mathbf{X}_1 \times \mathbf{Y}_2$ таке, що $(x_1, y_2) \in \mathbf{S}$, якщо існують $y_1 \in \mathbf{Y_1}$, $x_2 \in \mathbf{X}_2$, які задовольняють умовам $(x_1, y_1) \in \mathbf{S_1}$, $(y_1, x_2) \in \mathbf{R}$, $(x_2, y_2) \in \mathbf{S_2}$, де $\mathbf{R} \subset \mathbf{Y}_1 \times \mathbf{X}_2$ – відношення, що визначає зв'язок між y_1 і x_2. Таким чином, можна визначати які завгодно складні системи, виходячи із простих.

Управління системою здійснюється для досягнення бажаної цілі. Поняття ***цілі системи*** визначають як задачу досягнення бажаного стану виходів системи, визначеного ззовні (зовнішнім середовищем) або встановленого самою системою Саме ціль обумовлює спосіб і форму опису системи, тобто опис системи повинний бути цілеспрямованим.

Система може мати одну або кілька цілей. Із множини цілей системи повинна бути виділена головна ціль. Ціль повинна бути чисельною й мати розмірність. Для конкретизації цілі необхідно задати *критерії досягнення цілі* і *обмеження*, при яких розв‘язується ця задача

Критерій досягнення цілі це міра близькості до цілі. Критерій досягнення цілей може формулюватися як у якісній, так і в кількісній формі.

Обмеження це умови, що відображають вплив зовнішніх і внутрішніх факторів, які потрібно враховувати в задачі прийняття рішень. Обмеження, як правило, доповнюють (конкретизують) сформульовані цілі, а у ряді випадків можуть зробити цілі нереалізованими.

Різноманіття систем за складністю досить велике, й істотну допомогу при їхньому вивченні могла б надати їхня класифікація. Однак повної класифікації систем, зокрема складних систем на даний час не існує, більш того, не вироблені остаточно її принципи. Різні вчені пропонують різні трактування складної системи.

За класифікацією Г. Поварова[1.1] залежно від числа елементів, що входять у систему, розрізняють чотири класи складності систем: малі ($10 \ldots 10^3$ елементів); складні ($10^3 \ldots 10^7$ елементів); ультраскладні ($10^7 \ldots 10^{30}$ елементів); суперсистеми ($10^{30} \ldots 10^{200}$ елементів).

[1.1] Поваров Геллий Миколайович – радянський математик, професор кафедри кібернетики Національного дослідницького ядерного університету «МІФІ»

За трактуванням А. Берга[1.2] складна система описується, принаймні, на двох різних мовах, наприклад теорії диференціальних рівнянь і алгебри логіки.

Ю. Черняк[1.3] складної називає систему, яка будується для розв'язання багатоцільової, багатоаспектної задачі й відображає об'єкт із різних сторін у декількох моделях.

С. Вир[1.4] пропонує ділити системи на прості, складні й дуже складні. Прості - це найменш складні системи. Складні - це системи, що відрізняються розгалуженою структурою й більшою розмаїтістю, внутрішніх зв'язків. Дуже складна система - це складна система, яку докладно описати не можна. При цьому чіткої межі, що відокремлює прості системи від складних немає. Пізніше С. Вир запропонував відносити до простих систем ті, які мають до 10^3 станів, до складних - від 10^3 до 10^6 станів і до дуже складних - системи, що мають понад мільйон станів.

Розглянемо один з найбільш розповсюджених варіантів класифікації систем за складністю. Відповідно до цієї класифікації множина систем підрозділяється на *прості* системи*, складні* й *великі*.

Проста система – це система, модель якої досить просто описати з метою реалізації управління цією системою. Якщо ж результат управління, отриманий за допомогою моделі, буде несподіваним, то таку систему відносять до складної системи. Таким чином, ознакою простоти системи є достатність інформації для її управління.

На відміну від простих систем складні системи характеризуються численними й різноманітними зв'язками між окремими елементами системи й наявністю у системи функції призначення, яких немає у складових її частин.

Складна система характеризується:

– наявністю великої кількості різнорідних елементів і підсистем;

[1.2] Берг Аксель Іванович – радянський вчений у галузі радіотехніки та кібернетики, основоположник школи біологічної кібернетики та біотехнічних систем і технологій
[1.3] ЧЕРНЯК Юрій Ілліч – доктор економічних наук, професор, радянський вчений у галузі теорії систем і системного аналізу.
[1.4] Бір Ентоні Стаффорд – британський вчений у галузі кібернетики та оперативних досліджень, професор Манчестерського університету.

– складністю й неоднорідністю зв'язків між підсистемами;
– наявністю невизначеностей в описі системи;
– недостатністю інформації для ефективного управління цією системою.

Для переведення складної системи в розряд простої необхідне одержання відсутньої інформації про неї й включення її в модель.

Великі системи складаються з ряду складних систем, мають ієрархічну структуру з розвиненою процедурою прийняття рішень на різних рівнях.

Як ознаки великої системи пропонується використовувати різні поняття:

– поняття ієрархічної структури;
– наявність великих потоків інформації;
– велика кількість алгоритмів переробки інформації.

У кібернетиці великою системою прийнято вважати систему, для актуалізації моделі якої з метою управління бракує матеріальних ресурсів (машинного часу, ємності пам'яті, інших матеріальних засобів моделювання), тому способом переведення великих систем у прості або складні є створення нових більш сучасних засобів обчислювальної техніки.

Як видно з визначень, поняття великої й складної системи є різними. Однак у літературі ці поняття визначені не однозначно. Деякі автори взагалі не використовують цих понять, інші використовують їх як синоніми, а деякі вважають різницю між ними чисто кількісною.

Основні атрибути (властивості), які визначають систему, це цілісність, структурованість і цілеспрямованість.

Цілісність (єдність) означає, що система відділена від зовнішнього середовища; середовище може здійснювати на систему вплив (акцію) через входи й сприймати відгук (реакцію) на ці дії через виходи.

Структурованість визначає наявність установлених зв'язків і відносин між елементами усередині системи, підпорядкованість організації всієї системи певної цілі.

Цілеспрямованість – вимагає завдання деякої цілі - бажаного майбутнього стану системи, який може бути досягнутий в результаті діяльності, здійснюваної за певними правилами. Досягнення цілі свідчить про правильну роботу системи.

Наведені вище формальні визначення досить загальні; під них підпадають практично всі види систем, а також їх математичних моделей.

Функціонування системи - це процес, що поширюється у часі, тобто множини можливих входів і виходів **X, Y** – це множини функцій часу зі значеннями відповідно у множинах X, Y:

$$\mathbf{X} = \{ X : T \rightarrow X \},\ \mathbf{Y} = \{ Y : T \rightarrow \mathrm{Y} \},$$

де T – множина моментів часу, на якій розглядається система.

Система називається *визначеною*, якщо кожної вхідної функції $x(t)$ відповідає єдина вихідна функція $y(t)$. У протилежному випадку система називається *невизначеною.* Невизначеність зазвичай виникає через неповноту інформації про зовнішні умови роботи системи.

Числові величини, що пов'язані з системою характеризуються: *вхідними* (екзогенними) змінними $x(t)$, які впливають на систему ззовні або являють собою результат дії зовнішніх причин, і *вихідними* (ендогенними) змінними $y(t)$, що виникають у системі в результаті впливу внутрішніх причин. Коли вихідні (ендогенні) змінні характеризують стан або умови, то вони називаються **змінними стану** – $z(t)$. Поведінка системи залежить також від **параметрів** – числових величин, які можна задавати незалежно від впливу зовнішнього середовища і, які вважаються сталими на проміжку часу розглядання системи.

Значення змінних і параметрів визначають кількісну інформацію про систему. Інформація, що залишилася, тобто якісна інформація, визначає структуру системи. Різниця між змінними і параметрами, а також між параметрами і структурою може бути умовною, однак вона корисна в методичному відношенні. Так, типовим прийомом побудови математичних моделей системи, у тому числі з метою моделювання, є її параметризація - вибір в якості математичної моделі сімейства функцій, які залежать від кінцевої (зазвичай невеликої) кількості чисел – параметрів.

Всілякі аспекти дослідження систем здійснюються з використанням методології "*теорії систем*", складовою якої є системний аналіз.

1.2. Основи системного аналізу

Системний аналіз проводитися з метою прийняття тих чи інших рішень на етапі дослідження, в тому числі і моделювання, складних систем. Відомі різні способи прийняття рішень.

Наприклад, рішення бувають інтуїтивні. В цьому випадку рішення підказуються підсвідомістю. Такі рішення, як правило, приймаються миттєво і в цьому полягає їх ефективність.

Стандартні рішення прийматися на рівні здорового глузду, який представляє собою досвід людини та оточуючих його людей, отриманий в конкретних умовах.

Задачі, що не мають інтуїтивних і стандартних рішень на рівні здорового глузду, і які, в принципі, не можуть бути формалізовані, вирішуються з використанням системного аналізу, який є одним з напрямків *системного підходу*.

Системний підхід це комплексне вивчення складної *проблеми* як єдиного цілого (як системи), без концентрації уваги на окремих її складових. Такій підхід ґрунтується на визнанні того що навіть при оптимальних характеристиках підсистем поведінка системи в цілому може бути лише субоптимальною через взаємодії між окремими її частинами.

Проблема – це різниця між існуючою і бажаної системами, якщо цієї різниці немає, то немає і проблеми. Вирішити проблему – значить скорегувати стару систему або сконструювати нову, бажану. У широкому сенсі проблема це складне теоретичне або практичне питання, яке вимагає вивчення і розв'язання. Важливою передумовою успішного розв'язання проблеми служить її правильне формулювання.

Тому на ***першому етапі*** системного аналізу аналізується сутність проблеми, що виникла, и здійснюється її формулювання. Формулювання проблеми здійснюється на словесному (вербальному) рівні і, як правило, є досить розпливчастим. Усвідомлення проблема проходить декілька «стадій»: від неясного відчуття незадоволеності системою, до усвідомлення потреби, потім до виявлення проблеми й, нарешті, до її формулювання.

На цьому етапі також обирається метод розв'язання проблеми, оскільки, з одного боку, можна взятися за розв'язання проблеми, яка не піддається системному аналізу, а з іншого боку - вибрати проблему, яку можна більш економно вирішити, не використовуючи всю міць методів системного аналізу. На відміну від задачі проблема не має однозначного вирішення, саме невизначеністю проблема відрізняється від задачі.

Як тільки існування проблеми усвідомлене, потрібно спростити задачу настільки, щоб вона мала рішення (по можливості аналітичне), яке піддається якісному аналізу й має наочну інтерпретацію. Тому ***другий етап*** системного аналізу передбачає постановку задачі шляхом уточнення складної проблеми і її структуризацію (деталізацію) в серію завдань, що вирішуються з використанням принципу *системності*. Професор кібернетики Іллінойського університету Ешбі сформулював поняття системності в такий спосіб: «Будь-яка наука системна. Системність - це науковий спосіб спрощувати». Тому поняття ***системного аналізу*** можна сформулювати як сукупність методичних прийомів дослідження можливостей досягнення системою конкретної *цілі* шляхом виявлення та *спрощення* складної проблеми (розподіл її на ряд завдань) і з послідуючим розв'язанням цих завдань. При цьому важливо не випустити з розгляду найбільш суттєві зв'язки системи, що впливають на досягнення цілі дослідження.

Звідси випливає основний принцип системного аналізу - ***принцип цілі*** (бажаного майбутнього стану системи, який може бути досягнутий в результаті певної діяльності). Принцип цілі стверджує, що будь-яка складна система повинна розглядатися тільки з точки зору досягнення поставленої цілі, тобто система є засіб досягнення цілі.

На практиці, як правило, існує кілька цілей і задач і тому важливо, окрім формулювання головної, глобальної цілі системи, не упустити деякі з суттєвих серед інших, для чого на ***третьому етапі*** системного аналізу здійснюється встановлення деревовидної ієрархії цілей та задач. При цьому здійснюється дроблення (декомпозиція) загальної цілі й основної задачі дослідження на ряд більше простих (другорядних). Установлення пріоритетності тих або інших задач в ієрархічному ланцюжку - одна із центральних проблем системного аналізу.

Етап формулювання цілей включає також визначення критеріїв їх досягнення. При формуванні критеріїв головним є не їх кількість, а те, наскільки повно вони характеризують ціль. Тому прагнуть досягти компромісу між повнотою описування цілей та кількістю критеріїв.

Структуризація складної проблеми та принцип цілі диктують вибір рівня узагальнення і спрощення, на якому будується визначення та опис системи. Тому на ***четвертому етапі*** здійснюється структуризація системи – етап системного аналізу, смисл якого полягає в тому, що вся сукупність об'єктів і процесів, що мають відношення до поставленої цілі, насамперед розділяється на власне досліджувану систему й зовнішнє середовище, а потім здійснюється послідовне цілеспрямоване спрощення (декомпозиція) самої системи. Система послідовно поділяється на взаємопов'язані підсистеми й елементи (виділяються окремі складові частини досліджуваної системи), а можливі зовнішні впливи представляються у вигляді сукупності елементарних впливів. На основі цієї інформації будується моделі системи та підсистем і визначаються їх параметри у числовому виразі.

Використовуючи отримані моделі, здійснюють етап моделювання складних динамічних взаємозв'язків між різними аспектами проблеми. Робота з моделлю (моделювання) і є предметом дослідження системи. Саме на цьому етапі здійснюється пошук шляхів розв'язання проблеми з перевіркою та підтвердженням цього дослідним шляхом. Слід зазначити, процесам що моделюються може бути властива внутрішня невизначеність, що значно ускладнює дослідження системи. У цьому випадку повертаються до етапу структуризації цілей та задач і відповідно до декомпозиції системи та її моделей. Далі для вирішення кожної підзадачі користуються тією ж методикою, що й для всієї системи

Якщо в ході рішення деякі з підзадач знов виявляються занадто складними, то проводиться подальша декомпозиція: виникають підзадачі наступного рівня і т. д. Результатом цього процесу є структуризація: вихідна система набуває ієрархічну (багаторівневу) структуру.

Ієрархічна структура може бути зображена у вигляді розгалуженої блок-схеми (рис. 1.1). Вона називається деревом цілей, оскільки являє собою граф типу дерева. Відповідна структура виникає і в безлічі підцелей. Вершина верхнього рівня графа називається коренем.

Деревоподібна структура найбільш проста для аналізу та реалізації. Крім того, в ній завжди зручно виділяти ієрархічні рівні - групи елементів, що знаходяться на однаковому видаленні від верхнього елементу.

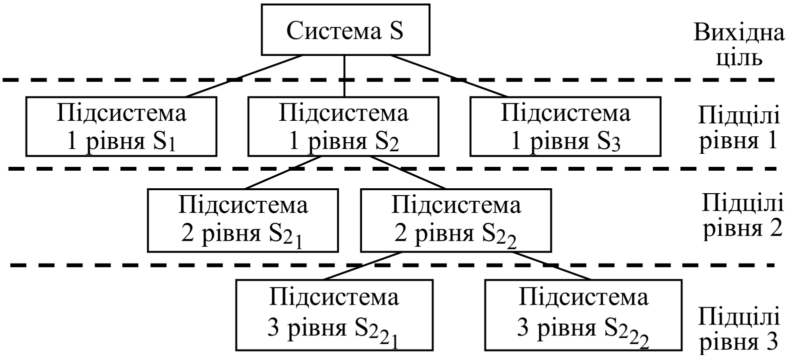


Рис. 1.1

Зазвичай операцію декомпозиції виконує експерт, і успіх декомпозиції визначається його досвідом та інтуїцією. Якщо доручити аналіз одного і того ж об'єкта різним експертам, то отримані деревовидні списки можуть різнитися.

Прагнучи при виконанні декомпозиції перейти від евристичного (інтуїтивного) підходу до більш усвідомленого (алгоритмічного), необхідно обґрунтувати, чому саме так, а не інакше ціле поділяється на частини. Обґрунтуванням можуть бути допущення, які приймають при отриманні спрощених моделей системи (підсистем), які використовуються при їх дослідженні шляхом моделювання. Тому основою декомпозиції є абстрактна, зазвичай математична, модель аналізованої системи, яка визначає структуру та властивості елементів системи, а також причинно-наслідкові зв'язки, що властиві системі й суттєві для досягнення мети моделювання.

Роздроблення моделі системи на моделі підсистем зазвичай проводиться за принципом «слабких» зв'язків, тобто так, щоб зв'язки між підсистемами були слабкіше, ніж зв'язки між елементами кожної підсистеми.

При виборі рівня опису системи доцільно керуватися наступним правилом: у модель системи повинні ввійти всі параметри, які забезпечують визначення характеристик, які цікавлять дослідника, системи на заданому часовому інтервалі її функціонування; інші параметри за можливістю варто виключити з моделі. Для тих компонентів, відносно яких відомо або передбачається, що вони більшою мірою впливають на точність результатів, ступінь детальності може бути вище інших

Однак розчленовування системи на частини порушує її цілісність, позбавляє систему емерджентних властивостей, тобто властивостей якими володіє тільки система в цілому, але не володіє, жоден її елемент окремо. Тому успіх системного аналізу складається не тільки й не стільки в розчленовуванні складного цілого на більш прості частини, а в тому, що, будучи з'єднаними належним чином, ці частини знову утворять єдине ціле. Це *агрегування* частин у ціле є ***кінцевим етапом*** аналізу, оскільки лише після цього можна пояснити ціле через його частини - у вигляді структури цілого.

У більшості випадків практичного застосування системного аналізу для дослідження властивостей і наступного оптимального управління системою можна виділити наступні основні етапи:

- вибір проблеми (аналіз проблеми і її формулювання);
- постановка задачі й обмеження ступеня її складності (структуризація задачі в серію завдань, з використанням принципу системності - принципу цілі);
- встановлення ієрархії цілей та задач і критеріїв їх досягнення;
- декомпозиція системи й побудова її моделей;
- моделювання (відшукання рішень по частинах за допомогою моделей, перевірка і підстроювання рішень під зовнішні умови);
- агрегування («склеювання» частин);
- вибір найкращого рішення;
- впровадження результатів.

Якщо на якомусь етапі виникають труднощі, то це означає, що задача виявилася занадто складною і знов її потрібно розбити на кілька простіших підзадач, тобто провести подальшу декомпозицію. Результатом цього процесу є структуризація: вихідна система набуває ієрархічну (багаторівневу) структуру.

Резюмуючи вище викладене, можна сформулювати основу математичного моделювання, як головного інструменту системного аналізу: «**не вирішуй складну задачу, не вирішивши спочатку більше просту!**». Проведення декомпозиції складної задачі дозволяє встановити якісні властивості її вирішення, виявити фактори, які найбільш сильно впливають на рішення складної задачі, і, головне, співвіднести результати математичного дослідження зі здоровим глуздом.

1.3. Декомпозиція та формалізація її принципів

Декомпозиція системи здійснюється виходячи з обраного рівня деталізації моделі, який, у свою чергу, обумовлюється цілями моделювання та вимогами до точності та вірогідності результатів моделювання.

На етапі декомпозиції здійснюються:

1. Визначення і структуризація складної проблеми в серію завдань;
2. Декомпозиція загальної цілі дослідження, шляхом побудови дерева цілей;
3. Виділення системи із середовища й опис факторів впливу;
4. Побудова моделі системи;
5. Декомпозиції системи та її моделі залежно від обраної стратегії декомпозиції.

Рівні деталізації іноді називають стратами, а процес виділення рівнів, стратифікацією. Деталізація системи та її моделі повинна проводитися до такого рівня, щоб для кожного елемента були відомі або могли бути отримані залежності його вихідних характеристик від вхідних впливів, істотних з точки зору обраного показника ефективності. Підвищення рівня деталізації опису системи дозволяє отримати більш точну її модель, але ускладнює процес моделювання і веде до зростання витрат часу на його проведення.

Глибина декомпозиції обмежується. Декомпозиція повинна припинятися, якщо необхідно змінити рівень абстракції – представити елемент як підсистему. Якщо при декомпозиції з'ясовується, що модель починає описувати внутрішній алгоритм функціонування елемента, то це означає вихід за межі цілі дослідження системи й тому необхідно припинити декомпозицію.

В автоматизованих методиках типової є декомпозиція моделі на глибину 5-6 рівнів. На таку глибину зазвичай здійснюється декомпозиція однієї з підсистем. Функції, які вимагають такого рівня деталізації, часто дуже важливі, і їхній детальний опис дає ключ до секретів роботи всієї системи.

Найбільш часто застосовують наступні стратегії декомпозиції:

– декомпозиція за життєвим циклом;
– декомпозиція за фізичним процесом;

– структурна декомпозиція;
– функціональна декомпозиція.

Декомпозиція за життєвим циклом. Ознака виділення підсистем – зміна закону функціонування підсистем на різних етапах циклу існування системи «від народження до загибелі». Рекомендується застосовувати цю стратегію, коли ціллю системи є оптимізація процесів і коли можна визначити послідовні стадії перетворення входів у виходи.

Декомпозиція за фізичним процесом. Ознака виділення підсистем – кроки виконання алгоритму функціонування підсистеми, стадії зміни станів. Хоча ця стратегія корисна при описі існуючих процесів, результатом її часто може стати занадто послідовний опис системи, який однак не буде повною мірою враховувати обмеження, які накладаються однією функцією на іншу. До того ж може виявитися схованою послідовність управління. Застосовувати цю стратегію доцільно, якщо ціллю моделі є тільки опис самого фізичного процесу.

Структурна декомпозиція (декомпозиція за підсистемами). Ознака виділення підсистем – сильний зв'язок між елементами за одним з типів відносин (зв'язків), які існують у системі (інформаційні, логічні, ієрархічні, енергетичні і т.п.). Силу зв'язку, наприклад, за інформацією можна оцінити коефіцієнтом інформаційного взаємозв'язку підсистем $k = N / N_0$, де N – кількість взаємно використовуваних інформаційних масивів у підсистемах, N_0 – загальна кількість інформаційних масивів. Для опису всієї системи повинна бути побудована складена модель, яка об'єднує всі окремі моделі. Рекомендується використовувати розкладання на підсистеми, тільки коли такий поділ на основні частини системи не змінюється. Нестабільність границь підсистем швидко знецінить як окремі моделі, так і їхнє об'єднання.

Найчастіше при дослідженні систем використовується стратегія **функціональної декомпозиції.** Функціональна декомпозиція базується на аналізі функцій системи. При цьому ставиться питання, що робить система, незалежно від того, як вона працює. Основою розбивки на функціональні підсистеми служить спільність функцій, виконуваних групами елементів.

Функціональна декомпозиція і опис її результатів на базі аналізу функції F окремих елементів модельованої системи класу S є основою формалізованого опису складних систем і побудови їх математичних моделей. Клас S включає кінцеву множину базових моделей $s_k \in S' \subset S, \quad k = \overline{1, N}$, модифікація яких дозволяє розглянути практично всі моделі, що цікаві для дослідження.

Кожна функція F може бути представлена у вигляді набору операцій $\Phi = \{\Phi_i\}_{i=1}$, виконуваних паралельно над інформацією. При цьому окремі підоперації Φ_i відповідають компонентам c_i системи, що вивчається. Послідовність Φ_i можна впорядкувати відповідно до характеру операцій (управління, генерування, обробка інформації). Процес розбивки можна продовжити з урахуванням розбиття системи на окремі компоненти, тобто

$$\Phi_i = \left\{\Phi_{ij}\right\} \quad i = \overline{1, I};\; j = \overline{1, I(i)};$$

$$\Phi_{ij} = \left\{\Phi_{ijk}\right\} \; i = \overline{1, I};\; j = \overline{1, I(i)};\; k = \overline{1, K(i, j)}.$$

Таким чином, зазначений процес розбиття, який відповідає графу-дереву T, корінь якого відповідає Φ, вершини першого рівня - Φ_i, другого рівня - Φ_{ij} і т.д., причому, Φ, Φ_i, Φ_{ij}, ... відображають компоненти c, c_i, c_{ij} і т. д. (рис. 1.2).

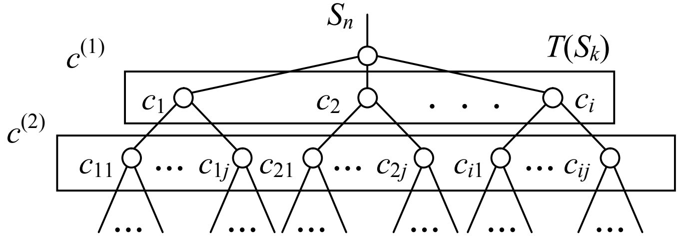


Рис. 1.2

Позначимо компоненти першого рівня $c^{(1)} = \{c_i\}$, другого $c^{(2)} = \{c_{ij}\}$ і т. д. Для l-го рівня визначимо множину відмінних один від одного компонент як

$$D^{(l)} = \{x \mid x \in X \subset c^{(l)} \wedge F(x') \neq$$
$$\neq F(x'') \forall x', \; x'' \in X, x' \neq x''\}, \quad l = 1, 2, \ldots$$

Розглянутий клас систем S' відображається коренем S дерева $T(S')$ при функціональній декомпозиції систем. При цьому основне завдання полягає в завданні множин $c^{(i)}$. Це здійснюється послідовним відбором компонент на кожному рівні у відповідності зі звичайним розчленуванням модельованої системи на підсистеми. Для оптимізації процесу декомпозиції може бути вибраний відповідний критерій (наприклад, число нетотожних компонентів в системі або в класі таких систем в цілому, складність системи та ін.).

Задамо для класу S' множину

$$D^{(l)}[T(S')] = \{y \mid y \in Y \subset Z = \bigcup_{k=1}^{N} D^{(l)}[T(S_k)] \wedge F(y') \not\equiv F(y'') \forall y',$$
$$y'' \in Y,\ y' \neq y''\},$$

яку можна визначити як сукупність нетотожних компонент l-го рівня для $S' \subset S$. Тоді повним описом S' є множина $P(S')$, яка містить елементи множин $D^{(l)}[T(S')]$, $l = 1, 2$ ... і необхідні елементи їх попарних декартових добутків. Очевидно, що $P(S')$ - упорядкована множина, що являє собою в загальному випадку нелінійну топологічну структуру. Більш зручна декомпозиція кожної системи $s_k \in S'$ у вигляді надлишкової деревовидної структури $T(S_k)$, яка має більш прості зв'язки між компонентами. Перехід до більш складній структурі доцільний на етапі програмної реалізації моделі.

Таким чином, функціональна декомпозиція зводиться до побудови дерева $T(S')$ і складанню таблиці функції T^F, де перераховуються вершини A дерева $T(S')$ і опис функції $F(A)$ відповідних компонент (крім вершини $A = S'$). Необхідно також вказати зв'язки, що існують між різними компонентами. Для цього можна запропонувати наступну процедуру.

Нехай вершина піддерева кореня $T(S')$, яка має нащадків, відображає об'єкт A, що складається з компонент $A_i = \overline{1,M}$. Позначимо $B(A)=\{A_1, ..., T(S')\}$. З метою стандартизації будемо вважати, що A і всі A_i мають по m входів $x_i(t)$ і т виходів $y_i(t)$ (при необхідності можна вводити фіктивні входи і виходи) і характеризуються операторами Ψ_k, $k=\overline{1,m}$. Зазвичай $m = 1...4$. Реальні зв'язки між A_i надаються у вигляді структурної схеми об'єкта A, яка описується набором $F(A)$, $F(A_i)$, $i = \overline{1,M}$, наведеному в T^F. Доповнимо $B(A)$ фіктивної компонентою A_0, входи якої є входи A, а виходи - виходи A. Вважаємо, що на зв'язки (канали) між A_i и A_j, накладені обмеження: до кожного входу приєднано не більше одного каналу.

Звідси випливає, що для опису декомпозиції вихідними даними служать $T(S')$, T^F, структурні схеми систем та їх підсистем. Елементам системи відповідають листя $T(S')$.

Функціонування об'єкта A повністю описується набором операторів $\{\Psi_k\}$ і виражається у вигляді

$$y = \Psi(x,t). \tag{1.1}$$

Таблиця T^Ψ, де перераховуються листя $T(S')$ і описи параметрів (1.1), повністю задає об'єкт A.

Стандартизована структурна схема об'єкта A, яка включає компоненти A_i, .., A_m і фіктивну компоненту A_0 зі входами відповідно $\boldsymbol{x}$ и $\boldsymbol{x}_i$, $i = \overline{1,m}$ і виходами $\boldsymbol{y}$ и $\boldsymbol{y}_i$, $i = \overline{0,M}$, де $\boldsymbol{x} = \{x_j\}$, $\boldsymbol{y} = \{y_j\}$, $\boldsymbol{x}_i = \{x_{ij}\}$, $\boldsymbol{y}_i = \{y_{ij}\}$) – може бути представлена у вигляді орграфа $G(A)$, вершинам якого відповідають x_{ij} і y_{ij}, ребрам зв'язки між x_{ij} і y_{ij}. Кількість вершин $G(A)$ дорівнює $2m \times (M+1)$. Орграф $G(A)$ має щонайменше $m \times (M+1)$ підграфов, кожен з яких однопов'язаний або складається з ізольованих вершин. Внаслідок припущення про однозначність входів A має місце співвідношення

$$x_i = R_i(y_0, \ldots, y_M)^T, \; i = \overline{0,M},$$

тут R_i є матричним представленням лінійного перетворення $m \times (M+1)$- мірного векторного простору виходів компонент об'єкта A в m-мірний векторний простір входів компоненти A_i.

Для складання математичної моделі в якості вихідних даних використовуються граф $T(S')$ і таблиці T^Ψ і T^R. Таблиця T^R визначається матрицею $\boldsymbol{R}$, що складається з підматриць $\boldsymbol{R}_i$.

Функціонування будь-якої системи з $S' \subset S$ описується математично узагальненою системою рівнянь, яка має ієрархічну, багаторівневу структуру і складається з систем нульового рівня і сімейств систем 1, 2, ..., (l_{max} – 1)-го рівнів, де l_{max} – максимальний рівень декомпозиції моделі системи. Чим більше рівнів системи модель відображає, тим ближче модель до оригіналу. На нульовому рівні модель дозволяє судити про те, чи існує взагалі оригінал.

Побудова та дослідження моделі системи є основним етапом системного аналізу. Тому поняття моделі системи взагалі й зокрема математичної моделі системи є однім з основних понять сучасної науки.

Контрольні запитання 

1. Сформулюйте поняття системи.
2. Що являє собою елемент системи?
3. Сформулюйте поняття цілі системи.
4. Наведіть найбільш розповсюджений варіантів класифікації систем за складністю.
5. Назвіть основні атрибути (властивості), які визначають систему.
6. Які величини визначають кількісну інформацію про систему?
7. Яка мета проведення системного аналізу?
8. Сформулюйте поняття системного підходу.
9. Сформулюйте поняття проблеми.
10. Сформулюйте основні задачі першого етапу системного аналізу.
11. Як професор Ешбі сформулював поняття системності?
12. Сформулюйте поняття системного аналізу.
13. У чому полягає смисл структуризації системи?
14. Які основні етапи можна виділити у більшості випадків практичного застосування системного аналізу?
15. Сформулюйте основу математичного моделювання, як головного інструменту системного аналізу.
16. Які задачі здійснюються на етапі декомпозиції?
17. Які стратегії декомпозиції застосовують частіше за все?

Глава 2. МОДЕЛІ ТА ЇХ КЛАСИФІКАЦІЯ

2.1. Модель як образ системи-оригіналу

Модель – це такій матеріальний або подумки представлений образ або прообраз (образ майбутнього), який в процесі дослідження заміщає систему-оригінал і використовується для дослідження її поведінки в заданих умовах як система-замінник. Модель відображає (копіює) *основні* (найбільш істотні) закономірності оригіналу й при цьому має перевагу в зручності поводження з нею.

Основні тому, що модель – це компроміс між неосяжною складністю оригіналу й обмеженими можливостями дослідження його поведінки. Як *образ* можуть служити макети, схеми, графи, математичні рівняння й т.п., а як *прообраз*, наприклад, зразки (зліпки), які використовуються як еталон для серійного відтворення виробу в іншому матеріалі.

Відносно своїх моделей реальна система буде служити *оригіналом* або *прототипом.*

В основу побудови моделей покладають такі принципи:

- реалізованість і завершеність;
- оптимальність і відповідність;
- конструктивна цілісність.

Принцип реалізованості і завершеності передбачає, що за моделлю, яка відповідає заданим вимогам можна побудувати реальну систему.

Оптимальність і відповідність передбачає правильність вибору цільової функції і обмежень. Конструктивна цілісність підкреслює, що модель складається з предикатів, які можуть бути реалізовані на практиці

Все різноманіття моделей можна розділити на *ізоморфні* (однакові за формою) і *гомоморфні* (подібні за формою) моделі.

Ізоморфні моделі - це моделі, що включають всі характеристики об'єкта - оригіналу, здатні, власне кажучи, замінити його. В ізоморфній моделі між моделлю й оригіналом спостерігається повна поелементна відповідність.

Можливість взаємного перенесення уявлень, понять і суджень з однієї системи на іншу ізоморфну їй систему лежить в основі концепції моделі. Ізоморфізм у даному випадку означає рівність форми реальної системи і моделі, яка має властивості рефлективності, транзитивності і симетричності.

Рефлективність передбачає, що копія даної системи служить її моделлю.

Властивість ***транзитивності*** підкреслює, що модель моделі є моделлю вихідної системи.

Згідно з умовою ***симетрії***, будь-яка система є модель кожної своєї моделі, тобто реальна система і її модель можуть мінятися місцями.

Якщо можна створити й спостерігати ізоморфну модель, то наші знання про реальний об'єкт будуть точними. У цьому випадку ми зможемо точно передбачити поведінку об'єкта-оригіналу. Класичними прикладами ізоморфізму є: *наочні* моделі предметів (малюнки, фотографії), топографічні карти місцевості, тощо.

Умова ізоморфізму моделі і оригінала вимагає подібності їх будови, яка передбачає рівночисельність предикатів (задана сукупність елементів і відносин). Виконання умови рівночисельності у ряді випадків виявляється складним або непотрібним, оскільки ніякого спрощення задачі при дослідженні тільки ізоморфних моделей не досягається. Спрощення може бути досягнуто у тому випадку, якщо знехтувати несуттєвими предикатами, тобто перейти від точного ізоморфного сприйняття до подання моделі як приблизного образу, що моделюється. Зазначена вимога приводить до поняття *гомоморфізму* (подібності) моделі реальної системи, яке також є рефлексивним і транзитивним, але несиметричним.

В основі ***гомоморфних моделей*** лежить не повна, а часткова подоба моделі досліджуваному об'єкту. Гомоморфний образ містить елементи й предикати (відносини), які відбивають лише істотні для проведення досліджень риси оригіналу. При цьому наявні у дослідника відомості про систему укрупнюються в більш компактну, зручну для подальшої обробки форму. У самому загальному випадку при побудові гомоморфних моделей дослідник відкидає ті характеристики, параметри об'єкта-оригіналу, які несуттєві для вивчення об'єкта. Вибір характеристик об'єкта-оригіналу, які при цьому зберігаються і увійдуть в модель, визначається цілями моделювання.

Отже, повна поелементна відповідність між моделлю й оригіналом у гомоморфних моделях відсутня. У наслідок цього спрощується побудова моделі й інтерпретація результатів дослідження. Прикладом гомоморфної моделі може служити, наприклад, план, а не детальна карта певної ділянці земної поверхні.

Заміна об'єкта його гомоморфною моделлю дозволяє проводити аналіз системи за допомогою *спрощеної* моделі, що дає серйозні переваги в методології й технології дослідження системи.

Практично у всіх науках про природу, живу та неживу, про суспільство, побудова та використання моделей є потужним знаряддям пізнання. Реальні об'єкти і процеси бувають настільки багатогранні і складні, що кращим (а іноді і єдиним) способом їх вивчення є побудова та дослідження моделі, що відображає лише якусь грань реальності, і яка, через те багаторазово більш проста, ніж ця реальність. Багатовіковий досвід розвитку науки довів на практиці плідність такого підходу. Більш конкретно, необхідність використання моделей на етапах досліджень визначається тим, що багато об'єктів (або проблеми, які стосуються цих об'єктів) безпосередньо досліджувати взагалі неможливо, або ж такі дослідження вимагають занадто багато часу і коштів

Смислове навантаження терміна «модель» багатопланове, тому існують різні варіанти класифікацій моделей систем.

2.2. Класифікація моделей

До класифікації моделей можна підходити з різних позицій, поклавши в основу класифікації різні принципи. Можна класифікувати моделі за *галузями знань*, наприклад, як історичні, економічні, біологічні, соціологічні тощо.

За *областю використання* можна застосувати класифікацію, зображену на рис. 2.1.

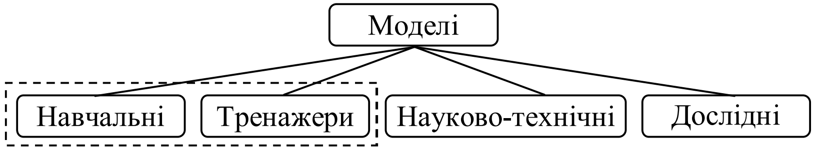


Рис. 2.1

Навчальні моделі це зображальні навчально-наочні посібники, які штучно відтворюють натуральні об'єкти і передають їх структуру, істотні властивості, зв'язки та відносини. При цьому допускається умовність у передачі властивостей оригінала, зменшення або збільшення розміру, схематизація в передачі принципів будови об'єктів, умовність забарвлення, тощо. Це можуть бути наочні посібники, анатомічні моделі, симулятори, навчальні програми.

Особливе місце серед навчальних моделей займають ***тренажери*** – спеціальні технічні засоби навчання заданої дидактичної спрямованості, які моделюють умови виробничої діяльності фахівця. Тренажери дозволяють створити оптимальні умови для ефективного формування професійних умінь і навичок, необхідних у практичній діяльності.

Науково-технічні моделі створюють для дослідження процесів і явищ шляхом проведення на них експериментів, в процесі яких коригуються прототипи систем. Як засіб осмислення реальних зв'язків науково-технічні моделі дозволяють упорядкувати і систематизувати знання про систему, встановити ресурси, необхідні для її проектування. До таких моделей можна віднести, наприклад, прилад, що імітує розряд блискавки, стенд для перевірки аерометричних приладів, на якому імітуються зміни земної атмосфери.

Дослідні моделі - це зменшені або збільшені копії проектованого об'єкта. Їх називають також натурними і використовують для дослідження об'єкта і прогнозування його майбутніх характеристик.

Наприклад, модель планера літака, що встановлена в аеродинамічній трубі, дозволяє оцінити основні характеристики динамічної стійкості літака, а модель гідростанції ще при розробці проекту допомагає вирішити гідротехнічні, екологічні та багато інших проблем.

Корисність авіаційних дослідних моделей як засобів експериментування проявляється в можливості створення важкодосяжних і аварійних умов польоту, виявлення причин несправностей і їх впливу на безпеку польотів.

Якщо поставити на перше місце *мету моделювання*, серед якої основними є: розуміння, управління, прогнозування, то можна прийти до наступної класифікації:

- дескриптивні моделі – моделі опису систем;
- оптимізаційні моделі – моделі керування об'єктами;

• багатокритеріальні моделі – моделі для розв'язування оптимізаційних завдань, в яких є декілька цільових функцій;

• ігрові моделі;

Моделюючи рух астероїда, що летить до Землі, дослідник, використовуючи ***дескриптивну модель*** польоту астероїда, описує ситуацію. За допомогою моделі аналізується траєкторія польоту астероїда, відстань, на якому він пройде від Землі, наслідки входу астероїда в земну атмосферу і т.д. Немає ніяких можливостей вплинути на рух астероїда або щось змінити в процесі моделювання, ставляться чисто описові мети.

В ***оптимізаційних моделях*** дослідник може впливати на процеси, намагаючись домогтися якоїсь мети. У цьому випадку в модель входить один або кілька параметрів, доступних змін. Наприклад, змінюючи режими роботи авіадвигуна, можна оптимізувати режим набору висоти за критерієм мінімізації витрат палива.

Часто доводиться оптимізувати процес за кількома параметрами відразу, причому цілі можуть бути, досить, суперечливими. Наприклад, знаючи ціни на продукти і потреби людини в їжі, організувати харчування великих груп людей (в армії, літньому таборі, тощо) якомога корисніше і якомога дешевше. Зрозуміло, що ці цілі зовсім не збігаються, тому при моделюванні буде кілька критеріїв, між якими треба шукати баланс. У цьому випадку говорять про ***багатокритеріальні моделі***.

Ігрові моделі - це військові, економічні, спортивні, ділові ігри. Вони як би репетирують поведінку об'єкта в різних ситуаціях з урахуванням можливої реакції з боку конкурента, союзника або супротивника. У спеціальному розділі сучасної математики – теорії ігор, що вивчає методи прийняття рішень в умовах неповної інформації, експерименти проводять саме на ігрових моделях – моделях для опису та дослідження конфліктних ситуацій, в яких різні учасники мають різні інтереси, які зазвичай не співпадають.

Моделі можуть бути класифіковані за ієрархічною ознакою модельованих систем. За мірою складності систем їх моделі можна розділити на такі рівні.

• ***Модель пасивної системи*** – це описова моделі системи, в якій дослідник немає можливості вплинути на її поведінку. Наприклад, тектонічні процеси і їх моделювання.

• ***Модель керованої системи*** – модель системи, яка реагує на керуючий вплив. Наприклад, модель динаміки польоту літака.

• ***Модель керуючої системи*** – модель системи, яка здійснює управління будь-якими процесами або об'єктами. Прикладами таких моделей є моделі автоматичних систем управління, роботів.

• ***Моделі інтелектуальних систем*** – моделі систем з власною системою прийняття рішень, що вирішують творчі задачі в конкретній предметній області, знання про яку зберігаються в пам'яті такої системи. Такі системи, сумісно з іншими подібними системами складають "ігрову систему", яка самостійно моделює ситуацію і відповідає на зовнішні впливи відповідно до своєї моделі. Прикладами служать: система суспільних відносин, моделювання театру військових дій і політичних ситуацій.

Залежно від того, як в моделях відображується динаміка процесів, що відбуваються, тобто з урахуванням ***фактора часу*** виділяють статичні та динамічні моделі.

Статична модель показує структуру системи та взаємини між елементами прототипу в статиці. При будівництві будинку розраховують міцність і стійкість до постійного навантаження його фундаменту, стін, балок – це статична модель будівлі. Схема бортового навігаційного комплексу, яка представлена у вигляді набору елементів, частина яких може бути об'єднана в більш великі структури – підсистеми, це статична модель (модель складу комплексу). Структурна схема бортового навігаційного комплексу, що показує зв'язок між підсистемами комплексу також є прикладом статичної моделі (моделі структури комплексу). Проста фотографія об'єкта - це його статична модель. Таким чином, статична модель - це як би одномоментний зріз інформації з об'єкту – зріз, що відображає оригінал в окремий момент часу.

Динамічна модель тісно пов'язана зі структурою, але розглядає відносини між елементами прототипу в часі (в динаміці), тобто динамічна модель призначена для вивчення особливостей функціонування та розвитку системи. При цьому під *функціонуванням* маються на увазі процеси, які відбуваються в системі (навколишньому середовищу) при реалізації фіксованої цілі. А *розвитком* називають те, що відбувається з системою при зміні її цілей, тобто характерною рисою розвитку є необхідність зміни структури системи.

Динамічна модель дозволяє побачити зміни об'єкта. На відміну від статичної моделі будівлі, його динамічна модель дозволяє оцінити вплив на конструкцію будівлі сейсмічних коливань, рухів ґрунтових вод, вітрових збурень і інших змінних в часі чинників. Прикладами динамічних моделей можуть служити: модель контуру системи автоматичного керування у вигляді набору типових динамічних ланок, які розкривають перетворення вхідних сигналів; часова діаграма роботи системи передачі даних; перехідна функція системи і т.п.

За способом подання інформації моделі діляться на дві великі групи (див. рис. 2.2): матеріальні й абстрактні. Назви цих груп як би показують, з чого «зроблені» моделі.

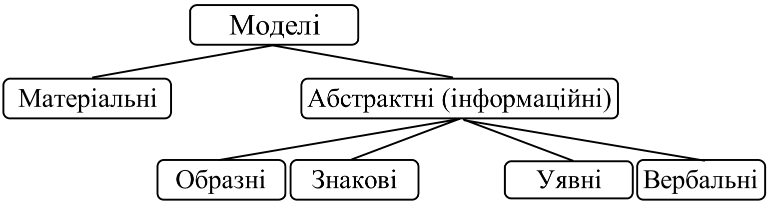


Рис. 2.2

Матеріальні моделі інакше можна назвати предметними, фізичними. Вони завжди мають реальне втілення і відтворюють геометричні та фізичні властивості об'єкта-оригінала.

Предметні моделі дозволяють уявити в матеріальної наочній формі об'єкти і процеси, недоступні для безпосереднього дослідження (дуже великі або дуже маленькі об'єкти, дуже швидкі або дуже повільні процеси та ін.).

Макети будівель і споруд дозволяють архітекторам вибрати найкращі містобудівні рішення, модель авіаційно-технічної бази (у зменшеному масштабі), призначена для вивчення нового технологічного процесу ремонту повітряного судна, моделі літаків і кораблів дозволяють інженерам обрати їх оптимальну форму.

Матеріальні моделі в деяких випадках можуть навіть зберігати субстанцію елементів прототипу. Субстанціональна матеріальна модель зосереджує увагу на матеріалі прототипу і зовні нагадує досліджуваний об'єкт. Прикладом такої моделі може служити шматок шини як модель шасі літака.

Абстрактні моделі це конструкції, побудовані за допомогою мислення. Найчастіше абстрактна модель дає лише якісні характеристики модельованого об'єкта чи явища. До абстрактних моделей не можна доторкнутися, оскільки вони не мають матеріального втілення. Основу таких моделей становить інформація про навколишню дійсність, тому ці моделі іноді називають інформаційні.

Абстрактні (інформаційні) моделі в свою чергу поділяються на уявно-вербальні і образно-знакові.

Уявні моделі формуються людиною в результаті роздумів, висновків, іноді у вигляді деякого образу. Прикладом уявної моделі є модель поведінки при переході через дорогу. Людина аналізує ситуацію на дорозі (який сигнал подає світлофор, як далеко знаходяться машини, з якою швидкістю вони рухаються і т. п.) і виробляє модель поведінки. Якщо ситуація змодельована правильно, то перехід буде безпечним, якщо ні, то може статися дорожньо-транспортна пригода.

Уявна модель може бути виражена розмовною формою. В цьому випадку вона називається ***вербальною*** (усною) і використовується людиною для передачі своїх думок іншим людям.

Образні моделі являють собою графічні образи об'єктів, що зафіксовані на матеріальних носіях інформації (папері, фото- і кіноплівці і т.п.).

До подібних моделей можна віднести: фотографії, рисунки, картини. Образними моделями є також графічні інформаційні моделі: карти, креслення, діаграми, графіки, схеми.

Незважаючи на наочність образних моделей, вони є проміжними, перехідними до вищої форми абстрактних моделей, перетворюючись зрештою в знакові моделі. ***Знакова модель*** це абстрактна інформаційна модель виражена знаками, тобто засобами будь-якої формальної мови. На відміну від образної моделі в знаковій, моделі більше слів і чисел. Знакова модель, що сформульована на мові математичної символіки, називається математичною моделлю

Математична модель являє собою формалізований опис системи на деякій абстрактній мові (у вигляді диференціальних, інтегральних або різницевих рівнянь; у вигляді логіко-математичних схем і алгоритмів; з використанням теорії множин, теорії графів, математичної статистики; тощо). Реалізовану на комп'ютері засобами програмного середовища математичну модель іноді називають ***комп'ютерною моделлю***. Комп'ютерні моделі, використовуються для чисельного математичного моделювання та візуалізації явищ і процесів, що відбуваються в моделі системи.

Відзначимо, що існують і інші підходи до класифікації абстрактних моделей, загально прийнята точка зору тут ще не встановилася.

Саме математичні моделі ми і будемо розглядати в подальшому як основний інструмент опису поведінки складних систем цивільної авіації.

Контрольні запитання

1. Сформулюйте поняття моделі взагалі.
2. Наведіть приклади образів та прообразів оригіналу.
3. Які принципи покладають в основу побудови моделей?
4. На які моделі можна поділити все різноманіття моделей?
5. Сформулюйте поняття ізоморфної моделі.
6. Що означає рефлективність моделі та системи?
7. Що означає симетричність моделі та системи?
8. Що означає транзитивність моделі та системи?
9. Що покладено в основу побудови гомоморфних моделей?
10. Що являє собою дескриптивна модель?
11. Що являє собою оптимізаційна модель?
12. На які рівні можна умовно поділіть моделі за мірою складності?
13. Наведіть приклади статичних моделей.
14. У чому головне призначення динамічних моделей?
15. На які дві великі групи діляться моделі за способом подання інформації?
16. На які групи можна поділити абстрактні (інформаційні) моделі?
17. Що являє собою вербальна модель?
18. Наведіть приклади образних моделей.
19. У чому головна відмінність знакової моделі від образної?
20. Сформулюйте поняття математичної (абстрактно-знакової) моделі.
21. Яку модель іноді називають комп’ютерною?

Глава 3. МАТЕМАТИЧНІ МОДЕЛІ СИСТЕМ ТА ЇХ КЛАСИФІКАЦІЯ

3.1. Поняття математичних моделей системи

Математична модель (абстрактна знакова модель) це формалізований опис системи з використанням математичної символіки у вигляді:

– математичних формул, що базуються на різних математичних теоріях (теорії диференціальних чи інтегральних рівнянь, теорії множин, абстрактної алгебри, математичної логіки, теорії ймовірності, тощо);
– схем алгоритмів;
– мов програмування або машинного коду.

Математична модель починається з розробки попереднього неформалізованого (концептуального) опису системи. Система поділяється на взаємопов'язані підсистеми й елементи (виділяються окремі найбільш істотні складові частини досліджуваної системи й зв‘язки між ними), а можливі зовнішні впливи середовища представляються у вигляді сукупності елементарних впливів. Формулюється концептуальна модель.

Розглядаючи описану таким чином реальну систему, будують моделі системи та підсистем відповідно до поставленої цілі і визначаються їх параметри при чому основний наголос вже робиться на формалізованому описі. Тобто, починаються спроби створення математичної моделі, яка будується на основі спрощень та ідеалізацій об’єкта.

Створюючи математичну модель, необхідно:

- виділити припущення, на яких буде базуватися математична модель;

- визначити вхідні дані та шукані результати;

- записати математичні співвідношення (формули, рівняння, нерівності), які зв’язують шукані результати з вхідними даними.

Щоб відповідати своєму призначенню моделі повинні володіти рядом властивостей, які визначаються як сутністю методології, так і конкретними умовами. Зазначимо деякі з них, які дозволяють в тій чи іншій мірі ототожнювати модель з системою-оригіналом і носять принциповий і універсальний характер.

Відповідність між моделлю і системою-оригіналом можна виразити наступними властивостями моделі:

- цілеспрямованість і повнота;
- скінченність;
- спрощеність;
- наближеність;
- адекватність.

Цілеспрямованість моделі полягає в тому, що для заданої системи-оригіналу необхідно створити таку модель, яка у повній мірі відображав би оригінал з точки зору заданої цілі моделювання. Ця ціль визначає, які властивості системи-оригіналу з безлічі несуттєвих вважаються істотними при проведенні певних досліджень і тому повинні бути у повної мері віддзеркалені у моделі. ***Повнота*** моделі полягає в тому, що вона має відображати всі істотні з точки зору цілі моделювання властивості оригіналу.

Скінченність моделі обумовлюється її цілеспрямованістю але визначає те, що модель відтворює лише скінчену кількість властивостей та відношень (навіть істотних з точки зору цілеспрямованості). До того ж що ресурси (інформаційні, фінансові, технічні й т.д.) моделювання, а також знання про оригінал як інтелектуальні ресурси кінцеві, а тому об'єктивно обмежують можливості моделювання. Тому дослідник в основному має справу з кінцевомірними моделями.

Скінченність моделей робить їх ***спрощеність*** неминучої, але в наукової практиці ця спрощеність є припустимою (для будь-якої цілі виявляється достатнім спрощене відображення дійсності). Спрощувати модель можна доти, поки зберігаються основні властивості, характеристики й закономірності, що властиві оригіналу.

Скінченність і спрощеність моделі характеризують якісні розходження між оригіналом и моделлю, а термін ***наближеність моделі*** характеризує кількісну сторону цього розходження.

Кількісну міру наближеності можна ввести шляхом порівняння, наприклад, грубої моделі з більше точної еталонної (повної, ідеальної) моделлю або з реальною моделлю. Наближеність моделі до оригіналу існує об'єктивно, тому що модель як інший об'єкт відбиває лише окремі властивості оригіналу. Тому ступінь наближеності моделі до оригіналу визначається метою моделювання. Намагання підвищити точність моделі призводить до її надмірного ускладнення, а отже, до зниження її практичної цінності.

Необхідною умовою для переходу від дослідження об’єкта до дослідження моделі і подальшого перенесення результатів на об’єкт дослідження – вимога адекватності моделі і об’єкта.

Адекватність моделі це відповідність моделі модельованому об'єкту або процесу. Це, мабуть, найголовніша властивість моделі, яка визначає можливість її використання. При цьому мається на увазі адекватність не взагалі, а адекватність за такими властивостям моделі, які є для дослідника істотними. Повна адекватність означає тотожність між моделлю і прототипом. Оскільки будь-яка модель простіша за оригінал, говорити про повну адекватність принаймні некоректно.

Остаточне судження про адекватність моделі може дати лише практика, тобто порівняння моделі з оригіналом на основі експериментів з об'єктом і моделлю. Перевірка на адекватність полягає в доказі факту, що точність результатів, отриманих з використанням моделі, буде не гірше точності розрахунків, зроблених на підставі експериментальних даних.

На неіснуючій, наприклад, на проектованій системі провести експерименти з подальшою перевіркою моделі цієї системи на адекватність, природно, не представляється можливим. Єдиний спосіб подолати цю перешкоду полягає в тому, щоб прийняти як еталон концептуальну модель системи (змістовний опис модельованої системи) або модель попередніх розробок схожої системи. Тоді оцінка адекватності полягає в перевірці того, наскільки коректно модель, що розробляється, відображає еталоні моделі.

Таким чином, властивість адекватності є найважливішою вимогою до моделі, але розробка високоточних і надійних методів перевірки адекватності залишається як і раніше складною задачею. Тому висновки щодо адекватності моделі не носять формальний характер, а ґрунтуються на досвіді та інтуїції дослідника.

Існують й інші вимоги, що до моделей систем-оригіналів (універсальність, економічність, здатність до розширення, результативність, потужність, наочність, тощо).

У рамках абстрактних схем вся безліч математичних моделей поділяється на дві великі групи – *аналітичні* та *імітаційні* моделі.

Аналітичні моделі описують процеси функціонування складних систем у вигляді явних функціональних залежностей – аналітичних виразів (звідси й походить їхня назва). Ці співвідношення можуть бути отримані шляхом теоретичних викладок, а також експериментально, в результаті обробки дослідних даних. У другому випадку аналітичну модель також називають емпіричною. Аналітичні моделі використовують при аналізі функціонування систем управління, для опису динаміки польоту, оптимізації параметрів систем і т.п.

Аналітичні моделі за кількістю аргументів (факторів), від яких залежить значення функції, діляться на однофакторні ($y = f(x)$) і багатофакторні ($y = f(x_1,x_2,\ldots,x_n)$). За кількістю параметрів, значення яких необхідно визначити, аналітичні моделі, діляться на однопараметричні ($y = ax,\ y = ae^x,\ y = x^b$) і багатопараметричні ($y = a + bx + cx^2$ и т.д.)

Аналітичні моделі також поділяються на декілька видів залежно від математичної проблеми:

- рівняння (алгебраїчні, трансцендентні, диференціальні, інтегральні);
- апроксимаційні задачі (інтерполяція, екстраполяція, чисельне інтегрування та диференціювання);
- оптимізаційні задачі.

Аналітичні моделі розробляються для дослідження системи, що моделюється, аналітичними методами, а коли не вдається знайти рішення у загальному вигляді, то воно шукається для конкретних початкових умов чисельними методами. Аналітичні рішення, що представлені логічними конструкціями, які виражають результати дослідження через вихідні дані, зазвичай зручніше та інформативніше рішень, отриманих чисельними методами. Однак все ж основним інструментом реалізації аналітичних моделей є чисельні методи, які дозволяють звести рішення задачі до обчислення кінцевого числа арифметичних дій над числами і отримати розв‘язок у вигляді числових значень. Проте, рішення, що отримується чисельними методами, зазвичай є наближеним, тобто містить деяку похибку.

Обмежуючими факторами, які перешкоджають широкому використанню аналітичних моделей, є труднощі вибору конкретних обмежень і припущень при ускладненні моделі і специфічний характер отриманих рішень, який залежить від заданих початкових умов.

Коли об'єкт моделювання настільки складний, що отримання його аналітичного опису з різних причин перетворюється в важко вирішувану проблему, його аналітична модель замінюється імітатором або імітаційної моделлю.

Імітаційна модель - це логіко-математичний опис досліджуваної системи, який забезпечує проведення експериментів на комп'ютері з метою аналізу та оцінки функціонування модельованої системи.

Наприклад, необхідно дізнатися, коли і де зустрінуться два об'єкти, починаючи рух з відстані D назустріч один одному зі швидкостями V_1 и V_2 відповідно.

За рахунок суттєвої ідеалізації (дорога ідеально пряма, швидкості об'єктів сталі, відсутні перешкоди для руху і т.п.) можна отримати просту *аналітичну модель*, яка має вигляд:

$$T = D/(V_1 + V_2);\ \ S_1 = V_1 \cdot T.$$

В даній аналітичній моделі протягнута низка обчислень від вихідних даних до виходу. Після обчислення правої частини першого рівняння її значення присвоюється змінної, що стоїть в лівій частині. Далі значення цієї змінної застосовано в правій частині наступного рівняння.

При імітаційному моделюванні замість аналітичного опису взаємозв'язків між входами і виходами досліджуваної системи будується алгоритм, що відображає послідовність розвитку процесів усередині системи. Тому основною відмінністю імітаційних моделей від аналітичних є обов'язкова наявність деякого лічильника, який дозволяє моделювати процес за кроками або за деталями процесу.

Повторюючи покроково розрахунок в циклі, на кожному етапі роботи алгоритму можна імітувати сукупність послідовних дій процесу (рис. 3.1). При цьому процес береться не в цілому, а в деталях, покроково.

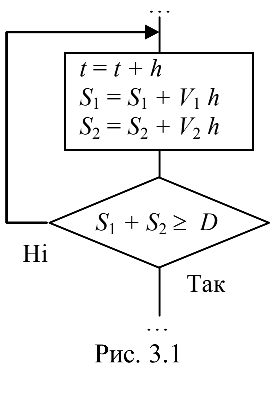


Рис. 3.1

Змінна t є координатою, а значить, відстежується лічильником з кроком h. Ідея імітації - просувати обидва об'єкти на величину $V \cdot h$ на кожному такті, де h – досить мала величина. Оскільки ми розглядаємо множини актів руху відокремлено, можна з плином часу міняти всі змінні моделі, наприклад, V. Якщо шлях пройдений першим об'єктом значний (S_1), то можна на деякий час влаштувати привал ($V_1 = 0$). Зупинка процесу імітації визначається сумою відстаней, пройдених об'єктами назустріч один одному, і порівнянням її з відстанню D.

Найбільший ефект застосування імітаційних моделей досягається при дослідженні систем, на функціонування яких істотно впливають випадкові чинники. Наприклад, на шляху руху об'єктів можна встановити перешкоду – шлагбаум, який працює за працює за імовірнісним (випадковим) законом. Якщо шлагбаум відкритий, то об'єкт може продовжувати рух, в іншому випадку він не має права цього робити.

Промоделювати імовірнісну роботу шлагбаума можна за допомогою генератора випадкових чисел. Частоту роботи шлагбаума, а також тривалість його закриття можна контролювати.

Оскільки алгоритм використовує випадкові числа в якості вихідних даних, то при моделюванні доведеться провести кілька експериментів, тому що результат одного експерименту випадковий і ні про що не говорить.

При багаторазових експериментах проводиться накопичення статистичних даних, які і є предметом досліджень. За цими даними можна отримати достатньо стійку статистику, і тоді оцінка характеристик модельованої системи проводиться на рівні статистичної обробки результатів моделювання. Тому імітаційне моделювання зазвичай прирівнюється до статистичного експерименту. Таким чином, математична статистика і теорія ймовірностей є математичними основами імітаційного моделювання.

Імітаційне моделювання застосовується для опису властивості системи практично будь-якої складності, але поведінка складових об'єктів цієї системи повинна бути логічно чітко сформульована. Обмеженнями для застосування імітаційного моделювання можуть бути лише недостатня кваліфікація виконавця, а також вимога щодо адекватності моделі і вимоги досягнення надто великої точності результату. Якщо складність аналітичної моделі з ускладненням модельованого об'єкта зростає з прискоренням, як показано на рис. 3.2, то складність імітаційної моделі, починаючи з деякого рівня S_0, зростає незначно.

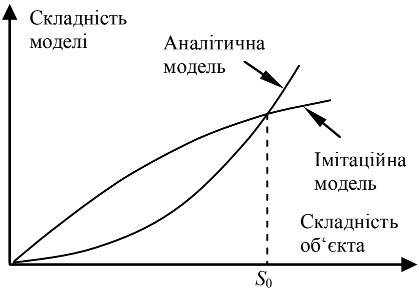


Рис. 3.2

Імітаційне моделювання дозволяє розкласти велику модель на частини, якими можна оперувати окремо, створюючи інші, більш прості або, навпаки, більш складні моделі. Імітаційні моделі дозволяють досить просто враховувати такі фактори, як наявність дискретних і безперервних елементів, нелінійні характеристики елементів системи, багаточисельні випадкові впливи та інші, які найчастіше створюють труднощі при аналітичних дослідженнях.

При розробці імітаційних моделей деякі взаємозв'язки в моделі можуть бути представлені і у вигляді аналітичних залежностей. Наприклад, при моделюванні польоту безпілотного літака відпрацювання команд, що надходять на борт літака, може бути описано на рівні логіки, а виникаючі при цьому перевантаження розраховуються аналітично.

До переваг імітаційного моделювання у порівнянні з аналітичними моделями можна віднести:

- простоту алгоритму;
- можливість багаторазового вимірювання параметрів моделі, які цікавлять дослідника;
- можливість дослідження складних сценаріїв поведінки системи;
- стійкість до випадкових збоїв комп'ютера, оскільки при великому числі реалізацій моделі збій в одній з них спотворить статистику несуттєво.

Недоліком імітаційного моделювання є те, що розв'язок, результат є чисельним, окремим, тобто справедливим тільки для конкретних значень вихідних даних. Щоб отримати функціональні залежності між параметрами досліджуваного процесу (системи) потрібно зробити дуже велику кількість варіантів моделювання. Якщо складність задачі, потрібна точність рішення, можливості математики і здатності дослідника дозволяють побудувати математичну аналітичну модель, то найкраще використовувати саме її.

Особливості математичних моделей залежать від галузі науки, в який вони використовуються. Тому математичні моделі у фізиці, біології, соціології і т. п. мають свою специфіку. Саме для математичних моделей є актуальною їхня класифікація, наприклад, за метою моделювання (дескриптивні, оптимізаційні, багатокритеріальні та ігрові моделі), або за мірою складності модельованої системи (моделі пасивної, керованої, керуючої або інтелектуальної системи). Проте класифікація математичних моделей систем, за математичним та логіко-математичним апаратом їх опису є найбільш злободенною.

3.2. Класифікація математичних моделей систем, за математичним та логіко-математичним апаратом їх опису

Математичні моделі можна класифікувати за Шеноном[3.1] як:

– статичні, динамічні,
– дискретні, безперервні,
– детерміновані, стохастичні.

Все різноманіття наявних типів математичних моделей (див. табл. 3.1) можна класифікувати за кількома основними ознаками..

Математичні моделі систем *Таблиця 3. 1*

Статичні	Динамічні
Дискретні	Безперервні
Лінійні	Нелінійні
Стаціонарні	Нестаціонарні
Зосереджені	Розподілені
Детерміновані	Недетерміновані: - Стохастичні (імовірнісні); - Нечіткі.

[3.1] Клод Елвуд Шенон – американський вчений, лауреат Нобелівської премії фахівець в області інформації, кібернетики, математики.

За категорією часу, математичні моделі класифікуються як *статичні і динамічні; дискретні і безперервні.*

3.2.1. Статичні і динамічні моделі

Математична модель системи називається статичною, якщо серед параметрів, що беруть участь в її опису, немає часового параметру. Наприклад, закон Ома для однорідної ділянки кола $I = U / R$ не описує перехідні характеристики зміни струму I в ланцюгу в залежності від характеру зміни напруги U і виду електричного опору R.

Статичні моделі відображають стан системи (дають її зріз) в кожен момент часу. У статичній моделі значення виходу системи $y(t)$ залежить від значення входу $x(t)$ тільки в той же момент часу $t = t_k$. Символічно ця властивість записується так:

$$y\,(t_k) = F\,[x(t_k)], \tag{3.1}$$

де F – символ деякого перетворення (оператора перетворення).

Проведення експериментів з вимірювання вихідної величини статичної моделі в різні моменти часу показує, що завжди при однаковому значенні вхідного сигналу результат буде один і той же, тобто один експеримент не залежить від іншого. Тому кажуть, що статичні системи не володіють «пам'яттю», а також описують явища без розвитку.

Крім явних функціональних залежностей (3.1), статичні моделі можуть задаватися неявно, наприклад, у вигляді рівняння $\Phi(y\,(t_k), x(t_k)) = 0$, яке має бути однозначно розв‘язуваним відносно $y(t_k)$.

Статичними моделями користуються, коли в рамках поставленого завдання (з точки зору досягнення обраної цілі) інерційністю і «пам'яттю» реальної системи можна знехтувати. Це можливо при виконанні ряду умов, до числа яких входять такі:

– система стійка, тобто перехідні процеси після стрибкоподібної зміни входів згасають. Кінцевий час згасання із заданою точністю позначимо через $t_{пер}$;
– входи змінюються повільно, тобто. $\Delta t_{вх} > t_{пер}$, де $\Delta t_{вх}$ – час між змінами вхідних впливів;

– виходи вимірюються рідко, тобто $\Delta t_{вих} > t_{пер}$, де ($\Delta t_{вих} = t_{k+1} - t_k$) проміжки між вимірами вхідних величин.

За допомогою статичних моделей системи описуються її статичні характеристики – залежності між вхідною і вихідною величиною у сталому режимі.

Математична модель, в якій розкриваються причинно-наслідкові зв'язки, що визначають процес переходу системи з одного стану в інший, називається ***динамічною моделлю***.

Модель динамічної системи характеризується своїм початковим станом і законом, за яким система переходить з початкового стану в інший, тобто динамічні системи, пам'ятають свій минулий стан (володіють пам'яттю).

Динамічні моделі, на відміну від статичних моделей, враховують зміни, що відбуваються в системі з плином часу, тобто динамічні моделі простежують поведінку систем. Тому в запису моделі динамічних систем присутня похідна, яка зв'язує минулий стан системи з теперішнім. Наприклад, динамічна система може бути представлена математичною моделлю у вигляді диференціального рівняння другого порядку:

$$a_0 \frac{d^2 y}{dt^2} + a_1 \frac{dy}{dt} + a_2 y = bU .$$

Чим більшою пам'яттю володіє система, тим більше станів з минулого впливають на теперішній стан, тим більша ступінь старшої похідної використовується в запису моделі.

Для різних об'єктів і систем розроблено велику кількість динамічних моделей, які описують процеси з різним ступенем детальності: від самого загального поняття динаміки руху взагалі, до формальних математичних моделей конкретних процесів, типу рівнянь руху в механіці або хвильових рівнянь в теорії поля.

Властивості динамічних систем визначають їх динамічні характеристики. Динамічна характеристика - це реакція системи на збурення (залежність зміни вихідних змінних і вхідних від часу).

3.2.2. Дискретні та безперервні моделі

Дискретні моделі описують дискретні, перервні процеси, а безперервні моделі представляють системи з неперервними процесами. Система, в якій використовуються дискретні сигнали, називається ***дискретною системою***. Система може бути дискретною або безперервною за входами X, за виходами Y і за часом T залежно від того, дискретними або безперервними є множини X, Y, T відповідно. При цьому під *дискретною* розуміється кінцева або зліченна множина, а під *безперервною* (континуальною - триваючу довго, без перерви) - будемо розуміти зв'язну (не розбиту на частини) числову множина. Системи, в яких дискретність множини X спричиняє дискретність Y, відносяться до класу цифрових автоматів, моделі яких використовуються при так званому дискретно-подієвому моделюванні (вид імітаційного моделювання). У таких системах під множинами X и Y розуміють дискретні події або стани.

Цифровий автомат - це модель пристрою, реакція якого залежить не тільки від входу, але і від того що було раніше, тобто від стану в попередній момент часу. Тому таку модель також називають кінцевим автоматом, підкреслюючи, що це математична модель пристрою з кінцевої пам'яттю, яка зберігає його стан. Залежно від стану автомат виконує ту або іншу дію.

Кінцевий автомат переробляє велику кількість вхідних дискретних сигналів X в множину вихідних сигналів Y, *не враховуючи* динамічний стан, який виникає під час перехідних процесів.

Залежно від дискретності або безперервності часу T розрізняють синхронні і асинхронні автомати.

Синхронні автомати (кінцеві синхронні автомати) являють собою опис логічних функцій, в яких відсутня така змінна, як час, оскільки синхронні моделі використовують для аналізу сталих станів. Прикладом синхронної моделі може служити, наприклад, логічна схема тригера.

У синхронних кінцевих автоматах моменти часу, в які автомат зчитує вхідні сигнали, визначаються синхронізуючими тактовими імпульсами зовнішнього генератора, які не синхронізовані з його вхідними сигналами. Протягом такту вхідний та вихідний сигнали, а також стан незмінні. Після чергового синхронізуючого сигналу відповідно до правил функціонування, які враховують поточний стан автомата, він переходить в новий стан і видає відповідний вихідний сигнал, після чого автомат може сприймати наступне значення вхідного сигналу.

Логіко-математичний опис роботи автомата зазвичай задається (див. рис. 3.3) *таблицею переходів* і *таблицею виходів.*

Таблиця переходів

Вхідний сигнал	Стан			
	a_0	a_1	a_2	a_3
x_1	a_1	a_2	a_3	a_3
x_2	a_0	a_0	a_0	a_0

Таблиця виходів

Вхідний сигнал	Стан			
	a_0	a_1	a_2	a_3
x_1	y_2	y_2	y_1	y_2
x_2	y_2	y_2	y_2	y_3

Рис. 3.3

В таблиці переходів показують в який стан a_i переходить автомат від того чи іншого вхідного сигналу (x_1, x_2) в залежності від свого поточного стану (a_0… a_3). У таблиці виходів показують, який вихідний сигнал y_i генерує автомат залежно від вхідного сигналу і поточного стану автомата.

Опис операторів переходів і виходів можуть бути задані і однією таблицею, за якою однозначно визначаються і переходи, і виходи автомата. Інший спосіб завдання автомата ґрунтується на *орієнтованих графах* (рис. 3.4).

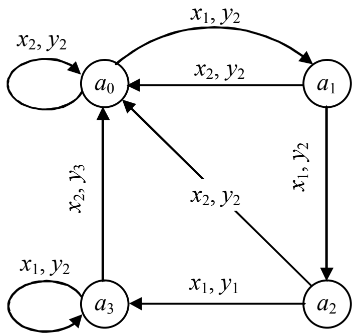


Рис. 3.4

Вершини графа ототожнюють внутрішній стан автомата, а кожна гілка графа (орієнтована лінія, стрілка якої указує наступний стан автомата) позначається вхідним сигналом, що викликає в автоматі відповідний даної гілки перехід, і вихідним сигналом, який виникає при цьому переході. Якщо деякий вхідний сигнал не змінює стан автомата, то відповідна гілка замикається на вершині графа, з якої вона виходить. Граф на рис. 3.4 відображає автомат, заданий таблицями переходів і виходів, наведеними на рис. 3.3.

Якщо цифровий автомат описує систему з безперервним часом, то відповідні системи називають ***асинхронними автоматами***. На відміну від синхронних автоматів, в асинхронних автоматах моменти переходів з одного стану в інший заздалегідь не визначені. Асинхронний автомат змінює свій стан тільки після зміни вхідного сигналу, що постійно присутній на вході автомата, при цьому зміни вхідного сигналу можуть відбуватися в довільні моменти часу. Інтуїтивне визначення асинхронного автомата означає, що він змінює свій стан тільки після зміни входів.

Виконавши перехід в новий стан, автомат повинен залишатися в цьому стані до наступної зміни сигналу на вході. Такий новий стан автомата називається стійким. Наведений опис дозволяє уточнити поняття асинхронного автомата: «***Автомат, всі стани якого стійкі, називається асинхронним***».

З урахуванням неодночасності перемикання станів реальних елементів автоматної схеми, а також неодночасність зміни станів вхідних сигналів асинхронний автомат іноді може кілька разів змінювати стан, видаючи відповідне число вихідних проміжних сигналів, поки не перейде в стійкий стан. Для виключення такого явища стійкість стану асинхронного автомата забезпечується тим чи іншим способом, наприклад введенням сигналів синхронізації.

Неформально схему асинхронного автомата можна представити у вигляді, показаному на рис. 3.5, де безінерційні логічні елементи зібрані в логічному перетворювачі (ЛП). Тут $x_1(t)$, ..., $x_n(t)$ – послідовність вхідних дискретних сигналів; $y_1(t)$, ..., $y_m(t)$ – послідовність вихідних дискретних сигналів, а $z_1(t)$, ..., $z_k(t)$ – послідовність проміжних вихідних сигналів. Вважається, що чергова зміна вхідних сигналів може статися лише тоді, коли в автоматі закінчиться перехідний процес. Елементи $\tau_1(t)$, ..., $\tau_k(t)$ призначені для затримки вхідних сигналів. У загальному випадку затримка приймає випадкове значення, що обмежене зверху величиною τ_{max}. Це і призводить до змагання різних ланцюгів асинхронного автомата при переході з одного стану в інший, вже стійкий стан.

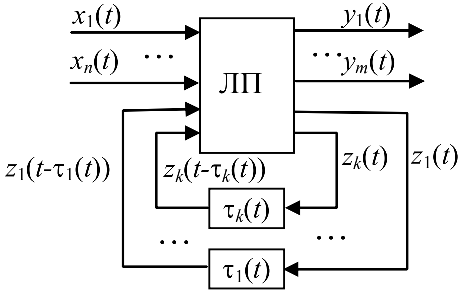


Рис. 3.5

Робота асинхронного автомата, так само як і автомата синхронного типу, може бути описана за допомогою таблиці переходів і виходів або за допомогою графа.

З *кінцевими автоматами* близько межує математичний об'єкт, який широко використовується останнім часом для моделювання дискретних систем, діям компонент яких притаманний паралелізм (обчислювальні системи і мережі, системи керування транспортними процесами, тощо). Цим об'єктом є мережі Петрі.

Мережі Петрі - апарат для моделювання динамічних дискретних систем (переважно асинхронних паралельних процесів). Формально мережі Петрі задається четвіркою $C = <P, T, I, O>$, де P і T – кінцева кількість *позицій і переходів*, I и O – множини вхідних і вихідних функцій.

Мережа Петрі (див. рис. 3.6) являє собою граф, що складається з вершин двох типів – ***позицій*** (позиціям P_i відповідають вершини, що зображені кружками) та ***переходи*** (переходам t_j відповідають вершини, що зображені потовщеними рисками), з'єднаних між собою дугами. Вершини одного типу не можуть бути з'єднані безпосередньо. Іншими словами, мережа Петрі представляє собою дводольний орієнтований граф[3.2].

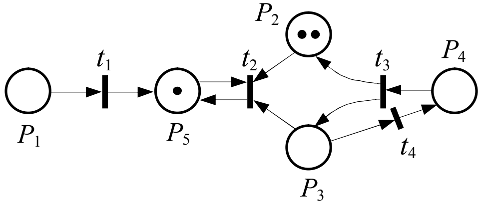


Рис. 3.6

У мережах Петрі вхідним функцій типу I відповідають дуги, що спрямовані від позицій до переходів, а вихідним функцій типу O – дуги від переходів до позицій.

У позиціях можуть розміщуватися ***мітки*** (***маркери***), які здатні переміщатися за мережею. Маркери зображуються точками в кружках. Розподіл маркерів за позиціями називають *маркуванням*. Маркери можуть переміщатися в мережі. Кожному стану мережі Петрі відповідає певна позиція (певний розподіл маркерів за позиціями).

Кожна зміна маркування є *подією*, причому кожна подія пов'язано з певним переходом. Вважається, що події відбуваються миттєво та різночасно при виконанні деяких умов.

[3.2] Граф, множину вершин якого можна розбити на два типи, при цьому не існує ребер, що з'єднують дві вершини одного і того ж типу.

Здійсненню події відповідає *спрацьовування* (збудження або запуск) переходу, при якому маркери з вхідних позицій цього переходу переміщаються у вихідні позиції. Послідовність подій створює модельований процес.

Правила спрацьовування переходів конкретизують таким чином: перехід спрацьовує, якщо для кожної з його вхідних позицій виконується умова $N_i \geq K_i$, де N_i – число маркерів в i-ої вхідної позиції, K_i – число дуг, що йдуть від i-ої позиції до переходу. При спрацьовуванні переходу число маркерів в i-ої вхідній позиції зменшується на K_i, а в j-ої вихідний позиції збільшується на M_j, де M_j – число дуг, що пов'язують перехід з j-ою позицією.

На рис. 3.7 показаний приклад розподілу маркерів за позиціями перед спрацьовуванням, це маркування записують у вигляді (2,2,3,1). Після спрацьовування переходу маркування стає іншим: (1,0,1,4).

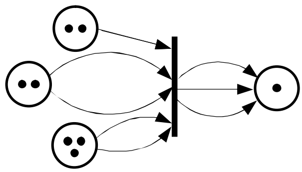


Рис. 3.7

Нарешті, для дослідження складних об'єктів, прикладами яких можуть служити технологічні процеси виконання певних робіт, крім мереж Петрі застосовують моделі систем масового обслуговування (СМО), які відносяться до категорії моделей динамічних дискретних систем.

Система масового обслуговування – складна система (підприємство, транспортна система, технологічна лінія і т.п.), яка здійснює обслуговування масового потоку заявок (транзактів), які надходять до неї. При цьому потік заявок зазвичай є випадковою величиною. Обслуговування заявок у СМО проводиться обслуговуючими пристроями. Найпростіша схема СМО зображена на рис.

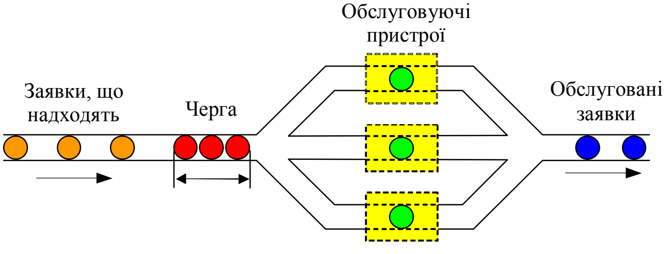


Рис. 3.8

Прикладами застосування теорії масового обслуговування в ЦА можуть служити системи управління повітряним рухом, процеси завантаження, вивантаження і підготовки до рейсів літаків в аеропорту, процеси обслуговування пасажиропотоків і т.п. Основною особливістю подібних систем є випадковий характер не тільки часу обслуговування, але і вхідного, і вихідного потоків.

Якщо СМО описуються послідовністю випадкових подій з кінцевим або зліченим числом наслідків, незалежних від минулого, то їх відносять до Марковських систем.

Аналітичному дослідженню піддаються тільки окремі випадки нелінійних СМО, які виділяються в окремі класи напівмарковських, лінійчатих та інших СМО. При цьому аналітичні моделі СМО вдається отримати при досить серйозних припущених, тому основним методом моделювання СМО є імітаційне.

Існує велика кількість моделей СМО і методів їх класифікації. Зазвичай використовуються такі класифікаційні ознаки: організація потоку заявок, характер створення черги, обмеження черги, кількість обслуговуючих каналів, дисципліна черги (рис. 3.9).

Рис. 3.9

Перш за все, СМО поділяються на одноканальні і багатоканальні залежно від числа приладів обслуговування, які можуть одночасно обслуговувати вхідні заявки.

Залежно від характеру утворення черги СМО можна розділити на два типи: системи з відмовами в обслуговуванні і системи з очікуванням обслуговування, тобто з чергою.

СМО з відмовами це такі системи масового обслуговування, в яких система відмовляє вимоги, яка надійшла в момент, коли всі обслуговуючі апарати зайняті, і тоді заявка йде зі СМО не обслуговуваної. Таким чином повністю відсутні умови для створення черги.

СМО з очікуванням - це такі системи, в яких заявка, яка надійшла, буде знаходитися в системі до тих пір, поки не закінчиться її обслуговування. СМО з очікуванням має велику кількість різновидів, які визначаються обмеженнями на чергу і різними законами очікування в черзі (дисципліною черги).

Особливістю СМО з очікуванням є характеристика самої черги (обмеження на чергу). Такі системи підрозділяються на СМО з необмеженим очікуванням (необмеженою кількістю місць в черзі) і СМО з обмеженим очікуванням (обмеженою кількістю місць у черзі). Перший варіант обмеження полягає в тому, що заявка завжди ставиться у чергу і колись буде обслужена. Другий варіант для постановки заявки в чергу вимагає наявності вільних місць в ній. І тільки в цьому випадку заявка ставиться в чергу. Але можливий і інший вид обмеження – за часом очікування заявкою початку обслуговування. В цьому випадку заявка залишає систему не опрацьованою, якщо тривалість її можливого перебування в черзі перевищує заданий ресурс часу очікування (це так звані «нетерплячі» заявки).

За дисципліною черги (дисципліною обслуговування), тобто за критерієм вибору з черги чергової заявки, СМО поділяються на системи з пріоритетом і без пріоритету. У першому випадку заявці, яка надійшла в СМО, присвоюється певний ***пріоритет*** (переважне право на обслуговування), відповідно до якого на обслуговування відправляється заявка з найвищим пріоритетом, незважаючи на те, що, можливо, вона прийшла в систему після приходу інших заявок.

При реалізації дисципліни черги «без пріоритету» можливі різні правила відбору заявок на обслуговування: «перший прийшов - перший обслужений» (дисципліна **FIFO**), «останній прийшов - перший обслужений» (дисципліна **LIFO**) або у випадковому порядку (наприклад, із заданими ймовірностями вибірки з черги).

У моделях СМО зазвичай виділяються елементарні складові: джерела вхідних потоків заявок, пристрої, накопичувачі, вузли.

Джерело вхідного потоку заявок це алгоритм, відповідно до якого обчислюються моменти появи заявок на виході джерела. Типовим незалежним джерелом є алгоритм формування значень випадкової величини із заданим законом розподілу.

Пристрої в імітаційній моделі представлені алгоритмами формування значень інтервалів (тривалостей) обслуговування. Найчастіше це алгоритми генерації значень випадкових величин із заданим законом розподілу. Але можуть бути пристрої з детермінованим часом обслуговування. Модель пристрою відображає також задану дисципліну обслуговування, оскільки в модель входить алгоритм, керуючий чергами на входах пристрою.

Накопичувачі моделюються алгоритмами визначення обсягів пам'яті, що займаються заявками, які надходять на вхід накопичувача. Зазвичай обсяг пам'яті, займаний заявкою, обчислюється як значення випадкової величини, закон і (або) числові характеристики розподілу можуть залежати від типу заявки.

Вузли виконують зв'язувальні, керуючі і допоміжні функції в імітаційній моделі, наприклад, для вибору напрямків руху заявок, зміни їх параметрів і пріоритетів, поділу заявок на частини, їх об'єднання, тощо.

Зазвичай кожному типу елементарної моделі СМО, відповідає певна процедура (підпрограма). Тоді повну модель СМО можна представити як алгоритм, який складається з упорядкованих звернень до цих процедур і відображає поведінку модельованої системи.

Розглянуті вище моделі систем на основі цифрових автоматів, мереж Петрі, а також моделі СМО належать до класу моделей систем з дискретними подіями, які використовуються при дискретно-подієвому моделюванні.

Зміни змінних стану[3.3] (процеси, що протікають в системах, представляються за допомогою змінних стану) при описі дискретно-подієвих процесів (в даному випадку подій) носять якісний характер і відбуваються миттєво в *певні* дискретні моменти часу, які називаються моментами здійснення подій. У разі ж безперервного процесу зміни мають *безперервний* характер і в якості змінних стану використовуються дійсні числа. У загальному випадку поведінка динамічних систем з безперервним часом описується рівняннями стану (системами диференціальних або різницевих рівнянь разом з рівняннями для вихідних величин).

$$\dot{x} = F(x, u, t), \quad y = G(x, u, t).$$

Функції F і G як функції змінних стану, часу t і вхідного (керуючого) сигналу u в загальному випадку є нелінійними і нестаціонарними. Рішення диференціального рівняння це і є поведінка динамічної системи, яку воно описує (або траєкторія системи в фазовому просторі). Безперервні моделі описують системи, в яких можливі появи континууму можливих станів.

При переході до дискретного подання часу і, відповідно, до дискретного подання змінних стану поведінку динамічних систем описуються дискретними аналогами моделей неперервних систем. Поведінка такої системи (траєкторія системи в фазовому просторі) описується послідовністю станів.

Класичним прикладом дискретних автоматичних систем є системи, що використовують в контурі управління цифрові регулятори. Моделювання такої системи, природно, повинно проводитися з використанням дискретної моделі. При моделюванні ж, наприклад, аналогових систем автоматичного управління можуть використовуватися як безперервні, так і дискретні моделі цієї системи. Зазначимо, що при моделюванні на цифрових комп'ютерах все одно відбувається перехід до дискретного подання часу через використання чисельного інтегрування.

Виходячи з категорії *властивості*, математичні моделі класифікуються як лінійні і нелінійні; стаціонарні і нестаціонарні; зосереджені і розподілені; детерміновані і недетерміновані, останні в свою чергу можна поділити на стохастичні (ймовірні) і нечіткі.

[3.3] Стан - це сукупність величин, які визначають (разом з вхідним впливом) майбутню поведінку системи.

3.2.3. Лінійні та нелінійні моделі

Лінійні моделі це моделі систем, властивості яких не змінюються при зміні їх стану, тобто параметри лінійної системи, які характеризують її властивості, не залежать від величин, що характеризують стан системи. Таке визначення охоплює як статичні, так і динамічні моделі.

Динаміка поведінки лінійних систем описується моделлю у вигляді лінійних диференціальних рівнянь: $\dot{x} = Ax + Bu, \quad y = Cx$ (до речі, звідси і пішла їх назва).

У лінійній системі виконується принцип суперпозиції[3.4] (накладення), наприклад, при додаванні входів системи виходи також складаються, а при множенні входу на будь-яке число вихід множиться на те ж число.

- Необхідними умовами лінійності системи є:
- гомогенність – при зміні амплітуди вхідного сигналу в k разів також в k разів змінюється амплітуда вихідного сигналу.
- адитивність – при підсумовуванні вхідних сигналів результуючий сигнал на виході буде дорівнювати сумі реакцій від вихідних сигналів.
- інваріантність – коли зсув вхідного сигналу в часі викликає аналогічне зміщення вихідного сигналу.
- статична лінійність – коли основні закони в системі описуються лінійними рівняннями.
- гармонійна вірність – якщо на вхід системи подати синусоїдальний сигнал, то на виході буде сигнал тієї ж частоти.

Лінійні моделі привабливі через більш доступний і добре відпрацьований математичний апарат.

Нелінійна модель це математична модель, що відображає стан і функціонування нелінійної системи таким чином, що деякі взаємозв'язки і елементи в ній приймаються нелінійними. У нелінійній системі не виконується принцип суперпозиції.

Властивості і характеристики нелінійних систем зазвичай залежать від їх стану.

[3.4] Суперпозиція функції – це функція від функції. Пусть $y = f(u)$, $u = g(x)$, тоді $y = f(g(x))$. Наприклад, $y = \mathrm{arctg}\,(x^2)$ або $y = \mathrm{arctg}\,(u)$, $u = x^2$.

Деякі види ланок, які обумовлюють нелінійність системи, наведені нижче:

- ланка релейного типу;
- ланка із зоною нечутливості;
- ланка з криволінійної характеристикою будь-якого поєднання;
- ланка, рівняння якої містить добуток змінних або їх похідних, або інші їх комбінації;
- нелінійна ланка з запізненням;
- імпульсна ланка;
- логічна ланка.

Зазвичай, реальні об'єкти і процеси мають в тій чи іншій мірі нелінійний характер, але в багатьох випадках виявляється можливим усвідомлено знехтувати нелінійними властивостями для того, наприклад, щоб скористатися добре розробленим математичним апаратом дослідження лінійних моделей для отримання попередніх результатів. Однак робити це потрібно обережно, об'єктивно оцінюючи похибки і обґрунтовуючи можливість такого спрощення.

Так, наприклад, будь-які фізичні датчики інформації мають зону нечутливості – типову нелінійність, яка враховує те, що при дуже малих сигналах на вході навіть самий чутливий вимірювальний прилад на виході показує «нуль», що означає відсутність вхідного сигналу. Все залежить від величини цієї зони нечутливості: в деяких випадках вона настільки мала, що їй можна знехтувати, і тоді модель переходить вже в категорію лінійних моделей.

Класичний перехід від нелінійної моделі типу $\dot{x} = F(x,u)$ до лінійної реалізується з використанням методу лінеаризації диференціальних рівнянь. В процесі лінеаризації передбачається, що процеси в системі протікають, мало відхиляючись від деякої опорної траєкторії $\{x_0(t), u_0(t)\}$, яка задовольняє рівнянню

$$\dot{x}_0 = F(x_0, u_0).$$

Тоді можна записати наближену *лінеаризовану* модель у відхиленнях від цього режиму:

$$\Delta\dot{x} = A(t)\Delta x + B(t)\Delta u, \tag{3.2}$$

де $\Delta x = x - x_0$; $\Delta u = u - u_0$; $A(t) = \frac{\partial F}{\partial x}(x_0(t), u_0(t)) + \frac{\partial F}{\partial u}(x_0(t), u_0(t))$.

3.2.4. Моделі стаціонарних і нестаціонарних систем

Якщо розрахунковий режим є сталим, тобто не залежить від часу, то коефіцієнти в (3.2) також не залежать від часу: $A(t) \equiv A$, $B(t) \equiv B$ и т. д. Такі системи називаються ***стаціонарними***. Для стаціонарної системи її вихідний сигнал, як реакція на будь-який заданий вхідний сигнал, однаковий для будь-якого моменту прикладення вхідного сигналу. Це означає, що в моделі стаціонарної системи параметри не змінюються в часі, в іншому випадку, система є ***нестаціонарної***. Відповідно в нестаціонарній моделі параметри моделі змінюються в часі за певним законом.

Особливо часто на практиці зустрічаються *стаціонарні лінійні безперервні системи*, описувані простими рівняннями

$$\dot{x} = Ax + Bu, \quad y = Cx. \tag{3.3}$$

Матриці A, B, C є параметрами моделі (3.3).

Наведене вище стосується і рівнянь ***дискретних за часом систем***. Рівняння дискретної системи у загальному випадку мають вигляд

$$x_{k+1} = F(x_k, u_k), \;\; y_k = G(x_k, u_k).$$

Дискретним аналогом рівнянь лінійної стаціонарної системи (3.3) є рівняння:

$$x_{k+1} = Px_k + Qu_k, \;\; y_k = Rx_k.$$

Поряд з рівняннями стану широке застосування знаходять також моделі, що формулюються в змінних «вхід - вихід». Для безперервного часу лінійна модель «вхід-вихід» одноканальної динамічної системи може бути представлена звичайним диференціальним рівнянням виду:

$$a_0 y^{(n)} + a_1 y^{(n-1)} + \ldots + a_{n-1}\dot{y} + a_n y = b_0 u^{(m)} + b_1 u^{(m-1)} + \ldots + b_m u.$$

Дана модель може бути переписана в операторній формі.

$$A(p)y(t) = B(p)u(t), \tag{3.4}$$

де $p = d/dt$ – оператор диференціювання;

$A(p) = a_0 p^n + a_1 p^{n-1} + \ldots + a_{n-1} p + a_n$;

$B(p) = b_0 p^m + b_1 p^{m-1} \ldots b_{m-1} p + b_m$,

причому в (3.4) завжди $m < n$.

Лінійні моделі за допомогою знову ж лінійних перетворень можна трансформувати в інші лінійні моделі. Так, наприклад, еквівалентною характеристикою системи, описуваної моделлю у вигляді лінійного диференціального рівняння, є ***передаточна функція***.

За визначенням передатна функція $W(p)$ – це відношення зображення вихідної величини $Y(p)$ до зображення вхідної величини $U(p)$ при нульових початкових умовах. Передаточна функція для безперервної системи, описуваної в вигляді диференціального рівняння (3.4), має вигляд

$$W(p) = \frac{Y(p)}{U(p)} = \frac{\sum_{i=0}^{m} b_i p^i}{\sum_{k=0}^{n} a_k p^k}. \qquad (3.5)$$

Покажемо такого роду трансформацію на прикладі найпростішого рівняння: $2\dot{y}(t) + 4y(t) = 5x(t)$. Застосувавши до нього перетворення Лапласа, отримаємо:

$$2py(p)+4y(p)=5x(p),$$

Виносимо за дужки $y(p)$

$$(2p+4)y(p)=5x(p)$$

З урахуванням (3.5) отримаємо:

$$W(p) = \frac{y(p)}{x(p)} = \frac{5}{2p+4} = \frac{1{,}25}{0{,}5p+1}.$$

Таким чином, вихідна система, яка описується рівнянням $2\dot{y}(t) + 4y(t) = 5x(t)$, трансформується в передаточну функцію аперіодичної ланки з коефіцієнтом посилення 1,25 і сталою часу, що дорівнює 0,5.

Зворотний перехід від опису у вигляді передаточних функцій до описів у вигляді диференціальних рівнянь неоднозначний. На практиці застосовуються кілька типових способів переходу від передаточної функції до рівнянь стану (рівнянням «вхід - вихід»). Ці способи базуються на використанні структурних схем, на зворотних перетвореннях Лапласа і на так званих канонічних поданнях системи.

3.2.5. Моделі зосереджених і розподілених систем

Одна і та ж динамічна система в різних умовах може розглядатися або як зосереджена, або як розподілена.

Якщо параметр, що описує властивість системи, в будь-яких її точках має однакове значення (хоча і може змінюватися в часі), то це система з **зосередженими** параметрами.

Математичні моделі **зосереджених** систем описуються у вигляді диференціальних рівнянь

$$\sum_{k=0}^{n} a_k \frac{d^k y(t)}{dt^k} = \sum_{i=0}^{m} b_i \frac{d^i u(t)}{dt^i},$$

або у вигляді кінцевих, різницевих рівнянь

$$\sum_{k=0}^{n} a_k y(n-k) = \sum_{i=0}^{m} b_i u(n-i),$$

де для фізично реалізованих систем виконується умова $m \le n$.

Під кінцевими рівняннями розуміють алгебраїчні і трансцендентні рівняння (рівняння, що містять показові, логарифмічні, тригонометричні або зворотні тригонометричні функції).

Якщо параметр, що описує властивість об'єкта, приймає різні значення в різних точках об'єкта, то кажуть, що він розподілений, а модель, що описує об'єкт, - ***розподіленою***.

Математичні моделі ***розподілених*** систем - це диференціальні рівняння в частинних похідних, інтегральні рівняння або звичайні рівняння із запізнілим аргументом (рівняння із запізненням). Тут змінні стану в кожен момент часу є функції однієї або кількох змінних.

Підсумовуючи вищевикладене, зазначимо, що, як правило, реальні об'єкти і процеси мають в тій чи іншій мірі нелінійний характер, але в багатьох випадках виявляється можливим усвідомлено знехтувати нелінійними властивостями, для того щоб скористатися простим математичним апаратом дослідження лінійних систем. Динамічні лінійні стаціонарні системи з зосередженими параметрами описуються звичайними лінійними диференціальними рівняннями з постійними коефіцієнтами. Для таких моделей гарантовано отримання аналітичного рішення (наприклад,теорема Пікара про достатні умови існування та єдності розв'язку задачі Коші).

Отже, системи, початково нелінійні, але лінеаризовані у межах опорних траєкторій, описуються за допомогою лінійних рівнянь, які є спрощеними у порівнянні з нелінійними. Однак при спрощення можуть не враховуватися важливі, цікаві, тонкі ефекти, пов'язані з проявом нелінійних властивостей. Тому, нехтуючи нелінійними властивостями системи, потрібно об'єктивно обґрунтовувати можливість такого спрощення.

3.2.6. Детерміновані і недетерміновані моделі

Ще один підхід до класифікації абстрактних моделей поділяє їх на детерміновані і недетерміновані.

У ***детермінованих моделях*** вхідні параметри піддаються вимірюванню однозначно і з будь-яким ступенем точності, тобто є детермінованими величинами. Відповідно, процес еволюції такої системи детермінований. Детерміновані моделі розглядають «точну» поведінку системи-замінника на заданому відрізку часу. Знаючи стан моделі в певний момент t_0 і значення вхідних сигналів в інтервалі $[t_0, t_s]$, можна точно визначити її стан в момент t_s.

Однак на практиці так буває рідко: опису реальних систем зазвичай властива невизначеність. Тому такі системи описуються ***недетермінованими моделями***.

Серед різних способів формалізації невизначеності найбільшого поширення набув стохастичний (імовірнісний) підхід, при якому невизначені величини вважаються випадковими. Тому недетерміновані моделі також називають ***стохастичними*** (імовірнісними) моделями. У таких моделях значення вхідних параметрів відомі лише з певним ступенем імовірності, тому процес еволюції системи є випадковим.

Для статичної моделі (3.1) невизначеність можна врахувати співвідношенням:

$$Y(t)=F(u(t))+ \varphi(t),$$

де $\varphi(t)$ – випадкова величина, яка в результаті досліду може прийняти те чи інше значення, причому невідомо заздалегідь, яке саме. Наприклад, похибка або завада вимірювань входів і виходів системи.

Узагальненням лінійної динамічної моделі (3.3) є імовірнісна модель, яка описується стохастичним диференціальним рівнянням:

$$\dot{x} = A(x) + B(u) + \varpi(t); \quad y = C(x),$$

де – $\varpi(t)$ являє собою деякий випадковий процес[3.5], котрий може розглядатися як вхідний вплив, тому x і y – також випадкові процеси.

Стохастичні диференціальні рівняння такого роду дозволяють отримати закони розподілу[3.6] і числові характеристики виходу (математичне очікування і дисперсію), якщо задані закони розподілу вхідних сигналів. Такі дані дуже важливі при дослідженні статистичної динаміки безперервних систем.

Серед типових математичних схем дискретних систем, які використовуються при дослідженні недетермінірованних систем ЦА широке застосування знайшли *імовірнісні* автомати, які моделюють системи, що функціонують у випадковому середовищі або знаходяться під впливом випадкових збурень.

Імовірнісні автомати формально описуються структурою

$$< X, Y, Z, P(z'y \mid z, x) >,$$

де X, Y и Z – відповідно кінцеві множини вхідних сигналів, вихідних сигналів і станів;

$P(z'y \mid z, x)$ – умовна імовірність[3.7] того, що імовірнісний автомат, отримавши на вході символ x і перебуваючи в стані z, перейде в новий стан z' і видасть на виході символ y.

Такий автомат являє собою дискретний стаціонарний перетворювач послідовності вхідних символів з пам'яттю, робота якого

[3.5] Випадковим називається процес, миттєві значення якого є випадковими величинами, залежними від деякого параметра, наприклад, часу або координати.

[3.6] Законом розподілу випадкової величини називається співвідношення, яке встановлює зв'язок між можливими значеннями випадкової величини і відповідними їм ймовірностями

[3.7] Законом розподілу випадкової величини називається співвідношення, яке встановлює зв'язок між можливими значеннями випадкової величини і відповідними їм ймовірностями в кожному такті залежить від стану його пам'яті і може бути описана статистично. У першому такті має бути зафіксовано початковий стан автомата або ж заданий початковий розподіл $P_0(z)$ ймовірностей станів.

Для моделювання стохастичних динамічних дискретних систем можуть використовуватися *стохастична мережа Петрі*, в якій переходам надається певна вага – тривалість (затримка) спрацьовування.

Стохастична мережа Петрі заснована на концепції стохастичних часових затримок. Переходам в стохастичній мережі Петрі зіставляють умовні імовірності їх спрацьовування або параметри розподілу затримок спрацьовування. При цьому ймовірності змін маркувань виявляються розподіленими, відповідно, за дискретної або безперервної часовою шкалою. Так, на рис. 3.10 представлений фрагмент мережі Петрі, який ілюструє конфліктну ситуацію – маркер в позиції P може запустити або перехід t_1, або перехід t_2. У стохастичній мережі передбачається імовірнісний вибір переходу, який має спрацювати в таких ситуаціях.

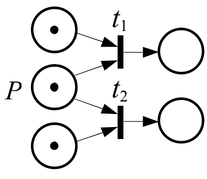


Рис. 3.10

Однак найчастіше при стохастичному підході до моделювання динамічних дискретних систем в якості типових математичних схем застосовуються *моделі систем масового обслуговування*.

При цьому характерними для роботи таких моделей є джерело випадкового потоку вхідного заявок на обслуговування і алгоритм генерації значень випадкових величин інтервалів обслуговування, які і визначають стохастичний характер процесу моделювання.

Розвинений понятійний і обчислювальний апарат теорії ймовірностей дозволяє дати конкретні рекомендації щодо вибору структури моделі системи і оцінці її параметрів.

Висновки і рекомендації засновані на ефекті усереднення: випадкові відхилення результатів вимірювань деякої величини від її очікуваного значення при підсумовуванні взаємно знищуються, і середнє арифметичне великого числа вимірювань виявляється близьким до очікуваного значення. Математичні формулювання цього ефекту даються *законом великих, чисел і центральною граничною* *теоремою.* Закон великих чисел гласить: якщо $\xi_1,\ldots,\xi_N$ – випадкові величини з математичним очікуванням $M_{\xi 1} = a$ і дисперсією $M(\xi_i - a)^2 = \sigma^2$, то

$$\frac{1}{N}(\xi_1 + \ldots + \xi_N) - a \approx 0 \qquad (3.6)$$

при достатньо великих N. Це говорить про принципову можливість точної оцінки M_{ξ_i} за проведеними вимірюваннями. Центральна гранична теорема, яка уточнює (3.6), стверджує, що

$$\frac{1}{N}(\xi_1 + \ldots + \xi_N) - a \approx \frac{\sigma}{\sqrt{N}}\xi\,, \qquad (3.7)$$

де ξ – стандартна нормально розподілена випадкова величина ($M_\xi = 0$, $M \cdot \xi^2 = 1$).

Оскільки розподіл величини ξ добре відомий і табульований, то співвідношення (3.7) дозволяє обчислювати похибки оцінки.

Зрозуміло, що формулюванням (3.6), (3.7) можна надати більш строгий вигляд, і це легко може бути зроблено за допомогою понять ймовірнісної збіжності. Труднощі виникають при спробі перевірити умови цих суворих стверджень. Наприклад, в законі великих чисел і в центральній граничній теоремі потрібні незалежність окремих вимірювань (реалізацій) випадкової величини і скінченність її дисперсії. Якщо ці умови порушуються, то можуть порушуватися і висновки. Наприклад, якщо всі виміри збігаються: $\xi_1 = \ldots = \xi_N$ то, хоча всі інші умови виконуються, про усереднення не може бути й мови.

Стохастичний підхід заздалегідь не можна застосовувати до одиничних експериментів. Але навіть якщо масовість і повторюваність експериментів є, статистичної стійкості може і не бути, а перевірити це – непроста справа. Відомі оцінки допустимого відхилення частоти від ймовірності засновані на центральній граничній теоремі або нерівності Чебишева і вимагають додаткових гіпотез про незалежність або про слабку залежність вимірювань. Дослідна ж перевірка умови незалежності ще складніша, оскільки вимагає додаткових експериментів.

Альтернативою стохастичного підходу для дослідження задач з невизначеністю став підхід, заснований на теорії нечітких множин.

Нечіткі моделі. У 1965 році американський математик індійського походження Л. Заде опублікував статтю під назвою «Fuzzy sets», що можна перекласти як «Нечіткі множини». У статті дано визначення понять нечітких множин і підмножин, призначене для опису систем, на поведінку яких зокрема, істотний вплив роблять знання, судження і емоції людини. Нечіткі множини пов'язані з розпливчастістю, невизначеністю і суб'єктивністю. При цьому класичне поняття множини розширюється за рахунок введення поняття *функції приналежності*.

Функція приналежності $\mu(x)$ (див. рис. 3.11) кількісно гра-

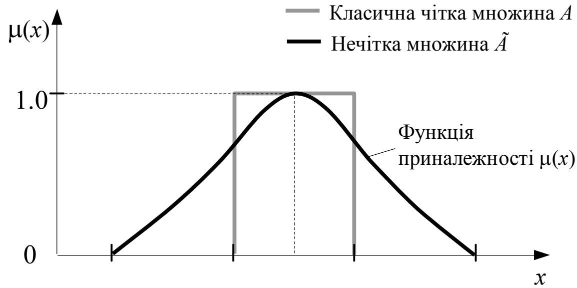


Рис. 3.11

дуює приналежність елементів фундаментальної множини простору розмірковувань $x \in \mathbf{X}$ нечіткої множині $\tilde{A}$. Значення 0 означає, що елемент не включений в нечітку множину, 1 описує повністю включений елемент. Значення між 0 і 1 характеризують нечітко включені елементи.

Функції приналежності елемента множині може приймати будь-які значення в інтервалі [0, 1], а не тільки значення 0 або 1, тобто функція приналежності вказує, в якій мірі елемент x належить нечіткій множині A

Для простору розміркувань $\mathbf{X}$ і даної функції приналежності μ: нечітка множина визначається як

$$\tilde{A} = \{(x, \mu_A(x)) \mid x \in \mathbf{X}\}.$$

Оскільки поняття множини лежить в основі всіх математичних конструкцій, стаття Л. Заде породила новий науковий напрям. Сталося «роздвоєння» математики: з'явилися нечіткі функції, нечіткі відносини, нечіткі рівняння, нечітка логіка.

Предметом нечіткої логіки, зокрема, є побудова нечітких моделей наближених міркувань людини, які є різновидом недетермінованих моделей. Найчастіше такі моделі застосовується в експертних системах, нейронних мережах і системах штучного інтелекту. Замість традиційних значень *Істина* і *Хибність* в нечіткій логіці використовується більш широкий діапазон значень, серед яких такі: *Істина, Хибність*, *Можливо*, *Іноді*, *Не пам'ятаю*, *Як би Так*, *Чому б Ні*, *Ще не вирішив*, *Не скажу* ...

В даний час теорія нечітких систем - це єдина теорія, яка математично оперує зі смисловим змістом слів людини. Можна навести приклади людино-машинних систем, які призначені для обробки нечітких знань. У технічних областях – автоматичне керування, інтелектуальні роботи, системи підтримки цілісності баз даних і системи забезпечення безпеки, розпізнавання зображень і мови та ін. В сфері бізнесу – допомога в прийнятті економічних рішень, маркетинг, поради щодо вкладення капіталу, допомога в підготовці контрактів, тощо. Крім того, оцінка стану навколишнього середовища, прогноз землетрусів, прогнози погоди і т. п.

Підсумовуючи викладену вище класифікацію математичних моделей, відзначимо, що для дослідження процесів і систем ЦА застосовуються в основному *аналітичні моделі*, що описуються звичайними диференціальними рівняннями (безперервними або дискретними), рідше рівняннями в частинних похідних. А для моделювання систем з випадковими вхідними впливами використовуються стохастичні диференціальні рівняння.

Іншим поширеним типом моделей реальних систем ЦА є *імітаційні моделі* з дискретними подіями, які відбуваються в ізольованих точках часової осі. Формалізація таких моделей проводиться в термінах теорії систем масового обслуговування, мереж Петрі, цифрових автоматів.

Як приклад систем, описуваних аналітичними моделями, можна привести модель системи автоматичного управління польотом, причому для вивчення її статистичних характеристик може використовуватися стохастична модель руху літального апарату. Моделі систем з дискретними подіями добре описують функціонування обчислювальних систем, систем управління повітряним рухом, виробничі процеси в цивільній авіації, процеси обслуговування пасажиропотоків і багато інших.

Контрольні запитання

1. У яких формах може бути наданий формалізований опис математичної моделі?
2. Що означає адекватність математичної моделі?
3. На які дві великі групи поділяється вся безліч математичних моделей?
4. У якому вигляді аналітичні моделі описують процеси функціонування складних систем?
5. Сформулюйте поняття імітаційної моделі.
6. Що є основною відмінністю імітаційних моделей від аналітичних?
7. При яких дослідженнях досягається найбільший ефект застосування імітаційних моделей?
8. Чому імітаційне моделювання зазвичай прирівнюється до статистичного експерименту?
9. Які фактори можна віднести до переваг, а які до недоліків імітаційного моделювання в порівнянні з аналітичним?
10. Наведіть класифікацію математичних моделей за Шеноном.
11. Яка математична модель системи називається статичною?
12. Яка математична модель системи називається динамічною?
13. За якими параметрами система може бути дискретною або безперервною?
14. Наведіть приклади моделей систем, які використовуються при дискретно-подієвому моделюванні.
15. У якому вигляді описується динаміка поведінки лінійних систем?
16. Який принцип не виконується в нелінійної системі?
17. Які системи називаються стаціонарними?
18. Якими математичними моделями описують розподілені системи?
19. Що означає термін детермінована модель?
20. Наведіть приклади імовірнісних моделей.
21. Що означає поняття «функція приналежності» в нечітких моделях?

Глава 4. ОГЛЯД МЕТОДІВ МОДЕЛЮВАННЯ

4.1. Аналітичне моделювання

Аналітичне моделювання - це таке моделювання, при якому вивчаються аналітичні (чисельно-математичні) моделі реального об'єкта у вигляді явних функціональних залежностей, а також передбачається здійснення однозначної обчислювальної процедури, що призводить до їх точного розв'язання. Експеримент, проведений методом аналітичного моделювання, нічим не відрізняється від традиційної для математики процедури обчислень. І скільки б разів ми не повторювали цю процедуру, результат завжди буде один і той же (при незмінних значеннях параметрів). Основою аналітичного моделювання є опис роботи системи у вигляді одного або декількох рівнянь. При цьому динаміка системи враховується при складанні диференціальних рівнянь, а не реалізується у вигляді послідовностей операцій (станів або процесів).

Понятійний (термінологічний) апарат аналітичних методів моделювання складають поняття класичної математики, такі як величина, формула, функція, рівняння, система рівнянь і т.п.

В основі аналітичних методів моделювання лежать математичні теорії різної складності, починаючи від апарату класичного математичного аналізу і закінчуючи такими новими розділами сучасної математики, як *математичне програмування, теорія ігор* і т.д. В їх число входять основи шкільної математики: *арифметика, алгебра, геометрія,* а також дисципліни, що розвиваються на їх базі: *лінійна алгебра, аналітична геометрія, інтегро-диференціальне числення, регресійний і функціональний аналізи* та інші напрямки, що формують предмети вузівської математики.

При моделюванні важливим моментом є первісна математична постановка задачі. Вона передбачає опис математичної моделі і зазначення цілі її дослідження. Залежно від цілей дослідження при аналітичному моделюванні може бути використаний різний математичний апарат. Для однієї і тієї ж математичної моделі можуть бути сформульовані і розв’язані різні математичні задачі.

Так, наприклад, в деяких задачах фізики безпосередній зв'язок між величинами, що описують процес, встановити не вдається. Але існує можливість отримати рівності, що містять похідні досліджуваних функцій – диференціальні рівняння, розв‘язок яких дозволяє отримати характеристики процесу (системи) як деякі функції параметрів її функціонування.

Відзначимо деякі області застосування диференціальних рівнянь при аналітичному моделюванні. Це моделювання систем автоматичного керування; розрахунок руху ракет, супутників і небесних тіл; розрахунок струмів в складних електричних ланцюгах; моделювання динаміки механічних і фізичних процесів в техніці; дослідження окремих питань в біології. Диференціальні рівняння також широко використовуються в моделях економічної динаміки.

Для розв‘язання апроксимаційних задач при аналітичному моделюванні широко використовуються математичні методи інтерполяції, екстраполяції, чисельного диференціювання та інтегрування.

Якщо ціллю моделювання є пошук найкращих способів управління і планування при заданих цілях і критеріях, то як математичний апарат можуть використовуватися математичні методи оптимізації систем і процесів, в основі яких лежать методи варіаційного числення, які засновані на знаходженні екстремумів функціоналів. Методи варіаційного числення широко застосовуються в різних областях математики, фізики, механіки, техніки.

Для теорії систем і системного аналізу найбільший інтерес представляє напрямок варіаційного обчислення, зазвичай коротко званий математичним програмуванням, який отримав особливий розвиток у другій половині XX століття і увійшов складовою частиною в сучасну теорію оптимальних процесів.

Аналітична модель може бути досліджена такими методами:

1) аналітичним, коли прагнуть отримати в загальному вигляді явні залежності для характеристик систем;

2) чисельним, коли не вдається знайти розв‘язок рівнянь в загальному вигляді тому їх вирішують для конкретних вихідних даних;

На практиці аналітичне моделювання нелінійної системи з використанням чисельних методів виявляється набагато більш ефективним, ніж з використанням аналітичних методів моделювання окремих частинних лінійних ланцюгів і пристроїв цієї системи. Так за даними чисельного моделювання можна отримати досить повні дані про поведінку модельованих систем і пристроїв, а також побудувати графіки, що описують цю поведінку.

Чисельні методи - це методи наближеного розв'язання задач прикладної математики, засновані на реалізації алгоритмів, які зводять все обчислення до послідовності арифметичних і логічних дій. В даний час розроблено цілий ряд алгоритмів, реалізації чисельних методів, що ідеально підходять для реалізації на цифрових ЕОМ, які "вміють" виконувати тільки арифметичні дії і логічні операції. Наприклад, алгоритми: чисельного інтегрування, чисельного розв'язання звичайних диференціальних рівнянь, чисельного розв'язання рівнянь в частинних похідних, інтерполяції і наближеного обчислення функцій і т. д.

У цивільній авіації аналітичне моделювання є невід'ємним етапом технологічного процесу розробки і випробувань складних авіаційних систем типу пілотажно-навігаційних комплексів (ПНК).

Ефективним способом випробувань ПНК, які доповнює льотні випробування і сприяють підвищенню їх результативності, є наземне моделювання. Воно полягає у відтворенні засобами обчислювальної техніки і автоматики процесів, ідентичних тим, що мають місце в ПНК в льотних умовах. У зіставленні з льотними випробуваннями перевага моделювання зумовлено тим, що воно вільно від ряду обмежень і труднощів, властивих льотному експерименту, таких, наприклад, як необхідність дотримання умов безпеки польотів, проблематичність повторного відтворення ситуації, а також меншими матеріальними і часовими витратами, що припадають на одиничний експеримент. Істотно більше можливостей у моделювання щодо залучення необхідного персоналу, за повнотою і доступності аналізу результатів, з отримання великих обсягів експериментальних даних.

Моделювання дозволяє також отримувати дані про можливу поведінку ПНК в режимах, які не допускаються в льотних випробуваннях з міркувань безпеки, у зовнішніх умовах, які зазвичай рідко відбуваються, в важкодоступних районах. Ці дані за умов збіжності (подібності) моделювання і льотних випробувань здатні доповнити результати льотних випробувань при вирішенні задач оцінювання характеристик ПНК заданим вимогам.

У більшості випадків при аналітичному моделюванні не потрібно відтворення інформаційних процесів в реальному масштабі часу. Темп їх може бути прискореним, реальним, уповільненим, тобто довільним.

Предметом вивчення методом аналітичного моделювання є переважно питання динаміки автоматичного управління рухом, точності витримування параметрів в заданих межах, логічної правильності здійснення основних функцій ПНК, реалізованих за допомогою схемних алгоритмічних засобів. Залежно від того, яка з функцій випробуваного ПНК відпрацьовується за допомогою аналітичного моделювання, змінюються вимоги до змісту математичної моделі досліджуваного ПНК, ступеня її повноти і детальності. Однак, в загальному випадку перелік окремих моделей, що становлять шукану модель, включає:

- модель руху літака;
- моделі зовнішніх геофізичних умов польоту;
- моделі бортових інформаційних систем в їх взаємодії з наземними засобами і зовнішнім середовищем;
- моделі бортових обчислювачів навігаційних, інформаційних, пілотажних комплексів;
- моделі виконавчих пристроїв системи автоматичного управління.

Взаємодія цих окремих моделей показано на структурній схемі (рис. 4.1).

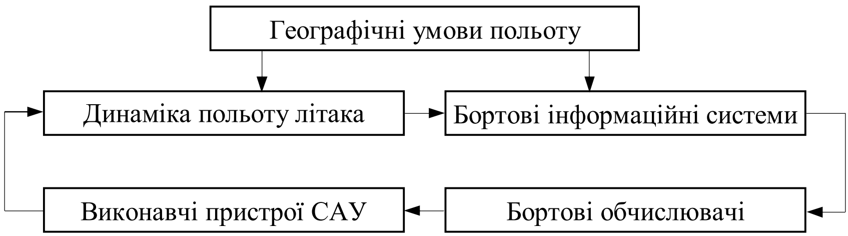


Рис. 4.1

При формуванні математичних моделей раціонально мати обмежений ряд окремих математичних моделей однієї і тієї ж системи від докладної до спрощеної, застосовуючи їх в залежності від специфіки завдання, і уникаючи при цьому надмірного різноманіття і зайвої універсальності, яка веде до громіздкості та складності.

Центральним для моделювання, особливо для аналітичного, є питання досягнення відповідності (збіжності) результатів моделювання і льотних випробувань. Найважливішим засобом досягнення такої відповідності є з'єднання льотного експерименту і моделювання в єдиний технологічний процес. Результати льотного експерименту обробляються для отримання і уточнення моделей, а матеріали моделювання доповнюють результати льотних випробувань. В процесі льотних випробувань можуть бути уточнені параметри моделі літака, зовнішніх збурень, похибок систем і датчиків. Це досягається застосуванням методів ідентифікації моделей.

Одним з уживаних способів досягнення відповідності аналітичного моделювання льотним випробуванням є використання записів реальних сигналів датчиків, виконаних в процесі польотів.

У цьому випадку в комп'ютері необхідно реалізувати лише модульні алгоритми бортового обчислювача інформаційного (навігаційного) комплексу. Найбільша ефективність при використанні цього способу досягається і випадку застосування на борту пристроїв запису польотної інформації з подальшим введенням цих записів в моделюючу ЦОМ. Зауважимо, що зазначений спосіб може бути застосований тільки для досліджень розімкнутих інформаційних трактів ПНК і не може використовуватися для замкнутих контурів управління.

Аналітичне моделювання із застосуванням записів реальних сигналів є, таким чином, гібридною формою моделювання, пов'язаної з натурним льотним експериментом. Розвиток обчислювальної техніки і засобів телеметричної передачі даних з борту на землю і назад дозволяють в перспективі створювати більш розвинені гібридні форми моделювання, які можуть функціонувати безпосередньо в процесі льотних випробувань.

При такому моделюванні сигнали датчиків, що передаються з борта на землю, в реальному масштабі часу обробляються за допомогою модельних алгоритмів бортового обчислювача, реалізованого в наземної ЦОМ. Результати обробки цих сигналів в придатному для сприйняття оператором вигляді надходять на екран дисплея експериментатора, а також на борт екіпажу, які, взаємодіючи один з одним, керують експериментом. Принципово можливо в цьому випадку і автоматичне відпрацювання бортовими системами сигналів, вироблюваних математичною моделлю, тобто здійснення моделювання замкнутих автоматичних контурів управління.

4.2. Напівнатурне моделювання

Аналітичне моделювання не вичерпує всіх аспектів досліджень ПНК. Більш детальне дослідження внутрішніх і зовнішніх факторів, що впливають на роботу реального ПНК, може здійснюватися за допомогою напівнатурного моделювання. Це моделювання, так само як і аналітичне, будучи наближенням до натурного експерименту (льотних випробувань), володіє великими можливостями в порівнянні з аналітичним. Це пояснюється тим, що більша частина явищ, яка відбувається в реальній апаратурі, має досить складний фізичний механізм, детальне математичне подання якого або невідоме, або надзвичайно складне. Проте напівнатурне моделювання дозволяє спостерігати й експериментально вивчати ці явища в їх безпосередньому вигляді внаслідок реального функціонування ПНК або його частин у взаємодії з математичними і фізичними моделями літака та його систем, відтворюваних засобами обчислювальної та імітаційної техніки.

Завдяки цьому напівнатурні моделювання дозволяє вирішувати завдання:

- оперативної наземної перевірки технічних доводочних заходів при перевірках ПНК в льотних випробуваннях;
- аналізу ситуацій, що виникають в льотних випробуваннях, здійснення діагностики різних відхилень в роботі ПНК;
- відпрацювання програм і методик льотних випробувань, апаратурних і програмних засобів реєстрації та обробки результатів випробувань;
- освоєння екіпажами та обслуговуючим персоналом прийомів поводження з апаратурою і її органами управління;
- оцінки ПНК в умовах впливу завад в лініях зв'язку, в колах електропостачання;
- відпрацювання ПНК в умовах впливу відмов систем, оцінювання способів їх парирування;
- попередньої ергономічної оцінки ПНК.

Задачі, які вирішувалися за допомогою аналітичного моделювання, напівнатурні модель дозволяють повторити з більшим ступенем подібності льотному експерименту. Це відбувається завдяки прояві динамічних властивостей та похибок реальної апаратури, яка включається в напівнатурні моделі.

Реалізація напівнатурні моделі ПНК може мати багато різновидів. Однак при будь-якому технічному втіленні напівнатурна модель включає такі структурні елементи:

- обчислювально-моделюючу систему на базі комп‘ютера;
- пристрої для зв'язку математичних моделей з реальною апаратурою ПНК, яка застосовується в схемі моделювання;
- реальну апаратуру ПНК разом з пристроями, які забезпечують її розміщення, за діяння та нормальне функціонування;
- фізичні моделі низки систем ПНК і літакових систем, сполучених з ПНК;
- пристрої керування й контролю роботи напівнатурної моделі.

Розглянемо загальний технічний вигляд зазначених структурних елементів і принципи їх взаємодії для найбільш уживаних різновидів напівнатурних моделей. Ці різновиди відрізняються повнотою включення реальної апаратури ПНК в схему моделювання. Найбільш вживаними є напівнатурні моделі навігаційних, інформаційних, пілотажних і пілотажно-навігаційних (в цілому) комплексів.

Для напівнатурні моделі навігаційного (інформаційного) комплексу характерно застосування в якості обчислювально-моделюючої системи цифрової обчислювальної машини (ЦОМ) або комп‘ютера. Це пояснюється великим обсягом, високою складністю і точністю виконуваних операцій, характерних для навігаційних завдань. Функціями ЦОМ у складі напівнатурні моделі ПНК є:

- реалізація в реальному масштабі часу математичних моделей;
- накопичення й обробка результатів функціонування напівнатурної моделі і подання їх експериментатору;
- забезпечення автоматизації ряду функцій підготовки експерименту;
- управління пристроями зв'язку моделей з реальною апаратурою ПНК.

Ці функції здійснюються програмними й апаратними засобами. Основними апаратними засобами є дисплей, пристрій друку, аналого-цифрові перетворювачі. Програмні засоби поряд з загальносистемними програмами включають окремі прикладні програми:

- формування вихідних даних, що характеризують трасу і геофізичні умови польоту;
- моделювання в реальному масштабі часу руху літака і відпо-

відних йому навігаційних (інформаційних) параметрів, які повинні вимірюватися навігаційними системами і деякими іншими бортовими датчиками;
- накопичення даних, що формуються в ЦОМ або вводяться ззовні, оперативна обробка цих даних з видачею результатів на зовнішні пристрої відображення в формі, зручної для аналізу;
- тестовий контроль обладнання, що сполучається з ЦОМ.

Реальна апаратура навігаційних (інформаційних) комплексів є основним об'єктом випробувань при напівнатурному моделюванні. Для забезпечення нормального її функціонування в складі напівнатурної моделі використовується система електропостачання, що імітує бортову резервовану систему електропостачання, та система обдування, що відтворює бортову систему повітряного охолодження апаратурних блоків. Для розміщення випробовується апаратури передбачаються спеціальні етажерки та стенди. До складу апаратури, яка підлягає випробуванню і включається в напівнатурну модель, можуть входити:

- автономні навігаційні системи інерціально-гіроскопічного та магнітометричного принципу роботи;
- радіотехнічні навігаційні системи та системи посадки, радіотехнічні вимірювачі висоти та шляхової швидкості;
- вимірювачі аерометричних висотно-швидкісних параметрів (системи повітряних сигналів);
- бортові обчислювачі навігаційних та інформаційних комплексів з пультами та індикаторами.

Залежно від типу ПНК даний перелік може коригуватися.

Зв'язок реального обладнання з математичними моделями, реалізованими в ЦОМ, здійснюється за допомогою керованих стимуляторів і імітаторів. Стимулятори є пристроями, що формують сигнали фізичної природи, які відповідають:

а) природі вхідного сигналу, що сприймається приймальним пристроєм або чутливим датчиком (тип "А");

б) природі проміжного сигналу, що сприймається перетворювачем або обчислювальним пристроєм, наступним за приймальним пристроєм (тип "Б").

Імітатори є пристроями, що формують електричні сигнали, відповідні вихідним сигналам систем і датчиків. Так, стимулятор типу "А" для системи повітряних сигналів (СПС) є пристроєм, що складається з двох керованих генераторів статичного і повного тиску, які підключаються до пнемовходів СПС. Стимулятор типу "Б" для тієї ж системи є принципово іншим пристроєм, який перетворює сигнали ЦОМ в електричні сигнали, які відповідні сигналам, що знімаються з датчиків тиску. Натомість імітатор СПС формує електричні вихідні сигнали СПС без безпосереднього використання реальної апаратури.

Для радіотехнічних систем можуть бути застосовані як стимулятори обох типів так й імітатори. Для інерціальної навігаційної системи (ІНС) або інерціальної курсовертикалі можна здійснювати стимуляцію її роботи за типом "А" тільки для функцій вимірювань кутів орієнтації: курсу, крену, тангажу. Для цього гіроблок розміщують на керованому триступеневому стенді, який відслідковує орієнтацію модельованого літака. При відтворенні обертання за курсом для забезпечення методичної точності повинна враховуватися поправка за кутовою швидкістю, що дорівнює різниці проекцій на вертикаль кутової швидкості обертання Землі в місці установки стенду і в місці імітованого польоту.

Стимуляція за типом "А" іншої функції ІНС, а саме визначення швидкості і координат, вимагає відтворення руху, тобто відтворення показів акселерометрів, а отже, неможлива в лабораторних умовах. Стимуляція за типом "Б" – відтворенням електричного вихідного сигналу акселерометра на вході обчислювача є ідеєю, яка досить просто може бути реалізована. Для повноти відтворення особливостей роботи, в цьому випадку потрібно передавати інформації про сигнали, які виробляються в обчислювачі ІНС, в моделюючу ЦОМ на моментні датчики моделі гіроплатформа, де повинна формуватися акселерометрична інформація. У найпростішому варіанті може застосовуватися імітація вихідних сигналів ІНС як за швидкістю та координатам, так і за кутами.

Стимуляція вимірника магнітного курсу - індукційного датчика (ІД) здійснюється з однаковим результатом або обертанням датчика в магнітному полі Землі, або обертанням спеціально створеного магнітного поля відносно датчика. Імітатором вихідних сигналів ІД можуть служити, як і для багатьох інших сучасних навігаційних систем, перетворювачі кодів розрахункових значень синуса і косинуса вихідного параметра (магнітного курсу для ІД) в змінні напруги, сукупність яких еквівалентна напругам, що знімаються з обмоток синусно-косинусних трансформаторів. Для більшого наближення до натури в якості опорної напруги для цих перетворювачів рекомендується брати сигнал однієї з вимірювальних обмоток ІД, нерухомо закріпленої у напрямку магнітного меридіана. Завдяки залученню при побудові напівнатурної моделі ПНК навігаційних систем (за допомогою стимуляторів або імітації їх вихідних сигналів) створюються умови для функціонування бортових обчислювачів, сигналізаторів та індикаторів у схемі напівнатурні моделі.

Ряд суміжних з ПНК бортових систем, таких як система аварійної сигналізації, можуть бути представлені фізичними моделями, що відтворюють функції модельованих систем, але не ідентичні їм.

Пристрої управління та контролю експерименту при напівнатурні моделюванні мають багато спільного, як з льотними випробуваннями, так і з аналітичним моделюванням. Як і в льотних випробуваннях при напівнатурні моделюванні застосовуються різного роду пристрої реєстрації. Структурна схема взаємодії описаних раніше елементів напівнатурної модель представлена на рис. 4.2.

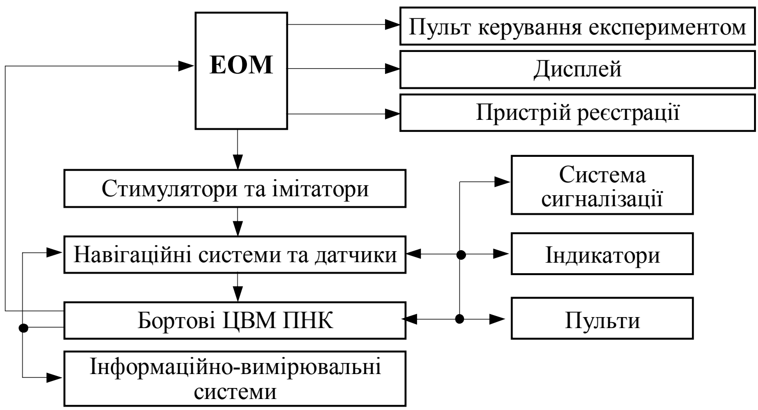


Рис. 4.2

Просторовий і швидкісний профілі модельованого польоту можуть змінюватися експериментатором за допомогою спеціальних органів управління. Однак на відміну від льотних випробувань так само, як і при аналітичному моделюванні, "політ" може бути перерваний в будь-якій точці і повторений заново. У процесі моделювання ЦОМ може накопичувати дані, виконувати в темпі "польоту" їх оперативну обробку і видавати результати обробки на дисплей експериментатора, а також у разі необхідності на друкувальний пристрій. Підготовка експерименту полягає у "передстартової" підготовки бортової апаратури, яка буде випробуватися, у тестовому контролі устаткування, що сполучається з ЦОМ, а також у введення вихідних даних в ЕОМ.

Напівнатурна модель пілотажного комплексу, яка часто іменується пілотажним стендом, в більшій мірі, ніж модель навігаційного комплексу, вимагає відтворення особливостей взаємодії з екіпажем та суміжними літаковими системами. Тому пілотажний стенд будується на основі фізичних моделей кабіни, органів і механізмів керування літаком. У багатьох випадках пілотажний стенд близький за побудовою до комплексного тренажера або навіть організовується на базі готового тренажера.

Структурна схема пілотажного стенда, що ілюструє взаємодію його основних елементів, приведена на рис. 4.3.

Для реалізації математичних моделей динаміки руху літака, двигунів і вимірників пілотажних параметрів застосовують ЦОМ (комп'ютер). Пристроями спряження в пілотажних стендах є стандартні пристрої: цифроаналогові перетворювачі, сельсин-трансформаторні перетворювачі, системи стеження, операційні підсилювачі, тощо. З їх допомогою здійснюється електричне стикування виходів моделі зі входами системи автоматичного управління (САУ) і приладами, а також виходів САУ і органів штурвального управління зі входами моделі. У процесі переходу від аналогових САУ до цифрових можуть трансформуватися і пристрої сполучення, перетворюючись в пристрої перетворення кодових сигналів.

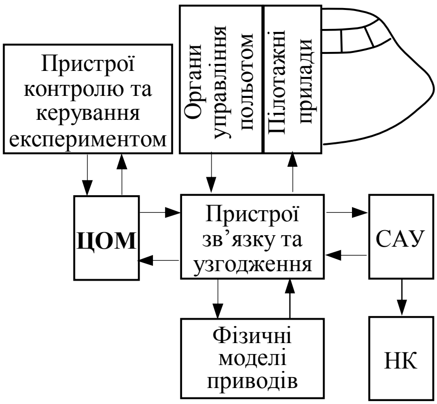


Рис. 4.3

4.3. Імітаційне моделювання

Зародження методу імітаційного моделювання пов'язане з роботою фон Неймана і Улама, в 1940-і роки вони застосували його для розв’язання завдань екранування ядерних випромінювань. Основний принцип методу імітаційного моделювання (його часто називають методом Монте-Карло, або методом статистичного моделювання) складається з побудови штучного імовірнісного процесу, параметри якого давали б рішення поставленої задачі, причому сама задача може і не бути ймовірнісної. Як характерні приклади можна привести статистичні моделі експлуатації за станом авіатехнічних систем, зміна параметрів яких описується випадковими процесами, а також моделі інформаційних систем ЦА, описувані в термінах теорії масового обслуговування.

Імітаційне моделювання - це окремий випадок математичного моделювання. Існує клас об'єктів, для яких з різних причин не розроблені аналітичні моделі, або не розроблені методи розв’язку отриманої моделі. В цьому випадку аналітична модель замінюється імітаційної – логіко-математичним описом реальної системи, що відображає процеси, які відбуваються в системі в часі при різних поєднаннях значень параметрів системи та зовнішнього середовища. Імітаційну модель можна розглядати як множину правил (автоматів, мереж, таблиць станів, і т.п.), які визначають в який стан система перейде в майбутньому з заданого поточного стану. Імітація у цьому випадку - це процес "виконання" моделі, який проводить її через (дискретні або безперервні) зміни стану в часі.

Таку модель можна «програти» в часі як для одного випробування, так і для заданої множини випробувань. При цьому результати будуть визначатися випадковим характером процесів. За цими даними можна отримати достатньо стійку статистику. Тому імітаційне моделювання зазвичай прирівнюється до статистичного експерименту, в якому результати експерименту обробляються з використанням методів математичної статистики. Як в будь-якому статистичному експерименті при імітаційному моделюванні важливу роль відіграє не тільки проведення, але і планування експерименту.

Імітаційне моделювання, як експериментування з моделлю реальної системи, є одним з найбільш широко поширених кількісних методів, використовуваних при розв’язання задач проектування системи, її оптимізації, прогнозування, тренажі т. д.

Особливістю методу імітаційного моделювання є можливість опису і відтворення взаємодії між різними елементами системи зі збереженням логічної структури системи, а також зі збереженням її поведінкових властивостей, тобто описати динаміку змін станів системи. Тому побудова імітаційної моделі полягає в описі структури та процесів функціонування модельованого об'єкта або системи. Щоб скласти імітаційну модель, необхідно:

• розробити статичний опис системи, тобто подати реальну систему як сукупність взаємодіючих елементів;

• алгоритмічно описати функціонування окремих елементів системи;

• розробити динамічний опис системи, тобто описати динаміку зміни станів системи (послідовність чергування в часі подій, що відбуваються в системі).

Логічна структура реальної системи відображається в моделі деякими програмними компонентами елементів системи, а стан цих елементів описується за допомогою змінних стану.

Функціонування окремих елементів системи описується моделюючим алгоритмом взаємодії елементів системи (обміну інформацією між елементами) і алгоритмом зміни змінних станів.

Ключовим моментом в імітаційному моделюванні є опис *динаміки зміни станів* системи. Система характеризується набором змінних станів, кожна комбінація яких описує конкретний стан системи. Отже, шляхом зміни значень цих змінних можна імітувати моменти здійснення подій (перехід системи з одного стану в інший). Таким чином, імітаційне моделювання - це опис *динамічної поведінки* системи шляхом просування її від одного стану до іншого за певними правилами. Динаміка (чергування в часі подій) в імітаційних моделях реалізується за допомогою *механізму просування модельного часу*.

Часто *імітаційний* підхід протиставляється *аналітичному*, в основі якого лежать аналітичні моделі реального об'єкта у вигляді явних функціональних залежностей, а моделювання передбачає здійснення обчислювальної процедури, яка призводить до їх точного розв'язання. Динаміка ж системи при аналітичному моделюванні враховується при складанні цих функціональних залежностей, наприклад, систем диференціальних рівнянь, а не реалізується у вигляді послідовностей здійснення в часі подій (операцій, що відбуваються в системі) як це реалізується при імітаційному моделюванні.. Статистичний експеримент при аналітичному моделюванні, на відміну від імітаційного моделювання являє собою підстановку деякої стохастичної вибірки вихідних даних у формули і обчислення результатів.

Підсумовуючи вище викладене, визначимо імітаційне моделювання як спосіб опису процесу функціонування системи, при якому необхідне покрокове моделювання, оскільки відсутні явно задані критерії оцінки стану та траєкторії розвитку модельованої системи.

Як варіант класифікації можна виділити такі види імітаційного моделювання (див. рис 4.4):

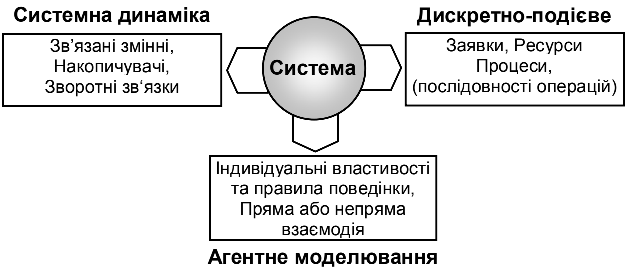


Рис. 4.4

Дискретно-подієве моделювання – підхід до моделювання, що пропонує абстрагуватися від безперервної природи подій і розглядати тільки основні події, що моделюється. При використанні дискретно-подієвого підходу динаміка системи задається за допомогою множини подій, які упорядковуються за часом виникнення. Наприклад, для опису роботи сканера можна вказати чотири події (рис. 4.5.): *укладка оригіналу* (e_1)*, початок сканування* (e_2), *закінчення сканування* (e_3), *видалення оригіналу* (e_4).

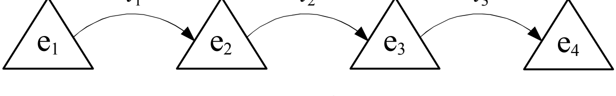


Рис. 4.5

Термін "дискретно-подієве моделювання" історично закріпився за моделюванням СМО потоків об'єктів деякої природи: літаків в зоні управління повітряним рухом, клієнтів банку, автомобілів на заправній станції, пасажирів в аеропортах і т.п. Перераховані вище системи не вичерпують усього різноманіття СМО. Наприклад, конвеєрні системи для поточного виробництва і збирання виробів також можуть розглядатися як СМО, але вони вимагають при аналізі врахування характеристик самих конвеєрів (наприклад, їх форми, швидкості) і алгоритмів збирання виробів. Крім того, великий клас систем включає такі процеси обслуговування, які вимагають для окремих операцій виконання специфічних умов, наприклад, наявність ресурсів конкретного типу.

У більш розширених класифікаціях дискретних видів імітаційного моделювання, крім подієво-орієнтованого моделювання виділяють процесно-орієнтовані, транзактно-орієнтовані і дія-орієнтовані види моделювання. У більш ранніх класифікаціях виділяли також агрегатні види моделювання, дія-орієнтовані види називали системами сканування активностей, а подієво-орієнтовані види моделювання – системами з розкладом подій.

Перевагою подієвого підходу є його швидкодія, яке досягається за рахунок можливості "стрибкоподібної" зміни модельного часу. Наприклад, при моделюванні комп'ютерної мережі можна знехтувати конкретними характеристиками комп'ютерів і здійснювати тільки облік їх завантаженості.

Деякі автори вважають, що дана парадигма моделювання, насправді є єдиним представником імітаційного моделювання як такого. При цьому більшість робіт з імітаційного моделювання присвячені викладанню виключно цього стилю моделювання.

Найбільш популярною подієвою мовою є мережі Петрі та їх модифікації (Е-мережі, Р-мережі). Даний підхід має серйозне і добре відпрацьоване математичне обґрунтування.

Системна динаміка - це вид імітаційного моделювання для вивчення комплексних систем, які з часом піддаються змінам. При такому моделюванні враховується глобальний вплив одних параметрів на інші в часі і причинно-наслідкові зв'язки між елементами системи, при цьому особлива увага приділяється зворотного зв'язку між цими елементами.

В основі системної динаміки лежить твердження, що структура системи обумовлює її поведінку. Тобто після того, як модель побудована, можна здійснити симуляцію числових показників елементів системи, яка у вигляді графіка або таблиці покаже поведінку системи.

Моделі в системній динаміці можна поділити на якісні і кількісні (з можливістю симуляції числових показників).

Концептуальна модель показує структуру проблеми і то, яким чином один елемент системи залежить від іншого. Існує дві форми взаємозв'язку між двома змінними: позитивна і негативна. *Менше знаєш, краще спиш* – приклад негативної полярності. *Далі в спір - більше слів* - приклад позитивної полярності.

Деякі взаємозв'язки між змінними містять зворотний зв'язок і утворюють так звані «петлі». Петлі бувають двох видів: підсилююча і стабілізуюча.

Приклад підсилюючої петлі: «*Чим більше яєць, тим більше курей вилупиться. Чим більше курей вилупиться, тим більше яєць вони знесуть*».

Приклад стабілізуючою петлі: «*Чим більше травоїдних, тим менше рослин для їх харчування залишиться на території; Чим менше рослин, тим менше буде кількість травоїдних*».

Щоб модель могла генерувати графік поведінки системи в часі, вона повинна бути кількісної і включати в себе статистичні дані. У кількісної моделі системної динаміки є свої елементи:

Резервуар - елемент, який акумулюється або зменшується з плином часу*;*

Потік - елемент, який впливає на стан резервуара (зменшує, збільшує або залишає незмінним)*;*

Змінна - елемент, який змінюється миттєво (не акумулюється);

З'єднання - елемент, який вказує взаємозв'язки між резервуарами, потоками і змінними.

У кількісної моделі системної динаміки найбільш важливим поняттям є поняття акумулювання. Класичний приклад поняття акумулювання – це басейн з двома трубами, через які вода заливається і виливається з басейна. Вода у басейні може або спадати або прибувати, проте у будь-якому випадку потрібен час оскільки це не миттєвий процес. Іншим важливим моментом, який стосується кількісної моделі системної динаміки, є часова затримка між процесами.

Таким чином, системна динаміка як метод дозволяє врахувати основні взаємозв'язки між елементами системи та часові затримки в динаміці її розвитку, що допомагає більш структурно дослідити існуючи комплексні проблеми і шляхи їх вирішення.

Такий вид моделювання найкраще допомагає зрозуміти сутність проблеми і виявити причинно-наслідкові зв'язки між об'єктами і явищами. За допомогою системної динаміки будують моделі бізнес-процесів, моделі розвитку міста, моделі виробництва, тощо.

Агентне моделювання – відносно новий напрямок в імітаційному моделюванні, яке використовується для дослідження децентралізованих систем, динаміка функціонування яких визначається не глобальними правилами і законами (як в інших парадигмах моделювання), а навпаки, коли ці глобальні правила і закони є результатом індивідуальної активності окремих об'єктів системи. Ціль агентних моделей – отримати уявлення про ці глобальні правила і загалом про поведінку системи, виходячи з припущень про індивідуальну поведінку її окремих активних об'єктів і про взаємодію цих об'єктів в системі. Агент – деяка сутність, що володіє активністю, автономною поведінкою, може приймати рішення відповідно до деякого набору правил, взаємодіяти з оточенням, а також самостійно змінюватися. Таким чином, агентне моделювання реалізує моделювання знизу до верху.

Розглянемо просту систему обслуговування в термінах агентного моделювання (рис. 4.6), де об'єкти (заявки, транзакції, і т.д.) входять в систему, обслуговуються (для цього потрібний ресурс) один або кілька разів, залежності від їх властивостей, затримуються на деякий час і виходять. Ці об'єкти і є агентами. Подія генерації чергового об'єкта (заявки) відповідає створенню нового агента. Після створення агент запитує обслуговування, але

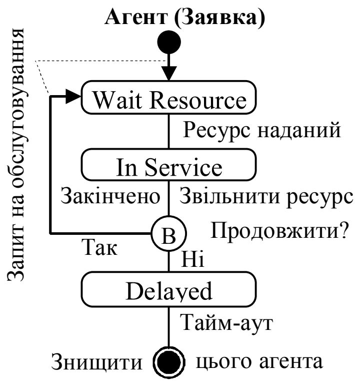


Рис.4.6

не обов'язково відразу ж його отримує, так що у агента є стан очікування – Wait Resource (відповідає заявці, що стоїть в черзі в блоці Service). Коли ресурс наданий, агент переходить у стан обслуговування – In Service (відповідає заявці, що отримала ресурс і яка затримується в тому ж блоці Service). Після закінчення обробки агент вирішує: повторити процедуру або перейти в стан затримки – Delayed. Після виходу з Delayed агент знищує себе, що відповідає заявці, яка виходить з потокової діаграми. Ресурси також можна моделювати агентами якщо в цьому є сенс (наприклад, ресурси це оператори, персонал з будь-якою індивідуальною поведінкою). У простому прикладі ресурси можуть мати два стани: вільний (Idle) і зайнятий (Busy). Для координації доступу до ресурсів необхідний якийсь механізм, реалізований або централізовано (диспетчер), або децентралізовано через взаємодію агентів.

Розглянемо більш ускладнену постановку задачі. Нехай в будь-який момент під час знаходження об'єкта в системі може статися щось, що змусить його негайно покинути систему незалежно від стадії обробки (див. рис. 4.7). Це може бути телефонний дзвінок, серцевий напад, сигнал тривоги і т.п. В агентний моделі це дуже легко врахувати: на карті станів агента формується складовий стан (Normal), що охоплює всі стани і відповідає нормальному режиму. З нього визначається перехід в новий стан – стан Emergency Process, який відповідне процесу "аварійного покидання" системи. Перехід буде спрацьовувати за особливою подією при цьому, залишаючи нормальний процес, можливо, доведеться звільнити захоплені ресурси. При дискретно-подієвому підході таку поведінку моделювати досить складно, доведеться вставляти спеціальний код в усі блоки, а в деяких інструментах це взагалі неможливо зробити.

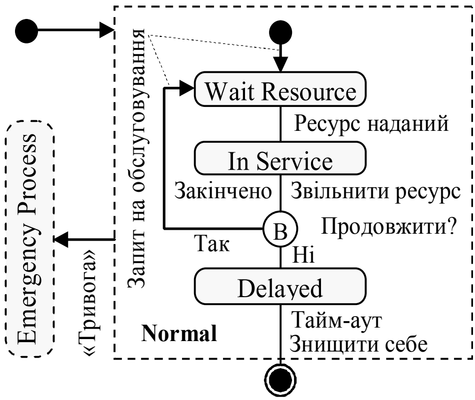


Рис. 4.7

4.4. Експертне і ситуаційне моделювання

Експертне і ситуаційне моделювання, за суттю, є імітаційними, але деякі фахівці виділяють їх в окремий клас методів моделювання.

4.4.1. Системи експертного моделювання

Експертне моделювання пов'язане з дослідженням успішних стратегій людської поведінки в самих різних областях людської діяльності: спорт, бізнес, творчість або вивчення іноземних мов. Причому зміст експертного моделювання, що імітує процеси міркування людини, полягає не тільки в тому, щоб зрозуміти, як бачить, чує, відчуває і діє людина, особливо успішна в якійсь галузі діяльності (експерт), а й змоделювати його поведінку, щоб інші люди змогли досягти найбільш успішних результатів в обраній ними галузі.

За своєю суттю, експертне моделювання - це один з ключових моментів методу нейролінгвістичного програмування[4.1], який виник з моделювання людської поведінки і процесів мислення і вже отримав широку популярність в психології.

Вихідними даними для систем експертного моделювання є декларативні і процедурні знання, тому їх також називають системами заснованими на знаннях або ***експертними системами***.

В експертних системах на відміну від систем імітаційного моделювання використовують стратегії вибору правил, формалізовані цілі і інші специфічні критерії. Проте, при моделюванні знань експерта, які представляють собою вербальне або графічне відображення системи, її зв'язків та закономірностей, експертне подання аналогічно імітаційному.

Основними моделями при експертному моделюванні (в експертних системах) є:

- Фрейм - структура даних для представлення стереотипної ситуації;

[4.1] Нейролінгвістичне програмування - це область знань, яка вивчає структуру суб'єктивного досвіду людей і займається розробкою покрокових стратегій та прийомів передачі даного досвіду іншим людям, а також для застосування цього досвіду в інших контекстах.

- Семантична мережа – конструкція двох основних компонентів: вузлів і дуг. Вузли моделюють поняття предметної області, дуги – відносини між парою понять.
- Логічні моделі. Основними компоненти даної моделі є: обробка фактів, правила і ствердження. Мовою логіки предикатів ствердження можуть бути представлені: правилами, які використовуються для доказу, і правилами висновок.

Структура та класифікація експертних систем

Експертна система **-** це комп'ютерна система, в яку включені знання фахівців про деяку проблемну область та яка в межах цієї області забезпечує автоматичну реалізацію експертних рішень.

Структура експертних систем (див. рис. 4.8) може включати в себе такі компоненти: база знань; база даних; вирішувач проблем (механізм висновку); інтерфейс користувача; редактор бази даних (модуль видобування знань та навчання); підсистема пояснень.

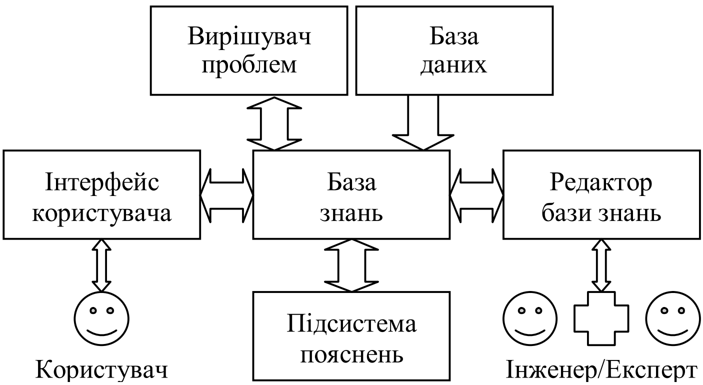


Рис. 4.8

Розглянемо вміст цих блоків.

«***Дані***» це неструктурований набір чисел, фактів і символів, отриманих з фактографічних джерел (бібліографічні відомості, повнотекстові документи, ін.), або самі джерела, підготовлені до розміщення та зберігання в пам'яті комп'ютера.

«***Знання***» це «зовнішня» до джерела даних інформація, яка призначена для кращого розуміння даних. інформація, щоб ця програма поводила себе «інтелектуально». «***Знання***» зазвичай трактуються як «*поняття високого рівня абстракції*».

База знань (БЗ) являє собою модель оцінок експерта або групи експертів, наданих у вигляді особливих правил, які мають логічну форму: **«якщо ...... то»**. Природно, що при створенні БЗ використовуються знання провідних фахівців в тій чи іншій проблемній області. При цьому в більшості експертних систем набір правил може бути розширений користувачем. Зазвичай, БЗ містить факти (статичні відомості про предметну область, різні довідники, енциклопедії, тезауруси) і правила – набір інструкцій, застосовуючи які до відомих фактів можна отримувати нові факти.

БЗ не тільки надають засоби для формалізації поводження з даними, але і дозволяють адаптувати знання до інших джерел і застосовувати їх до нових ситуацій.

Бази даних (БД) відрізняються від баз знань характерним рівнем абстракції, а також рівнем організації розміщених в них інформаційних масивів. Будь-яка сукупність філософських текстів може бути оброблена на комп'ютері як один безперервний, «повний» текст або як колекція інформації, яка трансформується в БД відповідно до певної заздалегідь заданої моделі знань.

Вирішувач проблем (ВП) забезпечує отримання коректних, науково значущих висновків на основі БЗ і наявного емпіричного матеріалу. Експертної системи без ВП не існує.

Суттєвим поняттям ідеології експертних систем є поняття «***механізму висновку***». Одні ВП роблять висновки на основі дедуктивної логіки, інші спираються на теорію нечітких множин, засоби розпізнавання образів, тощо. В цілому, суть конструкцій механізму висновку добре висловив Е. Файгенбаум[4.2]: «Власне це не висновок, а виявлення і видача знання».

Інтерфейс користувача призначений для перетворення інформації на мові предметної області у внутрішню інформацію системи. Інтерфейс користувача забезпечує взаємозв'язок користувача з експертною системою.

[4.2] Вчений у галузі теорії обчислювальних систем (США Стенфордський університет)

Редактор бази знань (компонент придбання знань) призначений для наповнення експертної системи знаннями за участю експерта з предметної області та інженера по знанням.

Мало отримувати рішення, потрібно представити його у такому вигляді, в якому це рішення буде зрозуміло користувачеві. За розв‘язок цієї проблеми відповідає **підсистема пояснення рішень**, яка дає відповідь на наступні питання: Як отримане рішення? Чому експортна система просить ту чи іншу інформацію?

Загальноприйнята класифікація експертних систем відсутня, однак найбільш часто експертні системи розрізняють за призначенням, за предметною областю, за методами представлення знань, за складністю і динамічністю. Класифікація експертних систем за динамічністю наведена на рис. 4.9.

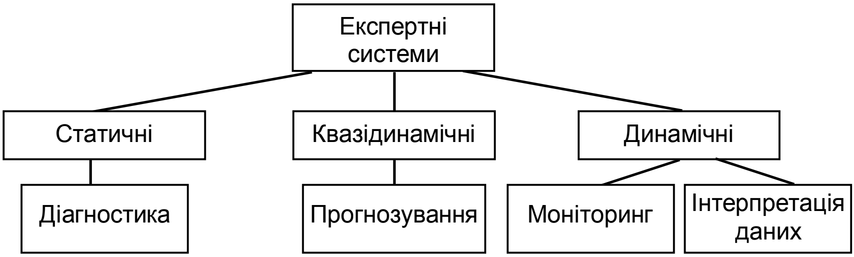


Рис. 4.9

- **Статичні** експертні системи розробляються в предметних областях, в яких база знань і інтерпретовані дані не змінюються в часі, наприклад експертні системи діагностики стану автомобіля.
- **Динамічні** експертні системи, які здатні відтворювати судження з приводу даних зі складною структурою (інтерпретувати дані), а також вирішувати завдання моніторингу (спостереження за станом об'єкта для визначення та передбачення моменту його переходу в граничний стан). У динамічної експертної системі враховуються зміни навколишнього середовища, що відбуваються за час виконання завдання. В архітектуру цієї системи в порівнянні зі статичної експертної системою вводяться два компоненти: підсистема моделювання зовнішнього оточення і підсистема зв'язку із зовнішнім оточенням. Остання здійснює зв'язок із зовнішнім оточенням через систему датчиків і контролерів. Крім того, традиційні компоненти статичної експертної системи (база знань і механізм вистнов-

ку) зазнають суттєвих змін, щоб відобразити часову логіку подій, які відбуваються у реальному світі.

• **Квазідинамічні** експертні системи орієнтовані на підтримку прийняття рішень дослідником. Ці експертні системи забезпечують виявлення альтернатив розв‘язання проблем. Вони можуть бути використані для побудови прогнозів розвитку складних ситуацій.

Області застосування експертних систем можуть бути згруповані в кілька основних класів: медична діагностика, контроль та управління, діагностика несправностей в механічних і електричних пристроях, навчання, тощо.

4.4.2. Ситуаційне моделювання

Ситуаційне моделювання, як і експертне – один з підходів до моделювання, який за своєю суттю, також є гілкою імітаційного моделювання.

Ситуаційне моделювання використовується при моделюванні складних систем (ігрових стратегій, процесів зростання, багатофакторної поведінки і т.д.). Такі об'єкти дослідження не піддаються строгому функціональному опису. Необхідність використання ситуаційного підходу для моделювання складних систем визначається такими властивостями цих систем:

• *Унікальність*. Кожен об'єкт має таку структуру і функціонує так, що система управління ним повинна будуватися з урахуванням всіх його якостей і до нього не можна застосувати будь-яку стандартну типову процедуру управління.

• *Відсутність формалізованої цілі існування*. Не для всіх об'єктів можна чітко сформулювати ціль їх існування.

• *Відсутність оптимальності*. Через відсутність цілі існування (в рамках теорії управління) для розглянутих об'єктів не можна побудувати об'єктивний критерій управління. Критерій управління стає суб'єктивним, цілком залежним від особи, що приймає рішення.

• *Динамічність*. З часом структура і процес функціонування об'єктів змінюються.

• *Неповнота опису*. Зазвичай, колектив експертів, які знають об'єкт управління, не в змозі відразу сформувати таку інформацію, якої б вистачило для створення системи управління об'єктом.

• *Велика розмірність*. Складна система, що характеризується великою розмірністю, що не дозволяє здійснювати її імітаційне моделювання за короткі терміни.

• *Неформалізована інформація*. Часто для прийняття рішення необхідно враховувати слабко формалізовані поняття.

При ситуаційному моделюванні такого роду об'єкти (складні процеси) подаються в категоріях мови ситуацій, основним елементом якої є поняття ситуації, у протилежність складної для прив'язки до конкретних ситуацій мови інтегро-диференціального обчислення.

Ситуацію можна визначати як синонім слова взаємозв'язок. Поняття ситуації також явно ототожнюється з поняттям стану, як поєднання умов та обставин, яке створює певну обстановку, становище. В імітаційному моделюванні під станом розуміється значення всіх характеристик об'єкта в заданий момент часу. У ситуаційному ж моделюванні поняття стану розширено: у нього також включається набір зв'язків між елементами об'єкта та їх значення

Підсумовуючи всі наведені визначення, ситуацію можна сформулювати таким чином: ***Ситуація*** в системі є сукупність характеристик об'єктів і зв'язків між ними, які складаються з постійних і причинно-наслідкових відносин (подій і процесів). Саме характеристики зв'язків визначаю тип ситуації. На рис 4.10. показаний варіант класифікації ситуацій.

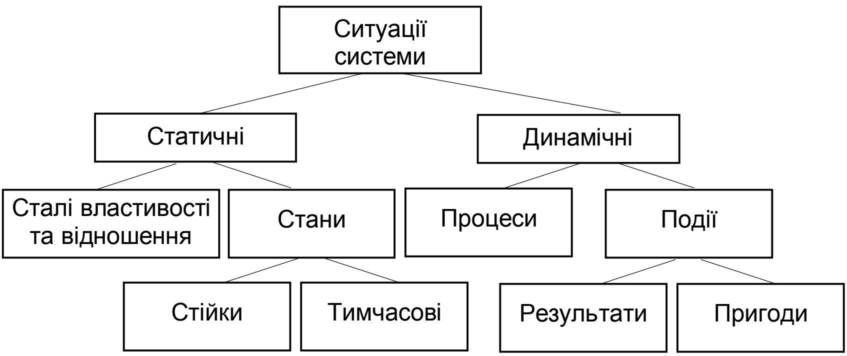


Рис. 4.10

Ситуаційне моделювання спирається на ряд розділів математики (теорія множин, математичні теорії подання нелінійних процесів і теорія катастроф), але ще додатково застосовує математичний апарат варіювання умовами. Останнє доповнення як інструмент моделювання з'явилося завдяки розвитку комп'ютерних програм таких як: об'єктно-орієнтоване програмування, case-технології, графічний інтерфейс та інші засоби візуалізації.

У відповідь на розвиток цих можливостей використовується той або інший теоретичний інструментарій, наприклад, ймовірнісне моделювання, нечітка логіка.

Процедура екстраполяції ситуацій є основою ситуаційного моделювання. Тому свідомо поставлена задача ситуаційного моделювання виводить споживача моделі, наприклад керівника підприємства, в змодельовану "віртуальну реальність", де можна перепробувати різні варіанти розвитку ситуацій. Змодельовані сценарії не обов'язково повинні втілитися у дійсність, але можуть застерегти – і в цьому їх безумовна корисність.

Методи ситуаційного моделювання

Для опису ситуацій використовуються семіотичні (ситуаційні) мови і моделі, серед яких можна виділити такі основні підходи:

• дискретні ситуаційні мережі;
• RX-коди;
• логіку предикатів;
• універсальний семантичний код.

Ситуаційна мережа являє собою складну семантичну (смислову) мережу. Кожна ситуація описується орієнтованим графом, вершини якого відповідають поняттям (ситуація), а дуги задають відносини між ними. За допомогою семантичної мережі можна описати будь-яку складну ситуацію,

RX-коди є мова бінарних відносин. RX-коди мають в якості ядерної конструкції запис такого виду:

$$x_1 = x_2 r_2 x_3,$$

де x_i - об'єкт або ситуація; r_i - відношення.

Універсальний семантичний код використовує в якості ядерної конструкції трійку S-A-O, яка відповідає суб'єкту S, що вчиняє дію A над об'єктом O.

Для реалізації в ЦОМ семіотичних мов використовують мови представлення знань. Найбільш близьким підходом до опису семіотичних конструкцій є семантична мережу. Однак мережі дуже повільні при використанні операцій пошуку, тому конструкції часто представляють за допомогою ***логіки предикатів***.

Методи представлення знань в ситуаційних системах і експертних системах аналогічні. Ще більше вони зблизилися після активного впровадження нечіткої логіки в технології експертних систем. При ситуаційному моделюванні активно використовуються імітаційні моделі, тому, ситуаційна мова включає засоби, що властиві мовам імітаційного моделювання: системний час, черги подій, тощо.

Для ситуаційного моделювання можна використовувати два методи:

• завдання вихідних даних і розрахунок виникаючих ситуацій;

• моделювання взаємозв'язків ситуацій.

Останній метод аналогічний структурному підходу в системах імітаційного моделювання. У ролі вершин мережі виступають ситуації. Якщо застосувати мережі Петрі, то вершинами (позиціями) будуть ситуації, а переходами – події.

Особливо можна виділити методи візуалізації ситуацій. Вони спрямовані на вирішення завдань оптимального відображення інформації на моніторах (сценарні методи, метод абстрактної мапи) і декомпозиції зображень за зрізами ситуаційної моделі.

Спосіб реалізації ситуаційного моделювання

Інструментом ситуаційного управління є ситуаційний центр. Ситуаційний центр це сукупність інтелектуально організованих робочих місць з автоматизованими операціями поповнення інформації (включаючи конвертори даних), процедурами побудови моделей, аналізу ситуації, прогону моделей, графічного подання зіграних сценаріїв. Блок-схема ситуаційного центру приведена на рис 4.11.

За ступенем складності, масштабу і розв'язуваним задачам ситуаційні центри діляться на три типи:

- *стратегічний ситуаційний центр* (складні, масштабні, відповідальні завдання, спрямовані на структурну і функціональну перебудову об‘єкта (або принаймні на стратегічний аналіз і прогноз);

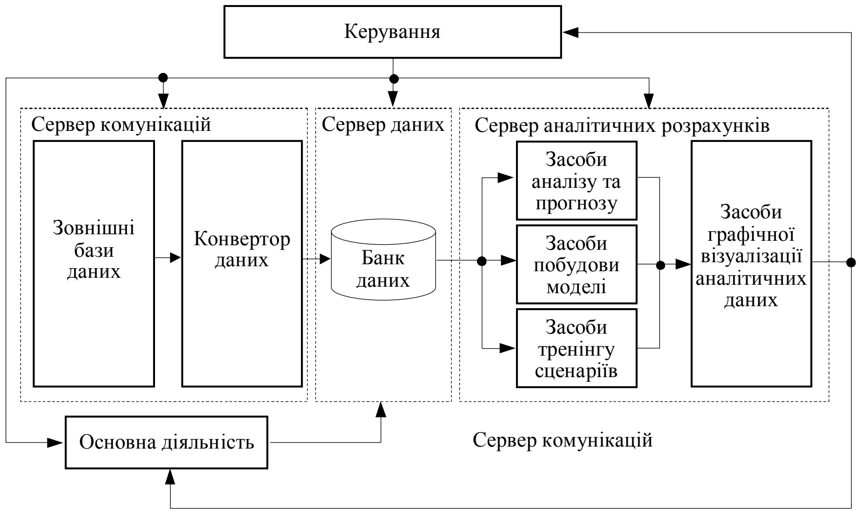


Рис. 4.11

• *оперативний ситуаційний центр*, який вирішує задачі автоматичної згортки оперативної інформації в ситуаційну модель, і надає підприємцю-користувачу можливість оперувати "модулями" свого бізнесу в реальному масштабі часу;

• *персональний ситуаційний центр*, який вирішує задачі експрес-оцінки ситуації, оперативного доступу до керованого об'єкту і який підтримує можливість першого керівника завжди "бути в курсі" незалежно від часу і свого місцезнаходження.

Стратегічні ситуаційні центри налаштовані на об'єкти класу: галузь, регіон, велике підприємство (холдинг), відомство, складний розподілений в просторі процес;

Оперативні ситуаційні центри налаштовані на об'єкти класу: підприємство (компанія), завдання, процес, проект, велика акція.

Персональні ситуаційні центри в певному сенсі індиферентні до масштабів керованого об'єкта, їх завдання, функції та склад визначаються скоріше суб'єктом, який вирішує, яка інформація йому знадобиться для оцінки ситуації при розв‘язанні тієї або іншої проблеми.

З усього різноманіття методів математичного моделювання в подальшому детально розглядатимуться тільки аналітичне та імітаційне моделювання, які на наш погляд є основними методами математичного моделювання.

Контрольні запитання

1. Сформулюйте поняття аналітичного моделювання.
2. Якими методами може бути досліджена аналітична модель?
3. Наведіть деякі області застосування диференціальних рівнянь при аналітичному моделюванні.
4. Сформулюйте поняття напівнатурного моделювання.
5. Які структурні елементи включає в себе напівнатурна модель при будь-якому технічному втіленні?
6. Як здійснюється зв'язок реального обладнання з математичними моделями при напівнатурному моделюванні?
7. Сформулюйте поняття імітаційного моделювання.
8. Чому імітаційне моделювання зазвичай прирівнюється до статистичного експерименту?
9. Що є ключовим моментом в імітаційному моделюванні?
10. Перелічіть деякі види імітаційного моделювання.
11. Як задається динаміка системи при дискретно-подієвому моделюванні?
12. Для вивчення, яких систем застосовують такий вид імітаційного моделювання як системна динаміка?
13. У чому полягає суть експертного моделювання?
14. Які компоненти може включати експертна система?
15. При моделюванні, яких систем доцільно використовувати ситуаційне моделювання?
16. Які мови і моделі можуть використовуватися для опису ситуацій?

Глава 5. ПРИНЦИПИ Й ЕТАПИ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ

Математичне (комп'ютерне) моделювання засноване на побудові дослідником залежностей або алгоритмів, що відображають відносини елементів системи і оцінки їхнього впливу на функціональні характеристики системи оригінала.

Розглянемо основні принципи моделювання, які у стислій формі віддзеркалюють той достатньо багатий досвід, що накопичений до теперішнього часу в області розробки математичних моделей і їхнього використання при комп'ютерному моделюванні.

1. ***Принцип інформаційної достатності***. При повній відсутності інформації про досліджувану систему побудова її моделі неможливо. При наявності повної інформації про систему її моделювання позбавлене сенсу. Існує деякий критичний рівень апріорних відомостей про систему (рівень інформаційної достатності), при досягненні якого може бути побудована її адекватна модель.
2. ***Принцип здійсненності***. Створювана модель повинна забезпечувати досягнення поставленої мети дослідження з імовірністю, що істотно відрізняється від нуля, і за кінцевий час.
3. ***Принцип множинності моделі***. Даний принцип є *ключовим.* Мова йде про те, що створювана модель повинна відображати в першу чергу ті властивості реальної системи (або явища), які впливають на обраний показник ефективності. Відповідно при використанні будь-якої конкретної моделі пізнаються лише деякі сторони реальності. Для більш повного її дослідження необхідний ряд моделей, що дозволяють із різних сторін і з різним ступенем детальності відображати розглянутий процес.
4. ***Принцип агрегування***. У більшості випадків складну систему можна представити як систему складену з підсистем (*агрегатів*), для адекватного математичного опису яких виявляються придатними деякі стандартні математичні схеми. Принцип агрегування дозволяє, крім того, досить гнучко перебудовувати модель залежно від завдань дослідження.
5. ***Принцип параметризації***. У ряді випадків модельована система має у своєму складі деякі відносно ізольовані підсистеми, що характеризуються певним параметром, у тому числі векто-

рним. Такі підсистеми можна заміняти в моделі відповідними числовими величинами, а не описувати процес їхнього функціонування. При необхідності залежність значень цих величин від ситуації може задаватися у вигляді таблиці, графіка або аналітичного виразу. Принцип параметризації дозволяє скоротити об'єм і тривалість моделювання. Однак треба мати на увазі, що параметризація знижує адекватність моделі.

Ступінь реалізації перерахованих принципів у кожній конкретній моделі може бути різною, причому це залежить не, тільки від бажання розробника, але й від дотримання ним технології моделювання. А будь-яка *технологія* припускає наявність певної *послідовності дій.*

Технологія комп'ютерного моделювання передбачає виконання таких етапів:

Етап 1. Формулювання та обґрунтування задачі, завдання цілі моделювання

Етап 2. Розробка концептуальної моделі

Етап 3. Формалізація моделі

Етап 4. Програмна реалізація.

Етап 5. Налагодження, тестування та перевірка адекватності моделі.

Етап 6. Планування модельних експериментів;

Етап 7. Реалізація плану експерименту (проведення дослідження)

Етап 8. Аналіз та інтерпретація результатів моделювання.

На початковому етапі моделювання виробляється загальний підхід до досліджуваної проблеми, формулюється завдання і визначаються цілі комп'ютерного моделювання. На цьому етапі потрібне глибоке розуміння сутності поставленої задачі. Іноді, правильно поставити задачу не менш складно, ніж її вирішити. Постановка задачі комп'ютерного моделювання - процес не формальний і не має загальних правил.

На рис. 5.1 представлена блок-схема ітераційного пропроцесу математичного моделювання, яка містить перелічені вище етапи.

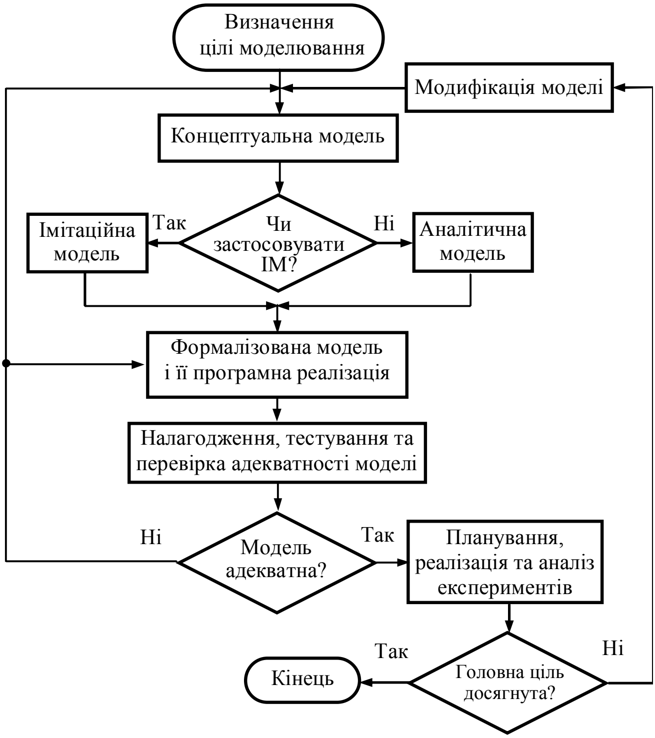


Рис. 5.1

Розглянемо докладніше зміст основних етапів комп'ютерного моделювання.

5.1. Визначення цілі моделювання

Сутність першого етапу комп'ютерного моделювання багато в чому ідентична начальним етапам розглянутого раніше системного аналізу. На цьому етапі аналізується й обґрунтовуються сутність проблеми, що виникла, і здійснюється її формулювання, формулюються задачі дослідження (ставиться задача комп'ютерного моделювання). Визначається клас систем і досліджувана система виділяється із зовнішнього середовища – проводиться межа між системою і середовищем. Задаються обмеження і цілі дослідження, які повинні бути реалізовані в результаті моделювання. Основні з них такі:

1) ***Розуміння***

Модель в цій ситуації потрібна для того, щоб зрозуміти, як створений конкретний об'єкт, яка його структура, основні властивості, закони розвитку та взаємодії з навколишнім світом. Приклади: модель руху планет Сонячної системи, модель кристалізації речовини, моделі хвильових рухів в різних середовищах, моделі деформації твердих тіл.

2) ***Прогнозування***

Модель використовується для того, щоб прогнозувати прямі і непрямі наслідки впливу на об'єкт різними способами. Приклади: модель атмосфери для прогнозу погоди, модель ринку, модель складної мережі або іншої технічної системи.

2) ***Управління***

Модель потрібна для того, щоб навчитися керувати об'єктом (або процесом) і визначити найкращі способи управління при заданих цілях і критеріях. Приклади: модель системи автоматичного управління, модель механізму, модель поведінки мас людей.

Однак ціль моделювання визначається не тільки ціллю досліджуваної операції, але і планованим способом використання результатів дослідження. Тому етап формулювання цілей включає також визначення критеріїв їх досягнення.

Наприклад, проблемна ситуація, яка вимагає ухвалення рішення, формулюється таким чином: змоделювати варіант побудови обчислювальної мережі, яка б мала мінімальну вартістю при дотриманні вимог з продуктивності та надійності. В даному випадку ціллю моделювання є не просто побудова моделі складної обчислювальної мережі, а такої моделі, яка забезпечувала б найкращу ступінь її пристосованості до вирішення поставленого завдання. Назвемо значення ступеня пристосованості моделі до виконання поставлених перед нею завдань ***критерієм ефективності моделювання***, в ролі якого в вище наведеній задачі виступає вартість. Під ефективністю моделювання будемо розуміти комплексну властивість процесу моделювання, як ступінь пристосованості до досягнення цілі.

Задача може бути сформульована інакше: з декількох варіантів конфігурації обчислювальної мережі вибрати найбільш надійний. Тут в якості критерію ефективності моделювання вибирається один з показників надійності (середнє напрацювання на відмову, імовірність безвідмовної роботи і т. д.), А ціллю моделювання є порівняльна оцінка варіантів мережі за цим показником.

Отже, при розробці конкретної моделі ціль моделювання повинна уточнюватися, для того щоб досягти найкращої результативності моделювання. Таки чином ціль моделювання визначається не тільки ціллю, з якою проводиться дослідження системи, але і вибором критерію ефективності моделювання, який забезпечує найкращу результативність моделювання.

Проте вибір цілі і критерію ефективності моделювання ще не визначає «архітектуру» майбутньої моделі, оскільки на цьому етапі не сформульована її концепція – не визначена концептуальна модель досліджуваної системи.

5.2. Розробка концептуальної моделі

На цьому етапі задається деякий «нематематичний» об'єкт (явище природи, конструкція, економічний план, виробничий процес). Виявляються основні особливості явищ, і зв'язку між ними на якісному рівні – формулюється концептуальна модель.

Концептуальна (змістовна) модель – це абстрактна модель, яка визначає структуру модельованої системи, властивості її елементів і причинно-наслідкові зв'язки, властиві системі і суттєві для досягнення цілі моделювання. Концептуальну модель іноді називають моделлю принципу дії, а якщо при описі досліджуваної системи увага приділяється особливостям роботи (функціонування) системи у взаємозв'язку з внутрішніми та зовнішніми елементами, то таку модель називають функціональною.

Графічним представленням концептуальних (функціональних) моделей служать структурні схеми, функціональні схеми, графи станів, часові діаграми і т. п. Зручною ілюстрацією концептуальної (функціональної моделі) є опис роботи вузла комутації повідомлень як одноканальної системи масового обслуговування, структура якої представлена на рис. 5.2

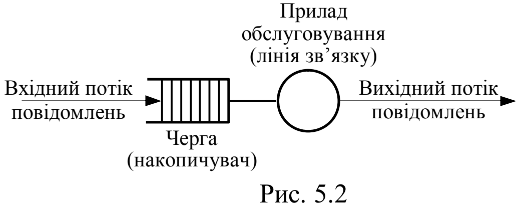


Рис. 5.2

В процесі розробки концептуальної моделі описується структура модельованої системи і її характеристики, на основі певних припущень виділяються найбільш суттєві елементи системи і причинно-наслідкові зв'язки між ними і зовнішнім середовищем, визначаються вхідні і вихідні дані, встановлюються цілі і обмеження.

Формалізований опис допомагає осмисленню моделі безпосередньо розробниками, що сприяє ефективній оцінці адекватності моделі та більш простих її модифікацій, а також взаєморозумінню між інженерами-системотехніками, математиками, програмістами і фахівцями з інших галузей. Такий опис служить базою для підвищення рівня абстрагування на наступних етапах.

Концептуальна модель повинна бути зрозумілою користувачеві; еволюційно незавершеною в тому сенсі, щоб дозволяти користувачу при необхідності вводити необхідні зміни; простою та надійною з точки зору досягнення заданих результатів.

Побудова концептуальної моделі включає такі етапи:

- визначення типу системи;
- опис зовнішніх впливів;
- декомпозиція системи.

Визначення типу системи

На цьому етапі здійснюється збір фактичних даних на основі: роботи з літературою та технічною документацією, проведення натурних експериментів, збору експертної інформації тощо. Також висуваються гіпотези щодо значень параметрів і змінних, для яких відсутня можливість отримання фактичних даних. Якщо отримані результати відповідають принципам інформаційної достатності та здійсненності, то вони можуть служити основою для віднесення модельованої системи до одного з відомих типів, аналогічних типам математичних моделей, розглянутих раніше. Найбільш важливі в цьому відношенні класифікаційні ознаки модельованої системи за типом поведінки наведені на рис. 5.3.

Одним з них є потужність множини станів (кількість станів) модельованої системи. За цією ознакою системи, як і математичні моделі, ділять на статичні і динамічні. ***Статичні моделі*** описують поведінку системи в будь-який момент часу, тобто множина її станів, містить один елемент. Якщо станів більше одного, або вони можуть змінюватися в часі, система називається динамічною. ***Динамічні моделі*** відображають поведінку об'єкта, процесу або системи у часі.

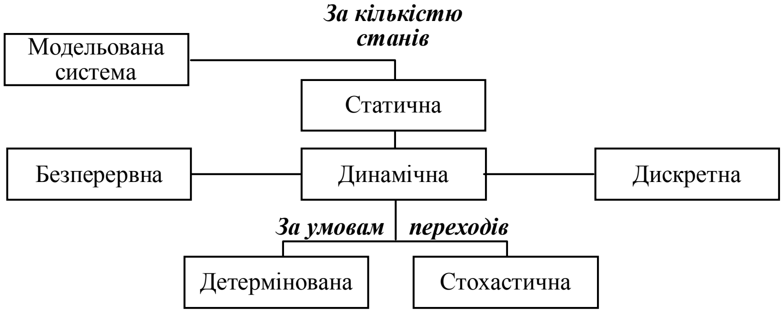


Рис. 5.3

Розрізняють два основних типи динамічних систем:

- безперервні (з безперервною множиною станів);
- дискретні (з дискретною множиною станів).

Системи з дискретними станами характеризуються тим, що в будь-який момент часу можна однозначно визначити, в якому саме стані знаходиться система. Для такої ідентифікації обов'язково потрібно знати ту ознаку, яка відрізняє один стан системи від іншого. Наприклад, при дослідженні систем масового обслуговування в якості такої ознаки зазвичай використовують число заявок в системі. Відповідно, зміна числа заявок в системі інтерпретується як перехід системи в новий стан. Дискретні системи підрозділяються на синхронні і асинхронні. У синхронних системах є точні часові позначки, в які відбуваються зміни стану, наприклад, тактовий генератор. В асинхронних системах зміна стану не прив'язана до часу. Такого типу динамічні системи з дискретними моментами зміни станів використовуються при дискретно-подієвому моделюванні.

Зміна станів може відбуватися і безперервно (зміна температури тіла при нагріванні). Безперервним може бути і множина станів, наприклад, зміна форми падаючої краплі. Такого типу динамічні системи є ***безперервними*** і найчастіше використовуються при аналітичному моделюванні. Безперервні системи діляться на системи з зосередженими параметрами і системи з розподіленими параметрами. У системах із зосередженими параметрами змінні залежать тільки від часу і не залежать від інших координат. Для систем з розподіленими параметрами змінні залежать як від часу, так і від інших координат. Залежно від сформульованої задачі дослідження одна і та ж система може розглядатися і як система з зосередженими параметрами і як система з розподіленими параметрами.

За умовами переходу з одного стану в інший розрізняють детерміновані системи і стохастичні.

У ***детермінованих*** системах новий стан залежить тільки від часу і поточного стану системи. Іншими словами, якщо є умови, що визначають перехід системи в новий стан, то для детермінованої системи можна однозначно вказати, в який саме стан вона перейде.

Для ***стохастичною*** системи можна вказати лише множину можливих станів переходу і, в деяких випадках, – імовірнісні характеристики переходу в кожний з цих станів.

Розглянута схема класифікації систем важлива не сама по собі. На етапі розробки концептуальної моделі вона, по-перше, дозволяє уточнити цілі і задачі моделювання і, по-друге, полегшує перехід до етапу формалізації моделі. Крім того, пізніше, на етапі оцінки якості розробленої моделі, знання класифікаційних ознак дає можливість оцінити ступінь її відповідності початковому задуму розробника.

Як правило, тип концептуальної системи з урахуванням поставлених цілей, оцінки складності завдання дослідження системи, наявних засобів, обмежень на час та на інші ресурси визначає вибір методу моделювання: аналітичне або імітаційне. Щоб вибір був вдалим, необхідно відповісти на два питання:

- До якого типу концептуальної системи може бути віднесено модельоване явище?
- З якою ціллю проводиться моделювання?

Відповіді на обидва ці питання можуть бути отримані в ході виконання двох перших етапів моделювання. Очевидно, в одних випадках більш доцільним є аналітичне моделювання, в інших - імітаційне (або поєднання того й іншого).

Внаслідок цього вибору реалізація інших етапів істотно розрізняється для кожного з підходів.

Опис зовнішніх впливів

При дослідженні ефективності функціонування системи[5.1] важливу роль відіграє коректний опис умов проведення досліджень. Зазвичай, він являє собою перелік і характеристики зовнішніх факторів, що впливають на систему. Сукупність діючих на систему факторів, які впливають на ефективність її функціонування, назвемо зовнішніми впливами (ЗВ).

У термінах імітаційного моделювання сукупність зовнішніх впливів, що впливають на ефективність застосування даної системи в рамках проведеної операції, називають ***робочим навантаженням*** відповідної системи.

Наприклад, нехай оцінюється продуктивність бортовий обчислювальної системи при управлінні польотом космічного корабля. За параметри робочого навантаження такої обчислювальної системи доцільно розглядати потік інформації, який підлягає обробці, і потік відмов, що призводить до порушення обчислювального процесу. Оцінки продуктивності обчислювальної системи матимуть сенс лише тому випадку, якщо відомо, для якого робочого навантаження вони отримані. Це ствердження справедливе для будь-якої задачі прийняття рішення, до якої б предметної області вона не відносилася.

Опис ЗВ (робочого навантаження) є не тільки важливим, але і досить складним завданням. Особливо у тих випадках, коли доводиться враховувати вплив випадкових факторів, або коли мова йде про ЗВ проектованої принципово нової системи. У зв'язку з цим багато авторів вводять поняття «моделі зовнішніх впливів», підкреслюючи порівнянність рівня складності опису власне системи і її робочого навантаження.

Модель зовнішніх впливів повинна володіти такими основними властивостями:

- сумісністю з моделлю системи;
- показністю;
- керованістю;
- системної незалежністю.

[5.1] Під ефективністю функціонування розуміють ступінь пристосування системи до виконання поставлених перед нею завдань.

Властивість ***сумісності*** передбачає, що, по-перше, ступінь деталізації опису ЗВ відповідає деталізації опису системи; по-друге, модель ЗВ повинна бути сформульована в тих же категоріях предметної області, що і модель системи (наприклад, якщо в моделі системи досліджується використання ресурсів, то модель ЗВ повинна бути виражена в запитах на ресурси).

Показність моделі ЗВ визначається її здатністю адекватно представити ЗВ відповідно до цілей дослідження. Іншими словами, модель ЗВ повинна відповідати цілям дослідження системи. Наприклад, якщо оцінюється пропускна здатність, то повинна вибиратися модель ЗВ, що «насичує» систему.

Під ***керованістю*** розуміється можливість зміни параметрів моделі ЗВ в деякому діапазоні, який визначається цілями дослідження.

Системна незалежність - це можливість перенесення моделі ЗВ з однієї системи на іншу зі збереженням її показності, тобто модель ЗВ не повинна залежати від конфігурації системи, що досліджується. Дана властивість найбільш важлива при розв‘язанні задач порівняння різних систем або різних модифікацій однієї системи. Концептуальна модель, що представлена в блоковому вигляді, – модель впливів зовнішнього середовища і власне модель процесу функціонування досліджуваної системи – є найбільш раціональною.

Декомпозиція

Аксіомою математичного моделювання є правило починати моделювання складної системи з найпростіших її варіантів. Тому розробка концептуальної моделі починається з побудови і аналізу окремих простих, але взаємопов'язаних, моделей даного об'єкту, процесу або системи. Надалі в результаті ітераційного процесу модель укрупнюється, уточнюється, і її відповідність об'єкту робиться повнішою.

Дроблення моделі складної системи на ряд простих моделей реалізується методом декомпозиції. Декомпозиція дозволяє замінити рішення однієї великої задачі рішенням серії менших, більш простих але взаємозв'язаних задач. Декомпозиція концептуальної моделі проводиться виходячи з обраного рівня деталізації моделі, який, в свою чергу, визначається трьома факторами:

- цілями моделювання;
- об'ємом апріорної інформації про систему;
- вимогами до точності і достовірності результатів моделювання.

Декомпозиція моделі виконується аналогічно розглянутої раніше декомпозиції системи. Для окремого елемента системи, отриманого в результаті декомпозиції, впливи з боку інших елементів уважаються входом, а вплив даного елемента на інші – виходом. Декомпозиція завжди умовна і може бути виконана багатьма різними способами.

Успіх комп'ютерного моделювання складних систем полягає не тільки і не стільки в розчленуванні складного цілого на більш прості частини, а в тому, що, будучи з'єднаними належним чином, ці частини знову утворюють єдине ціле - повний опис концептуальної моделі складної системи. Ця операція зворотна декомпозиції називається ***агрегуванням***. Агрегування окремих моделей складної системи в закінчену повну модель є завершальним етапом побудови концептуальної моделі системи.

5.3. Формалізація, програмна реалізація та оцінка адекватності моделі

Формалізацією називається процес побудови математичних (абстрактних знакових) моделей за допомогою деяких абстрактних мов (у вигляді диференціальних, інтегральних або різницевих рівнянь; у вигляді логіко-математичних схем і алгоритмів; з використанням теорії множин, теорії графів, математичної статистики; тощо). Таким чином, формалізований опис системи являє собою її математичну модель.

Формалізація моделі в термінах обраної математичної теорії дозволяє перейти від абстрактного формулювання концептуальної моделі до формулювання, що має конкретне математичний опис на формальних мовах. Мова алгебри дозволяє формалізувати функціональні залежності між величинами, а мова логіки дає можливість будувати формальні логічні моделі. Після формалізації модель для аналітичного моделювання постає у вигляді систем диференціальних, інтегральних або різницевих рівнянь, а для імітаційного моделювання – у вигляді логіко-математичних схем, алгоритмів, тощо.

На цьому етапі також, формалізуються обмеження, визначаються відповідні змінні та параметри, вхідні та вихідні дані.

Етап формалізації моделі один з найбільш складних етапів, для його реалізації потрібні спеціальні знання та вміння моделювання в певній предметній області (біологія, соціологія, технічні науки). Потрібно вміти для вирішення конкретного завдання виділити найбільш важливі характеристики і виявити методи вирішення даного завдання, що прийняті на практиці. При виборі методу враховуються знання користувача, його вподобання, а також вподобання розробника. Найчастіше тут використовуються чисельні методи, які хороше піддаються програмуванню. Зазвичай, для вирішення однієї і тієї ж задачі підходить кілька методів, що розрізняються точністю, стійкістю і т.д. Від правильного вибору методу часто залежить успіх всього процесу моделювання.

Після формалізації моделі і вибору методу вирішення поставленого завдання комп'ютерного моделювання починаються процедури алгоритмізації та програмування.

Поняття алгоритму – одне з основних в комп'ютерному моделюванні. Алгоритм це інструкція, що визначає послідовність дій, для виконання поставленого завдання. Число дій завжди кінцеве. Основними властивостями алгоритмів є дискретність і детермінованість: кожен крок алгоритму і перехід від кроку до кроку мають бути точно визначені.

Виділяють такі види алгоритмів: лінійний, розгалужений і циклічний. Для більш наочного уявлення алгоритму найчастіше використовується графічна форма – блок-схема, яка складається зі стандартних графічних об'єктів. Крім графічної форми алгоритми можуть формулюватися в словесній формі або у вигляді тексту.

Запис алгоритму на формальній мові іноді називають програмою, ототожнюючи слова «алгоритм» і «програма». Це не зовсім так, оскільки під алгоритмом, як правило, розуміють схему втілення основної ідеї розв‘язання задачі і в його основі лежить логічна складова, а не формальна. Програма ж являє собою інтерпретацію логічної схеми алгоритму певною мовою програмування для реалізації його на комп'ютері. Програма не тільки описує дії, але також оперує конкретними даними і містить атрибути, що забезпечують ефективний спосіб зв'язування інформації з машинним кодом. Основна програма може доповнюватися різними службовими підпрограмами (наприклад підпрограмами, що рисують або змінюють графічні образи в часі).

На етапі програмування алгоритм моделювання записується у вигляді тексту програми на обраної мові програмування, описуються масиви характеристик компонент, організовується збір даних про спостережувані змінні, розробляються програми візуалізації результатів моделювання.

На ранніх етапах комп'ютерного моделювання програми створювалися на мові машинних слів, наступним кроком стала мова Асемблера. Надалі з'явилися «мови високого рівня» (Алгол, Бейсік, Фортран, Паскаль, С ++, тощо). Крім універсальних алгоритмічних мов для програмної реалізація моделі можуть використовуватися пакети прикладних програм.

При аналітичному моделюванні програмуються алгоритми чисельних методів, наприклад, алгоритми розв'язання систем лінійних і нелінійних рівнянь, алгоритми інтерполяції та екстраполяції, алгоритми чисельного інтегрування та диференціювання і т.д.

При імітаційному моделюванні складаються програми імітації випадкових подій, процесів, транзактів, програмуються алгоритми модельованих систем (агрегатів, мереж, автоматів, систем масового обслуговування), алгоритми управління модельним часом (просування часу від події до події або просування часу з постійним кроком) і т. д.

При використанні пакетів прикладних програм програмування алгоритмів аналітичного і імітаційного моделювання не потрібно, оскільки вони вже закладені в базу пакетів прикладних програм.

Після розробки алгоритму та складання програми проводиться її налагодження, метою якої є виявлення і усунення помилок.

Основними джерелами помилок є порушення відповідності між схемою алгоритму і програми, записаної на мові програмування; невірне подання вихідних даних; неуважність при наборі програми і вихідних даних. Такого роду помилки виникають внаслідок неуважності програміста. Це такі помилки як втрата міток і описів масивів, дублювання міток, порушення балансу дужок. Можливі й помилки, пов'язані з втратою операторів, заміною букв в позначеннях змінних, відсутністю визначеності вихідних значень змінних, порушення адресації в масивах і т.д.

Враховуючи різноманітність джерел помилок, їх класифікують як синтаксичні, семантичні (смислові) і логічні помилки.

Синтаксичні помилки - це помилки в запису конструкцій мови програмування (помилки запису чисел, змінних, функцій, виразів, операторів, міток, підпрограм і т.д.).

Семантичні помилки - це помилки, пов'язані з неправильним змістом дій і використанням неприпустимих значень величин (функцій параметрів), що не відповідають її аргументам.

Логічні помилки - це помилки порушення логіки програми, що призводять до невірного результату. Подібного роду помилки, зазвичай, містяться в алгоритмах і вимагають їх ретельного аналізу.

Більшість середовищ програмування включають засоби налагодження, які забезпечують максимальну ефективність її застосування. Виявлення більшості синтаксичних і семантичних помилок в системах програмування автоматизовано і реалізується компіляторами і компонувальниками завдань на етапах трансляції програми на машинну мову і компонуванні програми.

Компілятори таких систем програмування видають діагностичне повідомлення про синтаксичні помилки в лістингу програми. Компонувальник задач систем програмування виявляє такі семантичні помилки, як невідповідність числа параметрів в описі підпрограми і в зверненні до неї, виклик неіснуючої програми, різні довжини спільного блоку пам'яті в окремих модулях, тощо.

Пошук же логічних помилок набагато менш формалізований; частина їх проявляється при виконанні програми як порушення процесу автоматичних обчислень і відображається або видачею діагностичних повідомлень програми, або відсутністю видачі результатів через нескінченного повторення однієї і тієї ж частини програми (зациклення), або появою непередбачуваної форми або змісту виведених результатів.

Для виявлення логічних помилок виконується трасування програми, тобто виконання програми або її ділянки, яке супроводжується виведенням в хронологічній послідовності інформації про події, що пов'язані з виконанням програми.

Існує багато помилок, які транслятор виявити не в змозі, якщо використовувані в програмі оператори сформовані правильно. Такого роду помилки можуть бути допущені на всіх етапах виконання завдання - від її постановки до оформлення. Ці помилки виявляються за допомогою тестування. Тест - це конкретний варіант значень вихідних даних, для якого наперед відомий очікуваний результат. Шляхом тестування також перевіряється *адекватність моделі*.

Як вже пояснювалося, адекватність моделі це відповідність моделі модельованому об'єкту або процесу. При цьому мається на увазі адекватність не взагалі, а адекватність за такими властивостям моделі, які є для дослідника істотними. Перевірка на адекватність полягає в тому, що результати тестування моделі, узгоджуються з результатами, отриманими шляхом теоретичних розрахунків, або з результатами експерименту на оригіналі.

Для оцінки результатів тестування слід використовувати знання фахівців предметної області. У разі невідповідності моделі реальному процесу повертаються до одного з попередніх етапів моделювання, який міг призвести до розробки невдалої моделі. Ітеративний процес тестування та вибору моделі триває доти, поки не будуть отримані прийнятні результати, тобто поки не буде підтверджена відповідність обраної моделі наявними даними про модельовану систему. Однак остаточне судження про адекватність моделі робиться за результатами експериментування з моделлю.

5.4. Планування та реалізація плану модельних експериментів. Аналіз результатів моделювання

Планування експерименту полягає у виборі його логічної структури і організації дій дослідника, які дозволяють вирішувати поставлену задачу при заданих обмеженнях за часом і вартістю.

Проведення модельних експериментів з визначення вихідних характеристик системи при різних значеннях змінних параметрів моделі є заключним етапом моделювання. Експерименти слід проводити за певним планом.

Планування експерименту це оптимальний розподіл ресурсів для досягнення поставлених цілей, яке дозволяє впорядкувати процес наукового пошуку і представити його у вигляді послідовності розв‘язання основних, окремих, а також додаткових завдань. Основне завдання дослідження формулюється на першому етапі комп'ютерного моделювання, а окремі задачі, які виступають в якості засобів розв‘язання основного, можна назвати проміжними. У міру розвитку дослідницького процесу окремі задачі можуть уточнюватися, виникати нові, і так до закінчення досліджень.

Планування модельного експерименту включає ряд етапів.

1**. *Встановлення цілі експерименту та його виду*** (контрольні, порівняльні, дослідницькі). Ціль експерименту повинна бути сформульована чітко і допускати кількісну оцінку. Будемо називати характеристику цілі, що задана кількісно, параметром оптимізації. Параметр оптимізації є реакцією модельованої системи на вхідні дії, які й визначають поведінку системи. Реакція модельованої системи багатогранна, багатоаспектна. Вибір того аспекту, який становить найбільший інтерес, як раз і задає ціль експерименту.

2. ***Вибір виду випробувань*** (нормальні, скорочені).

3. ***Вибір вхідних і вихідних параметрів***. Вхідні параметри (вхідні дії) повинні бути реєстрованими, але можуть бути детермінованими і керованими (залежними від дослідника), а також випадковими, тобто реєстрованими, але некерованими.

4. ***Складання плану проведення експерименту*** - кількість і порядок випробувань, спосіб збору, зберігання та документування даних.

Існує два основні варіанти постановки задачі планування модельного експерименту: стратегічне планування експерименту і тактичне планування

Експерименти з визначення вихідних характеристик системи зазвичай виконуються при різних значеннях керованих змінних параметрів моделі. При цьому кожен експеримент проводиться при певному поєднанні значень цих параметрів. Для проведення таких досліджень системи складається *стратегічний план*, який дозволяє отримати найбільш достовірне значення функції відгуку[5.2] на вхідні дії при фіксованому числі дослідів.

Сукупність методів зменшення тривалості модельного експерименту при забезпеченні статистичної достовірності результатів моделювання отримала назву *тактичного планування*. При тактичному плануванні модельного експерименту вибирається такий до-

[5.2] Функція відгуку розглядається як показник якості й ефективності модельованої системи.

пустимий план, при якому статистична оцінка функції відгуку може бути отримана із заданою точністю при мінімальному обсязі випробувань.

Планування модельних експериментів також має за мету визначення початкових умов, станів і параметрів моделі, які дозволяють швидко досягти сталого режиму.

Після складання плану проводиться власне обчислювальний експеримент шляхом реалізації прогонів моделі системи на комп'ютері відповідно до плану експерименту.

Проведення обчислювального експерименту можна умовно розділити на два етапи. Після першого етапу обчислювального експерименту, якщо потрібно, модель модифікується (уточнюється) як в напрямку її ускладнення (врахування додаткових ефектів і зв'язків в досліджуваному явищі щоб модель була більш адекватною дійсності), так і у напрямку її спрощення (нехтування деякими закономірностями та зв'язками заради досягнення практично прийнятного рішення). При цьому здійснюється повернення до одного з ранніх етапів моделювання, який міг призвести до розроблення невдалої моделі. Процес моделювання є ітеративним, тому на наступних етапах обчислювального експерименту цей цикл може повторюватися доки дослідник не переконається, що модель адекватна тому об'єкту, для якого вона складена.

При аналізі результатів моделювання проводиться статистична обробка даних експериментів та інтерпретація результатів моделювання. Інформація про результати модельних експериментів представляється у вигляді таблиць, графіків, діаграм, і схем, які дозволяють судити про показники ефективності системи і досягнення поставленої цілі.

До результатів моделювання відносять і аналіз чутливості моделі до варіацій її параметрів, виявляючи при цьому найбільш значущі параметри системи.

Планування, обробка та аналіз результатів експериментів особливо актуальні при проведенні імітаційного моделювання, як одного з видів статистичного моделювання.

Аналіз результатів моделювання дозволяє прийняти рішення: або дослідження буде продовжено, або закінчено.

Після завершення досліджень формується звіт за результатами виконаної роботи і надаються рекомендацій, наприклад, з проектування системи або з її модифікації.

У подальшому викладенні матеріалу вважається, що математичне формулювання задачі вже є, потрібно тільки вміти її розв'язувати на комп'ютері з використанням обраних чисельних методів, складаючи для цього алгоритми та програми розв’язку задачі.

Контрольні запитання

1. Перелічіть основні принципи моделювання.
2. Виконання, яких етапів передбачає технологія комп'ютерного моделювання?
3. У чому сутність першого етапу комп'ютерного моделювання?
4. Які цілі дослідження можуть бути досягнуті в результаті моделювання?
5. Що зазвичай розуміють під ефективністю моделювання?
6. Що зазвичай розуміють під концептуальною (змістовною) моделлю?
7. Які етапи включає в себе побудова концептуальної моделі?
8. Назвіть найбільш важливі класифікаційні ознаки модельованої системи за типом поведінки.
9. Що при моделюванні зазвичай розуміють під зовнішніми впливами?
10. Як в термінах імітаційного моделювання називають сукупність зовнішніх впливів?
11. Якими основними властивостями повинна володіти модель зовнішніх впливів?
12. Що мається на увазі під формалізацією моделі?
13. Сформулюйте поняття алгоритму?
14. У чому відмінність понять «алгоритм» і «програма»?
15. Як можна класифікувати помилки програм і алгоритмів моделювання?
16. Які помилки виявляють компілятор і компонувальник завдань середовища програмування на етапах трансляції?
17. Які основні етапи включає в себе планування модельного експерименту?

Глава 6. АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АНАЛІТИЧНОГО МОДЕЛЮВАННЯ СИСТЕМ З ВИКОРИСТАННЯМ ЧИСЕЛЬНИХ МЕТОДІВ

У ході розв'язання конкретної практичної задачі спеціаліст повинен, перш за все, визначити, до якого типу математичної задачі належить ця практична задача, вибрати чисельний метод для її розв'язання, розробити програмну реалізацію методу самостійно або вміти застосувати для її розв'язання один із відомих комп'ютерних математичних пакетів.

Чисельні методи дозволяють звести розв'язування задачі до виконання скінченного числа арифметичних і логічних дій з числами. При цьому розв'язок визначається як набір чисел, які надалі можуть бути інтерпретовані різним способом (наприклад, подані у вигляді таблиць, графіків, анімації тощо). Перевагами чисельних методів є: абсолютна універсальність, бо теоретично можуть бути застосовані для розв'язання будь-яких задач; добре пристосовані для реалізації на комп'ютері. Проте розв'язок, отриманий за допомогою чисельного методу, зазвичай є наближеним, тобто містить деякі погрішності. Джерелами погрішності є:

- погрішність вихідних даних;
- погрішність чисельного методу розв'язання;
- помилки округлення та розрахунків.

Чисельні методи бувають двох типів: прямі та ітераційні. В прямих методах розв'язок задачі досягається за скінченну кількість кроків методу після виконання останнього кроку, в ітераційних методах виконується ряд ітерацій методу до отримання наближеного розв'язку із заданою точністю. В основному чисельні методи є ітераційними. Ітерація – це повторення сукупності операцій або процедур для покращення наявного (поточного) наближеного розв'язку задачі.

Чисельні методи інтерполяції, екстраполяції, чисельного диференціювання й інтегрування, методи численного розв'язання диференціальних рівнянь, інші є основним апаратом аналітичних методів моделювання, і їх значущість тільки збільшуватиметься у міру вдосконалення комп'ютерної техніки.

6.1. Програмна реалізація чисельних методів розв‘язання систем лінійних рівнянь

Існує кілька методів розв‘язання системи лінійних алгебраїчних рівнянь (СЛАР), тобто знаходження безліч її рішень:

- розв‘язання СЛАР методом підстановки;
- розв‘язання СЛАР методом почленного додавання (віднімання) рівнянь системи;
- розв‘язання СЛАР за формулами Крамера;
- розв‘язання СЛАР за допомогою оберненої матриці;
- розв‘язання СЛАР методом Гауса.

Серед усіх методів найбільш універсальним способом розв‘язання СЛАР є метод виключення невідомих (метод Гаусса).

Система лінійних рівнянь може:

- мати єдиний розв‘язок.
- мати нескінченно багато розв‘язків.
- не мати розв‘язків (бути *несумісною*).

Метод Гаусса - найбільш потужний і універсальний інструмент для знаходження розв‘язку будь-якої СЛАР. Навіть в тих випадках, коли система має нескінченно багато розв‘язків або несумісна, метод послідовного виключення невідомих завжди призведе до результату.

Використання методу Гаусса передбачає виконання над розширеною матрицею СЛАР $\begin{cases} ax - by = -n \\ cx + dy = -m \end{cases}$, розв‘язок якої розшукується, деяких дії, які також називаються елементарними перетвореннями.

Існують такі елементарні перетворення:

- рядки матриці можна переставляти місцями. Наприклад[6.1],

$$\left(\begin{array}{cc|c} 1 & -1 & -5 \\ 2 & 1 & -7 \end{array}\right) \Rightarrow \left(\begin{array}{cc|c} 2 & 1 & -7 \\ 1 & -1 & -5 \end{array}\right);$$

[6.1] Вертикальна риска усередині матриць не несе ніякого математичного змісту - це просто відділення правих частин для зручності оформлення.

- якщо в матриці є (або з'явилися) пропорційні (як окремий випадок – однакові) рядки, то слід видалити з матриці всі ці рядки крім однієї. Наприклад,

$$\begin{array}{r} \\ \text{однаковий} \rightarrow \\ \text{однаковий} \rightarrow \\ \text{пропорційний} \rightarrow \end{array}\left(\begin{array}{cccc|c} 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & -1 & 2 & 3 \\ 0 & 1 & -1 & 2 & 3 \\ 0 & 2 & -2 & 4 & 6 \end{array}\right) \Rightarrow \left(\begin{array}{cccc|c} 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & -1 & 2 & 3 \end{array}\right);$$

- якщо в матриці в ході перетворень з'явився нульовий рядок, то його слід видалити;
- рядок можна помножити (поділити) на будь-яке число, відмінне від нуля. Наприклад,

$$\begin{array}{r} \text{ділимо на } (-3) \rightarrow \\ \text{множимо на } 2 \rightarrow \end{array}\left(\begin{array}{cc|c} -3 & 9 & 15 \\ 0{,}5 & 0 & 2{,}5 \end{array}\right) \Rightarrow \left(\begin{array}{cc|c} 1 & -3 & -3 \\ 1 & 0 & 5 \end{array}\right);$$

- до рядка матриці можна додати інший рядок, помножений на число, відмінне від нуля. При цьому змінюється тільки той рядок, до якого додають. Наприклад,

$$\left(\begin{array}{cc|c} 1 & -1 & -5 \\ 2 & 1 & -7 \end{array}\right) \begin{array}{l} \text{до другого рядку додали} \\ \text{переший помножений на } (-2) \end{array} \Rightarrow \left(\begin{array}{cc|c} 1 & -1 & -5 \\ 0 & 3 & 3 \end{array}\right).$$

Елементарні перетворення не змінюють розв‘язок СЛАР, ***але їх, ні в якому разі не можна використовувати при «класичних» діях з матрицями***.

Як приклад розв‘яжемо простеньку СЛАР типу $\begin{cases} x - y = -5 \\ 2x + y = -7 \end{cases}$ методом Гауса.

На першому етапі потрібно записати розширену матрицю системи: $\left(\begin{array}{cc|c} 1 & -1 & -5 \\ 2 & 1 & -7 \end{array}\right)$,

а потім за допомогою елементарних перетворень привести її до східчастого (трикутного) вигляду:

$$\left(\begin{array}{cc|c} 1 & -1 & -5 \\ 2 & 1 & -7 \end{array}\right) \begin{array}{l} \text{Щоб позбавитися від однієї} \\ \text{змінної у другому рядку,} \\ \text{(отримати нуль у 1-ої позиції)} \\ \text{до другого рядку додали} \\ \text{перший помножений на } (-2) \end{array} \Rightarrow \left(\begin{array}{cc|c} 1 & -1 & -5 \\ \boxed{0} & 3 & 3 \end{array}\right),$$

$\left(\begin{array}{cc|c} 1 & -1 & -5 \\ \boxed{0} & 3 & 3 \end{array}\right)$ Для того щоб отримати одиницю у 2-ої позиції 2-ого рядку ділимо його на 3 $\Rightarrow \left(\begin{array}{cc|c} 1 & -1 & -5 \\ 0 & 1 & 1 \end{array}\right)$.

В результаті елементарних перетворень отримана система

$$\begin{cases} x - y = -5 \\ \quad\;\; y = 1 \end{cases},$$

яка еквівалентна вихідної системі рівнянь:

Тепер систему потрібно «розкрутити» в зворотному напрямку – від низу до верху, цей процес називається *зворотним ходом* методу Гауса.

У нижньому рівнянні є вже готовий результат: $y = 1$, тоді з першого рівняння виходить $x = -4$

В даний час розроблено багато програмних реалізацій методу Гаусса, в яких реалізована двохетапна обчислювальна схема: етап прямого ходу – приведення системи до трикутного вигляду; етап зворотного ходу – визначення невідомих.

Нижче наведено варіант програмної реалізації методу Гаусса в середовищі програмування Делфі, який не претендує на оптимальність і повноту реалізації.

```
****************************************************
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls;
const
n=4;  m=n+1;// Задається розмірність розширеної матриці СЛАР
Type
mat = array[1..n,1..m] of real;
mat1= array[1..n] of real;
  TForm1 = class(TForm)
...
...
procedure Button1Click(Sender: TObject);
      private
      public
end;
Var
  Form1: TForm1;
  d,s: Real;{ допоміжні змінні }
  k, j, i:Integer; { лічильники }
  a : mat; { коефіцієнти розширеної матриці СЛАР }
  b : mat1;{ розв‘язання СЛАР }
```

```
procedure TForm1.Button1Click(Sender: TObject);
begin
...
... // Тут будь-яким способом вводяться коефіцієнти матриці a[i,j]
    ...
{Прямий хід:
      - Послідовно всі коефіцієнти рядків розширеної матриці, починаючи з останніх коефіцієнтів, діляться на перший коефіцієнт рядка, - в наслідок чого перший коефіцієнт перетворюється в одиницю. Перший рядок готовий.
      - З решти рядків матриці віднімається перший рядок, - в наслідок чого перші коефіцієнти рядків перетворюються в нуль.
      - Знову все коефіцієнти рядків, що залишилися, починаючи з другого і починаючи з останніх коефіцієнтів, діляться, але вже на другий коефіцієнт другого рядка, в результаті чого він перетворюється в одиницю. Другий рядок готовий.
      - З решти рядків матриці, починаючи з третього, віднімається вже другий рядок, - в наслідок чого перші коефіцієнти рядків, що залишилися, перетворюються в нуль і т.д.}
   for i:=1 to n-1 do
   begin
   for k:=m downto i do // починаючи з останніх коефіцієнтів
   for j:=i to n do
   a[j,k]:=a[j,k]/a[j,i];// отримаємо одиницю
   for j:=i+1 to n do
   for k:=i to m do
   a[j,k]:= a[j,k]-a[(i),k];// отримуємо нуль
   end;
{Зворотний хід: Коріння СЛАР b [i] знаходять, починаючи з останнього, - b [n]. За знайденими коріння, використовуючи метод підстановки, розшукуються інші корені СЛАР}
   for j:=1 to n do
   b[j]:=0;// Онулення матриці розв‘язків
   for i:=n downto 1 do
   begin
   for j:=n downto 1 do
   begin
   s:=b[j]*a[i,j];// підстановка знайдених на попередніх кроках розв‘язків
   d:=d+s;
   end;
   b[i]:=(a[i,m]-d)/a[i,i];
   d:=0;// Онулення раніше розрахованих допоміжних змінних
   s:=0;
   end;
      { Виведення результатів розв‘язання СЛАР }
      LABEL77.Caption:='W= '+floattostrf(b[n],ffFixed,6,4);
      LABEL78.Caption:='Z= '+floattostrf (b[n-1],ffFixed,6,4);
      LABEL79.Caption:='Y= '+floattostrf (b[n-2],ffFixed,6,4);
      LABEL80.Caption:='X= '+floattostrf (b[n-3],ffFixed,6,4);
end;
      end.
```

6.2. Програмна реалізація чисельних методів розв‘язання нелінійних рівнянь

Розв‘язання нелінійних (зокрема, трансцендентних) рівнянь виду

$$f(x) = 0, \tag{6.1}$$

або

$$f_1(x) = f_2(x) \tag{6.2}$$

полягає в знаходженні одного або всіх коренів на заданому відрізку $[x_н, x_к]$, на якому, є корені рівняння.

Графічна інтерпретація розв‘язання рівнянь виду (6.1) зводиться до пошуку аргументу x_0, при якому функція $f(x)$ перетинає вісь абсцис (рис. 6.1, а). Розв‘язання рівнянь виду (6.2) інтерпретується як пошук точки перетину функцій $f_1(x)$ і $f_2(x)$ (рис. 6.1, *б*).

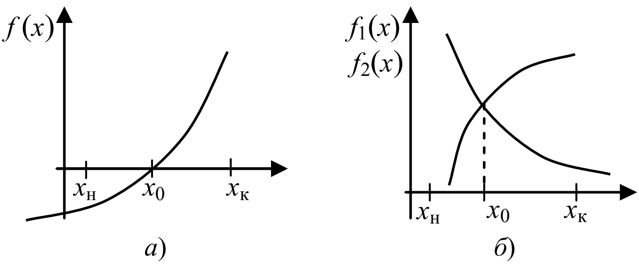


Рис. 6.1

Задачі типу (6.2) вирішуються, наприклад, в бортових контролерах зльоту, при визначенні швидкості «прийняття рішення» про продовження або припинення розбігу, наприклад, при часткової втрати тяги двигуна. При цьому функція $V_1(L)$ описує графік розбігу, а функція $V_2(L)$ – графік гальмування (див. рис. 6.2).

При чому основні параметри розбігу та гальмування залежався від багатьох чинників: злітної масі літака; тяги двигуна; стану злітно-посадкової смуги, який характеризується коефіцієнтом тертя; силою та напрямком вітру.

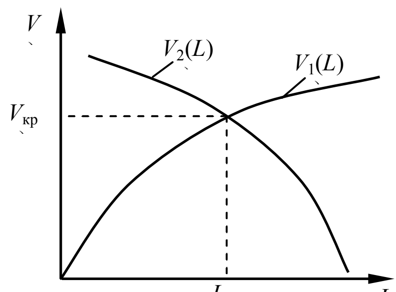


Рис. 6.2

Знаходження точки перетину кривих $V_1(L)$ і $V_2(L)$, тобто розв‘язання нелінійного рівняння типу (6.2) визначає критичні параметри розбігу $L_{кр}$ і $V_{кр}$. Задача логічного пристрою полягає в тому, щоб визначити, чи буде досягнута критична швидкість $V_{кр}$ до моменту пробігу критичної відстані $L_{кр}$. Тільки в цьому випадку можна продовжувати зліт. При цьому треба мати на увазі, що рішення про продовження або припинення розбігу повинне бути прийняте до досягнення критичних значень швидкості та відстані, оскільки в іншому випадку ні продовження, ні припинення розбігу не забезпечать необхідної безпеки зльоту.

Розв‘язання таких завдань графічним методом не завжди задовольняє вимогам з точності, а аналітичне розв‘язання не завжди можливо отримати через складність рівнянь, тому застосовують наближені чисельні методи. В цьому випадку зазвичай локалізують кожен корінь на своєму відрізку $[x_{н},\ x_{к}]$, і тоді знаходження коренів зводиться до процедури ітераційного зменшення відрізків локалізації кореня $[x_{н},\ x_{к}]$ одним з відомих методів. До чисельних методів розв'язання нелінійних (зокрема, трансцендентних) рівнянь відносяться:

- Метод дихотомії (метод поділу відрізка навпіл);
- Метод простих ітерацій;
- Метод хорд;
- Метод Ньютона (метод дотичних);
- Метод Чебишева.

Як приклад розглянемо детально метод простих ітерацій.

Метод простих ітерацій заснований на поданні рівнянь (6.1) і (6.2) у вигляді

$$x = \xi(x) \tag{6.3}$$

і багаторазовому застосуванні ітераційної формули

$$x_{n+1} = \xi(x_n), \tag{6.4}$$

до тих пір, поки виконується умова

$$|x_{n+1} - x_n| \geq \varepsilon, \tag{6.5}$$

де n – порядковий номер ітерації; ε – задана точність обчислення кореня нелінійного рівняння.

Розглянемо порядок перетворень для отримання рівняння у вигляді (6.3) та ітераційної формули (6.4).

Якщо вихідне рівняння подано виразом (6.1), то його слід перетворити до виду (6.2). Далі можемо записати для лівої частини рівняння (6.2)

$$y = f_1\,(x), \tag{6.6}$$

а для правої частини рівняння (6.2)

$$y = f_2\,(x). \tag{6.7}$$

Рівняння (6.6) і (6.7) утворюють систему рівнянь:

$$\begin{cases} y = f_1(x); \\ y = f_2(x). \end{cases}$$

З рівняння (6.7) виразимо аргумент x через значення функції y:

$$x = \xi_2(y). \tag{6.8}$$

Наприклад, якщо функція (6.7) надана як $y = kx+b$, то аргумент x через значення функції y можемо записати як $x = (y - b)/k$, а якщо функція y подана у вигляді $y = \sin(x)$, то отримаємо $x = \arcsin(y)$. Потім в знайдений вираз (6.8) підставляємо значення y з (6.6) і отримуємо

$$x = \xi_2\,(f_1\,(x)) = \xi\,(x). \tag{6.9}$$

З (6.9) знаходимо ітераційну формулу

$$x_{n+1} = \xi_2\,(f_1\,(x_n)), \tag{6.10}$$

яку необхідно застосовувати доти, поки виконується умова (6.5). Формула (6.10) дозволяє за одну ітерацію обчислювати значення аргументу x_{n+1}, використовуючи попереднє значення x_n,. На наступній ітерації в (6.10) підставляємо отримане значення x_{n+1} обчислюємо наступне значення аргументу. Описана ітераційна процедура зображена на рис. 6.3, а її алгоритм – на рис. 6.4.

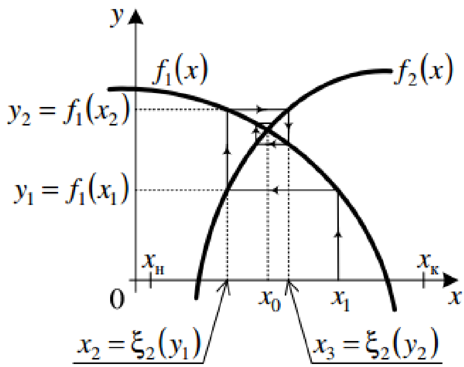


Рис. 6.3

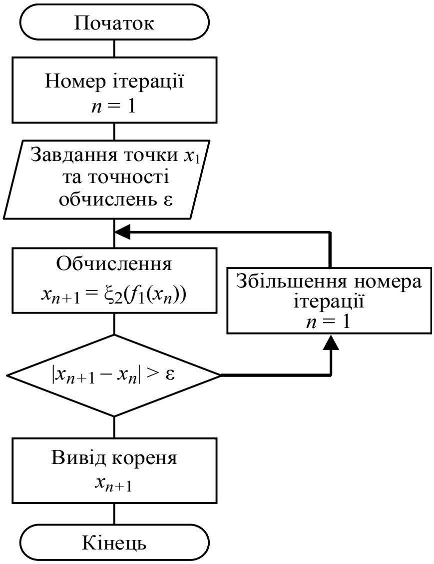


Рис. 6.4

Отже для розв‘язання нелінійного рівняння методом простих ітерацій необхідно виконати наступні дії:

1. Використовуючи перетворення (6.3) ... (6.10) отримати ітераційну формулу.
2. На заданому інтервалі аргументів обрати значення вихідного аргументу x_1.
3. Розв‘язати отриману ітераційну формулу, підставивши в неї значення аргументу.
4. Використовуючи нове обчислене значення аргументу і старе значення аргументу перевірити умову (6.5).
5. Якщо умова виконується, то перейти до пункту 3, підставляючи в ітераційну формулу обчислене значення аргументу.

Якщо умова не виконується, то процедура обчислення закінчується і вважається, що останнє значення аргументу x_{n+1} і є обчисленим з заданою точністю ε коренем x_0 нелінійного рівняння, тобто $x_0 = x_{n+1} \pm \varepsilon$.

Нижче наведений лістинг програмної реалізації методу простих ітерацій для розв‘язання нелінійного рівняння виду:

$$y = 4/(5 - 0{,}05x^5).$$

```
******************************************************************************
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,Dialogs, StdCtrls;
Type
    TForm1 = class(TForm)
    Button1: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    procedure Button1Click(Sender: TObject);
            private   { Private declarations }
            public    { Public declarations }
    end;
Var
    Form1: TForm1;
    K,x0,xn,xn1,q,e,d,d1:real;
{ x0-початкове наближення; e-точність; xn1-поточний розв‘язок;
xn-попередній розв‘язок; d-похибка; K-кількість ітерацій}
    implementation
    {$R *.dfm}
function F(x:real):real;
begin
    F:=4/(5-(0.05*(x*x*x*x*x))); ;// Задане рівняння
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
   x0:=StrToFloat(edit1.Text);// Завдання початкового наближення
   e:=StrToFloat(edit2.Text); );// Завдання точності розв‘язання
   xn:=x0;
   K:=0;
   repeat
        K:=K+1;// Підрахунок кількості ітерацій
        xn1:=xn;
        xn:=F(xn1);
        d1:=abs(xn1-xn);
   until d1<e;//Умова закінчення ітераційного процесу
        {Виведення результатів обчислення}
        edit3.text:= 'x = '+floattostrf(xn,fffixed, 10,8);
        edit4.text:= 'K = '+floattostrf(K,fffixed, 10,8);
end;
        end.
******************************************************************************
```

Чисельні методи розв‘язання лінійних і нелінійних рівнянь часто є складовими інших чисельних методів, використовуваних при аналітичному моделюванні систем.

6.3. Програмна реалізація алгоритмів інтерполяції

Задача наближення функцій (апроксимація) за методом інтерполяції виникає при розв'язанні багатьох задач чисельного аналізу: чисельного інтегрування і диференціювання, розв'язання диференціальних рівнянь, розв'язання систем нелінійних рівнянь, задач оптимізації та ін.

На практиці деякі характеристики об'єкта отримують експериментально. Наприклад, за результатами продування в аеродинамічній трубі на різних швидкостях вимірюють значення коефіцієнта аеродинамічної піднімальної сили c_{y_a}, формуючі таблицю $c_{y_a}(V)$, а потім табличні значення апроксимують графіком – кривою, яка не обов'язково з'єднує всі точки таблиці (див. рис. 6.5).

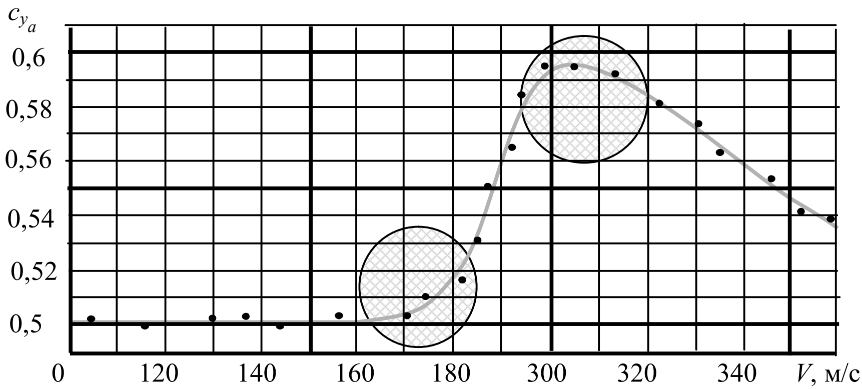


Рис.6.5

При комп'ютерному моделюванні навпаки вихідні графічні залежності переводять в табличні значення. Так, при моделюванні динаміки руху літака в табличні значення переводять залежності коефіцієнтів аеродинамічних сил і моментів літака від параметрів польоту, які у вигляді графіків наведені в технічній документації.

З таблиці легко визначити значення функції, якщо значення аргументу дорівнює одному з табличних значень x_i. Якщо ж значення аргументу не влучає в точки таблиці, а знаходиться між ними, то для обчислення значення функції необхідно описати її якимось чином між дискретними точками таблиці. Одним з методів опису функції між відомими точками таблиці є інтерполяція функції.

6.3.1. Лінійна інтерполяція

При лінійній інтерполяції передбачається, що залежність $y = f(x)$ на кожній ділянці між дискретними точками таблиці є лінійною. Таке припущення допустимо при монотонному характері функції на аналізованих ділянках, і дозволяє виконувати наближені розрахунки. Для отримання інтерполяційного значення функції при заданому значенні аргументу, через дві сусідні точки (y_i, x_i) и (y_{i+1}, x_{i+1}) залежності $y = f(x)$ проводять пряму лінію (рис. 6. 6).

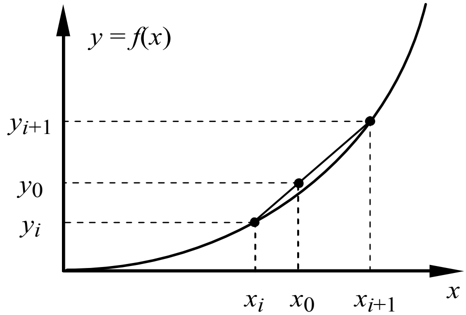


Рис. 6. 6

Для будь-яких точок (y, x), що належать цій прямій, очевидним є співвідношення:

$$\frac{x - x_i}{y - y_i} = \frac{x_{i+1} - x_i}{y_{i+1} - y_i},$$

з якого можна отримати шукане значення функції y_0 при заданому значенні аргументу x_0

$$y_0 = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x_0 - x_i).$$

Точність обчислення значення функції методом лінійною інтерполяції може бути недостатньою, оскільки іноді інтерполюючий відрізок прямої лінії не збігається з реальною кривою, яка описує інтерпольовану функцію. Для зменшення похибки можна на нелінійних ділянках функції збільшити частоту її табулювання, тобто для цих ділянок при формуванні таблиці зменшувати крок зміни аргументу. На рис. 6.5 виділені області функції $c_{y_a} = f(V)$, в яких доцільно збільшувати частоту табулювання.

При написанні програми, що реалізує алгоритм методу лінійної інтерполяції, рекомендується застосувати автоматизований пошук інтервалу характеристики $y = f(x)$, на якому виконується розрахунок.

Приклад фрагмента алгоритму пошуку заданого інтервалу табульованої функції зображений на рис. 6.7.

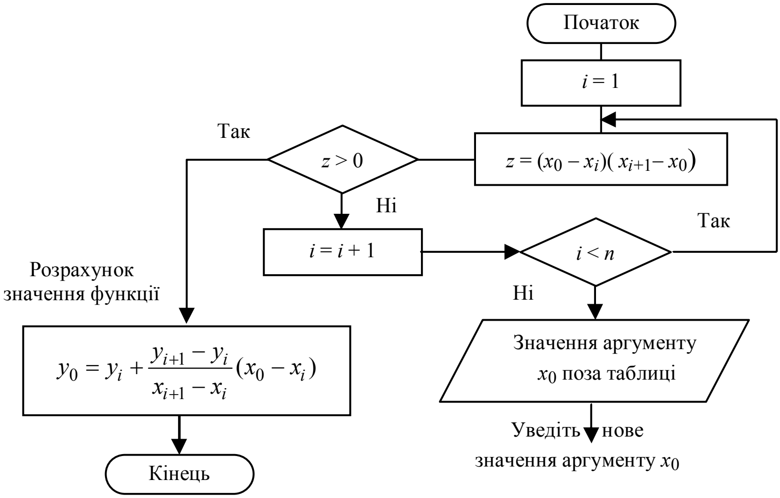


Рис. 6.7

Природно, що при проведенні модельного експерименту, наприклад з дослідження складної системи, методи інтерполяції нелінійних функцій, які описують зміну деяких параметрів системи, оформляються як підпрограми загальної програми експерименту. Найчастіше лінійна інтерполяція використовується для обчислення табличних значень графічної функції, яка отримана експериментально. Наприклад, як вже зазначалося, для обчислення коефіцієнтів аеродинамічних сил і моментів від параметрів польоту при моделюванні динаміки руху літака.

Табличні значення нелінійної функції можуть задаватися у вигляді, наприклад, двомірного масиву даних і формуватися безпосередньо в програмі експерименту, а можуть зчитуватися з зовнішніх файлів. Наприклад, з таблиць, створених в будь-якому текстовому редакторі (тільки не в Word), наприклад в Notepad або Excel.

Програмна реалізація процедури зчитування таблиці даних з Excel наведена нижче

```
******************************************************
uses
  Windows, Messages, …. ComObj,
…
…
var
  Form1: TForm1;
  Excel: Variant;
implementation
…
procedure ReadExel(var FileName:String; m:mat);
var
WorkSheet: OLEVariant;
Rows,Cols,i,j: integer;
begin
  { Ініціалізація Excel}
Excel := CreateOleObject('Excel.Application');
{ Відкриваймо книгу }
Excel.Workbooks.Open(FileName);
{Шлях до файлу прописується в основній програмі, наприклад
FileName:='C:\interpol\kniga1.xls'}
WorkSheet:=Excel.ActiveWorkbook.ActiveSheet;//Одержуємо
                                           //активний лист
 {визначаємо кількість рядків і стовпців таблиці}
 Rows:=WorkSheet.UsedRange.Rows.Count; //рядки
 Cols:=WorkSheet.UsedRange.Columns.Count; // стовпці
{Зчитуємо дані з Excel в двовимірний масив m[j,i]}
for i := 1 to Rows do     // i-рядки
  begin
    for j := 1 to Cols do // j-стовпці;
  begin
  m[j,i]:=WorkSheet.UsedRange.Cells[i,j].Value;
  end;
  end;
  {Закриваємо книгу}
Excel.ActiveWorkbook.Close;
end;
******************************************************
```

Головним недоліком лінійної інтерполяції є те, що інтерполююча функція не володіє гладкістю[6.2], тому її перша похідна терпить розрив у вузлах інтерполяції (в табульованих точках).

Відмінним від лінійної інтерполяції методом, що дозволяє підвищити гладкість апроксимації шуканих залежностей, є глоба-

[6.2] Властивість функції мати безперервну похідну на всієї множині свого визначення.

льна поліноміальна інтерполяція, тобто інтерполяція всього інтервалу табулювання за допомогою єдиного інтерполяційного многочлена (поліноми Лагранжа, Ньютона, Стірлінга), значення якого в табульованих точках збігаються зі значеннями функції в цих точках.

Однак глобальна поліноміальна інтерполяція не завжди дає задовільні результати. Так, наприклад, інтерполяційна функція може мати значні відхилення від апроксимуючої кривою на ділянках між табульованими точками, а спроба зменшити ці відхилення за рахунок підвищення ступеня інтерполяційного полінома (збільшення кількості табульованих точок) призводить до збільшення похибки через виникнення так званої хвилястості.

Для зниження похибок інтерполяція застосовують, ***кусково-поліноміальну інтерполяцію***, коли весь інтервал табулювання розбивається на часткові відрізки (шматочки) і на кожному з них табличну функцію замінюють, на відміну від лінійної інтерполяції, поліномом невисокого ступеня. Нелінійні функції здатні більш точно описати поведінку табульованої функції на заданому інтервалі ніж лінійні.

Однак якщо потрібно, щоб апроксимуюча функція була гладкою, то кусково-поліноміальна інтерполяція неприйнятна. У цьому випадку застосовують сплайн-інтерполяцію заданої гладкості.

6.3.2. Сплайн-інтерполяція

Сплайном називається функція, область визначення якої розбита на кінцеве число відрізків, на кожному з яких сплайн збігається з деяким поліномом, причому для кожного відрізка цей поліном власний. Зауважимо, що кусково-лінійна інтерполяція також є сплайн-інтерполяція, для якої на кожному відрізку функцію апроксимують лінією, але ніяких умов гладкості в даному випадку не накладають. На практиці найчастіше застосовується кубічна сплайн-інтерполяція, тобто при сплайн-інтерполяції використовують інтерполяційні поліноми третього порядку і додатково потребується безперервність першої та другої похідних сплайна на всьому інтервалі табулювання.

Іноді для збільшення точності обчислень можуть використовуватися поліноми й більш високого порядку. Так в основу алгоритмів обчислення аерометричних параметрів цифрових систем повітряних сигналів покладений метод сплайн-інтерполяції табличних градуйовочних залежностей для стандартної атмосфери. В обчислювачі весь інтервал табулювання розбиваються на 16 підінтервалів, у кожному з яких шукана функція апроксимується поліномом (від 3-го до 5-го порядку).

Основними характеристиками сплайна є:

• ступінь сплайна – максимальна з ступенів інтерполяційних поліномів;

• гладкість сплайна – кількість безперервних похідних, які має сплайна на всьому інтервалі табулювання;

• дефект сплайна – різниця між ступенем і гладкістю сплайна.

Наприклад, кусково-лінійний сплайн має ступінь 1, гладкість 0 і дефект 1. Кусково-кубічний сплайн має ступінь 3, гладкість 2 і дефект 1.

При використанні сплайн-інтерполяції інтерполююча функція $S(x)$, що «склеєна» з окремих інтерполюючих поліномів, проходить через систему з n вузлів, за які використовують табульовані точки. При кубічної сплайн-інтерполяції в якості інтерполюючих поліномів використовують поліноми третього порядку, які на інтервалі $x \in [x_{i-1}, x_i]$ записують у вигляді

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 .$$

Коефіцієнти a_i, b_i, c_i, d_i поліномів інтерполяції на кожному інтервалі $i = \overline{1, n-1}$ визначаються з умов сполучення у вузлах, що забезпечується на межах інтервалів $[x_i; x_{i+1}]$ безперервністю, як інтерполюючої функції $S(x)$, так і її першої $S'(x)$ та другої $S''(x)$ похідних, тобто:

$$\left.\begin{aligned} S(x) &= S_i(x) = S_{i+1}(x), \\ S'(x) &= S_i'(x) = S_{i+1}'(x), \\ S''(x) &= S_i''(x) = S_{i+1}''(x), \end{aligned}\right\} \quad \text{при } i = \overline{1, n-1}. \qquad (6.11)$$

При цьому на кожному з відрізків $[x_i; x_{i+1}]$ коефіцієнти i-их поліномів інтерполяції різні. Коефіцієнти a_i визначаються як табличні значення інтерполюючої функції $y(x)$ в i-их вузлах: $a_i = y(x_i), \quad i = \overline{1, n-1}$.

Для визначення решти $3(n-1)$ невідомих коефіцієнтів поліномів інтерполяції b_i, c_i, d_i крім умов безперервності задають додаткові, так звані, граничні умови – значення першої або другої похідної функції на межах (x_1, x_n) всього інтервалу табулювання інтервалу. Якщо ці значення невідомі, то можна отримати достатньо хороші результати апріорі задавши другу похідну на межах (x_1, x_n) рівною нулю тобто

$$\begin{aligned} &S_1''(x_1) = 2c_1 = 0, \\ &S_{n-1}''(x_n) = 2c_{n-1} + 6d_{n-1}(x_n - x_{n-1}) = 0. \end{aligned} \tag{6.12}$$

Розкриваючи умови безперервності (6.11) і переписуючи їх в більш наочній формі, отримаємо, опускаючи проміжні викладення, спільно з формулами граничних умов (6.12) систему лінійних рівнянь для пошуку невідомих коефіцієнтів b_i, c_i, d_i для кожного ($n-1$)-го інтервалу.

$$\begin{aligned} &y(x_i) + b_i(x_{i+1} - x_i) + c_i(x_{i+1} - x_i)^2 + d_i(x_{i+1} - x_i)^3 = y(x_{i+1}), \\ &b_i + 2c_i(x_{i+1} - x_i) + 3d_i(x_{i+1} - x_i)^2 = b_{i+1}, \\ &2c_i + 6d_i(x_{i+1} - x_i) = 2c_{i+1}, \\ &c_1 = 0, \\ &c_{n-1} + 3d_{n-1}(x_n - x_{n-1}) = 0, \end{aligned} \tag{6.13}$$

тут у першому рівнянні $i = \overline{1, n-1}$, а у другому і третьому $i = \overline{1, n-2}$.

Але, перш ніж розв‘язувати цю систему, вигідно перетворити її так, щоб невідомими була тільки одна група коефіцієнтів c_i.

Знову ж, опускаючи проміжні викладки, отримаємо

$$\Delta x_{i-1} c_{i-1} + \xi_i c_i + \Delta x_i c_{i+1} = r_i, \tag{6.14}$$

тут $\Delta x_i = x_i - x_{i-1}$;

$\Delta x_{i-1} = x_{i-1} - x_{i-2}$;

$\xi_i = 2(\Delta x_i + \Delta x_{i-1})$;

$r_i = 3\left[\left(y(x_i) - y(x_{i-1})\right)/\Delta x_i - \left(y(x_{i-1}) - y(x_{i-2})\right)/\Delta x_{i-1}\right]$;

$y(x_i)$ – значення функції в табульованих точках.

Коефіцієнти c_1 и c_{n+1} отримують з умов вільних кінців сплайна. Зазвичай вимагають нульову кривизну на кінцях сплайна, тому в рівнянні (6.14) слід покласти

$$c_1 = c_{n+1} = 0. \qquad (6.15)$$

Отже, $(n - 1)$ рівнянь виду (6.14) разом з умовами (6.15) створюють систему лінійних алгебраїчних рівнянь для визначення коефіцієнтів c_i.

У кожне з рівнянь типу (6.14) входять три невідомих з послідовними значеннями індексів c_{i-1}, c_i, c_{i+1}.

Для вирішення подібних рівнянь доцільно застосовувати так званий метод прогонки, що є окремим випадком методу виключення Гаусса. Так само, як і метод Гаусса, метод прогонки розділяється на два етапи – прямий та зворотний.

В процесі виконання прямого ходу прогонки обчислюються значення прогоночних коефіцієнтів k_i і m_i, за допомогою яких встановлюється лінійний зв'язок для будь-яких сусідніх коефіцієнтів c_i и c_{i+1}.

$$c_i = k_i - m_i c_{i+1}. \qquad (6.16)$$

Не наводячи детальних викладок, запишемо вирази для прогоночних коефіцієнтів, починаючи з k_2 и m_2.

$$k_i = \frac{r_i - \Delta x_{i-1} k_{i-1}}{z_i}; \quad m_i = \frac{\Delta x_i}{z_i},$$

де $z_i = \xi_i - \Delta x_{i-1} m_{i-1}$.

З огляду на співвідношення $c_1 = k_1 - m_1 c_2$ і гранична умова (6.15), а також вважаючи $c_2 \neq 0$, початкові коефіцієнти k_1 і m_1 задають рівними нулю.

На зворотному ходу прогонки, використовуючи прогоночні коефіцієнти за формулою (6.16), визначають значення шуканих величин c_i. При цьому знову ж з урахуванням $(c_n = k_n - m_n \cdot c_{n+1})$ і співвідношення (6.15) вважають $c_n = k_n$.

Після визначення всіх коефіцієнтів c_i, на їх основі обчислюються інші коефіцієнти сплайнів. При цьому коефіцієнти a_i визначаються як вузлові значення функції, що апроксимується: $a_i = y(x_i)$, а коефіцієнти b_i, d_i обчислюються, використовуючи вже обчислені значення c_i, за такими формулами:

$$b_i = \frac{y(x_i) - y(x_{i-1})}{\Delta x_i} - \frac{(2c_i + c_{i+1})}{3};$$

$$d_i = \frac{c_{i+1} - c_i}{\Delta x_i}.$$

Після знаходження всіх коефіцієнтів інтерполюючу функцію $S(x)$ можна розрахувати в будь-якій точці на заданому інтервалі апроксимації.

Нижче наведений приклад програмної реалізації кубічної сплайн-інтерполяції.

```
   *********************************************************
procedure Coeff_C(n:integer; var x,y,c:vector);
{ Обчислення коефіцієнтів сплайна "C"}
{ n - кількість табульованих точок; x - значення аргументу в табульованої точці; y - значення функції в табульованої точці; c - значення масиву коефіцієнтів сплайна "C", що повертається}
    var
    i:integer;
    Dx1,Dx2,ei,ri,zi:real;
    m,k:vector; //масив прогоночних коефіцієнтів
  begin
     k[1]:=0; m[1]:=0;
{Прямий хід - розрахунок прогоночних коефіцієнтів}
     for i:=2 to n do
     begin
       Dx1:=x[i]-x[i-1];
       Dx2:=x[i-1]-x[i-2];
       ei:=2*(Dx2+Dx1);
       ri:=3.0*((y[i]-y[i-1])/Dx1-(y[i-1]-y[i-2])/Dx2);
       zi:=ei-Dx2*m[i-1];
       k[i]:=(ri-Dx2*k[i-1])/zi;
       m[i]:=Dx1/zi;
     end;
 {Зворотний хід прогонки - формування масиву коеф. сплайна}
       c[n]:=k[n];c[1]:=0;
     for i:=n-1 downto 2 do
       c[i]:=k[i]-m[i]*c[i+1];// розрахунок коефіцієнтів "C"
  end;
procedure Spl (n:integer; var x,y,c:vector; x0:real; var Sp:real); { побудова сплайна }
{ x, y - значення аргументу та функції у табульованої точці, c - масив коефіцієнтів "C", знайдений процедурою Coeff_C, x0 - значення x, для якого обчислюється значення сплайна Sp}
```

```
    var
        i,j:integer;
        a,b,d:real;// коефіцієнти сплайна
        Dxi, Dx0:real;
 begin
      i:=1;
      while (x0>x[i]) and (i<>n) do
      i:=i+1; //Пошук сусіднього вузла
     { Проміжні змінні і коефіцієнти }
     Dxi:=x[i]-x[i-1];
     Dx0:=(x0-x[i-1]); //(x0-xi)
{Розрахунок коефіцієнтів сплайна на інтервалі x[i]…x[i-1],
    за знайденими процедурою Coeff_C коефіцієнтам c[i]}
    a:=y[i-1];
    b:=((y[i]-y[i-1])/Dxi)-(Dxi*(2*c[i]+c[i+1])/3);
    d:=(c[i+1]-c[i])/(3*Dxi);
    {Побудова сплайна}
     Sp:=a+(b*Dx0)+(c[i]*sqr(Dx0))+(d*Dx0*sqr(Dx0));
 end;
    uses
      Windows, Messages, …. ComObj,
    …
    var
      Form1: TForm1;
      Excel: Variant;
    …
    procedure TForm1.Button1Click(Sender: TObject);
begin
    if k=3 then
    begin // Очищення вікна інтерфейсу
    k:=0;  Series1.Clear; Series2.Clear; Series3.Clear;
    end;
    ReadExel(Rows,FileName,m);// Зчитування даних у масив m
    n:=Rows;
    for i:=1 to n do
      begin   // Перезапис масиву m в масиви x и y
      x[i-1]:=m[1,i];
      y[i-1]:=m[2,i];
      end;
Coeff_C (n,x,y,c); // Розрахунок масиву коефіцієнтів c[i]
    {Завдання аргументу X, який зростає з нуля}
    x0:=0;
 repeat
    x0:=x0+0.5;
    {Пошук інтервалу}
    i:=1;
    repeat
    z:=(x0-m[1,i])*(m[1,i+1]-x0);
```

```
      i:=i+1;
      until z>=0;
      { Розрахунок значень сплайна для кожного аргументу X0}
       Spl (n,x,y,c,X0,Sp);
      i:=i-1;
{Розрахунок значень лінійної інтерполюючої функції для кож-
ного аргументу X0 з метою проведення порівняльного аналізу}
y0:=m[2,i]+((m[2,i+1]-m[2,i])/
    (m[1,i+1]-m[1,i]))*(x0-m[1,i]);
{Графічне виведення інтерполюючої функцій y0 = f (x0) і
сплайна Sp = f (x0) для проведення порівняльного аналізу }
      if RadioButton1.Checked then
      Series1.AddXY(x0,(y0));
      if RadioButton3.Checked then
      Series3.AddXY(x0,(y0));
      if RadioButton2.Checked then
      Series2.AddXY(x0,(Sp));
 until x0>=Xk;
k:=k+1;
 end;
end.
```

****

Рис. 6.8 ілюструє результати роботи програми з інтерполяції фрагменту табличне заданої функції, наведеної на рис. 6.5. Для порівняльного аналізу на рис. 6.8 наведені також результати лінійної інтерполяції з різною частотою табулювання.

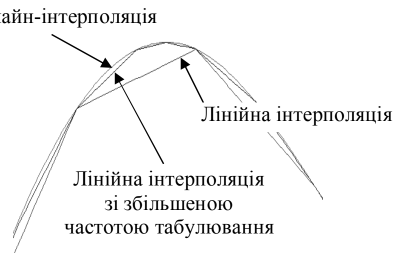


Рис. 6.8

Сплайн-інтерполяція широко використовується при моделюванні, в тому числі і при дослідженнях різних режимів польоту літака. Наприклад, використовуючи сплайн-інтерполяцію можна змоделювати рельєф підстильної місцевості, а наклавши на інтерполюючу функцію шумову складову, максимально наблизити отриману модель до реального рельєфу. Побудовану модель потім можна використовувати при дослідженнях маловисотних режимів польоту.

6.4. Чисельне диференціювання та інтегрування функцій

Диференціювання та інтегрування є окремими випадками функцій, визначених на функціональному просторі[6.3]. При цьому кожній функції функціонального простору ставиться у відповідність або знову функція (при відшукуванні похідної або невизначеного інтеграла), або деяке число (якщо розшукається похідна в певній точці або визначений інтеграл). У багатьох випадках значення цих функцій не можуть бути знайдені точно з використанням методів диференціального та інтегрального обчислення. Тоді вдаються до наближеного розв‘язання цих завдань, використовуючи методи чисельного диференціювання та інтегрування, що засновані на заміні одного функціонального простору іншим. В основі заміни лежить вже розглянутий раніше метод наближення – інтерполювання.

6.4.1. Чисельне диференціювання

Чисельне диференціювання це сукупність методів обчислення значення похідної функції $y = f(x)$ в заданих точках у випадках, коли аналітичний вигляд функції $f(x)$ невідомий (функція задана неявно), або дуже складний, або функція $f(x)$ задана таблицею. В основі чисельного диференціювання функції $f(x)$, заданою таблицею лежить апроксимація функції, від якої береться похідна, інтерполяційним поліномом.

Похідна функції це границя відношення приросту функції до приросту незалежної змінної при прагненні до нуля приросту незалежної змінної:

$$\frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x}.$$

При чисельному диференціюванні заміняють відношення нескінченно малих приростів функції й аргументу відношенням кінцевих різниць:

$$\frac{dy}{dx} \approx \frac{\Delta y}{\Delta x}.$$

[6.3] Функціональний простір це сукупність функцій з визначеним для них поняттям близькості.

Очевидно, що чим менше буде приріст аргументу, тим точніше чисельне значення похідної.

Привабливість чисельного підходу пояснюється наявністю простих формул, за допомогою яких похідні в заданих точках можна приблизно обчислити за кількома значеннями функції в цих та близьких до них точках. При цьому точність чисельного диференціювання залежить від кількості задіяних точок.

Найчастіше при реалізації чисельного диференціювання використовують двоточкові методи.

Для двоточкових методів при обчисленні похідних використовується значення функції у двох точках. Приріст аргументу задається трьома способами, відкладаючи Δx вправо, уліво й в обидва боки від досліджуваної точки. Відповідно виходить три двоточкових методи чисельного диференціювання:

1 – диференціювання вперед $\dfrac{dy}{dx} \approx \dfrac{\Delta y}{\Delta x} = \dfrac{y(x+\Delta x)-y(x)}{\Delta x}$;

2 – диференціювання назад $\dfrac{dy}{dx} \approx \dfrac{\Delta y}{\Delta x} = \dfrac{y(x)-y(x-\Delta x)}{\Delta x}$;

3 – симетричне диференціювання $\dfrac{dy}{dx} \approx \dfrac{\Delta y}{\Delta x} = \dfrac{y(x+\Delta x)-y(x-\Delta x)}{2\Delta x}$.

Суть зазначених методів проілюстрована на рис. 6.9. Чисельне значення тангенса кута α утвореного дотичної до графіка $y = f(x)$ і віссю абсцис, показує точне значення похідної (геометричний зміст похідної). Тангенси кутів α_1, α_2, α_3 відповідають наближеним значеннями похідних, невизначених методами 1, 2, 3 відповідно. При цьому $\Delta x = h$ називають кроком чисельного диференціювання.

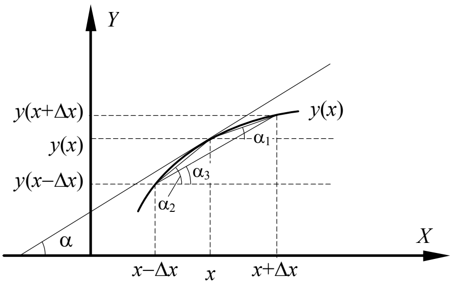


Рис. 6.9

Похибка чисельного диференціювання визначається двома внесками. Перший внесок Δ_1 називається похибкою апроксимації і він пов'язаний із заміною оператора диференціювання оператором кінцевої різниці. Цей внесок в похибку залежить від інтенсивності зміни функції $f(x)$ на цьому кроці, а саме від $f''(x)$, а також від кроку чисельного диференціювання h:

$$\Delta_1 = \max\left|f''(x)\right| \cdot \frac{h}{2}.$$

Похибка апроксимації Δ_1 прямує до нуля при $h \to 0$.

Другий внесок у похибку обчислень – Δ_2 пов'язаний з неточністю машинних обчислень, тобто з помилками округлення (машинною похибкою $\delta_м = 10^{-8}\ldots10^{-4}$):

$$\Delta_2 = \frac{2\delta_м}{h}.$$

Похибка округлення Δ_2 при $h \to 0$ прямує до нескінченності.

З мінімуму сумарної похибки як функції кроку:

$$\Delta_1 - \Delta_2 = 0,$$

можна визначити оптимальний крок чисельного диференціювання:

$$h_{опт} = 2\sqrt{\frac{\delta_м}{\max\left|f''(x)\right|}}.$$

На практиці значення h обирають залежно від величини точки x_0, функції $f(x)$, для якої потрібно розрахувати похідну, наприклад, $h = h_R \cdot |x_0|$, де h_R знаходиться у межах від 10^{-6} до 10^{-2} залежно від поведінки функції $f(x)$ в околицях точки x_0.

При табличному задаванні функції значення похідних розраховуються тільки у вузлових точках, тому крок чисельного диференціювання визначається частотою табулювання.

Існують також розрахункові формули для триточкової схеми чисельного диференціювання. Але у практиці модулювання алгоритмів інформаційних систем цивільної авіації вони не застосовуються.

Так в алгоритмах цифрової системи повітряних сигналів СВС-2Ц-1М для визначення вертикальної швидкості V_y здійснюється чисельне диференціювання поточних значень абсолютної барометричної висоти з використанням 2-го методу двоточкового чисельного диференціювання:

$$V_y \approx \frac{\Delta H_{абс}}{\Delta t} = \frac{H_{абс}(t) - H_{абс}(t-h)}{h},$$

де $H_{\text{абс}}(t-h)$, $H_{\text{абс}}(t)$ – значення абсолютної барометричної висоти, що обчислені на попередньому та на поточному циклах обчислень відповідно; h – крок чисельного диференціювання, який визначається частотою опитування датчика статичного тиску, за інформацією якого за табличними градуйовочними залежностями визначаються значення абсолютної барометричної висоти.

Спрощена блок-схема алгоритму визначення вертикальної швидкості V_y з використанням методу чисельного диференціювання зображена на рис. 6.10.

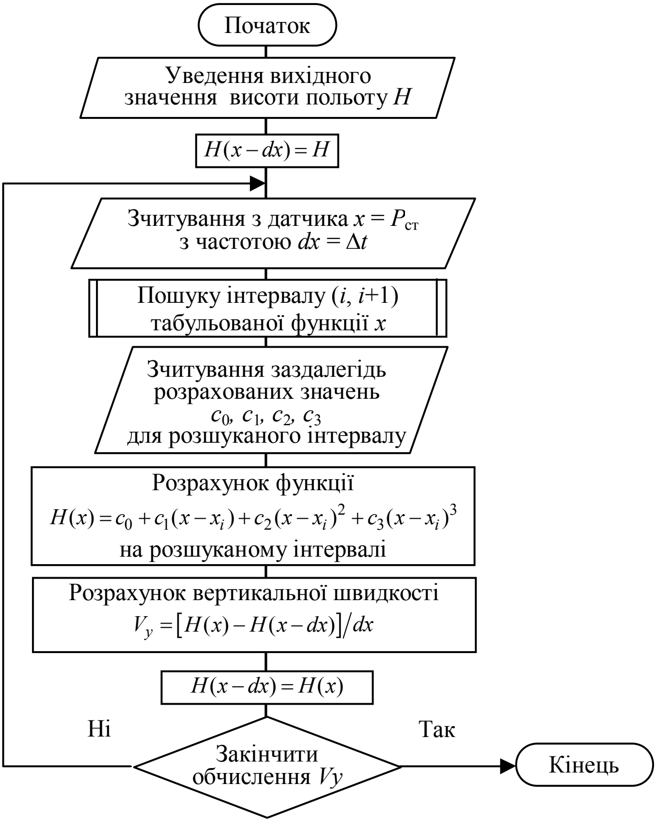


Рис. 6.10

6.4.2. Чисельне інтегрування

Визначеним інтегралом $S = \int_a^b f(x)dx$ від безперервної функції $f(x)$ на кінцевому відрізку $[a, b]$ називається кінцева границя її інтегральної суми, при прагненні до нескінченності числа ділянок розбиття, довжина яких прагне до нуля. Для знаходження точного значення визначеного інтеграла за формулою Ньютона-Лейбніца необхідно первісну підінтегральну функції надати у вигляді елементарної функції, що у багатьох випадках не вдається. У цьому випадку використовують методи чисельного інтегрування, які дозволяють отримати значення визначеного інтеграла з необхідним ступенем точності.

Значення визначеного інтеграла $S = \int_{x_н}^{x_к} f(x)dx$ з межами інтегрування $x_н$ и $x_к$ можна трактувати як площину криволінійної трапеції, що виділена на рис. 6.11 штрихуванням.

Задача чисельного інтегрування полягає у заміні інтеграла $S = \int_{x_н}^{x_к} f(x)dx$ кінцевими сумами виду, $S \approx \sum_{i=1}^{n} c_i f(x_i)$, званими квадратурними формулами, де x_i – вузли, а c_i – ваги квадратурної формули.

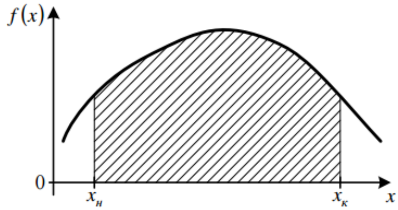


Рис. 6.11

В основі методів чисельного інтегрування лежить розбиття інтервалу, на якому виконується інтегрування, не кілька відрізків і визначення суми площ простих фігур на отриманих відрізках. Найпоширенішими методами чисельного інтегрування є: методи прямокутників, метод трапецій та метод Сімпсона.

<u>Методи прямокутників</u>

Методи прямокутників відносяться до найпростіших методів чисельного інтегрування. Методи прямокутників дозволяють визначити значення інтеграла як суму площ прямокутників, одна зі сторін яких є кроком (відрізком інтервалу) інтегрування h – відстанню між вузлами $(x_{i+1} - x_i)$.

Якщо за другу сторону відрізку використовується значення функції на лівій межі відрізка (див. рис. 6.12), то наближене значення інтеграла, обчислюється методом лівих прямокутників, формула якого записується у вигляді:

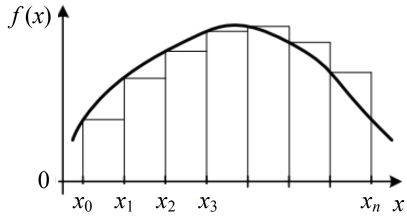


Рис. 6.12

$$S \approx \sum_{i=0}^{n-1} (x_{i+1} - x_i) f(x_i).$$

При сталому кроку інтегрування формула чисельного інтегрування за методом лівих прямокутників спрощується:

$$S \approx h \sum_{i=0}^{n-1} f(x_i).$$

Якщо в якості другої сторони відрізка використовується значення функції на правій межі відрізка (див. рис. 6.13), то наближене значення інтеграла, обчислюється методом правих прямокутників, формула якого записується у вигляді:

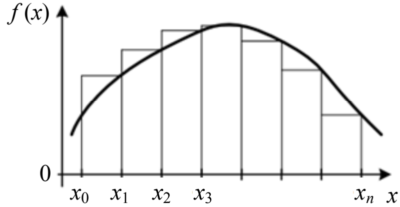


Рис. 6.13

$$S \approx \sum_{i=1}^{n} (x_i - x_{i+1}) f(x_i),$$

а при сталому кроці інтегрування:

$$S \approx h \sum_{i=1}^{n} f(x_i).$$

Якщо друга сторона відрізка є значенням функції на середині відрізка (рис. 6.14), то наближене значення інтеграла, обчислюється методом середніх прямокутників, формула якого записується у вигляді:

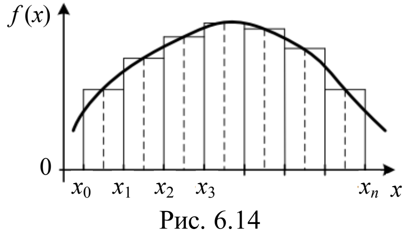


Рис. 6.14

$$S \approx \sum_{i=0}^{n-1}(x_{i+1}-x_i)\, f\left(\frac{x_i+x_{i+1}}{2}\right).$$

Для сталого кроку інтегрування дану формулу можна перетворити до вигляду:

$$S \approx h\sum_{i=0}^{n-1} f(x_i+\frac{h}{2}).$$

Точність інтегрування методами прямокутників визначається кроком інтегрування. Чим менше крок, тим точніше результат чисельного інтегрування.

Метод трапецій

Метод трапецій дозволяє визначити значення інтеграла як суму площ трапецій, на які розбитий інтервал інтегрування (рис. 6.15). Оскільки площа S_i трапеції $x_iP_iP_{i+1}x_{i+1}$ на інтервалі x_i, x_{i+1} дорівнює

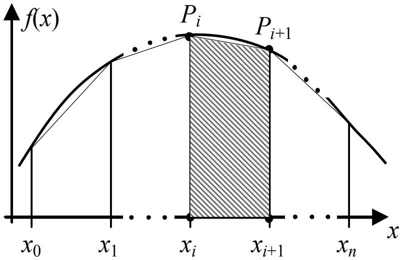


Рис. 6.15

$$S_i = (x_{i+1}-x_i)\frac{f(x_{i+1})+f(x_i)}{2},$$

то, складаючи площі всіх прямолінійних трапецій, одержимо наближене значення інтеграла, що обчислене методом трапецій:

$$S \approx \sum_{i=1}^{n}(x_{i+1}-x_i)\frac{f(x_{i+1})+f(x_i)}{2}.$$

При використанні сталого кроку інтегрування ($h = x_{i+1}-x_i$) формула обчислення інтеграла методом трапецій може бути перетворена до виду:

$$S \approx h\left(\frac{f(x_0)+f(x_n)}{2}+\sum_{i=1}^{n} f(x_i)\right).$$

Метод трапецій дає більш точний результат, ніж методи прямокутників.

Метод Сімпсона

Значне підвищення точності наближених формул чисельного інтегрування надає метод Сімпсона (метод парабол). Ідея методу базується на тому, що на окремому інтервалі визначення функції

$y = f(x)$ дуга певної параболи у загальному випадку тісніше примикає до кривої $y = f(x)$, ніж хорда, що з'єднує кінці дуги цієї кривою (метод трапецій). Тому значення площ, які відповідають елементарним трапеціям, обмежених зверху дугами парабол, є більш близькими до значень площ, що відповідають криволінійним трапеціям, обмеженим зверху кривою $y = f(x)$, ніж значення площ, які відповідають прямолінійним трапеціям.

Суть методу полягає в тому, що інтегрована функція $f(x)$ (рис. 6.16) замінюється на кожному n-ому відрізку інтервалу інтегрування (на кожному кроці інтегрування h), фрагментом параболи, яка проходить через початок x_i, кінець (x_{i+1}) і середину (x_i+0,5h) відрізку (рис. 6.16). Інтеграли, від фрагментів парабол другого ступеня, обчислюються за формулою Ньютона-Лейбніца.

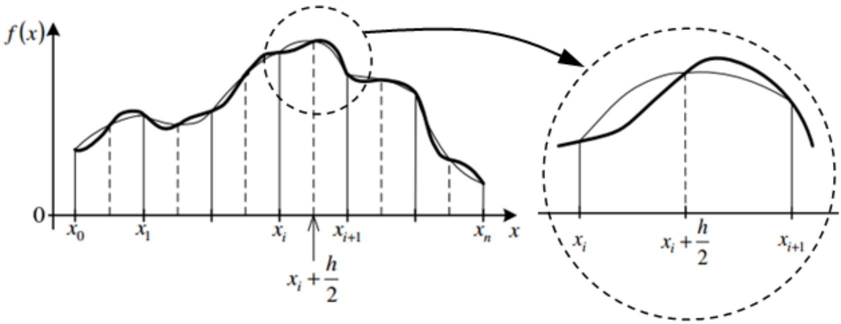


Рис. 6.16

Обов'язковою вимогою, яка випливає з геометричного змісту методу парабол, є те, що n повинно бути ***парним***.

Опускаючи висновки, формулу Сімпсона у разі рівномірного кроку інтегрування можна подати у вигляді:

$$S = \int_{x_0}^{x_n} f(x)dx \approx \frac{h}{3}\left(f(x_0) + 2\sum_{j=1}^{n-1} f(x_{2j}) + 4\sum_{j=1}^{n} f\left(x_{2j-1}\right) + f(x_n) \right).$$

Фрагмент програми інтегрування нелінійної функції $f(x) = \dfrac{x}{4 + x^4}$ з використанням методу Сімпсона для середовища програмування Delphi наведено нижче.

```
***********************************************************
function F(x:real):real;
  begin
   F:=x/(4+(x*x*x*x));
  end;
procedure TForm1.Button1Click(Sender: TObject);
begin
 n:=StrToInt(edit1.Text);//кількість відрізків розбиття
 a:=StrToInt(edit2.Text);//початок інтервалу інтегрування
 b:=StrToInt(edit3.Text);//кінець інтервалу інтегрування
 h:=(b-a)/(2*n);         //крок інтегрування
 sum2i:=0;  sum2i_1:=0;  //обнуління вихідних значень
    for i:= 1 to (n-1) do //перебір парних вузлів X2i
   begin
    x2i:=a+(2*i)*h;
    sum2i:=sum2i+F(x2i);
   end;
     for i:= 1 to n do     // перебір непарних вузлів X2i-1
    begin
    x2i_1:=a+((2*i)-1)*h;
    sum2i_1:=sum2i_1+F(x2i_1);
    end;
    s0:= F(a);
    sn:= F(b);
s:=(h/3)*( s0+ sn +2*sum2i+4*sum2i_1); //формула  Сімпсона
{Результат інтегрування}
    edit4.text:= 's = '+floattostrf(s, fffixed, 10,8); //
end;
*********************************************************
```

Метод Сімпсона дає найбільш точний результат інтегрування, у порівнянні з методами трапецій та прямокутників.

<u>Метод сплайн-квадратур</u>

Раніше була розглянута лінійна інтерполяція, а також більш точна, хоча й більш складна сплайн-інтерполяція нелінійної функції. Природним є прагнення і при чисельному інтегруванні скористатися положеннями теорії сплайнів. Метод обчислення певних інтегралів, що базується на теорії сплайнів, називається методом сплайн-квадратури.

Припустимо необхідно обчислити інтеграл вигляду:

$$S = \int_{x_0}^{x_n} f(x)dx \, .$$

При використанні сплайн-квадратури інтервал $[x_0, x_n]$ розбивається на ділянки ($h_i = x_i - x_{i-1}$, $i = \overline{1,n}$, на кожному з яких підінтегральна функція $f(x)$ замінюється кубічним поліномом $\varphi_i(x)$:

$$\varphi_i(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3,$$

де $x \in [x_{i-1}, x_i]$.

Тоді інтеграл S запишеться як сума інтегралів кубічних поліномів сплайна:

$$S \approx \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} \varphi_i(x)dx = \sum_{i=1}^{n} (a_i h_i + \frac{b_i h_i^2}{2} + \frac{c_i h_i^3}{3} + \frac{d_i h_i^4}{4}).$$

Дана формула спрощується при застосуванні виразів для коефіцієнтів a, b, и d:

$$S \approx \sum_{i=1}^{n} \frac{h_i(f_i - f_{i-1})}{2} - \sum_{i=1}^{n} \frac{h_i^3(c_i + c_{i+1})}{12}.$$

Нескладно побачити, що перша сума цієї формули є формулою трапецій, а друга сума – це поправочна складова для формули трапецій, що застосована до сплайнів. При малих значеннях h_i коефіцієнти c_i и c_{i+1} близькі за величиною, а коефіцієнт $d_j \to 0$, тому

$$c_i + c_{i+1} \approx \varphi''(x), \quad x \in [x_{i-1}, x_i].$$

Виходить, що похибка сплайн-квадратури менше, ніж похибка методу трапецій. Однак алгоритм сплайн-квадратури складніше алгоритмів методів трапецій і алгоритму Сімпсона за рахунок необхідності визначення коефіцієнтів сплайна a_i, b_i, c_i, d_i на кожної i-тої ділянці. Тому раціонально використовувати сплайн-квадратури тоді, коли сплайни вже застосовуються для інтерполяції складних залежностей, тобто тоді коли коефіцієнти сплайна вже розраховані.

Методи чисельного інтегрування використовуються, наприклад, при моделюванні аеродинаміки крила складної конфігурації, площа якого може бути розрахована методами чисельного інтегрування.

6.5. Програмні реалізації методів розв‘язання звичайних диференціальних рівнянь

Програмні засоби ЦОМ надають широкі можливості для дослідження характеристик безперервних систем, математичний опис яких зазвичай задається у вигляді звичайних диференціальних рівнянь.

Звичайним диференціальним рівнянням (ЗДУ) *n-го* порядку називається рівняння вигляду:

$$F(x, y, y', y'', ..., y^{(n)}) = 0, \tag{6.17}$$

де F – відома функція $(n + 2)$-х змінних, x – незалежна змінна на інтервалі (a,b), $y(x)$ – невідома функція, яка називається розв’язком (або інтегралом) ЗДУ на інтервалі (a, b), і яка при підстановці в рівняння перетворює його у тотожність.

Диференціальні рівняння, розв’язанні відносно старшої похідної, називають рівняннями у канонічному вигляді:

$$y^{(n)} = f(x, y, y', y'', ..., y^{(n-1)}).$$

Це рівняння і еквівалентне йому рівняння (6.17) мають безліч розв’язків. Щоб виділити єдиний розв’язок рівняння n–го порядку задають n початкових умов: $y(x_0) = y_0$, $y'(x_0) = y_1$, $y''(x_0) = y_2$, … , $y^{(n-1)}(x_0) = y_{n-1}$, які називають даними Коші. Задачу пошуку розв’язку $y = y(x)$, що задовольняє початковим умовам називають задачею Коші.

Для розв’язку задачі Коші використовуються різні моделюючі алгоритми, що засновані на методах чисельного інтегрування. Чисельні методи розв'язання задач Коші виявляються більш ефективними навіть при наявності аналітичних рішень через відомі функції.

6.5.1. Моделюючі алгоритми

При дослідженні систем ЦА застосовуються в основному динамічні моделі, що описуються ЗДУ, для яких незалежної змінної x (аргументом похідних) є час. Моделюючі алгоритми чисельного розв’язку таких ЗДУ будуються на принципі Δt і можуть бути реалізовані з використанням універсальних мов програмування (Паскаль, С ++ та ін.) або проблемно-орієнтованих мов.

При використанні методів чисельного інтегрування ЗДУ довільного ступеня, що записані в канонічному вигляді:

$$y^{(n)} = f(t, y, y', y'', ..., y^{(n-1)}),$$

подаються у вигляді системи ЗДУ першого порядку, яка має нормальну форму Коші:

$$\frac{dy_1}{dt} = f_1(y_1(t), y_2(t), ...y_n(t)),$$

$$\frac{dy_2}{dt} = f_2(y_1(t), y_2(t), ...y_n(t)),$$

....

$$\frac{dy_n}{dt} = f_n(y_1(t), y_2(t), ...y_n(t)).$$

При формулюванні задачі Коші система доповнюється вихідними умовами та вхідними впливами, які можуть бути як керуючими, так і збурюючими.

Розв’язок в дискретних точках часової осі t_0, t_1, t_2 ... можливий з довжиною кроку $h_n = t_{n+1} - t_n$ і базується на покрокових алгоритмах, які називаються різницевими схемами.

Розглянемо застосування деяких чисельних методів розв’язку (інтегрування) ЗДУ на прикладі рівняння

$$y' = \frac{dy}{dt} = f(t, y).$$

Історично першим і найбільш простим способом чисельного розв'язання задачі Коші для ЗДУ першого порядку є метод Ейлера. В його основі лежить апроксимація похідної відношенням кінцевих збільшень залежною (y) і незалежної (t) змінних між вузлами Δt часової осі рівномірної сітки:

$$y' = \frac{dy}{dt} \approx \frac{\Delta y}{\Delta t} = \frac{y_{i+1} - y_i}{t_{i+1} - t_i} = f(t, y),$$

де y_{i+1} – шукане значення функції в точці t_{i+1}.

Перетворюючи дане рівняння і враховуючи на рівномірність сітки (кроку) інтегрування $h = t_{i+1} - t_i$, отримаємо ітераційну формулу Ейлера для обчислення y_{i+1} за відомим значенням y_i у точці t_i:

$$y_{i+1} = y_i + h \cdot f(t, y).$$

Аналіз формули Ейлера показує, що для наближеного обчислення інтеграла в методі Ейлера використовується найпростіша формула інтегрування – формула методу лівих прямокутників. Якщо шукана функція на відрізку інтегрування сильно відрізняється від лінійної, то похибка обчислення буде значною.

Приклад

Дослідити методом аналітичного моделювання динаміку поведінки літака при ступінчастому відхиленні керма висоти Δδв.

Лінеаризовані рівняння поздовжнього короткоперіодичного руху літака мають вигляд:

$$\begin{aligned} &\Delta\dot{\Theta} = a_y^{\alpha}\Delta\alpha; \\ &\Delta\dot{\omega}_z = a_{m_z}^{\alpha}\Delta\alpha - a_{m_z}^{\omega_z}\Delta\omega_z + a_{m_z}^{\delta_\text{в}}\Delta\delta_\text{в}; \\ &\Delta\dot{\vartheta} = \Delta\omega_z; \\ &\Delta\alpha = \Delta\vartheta - \Delta\Theta, \end{aligned} \tag{6.18}$$

де ΔΘ ,$\Delta\omega_z$, Δϑ, Δα – малі прирости параметрів поздовжнього руху (кута нахилу траєкторії, кутової швидкості тангажа, кута тангажа та кута атаки відповідно).

Блок-схема моделюючого алгоритму для розв'язку рівнянь поздовжнього короткоперіодичного руху літака методом Ейлера має вигляд (рис. 6.17). Вихідними даними для проведення моделювання є:

- крок інтегрування – sh;
- час інтегрування – Tm;
- значення коефіцієнтів моделі;
- початкові значення Δωz, Δα, Δϑ, ΔΘ, наприклад, задаються рівними нулю;
- величина збурюючого впливу (відхилення руля висоти $\Delta\delta_в$).

Фрагменти програми моделюючого алгоритму наведені нижче.

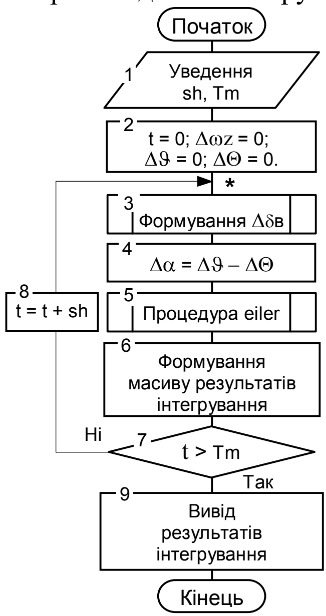


Рис. 6.17

Всі рівняння, що підлягають інтегруванню (у нашому випадку перші три рівняння системи (6.18)), записуються у вигляді функцій елементів px[i] = f(x[i]) масивів px, x и оформляються окремою процедурою, в нашому прикладі процедурою ody (var x, px: mat). Процедура опису правих частин рівнянь поздовжнього короткоперіодического руху літака на мові Paskal має вигляд:

```
*************************************************
procedure ody(var x,px:mat);
{x,px – масиви значений функцій та їх похідних;
mat = array[1..n] of real; (у даному випадку n = 3)}
begin
  px[1]:=-a11*alf;
  px[2]:=-a21*alf-a22x[2]+a23*db;
  px[3]:=x[2];
end;
{x [1] - кут нахилу траєкторії, x [2] - кутова швидкість
тангажа, x [3] - кут тангажа; px [i] - похідні цих парамет-
рів; a11, a21, a22, a23 - коефіцієнти моделі; db - іденти-
фікатор відхилення руля висоти, alf – ідентифікатор кута
атаки (обчислюється в основній програмі)}
*******************************************************
```

Процедура чисельного розв'язання диференціальних рівнянь поздовжнього короткоперіодичного руху літака методом Ейлера на мові Paskal має вигляд:

```
*****************************************************
procedure eiler(var x,px:mat); //Метод Эйлера
var
  i:integer;
begin
  for i:=1 to n do
  begin
{звернення до процедури ОДУ для обчислення функцій px}
  ody (x,px);
{Інтегруванні з кроком sh і формування масиву результа-
тів
 інтегрування на даному етапі}
   x[i]:=x[i]+sh*px[i];
   end;
end;
*************************************************************
```

Основна процедура моделюючого алгоритму запускається натисканням кнопки Button1 на панелі інтерфейсу форми Form1 середовища програмування Делфі.

```
*************************************************
procedure TForm1.Button1Click(Sender: TObject);
begin
Tm:=StrToFloat(Edit3.Text);//Завдання часу моделювання
sh:=StrToFloat(Edit2.Text);// Завдання кроку
      { Завдання коефіцієнтів моделі}
a11:= 0.59; a21:= 5.28; a22:= 0.5; a23:= 1.5;
t:=0; stop:=0;   // Обнулення лічильника часу
        for i:=1 to 3 do// Обнулення масивів
      begin
        x[i]:=0; px[i]:=0;
      end;
            {Очищення вікна графіків}
Series1.Clear; Series2.Clear; Series3.Clear; Series4.Clear;
    {Початок нескінченного циклу}
    Repeat
   db:=1; // формування ступінчастого відхилення руля висо-
ти
   alf:= x[3]-x[1]; // Розрахунок кута атаки
    {Графічне виведення результатів інтегрування}
         Series1.AddXY(t, z[1]);  Series2.AddXY(t, x[2]);
         Series3.AddXY(t, x[3]); Series4.AddXY(t, x[4]);
       t:=t+sh; // Лічильник часу
{Звернення до процедури інтегрування, яка в свою чергу
звертається до процедури ody - процедурі опису правих
частин рівнянь }
       eiler(x,px);
    {Зупинка процедури після закінчення часу інтегрування}
       if t>=Tm then stop:=1;
    Until   stop=1;
   end;
***********************************************************
```

Метод Ейлера є методом першого порядку – має похибку на кроці $O(h^2)$ і похибка в цілому $O(h)$.

Підвищити точність обчислень можна за допомогою модифікованого методу Ейлера з перерахунком наступного виду:

прогноз: $\tilde{y}_{n+1} = y_n + hf(x_n, y_n)$;

корекція: $y_{n+1} = y_n + h\dfrac{f(x_n, y_n) + f(x_{n+1}, \tilde{y}_n)}{2}$.

Однак цей метод вже є різновидом методів Рунге-Кута, які базуються на формулі інтегрування Сімпсона, що використовує значення підінтегрального виразу у трьох точках. У методах Рунге-Кута в якості додаткової точки використовують середину ($0{,}5h$) кроку інтегрування.

При використанні різних методів наближеного обчислення значень функцій $f(y_{n+0,5h})$ і $f(y_{n+h})$ в точках $x_{n+0,5h}$ і x_{n+h}, отримують вирази для методів Рунге-Кута різного порядку точності.

В обчислювальній практиці найчастіше використовується метод Рунге-Кута четвертого порядку:

$$y_{n+1} = y_n + \frac{1}{6}(k_0 + 2k_1 + 2k_2 + k_3),$$

де $k_1 = h \cdot f(y_n, x_n)$; $k_2 = h \cdot f(y_n + \dfrac{k_1}{2},\ x_n + h/2)$;

$k_3 = h \cdot f(y_n + \dfrac{k_2}{2},\ x_n + h/2)$; $k_4 = h \cdot f(y_n + k_3,\ x_n + h)$.

Тут на кожному кроці розв‘язання потрібно чотири обчислення функції $f(y,\ x)$, що може бути трудомістким. Методи Рунге-Кута легко програмувати, вони стійкі для широкого класу задач і є такими, що можуть «стартувати» самостійно, тобто для обчислення y_{n+1} необхідно лише попереднє значення y_n.

Процедура чисельного розв‘язання диференціальних рівнянь поздовжнього короткоперіодичного руху літака для попереднього прикладу методом Рунге-Кута на мові Paskal має вигляд:

```
***********************************************************
       procedure rk(var x,px:mat); //Метод Рунге-Кута
       var
         z,k1,k2,k3,k4 :mat;
begin
    for i:=1 to n do //n - розмір масиву mat
    begin
     z[i]:=x[i];
  ody (x,px); //1-е звернення до процедури оду
         k1[i]:=sh*px[i]; x[i]:=x[i]+k1[1]/2;
  ody (x,px); //2-е звернення до процедури оду
         k2[i]:=sh*px[i]; x[i]:=x[i]+k2[i]/2;
  ody (x,px); //3-е звернення до процедури оду
         k3[i]:=sh*px[i]; x[i]:=x[i]+k3[i];
  ody (x,px); //4-е звернення до процедури оду
         k4[i]:=sh*px[i];
{Формування масиву результатів інтегрування на даному етапі}
     x[i]:=z[i]+(1/6)*(k1[i]+2*k2[i]+2*k3[i]+k4[i]);
    end;
end;
***********************************************************
```

Розглянуті методи Ейлера і Рунге-Кута використовують значення функції на одному попередньому кроці, тому вони відносяться до так званих однокрокових методів. Точність обчислень можна підвищити, за рахунок застосування ***багатокрокових методів,*** в яких при знаходженні розв‘язку в певній точці x_n використовується інформація про значеннях функції f_{n-1}, f_{n-2}, ... в попередніх точках x_{n-1}, x_{n-2}, ..., де $f_i = f(x_i, t_i)$. Інтеграл при цьому обчислюється з використанням квадратурної формули:

$$y_{i+1} \approx y_i + h\sum_{j=1}^{k} \lambda_j f(x_{i+1-j}, y_{i+1-j}),$$

де λ_j – квадратурні коефіцієнти.

Отримане таким чином сімейство формул називається явною k-крокової схемою Адамса (методи Адамса-Башфорта).

Наприклад, чотирьох крокова явна формула Адамса має вигляд:

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}),$$

де f_n, f_{n-1}, f_{n-2}, f_{n-3} – значення функції $f(x,y)$, що розраховані на чотирьох попередніх кроках.

Якщо для побудови інтерполяційного полінома використовувати k вузлів, починаючи з x_{i+1}, то можна отримати формули інтегрування ЗДУ, відомі як неявні схеми Адамса (або методи Адамса-Моултона). Неявними ці формули називаються тому, що значення шуканої функції y_{i+1} в (i+1)-м вузлі виявляється одночасно і в лівій, і в правій частинах рівності:

$$y_{i+1} \approx y_i + h\sum_{j=0}^{k} \lambda_j f(x_{i+1-j}, y_{i+1-j}).$$

Наприклад, чотирьох крокова неявна формула Адамса-Моултона має вигляд:

$$y_{n+1} = y_n + \frac{h}{24}\left[9f(x_{n+1}, y_{n+1}) + 19f(x_n, y_n) - 5f(x_{n-1}, y_{n-1}) + f(x_{n-2}, y_{n-2})\right].$$

Це рівняння є неявним відносно y_{n+1}, оскільки y_{n+1} зустрічається і в лівій, і правій його частині. Зазвичай таке рівняння не розв‘язується, а використовується метод "прогноз-корекції". У методі "прогноз-корекції" значення y_{n+1} в правій частині рівняння замінюється на розраховане за будь-якою явною формулою, наприклад, за формулою Адамса-Башфорта, званої в даному випадку предиктором (прогнозом).

Як приклад методу "прогнозу-корекції" наведемо метод Адамса четвертого порядку:

– предиктор

$$y_{n+1} = y_n + \frac{h}{24}(55f_n - 59f_{n-1} + 31f_{n-2} - 9f_{n-3}); \qquad (6.19\ а)$$

– коректор

$$y_{n+1} = y_n + \frac{h}{24}(9f_n + 19f_{n-1} - 5f_{n-2} + f_{n-3}). \qquad (6.19\ б)$$

Обчислення y_{n+1} виробляються в даному випадку в такій послідовності: на початку визначається початкове наближення y_{n+1} за формулою (6.19 *а*); потім обчислюється значення функції $f_{n+1} = f(x_{n+1}, y_{n+1})$, а потім поклавши $i + 1 = i$, визначається більш точне значення y_{n+1} за формулою (6.19 *б*).

Перевага багатокрокових методів Адамса при розв‘язанні ЗДУ полягає в тому, що в кожному вузлі розраховується тільки одне значення правої частини ЗДУ – функції $f(x,y)$. До недоліків можна віднести неможливість старту багатокрокового методу з єдиної початкової точки, оскільки для обчислень за k-крокової формулою необхідне знання значення функції в k вузлах. Тому доводиться $(k–1)$ розв‘язок в перших вузлах x_1, x_2, …, x_{k-1} отримувати за допомогою будь-якого однокрокового методу, наприклад за допомогою методу Рунге-Кута 4-го порядку.

До методів чисельного розв‘язку диференціальних рівнянь також відносяться методи екстраполяції, які базуються на алгоритму виду (6.19 *а*), де y_{n+1} екстраполюється за раніш виявленими попередніми значенням y_{n-i} та f_{n-i}, $i=\overline{1,k}$.

Похибки методів чисельного інтегрування діляться на два види: похибки дискретизації, які залежать від вибору кроку і похибки обраного методу розв'язання. Оцінка точності обраного методу визначається порядком локальної та глобальної похибки. При цьому локальної похибкою вважається похибка на даному етапі обчислень за умови, що попередні значення точні й немає похибок округлення, які пов'язані з кроком інтегрування. Глобальна ж похибка це похибка в цілому (накопичена похибка) – похибка в останній точці довільного кінцевого відрізку інтегрування.

Важливим показником ефективності чисельного методу розв'язання диференціальних рівнянь є його стійкість, тобто спадання глобальної похибки з ростом t. Стійкість залежить від знаку похідної $\partial f/\partial x$, який може змінюватися для нелінійних рівнянь, тобто рівняння може бути стійким в одних областях і нестійким в інших. Для системи рівнянь ситуація більш складна.

При аналітичному моделюванні систем з випадковими вхідними впливами $u(t_n)$ використовуються стохастичні диференціальні рівняння, чисельне розв‘язання яких дозволяє знайти реалізацію випадкового процесу $y(t_n)$ за формованою в процесі інтегрування реалізацією випадкового вхідного впливу.

6.6. Програмна реалізація випадкових факторів при аналітичному моделюванні стохастичних систем

Математичне моделювання дозволяє досліджувати поведінку різних систем з урахуванням впливу випадкових факторів. При аналітичному моделюванні систем з випадковими вхідними впливами використовуються стохастичні диференціальні рівняння (СДУ) виду:

$$\dot{x} = A(x) + B(u) + \varpi(t).$$

Неважко бачити, що відмінність СДУ від ЗДУ полягає в наявності в правій частині додаткової складової $\varpi(t)$, що являє собою певний випадковий процес, який може розглядатися як вхідний вплив. Саме випадкові процеси є основними випадковими чинниками при аналітичному моделюванні стохастичних систем. Це можуть бути випадкові зовнішні або вхідні впливи, похибки або перешкоди вимірювань, тощо.

В основі методів моделювання випадкових факторів лежить використання рівномірно розподілених на інтервалі [0; 1] випадкових чисел (ВЧ), які генеруються певними способами.

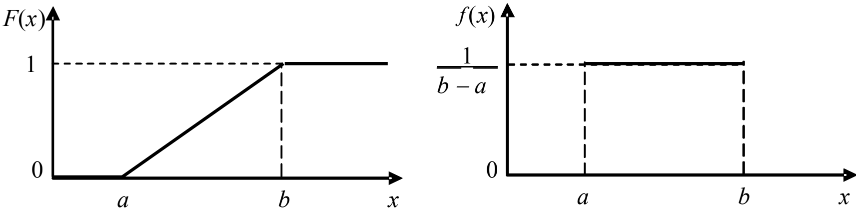
Безперервна випадкова величина має рівномірний розподіл в інтервалі $[a,\ b]$ якщо її функція розподілу[6.4] $F(x)$ і функція щільності [6.5] $f(x)$ мають вигляд:

$$F(x) = \begin{cases} 0, & \text{при } x < a \\ \dfrac{x-a}{b-a}, & \text{при } a \le x \le b \\ 1, & \text{при } x > b \end{cases} \quad f(x) = \begin{cases} 0, & \text{при } x < a \\ 1/(b-a), & \text{при } a \le x \le b \\ 0, & \text{при } x > b \end{cases}.$$

Графічне подання функцій $F(x)$ и $f(x)$ безперервної рівномірно розподіленої випадкової величини наведено на ріс.6.18,*а* і ріс.6.18,*б* відповідно.

[6.4] Функція розподілу випадкової величини $F(X)$ це кількісна характеристика ймовірності P події ($X < x$), тобто $F(X) = P(X < x)$ де x – деяка поточна змінна. Функцію розподілу іноді називають інтегральним законом розподілу.

[6.5] Функція щільності – похідна (якщо вона існує) від функції розподілу.



а)   *б*)

Рис. 6.18

Математичне сподівання[6.6] m_x, дисперсія[6.7] σ_x^2 і середньоквадратичне відхилення[6.8] σ_x безперервної рівномірно розподіленої випадкової величини відповідно дорівнюють:

$$m_x = \int_a^b xf(x)dx = \frac{1}{b-a}\int_a^b xdx = \frac{1}{b-a}\frac{x^2}{2}\Big|_a^b = \frac{b+a}{2};$$

$$\sigma_x^2 = \int_a^b (x-m_x)^2 f(x)dx = \frac{(b-a)^2}{12}; \quad \sigma_x = \sqrt{\sigma_x^2} = \frac{b-a}{2\sqrt{3}}.$$

Розрізняють три способи отримання ВЧ:

• за допомогою спеціальних електронних пристроїв, званих генераторами випадкових чисел;

• алгоритмічний, що реалізується за спеціальними алгоритмами на ЦОМ;

• табличний.

За ***генератори*** «істинно» випадкових чисел використовуються електронні пристрої, побудовані на джерелах природних випадкових шумів зі спектром, близьким до рівномірного розподілу. Найвідомішим з таких пристроїв був електронний імпульсний генератор, керований джерелом шуму, розроблений широко відомою фірмою RAND Corporation. Використовувалися й інші подібні генератори – наприклад що базуються на перетворенні природного випадкового шуму при радіоактивному розпаді. Всі ці генератори володіють двома недоліками:

[6.6] Математичне сподівання - середнє ймовірнісне значення випадкової величини.

[6.7] Дисперсія випадкової величини - міра розкиду даної випадкової величини, тобто її відхилення від математичного сподівання.

[6.8] Середньоквадратичне відхилення - показник розсіювання значень випадкової величини відносно її математичного сподівання.

- неможливо повторно отримати одну й ту саму послідовність випадкових чисел, що буває необхідно при експериментах з імітаційної моделлю;
- технічно складно реалізувати фізичні генератори, здатні тривалий час видавати випадкові числа "необхідної якості".

При використанні ***алгоритмічного методу*** випадкові числа генеруються апаратно або програмно на ЦОМ і вони називаються псевдовипадковими, проте їх статистичні властивості збігаються зі статистичними властивостями «істинно» випадкових чисел.

Алгоритмічний метод передбачає машинну імітацію псевдовипадкових чисел (ПВЧ) за спеціальною програмою, в який кожне наступне число отримують шляхом обробки попередніх. До складу практично всіх сучасних систем програмування входять спеціальні функції генерації випадкових чисел, які зазвичай називають датчиками або генераторами випадкових чисел. В імовірнісному розумінні такі числа не є випадковими, оскільки метод їх отримання повністю детермінований, тому при використанні в імітаційних моделях вони повинні піддаватися статистичним перевіркам (тестуванню).

Алгоритми для отримання ПВЧ на ЦОМ повинні забезпечувати незалежність або хоча б слабкі кореляційні зв'язки між реалізаціями ПВЧ в межах використовуваного відрізка послідовності та мале число операцій ЦОМ на їх вироблення. На практиці використовують безліч таких алгоритмів.

Найбільш широко застосовують для формування ПВЧ рекурентні формули першого порядку:

$$x_{i+1} = f(x_i), \tag{6.20}$$

тобто кожне чергове випадкове число x_{i+1} отримують в результаті деякого перетворення f над попереднім числом x_i або групою попередніх випадкових чисел. Перетворення f – це деяке перемішування попередніх чисел і виконання деяких арифметичних операцій.

До алгоритмів виду (6.20) відносять найбільш популярний, лінійний конгруентний метод, розроблений Д. Лемером в 1949 році. Для генерації послідовності випадкових чисел в конгруентному методі, використовується така формула:

$$x_{i+1} = f(x_i) = (Ax_i + C) \bmod M .$$

де x_{i+1}, x_i – чергове і попереднє випадкові числа відповідно; A, C – цілі невід'ємні числа; M – достатнє велике ціле додатне число (чим більше M, тим довше неповторювана послідовність). Операція взяття за модулем (mod M) являє собою обчислення залишку від ділення числа на M, наприклад, 24 mod 10 = 4.

При вдалому виборі вихідних чисел послідовність, що генерується, буде містити випадкові числа. Наприклад, стандартний генератор випадкових чисел в Delphi використовує значення A = 134775813, C = 1 и M = 232, а значення x_0 можна задавати безпосередньо або використовувати процедуру ***Randomize*** для обчислення його на основі показань системного годинника.

При програмуванні на різних алгоритмічних мовах завжди можна скористатися стандартними функціями, що реалізують отримання послідовностей випадкових величин, рівномірно розподілених в інтервалі [0,1]. Це функція ***Random***. Алгоритм її роботи, як правило, невідомий. Тому особливо ретельно потрібно перевіряти якість послідовності величин, одержуваних за допомогою цього генератора.

Табличний метод отримання ВЧ не є самостійним. ВЧ, що сформовані за допомогою генератора випадкових чисел або будь-якого іншого алгоритму на ЦОМ, заносяться в зовнішню пам'ять машини і зчитуються у ході виконання завдання в будь-якому заданому порядку. Слід мати на увазі, що «ідеальної» таблиці для будь-яких додатків не існує.

Генеровані ВЧ є основою формування випадкових процесів або випадкових функцій, які використовуються при стохастичному моделюванні. В першу чергу це стосується стаціонарних випадкових процесів типу білого шуму**.**

Білий шум є стаціонарним випадковим процесом $x(t)$ з нульовим математичним сподіванням і постійної спектральної щільністю[6.9] $S_x(\omega)$, яку називають інтенсивністю білого шуму і яка дорівнює дисперсії значень $x(t)$.

[6.9] Спектральна щільність (спектральна щільність потужності) – функція, що описує розподіл потужності (узагальнене поняття кількості) сигналу залежно від частоти, тобто потужність, що припадає на одиничний інтервал частоти.

$$S_x(\omega) = \sigma^2;$$
$$\mu_x(t) = 0.$$

Усі спектральні складові білого шуму рівномірно розподілені по всьому діапазону задіяних частот. З огляду на те, що спектральна щільність потужності цього шуму однакова на всіх частотах, він і отримав свою назву «білий» (за аналогією з білим світлом, який містить електромагнітні хвилі частот всього видимого діапазону електромагнітного випромінювання).

У природі і техніці суто білий шум не зустрічається (з огляду на те, що такий сигнал мав би нескінченну потужність), однак під категорію білих шумів потрапляють будь-які шуми, спектральна щільність яких однакова (або слабо відрізняється) в розглянутому діапазоні частот.

Білий шум некорельований за часом (або за іншим аргументом), тому його значення не визначаються в часовій (або будь-який іншої розглянутої аргументної) області. Набори, сигналів, можуть бути довільними, проте постійна складова таких сигналів повинна дорівнювати нулю.

Дискретний білий шум - це просто послідовність незалежних (тобто статистично не зв'язаних один з одним) чисел. З використанням генератора псевдовипадкових чисел, дискретний білий шум можна отримати, використовуючи функцію ***Random***, таким чином:

```
Randomize;
x[i] = 2 * (Random - 0.5).
```

У даному випадку x - масив дискретного білого шуму, який має рівномірний розподіл від -1 до 1 (с інтенсивністю, що дорівнює одиниці).

Якщо прибрати з програми процедуру ***Randomize***, то кожен раз при запуску програми буде генеруватися одна і та ж реалізація білого шуму.

Білий шум активно використовується при моделюванні датчиків інформаційних систем, електронних фільтрів, систем управління, тощо. Коли на вхід такої системи подається білий шум, то на виході отримують сигнал, який є відгуком системи на цій прикладений вплив.

Наприклад, похибка барометричного висотоміра у визначенні абсолютної висоти може буті описана таким співвідношенням:

$$\Delta H = \Delta h_{\text{с}} + \sigma_h \varepsilon_h ,$$

де $\Delta h_{\text{с}}$ – квазістаціонарна похибка виміру барометричної висоти;

σ_h – флуктуаційна складова білошумової похибки;

ε_h – дискретний білий шум з одиничною інтенсивністю.

У свою чергу дискретна модель еволюції квазістаціонарної похибки виміру може бути представлена в наступному вигляді:

$$\Delta h_{\text{с},k+1} = \Delta h_{\text{с},k} + \sigma_{\xi\text{с}} \xi_k ,$$

де $\sigma_{\xi\text{с}}$ – заданий параметр; ξ_k – стандартний дискретний білий шум з одиничною інтенсивністю.

Для моделей похибок сучасних барометричних висотомірів можна рекомендувати наступні значення параметрів:

$$\sigma_h = (0,5 \div 1)\text{м}; \quad \sigma_{\xi\text{с}} = (0,05 \div 0,02)\text{м}; \quad \Delta h_{\text{с},0} = (3 \div 5)\text{м},$$

де $\Delta h_{\text{с},0}$ – початкове середньоквадратичне значення похибки $\Delta h_{\text{с}}$.

Але іноді в задачах математичного моделювання потрібно описати білошумну складову процесу, що має не рівномірний, а адитивний нормальний розподіл. Термін «адитивний» означає, що даний вид шуму шумується з корисним сигналом.

При нормальному розподілі, яке також називають розподілом Гауса, функція щільності, співпадає з функцією Гауса і має вигляд:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} ,$$

де μ – математичне сподівання розподілу;

σ – середньоквадратичне відхилення розподілу;

σ^2 – дисперсія розподілу.

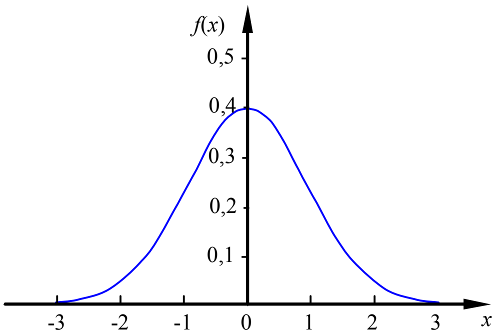


Рис. 6.19

Стандартним нормальним розподілом (див. рис. 6.19) називається нормальний розподіл з математичним сподіванням $\mu = 0$ і стандартним середньоквадратичним відхиленням $\sigma = 1$.

Оскільки більшість генераторів ПВЧ на виході дають тільки рівномірний розподіл, то часто виникає необхідність його перетворення в будь-який інший, зокрема, в нормальний гаусовський розподіл. Для цього можна скористатися найбільш поширеним перетворенням Боксу-Мюллер, який має два варіанти реалізацій.

1 варіант. З використанням функції `Random` генеруються дві незалежні випадкові величини r и φ, що рівномірно розподілені в інтервалі (0, 1). Потім за формулами:

$$z_0 = \cos(2\pi\varphi)\sqrt{-2\ln r}, \quad z_1 = \sin(2\pi\varphi)\sqrt{-2\ln r}$$

обчислюються значення z_0 и z_1, які й будуть незалежними та нормально розподіленими величинами з математичним сподіванням $\mu = 0$ і дисперсією $\sigma^2 = 1$.

2 варіант. Знову ж, генеруються дві незалежні випадкові величини x и y, що рівномірно розподілені на відрізку $[-1, 1]$. Потім обчислюється величина $s = x^2 + y^2$. Якщо одна з вимог $0 < s \leq 1$, то значення x и y слід згенерувати заново. Як тільки умова $0 < s \leq 1$ виконується, то за формулами:

$$z_0 = x \cdot \sqrt{\frac{-2\ln s}{s}}, \quad z_1 = y \cdot \sqrt{\frac{-2\ln s}{s}}$$

розраховуються значення z_0 и z_1, які, як і в першому випадку, будуть незалежними величинами, які задовольняють стандартному нормальному розподілу.

Існують і інші програмні реалізації такого перетворення. Наприклад, в Delphi функція `RandG(0,1)` генерує випадкові числа зі стандартним гаусовським розподілом.

При стохастичному моделюванні систем управління й обробки інформації часто виникає необхідність формування моделей випадкових зовнішніх впливів або похибок вимірювань з заданими властивостями. Ці моделі, як правило, представляються випадковими стаціонарними процесами, із заданою спектральної щільністю $S_x(\omega)$. Щоб сформувати стаціонарний процес з заданими властивостями зазвичай використовують метод формуючого фільтру.

Формуючим фільтром $\Phi(j\omega)$ називають динамічну ланку, яка формується з процесу $\xi(t)$ типу «білий шум», який має спектральну щільністю S_0, випадковий процес $x(t)$ із заданою спектральною щільністю:

$$S_x(\omega) = \left|\Phi(j\omega)\right|^2 S_0 .$$

Для заданої спектральної щільності $S_x(\omega)$ передаточна функція $\Phi(p)$ формуючого фільтру має вигляд:

$$\Phi(p) = \frac{S_x(p)}{\sqrt{S_0}} = \frac{1}{\sqrt{S_0}} \frac{F_m(p)}{H_n(p)} , \tag{6.21}$$

де $S_x(p) = \sqrt{\left|S_x(\omega)\right|} = \dfrac{F_m(p)}{H_n(p)}$;

$F_m(p)$ і $H_n(p)$ – поліноми степені m і ($n > m$), які описують задану спектральну щільність $S_x(p)$.

Зазвичай вважають, що процес $\xi(t)$ є гаусовським, нормованим умовою $S_0 = 1/2\pi$.

Передаточній функції (6.21) відповідає диференційне рівняння:

$$H_n(D)x(t) = \sqrt{\frac{1}{S_0}} F_m(D)\xi(t); \qquad D = d/dt . \tag{6.22}$$

Від цього рівняння за допомогою відомих перетворень можна перейти до системи диференціальних рівнянь першого порядку

Рівняння формуючого фільтра при моделюванні на ЦОМ, виходить з формули (6.22) при $S_0 = h/2\pi$ і має вигляд:

$$H_n(D)x(t) = \sqrt{\frac{2\pi}{h}} F_m(D)\xi(t) ,$$

де h – крок інтегрування диференціальних рівнянь формуючого фільтру.

Нагадаємо, що при інтегруванні системи диференціальних рівнянь методом Рунге-Кута на кожному кроці кілька разів обчислюються праві частини. Тому, складаючи програми, слід передбачити, щоб при обчисленні правих частин використовувалася одна і та ж для даного кроку інтегрування випадкова величина $\xi_k(t)$. Наведений метод, заснований на моделюванні за допомогою ЦОМ формуючого фільтра, має методичною помилкою, величина якої зменшується при $h \to 0$.

Найбільш характерним прикладом стохастичного моделювання систем цивільної авіації є моделювання польоту літака в зонах турбулентної атмосфери, які характеризуються хаотичним, поривчастим, неврегульованим рухом повітря.

Вплив турбулентності атмосфери на політ проявляється в різких і раптових змінах режиму горизонтального польоту. При попаданні в порив попутного вітру літак як би «провалюється», а при раптовому зустрічному пориві вітру апарат кидає вгору. Аналогічна картина виникає при впливі на літак вертикальних поривів вітру. Коли пориви вітру діють безупинно один за іншим, літальний апарат то «провалюється» вниз, то підкидається вгору; глибина «провалу» і висота підйому апарату тим більше, чим сильніше поривчастість вітру, тобто чим більше турбулентність. «Провали» іноді досягають 100-200 м.

Різкі та раптові зміни режиму польоту внаслідок і впливу турбулентності зазвичай називають «бовтанка» або рему. Рему залежить від фізичного стану атмосфери, пори року, часу доби, тощо.

При моделюванні турбулентної атмосфери робиться припущення, що турбулентні пориви являють собою локально-нормальний випадковий процес, розділений в просторі на зони з різними параметрами.

Теоретичні та експериментальні дослідження атмосферної турбулентності дають такі результати:

1. Величина пульсації швидкості в межах об‘єму, який займає літак звичайних розмірів, істотно не змінюється.

2. Пульсація швидкості вітру є стаціонарним процесом. Компоненти цієї швидкості w_x, w_y, w_z є незалежними. Статистичні характеристики пульсацій швидкості вітру в поперечних напрямках w_y и w_z однакові.

3. Спектральні щільності складових атмосферної турбулентності w_x і w_y є загальновизнаними і описуються, наприклад, моделями Драйдена:

$$S_{w_x}(\omega) = \sigma_u^2 \frac{L}{\pi V} \frac{1}{1+\left(\frac{L}{V}\right)^2 \cdot \omega^2}; \quad S_{w_y}(\omega) = \sigma_u^2 \frac{L}{2\pi V} \frac{1+3\cdot\left(\frac{L}{V}\right)^2 \cdot \omega^2}{\left[1+\left(\frac{L}{V}\right)^2 \cdot \omega^2\right]^2}, \quad (6.23)$$

або моделями Кармана. Модель Кармана для спектральної щільності атмосферної турбулентності w_y має вигляд:

$$S_{w_y}(\omega) = \sigma_u^2 \cdot \frac{L^*}{\pi} \cdot \frac{1 + \frac{8}{3}\left(1,339 \cdot L^* \cdot \omega\right)^2}{\left[1 + \left(1,339 \cdot L^* \cdot \omega\right)^2\right]^{\frac{11}{6}}}.$$

У наведених вище моделях σ_u – середньоквадратичне значення швидкості випадкового вітру; L - лінійний масштаб турбулентності; V – швидкість польоту, L^* – інтегральний масштаб турбулентності.

Атмосферу вважають практично спокійною, якщо $\sigma_u >$ 0,5 м/с, і дуже збуреною, якщо $\sigma_u >$ 2,5 м/с. Лінійний масштаб турбулентності в залежності від метеоумов, пори року та інших чинників приймає значення в інтервалі L = (100…1000) м.

При дослідженні динаміки польоту літака спектральна щільність складових атмосферної турбулентності може бути виділена з білого шуму за допомогою формуючого фільтра, тобто замість (6.23) можна використовувати

$$S_w(\omega) = \left|\Phi(j\omega)\right|^2 S_0,$$

де $\Phi(j\omega)$ – формуючий фільтр; S_0 – інтенсивність білого шуму

Якщо за інтенсивність білого шуму прийняти величину

$$S_0 = \frac{\sigma_u^2 L}{2\pi \cdot V},$$

то, наприклад, для вертикальної складової атмосферної турбулентності можна отримати передаточну функцію формуючого фільтра у вигляді:

$$\Phi_{w_y}(p) = \frac{\sqrt{3T_ф p + 1}}{(T_ф p + 1)},$$

а для горизонтальної складової атмосферної турбулентності передаточна функція формуючого фільтра буде мати вигляд:

$$\Phi_{w_x}(p) = \frac{1}{(T_ф p + 1)}, \qquad (6.24)$$

де $T_ф = L/V$.

Передаточної функції (6.24) відповідає диференціальне рівняння одиничної реалізації випадкового стаціонарного процесу, яке описує горизонтальну складову w_x атмосферної турбулентності в часовій області:

$$\frac{dw_x}{dt} = \frac{S_0 \cdot \xi(t) - w_y}{T_ф}. \tag{6.25}$$

А використовуючи перехід від передаточної функції до рівнянь стану, що базується на канонічних поданнях системи, отримаємо запис рівнянь одиничної реалізації вертикальної складової w_y атмосферної турбулентності в так званій першій канонічній формі:

$$\begin{aligned} &\frac{dx}{dt} = \frac{S_0 \cdot \xi(t) - x}{T_ф}; \\ &w_y(t) = \sqrt{3S_0 \cdot \xi(t) - 2x}. \end{aligned} \tag{6.26}$$

У (6.25), (6.26) $\xi(t)$ – одинична реалізація дискретного білого шуму з одиничною інтенсивністю; S_0 – інтенсивність білого шуму.

Дія на літак випадкового пориву вертикального вітру w_y рівнозначна зміні кута атаки на значення α_w. Якщо відношення V/w_y достатньо велике, то приріст кута атаки від швидкості вертикального вітру приблизно має вигляд:

$$\alpha_w = \frac{w_y}{V}.$$

Отже, дія вітрових збурень на динаміку поздовжнього руху літака враховується шляхом додавання до геометричного рівняння $\alpha = \vartheta - \Theta$ складової приросту кута атаки від швидкості вертикального вітру:

$$\alpha = \vartheta - \Theta + \alpha_w. \tag{6.27}$$

Зустрічні й попутні турбулентні пориви горизонтального вітру треба врахувати при обчисленнях повітряної швидкості. Зокрема, обчислюючи проекції сил на поздовжню вісь траєкторної системи координат, слід враховувати поздовжню складову вітру w_x:

$$m(\dot{V} + \dot{w}_x) = P\cos\alpha - X_a - G\sin\theta. \tag{6.28}$$

Після лінеаризації рівняння (6.28), з урахуванням $X_a = X_a(V, \alpha)$, $P = P(V, \delta_p)$, отримаємо:

$$\Delta\dot{V} + a_x^V \Delta V + a_x^{\theta} \Delta\theta + a_x^{\alpha} \Delta\alpha = a_x^{\delta_p} \Delta\delta_p + \Delta\dot{w}_x . \qquad (6.29)$$

Рівняння (6.29) сумісно з рівняннями поздовжнього руху (6.18), в яких враховані: вплив швидкості польоту на підйомну силу та аеродинамічний момент тангажа (коефіцієнти a_y^V, $a_{m_z}^V$) та приріст кута атаки згідно (6.28) від швидкості вертикального вітру (в останньому рівнянні (6.18)) формують математичну модель поздовжнього руху літака в турбулентній атмосфері:

$$\begin{aligned}
&\Delta\dot{V} + a_x^V \Delta V + a_x^{\Theta} \Delta\Theta + a_x^{\alpha} \Delta\alpha = \Delta\dot{w}_x;\\
&\Delta\dot{\Theta} + a_y^V \Delta V + a_y^{\alpha} \Delta\alpha = 0;\\
&\Delta\dot{\omega}_z + a_{m_z}^V \Delta V + a_{m_z}^{\alpha} \Delta\alpha + a_{m_z}^{\omega_z} \Delta\omega_z = a_{m_z}^{\delta_\text{в}} \Delta\delta_\text{в};\\
&\Delta\dot{\vartheta} = \Delta\omega_z;\\
&\Delta\alpha = \Delta\vartheta - \Delta\Theta + \Delta\alpha_w.
\end{aligned}$$

Щоб оцінити вплив турбулентності на інтенсивність «бовтанки» літака, систему рівнянь необхідно доповнити ще одним співвідношенням:

$$\Delta\dot{H} = a_H^V \Delta\Theta,$$

де $a_H^V = V_0 \cos\Theta_0$, (V_0, Θ_0) – швидкість і кут нахилу траєкторії незбуреного руху.

Блок-схема моделюючого алгоритму (див. рис. 6.17) процедури чисельного розв‘язання диференціальних рівнянь, які описують політ літака в турбулентної атмосфері аналогічна, описаному в п. 6.4. Відмінними рисами алгоритму є: необхідність додаткових вихідних даних і наявність процедур формування випадкових стаціонарних процесів, які описують складові атмосферної турбулентності.

Додатковими вихідними даними є: σ_u – середньоквадратичне значення швидкості випадкового вітру; L – лінійний масштаб турбулентності; V – швидкість польоту.

Додаткові процедури реалізуються в блоках:

блок 3 – формування інтенсивності білого шуму постійної часу формуючого фільтра;

блок 3* – формування дискретного білого шуму одиничної інтенсивності для вертикальній складовій атмосферної турбулентності;

блок 4* – формування дискретного білого шуму одиничної інтенсивності для горизонтальній складовій атмосферної турбулентності;

блок 5* – формування одиничної реалізації вертикальної складової w_y атмосферної турбулентності відповідно з (6.26) та приросту кута атаки від швидкості вертикального вітру.

У місце, зазначене (*) в алгоритмі (див. рис. 6.17), замість блоків 3, 4 потрібно вставити фрагмент, показаний на рис. 6.20.

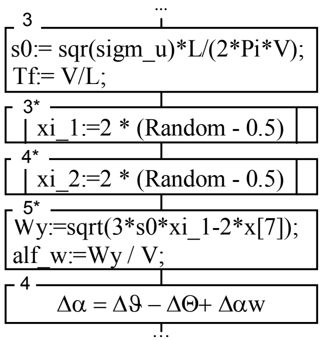


Рис. 6.20

При програмуванні, в процедуру опису правих частин рівнянь (procedure odi) включають всі рівняння, що підлягають інтегруванню, у тому числі рівняння поздовжнього руху літака та рівняння реалізацій випадкових процесів, що описують складові атмосферної турбулентності. Нижче наводиться приклад програмного забезпечення процесу моделювання польоту літака в турбулентної атмосфері.

```
*****************************************************************

procedure ody(var x,px:mat);

{x,px – масиви значень функцій і їх похідних: x[1],x[2],
x[3], x[4], x[5]– швидкість, кут нахилу траєкторії, кутова
швидкість тангажа, кут тангажа і висота польоту відповідно;
x[6]– поздовжня складова випадкового вітру wx, x[7] – допо-
міжна змінна формування  вертикальної складової випадкового
вітру wy; Tf – стала часу формуючого фільтру}

 mat = array[1..n] of real; (в даному випадку n = 7)}

     begin
       px[1]:=-a11*x[1]-a12*x[2]-a13*alfa+px[6];
       px[2]:=-a21*x[1]-a22*alfa;
       px[3]:=-a31*x[1]-a32* alfa -a33*x[3];
       px[4]:=x[3];
```

```
        px[5]:=a51*x[2];
        px[6]:=(s0*xi_1-x[6])/Tf;
        px[7]:=(s0*xi_2-x[7])/Tf;
      end;
***********************************************************
```

Зміни в лістингу основної процедури моделюючого алгоритму, наведеного у п. 6.4, надані нижче.

```
***********************************************************
      procedure TForm1.Button1Click(Sender: TObject);
 begin
      …
    s0:= sqr(sigm_u)*L/(2*Pi*V);
    Tf:= V/L;
    { Початок нескінченного циклу }
  Repeat
        Randomize;
    xi_1:=2 * (Random - 0.5);
    xi_2:=2 * (Random - 0.5);
    Wy:= sqrt(3*s0*xi_1-2*x[7]);
    alf_w:= Wy/V;
    alf:= x[3]-x[1]+ alf_w; // Розрахунок кута атаки
    {Графічне виведення результатів інтегрування}
   Series1.AddXY(t, x[1]); Series2.AddXY(t, x[2]);
       t:=t+sh; // відлік часу
{Звернення до процедури інтегрування, яка в свою чергу звертається до процедури ody - процедурі опису правих частин рівнянь}
       rk(x,px);
    {Зупинка рішення після закінчення часу інтегрування}
       if t>=Tm then stop:=1;
  Until   stop=1;
 end;
***********************************************************
```

При програмуванні методів чисельного інтегрування необхідно брати до уваги такі фактори, як очікувану точність, трудомісткість обчислення на ЦОМ, простоту використання та модифікації програми та ін. Програми, що реалізують один і той же моделюючий алгоритм, можуть істотно відрізнятися за своїми характеристиками. Суттєво спростити діалог з комп'ютером дозволяє використання універсальних пакетів прикладних програм та інтегрованих середовищ моделювання, для яких є обґрунтування області застосування та контрольні приклади. Модульний склад пакетів відповідає модульній структурі використовуваних підходів до програмування задач і широко застосовується при виборі необхідних методів розв‘язання для певного класу задач.

Контрольні запитання

1. Виконання, яких етапів передбачає програмна реалізація процедури розв'язання СЛАР методом Гауса?
2. Які завдання вирішуються в процесі виконання прямого ходу прогону при реалізації сплайн-інтерполяції?
3. У чому полягає задача чисельного диференціювання?
4. Як вирішується задача відшукання значення похідної від функції, заданої таблично?
5. Перелічіть двоточкові методи чисельного диференціювання?
6. Накреслить блок-схема алгоритму визначення вертикальної швидкості з використанням методу чисельного диференціювання?
7. Перелічіть кілька методів чисельного інтегрування?
8. Запишіть фрагмент програмної реалізації метода чисельного інтегрування Сімпсона.
9. Запишіть фрагмент програмної реалізації процедури чисельного розв'язання диференціальних рівнянь методом Рунге-Кута.
10. Яка функція програмного забезпечення генерує випадкові числа типу дискретного білого шуму?
11. Яке призначення процедури **`Randomize`** при реалізації білого шуму?
12. Який метод використовується для формування випадкового стаціонарного процесу з заданими властивостями?

Глава 7. КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ З ВИКОРИСТАННЯМ ІНТЕГРОВАНИХ ПАКЕТІВ ПРИКЛАДНИХ ПРОГРАМ

Сучасні програми чисельного моделювання систем і процесів стають усе більш автоматизованими, полегшуючи користувачу процес постановки й розв'язання широкого класу складних задач. Ще більший ефект дають сучасні можливості якісного візуального подання результатів.

Розглянемо програмне забезпечення персональних комп'ютерів, що використовується на різних етапах математичного моделювання. В останні роки в розвитку програмного забезпечення для персональних ЕОМ простежується тенденція застосування інтегрованих пакетів прикладних програм, які включають поряд зі спеціалізованими програмами програми підготовки звітів і багато чого ін.

В останні роки в науково-технічних колах світу дістала широке розповсюдження інтегроване середовище для проведення математичних розрахунків, проектування й моделювання - MATLAB компанії MathWorks.

<u>Пакет MAT LAB</u>

Система MATLAB призначена для виконання інженерних і наукових розрахунків і високоякісної візуалізації одержуваних результатів. Ця система застосовується в математиці, обчислювальному експерименті, імітаційному моделюванні, фінансових розрахунках і ряді інших галузей.

MATLAB - це набір прикладних програм для виконання завдань і розв'язання технічних обчислень на основі мови програмування четвертого покоління. Сам по собі пакет прикладних рішень MATLAB являє собою не тільки мову програмування нового покоління, але й окреме обчислювальне середовище. За допомогою MATLAB можна створювати користувальницькі інтерфейси, виконувати обчислення з матрицею, функціями й іншими даними, алгоритми, а також встановлювати зв'язки з програмами, що працюють на інших мовах програмування.

Сьогодні MATLAB використається в безлічі галузей, серед яких обробка сигналів і зображень, проектування систем керування, фінансові розрахунки й медичні дослідження. Його відкрита архітектура уможливлює використання MATLAB і супутніх продуктів для дослідження даних і створення власних інструментів, що використовують функціональні можливості MATLAB.

Джерела MATLAB пов'язані з розв'язанням задач лінійної алгебри й підвищенням ефективності використання математичних пакетів LINPACK і EISPACK, призначених для роботи з матрицями. Якщо згадати, що джерелом математичних моделей лінійних систем з постійними параметрами служать лінійні диференціальні або різницеві рівняння з постійними коефіцієнтами, то стає зрозумілим, що теоретична основа цих рівнянь - лінійна алгебра. Тому ядро інтегрованої системи MATLAB становить основу для більшості процедур пакетів Control System Toolbox, System Identification Toolbox, Frequency Domain Identification і ряду ін.

Крім ядра, що містить обчислювальні алгоритми загального призначення, у пакеті MATLAB реалізовано кілька десятків так званих тулбоксів (бібліотек спеціалізованих підпрограм), призначених для розв'язання різноманітних практичних задач створення, аналізу й оптимізації систем.

Тулбокси (toolboxes - тулбокси, тобто набори інструментів) – більше, ніж набір корисних функцій. Вони являють собою останнє слово в розробці (дослідженнях) у таких областях, як керування, обробка сигналів, ідентифікація систем і багатьох ін.

MATLAB - дуже зручна система для розробки додатків, дозволяючи ефективно створювати програми, виконувані як у середовищі MATLAB, так і незалежно. Крім багатих можливостей мови MATLAB для створення алгоритмів, у ньому містяться також широкі можливості для проектування користувальницького інтерфейсу.

Операції, які виконуються при створенні користувальницького інтерфейсу в середовищі MATLAB, аналогічні за простотою та наочністю операціям візуального програмування, виконуваним у середовищі програми Visual Basic.

<u>Пакет MATH CAD</u>

Інша сторона розвитку програмного забезпечення - орієнтація на "непрограмуючого користувача". У цьому випадку користувач такого пакета одержує можливість зосередитися на сутності самої задачі, а не на способах її програмної реалізації. У свою чергу, користувач повинен ясно представляти можливості використовуваного пакета й закладених у ньому методів, а також уміти вибрати необхідний пакет, що відповідає розв'язуваній задачі.

Всі етапи створення й використання математичної моделі легко простежити при роботі з пакетом MATHCAD фірми "MathSoft Inc." (USA).

MATHCAD - універсальний математичний пакет, призначений для виконання інженерних і наукових розрахунків. Математичне забезпечення пакета дозволяє вирішувати багато задач, наприклад, в об'ємі вузу.

Розроблювачі цього пакета вдосконалюють пакет від версії до версії. У цей час існують версії починаючи з MATHCAD 2000 і MATHCAD 2001, які володіють великими можливостями. Існують оригінальна (англомовна) і русифікована версії програми.

Від калькулятора пакет MATHCAD відрізняють обчислення з довільною точністю, робота з різними типами даних (комплексні, вектори, матриці), використання бібліотеки математичних функцій, яка може бути доповнена власними програмами.

Основна перевага пакета перед типовими мовами програмування - природна математична мова, на якій формулюється розв'язувана задача.

Пакет об'єднує в собі: редактор математичних формул, інтерпретатор для обчислень, бібліотеку математичних функцій, процесор символьних перетворень, текстовий редактор, графічні засоби подання результатів. Пакет MATHCAD відноситься до інтегрованих пакетів, тобто дозволяє не тільки зробити обчислення, але й отримати документ - підсумковий звіт з коментарями, формулами, таблицями й графіками.

До позитивних якостей MATHCAD варто віднести відкритість - все наведене в документі може бути відтворено, а інтеграція в одному документі вихідних даних, методу розв'язання й результатів дозволяє зберегти настроювання для розв'язання подібних задач.

<u>Пакет Lab VIEW</u>

LabVIEW – це графічне середовище розробки і платформа для виконання програм, створених на графічній мові програмування «G» фірми National Instruments (США). Створення програм являє собою процес побудови блок-діаграми з використанням стандартних графічних образів (іконок) з бібліотеки LabVIEW. Будь-яка програма, що створена в LabVIEW є віртуальним приладом, який має «лицьову панель» (іконки вводу-виводу для керування приладом: перемикачі, кнопки, світлодіоди, табло, лампочки, тощо) та саму «блок-схему» (логіка роботи програми). Всі частини програми з'єднані між собою комутаційними лініями, за якими відбувається передача даних.

Пакет LabVIEW дозволяє працювати зі створеними віртуальними приладами й інструментами апаратно-програмного комплексу LabVIEW. При цьому LabVIEW надає можливість одночасного використання функцій багатьох приладів різного призначення, а також функцій обробки результатів вимірювань і багато чого ін. Наробітки в області комп'ютерної графіки дозволили розроблювачам LabVIEW додати нові тривимірні елементи користувальницького інтерфейсу для створення професійно оформлених вимірювальних систем.

Основна особливість LabVIEW з погляду користувача – це графічна інтерпретація мови програмування цієї системи. Графічна мова - мова функціональних блок-діаграм - дозволяє значно спростити створення, наприклад, програм керування зовнішнім об'єктом або програм обробки даних, отриманих від зовнішнього пристрою. Разом з тим у середовищі LabVIEW можливе використання програм на мові Си. Можливості системи розширюються за рахунок додаткових бібліотек роботи з базами даних (SQL, Toolkit), обробці зображень (Convert VI), PID-регулюванню (PID Control).

При доповнені персонального комп'ютера багатофункціональними вбудованими платами вводу/виводу аналогової й цифрової інформації стандартних інтерфейсів (VME, IEEE, RS) та мереж промислових контролерів (зовнішніх адаптерів) середовище LabVIEW можна використовувати для програмної підтримки автоматизованих систем наукових досліджень (АСНД) і автоматизованих систем управління (АСУ). При цьому пакет прикладних програм LabVIEW забезпечує: обмін інформацією із зовнішніми вимірювальними та керуючими пристроями та пристроями стеження;

аналіз і обробку отриманої інформації; збирання, зберігання й передачу (у тому числі й через мережі) інформації; підтримку математичного експерименту (роботу з віртуальними інструментами); роботу користувальницького інтерфейсу АСУ й АСНД.

Для користувача створеної АСНД керування експериментом зводиться до роботи з лицьовою панеллю віртуальної установки на екрані монітора, за допомогою якої він спостерігає за необхідними параметрами й керує програмно-апаратним комплексом АСНД.

При реалізації програмної підтримки автоматизованих систем управління LabVIEW може використовуватися в системах збору, обробки та передачі даних, а також для контролю й управління технічними об'єктами та технологічними процесами. У середовищі LabVIEW можуть бути створені віртуальні прилади, які моделюють як окремі функції вимірювального або керуючого комплексу, так і весь комплекс у цілому; можливо повне моделювання експерименту. В останньому випадку немає необхідності підключати зовнішні та погоджувальні пристрої. Користувач працює з віртуальними пристроями, які забезпечують достатній ступінь адекватності процесів, що моделюються, реальним процесам.

Пакет STATISTICA

У багатьох природно-наукових областях статистичні методи були й залишаються важливою складовою процедури обробки результатів вимірювань. Це стосується практично всіх галузей знань: фізики, хімії, біології, геології, метеорології й багатьох ін. Сучасні програми для статистичної обробки даних дозволяють застосовувати складні сучасні методи аналізу навіть у тих областях, де раніше такі дослідження були надзвичайно трудомісткими й, отже, проводилися досить рідко. Деякі приклади застосування системи STATISTICA для обробки експериментальних даних наведені в документації на систему.

Методи математичної статистики вивчають не тільки студенти природно-наукових спеціальностей, але й економісти, інженери, психологи, соціологи й багато інших фахівців. Тому курс математичної статистики входить у програму більшості вищих навчальних закладів, а невід'ємною його частиною стає освоєння відповідного програмного забезпечення. Досвід роботи показав, що система

STATISTICA може слугувати не тільки ефективним інструментом для наукових досліджень, але й надзвичайно зручним середовищем для навчання методам статистичного аналізу. Перелічимо продукти фірми StatSoft:

- STATISTICA 5.5 – потужний пакет статистичного й графічного аналізу даних;
- Quick STATISTICA - базовий набір найбільше часто використовуваних статистичних методів, до яких додаються всі графічні можливості системи STATISTICA та мови програмування;
- Power analysis – додаток для аналізу необхідних умов одержання надійних статистичних результатів;
- Neural Networks – універсальна програма для здійснення аналізу в нейронних мережах.
- Student Edition of STATISTICA – коротка версія пакету STATISTICA для студентів.

<u>Інші математичні пакети</u>

Все різноманіття математичних пакетів не обмежується перерахованими вище системами. Не претендуючи на повноту, перелічимо деякі: MAPLE VI - система символьних перетворень (входить в MATLAB і частково входить в MATHCAD); MATHEMATICA - потужна система для аналітичних рішень.

<u>Графічні пакети</u>

Графічні пакети призначені для візуалізації результатів розрахунків. У якості найбільш відомих назвемо наступні продукти: компанії Golden Software: Surfer 7.0 - для побудови просторових поверхонь, ліній рівня й карт; Grapher 2 - для побудови двовимірних графіків, а також Map Viewer 3.0 і Didger 2.0, інших компаній: Harvard ChartXL 2.0 - професійна програма побудови графіків за формулами, Math Plotter 3.6 - програма побудови різноманітних графіків, Super Graph 2.14 - програма аналізу графічних залежностей, математичний пакет, будує графіки в будь-яких системах координат, Graphit 2000 Professional Edition - програма побудови графіків за математичними виразами з багатьма можливостями їхнього редагування та інші.

7.1. Практичне застосування MATLAB

Пакет MATLAB широко використовується при розв'язанні задач, пов'язаних з матричними обчисленнями. Назва пакета утворена шляхом скорочення від MATrix LABoratory (матрична лабораторія). Операції й команди в MATLAB досить природні и аналогичны математичного запису формул на папері. MATLAB створювався як пакет програм, що реалізують найбільш ефективні обчислювальні алгоритми лінійної алгебри. MATLAB дає користувачу можливість швидко виконувати різні операції над векторами й матрицями, такі як множення й обертання матриць, обчислення визначників, знаходження власних чисел і векторів. Крім того, в MATLAB входять операції обчислення звичайних функцій (алгебраїчних, тригонометричних, логічних), розв'язання алгебраїчних і диференціальних рівнянь, операції побудови графіків і ряд інших, тобто він поєднує в собі чисельні розрахунки, візуалізацію й програмування.

MATLAB є мовою високого рівня. За окремими його командами можна виконувати такі складні операції, як знаходження коренів поліномів, розв'язання лінійних і нелінійних алгебраїчних рівнянь, моделювання лінійних динамічних систем. Зазначені операції є елементарними функціями MATLAB.

У пакет входить безліч гарна перевірених чисельних методів (решателей), оператори графічного подання результатів, засоби створення діалогів. Використовуючи пакет MATLAB, можна, як з кубиків, побудувати досить складну математичну модель, або написати свою програму. MATLAB застосовується для розв'язання найрізноманітніших наукових, технічних і економічних задач, а професійні додатки MATLAB розширюють його можливості в конкретних областях.

MATLAB - інтерактивна система, у якій основним елементом даних є масив. Основні області застосування MATLAB:

1. Математичні обчислення.
2. Розробка алгоритмів.
3. Моделювання.
4. Аналіз даних і візуалізація.
5. Наукова й інженерна графіка.

6. Розробка додатків, включаючи графічний інтерфейс користувача.

Система MATLAB складається з 5 основних частин:

1. ***Мова MATLAB.*** Це мова матриць і масивів високого рівня з керуванням потоками, функціями, структурами даних, вводом-виводом і з особливостями об'єктно-орієнтованого програмування. Все це дозволяє створювати як невеликі програми, так і складні програмні комплекси.
2. ***Середовище MATLAB.*** Це набір інструментів і пристосувань, з якими працює користувач або програміст MATLAB. Воно містить у собі засоби для керування змінними в робочому просторі, вводом і виводом даних, а також створення, контролю й налагодження М-файлів і додатків MATLAB.
3. ***Керована графіка.*** Це графічна система MATLAB, яка містить у собі команди високого рівня для візуалізації двох і тривимірних даних, обробки зображень і анімації. Ця система також має засоби для створення GUI (графічного інтерфейсу користувача).
4. ***Бібліотека математичних функцій.*** Це велика колекція обчислювальних алгоритмів.
5. ***Програмний інтерфейс.*** Це бібліотека, що дозволяє створювати програми на мовах C^{++} й Фортран, які використовують обчислювальні можливості MATLAB.

Професійні додатки MATLAB

Спеціалізовані засоби, що зібрані в пакети, називані **ToolBox** – набори функцій для розв'язання задач різних класів. Найбільш популярні з них:

1. ***Communications Toolbox*** – проектування, моделювання й аналіз систем зв'язку й передачі інформації.
2. ***Control System Toolbox*** – проектування, моделювання й аналіз систем автоматичного керування.
3. ***Image Processing Toolbox*** – цифрова обробка й аналіз зображень.

4. ***Optimization Toolbox*** **–** оптимізація лінійних і нелінійних функцій.
5. ***Partial Differential Equation Toolbox*** – розв'язання задач математичної фізики.
6. ***Statistics Toolbox*** – статистичні обчислення.
7. ***Symbolic Math Toolbox*** – інтегроване середовище для символьних обчислень.
8. ***Simulink*** – інтерактивне середовище для моделювання й аналізу широкого класу динамічних систем.

Деякі важливі характеристики тулбоксів:

- кожний тулбокс побудований на програмах, надійність і точність яких перевірена багаторічним досвідом;
- всі тулбокси сумісні й легко інтегруються не тільки з MATLAB, але й з Simulink і будь-яким іншим вже встановленим тулбоксом;
- всі тулбокси написані в коді відкритої архітектури MATLAB, тобто можна прочитати всі *m*-файли, зробити до них свої додавання або використовувати їх як шаблони при створенні власних функцій;
- кожний тулбокс може функціонувати на будь-якій комп'ютерній платформі, на якій працює MATLAB.

Simulink

Це інтерактивне середовище для моделювання й аналізу широкого класу динамічних систем, включаючи дискретні, безперервні й гібридні, нелінійні й розривні системи. Моделювання процесу відбувається шляхом переміщення блоків діаграм на екрані і їхній маніпуляції. Simulink сполучає в собі наочність аналогових машин[7.1] і точність цифрових обчислювальних машин. Simulink є додатком до пакета MATLAB і забезпечує користувачу доступ до всіх можливостей цього пакета, у тому числі до великої бібліотеки чисельних методів.

[7.1] В аналогових обчислювальних машинах на спеціальному комутаційному полі входи й виходи символьних зображень окремих операційних блоків, з'єднувалися проводами й шинами зв'язку в ланцюги аналогової моделі розв'язання конкретної задачі.

При моделюванні з використанням Simulink реалізується принцип візуального програмування, відповідно до якого, користувач на екрані з бібліотеки стандартних блоків створює модель пристрою й здійснює розрахунки. При цьому, на відміну від класичних способів моделювання, користувачу не потрібно досконально вивчати мову програмування й чисельні методи математики, а досить загальних знань, що вимагаються при роботі на комп'ютері.

При моделюванні користувач може вибирати метод розв'язання диференціальних рівнянь, а також спосіб зміни модельного часу (з фіксованим або змінним кроком). У ході моделювання є можливість стежити за процесами, що відбуваються в системі. Для цього використовуються спеціальні пристрої спостереження, що входять до складу бібліотеки Simulink. Результати моделювання можуть бути представлені у вигляді графіків або таблиць.

Simulink є досить самостійним інструментом MATLAB і при роботі з ним зовсім не потрібно знати сам MATLAB і інші його додатки. З іншого боку доступ до функцій MATLAB та інших його інструментів залишається відкритим і їх можна використовувати в Simulink.

Частина пакетів вхідних до складу Simulink має власні інструменти (наприклад, LTI-Viewer додатки до Control System Toolbox - пакета для розробки систем керування). Додаткові пакети розширення Simulink дозволяють вирішувати весь спектр задач від розробки концепції моделі до тестування, перевірки, генерації коду й апаратної реалізації.

Інтерактивне середовище Simulink, дозволяє використовувати вже готові бібліотеки блоків для моделювання електротехнічних, механічних і гідравлічних систем (наприклад, Power System Blockset - моделювання електротехнічних пристроїв, Digital Signal Processing Blockset - набір блоків для розробки цифрових пристроїв і т.д), а також застосовувати розвинений модельно-модельно-орієнтований підхід при розробці систем керування, засобів цифрового зв'язку й пристроїв реального часу.

При роботі з Simulink користувач має можливість модернізувати бібліотечні блоки, створювати свої власні, а також створювати нові бібліотеки блоків.

Контрольні запитання 

1. Які задачі можна виконувати за допомогою системи MATLAB?
2. Що являє собою система MATLAB?
3. Які засоби об'єднує в собі пакет MATHCAD?
4. Чому пакет MATHCAD відноситься до інтегрованих пакетів прикладних програм?
5. На якого користувача з точки зору програмного забезпечення зорієнтований пакет MATHCAD?
6. Чім пакет MATHCAD відрізняється від калькулятора?
7. У чому основна перевага пакета MATHCAD перед типовими мовами програмування?
8. Що являє собою пакет LabVIEW?
9. У чому полягає процес створення програм у середовищі LabVIEW?
10. Що являє собою будь-яка програма, яка створюється в LabVIEW?
11. Програмну підтримку яких автоматизованих систем забезпечує пакет прикладних програм LabVIEW?
12. Перелічіть продукти фірми StatSoft для статистичної обробки даних.
13. Яке призначення програми Neural Networks?
14. Перелічіть найбільш відомі графічні пакети візуалізації результатів розрахунків.
15. Який основний елемент даних використовується в інтерактивній системі MATLAB?
16. Перелічіть основні області застосування MATLAB.
17. З яких основних частин складається система MATLAB?
18. Що являють собою спеціалізовані пакети ToolBox в системі MATLAB?
19. Перелічіть найбільш популярні тулбокси.
20. Що являють собою середовище Simulink?
21. Який принцип програмування реалізується в середовищі Simulink?

Глава 8. ОСОБЛИВОСТІ МЕТОДОЛОГІЇ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ

Рішення про вибір методу моделювання приймається на початкових етапах моделювання. Як правило, відповідь на це питання визначається вибором концептуальної системи з урахуванням поставлених цілей і оцінки складності задачі дослідження системи. Саме складність задачі диктує спосіб подання моделі, який буде використовуватися при її опису. Очевидно, в одних випадках більш доцільним є аналітичне моделювання, в інших - імітаційне (або їхня комбінація).

Імітаційні моделі (ІМ) систем можуть бути представлені типовими математичними схемами: агрегатів, мереж Петрі, систем масового обслуговування, кінцевих або імовірнісних автоматів, тощо. Тип ІМ в основному залежить від вибору методу опису динаміки системи, від формального (математичного) опису випадкових факторів, які підлягають урахуванню в моделі й від вибору механізму зміни модельного часу. Класифікація ІМ схематично показана на рис. 8.1.

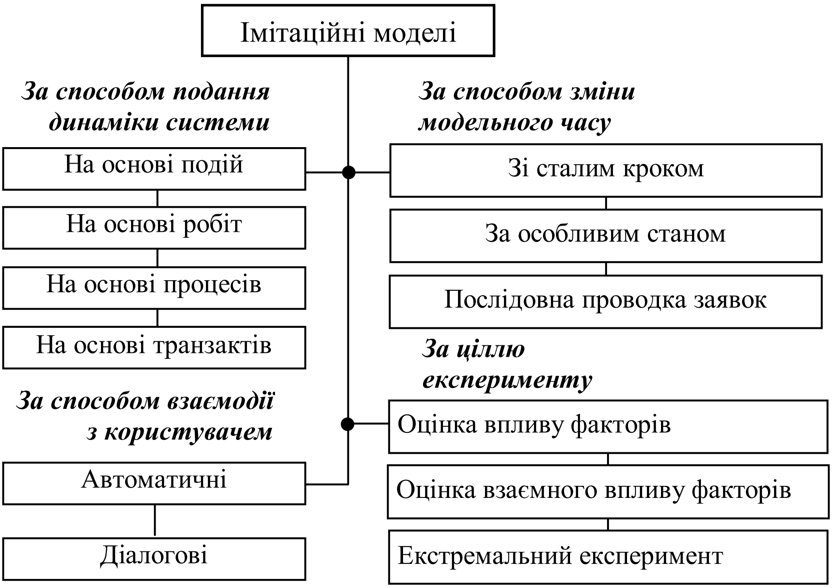


Рис. 8.1

Найбільш важлива класифікаційна ознака – *спосіб подання в моделі динаміки* (*руху*) *системи*. Вона може бути описана за допомогою подій, робіт (активностей), процесів і транзактів.

Іншою важливою класифікаційною ознакою є *спосіб зміни модельного часу*. За цією ознакою розрізняють моделювання зі сталим кроком, моделювання за особливими станами та моделювання з використанням принципу послідовної проводки заявок.

Ці поняття є ***основними*** в теорії імітаційного моделювання.

У більшості випадків кінцевою ціллю моделювання (ціллю експерименту) є оптимізація деяких параметрів системи. Залежно від цього, застосовується один з трьох найбільш поширених видів імітаційних експериментів:

• дослідження відносного впливу різних чинників на значення вихідних характеристик системи;

• знаходження аналітичної залежності між вихідними характеристиками і факторами, які цікавлять дослідника;

• знаходження оптимальних значень параметрів системи (так званий «екстремальний експеримент»).

Вид експерименту впливає не тільки на вибір схеми його формалізації, але також на побудову плану експерименту та вибір методу обробки його результатів.

З точки зору організації взаємодії дослідника з моделлю, в ході експерименту ІМ діляться на автоматичні та діалогові.

• Автоматичними називаються ІМ, взаємодія користувача з якими зводиться тільки до введення вихідної інформації та управління початком і закінченням роботи моделей.

• Діалоговими називаються ІМ, які дозволяють досліднику активно керувати ходом моделювання: припиняти сеанс моделювання, змінювати значення параметрів моделі, коригувати перелік реєстрованих даних і т. д.

Наявність процедури імітації випадкових факторів визначає імітаційне моделювання як спеціальним чином організований статистичний експеримент, в основі якого лежить метод статистичних випробувань (метод Монте-Карло). В наслідок проведення одиничного модельного експерименту отримують тільки одну реалізацію випадкової події. Тому для збору статистичної інформації про роботу об'єкта, що моделюється, проводять серію випробувань і отримують множину окремих значень спостережуваних характеристик (вибірку). Отримані статистичні дані обробляються методами математичної статистики і надаються у вигляді чисельних оцінок величин, які цікавлять дослідника.

Схема імітаційного (статистичного) моделювання як статистичного експерименту показана на рис. 8.2. Вона багато в чому нагадує схему проведення натурних експериментів в інженерній практиці, коли для отримання достатньо точної оцінки показників ефективності необхідно багаторазово повторювати експерименти, а сама система (або її модель) розглядається при цьому як чорний ящик.

Рис. 8.2

Пояснимо схему статистичного моделювання, орієнтованого на використання ЦОМ. Припустимо, що необхідно імітувати роботу системи на кінцевому інтервалі часу $[0,\ T_{м}]$. Тоді моделювання включає такі етапи.

1. Введення вихідних даних (присвоєння змінним станів z_1, ..., z_n вихідних значень z_{1_0}, ..., z_{n_0}; завдання довжини інтервалу $T_{м}$; введення інших вихідних даних про модель досліджуваної системи).

2. Завдання значень $k = 1$ лічильнику прогонів.

3. Формування поточних значень реалізації випадкових факторів (вхідних впливів, x_i, параметрів моделі a_i, тощо) на інтервалі $[0,\ T_{м}]$.

4. Обчислення поточних значень вихідних змінних y_i відповідно до моделюючого алгоритму, що імітує функціонування системи на інтервалі $[0,\ T_{м}]$.

5. Перевірка умови виконання заданого числа прогонів N, достатніх для отримання оцінок показників ефективності із зада-

ною точністю. Якщо ця умова не виконується, то цикл повторюється (перехід до етапу 6), інакше – перехід до етапу 7.

6. Присвоєння лічильнику прогонів значення $k = k + 1$. Почати новий прогін – перейти до етапу 3.

7.Обробка отриманих даних про роботу моделі за N прогонів методами математичної статистики.

8. Вивід оцінок показників ефективності та інших результатів статистичного моделювання.

У даній схемі обробка результатів моделювання не може дати точних значень необхідних критеріїв якості функціонування системи за кінцеве число прогонів. Однак якщо число прогонів N досить велике, то оцінки необхідних критеріїв набувають відповідно до закону великих чисел теорії ймовірностей статистичну стійкість, тобто зі збільшенням числа випробувань середнє арифметичне випадкових величин, отриманих в результаті одиничних експериментів, прагне до середнього арифметичного математичних сподівань і перестає бути випадковим. Це і є обґрунтуванням використання метода імітаційного моделювання як статистичного для аналізу характеристик складних систем.

При статистичному моделюванні, зазвичай, виконується умова незалежності статистичних даних, одержуваних в різних прогонах, а за показники ефективності функціонування системи, що досліджується, використовуються: ймовірності деяких подій; числові характеристики випадкових величин та процесів; розподіл ймовірностей. Так, наприклад, при вивченні характеристик надійності авіатехніки цікавляться ймовірністю безвідмовної роботи за певний інтервал часу, а при аналізі функціонування системи керування повітряним рухом - середнім рівнем безпеки та регулярності польотів.

На вихідному етапі формалізації алгоритмів імітаційного моделювання необхідно обрати метод опису динаміки системи.

8.1. Опис динаміки системи при імітаційному моделюванні

Динаміка системи описується на основі подій, робот, процесів або транзактів.

При моделюванні дискретних систем провідну роль відіграє поняття "подія". ***Подія*** являє собою миттєву зміну деякого елемента системи або стану системи в цілому.

Подія характеризується:

• умовами (або законом) виникнення;

• типом, який визначає порядок обробки (дисципліну обслуговування) даної події;

• нульовою тривалістю.

Зазвичай події поділяють на дві категорії:

• події слідування, які керують ініціалізацією процесів (або окремих робіт всередині процесу);

• події зміни станів (елементів системи або системи в цілому).

У випадку, якщо модель будується з метою вивчення причинно-наслідкових зв'язків, властивих системі, то опис її поведінки в термінах подій є "самодостатнім". Прикладом такого завдання є проектування користувальницького інтерфейсу програмного продукту, коли розробнику необхідно оцінити взаємну узгодженість елементів інтерфейсу (кнопок, прапорців, перемикачів, діалогових вікон, тощо).

Проте якщо дослідника цікавить не тільки логіка зміни станів модельованої системи, але і часові параметри її роботи, то механізм подій служить основою для подання в моделі інших описів динаміки системи, а саме: робіт, процесів і транзактів.

Робота (активність) - це одинична дія системи з обробці (перетворення) вхідних даних.

Залежно від природи модельованої системи під вхідними даними можна розуміти інформаційні дані або будь-які матеріальні ресурси. Кожна з робіт характеризується часом виконання та споживаними ресурсами. Відповідно, за допомогою моделей, описаних в термінах робіт, можуть вирішуватися завдання з оцінки якості розподілу ресурсів системи, її продуктивності, надійності, тощо.

При моделюванні складних систем часто мають місце ситуації, коли деякі роботи створюють стійку повторювану послідовність. У таких випадках буває зручніше перейти до опису моделі на основі процесів.

Під ***процесом*** розуміють логічно зв'язаний набір робіт. Деякі процеси можуть розглядатися як робота в процесі вищого рівня. І, навпаки, при підвищенні ступеня деталізації опису системи деяка робота може бути розділена на більш дрібні складові і, таким чином, перетворитися в процес.

Будь-який процес характеризується сукупністю статичних та динамічних характеристик.

До *статичних* характеристиках процесу відносяться:

- тривалість;
- результат;
- споживані ресурси;
- умови запуску (активації) та зупинки (переривання).

У загальному випадку статичні характеристики процесу не змінюються в ході його реалізації, однак при необхідності будь-яка з них може бути подана в моделі як випадкова величина, що розподілена за заданим законом.

Динамічною характеристикою процесу є його стан (активний або знаходиться в стані очікування).

Моделювання в термінах процесів здійснюється в тих же випадках, що і в термінах робіт (тобто коли система оцінюється за будь-яким часовим показником, або з точки зору споживаних ресурсів). Наприклад, при оцінці продуктивності обчислювальної мережі обробка завдань може бути подана в моделі як сукупність відповідних процесів, які використовують ресурси мережі (оперативну пам'ять, місце на жорстких дисках, процесорний час, тощо).

При описі модельованої системи в термінах робіт і процесів використовуються як події слідування (для відображення часових параметрів системи), так і події зміни станів (для подання логіки взаємодії процесів, або виконуваних робіт).

Ще один спосіб імітаційного моделювання систем заснований на використання поняття транзакта.

Транзакт - це деяке повідомлення (заявка на обслуговування), яке надходить на вхід системи і підлягає обробці. В рамках однієї ІМ можуть розглядатися транзакти декількох типів. Кожен транзакт характеризується відповідним алгоритмом обробки та необхідними для його реалізації ресурсами системи. З огляду на це, проходження транзакта по системі можна в деяких випадках розглядати як «обслуговування заявки» – послідовну активізацію процесів, що реалізують його обробку.

При розробці алгоритмів імітаційного моделювання як статистичного основна увага приділяється питанням генерування випадкових величин і отримання послідовностей випадкових величин, розподілених за заданими законами.

8.2. Формування випадкових факторів при імітаційному моделюванні

Імітаційне моделювання дозволяє досліджувати поведінку різних систем з урахуванням впливу випадкових факторів. Якщо при аналітичному моделюванні стохастичних систем основними випадковими чинниками є випадкові процеси, то при імітаційному моделюванні ці фактори найчастіше відображаються в ІМ як випадкові величини (дискретні або безперервні) або як випадкові події (потоки випадкових подій).

Наприклад, якщо за допомогою створюваної ІМ передбачається дослідити надійність обчислювальної системи, то виникнення відмови доцільно подати в моделі як випадкову подію. Проте якщо модель призначена для оцінки часових параметрів процесу обслуговування пасажирів в аеропорту, то інтервал часу до появи чергового пасажира зручніше описати як випадкову величину, розподілену за деяким законом.

Як вже зазначалося, в основі методів моделювання випадкових факторів лежить використання рівномірно розподілених на інтервалі [0; 1] ВЧ, які генеруються певними способами.

8.2.1. Моделювання випадкових подій

Використовуючи послідовність ВЧ, можна моделювати результати незалежних і залежних випробувань. Задача подібного роду виникає при побудові алгоритмів, що імітують недостовірне обслуговування клієнтів, надійність апаратури, наявність запасу матеріалів, тощо.

Для моделювання *випадкової події A*, ймовірність якої дорівнює P_C, достатньо сформувати одне ВЧ r, рівномірно розподілене на інтервалі [0,1]. При попаданні r в інтервал $[0, P_C]$ вважається, що подія A настала, в іншому випадку не настала.

Наприклад, ймовірність відмови обчислювальної системи становить 0,3. Щоб визначити, чи виникне відмова на черговому кроці моделювання, необхідно згенерувати за допомогою генератора одне ВЧ r і порівняти його з ймовірністю відмови (рис. 8.3).

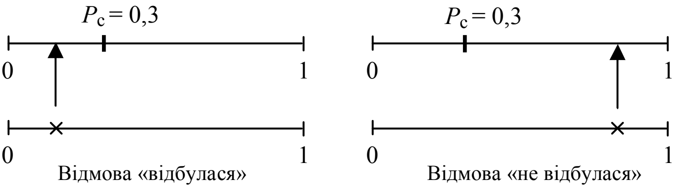


Рис. 8.3

При значному числі експериментів частота потрапляння чисел в інтервал від 0 до 0.3 буде наближатися до ймовірності $P_C = 0.3$, а частота потрапляння чисел в інтервал від 0.1 до 1 буде наближатися до $P_C = 0.7$. Фрагмент алгоритму представлений на рис. 8.4.

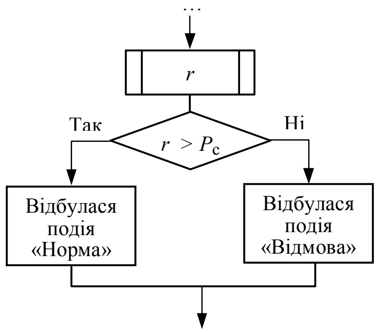


Рис. 8.4

Для моделювання повної групи N *несумісних подій* $A=\{A_1, A_2, ... A_N\}$ з вірогідністю відповідно P_1, P_2,..., P_N також досить одного значення r: подія A_i з групи A вважається такою, що настала, якщо виконується умова:

$$A = A_i \left| \sum_{i=0}^{n-1} P_i < r < \sum_{i=0}^{n} P_i \quad n = 1, N \right. .$$

Припустимо, що в кожен момент часу може відбуватися звернення вигляду $A = A_i \left| \sum_{i=0}^{n-1} P_i < r < \sum_{i=0}^{n} P_i \right.$ тільки до одного з трьох модулів оперативної пам'яті обчислювальної системи. Ймовірності звернення до кожного з них P_1, P_2 и P_3 дорівнюють відповідно 0.3, 0.5 и 0.2. Щоб дізнатися, з якого саме модуля будуть зчитані дані, необхідно визначити, в який інтервал потрапить отримане від генератора ВЧ r (рис. 8.5).

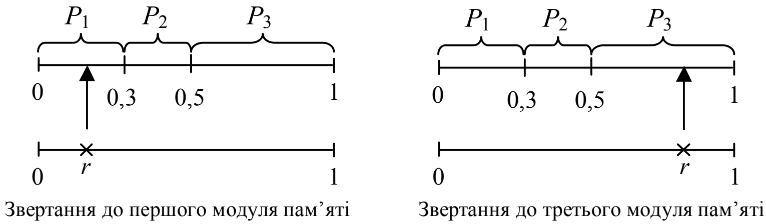


Рис. 8.5

Якщо група подій A не повна, то додатково вводять фіктивну подію A_{N+1} з ймовірністю P_{N+1} такою, що сума ймовірностей перетворюється в одиницю:

$\sum\limits_{i=0}^{N+1} P_i = 1$, тобто доповнюють групу A до повної.

Після цього генерують число r і перевіряють вказану вище умову. При $A = A_{N+1}$ вважається, що жодна подія з вихідної групи A не відбулася.

На рис. 8.6 показана блок-схема, яка реалізує описаний алгоритм. Алгоритм визначає за допомогою фільтра, побудованого у вигляді низки умовних операцій (IF), в якій з інтервалів: від 0 до P_1; від P_1 до ($P_1 + P_2$); від ($P_1 + P_2$) до ($P_1 + + P_2 + P_3$) і так далі – потрапило число, що генерується генератором ВЧ. Відповідно у великі відрізки ВЧ будуть потрапляти частіше (ймовірність появи цих подій більше), в менші відрізки - рідше. Якщо число потрапило в якийсь з інтервалів (що відбудеться завжди й обов'язково), то це відповідає здійсненню пов'язаної з ним події.

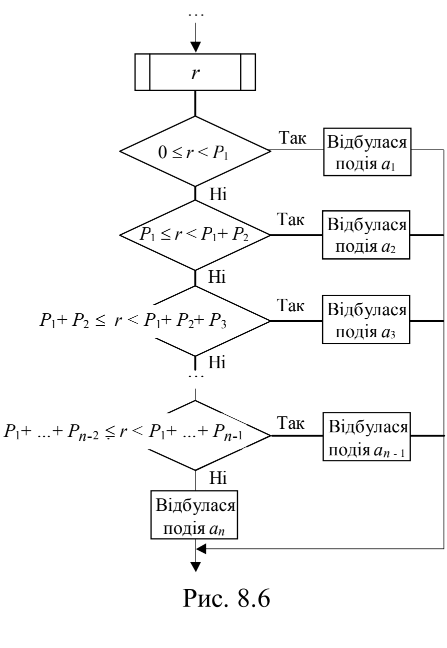


Рис. 8.6

Для імітації залежних подій A и B (B залежить від

A) необхідно знати безумовну ймовірність $P(A)$ події A і умовні ймовірності $P(B/A)$ і $P(B/\bar{A})$. Спочатку описаним вище способом імітується поява події A, залежно від результату вибирається одна з ймовірностей $P(B/A)$ або $P(B/\bar{A})$, і за тією ж технологією визначається поява події B.

Розглянемо принципи моделювання сумісних подій. Нехай події A_1 і A_2 незалежні та сумісні. Імовірність появи події A_1 дорівнює $P(A_1)$, а події $\tilde{A_2}$ $P(A_2)$. Можливі два способи моделювання.

1. Із сукупності ВЧ обираємо два числа ξ_1 і ξ_2 та перевіряємо умови $\xi_1 \le P(A_1)$; $\xi_2 \le P(A_2)$. Якщо обидві умови виконані, то наслідком випробування є подія A_1A_2; якщо перша умова істинна, а друга – хибна, то наслідком випробування є подія $A_1\overline{A}_2$ и т. д.

2. Сумісні незалежні випробування A_1 і A_2 мають повну групу наслідків A_1A_2, $A_1\overline{A}_2$, $\overline{A}_1A_2$, $\overline{A}_1\overline{A}_2$, з ймовірностями:

$$P(A_1)P(A_2);\ P(A_1)[1-P(A_2)];$$
$$[1-P(A_1)]P(A_2);\ [1-P(A_1)][1-P(A_2)].$$

Обираючи ВЧ і порівнюючи їх із зазначеними ймовірностями, знаходимо можливість того чи іншого наслідку.

Нехай сумісні події A_1 і A_2 незалежні та відбуваються з ймовірностями $P(A_1) = p_1$ и $P(A_2) = p_2$. Імовірність появи події A_2 за умови, що A_1 відбулася, задана і дорівнює $P\{A_2/A_1\} = p_3$. Моделювання наслідків залежних випробувань, як і в попередньому випадку, може здійснюватися двома способами.

1. Із сукупності ВЧ ξ виберемо деяке число ξ_1 і перевіримо умову $\xi_1 \le p_1$. Якщо умова виконана (тобто відбулася подія A_1), то обираємо нове ВЧ ξ_2 і перевіряємо умову $\xi_2 \le p_2$.

Якщо ця умова істинна, то результатом сумісного випробування є подія A_1A_2; якщо умова не виконана, то – $A_1\overline{A}_2$.

У процесі моделювання може виявитися, що умова $\xi_1 \le p_1$ не виконана, тоді на другому етапі значення ξ_2 необхідно порівнювати з ймовірністю $P(A_2/\overline{A}_1) = (p_2 - p_1p_3)/(1-p_1)$

2. Другий спосіб моделювання зводиться до попереднього визначення ймовірностей всіх наслідків залежних випробувань, які складають повну групу.

В результаті випробувань можуть відбутися такі події: $A_1 A_2$, $A_1 \overline{A}_2$, $\overline{A}_1 A_2$, $\overline{A}_1 \overline{A}_2$. Ймовірності їх наслідків:

$$P(A_1 A_2) = p_1 p_3;$$
$$P(A_1 \overline{A}_2) = p_1(1 - p_3);$$
$$P(\overline{A}_1 A_2) = (1 - p_1)\frac{p_2 - p_1 p_3}{1 - p_1} = p_2 - p_1 p_3;$$
$$P(\overline{A}_1 \overline{A}_2) = (1 - p_1)[1 - P(A_2 / \overline{A}_1)].$$

Процедура моделювання зводиться до вибору одного з наслідків відповідно до отриманих ймовірностей.

8.2.2. Формування випадкових величин

Більшої інформативністю, у порівнянні з такими статистичними характеристиками випадкових чисел як математичне сподівання, дисперсія, для інженера має закон розподілу ймовірності випадкової величини X. Припустимо, що X приймає випадкові значення з деякого діапазону. Наприклад, X – об'єм електроліту, що заливається в акумулятор. Цей об'єм може відхилятися від запланованого ідеального значення під впливом різних факторів, які не можна врахувати, тому він є випадковою слабо прогнозованою величиною. Але при тривалому спостереженні за процесом заправлення можна визначити, у скільки акумуляторів з 1000 було залито електроліту X_1 (позначимо N_{X1}), у скільки – X_2 (позначимо N_{X2}) і так далі. У результаті можна побудувати гістограму частості об'ємів, відкладаючи для X_1 величину $N_{X1}/1000$, а для X_2 величину $N_{X2}/1000$ і так далі. (Якщо бути точним, N_{X1} – це кількість акумуляторів, в яких об'єм залитого електроліту не просто дорівнює X_1, а знаходиться в діапазоні від $X_1 - \Delta/2$ до $X_1 + \Delta/2$, де $\Delta = X_1 - X_2$). Важливо, що сума всіх частостей буде дорівнює 1 (сумарна площа гістограми незмінна). Якщо X змінюється безперервно, експериментів проведено багато, то гістограма поступово перетворюється в графік розподілу ймовірності випадкової величини. На рис. 8.7 *а* показаний приклад гістограми дискретного розподілу, а на рис. 8.7 *б* – варіант безперервного розподілу випадкової величини.

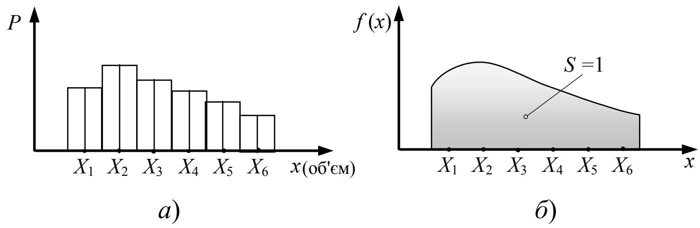


Рис. 8.7.

При моделюванні ***безперервних*** випадкових величин із заданим законом розподілу можуть використовуватися три методи:

- метод нелінійних перетворень;
- метод композицій;
- табличний метод.

Перші два методи вимагають від розробника моделі серйозною математичної підготовки. Третій метод, умовно названий «табличним», заснований на заміні закону розподілу безперервної випадкової величини спеціальним розрахунковим співвідношенням, яке дозволяє обчислювати значення випадкової величини за значенням випадкового числа, рівномірно розподіленого на тому ж інтервалі [0,1]. Такі співвідношення отримані практично для всіх найбільш поширених видів розподілів і наведені в довідниковій літературі. Як приклад нижче дані розрахункові співвідношення для двох законів розподілу:

– показового

$$x = -\frac{1}{\lambda}\ln(r),$$

тут λ – параметр показового розподілу, r – рівномірно розподілене випадкове число;

– нормального

$$x = m + s\left[\sum_{i=1}^{12} r_i - 6\right],$$

де m, s – параметри нормального закону розподілу, r_i – рівномірно розподілене випадкове число.

Можна моделювання безперервних випадкових величин із заданим законом розподілу замінити моделюванням дискретних випадкових величин. Для цього безперервний закон розподілу ймовірності події необхідно перетворити в дискретний. Для цього необхідно генеровані випадкові числа перетворити в заданий користувачем закон розподілу ймовірності, наприклад, в нормальний або експонентний закон. Одним з варіантів такого перетворення є ***метод східчастої апроксимації.***

На рис. 8.8 графічно показаний перехід від довільного безперервного закону розподілу до дискретного.

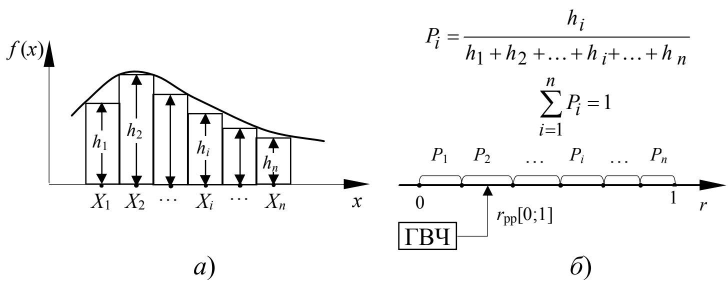


Рис. 8.8.

Тут h_i – висота i-го стовпця, $f(x)$ – розподіл імовірності (показує наскільки ймовірна деяка подія x).

Значення h_i операцією нормування необхідно перевести в одиниці ймовірності появи значень x з інтервалу $x_i < x \le x_{i+1}$:

$$P_i = h_i/(h_1 + h_2 + \ldots + h_i + \ldots + h_n).$$

Операція нормування забезпечує суму ймовірностей всіх n подій рівну одиниці:

$$\sum_{i=1}^{n} P_i = 1.$$

На рис. 8.8*а* показано відображення одержуваних ймовірностей на інтервал $r_{рр}[0; 1]$, а на рис. 8.8*б* – генерація випадкових подій з використанням генератора рівномірно розподілених випадкових чисел (ГВЧ).

Зауважимо, що всередині інтервалу $x_i < x \le x_{i+1}$ значення x тепер не помітне, тобто однаково. Метод округлює точність початкової постановки задачі, переходячи від безперервного закону розподілу до дискретного. Тому слід враховувати кількість розбивок n з умов точності подання.

На рис. 8.9 показаний фрагмент алгоритму, що реалізує описаний метод. Алгоритм генерує ВЧ, рівномірно розподілене від 0 до 1. Потім, порівнюючи границі відрізків, що розташовані на інтервалі від 0 до 1 і які являють собою ймовірності P появи тих чи інших випадкових величин X, визначає в циклі, яка з випадкових подій i в результаті цього виникла.

Переходячи до моделювання дискретних випадкових величин, зазначимо, що тут найбільш часто використовуються два методи:

- метод послідовних порівнянь;
- метод інтерпретації.

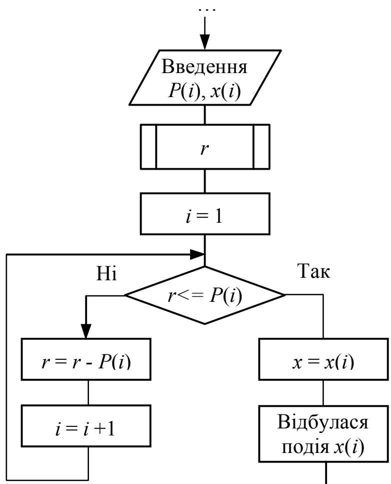


Рис. 8.9.

<u>Метод послідовних порівнянь</u>

Число r послідовно порівнюють зі значенням суми $P_1+P_2+...$, де P_1 – ймовірність найменшого значення випадкової величини Y; P_2 – ймовірність другого за величиною значення. При першому виконанні умови $r > \sum_{i=1}^{n} P_i$ перевірка припиняється і вважається, що дискретна випадкова величина Y набула значення y_i:

Процес можна прискорити, застосовуючи методи оптимізації перебору: метод дихотомії, метод ранжирування і т. д. Величини P_i розраховують за функціями розподілу ймовірності, відповідними модельованому закону.

<u>Метод інтерпретації</u>

Метод заснований на фізичному трактуванні модельованого закону розподілу. Наприклад, біноміальний розподіл описує число успіхів в n незалежних випробуваннях з ймовірністю успіху в кожному випробуванні P і ймовірністю невдачі $g = 1 - P$.

При моделюванні цього розподілу за допомогою методу інтерпретації вибирають n незалежних ВЧ, рівномірно розподілених на інтервалі [0,1], і підраховують кількість тих з них, які менше P. Це число і є модельованою випадковою величиною.

В одної і той самої імітаційної моделі можуть фігурувати різні випадкові фактори, одні можуть бути подані як випадкові події, інші – як випадкові величини. Більш того, якщо моделюється поведінка досить складної системи, то її функціонування може бути пов'язано з виникненням декількох типів подій і з врахуванням великої кількості випадкових величин, розподілених за різними законами. Якщо ж моделювання всіх випадкових факторів засноване на використанні одного датчика, який генерує одну «загальну» послідовність ВЧ, то з математичної точки зору їх не можна вважати незалежними. У зв'язку з цим для моделювання кожного випадкового фактора намагаються використовувати окремий генератор ВЧ, або, принаймні, забезпечувати створення нової послідовності ВЧ. У багатьох спеціалізованих мовах і пакетах моделювання така можливість передбачена.

8.2.3. Моделювання випадкових потоків подій

Випадкові потоки являють собою специфічний точковий випадковий процес. Зокрема переходи в СМО з одного стану в інший відбуваються під впливом цілком певних подій – надходження заявок та їх обслуговування.

Потоком подій називається послідовність однорідних подій, які відбуваються одна за іншою в деякі моменти часу. Наприклад, це потік відмов або потік заявок на обслуговування обчислювального комплексу. Події потоку відбуваються в заздалегідь невідомі (зазвичай випадкові) моменти часу t_1, t_2,…, тому потік подій зручно зображувати низкою точок на осі часу t (рис. 8.10).

Регулярним (або детермінованим) потоком подій називається

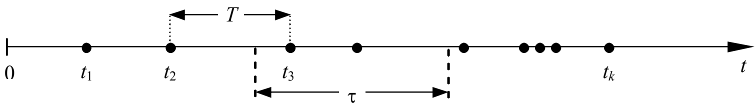


Рис. 8.10.

потік, в якому події слідують одна за іншою через однакові проміжки часу**.**

Важливою характеристикою будь-якого потоку є число m подій, які відбулися за час τ. Якщо m випадкова величина з можливими значеннями $x \in \{0, 1, 2, ...\}$, то розглядають або ймовірності

***p*(*m* = *x*),** або математичні сподівання m_x і дисперсії σ_x^2, які залежать від τ.

Потік подій називається стаціонарним, якщо при будь-якому *x* величина $p(m = x)$ залежить тільки від довжини ділянки τ і не залежить від її розташуванні на осі часу.

Ординарним потоком подій називається такий потік, для якого ймовірність $p(m \geq 2)$ потрапляння на відрізок τ дуже мала в порівнянні з імовірністю $p(m = x)$. Тобто практично неможливо потрапляння двох і більше подій на досить малий інтервал τ.

Потоком без післядії називається потік подій, в якому для будь-яких двох неперетинних інтервалів τ_1 і τ_2 осі *t* число подій m_1, що потрапляють на τ_1, не залежить від того, скільки подій m_2 потрапило на τ_2. Це говорить про те, що події, які створюють потік, з'являються в послідовні моменти часу незалежно один від одного.

Потік подій називається ***найпростішим*** (або стаціонарним пуасоновським), якщо він володіє одночасно трьома властивостями: стаціонарністю, ординарністю і відсутністю післядії. Інтерес до цього потоку викликаний простотою формального опису його статистичних характеристик. Крім того, при складанні декількох незалежних випадкових потоків, стаціонарних і ординарних, утворюється сумарний потік, який за своїми характеристиками наближається до найпростішого. Тому, при дослідженні реальних потоків прагнуть звести їх до найпростіших. Для цього нестаціонарні потоки подають як стаціонарні на обмеженому відрізку часу. Неординарні потоки зводять до ординарних, розглядаючи кілька одночасно наступаючих подій як одну подію і т. д.

Стаціонарний пуасоновський потік це потік подій, який не має післядії, є ординарним, але нестаціонарним.

У найпростішому потоці (як стаціонарному, так і нестаціонарному) величина *m* підпорядковується розподілу Пуассона:

$$p(m = x) = \frac{(\lambda\tau)^x}{x!} e^{-\lambda x}.$$

Тут $p(m = x)$ – ймовірність того, що в проміжок часу τ потрапляє рівно *x* подій; λ – інтенсивність потоку (середнє число подій в одиницю часу).

Якщо a – середнє число подій, яке припадає на проміжок τ, то для стаціонарного найпростішого потоку $a = \lambda\cdot\tau$, а для нестаціонарного пуасоновского потоку $\lambda = \lambda(\tau)$, $a = \int\limits_{t_0}^{t_0+\tau} \lambda(t)dt$.

Важливою характеристикою потоку подій є закон розподілу довжини проміжку між сусідніми подіями. Для найпростішого потоку з інтенсивністю λ довжина цього проміжку T (рис. 8.10) розподілена за показовим (експоненціальним) закону з щільністю:

$$f(t) = \begin{cases} 0, & t < 0; \\ \lambda e^{-\lambda t}, & t \geq 0. \end{cases}$$

Числові характеристики випадкової величини, розподіленої за показовим законом, а саме, математичне очікування і дисперсія відповідно дорівнюють:

$$m_t = \frac{1}{\lambda}, \quad D_t = \sigma_t^2 = \frac{1}{\lambda^2}.$$

Потоком Пальма (або потоком з обмеженої післядією) називається потік подій, для якого проміжки часу між послідовними подіями T_1, T_2,... , T_n являють собою незалежні, однаково розподілені випадкові величини.

Вочевидь, що найпростіший потік є окремим випадком потоку Пальма, коли інтервали між подіями мають показовий розподіл. Наприклад, потік несправностей системи, яка складається з певної кількості елементів, утворює потік Пальма за умовою, що елементи системи незалежно один від одного виходять з ладу, але тут же несправний елемент замінюється новим.

Особливий клас потоків Пальма створюють ***потоки Ерланга***, які отримують шляхом прорідження найпростіших потоків, тобто відкиданням деяких подій як таких, що не відбулися. Якщо в найпростішому потоці зберігається кожна k-та подія (відраховуючи від умовно першої), а інші просто не враховуються, то виникає потік Ерланга k-го порядку (k = *1*,2,...), який позначається як $Э_k$. На рис. 8.11 наведений потік Ерланга 3-го порядку (дві події найпростішого потоку пропускаються, а третя враховується).

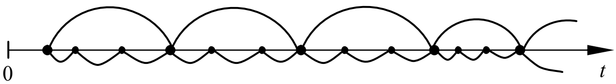


Рис. 8.11

Вочевидь, що інтервал часу T_k між будь-якими сусідніми подіями в потоці $Э_k$ – є сума k незалежних випадкових величин, а саме відстаней між відповідними подіями найпростішого потоку. Ці незалежні випадкові величини розподілені за показовим законом:

$$f_k(t) = \frac{\lambda(\lambda t)^{k-1}}{(k-1)!} e^{-\lambda t}, \quad t > 0 .$$

При $k = 1$ отримують:

$$f_1(t) = e^{-\lambda t}, \; t > 0 ,$$

тобто найпростіший потік є потоком $Э_1$. Числові характеристики величини T_k (математичне очікування та дисперсія) такі:

$$m_k = \frac{k}{\lambda}, \quad \sigma_t^2 = \frac{k}{\lambda^2} .$$

Слід звернути увагу на те, що потоки Ерланга є стаціонарними і простими, але володіють післядією, яка посилюється зі збільшенням k.

Приклади моделювання випадкових потоків подій.

Розглянемо потік виробів, які надходять на технологічну операцію. Вироби надходять випадковим чином – у середньому вісім штук за добу (інтенсивність потоку $\lambda = 8/24$ [од/год]). Необхідно промоделювати цей процес протягом $T_н = 100$ годин при $m = 1/\lambda = 24/8 = 3$, тобто в середньому одна деталь за три години. На рис. 8.12 представлений алгоритм, що генерує такий потік випадкових подій.

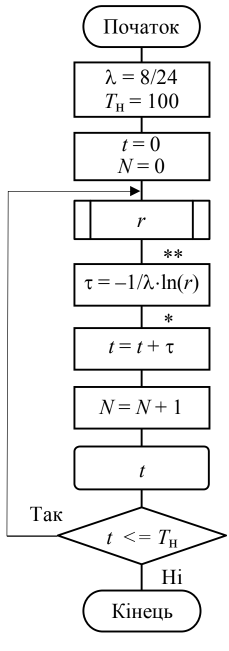


Рис.8.12

Якщо відомо, що потік не є ординарним, то необхідно моделювати крім моменту виникнення події ще й число подій, яке могло з'явитися в цей момент.

Наприклад, вагони на залізничну станцію прибувають в складі потяга у випадкові моменти часу (ординарний потік потягів). Але при цьому в складі потяга може бути різна (випадкова) кількість вагонів. У цьому випадку про потік вагонів говорять як про потік неординарних подій.

Припустимо, що $M_k = 10$, $\sigma = 4$, тобто, в середньому в 68 випадках зі 100 приходить від 6 до 14 вагонів у складі потяга і їх число розподілено за нормальним законом. В місце, зазначене (*) в попередньому алгоритмі (див. рис. 8.12), потрібно вставити фрагмент, показаний на рис. 8.13.

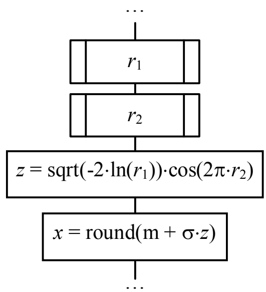


Рис.8.13

У ряді випадків інтенсивність потоку може змінюватися з часом $\lambda(t)$. Такий потік називається ***нестаціонарним***. Наприклад, середня кількість за годину машин швидкої допомоги, які покидають станцію за викликами населення великого міста, протягом доби може бути різною. Відомо, наприклад, що найбільша кількість викликів припадає на інтервали з 23 до 01 години ночі і з 05 до 07 ранку, тоді як в інші години вона вдвічі менше (див. рис. 8.14).

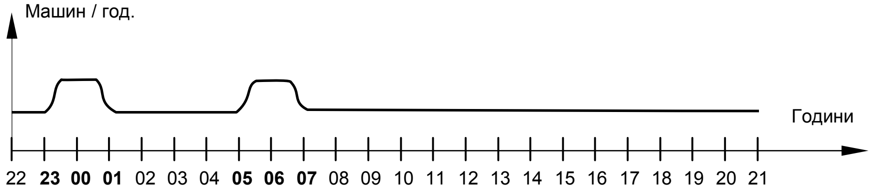


Рис. 8.14

У цьому випадку розподіл $\lambda(t)$ може бути заданий або графіком, або формулою, або таблицею. А в алгоритмі, показаному на рис. 8.13, в місце, позначене (**), потрібно буде вставити фрагмент, який формує даний розподіл $\lambda(t)$.

8.3. Керування модельним часом

8.3.1. Види подання часу в моделі

Імітаційний експеримент являє собою спостереження за поведінкою системи протягом деякого проміжку часу. Звичайно, не у всіх статистичних випробуваннях фактор часу відіграє провідну роль, а в деяких і взагалі може не розглядатися. Але значно більше задач, в яких оцінка ефективності модельованої системи прямо пов'язана з характеристиками часу її функціонування. До них відносяться задачі з оцінці продуктивності, деякі задачі з оцінці надійності, якості розподілу ресурсів, а також всі задачі, пов'язані з дослідженнями ефективності процесів обслуговування. Характерною рисою більшості практичних задач є те, що швидкість протікання розглянутих у них процесів значно нижче швидкості реалізації модельного експерименту. З іншого боку, навіть ті імітаційні експерименти, в яких параметри часу роботи системи не враховуються, вимагають для своєї реалізації певних витрат часу роботи комп'ютера (машинного часу).

У зв'язку із цим при розробці практично будь-якої імітаційної моделі й плануванні проведення модельних експериментів необхідно співвідносити між собою три подання часу:

- реальний час, у якому відбувається функціонування системи, що імітується;
- модельний (або, як його ще називають, системний) час, у масштабі якого організується робота моделі;
- машинний час, який показує витрати часу обчислювальної машини на проведення імітації.

За допомогою механізму модельного часу розв’язуються такі задачі:

- відображається перехід модельованої системи з одного стану в інший;
- здійснюється синхронізація роботи компонент моделі;
- змінюється масштаб часу функціонування досліджуваної системи;
- здійснюється керування ходом модельного експерименту;
- моделюється квазіпаралельна реалізація подій у моделі.

Приставка «квазі» у даному випадку відображає послідовний характер обробки подій (процесів) в ІМ, які в реальній системі виникають (протікають) одночасно.

Необхідність розв‘язання останньої задачі пов'язана з тим, що в розпорядженні дослідника знаходиться, зазвичай, однопроцесорна обчислювальна система, а модель може містити значно більше число одночасно працюючих підсистем. Тому дійсно паралельна (одночасна) реалізація всіх, компонент моделі неможлива. Навіть використання, так званої, розподіленої моделі, яка реалізована на декількох вузлах обчислювальної мережі, не гарантує, що число вузлів буде збігатися з числом одночасно працюючих компонент моделі. Слід зазначити, що реалізація квазіпаралельної роботи декількох компонент моделі є досить складним технічним завданням.

Раніше були названі методи реалізації механізму модельного часу – зі сталим кроком, за особливими станами та метод послідовної проводки заявок.

Вибір методу реалізації механізму модельного часу залежить від призначення моделі, її складності, характеру досліджуваних процесів, необхідної точності результатів, тощо.

8.3.2. Зміна часу зі сталим кроком

При використанні даного методу відлік системного часу здійснюється через фіксовані, обрані дослідником інтервали часу. Події в моделі вважаються такими, що настали в момент закінчення цього інтервалу. Похибка у вимірюванні часових характеристик системи в цьому випадку залежить від величини кроку моделювання Δt. Метод сталого кроку доцільно використовувати в тому випадку, якщо:

- події відбуваються регулярно, їх розподіл у часі досить рівномірний;
- число подій велике і моменти їх появи досить близькі;
- неможливо заздалегідь визначити моменти появи подій.

Даний метод управління модельним часом досить просто реалізувати у тому разі, коли умови появи подій всіх типів у моделі можна подати як функцію часу.

Нехай, наприклад, подія полягає в тому, що літак перетинає деякий повітряний кордон, відстань до якого дорівнює R. Якщо літак рухається прямо з постійною швидкістю V, то можна обчислювати шлях, пройдений літаком, за інтервал часу Δt: $S = S_0 + V\cdot\Delta t$.

Відповідно, подія вважається такою, що відбулася, якщо виконується умова $S > R$, а момент часу настання події приймається рівним $n \cdot \Delta t$, де n – номер кроку моделювання, умова стало істинним.

У загальному вигляді алгоритм моделювання зі сталим (постійним) кроком наведений на рис. 8.15 (на рисунку $t_М$ – поточне значення модельного часу, $T_М$ — заданий інтервал моделювання), а для розглянутого вище прикладу з літаком — на рис. 8.16.

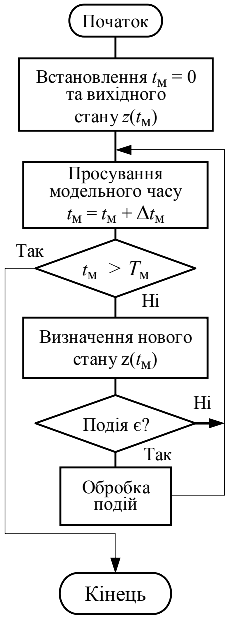


Рис.8.15

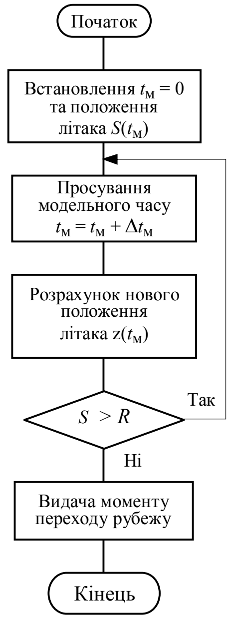


Рис.8.16

Зверніть увагу на те, що на відміну від узагальненого алгоритму, в наведеному прикладі моделювання завершується не після закінчення заданого інтервалу часу, а при настанні події, яка нас цікавить. У зв'язку з цим необхідно ще раз підкреслити, що при моделюванні зі сталим кроком результат моделювання безпосередньо залежить від величини цього кроку. Причому, якщо крок буде занадто великим, то результат, швидше за все, буде невірним: момент закінчення чергового кроку дуже рідко буде збігатися з реальним моментом перетину літаком заданого кордону. Така ситуація показана на рис. 8.17, де використані такі позначення:

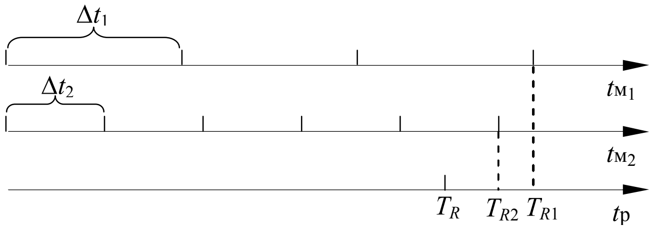


Рис. 8.17

$t_{М1}$ – вісь модельного часу при використанні кроку Δt_1;

$t_{М2}$ – вісь модельного часу при використанні кроку Δt_2;

$t_р$ – вісь реального часу;

T_R – реальний момент перетину літаком кордону;

T_{R1}, T_{R2} – моменти перетину кордону, що отримані для відповідних величин Δt.

Наведений приклад показує, що вибір величини кроку моделювання є складним і дуже суттєвим питанням. Універсальної методики розв'язання цієї проблеми не існує, але у багатьох випадках можна використовувати один з таких підходів:

- приймати величину кроку, яка дорівнює середньому значенню інтенсивності виникнення подій різних типів;
- вибирати величину кроку, яка дорівнює середньому інтервалу між найбільш частими (або найбільш важливими) подіями.

8.3.3. Зміна часу за особливими станами

При моделюванні за особливими станами системний час щоразу змінюється на величину, що суворо відповідає інтервалу часу до моменту настання чергової події. В цьому випадку події обробляються у порядку їх настання, а подіями, що настали одночасно, вважаються тільки ті, які є одночасними у дійсності.

Для реалізації моделювання за особливими станами потрібна розробка спеціальної процедури планування подій (так званого календаря подій).

Якщо відомий закон розподілу інтервалів між подіями, то таке прогнозування труднощів не становить: досить до поточного значення модельного часу додати величину інтервалу, отриману за допомогою відповідного датчика.

Нехай, наприклад, за літаком, що фігурував у попередньому описі моделювання зі сталим кроком, спостерігає диспетчер. Він вводить інформацію про літак, причому інтервали між введеннями двох сусідніх повідомлень є випадковими величинами, розподіленими за нормальним законом із заданими параметрами.

Ілюстрація до такої ситуації наведена на рис. 8.18.

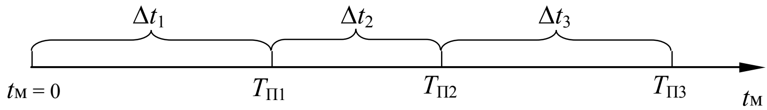


Рис. 8.18

Тут $T_{\Pi i}$ – момент введення чергового повідомлення; Δt_i – випадковий інтервал; $T_{\Pi 1} = t_{М} + \Delta\ t_1$; $T_{\Pi 2} = t_{М} + \Delta\ t_2$; $T_{\Pi 3} = t_{М} + \Delta\ t_3$.

Якщо ж момент настання події визначається деякими логічними умовами, то необхідно сформулювати ці умови і перевіряти їх істинність для кожного наступного кроку моделювання. Практика показує, що складності в реалізації механізму зміни часу за особливими станами пов'язані у першу чергу з коректним описом таких умов. Труднощі ще більш зростають, якщо в моделі фігурують кілька типів взаємопов'язаних подій.

Узагальнена схема алгоритму моделювання за особливими станами представлена на рис. 8.19, де $t_{п_i}$ – прогнозований момент настання i-ої події.

Моделювання за особливими станами доцільно використовувати, якщо:

- події розподіляються в часі нерівномірно або інтервали між ними великі;
- пред'являються підвищені вимоги до точності визначення взаємного положення подій в часі;
- необхідно враховувати наявність одночасних подій.

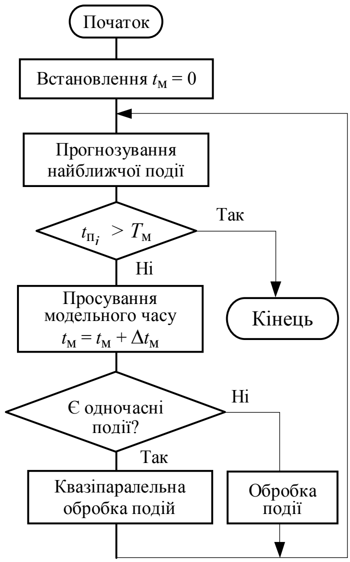


Рис.8.19

Додаткова перевага методу полягає в тому, що він дозволяє економити машинний час, особливо при моделюванні систем періодичної дії, в яких події тривалий час можуть не відбуватися.

Щоб «відчути різницю» в використанні двох методів управління модельним часом, повернемось до нашого прикладу з диспетчером. Доповнимо його наступною умовою: необхідно підрахувати число повідомлень, які встигне ввести диспетчер протягом заданого інтервалу моделювання.

Перш за все, необхідно відповісти на питання: що розуміти під «особливими станами», які повинні впливати на зміну модельного часу? На практиці зазвичай замість станів розглядають події, що визначають зміну станів модельованого процесу, Для процесу введення інформації диспетчером такий перехід виконується досить просто: подія – це введення чергового повідомлення; іншими словами, введення чергового повідомлення «просуває» модельний час на відповідний інтервал. Так, якщо інтервали між повідомленнями підкоряються нормальному закону з параметрами m и s, то чергове i-те значення модельного часу $t_m(i)$ визначається так:

$$t_m(i) = t_m(i-1) + \mathrm{norm}(m,s).$$

У цьому виразі доданок norm(m, s) означає звернення до генератора ВЧ, розподілених за нормальним законом.

Алгоритм роботи моделі наведено на рис. 8.20 (де N - число введених повідомлень).

Підводячи підсумки, відзначимо.

- Вибір механізму зміни модельного часу визначає технологію реалізації ІМ.
- На вибір методу моделювання впливає цілий ряд факторів, однак, визначальним є тип модельованої системи: для дискретних систем, події в яких розподілені в часі нерівномірно, зручнішим є зміна модельного часу за особливими станами.

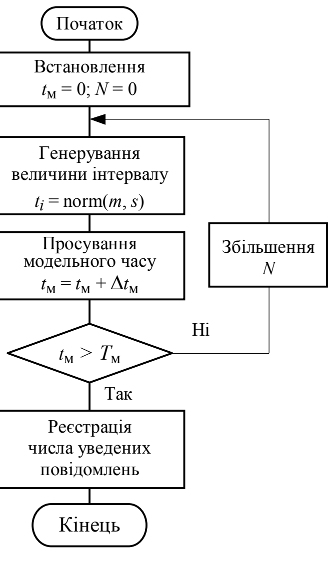


Рис.8.20

- Якщо в моделі повинні бути представлені компоненти реальної системи, робота яких вимірюється в різних одиницях часу, то вони повинні бути попередньо приведені до єдиного масштабу.

8.3.4. Моделювання послідовної проводки заявок

Принцип послідовної проводки заявок полягає в тому, що кожна заявка відстежується від моменту надходження її в систему до моменту її виходу з системи. І тільки потім розглядається наступна заявка.

Розглянемо роботу алгоритму на прикладі двоканальної системи масового обслуговування заявок з двома місцями в загальній черзі (див. рис. 8.21).

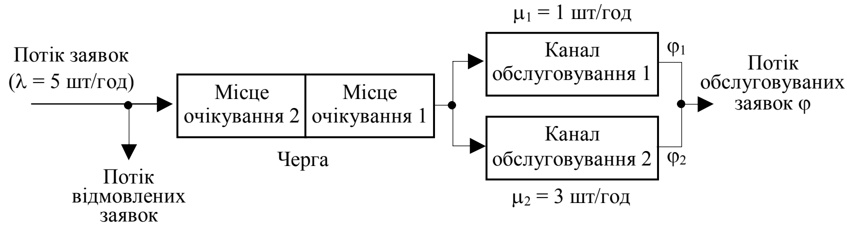


Рис. 8.21.

Позначимо: λ – інтенсивність приходу заявки; μ_i – інтенсивність обслуговування заявки.

Побудуємо алгоритм, який визначає ймовірності обслуговування заявок і відмови заявок, а також пропускну здатність системи. Блок-схема алгоритму показана на рис. 8.23

Генеруємо заявку (блоки 3, 4 - тут і далі див. блоки на рис. 8.23). Випадковий момент часу, коли заявка прийшла в систему, дорівнює $T_{\text{в}}$. Час між двома випадковими заявками імітується формулою $\tau = -1/\lambda \cdot \mathrm{Ln}(r)$, яке додається до $T_{\text{с}}$ попередньої заявки $T_{\text{с}} = T_{\text{с}} - 1/\lambda \cdot \mathrm{Ln}(r)$. Врахуємо цей факт в лічильнику N (блок 4) заявок, що надійшли.

Послідовно порівнюємо $T_{\text{с}}$ у порядку пріоритетів (блоки 5, 6, 7, 8) з часом звільнення: каналу 1 – T_1; каналу 2 – T_2; місця в черзі 1 – T_3; місця в черзі 2 – T_4. Як тільки виявляється факт, що місце в системі вільно (див. рис. 8.22): $T_{\text{с}} > T_1$, або $T_{\text{с}} > T_2$, або $T_{\text{с}} > T_3$, або $T_{\text{с}} > T_4$, негайно заявка переводиться на вільне місце і імітується її обробка на цьому місці. В іншому випадку заявка отримує відмову (блок 11).

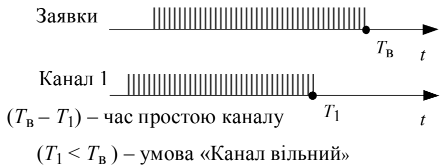


Рис. 8.22

Припустимо, що звільнилося місце у першому каналі. Обробка полягає в тому, що обчислюється час простою цього каналу до приходу заявки (наприклад, $T_с – T_1$), і цей час додається у лічильник часів простою (блок 15). Обчислюється наступний час зміни стану каналу – модифікується змінна T_1, яка вкаже надалі, коли знову звільниться канал.

Величина T_1 змінюється на величину $\tau := –1/\mu_1 \cdot \mathrm{Ln}(r)$ – час обслуговування, який відлічується від початку обслуговування $T_с$. Лічильник обслужених заявок $N_{об}$ збільшується на одиницю.

Аналогічно обробка відбувається і в другому каналі, якщо заявка потрапляє саме туди (блок 14).

Особливість обробки заявки в черзі полягає в тому, що перше місце в черзі звільняється, коли звільняється місце в одному з каналів, звичайно, заявка йде туди, де місце звільняється раніше (блоки 5, 6). Друге місце в черзі звільняється одночасно з першим, оскільки заявка в черзі пересувається на перше вільне місце (блок 12).

Далі алгоритм генерує в циклі наступну заявку (блоки 3, 16). Зупинка моделювання відбувається тоді, коли кожна лінійка буде заповнена до T_k (блок 16). Після цього відбувається обробка статистичних результатів, накопичених в лічильниках (блок 17). Імовірність оцінюється частотою появи події, яка обчислюється як відношення кількості подій, що з'явилися, до кількості можливих появ.

Природно, треба пам'ятати, що чим більше час моделювання, тим точніше буде обчислений результат.

Необхідно уважно стежити, щоб всі шукані змінні увійшли в інтервал оголошеної точності, тільки тоді можна припинити моделювання і бути впевненим у результаті.

Для підвищення ефективності алгоритму (зменшення часу його роботи) можна відкинути нехарактерну частина реалізації – зазвичай це початковий етап роботи системи, «вихід її на режим».

Зауважимо також, що не важливо, чи маємо ми справу з однією довгою реалізацією або з великою кількістю коротких реалізацій

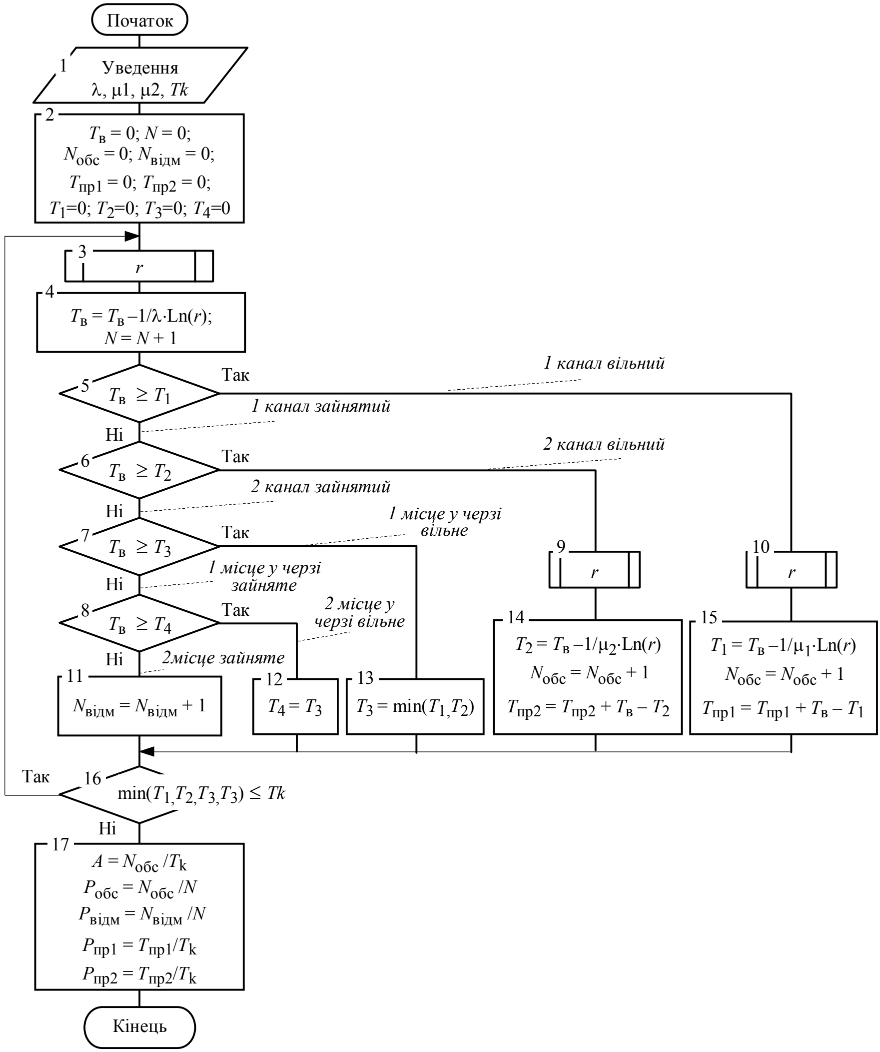


Рис. 8.23

(у яких, звичайно, вирізаний етап «вихід на режим»), в сумі дають реалізацію такої ж довжини, тобто статистичний результат буде тим же. Цей висновок встановлює рівність усереднeній за ансамблем реалізацій усередненню за часом.

Наостанок відзначимо, що на практиці зазвичай застосовують комбінації всіх трьох методів.

Контрольні запитання

1. Наведіть класифікацію імітаційних моделей.
2. Назвіть найбільш важливі класифікаційні ознаки імітаційних моделей.
3. Накресліть схему імітаційного моделювання як статистичного експерименту.
4. Як описується динаміка системи при імітаційному моделюванні?
5. Що являє собою подія при імітаційному моделюванні?
6. Що являє собою процес при імітаційному моделюванні?
7. Що являє собою транзакт при імітаційному моделюванні?
8. Як здійснюється моделювання випадкової події A, ймовірність якої дорівнює P_c?
9. Опишіть один із способів моделювання спільних подій.
10. Які методи використовуються при моделюванні безперервних випадкових величин із заданим законом розподілу?
11. Яке явище називають потоком подій?
12. Який потік подій називається ординарним?
13. Який потік подій називається регулярним?
14. Який потік подій називається потоком без післядії?
15. Який потік подій називається найпростішим?
16. Який потік подій називається нестаціонарних пуасоновським потоком?
17. Який потік подій називається потоком Пальма?
18. Які існують методи реалізації механізму модельного часу?
19. Коли доцільно використовувати метод постійного кроку?
20. Коли доцільно використовувати метод моделювання за особливими станами?

Глава 9. ДОСЛІДЖЕННЯ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ МЕТОДАМИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ

У діяльності будь-яких підприємств, в тому числі підприємств цивільної авіації, існують певні налагоджені процеси виконання ряду робіт – технологічні процеси. Технологічний процес це послідовність виконуваних операцій, необхідних для отримання необхідного результату (виконання певного виду робіт), що утворюють разом єдиний процес. Технологічні процеси добре описуються моделями систем масового обслуговування (СМО).

Система масового обслуговування – система, яка здійснює обслуговування вимог, які в неї надходять. Формальний опис процесів у вигляді СМО широко застосовується в самих різних областях науки та практики. Теоретичною базою побудови і дослідження СМО є теорія масового обслуговування як область прикладної математики.

У практичній діяльності підрозділів цивільної авіації часто створюються ситуації, коли виникає великий попит на обслуговування, а підрозділ має лише обмеженим числом обслуговуючих одиниць і не здатний задовольнити всі заявки, що надходять, на обслуговування. Заявками можуть бути замовлення на ремонт, задачі, які вирішуються в обчислювальній системі, пасажири в аеропортах, які проходять реєстрацію на рейси, літаки, що входять в зону дії диспетчера посадки та ін.

Прикладами таких систем можуть служити авіаремонтні майстерні або система резервування квитків на літаки. Іншими прикладами застосування теорії масового обслуговування можуть служити мінімізація часу посадки літаків, організації в аеропорту процесів завантаження, вивантаження і підготовки до рейсів повітряних суден, процеси обслуговування пасажиропотоків, що забезпечують найменшу чисельність обслуговуючого персоналу при заданій довжині черги. Основною особливістю подібних систем є випадковий характер не тільки часу обслуговування, але і вхідного і вихідного потоків заявок.

У п. 3.2 були розглянуті типові імітаційні моделі СМО та їх елементарні складові:

- джерела вхідних потоків заявок,
- пристрої,

- накопичувачі,
- вузли,

а також основні класифікаційні ознаки моделей СМО:

- організація потоку заявок (вимог, транзактів),
- характер утворення черги,
- обмеження черги,
- кількість обслуговуючих каналів,
- дисципліна черги і т.д.

Зазвичай кожному типу елементарної моделі СМО, відповідає певна процедура (підпрограма). Тоді повну модель СМО можна представити як алгоритм, що складається з упорядкованих звернень до цих процедур і відображає поведінку модельованої системи.

Аналіз модельованої системи на основі СМО, зазвичай, носить статистичний характер, оскільки параметри заявок, що надходять в систему, є випадковими величинами. При цьому дослідників цікавлять, перш за все, такі параметри, як продуктивність (пропускна здатність) системи, що забезпечує найменшу чисельність обслуговуючого персоналу при заданій довжині черги; тривалість обслуговування заявок в системі, наприклад, мінімізація часу посадки літаків; ефективність використовуваного в системі обладнання. Типовими вихідними параметрами в СМО вважаються числові характеристики таких величин, як час обслуговування заявок в системі, довжини черг заявок, час очікування обслуговування в чергах, завантаження пристроїв системи, а також ймовірність обслуговування в задані терміни.

Прийнято оцінювати ефективність СМО за такими показниками, як:

- ***Середня тривалість перебування заявки в СМО*** - клієнт вибирає для свого обслуговування ту систему, в якій, при однаковій якості обслуговування, він проведе менше часу;

- ***Ймовірність настання заданого події***, наприклад, ймовірність того, що заявка, що прийшла в СМО, буде обслужена;

- ***Середня кількість зайнятих каналів***, наприклад, якщо середня кількість зайнятих каналів значно менше загальної кількості каналів в системі, то це говорить про те, що деякі канали постійно простоюють, а це вже економічні втрати;

- ***Середня кількість зайнятих місць в черзі*** - при великої середній кількості зайнятих місць в черзі велика ймовірність відмови в обслуговуванні, тому виникає потреба в збільшенні кількості каналів або в збільшенні інтенсивності обслуговування заявок наявними каналами і ін.

Наведений невеликий перелік характеристик СМО вже свідчить про їх тісний взаємозв'язок і суперечливість. Поліпшення показників одних з них зазвичай тягне погіршення інших. Звідси випливає, що при проектуванні СМО бажано встановити розумну рівновагу між різними показниками ефективності системи.

Для розв‘язання цього завдання бажано вибрати ***узагальнений показник*** (C) ***ефективності*** СМО, наприклад критерій економічної ефективності, який би враховував як витрати звернення ($C_{вз}$), так і витрати споживання ($C_{вс}$):

$$C = (C_{вз} + C_{вс}) \to \min .$$

Перший доданок узагальненого показника включає витрати, наприклад, на експлуатацію системи і перерву в роботі каналів обслуговування. Другий доданок – втрати, пов'язані з відходом не обслужених заявок і з перебуванням цих заявок у черзі.

Точність чисельних оцінок ефективності СМО залежить як від форми подання взаємозв'язків між показниками ефективності СМО, так і від цілей дослідження. Теоретичні висновки і аналітичні вирази, як правило, дуже незручні для інженерних розрахунків. Тут фахівцям-практикам рекомендують застосування чисельних методів розв'язання систем рівнянь, а також поєднання формульних обчислень з чисельним аналізом.

Аналіз характеристик процесу обслуговування набагато простіше проводити методами ***імітаційного моделювання***. Тому цей метод є основним при аналізі СМО, а аналітичні дослідження використовують при попередній оцінці різних варіантів систем.

Для формулювання імітаційних моделей можна використовувати мови програмування загального застосування, а також спеціальні мови імітаційного моделювання.

9.1. Імітаційне моделювання систем масового обслуговування

Імітаційне моделювання застосовують при вивченні складних систем СМО. Результати моделювання допомагають визначити найбільш перспективні напрямки теоретичних досліджень, виявити взаємозв'язки між характеристиками системи, знайти чисельні оцінки параметрів процесів, що протікають в СМО. Основною особливістю СМО є випадковий характер не тільки часу обслуговування, але і параметрів вхідного і вихідного потоків.

Найчастіше метод імітаційного моделювання застосовується:

- для визначення оптимальних способів організації вхідних потоків заявок, для яких можна реалізувати задану якість обслуговування;
- для автоматизованого регулювання якості обслуговування при заданих обмеженнях;
- для знаходження «вузьких» місць в обслуговуючій системі з метою побудови найкращої структури СМО, яка у змозі забезпечити виконання заданої цільової функції.

9.1.1. Моделювання одноканальних систем масового обслуговування

Розглянемо імітаційну модель одноканальної СМО з обмеженим часом очікування заявками початку обслуговування. Опис принципів роботи цієї моделі базується на знаннях принципів моделювання потоків подій, викладених раніше.

Процес функціонування системи розглядається за період часу $[0, T]$ На вхід у випадкові моменти часу надходять заявки. Якщо в момент надходження чергової заявки канал вільний, то заявка приймається до негайного обслуговування. В іншому випадку заявка стає в чергу на обслуговування. Час перебування заявки в черзі обмежений. Тривалість обслуговування заявок також є випадковою величиною. Показниками ефективності СМО можна вважати величини ймовірності того, що заявка, яка прийшла в систему, буде обслужена, або продуктивність аналізованої СМО.

Суть підходу до імітації роботи СМО проста та зрозуміла – випадковий процес роботи системи на інтервалі $[0, T]$ розглядається як детермінований, але з випадково вибраними параметрами. Для забезпечення статистичної стійкості шуканих результатів процес моделювання повторюється задану кількість раз N_1. Показники ефективності формуються на основі статистичної обробки результатів окремих реалізацій процесу моделювання.

Отже, розглядається одноканальна СМО, в яку надходять заявки, що утворюють ординарний потік однорідних подій із заданим законом розподілу $f(t)$ інтервалів між моментами появи заявок. Заявки надходять у випадкові моменти часу t_i, вони обслуговуються протягом часу τ*; і обслуговування закінчується в момент t^*_i (див. рис. 9.1). Випадкова величина тривалості обслуговування τ* визначається відповідно до заданого закону розподілу $f(\tau^*)$.

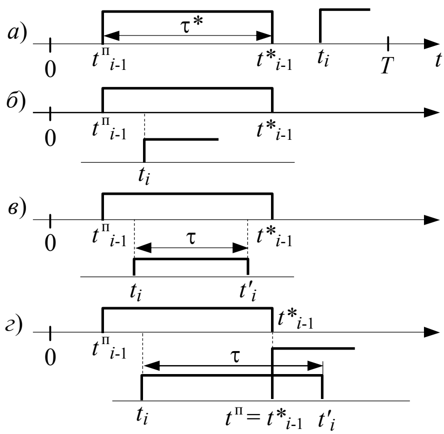


Рис. 9.1

Доля наступної заявки залежить (рис. 9.1 *а*) від моменту її появи t_i відносно моменту закінчення обслуговування попередньої заявки t^*_{i-1}. Проте, якби розглядалася СМО з відмовами і чергова заявка надійшла б раніше настання моменту закінчення обслуговування попередньої, тобто $t_i < t^*_{i-1}$, то така заявка отримала б відмову (рис. 7.6, *б*), якщо ж $t_i \geq t^*_{i-1}$, то заявка була б обслужена (див. рис. 7.6, *а*).

У зв'язку з тим, що в аналізованій СМО має враховуватися обмеження часу можливого очікування, необхідно формувати τ_i – випадкову величину тривалості очікування i-ою заявкою початку обслуговування з відповідним законом розподілу $f(\tau)$. Коли $t_i < t^*_{i-1}$, то i-я заявка отримує відмову (рис. 7.6, *в*), якщо $t'_i < t^*_{i-1}$, де $t'_i = t_i + \tau_i$ – момент закінчення очікування i-ою заявкою початку обслуговування. Якщо ж $t'_i \geq t^*_{i\text{-}1}$ (рис. 7.6, *г*), то обслуговування i- ої заявки почнеться в момент часу $t''_i = t^*_{i\text{-}1}$.

Фрагмент блок-схеми моделюючого алгоритму одноканальної СМО представлений на рис. 9.2.

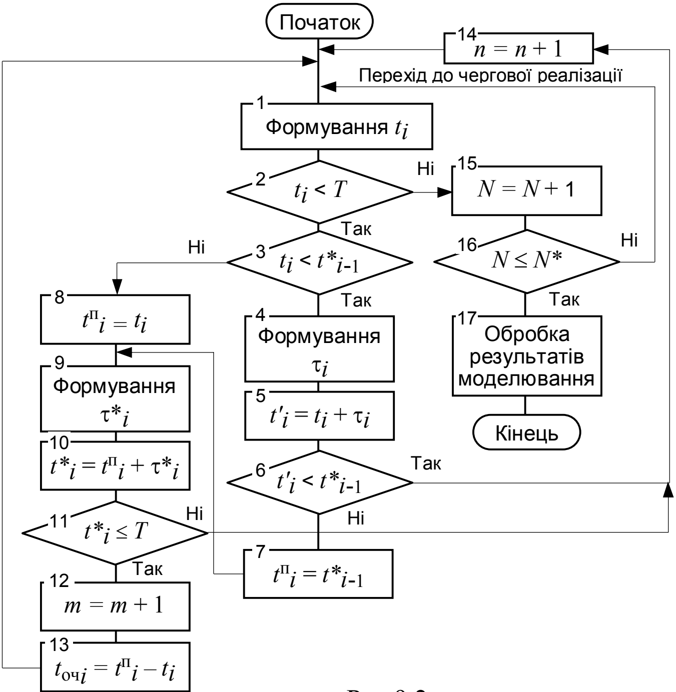


Рис.9.2

Вихідні дані для моделювання: закони розподілу $f(t)$, $f(\tau)$ и $f(\tau^*)$, величини T и N_1, а також стан системи при $t = 0$, наприклад, $t^*_{i\text{-}1} = t_i = 0$, $m = n = N = 0$. Вважається, що заявки, для яких момент появи $t_i \geq T$, в систему не потрапляють і не обслуговуються. Заявки, для яких час закінчення обслуговування $t^*_i > T$, вважаються такими, що отримали відмову, а для яких $t^*_i \leq T$, є забезпеченими.

Нижче дається опис більшості блоків, які використовуються у схемі моделюючого алгоритму:

1 – формування чергового моменту t_i надходження заявки в систему;

2 – перевірка умови $t_i < T$ приналежності моменту надходження чергової заявки t_i інтервалу дослідження системи $[0, T]$;

3 – перевірка умови $t_i < t^*_{i-1}$, де t^*_{i-1} – момент закінчення обслуговування попередньої заявки;

4 – формування випадкового значення тривалості очікування τ_i відповідно до закону розподілу $f(\tau)$;

5 – обчислення моменту $t'_i = t_i + \tau_i$, де t'_i – момент часу коли заявка покидає систему, якщо вона не буде прийнята для обслуговування;

6 – перевірка умови $t'_i < t^*_{i-1}$ (заявка залишає систему раніше, ніж звільняється канал);

7 – вибір за момент початку обслуговування i-ої заявки моменту закінчення обслуговування $(i–1)$-ої заявки $t^{н}_i = t^*_{i-1}$;

8 – вибір за момент початку обслуговування i-ої заявки моменту її надходження в систему $t^{н}_i = t_i$;

9 – формування тривалості обслуговування τ^*_i заявки (часу зайнятості каналу) відповідно до закону розподілу $f(\tau^*)$;

10 – обчислення моменту закінчення обслуговування i-ої заявки (моменту звільнення каналу) $t^*_i = t^{н}_i + \tau^*_i$,

11 – перевірка умови $t^*_i \le T$;

12 – підрахунок кількості обслугованих заявок m;

13 – обчислення $t_{очi} = t^{н}_i - t_i$ (тривалості перебування в черзі i-ої заявки);

14 – підрахунок кількості заявок n, які отримали відмову;

15 – підрахунок кількості реалізацій N;

16 – перевірка умови $N < N_1$, де N_1 – кількість реалізацій, необхідних для забезпечення заданої точності.

Можна розглянуту модель СМО перетворити в модель з відмовами, в якій заявка, що застала канал зайнятим, отримує відмову. Для цього потрібно в моделі (див. рис. 9.2) ліквідувати блоки 4 ... 7 і з блоку 3 за ознакою «Так» переходити в блок 14.

Можна також перетворити описану модель СМО в модель з необмеженим очікуванням заявкою початку обслуговування. Для цього потрібно в моделі ліквідувати блоки 4 ... 6 та за ознакою «Так» з блоку 3 переходити в блок 7.

9.1.2. Моделювання багатоканальних систем масового обслуговування

Нехай багатоканальна СМО має k каналів, цілком ідентичних (з точки зору тривалості обслуговування) і рівноправних (в сенсі можливості використання для обслуговування даної заявки будь-якого вільного каналу). У систему як і у попередньому прикладі надходить ординарний потік заявок із заданим законом розподілу $f(t)$. Час можливого очікування до початку обслуговування τ – випадкова величина з законом розподілу $f(\tau)$. Тривалість обслуговування τ^* має закон розподілу $f(\tau^*)$. Процес функціонування системи розглядається в інтервалі часу $[0, T]$. Блок-схема алгоритму багатоканальної СМО представлена на рис. 9.3.

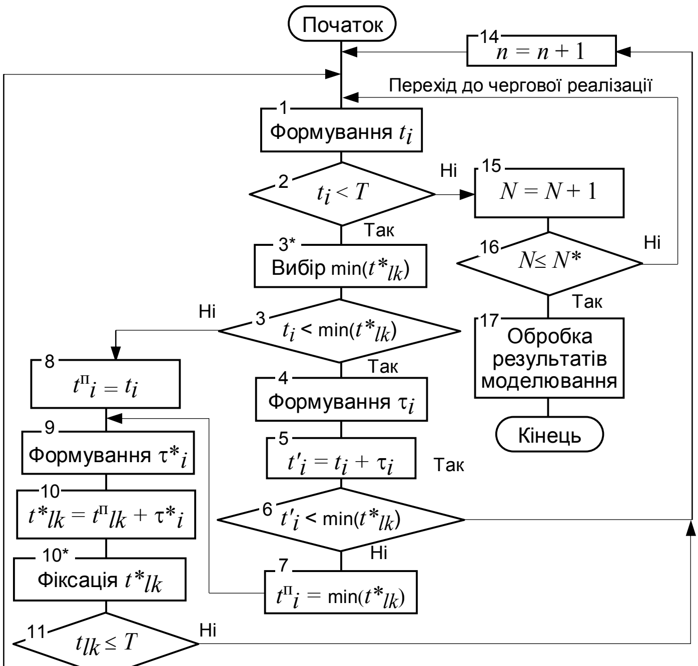


Рис. 9.3

Заявки приймаються до обслуговування в порядку черги (у порядку надходження в систему). Якщо для обслуговування даної заявки є кілька вільних каналів, то канали залучаються до обслуговування у порядку черги (в першу чергу залучається той канал, який раніш інших звільнився від обслуговування).

Описана тут система близька за характером до одноканальної СМО. Тому багато блоків попереднього прикладу можуть бути використані для побудови і цього алгоритму. Тут також зберігаються всі використані вище позначення.

Проте у блок-схемі алгоритму багатоканальної СМО (рис. 9.3) додатково використані такі блоки:

3 - вибір мінімального t^*_{lk};

3* - перевірка умови $t_i < \min(t^*_{lk})$;

6 - перевірка $t'_i < \min(t^*_{lk})$;

7 - вибір за момент початку обслуговування $t^{п}_i$, величини $\min(t^*_{lk})$;

10, 10* – виявлення та фіксація t^*_{lk} – моменту закінчення обслуговування l-ої заявки k-м каналом (моменту звільненням k-го каналу – $k = \overline{1,K}$): $t^*_{lk} = t^{п}_{lk} + \tau^*_i$, де $t^{п}_{lk}$ - момент початку обслуговування l-ої заявки k-м каналом.

Програми імітації СМО можуть реалізовувати покроковий або подієвий метод моделювання. При програмуванні застосовуються як спеціальні мови імітаційного моделювання, так і звичайні мови високого рівня (Фортран, Паскаль, Делфи, С ++, тощо.

У програмах зазвичай створюються процедури и функції, і тоді програму можна уявити собі як алгоритм, що складається з упорядкованих звернень до цих процедур і відображає поведінку СМО.

Наприклад, необхідно створити програму, яка імітує роботу касового залу аеропорту як СМО з продажу авіаквитків. За уведеними параметрами роботи системи: середньому часу появи пасажира, часу його обслуговування в касі, кількості кас, кількості пасажирів дозволяє адміністрації аеровокзалу провести аналіз кількості необхідних кас з продажу авіаквитків (апаратів обслуговування) та їх завантаженості, стану модельованої СМО в певний момент часу і ефективності її роботи. Серед пасажирів можуть бути пільговики, які обслуговуються позачергово. Час появи і час обслуговування пільговиків також може бути заданий користувачем. Черга до кожної каси своя. Вибір пасажиром каси відбувається за принципом: там де менше народу.

У прикладі програми, написаною на мові Делфи використовуються глобальні змінні `Time_ Pilgov` та `Time_Pasag`, які описують час, пройдений з моменту появи пасажира. На кожному кроці чергової реалізації відбувається інкрементація цих змінних. Якщо `Time_Pilgov` стає рівним `Random_Pilgov`, значить, настав час появи пільговика, я якщо `Time_Pasag` дорівнює `Random_Pasag` – час появи звичайного пасажира. Для додавання пасажира використовуються процедури `Add_Pilg` та `Add_Pas`, які додають пасажира у саму коротку чергу, в її початок або у кінець, залежно від статусу пасажира. Одночасно відбувається переприсвоювання змінних `Random_Pilgov` і `Random_Pasag` , та обнуління `Time_Pilgov` і `Time_Pasag`.

На кожному кроці також виконується циклічний обіг всіх апаратів обслуговування (у нашому випадку кас з продажу авіаквитків), при цьому виконуються наступні дії:

Якщо каса працює, то збільшується час її роботи, користувачу може надаватися поточна інформація про кількість пасажирів, що заходяться у черзі, і про то що канал занятий. Збільшується час перебування у черзі кожного пасажира, який стоїть у цієї черзі.

Якщо час обслуговування пасажира в цієї касі закінчився (`T_Obsl=Random_Obsl`), то збільшується кількість обслуговуваних апаратом (касою) заявок (проданих квитків), збільшується загальна кількість пасажирів, що придбали квитки, скорочується черга біля цієї каси.

Якщо черга нерівна нулю, то обслуговування отримує пасажир, який перебуває у черзі першим.

Якщо черга дорівнює нулю, то, то обслуговуючий апарат (каса) переходить у стан "вільний" (`Status=False`). Якщо каса вільна, то збільшується час її простою і користувачу, якщо це передбачено програмою, може надаватися поточна інформація про то що канал вільний. Оновлюються данні у файлах статистики.

Якщо кількість обслуговуваних пасажирів досягла необхідного або користувач зупинив програму, натиснувши кнопку `STOP`, то робота таймера зупиняється і запускається процедура формування звіту про експеримент. Звіт відображається на екрані користувача, а також додається у файл статистики.

Фрагменти програми імітації такої СМО з продажу авіаквитків на мові Делфи наведені нижче.

Наприклад, функцією `RandomTime (Min: Integer; Max: Integer): Integer`, що формує випадкову величину в заданому проміжку, може задаватися час обслуговування пасажира – `Random_Obsl` або час, через який, наприклад, з‘явиться звичайний пасажир і пільговик: `Random_Pasag` і `Random_Pilgov`. За заданими значеннями `Max` та `Min` функція `RandomTime` повертає випадкове число в заданому проміжку.

```
*****************************************************
Function TForm1. RandomTime;
   Begin
     Result: =Random (Max-Min+1) +Min;
   End;
*****************************************************
```

Програма імітації СМО оперує з двома класами `TKaca` и `TPasagir`. `TKaca` являє собою клас, що містить у собі поля, які характеризують апарати обслуговування (каси).

```
*****************************************************
TKaca = object // Апарат обслуговування
Status: Boolean; // Статус. Занятий або вільний
T_Obsl: Integer;
R_Obsl: Integer;
Zayavki: Integer; // кількість заявок на касу
Rabota: Integer; // Час роботи
Prostoy: Integer; // Час простою
Queue_Len: Integer; // Довжина черги
Queue: ARandomTime [1. .1000] of TPasag;
{Масив пасажирів}
  End;
*****************************************************
```

`Status` - логічна змінна, яка вказує занята каса або ні. (`True` – занята, `False` – вільна). `T_Obsl` – поле, що вказує час з початку обслуговування касою пасажира. `R_Obsl` – час потрібний для обслуговування певного пасажира. R_Obsl задається випадково, у проміжку, заданому користувачем. Поле Zayavki надає числове значення кількості заявок (пасажирів), обслугованих цією касою. Поля Rabota та Prostoy вказують час, який каса працювала і який простояла. Поле Queue_Len відповідає кількості пасажирів, що перебувають у черзі. Поле Queue є масив типу TPasag, в якому зберігається інформація про пасажирів, що перебувають у черзі.

Клас TPasagir містить тільки два поля: Status – звичайний пасажир або пільговик, (0 – звичайний пасажир, 1 – пільговик) та T_in_Queue – час, який пасажир знаходився у черзі.

```
*************************************************
TPasagir = object // пасажирів у черзі
Status: Byte; // статус (може бути пільговиком)
T_in_Queue: Integer; // Час проведений у черзі
End;
*************************************************
```

Як приклад нижче наведені процедури Add_Pilg та Add_Pas – додавання пасажира, який з‘явився у касовому залі, у саму коротку чергу, в її початок або у кінець, залежно від статусу пасажира.

Час появи пасажира у касовому залі це інтервал часу, у проміжку якого існує імовірність появи пасажира. Час появи формується функцією RandomTime. Середнье значення появи пасажира буде прагнути до середнього значення заданого інтервалу.

```
*************************************************
Procedure TForm1.Add_Pilg;
Var
i, Min,Min_i: Integer;
Begin      //Пошук каси з мінімальною довжиною черги
Min: =Kaca[1]. Queue_Len;
Min_i: =1;
For i: =2 to Kaca_Count do
Begin
if Kaca[i]. Queue_Len<Min Then
Begin
Min: = Kaca[i]. Queue_Len;
Min_i: =i;
End;
End;
Inc (Kaca[Min_i]. Queue_Len);
```

```
If Kaca[Min_i]. Queue_Len>Max_Queue Then Max_Queue: =
Kaca[Min_i]. Queue_Len;
For i: = Kaca[Min_i]. Queue_Len DownTo 2 do
Begin
Kaca[Min_i]. Queue [i]: = Kaca[Min_i]. Queue [i-1];
End;
Kaca[Min_i]. Queue [1]. Status: =1; // 1-ий у черзі
Kaca[Min_i].  Queue [1]. T_in_Queue: =0;
if Kaca[Min_i]. Status=False Then
Kaca[Min_i].R_Obsl: =RandomTime(ServeMin,ServeMax);
Kaca[Min_i]. Status: =True;
Inc (Pilg_Count);
End;
***********************************************
Procedure TForm1. Add_Pas;
Var
i, Min,Min_i: Integer;
Begin     //Пошук каси з мінімальною довжиною черги
Min: = Kaca[1]. Queue_Len;
Min_i: =1;
For i: =2 To Kaca_Count Do
Begin
if Kaca[i]. Queue_Len<Min Then
Begin
Min: = Kaca[i]. Queue_Len;
Min_i: =i;
End;
End;
Inc (Kaca[Min_i]. Queue_Len);
If Kaca[Min_i]. Queue_Len>Max_Queue Then Max_Queue: =
Kaca [Min_i]. Queue_Len;
Kaca[Min_i].Queue [Kaca[Min_i].Queue_Len].Status: =0;
{Останній у черзі}
Kaca[Min_i]. Queue [Kaca[Min_i]. Queue_Len].
T_in_Queue: =0;
if Kaca [Min_i]. Status=False Then
Kaca[Min_i]. R_Obsl: =TimeRandom (ServeMin,ServeMax);
Kaca [Min_i]. Status: =True;
End;
***********************************************
```

В останній час окрім мов високого рівня при програмуванні СМО застосовуються спеціальні мови імітаційного моделювання.

9.1.3. Моделювання систем масового обслуговування з використанням мов імітаційного моделювання

Серед мов імітаційного моделювання розрізняють мови, орієнтовані на опис подій, засобів обслуговування або маршрутів руху заявок (процесів). Вибір мови моделювання визначає структуру моделі та методику її побудови.

Орієнтація на пристрої характерна для функціонально-логічного та більш детальних ієрархічних рівнів опису об'єктів.

Для опису імітаційних моделей на системному рівні (такі моделі іноді називають мережевими імітаційними моделями) частіше за все використовують мови, орієнтовані на події або процеси. Прикладами перших можуть служити мови Сімскріпт, SMPL і ряд інших. До числа других відносяться мови Симула, SOL. Найбільш популярною мовою мережевих імітаційних моделей є мова GPSS (General Purpose Simulation System).

Мови імітаційного моделювання реалізуються в програмно-методичних комплексах моделювання СМО, які мають ту або іншу ступінь спеціалізації. Так, комплекси на базі мови GPSS можна використовувати в багатьох додатках, але є спеціалізовані комплекси для моделювання обчислювальних мереж, систем управління підприємствами, тощо.

При використанні мов, орієнтованих на процеси, у складі системи імітаційного моделювання (СІМ) виділяються елементарні частини (моделі), що були розглянуті у п. 3.2.2. Ними можуть бути, джерела вхідних потоків заявок, пристрої, накопичувачі та вузли.

Зазвичай кожному типу елементарної моделі, за винятком лише деяких вузлів, в програмній системі відповідає певна процедура (підпрограма). І тоді СІМ можна уявити як алгоритм, що складається з упорядкованих звернень до цих процедур і відображає поведінку модельованої системи.

У процесі моделювання відбуваються зміни модельного часу, який найчастіше приймається дискретним і вимірюється в тактах. Час змінюється після того, як закінчена імітація чергової групи подій, що відносяться до поточного моменту часу t_k. Імітація супроводжується накопиченням в окремому файлі статистики таких даних, як кількість заявок, які вийшли із системи обслуженими і не обслуженими, сумарний час зайнятого стану для кожного з пристроїв, середні довжини черг і т. п. Імітація закінчується, коли поточний час перевищить заданий відрізок часу або коли вхідні джерела сформують задане число заявок. Після цього виконується обробка накопичених у файлах статистики даних, що дозволяє отримати значення необхідних вихідних параметрів.

У програмах імітаційного моделювання СМО переважно реалізується ***подієвий метод*** організації обчислень. Сутність подієвого методу полягає у відстеженні на моделі послідовності подій у тому ж порядку, в якому вони відбувалися б у реальній системі. Обчислення виконують тільки для тих моментів часу і для тих частин (процедур) моделі, до яких відносяться здійснювані події.

Іншими словами, звернення на черговому такті модельованого часу здійснюються тільки до моделей тих обслуговуючих апаратів (ОА) (пристроїв, накопичувачів, вузлів, тощо), на входах яких в цьому такті відбулися зміни. Оскільки зміни станів в кожному такті зазвичай спостерігаються лише у малої частки ОА, то подієвий метод може істотно прискорити моделювання у порівнянні з методами, в яких на кожному такті аналізуються стани всіх елементів моделі

Розглянемо можливу схему реалізації подієвого методу імітаційного моделювання.

Моделювання починається з перегляду операторів генерування заявок, тобто зі звернення до моделей джерел вхідних потоків. Для кожного незалежного джерела таке звернення дозволяє розрахувати момент генерації першої заявки. Цей момент разом з ім'ям (посиланням на заявку) заноситься в список майбутніх подій (СМП), а відомості про генеровану заявці – в список заявок (СЗ). Запис в СЗ включає в себе ім'я заявки, значення її параметрів (атрибутів), місце, яке займає заявка у даний момент в СІМ. У СМП події упорядковуються за збільшенням моментів їх здійснення.

Потім з СМП вибирають сукупність відомостей про події, що відносяться до найбільш раннього моменту часу. Ця сукупність переноситься в список поточних подій (СПП), з якого беруться посилання на події. Звернення за посиланням до СЗ дозволяє встановити місце в СІМ заявки *А,* з якою пов'язано модельована подія.

Нехай цим місцем є пристрій *X*. Тоді програма моделювання виконує наступні дії (рис. 9.4):

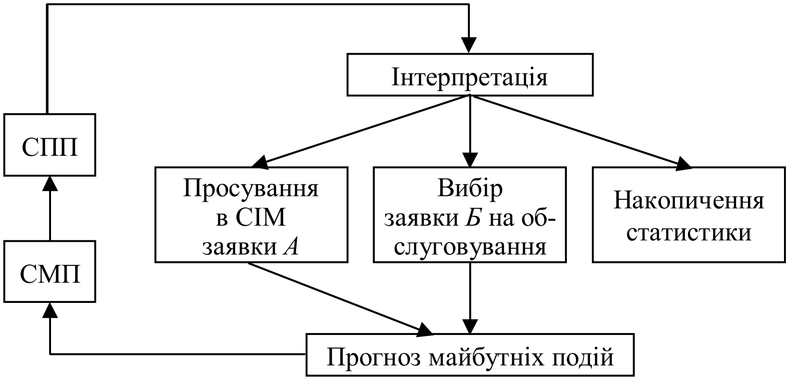


Рис. 9.4

– Змінює параметри стану пристрою *X*. Наприклад, якщо заявка *А* звільняє *X*, а черга до *X* не була порожньою, то відповідно до заданої дисципліни обслуговування з черги до *X* вибирається заявка *B* і надходить на обслуговування в *X*;

– Прогнозується час настання наступної події, пов'язаної із заявкою *B*, шляхом звернення до моделі пристрою *X*, в якому розраховується тривалість обслуговування заявки *B*. Відомості про цю майбутню подію заносяться в СМП і СЗ;

– Відбувається імітація руху заявки *А* в СІМ за маршрутом, який визначається заданою програмою моделювання, до тих пір, поки заявка не прийде на вхід деякого ОА. Тут заявка або затримується в черзі, або шляхом звернення до моделі цього ОА прогнозується настання майбутньої події, пов'язаної з подальшою долею заявки *А*. Відомості про майбутню подію також заносяться в СМП і СЗ;

– В файл статистики додаються необхідні дані.

Після відпрацювання всіх подій, які відносяться до моменту часу t_k, відбувається збільшення модельного часу до значення, відповідного найближчої майбутньої події, і розглянутий процес імітації повторюється.

Найчастіше програми орієнтовані на опис процесів в програмно-методичних комплексах моделювання СМО пишуться на мові імітаційного моделювання GPSS. Програма на мові GPSS являє собою послідовність операторів (їх називають блоками), які відображають події, що відбуваються в СМО при переміщеннях транзактів. Оскільки в інтерпретаторах GPSS реалізується подієвий метод і в СМО може бути одночасно багато транзактів, то інтерпретатор буде по черзі виконувати різні фрагменти програми, імітуючи просування транзактів в поточний момент часу до їх затримки в деяких пристроях або чергах.

Оператори[9.1] GPSS мають такий формат:

```
<мітка> <ім'я_оператора> <поле_операндів> [<коментар>]
```

Мітка може займати позиції, починаючи з 2-ої, ім'я оператора – с 8-ої, поле операндів – с 19-ої, а коментар обов'язково відділяється від поля операндів пробілом.

Поле операндів може бути пустим або мати один або більше операндів, які позначуються символами `A`, `B`, `C`, ..., що ідентифікують застосовані у програмі блоки. Операндами можуть бути ідентифікатори пристроїв, накопичувачів, службові слова, а також ***стандартні числові атрибути*** (СЧА), до яких відносяться величини, які часто зустрічаються у різних задачах. Як приклад СЧА можна навести такі операнди:

- `AC1` – поточний час,
- `FN` – функція,
- `RN` – випадкова величина,
- `RN1` – випадкова величина, яка рівномірно розподілена у діапазоні [0, 1],
- `P` – параметр транзакта (кожен транзакт може мати не більш L параметрів, де L залежить від інтерпретатора),
- `V` – цілочисленна змінна (дійсна та булева змінні позначаються FV і BV відповідно),
- `X` – збережена змінна (змінна, для якої автоматично підраховується статистика),
- `K` – константа,
- `S` – об'єм зайнятої пам'яті в накопичувачі,

[9.1] Оператор – це деяка дія, яка виконується з операндом (спеціальна величина, яка обробляється у програмі).

- F – стан пристрою,
- Q – поточна довжина черги, тощо.

Посилання на ідентифікатори записуються у вигляді:

<СЧА>$<ідентифікатор>,

наприклад, Q$ORD означає чергу ORD, а FN$COS – посилання на функцію COS.

Наведемо приклади операторів мови GPSS, які найчастіше зустрічаються у програмно-методичних комплексах моделювання СМО.

Так джерела заявок зазвичай описуються блоком

```
GENERATE   A,B,C,D,E
```

Тут A і B слугують для завдання інтервалів між появою заявок, при цьому можна використовувати один з наступних варіантів:

- інтервал це рівномірно розподілена у діапазоні [A–B, A+B] випадкова величина;
- інтервал це значення функції, яке вказано в B і помножено на A;

C – затримка у виробленні першого транзакта; D - число заявок, що виробляє джерело; E - пріоритет заявок. Якщо D пусто, то число генерованих транзактів необмежене. Наприклад:

```
GENERATE   6,FN$EXP,,15
```

Цей оператор описує джерело, яке виробляє 15 транзактів з інтервалами, що дорівнюють добутку числа 6 і значення функції EXP.

Другий приклад:

```
GENERATE   36,12
```

У цьому прикладі кількість транзактів необмежена, а інтервали між транзактами – випадкові числа у діапазоні [24, 48].

Функції, на які є посилання в операторах, мають бути описані за допомогою блоку такого типу:

```
M      FUNCTION   A,B
```

За ним йде рядок, що починається з першої позиції :

```
X1,Y1/X2,Y2/X3,/.../Xn,Yn
```

Тут мітка `M` – ідентифікатор функції, `A` – аргумент функції, `B` – тип функції, `Xi` та `Yi` – координати вузлових точок функції, заданої табличне. Наприклад:

```
EXP   FUNCTION   RN1,C12
0,0/.2,.22/.4,.51/.5,.6/.6,.92/.7,1.2/.8,
 1.61/.9,2.3/.95,3/.99,4.6/.999,6.9/1,1000
```

Це опис безперервної (**C**) функції EXP, заданої табличне 12-ма вузловими точками, аргументом якої є випадкова величина (RN1), що рівномірно розподілена у діапазоні [0, 1].

Оператори заняття транзактом і звільнення від обслуговування пристрою `A` формулюються наступним чином:

```
SEIZE      A
RELEASE    A
```

Затримка у русі транзакта по СМО описується оператором:

```
ADVANCE    A,B
```

`A` та `B` мають той же зміст, що й в операторі `GENERATE`.

Наприклад, обслуговування транзакта у пристрої WST тривалістю *a* одиниць часу, де *a* - рівномірно розподілена у діапазоні [7,11] випадкова величина, описується таким фрагментом програми:

```
SEIZE      WST
ADVANCE     9,2
RELEASE    WST
```

Аналогічно описується заняття транзактом пам'яті у накопичувачі:

```
ENTER      A,B
```

Тут крім ім'я накопичувача (`A`) вказується об'єм займаної пам'яті (`B`).

Звільнення `B` вічок пам'яті у накопичувачі `A` виконується оператором:

```
LEAVE      A,B
```

Для накопичувачів у моделі необхідно задавати загальний об'єм пам'яті, що робиться наступним чином:

```
M       STORAGE     A
```

Тут: `M` – ім'я накопичувача, `A` – об'єм його пам'яті.

Якщо транзакт приходить на вхід зайнятого пристрої або на вхід накопичувача з недостатнім об'єм вільної пам'яті, то транзакт затримується у черзі до цього пристрою або накопичувача. Стеження за станом пристроїв та черг виконує інтерпретатор. Але якщо у моделі потрібно посилатися на довжину черги або збирати статистику про її довжину, то необхідно явне зазначення цієї черги у моделі. Робиться це за допомогою операторів входу в чергу и виходу з черги:

```
QUEUE       A,B
DEPART      A,B
```

Згідно з цим оператором черга `A` збільшується і зменшується на `B` одиниць відповідно, якщо `B` = 1, то поле `B` можна залишити пустим.

Рух транзактів виконується за маршрутом, заданим послідовністю операторів у моделі. Якщо потрібно змінити природний порядок, то застосовують оператор переходу. Оператор умовного переходу має вигляд:

```
TEST XX    A,B,C
```

Відповідно до наведеного оператору умовного переходу перехід до оператору, поміченого міткою `C`, відбувається, якщо не виконується умова `A XX B`, де `XX` $\in$ `{E, NE, L, LE, G, GE}`; `E` – дорівнює; `NE` – не дорівнює; `L` – менше; `LE` - менше або дорівнює; `G` - більше; `GE` - більше або дорівнює (`XX` завжди розміщується в позиціях 13 та 14).

9.1.4. Моделювання систем масового обслуговування у термінах мереж Петрі

Аналіз складних систем можна здійснювати шляхом імітаційного моделювання СМО, які подаються моделями мереж Петрі. При цьому задають вхідні потоки заявок и визначають відповідну реакцію системи. Вихідні параметри СМО розраховують шляхом обробки накопиченого при моделюванні статистичного матеріалу.

<u>Приклад 1</u>

Потрібно описати за допомогою мережі Петрі роботу групи користувачів на єдиної робочої станції WS при заданих характеристиках потоків запитів на користування WS та з урахуванням характеристики задачі, яка надходить разом з певним запитом. Мережа Петрі для цього прикладу показана на рис.9.5.

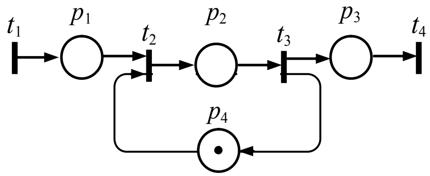


Рис. 9.5.

Тут переходи зв‘язані з наступними подіями: t_1 – надходження запиту на користування WS, t_2 – зайняття станції, t_3 – звільнення станції, t_4 - вихід обслуговуваної заявки; позиція p_4 використовується для відображення сану WS: якщо у p_4 є мітка, то WS вільна и заявка, що надійшла, збуджує спрацювання переходу t_2; доти ця заявка не буде обслугована, мітки у p_4 не буде, отже, и запити, що надійшли у позицію p_1 змушені очікувати спрацювання переходу t_3.

<u>Приклад 2</u>

Потрібно описати за допомогою мережі Петрі процеси виникнення та усунення несправностей в деякої технічної системі, яка складається з множини однотипних блоків; у запасе є один справний блок; відомі статистичні дані про інтенсивності виникнення відмов і тривалості таких операцій, як пошук несправностей, заміна та ремонт блоку, що відмовив. Пошук і заміну блоку, що відмовив, виконує одна бригада, а ремонт заміненого блоку – друга бригада. Мережа Петрі для цього прикладу показана на рис.9.6. Відмітимо, що при числі міток в позиції, рівному M, можна в ній не ставити M точок, а записати в позиції значення M.

У наведеному прикладі значення M в позиції p_2 відповідає числу наявних у системі блоків. Переходи відображають наступні події: t_1 – відмова блоку, t_2 – пошук несправного блоку, t_3 – заміна несправного блоку, t_4 – закінчення ремонту.

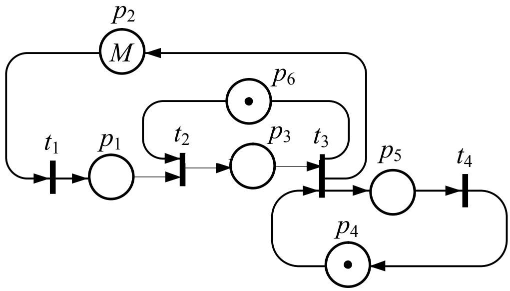


Рис. 9.6.

Очевидно, що при непустої позиції p_2 перехід t_1 спрацьовує, але із затримкою, що дорівнює обчисленому випадковому значенню модельованого відрізку часу між відмовами. Після виходу маркера з t_1 він потрапляє через p_1 у t_2, якщо є мітка у позиції p_6, це означає, що бригада фахівців, що обслуговує систему вільна и може починати пошук виниклої несправності. У переході t_2 мітка затримується на час, що дорівнює випадковому значенню тривалості пошуку несправності. Далі маркер опиняється у p_3 і, якщо є запасний блок (маркер у p_4), то запускається перехід t_3 , з котрого маркери вийдуть у p_2, p_5 та у p_6 через відрізок часу, що потрібний для заміни блоку. Після цього у t_4 імітується відновлення несправного блоку.

Ця модель описує функціонування системи в умовах, коли відмови можуть виникати і в робочому, і в несправному станах системи. Тому не виключені ситуації, при яких більш ніж один маркер опиниться у позиції p_1.

У багатьох задачах динамічні об’єкти можуть бути різних типів, и для кожного типу необхідно уводити свої алгоритми поведінки у мережі. У цьому випадку кожний маркер має мати хоч би один параметр, що позначує тип маркеру. Такий параметр зазвичай називають кольором. Колір можна застосовувати як аргумент у функціональних мережах. Таку мережу називають *кольоровою мережею Петрі.*

Серед інших різновидів мереж Петрі можна згадати *інгиbиторні мережі Петрі*, в яких можливо існування так званих заборонних (інгибиторних) дуг. Наявність маркеру у вхідній позиції, яка зв’язана з переходом інгибиторною дугою, означає заборону спрацювання переходу.

Контрольні запитання

1. Дайте визначення поняття технологічний процес.
2. Дайте визначення поняття СМО.
3. У чому основна особливістю СМО?
4. Наведіть приклади елементарних складових типових імітаційних моделей СМО.
5. Наведіть приклади основних класифікаційних ознак моделей СМО.
6. Які параметри модельованої СМО перш за все цікавлять дослідників при проведенні її аналізу?
7. Числові характеристики яких величин є типовими вихідними параметрами СМО?
8. За якими показниками прийнято оцінювати ефективність СМО?
9. Для розв‘язання яких завдань найчастіше застосовується метод імітаційного моделювання СМЩ?
10. Які параметри СМО зазвичай мають випадковий характер?
11. У чому суть підходу до імітації роботи СМО?
12. На основі чого формуються показники ефективності СМО?
13. Які вихідні дані задаються при моделюванні одноканальної СМО?
14. Як забезпечується статистична стійкість результатів моделювання СМО?
15. Які дії виконуються на кожному кроці наведеного прикладу СМО з продажу авіаквитків при циклічному обігу всіх апаратів обслуговування?
16. Які поля містить у собі об‘єкт «**`TKaca`**», який характеризує апарати обслуговування (каси) у наведеному прикладу СМО з продажу авіаквитків?
17. Наведіть приклади мов імітаційного моделювання?
18. Яким можна уявити собі алгоритм СІМ, якщо кожному типу елементарної моделі відповідає певна процедура?
19. Яку мережу називають кольоровою мережею Петрі?

Глава 10. ПЛАНУВАННЯ ІМІТАЦІЙНИХ ЕКСПЕРИМЕНТІВ

Загальні задачі планування модельних експериментів вже розглядалося в п.5.4, де підкреслювалося, що планування експериментів особливо актуально при проведенні імітаційного моделювання, як одного з видів статистичного моделювання.

Існує два основних варіанта постановки задачі планування імітаційного експерименту:

- зі всіх допустимих потрібно обрати такий план, який дозволив би отримати найбільш достовірне значення функції відгуку f (x) при фіксованої кількості дослідів;
- зі всіх допустимих потрібно обрати такий план, при якому статистична оцінка функції відгуку може бути отримана з достатньою точністю при мінімальному обсязі випробувань.

Розв‘язання задачі планування у першій постановці є стратегічним плануванням імітаційного експерименту, а у другій – тактичним плануванням.

10.1. Стратегічне планування імітаційного експерименту

При стратегічному плануванні виникають такі проблеми:

- встановлення стохастичної збіжності результатів, яка визначає тривалість окремих етапів експерименту і необхідність пошуку методів його скорочення;
- пошук способів аналізу систем при значному числі варійованих вхідних змінних;
- визначення багатокомпонентності функції реалізації, виникаючих при спостереженні декількох різних вихідних змінних;
- вибір найкращих способів пошуку екстремумів шуканої функції;
- вимірювання й обробка функцій реакції моделі.

В цілому, ціль стратегічного планування імітаційних експериментів це отримання максимального об‘єму інформації про досліджувану систему в кожному експерименті (спостереженні). Іншими словами, стратегічне планування дозволяє дати відповідь на таке запитання: при якому сполученні рівнів зовнішніх і внутрішніх факторів може бути отримана найбільш повна та достовірна інформація про поведінку системи.

Тому при стратегічному плануванні необхідно:

1. Провести ідентифікацію усіх факторів, що впливають на експеримент;
2. Здійснити вибір рівнів факторів.

Під *ідентифікацією факторів* розуміють їх ранжирування за ступенем впливу на значення спостережуваної змінної (показника ефективності).

За результатами ідентифікації доцільно розділити всі фактори на дві групи первинні та вторинні. *Первинні* – це фактори, в дослідженні впливу яких експериментатор заінтересований безпосередньо. *Вторинні* – це фактори, які не є предметом дослідження, проте впливом яких нехтувати неможна.

Вибір рівнів факторів здійснюється з врахуванням двох суперечливих вимог:

- рівні фактору повинні перекривати (заповнювати) увесь можливий діапазон його змін;
- загальна кількість рівнів за всіма факторами не має приводити до надмірного обсягу моделювання.

Відшукання компромісного рішення, що задовольняє цим вимогам, і є задачею стратегічного планування експерименту.

Способи побудови стратегічного плану

Експеримент, в якому реалізуються усі можливі сполучення рівнів факторів, називаються ***повним факторним експериментом*** (ПФЕ).

Загальне число різних комбінацій рівнів в ПФЕ для k факторів можна обчислити як:

$$N = l_1 . l_2 . l_3 \dots l_k,$$

тут l_i – число рівнів i-го фактора.

Якщо число рівнів для всіх факторів однаково, то $N = L^k$ (L – число рівнів).

Недолік ПФЕ – це великі часові витрати на підготовку та проведення експериментів.

Наприклад, якщо в моделі відображені 3 фактори, що впливають на значення вибраного показника ефективності, кожен з котрих має 4 можливих рівня (значення), то план проведення ПФЕ включатиме 64 експериментів ($N = 43$). Якщо при цьому кожний з них триває хоча б одну хвилину (з урахуванням часу на зміну значень факторів), то на одноразову реалізацію ПФЕ потрібно більше години.

Тому використання ПФЕ доцільно тільки у тому випадку, якщо у ході імітаційного експерименту досліджується взаємний вплив усіх факторів, що фігурують в моделі.

Якщо такі взаємодії вважають відсутніми або їх ефектом нехтують, проводять частковий факторний експеримент (ЧФЕ).

Відомі і застосовуються на практиці різні варіанти побудови планів ЧФЕ. Ми розглянемо тільки деякі з них.

Рандомізований план передбачає вибір поєднання рівнів для кожного прогону випадковим чином. При використанні цього методу відправною точкою у формуванні плану є число експериментів, які вважає за можливим (або необхідним) провести дослідник.

Латинський план (або «латинський квадрат») використовується в тому випадку, коли проводиться експеримент з одним первинним фактором і декількома вторинними. Суть такого планування така. Якщо первинний фактор A має l рівнів, то для кожного вторинного фактора також вибирається l рівнів. Вибір комбінації рівнів факторів виконується на основі спеціальної процедури.

Наприклад, в експерименті використовується первинний фактор A і два вторинних фактора B і C, число рівнів факторів l дорівнює 4. Відповідний план можна представити у вигляді квадратної матриці розміром $l \times l$ (4×4) відносно рівнів фактора A. При цьому матриця будується таким чином, щоб в кожному рядку і в кожному стовпці даний рівень фактора A зустрічався тільки один раз:

Значення фактора B	Значення фактора C			
	C_1	C_2	C_3	C_4
B_1	A_1	A_2	A_3	A_4
B_2	A_2	A_3	A_4	A_1
B_3	A_3	A_4	A_1	A_2
B_4	A_4	A_1	A_2	A_3

В результаті маємо план, що вимагає $4\times4 = 16$ прогонів, на відміну від ПФЕ, для якого потрібно $4^3 = 64$ прогонів.

Експеримент зі зміною факторів по одному. Суть його полягає в тому, що один з факторів «пробігає» все l рівнів, а інші n–1 факторів підтримуються постійними. Такий план забезпечує дослідження ефектів кожного фактора окремо. Він вимагає всього $N = l_1 + l_2 + ... + l_n$ прогонів (l_i число рівнів i-го фактору).

Для розглянутого вище прикладу (3 фактори, які мають по 4 рівня) N = 4 + 4 + 4 = 12. Ще раз підкреслимо, що такий план застосовується (як і будь-який ЧФЕ) тільки при відсутності взаємодії між факторами.

Дробовий факторний експеримент. Кожен фактор має два рівні – нижній і верхній, тому загальне число варіантів експерименту $N = 2k$, де k – кількість факторів. Матриці планів для значень $k = 2$ і $k = 3$ наведені нижче

Матриця планів для $k = 2$

Номер експерименту	Значення факторів	
	x_1	x_2
1	0	0
2	0	1
3	1	0
4	1	1

Матриця планів для $k = 3$

Номер експерименту	Значення факторів		
	x_1	x_2	x_3
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

Плани, що побудовані за таким принципом, у ряді випадків мають певні властивості (симетричності, нормованості, ортогональності і ротабельності), які забезпечують підвищення якості проведених експериментів.

- Симетричні плани відрізняються більшою впорядкованістю в розташуванні точок, зокрема алгебраїчна сума елементів вектор-стовпця для кожного фактора, дорівнює нулю.
- Нормовані плани це плани, для яких існує умова нормування, а саме – сума квадратів елементів кожного стовпчика повинна дорівнювати числу дослідів.

- Ортогональні плани володіють такою властивістю̃ сума почлених добутків будь-яких двох вектор-стовпців матриці дорівнює нулю.
- Рототабельні плани характеризуються тим, що в будь-якій точці простору факторів стандартне відхилення побудованої реакції залежить тільки від відстані цієї точки від центру і не залежить від напрямку. Таким чином, якщо «обертати» точки плану навколо центру, дисперсія передбачуваної реакції в будь-якій точці простору факторів буде однією і тією ж.

10.2. Тактичне планування імітаційного експерименту

Проблеми тактичного планування експерименту з імітаційної моделлю пов'язані із суттю статистичного моделювання, а закономірності, які цікавлять дослідника, можуть бути отримані в результаті статистичної обробки необхідного числа дослідів. Оцінки невідомих параметрів і змінних с їхніми взаємозв'язками флуктуірують відносно істинних значень і наближаються до них тільки у разі досить великих вибірок. Справедливо таке положення: чим більше чисельність експериментів, тим менше флуктуації результатів, а значить, і більш точні висновки будуть зроблені за дослідними даними. Завдання досліджень полягає в знаходженні компромісу між витратами машинного часу, точністю отриманих оцінок шуканих характеристик і ступенем довіри до результату. В рамках цього компромісу можна знайти способи збільшення точності при такому ж обсязі досліджень або деякі спеціальні методи обробки даних, які дозволяють при тій же довжині вибіркових сукупностей отримати більше корисної інформації.

Сукупність методів встановлення необхідного обсягу випробувань відносять до тактичного планування експериментів.

Оскільки точність оцінок спостережуваної змінної характеризується її дисперсією, то основу тактичного планування експерименту складають так звані методи зниження дисперсії.

<u>Формування простої випадкової вибірки</u>

Оскільки імітаційне моделювання являє собою статистичний експеримент, то при його проведенні необхідно не тільки отримати достовірний результат, а й забезпечити його «вимір» із заданою точністю.

Різниця понять «достовірний результат» і «точний результат» можна пояснити за допомогою рис. 10.1, на якому:

- y, y_x – дійсне і хибне значення спостережуваної змінної y;
- b, b_x – довірчі інтервали вимірювання величин y і y_x.

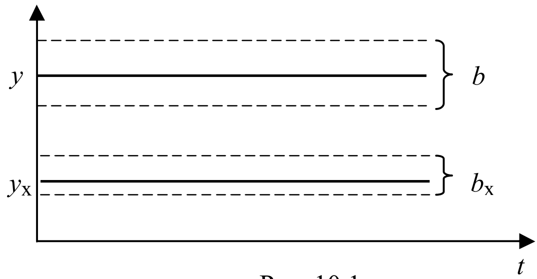


Рис. 10.1

У загальному випадку обсяг випробувань (величина вибірки), необхідний для отримання оцінок спостережуваної змінної із заданою точністю, залежить від таких факторів:

- виду розподілу спостережуваної змінної у (при статистичному експерименті вона є випадковою величиною);
- корельованістю між собою елементів вибірки;
- наявності та тривалості перехідного режиму функціонування модельованої системи.

Якщо дослідник не володіє переліченою інформацією, то у нього є єдиний спосіб підвищення точності оцінок дійсного значення спостережуваної змінної – багаторазовий прогін моделі для кожного сполучення рівнів факторів, обраного на етапі стратегічного планування експерименту. Такий підхід отримав назву «формування простої випадкової вибірки» (скорочено – ПВВ). Іншими словами, при використанні ПВВ кожен «пункт» стратегічного плану просто виконується повторно певне число разів. При цьому загальне число прогонів моделі, що необхідне для досягнення цілі моделювання, дорівнює добутку $N_C \cdot N_T$ (N_C – число сполучень рівнів факторів за стратегічним планом; N_T – число прогонів моделі для кожного сполучення, що обчислене при тактичному плануванні).

Наприклад, якщо для повного факторного експерименту $N_C = 64$, а для забезпечення необхідної точності оцінок N_T має дорівнювати 20, то загальне число прогонів моделі 1280. Необхідний час для проведення всіх випробувань (по хвилині на кожне) буде більш 20 годин. Тобто дослідник повинен працювати майже добу без сну і відпочинку. Такий режим роботи може вивести з ладу не тільки людини, а й комп'ютер. Тому навіть при використанні ПВВ до початку випробувань необхідно визначити той мінімальний обсяг вибірки, який забезпечить необхідну точність результатів.

Якщо випадкові значення спостережуваної змінної не корельовані та їхній розподіл не змінюється від прогону до прогону, то вибіркове середнє можна вважати нормально розподіленим. У цьому випадку число прогонів $N_{\text{Т}}$, яке необхідне для того, щоб дійсне середнє спостережуваної змінної лежало в інтервалі $y \pm b$ з ймовірністю $(1 - \alpha)$, визначається таким чином:

$$N_{\text{Т}} = \frac{Z^2 D_y}{b^2},$$

тут Z – значення нормованого нормального розподілу, яке визначається за довідковою таблицею при заданому рівні значимості $\alpha / 2$; D_y – дисперсія; b – довірчий інтервал.

Якщо необхідне значення дисперсії D_y до початку експерименту невідомо, доцільно виконати пробну серію з L прогонів і обчислити на її основі оцінку дисперсії, значення якої підставляють у вираз і отримують попередню оцінку числа прогонів $N_{\text{Т}}$. Потім виконують $(N_{\text{Т}} - L)$ прогонів, що залишилися, періодично уточнюючи оцінку D_y і число прогонів $N_{\text{Т}}$.

<u>Методи зниження дисперсії</u>

Основний недолік методів планування, заснованих на використанні простої випадкової вибірки це повільна збіжність вибіркових середніх до дійсних середніх з ростом обсягу вибірки $N_{\text{Т}}$ (пропорційно значенню квадратного кореня з $N_{\text{Т}}$). Це призводить до необхідності використання методів зменшення похибок, які не потребують збільшення $N_{\text{Т}}$. Такі методи називаються методами зниження дисперсії та діляться на три групи:

- активні (передбачають формування вибірки спеціальним чином);

- пасивні (застосовуються після того, як вибірка вже сформована);
- непрямі (в яких для отримання оцінок спостережуваної змінної використовуються значення деяких допоміжних величин).

Активних методів зниження дисперсії відомо досить багато. Вибір конкретного методу визначається, як правило, специфікою моделі та цілями експерименту. Розглянемо ті з них, які спрямовані на зниження впливу перехідного періоду. Вибір пояснюється тим, що наявність і тривалість перехідного режиму істотно впливають на якість результатів моделювання (у сенсі точності). Разом з тим, більшість ІМ використовується для вивчення функціонування системи у сталому режимі.

Сталим (стаціонарним) називається такий стан моделі, коли послідовні спостереження відгуку у сталому стані мають деякий граничний стаціонарний розподіл ймовірностей і не залежать від часу.

Стаціонарність режиму моделювання характеризує собою деяку встановлену рівновагу процесів у моделі системи, яка не залежить від часу (тобто якщо імітувати і далі, то нової інформації не отримаєш і продовження імітації буде безглуздою тратою часу).

Існує три основні методи зменшення помилок, обумовлених наявністю перехідного періоду:

- значне збільшення числа прогонів;
- виключення з розгляду перехідного періоду;
- ініціалізація моделі при деяких спеціально обраних вихідних умовах.

На практиці для зниження впливу перехідного періоду використовують один з відомих методів:

- метод повторення;
- метод підінтервалів;
- метод циклів.

Метод повторення. При використанні цього методу кожне спостереження отримують за допомогою окремого прогону моделі, причому всі прогони починаються при одних і тих же вихідних умовах, але використовуються різні послідовності випадкових чисел.

Перевагою методу є статистична незалежність одержуваних спостережень. Недолік полягає в тому, що спостереження можуть виявитися сильно зміщеними при зміні вихідних умов.

Метод підінтервалів. Даний метод заснований на розбитті кожного прогону моделі на рівні проміжки часу. Початок кожного інтервалу збігається з початком чергового етапу спостережень (на рис. 10.2 в якості спостережуваної змінної використовується довжина черги заявок Q).

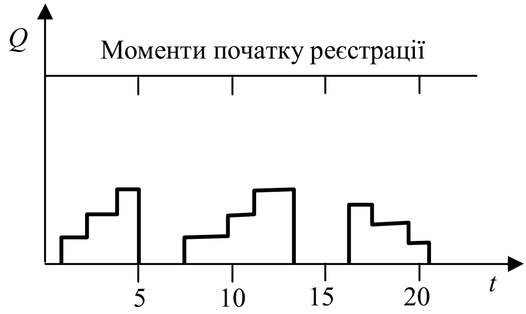


Рис. 10.2

Перевагою методу є те, що вплив перехідних умов з часом зменшується, і спостереження точніше відображають поведінку системи в стаціонарному режимі. До недоліків цього методу відноситься те, що значення спостережуваних змінних, що отримані на початку чергового інтервалу, залежать від кінцевих умов попереднього інтервалу (тобто між інтервалами існує автокореляція).

Метод циклів. При використанні методу циклів вплив автокореляції зменшується за рахунок вибору інтервалів такими, щоб в їх вихідних точках умови були однаковими. Наприклад, за такі умови можна розглядати довжину черги заявок на обслуговування. У цьому випадку зручно вибрати початок чергового інтервалу таким, щоб він збігається з моментом, коли довжина черги стає рівною нулю. Недолік методу – менше у порівнянні з методом підінтервалів число одержуваних спостережень (рис. 10.3).

Рис. 10.3

Метод стратифікованої вибірки. Даний метод відноситься до групи пасивних методів зниження дисперсії. Пасивні методи впливають на підготовку та проведення експерименту, але реалізуються на етапі обробки та аналізу результатів моделювання. Суть методу стратифікованої вибірки полягає в наступному.

Вибірка розділяється на шари, на так звані страти. При цьому необхідно, щоб значення елементів вибірки якомога менше відрізнялися всередині однієї страти і якомога більше між різними стратами. Усередині кожної страти здійснюється випадковий відбір елементів і обчислюється середнє значення страти y_i. Отримані оцінки використовують для обчислення математичного сподівання за вибіркою в цілому:

$$y = \frac{1}{N}\sum_{i=1}^{k} N_i y_i \ ,$$

тут N, N_i – об'єм всієї вибірки та i-ої страти відповідно;
k – кількість страт.

Якщо вважати, що оцінки y_i незалежні, то дисперсія за вибіркою в цілому дорівнює:

$$D_y = \frac{1}{N}\sum_{i=1}^{k} N_i D_{y_i} \ ,$$

тут D_{yi} – дисперсія для i-ої страти.

При вдалому виборі страт величини D_{yi} будуть малі, тому й вибіркова дисперсія D_y буде переважніше, ніж для оцінки, отриманою методами простої випадкової вибірки.

Непрямі методи зниження дисперсії засновані на тому, що часто деякі з вихідних характеристик моделі отримати (обчислити) легше, ніж інші. Їх використання передбачає не тільки досить глибоке знання суті процесів, що протікають в системі, а й наявність формального опису взаємної залежності параметрів моделі.

Контрольні запитання 

1. Які два основних варіанта постановки задачі планування імітаційного експерименту існують і як ці варіанти називають?
2. Які проблеми виникають при стратегічному плануванні імітаційного експерименту?
3. Що розуміють під ідентифікацією факторів?
4. На які дві групи доцільно розділити всі фактори при стратегічному плануванні імітаційного експерименту?
5. Що являють собою групи первинних та вторинних факторів?
6. З врахуванням яких суперечливих вимог здійснюється вибір рівнів факторів?
7. Який експеримент називають ПФЕ?
8. Як можна обчислити загальне число різних комбінацій рівнів в ПФЕ для *k* факторів?
9. У чому головний недолік ПФЕ?
10. Наведіть варіанти побудови планів ЧФЕ, що застосовуються на практиці?
11. Яким чином передбачає вибір поєднання рівнів для кожного прогону рандомізований план?
12. У якому випадку доцільно використовувати латинський план?
13. У чому суть експеримент зі зміною факторів по одному?
14. Наведіть матрицю планів для дробового факторного експерименту з двома факторами, які мають нижній і верхній рівень.
15. Що відносять до тактичного планування експериментів?
16. Які методи складають основу тактичного планування експерименту?
17. Від яких факторів залежить обсяг випробувань, необхідний для отримання оцінок спостережуваної змінної із заданою точністю?
18. Який основний недолік методів тактичного планування, заснованих на використанні простої випадкової вибірки?
19. На які групи при тактичному плануванні діляться методи зниження дисперсії?

Глава 11. ОБРОБКА Й АНАЛІЗ РЕЗУЛЬТАТІВ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ

Рішення, що приймаються дослідником за результатами імітаційного моделювання, можуть бути конструктивними тільки при виконанні двох основних умов:

- отримані результати мають необхідну точність та достовірність;
- дослідник здатний правильно інтерпретувати отримані результати, і знає, яким чином вони можуть бути використані.

Можливість виконання першої умови закладається, в основному, ще на етапі розробки ІМ і частково – на етапі планування експерименту. Достовірність результатів моделювання передбачає, що модель, за допомогою якої вони отримані, не тільки є адекватною, але відповідає і деяким додатковим вимогам, що пред'являються до ІМ.

При аналізі результатів моделювання здійснюється статистична обробка даних експериментів та інтерпретація результатів моделювання. До результатів моделювання відносять і аналіз чутливості моделі до варіацій її параметрів і, якщо це потрібно, її калібрування, тобто корекцію з ціллю приведення моделі у відповідність до встановлених вимог.

Під аналізом чутливості розуміють перевірку стійкості процесу функціонування системи до можливих відхилень значень її параметрів. Для аналізу залежностей характеристик системи до змін її параметрів і до зовнішніх впливів можна скористатися кореляційним, дисперсійним або регресійним методами.

Загальний процес калібрування моделі, зазвичай, носить ітераційний характер і складається з трьох основних етапів:

- глобальні зміни моделі (наприклад, введення нових процесів, зміна типів подій і т. д.);
- локальні зміни (зокрема, зміна деяких законів розподілу модельованих випадкових величин);
- зміна спеціальних параметрів, які називаються калібрувальними.

Конкретна процедура калібрування складається з трьох кроків, кожен з яких також є ітераційним (рис. 11.1)

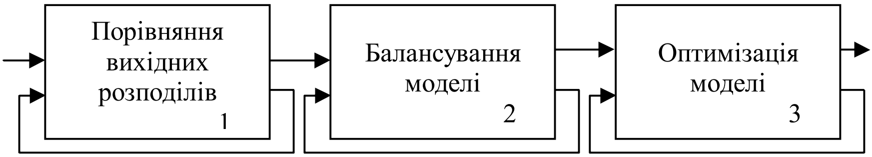


Рис. 11.1

1 крок. *Порівняння вихідних розподілів.* Мета цього кроку – оцінка адекватності ІМ. Критерії порівняння можуть бути різні. Зокрема, може використовуватися величина різниці між середніми значеннями відгуків моделі і системи. Усунення відмінностей на цьому кроці засноване на внесення глобальних змін.

2 крок. *Балансування моделі.* Основне завдання цього кроку – оцінка стійкості та чутливості моделі. За його результатами, як правило, проводяться локальні, але можливо й глобальні зміни.

3 крок. *Оптимізація моделі.* Мета цього кроку – забезпечення необхідної точності результатів. Тут можливі три основних напрямки робіт:

- додаткова перевірка якості датчиків випадкових чисел;
- зниження впливу перехідного режиму;
- застосування спеціальних методів зниження дисперсії.

У деяких випадках імітаційна модель складної системи може бути реалізована у вигляді набору окремих моделей її підсистем. При проведенні експериментів з такою моделлю з ціллю скорочення витрат часу доцільно замінити, використовуючи принцип параметризації, моделювання роботи однієї з підсистем деяким числовим параметром або випадковою величиною, розподіленою за заданим законом. Щоб така заміна була коректною, дослідник повинен мати у своєму розпорядженні опис залежності даного числового параметра від часу та інших факторів, що фігурують в моделі.

При імітаційному моделюванні підбір законів розподілів виконується на основі статистичних даних, отриманих в ході експерименту.

11.1. Підбір параметрів розподілів

В основі процедури пошуку закону розподілу деякої величини за експериментальними даними лежить перевірка статистичних гіпотез.

Статистична гіпотеза на основі аналізу статистичних даних формулює ствердження відносно значень одного або кількох критеріїв розподілу деякої величини або про саму форму розподілу.

Зазвичай вибирають дві вихідні гіпотези: основну – H_0 та альтернативну їй – H_1.

Статистична перевірка гіпотези – це процедура з'ясування, чи слід прийняти основну гіпотезу H_0 або відкинути її.

Якщо в результаті перевірки гіпотеза H_0 помилково відкидається, то має місце похибка першого роду (яка характеризується більш важкими наслідками); якщо гіпотеза H_0 приймається при істинності H_1 – похибка другого роду.

Ймовірності похибок I і II роду (α і β) залежать від критерію, на підставі якого буде вибиратися одна з гіпотез. Очевидно, що ймовірності цих двох похибок взаємопов'язані, тобто чим більше значення α, тим менше β, і навпаки. Звичайне розв‘язання цієї дилеми полягає в тому, що вибирають деяке фіксоване значення α (як правило, 0.05, 0.1, 0.001) і сподіваються, що β буде також мало. Фіксоване значення α називається *рівнем значимості*.

Для обраного значення α визначається так звана критична область B, яка задовольняє умові:

$$P(Z \in B \mid H_0 \text{ вірна}) \le \alpha,$$

де Z – контрольна величина (критерій), що являє собою деяку функцію від вибірки (результатів експерименту).

Перевірка гіпотези відбувається таким чином. Здійснюється вибірка (експеримент), на підставі чого обчислюється z – окреме значення критерію Z. Якщо $z \in B$, то від гіпотези H_0 відмовляються. Якщо z не належить B, то говорять, що отримані спостереження *не суперечать* прийнятій гіпотезі.

Зрозуміло, що перш ніж висувати гіпотезу щодо значень параметрів розподілу, необхідно визначити вид самого закону розподілу. Найбільш поширений на практиці й досить ефективний метод підбору закону розподілу заснований на використанні графічного подання експериментальних даних.

Вони відображаються у вигляді так званої гістограми відносних частот, яка може бути побудована як вручну, так і за допомогою відповідних інструментальних засобів, які входять до складу більшості пакетів математичного моделювання, зокрема до складу MATLAB. Зовнішній вигляд гістограми показаний на рис. 11.2.

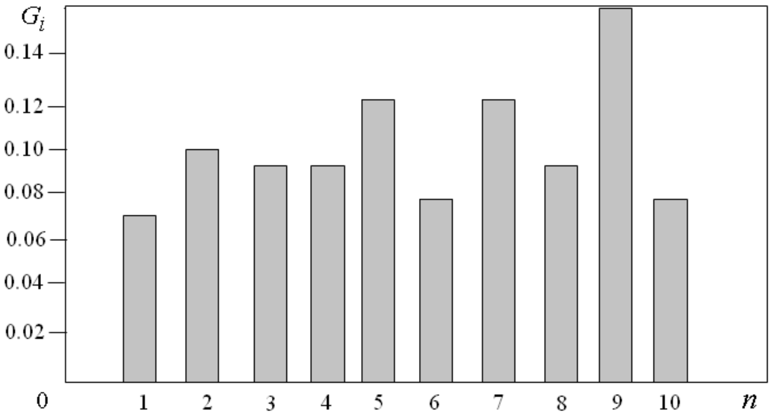


Рис. 11.2

Для ефективного використання графічних засобів пакету MATLAB треба знати методику побудови гістограми відносних частот.

1. Обчислюється величина інтервалу гістограми:

$$d = (y_{\max} - y_{\min})/n,$$

де $(y_{\max} - y_{\min})$ – діапазон зміни спостережуваної змінної; n – число інтервалів, обраних дослідником.

2. За результатом (або у процесі) моделювання визначається число влучень значень y в i-ий інтервал.

3. Обчислюється відносна частота влучень спостережуваної змінної в кожен інтервал:

$$G_i = R_i/N,$$

де R_i – число влучень у i-ий інтервал; N – загальне число вимірювань (обсяг вибірки).

4. На кожному i-ому інтервалі будується прямокутник зі сторонами $d \times G_i$. Сума площ прямокутників гістограми дорівнює одиниці.

Для найбільш часто використовуваних статистичних гіпотез розроблені критерії, які дозволяють проводити перевірку гіпотез з найбільшою вірогідністю. Розглянемо основні з них.

t-критерій. Цей критерій служить для перевірки гіпотези про рівність середніх значень двох нормально розподілених випадкових величин X і Y в припущенні, що дисперсії їх рівні (хоча і невідомі). Порівнянні вибірки можуть мати різний обсяг n_1 і n_2.

Як критерій використовують величину T:

$$T = \frac{\bar{X} - \bar{Y}}{\sqrt{(n_1 - 1)D_x + (n_2 - 1)D_y}} \sqrt{\frac{n_1 n_2 (n_1 + n_2 - 2)}{n_1 + n_2}}. \qquad (11.1)$$

Величина T підпорядковується t-розподілу Стьюдента.

Критичне значення для t-критерію ($t_{кр}$) визначається за таблицею для обраного значення α і числа ступенів свободи

$$k = n_1 + n_2 - 2.$$

Якщо обчислене за вказаною формулою значення T задовольняє нерівності $T > t_{кр}$, то гіпотезу H_0 відкидають.

Виконання нерівності $t_{кр} > T$ є підтвердженням припущення про приналежність випадкових величин X і Y до однієї генеральної сукупності (гіпотеза H_0). Вираз (11.1) можна застосовувати, якщо розподіли випадкових величин не мають кількох вершин і не занадто асиметричні.

F-критерій. Цей критерій служить для перевірки гіпотези про рівність дисперсій D_x и D_y за умови, що x і y розподілені нормально.

Гіпотези такого роду мають велике значення в техніці, оскільки дисперсія є мірою таких характеристик, як похибки вимірювальних приладів, точність технологічних процесів, точність наведення при стрільбі, тощо.

За контрольну величину тут використовується відношення дисперсій $f = D_x/D_y$ (або D_y/D_x – більша дисперсія повинна бути у чисельнику).

Величина f підпорядковується F-розподілу (розподілу Фішера) з (m_1, m_2) ступенями свободи ($m_1 = n_1 - 1$, $m_2 = n_2 - 1$). Перевірка гіпотези здійснюється так.

Для величини $a = \alpha/2$ і величин m_1, m_2 за таблицею F-розподілу вибирають значення F_a, m_1, m_2. Якщо f, що обчислена за вибіркою, більше цього критичного значення, гіпотеза повинна бути відхилена з ймовірністю похибки α.

Критерії згоди. Критерії згоди використовуються для перевірки того, чи задовольняє розглянута випадкова величина даному закону розподілу.

Критерій згоди Пірсона (χ^2) використовується для перевірки гіпотези H_0 про те, що $F_y(y) = F_0(y)$, де $F_y(y)$ – дійсний розподіл випадкової величини y; $F_0(y)$ – гіпотетичний розподіл. Перевірка проводиться таким чином.

1. Область значень випадкової величини y розбивається (довільно) на k неперетинних множин («класів»).

2. У результаті n дослідів формується вибірка (y_1, ...y_n).

3. Обчислюється контрольна величина χ^2

$$\chi^2 = \left(\sum_{i=1}^{k} \frac{M_i^2}{np_i} \right) - n,$$

тут M_i – число значень y, що потрапили в i-ий клас; p_i – теоретична ймовірність попадання значення y в і-ий клас для розподілу $F_0(y)$.

4. За таблицею χ^2 розподілів знаходять критичне значення $\chi\alpha^2$ для рівня значущості α и $m = k - 1$ ступенів свободи. Якщо $\chi^2 \geq \chi\alpha^2$, то гіпотеза відкидається.

Крітерій Колмогорова-Смирнова призначений для перевірки гіпотези H_0: «випадкова величина x має розподіл $f(x)$», тобто про приналежність аналізованої вибірки деякому повністю відомому закону розподілу.

Нехай x_n – вибірка незалежних однаково розподілених випадкових величин, $f_n(x)$ – емпірична функція розподілу, $f(x)$ - деяка "дійсна" функція розподілу з відомими параметрами. Статистика критерію визначається виразом:

$$D_n = \sup_x \left| f_n(x) - f(x) \right|.$$

Позначимо через H_0 гіпотезу про те, що вибірка підпорядковується розподілу $f(x) \in \mathbf{C}^1(\mathbf{X})$. Тоді за теоремою Колмогорова-Смирнова при справедливості гіпотези, що перевіряється:

$$\lim_{n \to \infty} P(\sqrt{n} D_n \le t) = K(t).$$

Гіпотеза H_0 відкидається, якщо статистика перевищує квантиль[11.1] розподілу K_α заданого рівня значущості α, і приймається в іншому випадку.

Критичне значення критерію, як і у попередніх випадках, знаходиться за таблицею.

Схематично алгоритм застосування критерію Колмогорова-Смирнова можна представити таким чином (див. рис. 11.3):

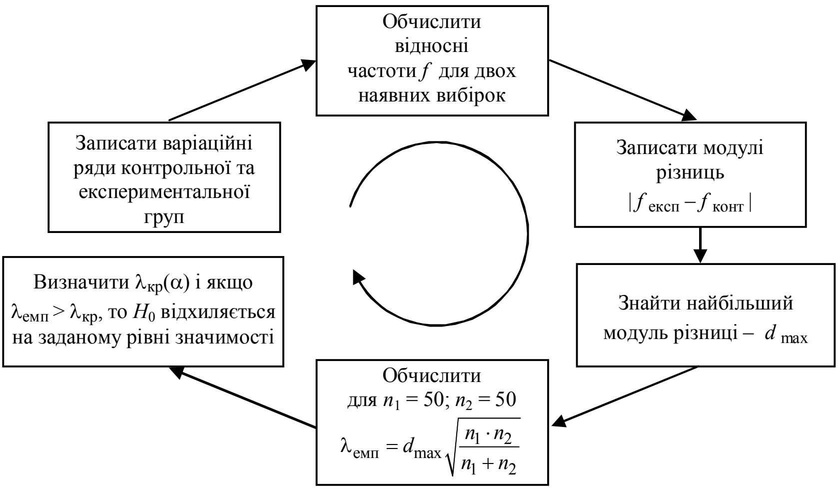


Рис. 11.3

Проілюструємо використання критерію Колмогорова-Смирнова на прикладі.

[11.1] Квантиль - значення, яке задана випадкова величина не перевищує з фіксованою ймовірністю.

При вивченні творчої активності студентів були отримані результати для експериментальних і контрольних груп (табл. 11.1). Чи є значущими відмінності між контрольною та експериментальної групами?

Таблиця 11.1

Рівень засвоєння	Частота в експериментальної групі	Частота в контрольної групі
Добрий	172 чол.	120 чол.
Задовільний	36 чол.	49 чол.
Незадовільний	15 чол.	36 чол.
Обсяг вибірки	n_1=172+36+15=223	n_2=120+49+36=205

Обчислюємо відносні частоти *f*, що дорівнюють частці від ділення частот на обсяг вибірки, для двох наявних вибірок.

Далі визначаємо модуль різниці відповідних відносних частот для контрольної та експериментальної вибірок. В результаті вихідна таблиця (табл. 11.2) прийме такий вигляд:

Таблиця 11.2

Відносна частота експериментальної групи ($f_{\text{експ}}$)	Відносна частота контрольної групи ($f_{\text{контр}}$)	Модуль різності частот $\lvert f_{\text{експ}} - f_{\text{контр}}\rvert$
172/223 ≈ 0.77	120/205 ≈ 0.59	0.18
36/223 ≈ 0.16	49/205 ≈ 0.24	0.08
15/223 ≈ 0.07	36/205 ≈ 0.17	0.1

Серед отриманих модулів різниць відносних частот вибираємо найбільший модуль, який позначається $d_{\max}$. У розглянутому прикладі $0.18 > 0.1 > 0.08$, тому $d_{\max} = 0.18$.

Емпіричне значення критерію $\lambda_{\text{емп}}$ визначається за допомогою формули:

$$\lambda_{\text{емп}} = d_{\max} \cdot \sqrt{\frac{n_1 \cdot n_2}{n_1 + n_2}}\,.$$

Щоб зробити висновок про схожість за даним критерію двох груп, порівняємо експериментальне значення критерію з його критичним значенням, що визначається за спеціальною таблицею, виходячи з рівня значущості α. За нульову гіпотезу приймемо створдження про те, що порівнювані групи незначно відрізняються один від одного за рівнем засвоєння. При цьому нульову гіпотезу слід прийняти в тому випадку, якщо спостережуване значення критерію не перевищує його критичного значення:

$$\alpha_{емп} = 0{,}18 \cdot \sqrt{\frac{223 \cdot 205}{223 + 205}} \approx 1{,}86\,.$$

Вважаючи, що α = 0,05, за таблицею визначаємо критичне значення критерію: $\lambda_{кр}$ (0,05) = 1,36.

Таким чином, $\lambda_{емп}$ = 1,86 > 1,36 = $\lambda_{кр}$. Отже, нульова гіпотеза відкидається, і групи за розглянутою ознакою відрізняються істотно.

Зауважимо, що обсяги розглянутих вибірок повинні бути досить великими: $n_1 \geq 50$, $n_2 \geq 50$.

Зрозуміло, проведення вручну розрахунків, необхідних для перевірки статистичних гіпотез, вимагає значних витрат часу та сил. Тому більшість сучасних математичних пакетів мають у своєму складі засоби, які дозволяють звести до мінімуму число операцій, що виконуються вручну.

11.2. Оцінка впливу і взаємозв'язку факторів

Як правило, кількісна оцінка ступеня впливу того чи іншого фактора на спостережувану змінну (показник ефективності) викликає значну складність, особливо при наявності взаємного впливу факторів. Найбільш простий та доступний спосіб вирішення цієї проблеми полягає у використанні результатів оцінки чутливості моделі. Ці результати складно уявити в формі аналітичної залежності, однак таке подання може виявитися корисним для багатьох практичних завдань, пов'язаних як з розробкою моделей, так і з прийняттям рішень щодо результатів експериментів.

Пошук аналітичних залежностей, що пов'язують між собою різні параметри, які фігурують в моделі, може бути заснований на спільному використанні групи методів математичної статистики: дисперсійного, кореляційного та регресійного аналізу.

Однофакторний дисперсійний аналіз.

Суть однфакторного дисперсійного аналізу зводиться до визначення впливу на результат моделювання одного обраного фактору.

Нехай, наприклад, дослідника цікавить середня інтенсивність відмов комп'ютера, і в створеній ним моделі враховані такі фактори: інтенсивність надходження завдань користувачів, інтенсивність звернень в оперативну пам'ять, часові характеристики вирішуваних завдань та інтенсивність звернень до жорсткого диска. Якщо попередні дані говорять про те, що основною причиною відмов є ненадійна робота жорсткого диска, то як аналізований фактор доцільно вибрати інтенсивність звернень до диска. Задача аналізу в даному випадку полягає в тому, щоб оцінити вплив зазначеного фактора на середнє число відмов.

Формально постановка задачі однофакторного дисперсійного аналізу така. Нехай аналізований фактор x має l рівнів. Для кожного з них отримана вибірка значень спостережуваної змінної y:

$$y_j(1),\ y_j(2),\ ...y_j(l),\ j=1,\ ....\ n,$$

де n – обсяг вибірки (число спостережень).

Необхідно перевірити гіпотезу H_0 про рівність середніх значень вибірок (тобто про незалежність значень y від значень досліджуваного фактора x). Рівняння однофакторного дисперсійного аналізу має вигляд:

$$y_{ij} = m + a_i + e_{ij},$$

y_{ij} – j-е значення y в i-ої серії дослідів; m – генеральне середнє випадкової величини y (тобто середнє значення спостережуваної змінної, що обумовлене її «сутністю»); a_i – невідомий параметр, який відображає вплив фактора x («ефект» i-го значення фактора x); e_{ij} – похибка вимірювання y.

Для перевірки гіпотези H_0 використовують F-критерій і переходять від перевірки значущості відмінностей середніх до перевірки значущості відмінностей двох дисперсій:

- генеральної (обумовленої похибками вимірювань) – D_0;
- факторної (обумовленої зміною фактора x) – D_x.

Значення F-критерію обчислюється також, як відношення D_x/D_0 або D_0/D_x (в чисельнику повинна стояти більша з дисперсій); потім за таблицею F-розподілів знаходять його критичне значення $F_{кр}$ для заданого рівня значущості та числа ступенів свободи. Якщо $F > F_{кр}$, то гіпотезу H_0 відкидають, тобто відмінності є значущими (фактор x впливає на значення y).

Багатофакторний дисперсійний аналіз

Багатофакторний дисперсійний аналіз (БДА) дозволяє оцінювати вплив на спостережувану змінну вже не одного, а довільного числа факторів. Точніше, БДА дозволяє вибрати з групи факторів, що беруть участь в експерименті, ті, які дійсно впливають на його результат.

Методику проведення багатофакторного дисперсійного аналізу розглянемо для часткового факторному експерименту, проведеного згідно латинського плану.

Нехай в експерименті розглядаються один первинний фактор і два вторинних, кожен з яких має n рівнів (тобто обсяг випробувань дорівнює $N = n^2$), Позначимо через y_{ijk} результат експерименту за умови, що фактор a знаходився на рівні i, фактор b – на рівні j, фактор c – на рівні k. Множину значень, які може приймати впорядкована трійка (i, j, k), позначимо через L. У цьому випадку рівняння дисперсійного аналізу виглядає таким чином:

$$y_{ijk} = m + a_i + b_j + g_k + e_{ijk}\,,$$

тут m – генеральне середнє випадкової величини y;
a_i, b_j, g_k – невідомі параметри («ефекти» відповідних факторів).

Розв‘язання задачі дисперсійного аналізу полягає в перевірці гіпотез про незалежність результатів вимірювань від факторів a, b, c:

$$H_a:\ a_i = 0\,,\qquad i = 1,n;$$

$$H_b:\ b_j = 0\,,\qquad j = 1,n;$$

$$H_c:\ g_k = 0,\qquad k = 1,n.$$

Для цього за методом найменших квадратів (МНК) знаходять оцінки параметрів m, a_i, b_j, g_k, мінімізуючи за вказаними змінним (по черзі) функцію:

$$ss = \sum_L (y_{ijk} - m + a_i + b_j + g_k)^2\,.$$

Потім за кожним фактором обчислюється F-статистика. Величина F є мірю втрат при прийнятті гіпотези H_0. Чим більше F, тим гірше модель, що відкидає вплив відповідного фактора. Отже, якщо обчислене значення F більше $F_{кр}$, знайденого за таблицею для деякого рівня значущості (і $n - 1$, $n^2 - 3n + 2$ ступенів свободи), то гіпотеза відкидається.

Необхідно відзначити, що дисперсійний аналіз може використовуватися для оцінювання впливу факторів, що мають як кількісний, так і якісний характер, оскільки в рівнянні дисперсійного аналізу фігурують не самі фактори, а тільки їх «ефекти».

У разі, якщо всі фактори носять кількісний характер, взаємозв'язок між ними і спостережуваною змінної може бути описаний за допомогою рівняння регресії.

Кореляційний та регресійний аналіз

Кореляційний та регресійний аналіз – це два близьких методу, які зазвичай використовуються спільно для дослідження взаємозв'язку між двома або більше безперервними змінними.

Результати *кореляційного аналізу* дозволяють робити статистичні висновки про ступінь залежності між змінними.

Величина лінійної залежності між двома змінними вимірюється за допомогою *простого* коефіцієнта кореляції, величина залежності від декількох – за допомогою *множинного* коефіцієнта кореляції.

У кореляційному аналізі використовується також поняття *окремого* коефіцієнта кореляції, який вимірює лінійний взаємозв'язок між двома змінними, без урахування впливу інших змінних.

Якщо кореляційний аналіз дозволив встановити наявність лінійної залежності спостережуваної змінної від однієї або більше незалежних змінних, то форма залежності може бути уточнена методами регресійного аналізу.

Для цього будується так зване рівняння регресії, яке пов'язує залежну змінну з незалежними і містить невідомі параметри. Якщо рівняння лінійно відносно параметрів (але необов'язково лінійно відносно незалежних змінних), то форма регресії лінійна, в іншому випадку регресія нелінійна.

Розглянемо простий кореляційний аналіз, тобто метод визначення взаємозв'язку між двома змінними.

Позначимо їх x і y. Незалежно від способу отримання вибірки є два попередніх кроку для визначення існування та ступеня лінійної залежності між x і y. Перший крок передбачає графічне відображення точок (x_i, y_i) на площині (x, y), тобто побудову діаграми розсіювання. Аналізуючи діаграму розсіювання(рис. 11.4 *а*), можна з‘ясувати, чи припустимо припущення про лінійну залежність між x і y.

Якщо коефіцієнт r_{xy} не дорівнює нулю, то на другому кроці обчислюється його точне значення. Чим більше за абсолютним значенням r_{xy}, тим сильніше лінійна залежність між змінними. При $|r_{xy}| = 1$ (рис. 11.4 *б*) має місце функціональна лінійна залежність між x і y виду $y = b_0 + b_1 x$, причому якщо $r_{xy} = +1$, то говорять про додатну кореляцію, тобто більші значення однієї величини відповідають більшим значенням іншої; при $r_{xy} = -1$ має місце від’ємна кореляція; при $0 < r_{xy} < 1$ імовірна або лінійна кореляція з розсіюванням (рис. 11.4 *в*), або нелінійна кореляція (рис. 11.4 *г*).

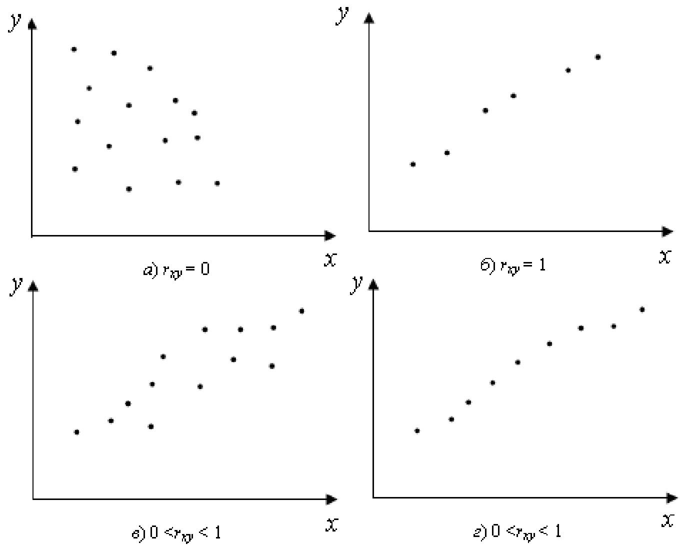


Рис. 11.4

При аналізі результатів імітаційного моделювання необхідно мати на увазі, що навіть встановлення щільної залежності між двома змінними, не є прямим доказом їх причинно-наслідкового зв'язку.

Можливо, має місце стохастична залежність, яка обумовлена, наприклад, корельованістю послідовностей псевдовипадкових чисел, що використовуються в ІМ. Тому результати кореляційного аналізу доцільно уточнити, шляхом регресійного аналізу.

<u>*Регресійний аналіз*</u> дозволяє розв'язувати дві задачі:

- встановлювати наявність можливого причинного зв'язку між змінними;
- передбачати значення змінної за значеннями незалежних змінних (ця можливість особливо важлива в тих випадках, коли прямі вимірювання залежної змінної ускладнені).

Якщо передбачається лінійна залежність між x і y, то вона може бути описана рівнянням, яке називається *простою лінійною регресією* y за x, що має вигляд:

$$y_i = b_0 + b_1 \cdot x_i + e_i,\ i = 1, ... n,$$

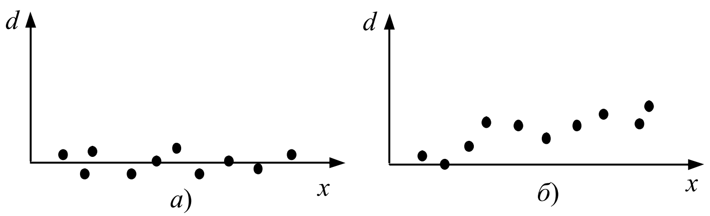
тут n – обсяг випробувань; b_0 і b_1 – невідомі параметри (b_1 називають *коефіцієнтом регресії*); e_i – випадкові похибки випробувань.

Мета регресійного аналізу – знайти найкращі в статистичному плані оцінки параметрів b_0 і b_1. А знаючи значення b_0 и b_1, можна знайти оцінку змінної y при $x = x_i$:

$$\widehat{y}_i = \widehat{b}_0 + \widehat{b}_1 x_i .$$

Різниця між спостережуваним і оціненим значенням y при $x = x_i$ називається відхиленням (або залишком) $d_i = y_i - \hat{y}_i$. Відхилення можуть бути використані для перевірки адекватності отриманої моделі. Для цього будується графік $d = f\ (y)$ або $d = f\ (x)$ (рис. 11.5) і з його вигляду робиться попередній висновок про ступінь адекватності моделі:

– модель адекватна (рис. 11.5 *а*);



– необхідно ввести додаткову незалежну змінну (рис. 11.5 *б*).

Рис. 11.5

У випадку декількох незалежних змінних має місце *множинна лінійна регресія*:

$$y = b_0 + b_1 x_1 + ... + b_k x_k + e.$$

Тут для пошуку оцінок b_j також використовується МНК.

У випадку нелінійної регресії основою для побудови регресійної моделі знову-таки є МНК. Однак в цьому випадку для пошуку оцінок b_j будується система нелінійних рівнянь (відносно b_j), а для її розв‘язання використовуються різні ітераційні методи.

Контрольні запитання

1. Що розуміють під аналізом чутливості моделі?
2. Якими методами користуються для аналізу залежностей характеристик системи до змін її параметрів і до зовнішніх впливів?
3. З яких етапів складається процес калібрування моделі?
4. З яких кроків складається конкретна процедура калібрування моделі?
5. Яка мета першого кроку процедури калібрування моделі?
6. Яке завдання другого кроку процедури калібрування моделі?
7. Яка мета третього кроку процедури калібрування моделі?
8. Яким чином при моделюванні можна замінити окремі моделі підсистем складної системи?
9. Що лежить в основі процедури пошуку закону розподілу деякої величини за експериментальними даними?
10. Які твердження формулює статистична гіпотеза?
11. Перелічіть критерії, які дозволяють проводити перевірку найбільш часто використовуваних статистичних гіпотез.
12. Для перевірки якої гіпотези служить F-критерій?
13. Для перевірки якої гіпотези служать критерії згоди Пірсона та Колмогорова-Смирнова?
14. Які методи математичної статистики використовуються для оцінки впливу і взаємозв'язку факторів при імітаційному моделюванні?
15. Сформулюйте постановку задачі однофакторного дисперсійного аналізу.
16. З якою метою проводять кореляційний та регресійний аналіз?
17. Що надають результати кореляційного аналізу?
18. Які задачі дозволяє розв'язувати регресійний аналіз?

СПИСОК ЛІТЕРАТУРИ

1. *Бахвалов Н.С., Жидков Н.П., Кобельков Г.М.* Численные методы. М.: Бином, 2008.

2. *Букетов А.В.* Ідентифікація і моделювання технологічних об'єктів та систем. Тернопіль: СМП „Тайп“ 2009 – 234 с.

3. *Бычков С. П., Храмов А. А.* Разработка моделей в системе моделирования GPSS: учеб. пособие. – М.: МИФИ, 1997.

4. *Громов Ю.Ю., Земской Н.А., Лагутин А.В.* и др. Системный анализ в информационных технологиях. Тамбов: Изд-во ТГТУ, 2004. 176 с.

5. *Джексон П.* Введение в экспертные системы. : Пер. с англ. : Уч. Пос. – М. :. Изд. дом "Вильямс", 2001.

6. *Єріна А. М.* Статистичне моделювання та прогнозування. К.: КНЕУ, 2001. – 223 с.

7. *Замятина О. М.* Компьютерное моделирование – Томск: Изд-во ТПУ, 2007. – 121 с.

8. *Кобзарь А. И.* Прикладная математическая статистика. Справочник для инженеров и научных работников. — М.: Физматлит, 2006. — 816 с.

9. *Козин Р.Г.* Алгоритмы численных методов линейной алгебры и их программная реализация. М.: МИФИ, 2012. – 192с.

10. *Лебідь Р.Д., Жуков І.А., Гузій М.М.* Математичні методи в моделюванні систем: навч. Посібник К.: КМУЦА, 2000 – 320 с.

11. *Овсянников А.В.* Математическое моделирование нестационарных случайных процессов на основе стохастических дифференциальных уравнений // Математика и математическое образование. Теория и практика: Межвуз. Сб. научн. Тр. Вып.4 – Ярославль: Изд. ЯГТУ, 2004.

12. *Снетков Н.Н.* Имитационное моделирование экономических процессов. – М.: Изд. центр ЕАОИ, 2008. – 228 с

13. *Советов Б. Я., Яковлев С. А.* Моделирование систем: Учебник для вузов. М., Высшая школа, 2001 – 258 с.

14. *Томашевський В.М.* Моделювання систем. - К.: Робоча група BHV, 2005 – 352 с.

Навчальне видання

ФІЛЯШКІН Микола Кирилович
КАЛІНИЧЕНКО Володимир Володимирович
КЕМЕНЯШ Юрій Михайлович
ТУПІЦИН Микола Федорович

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МОДЕЛЮВАННЯ СИСТЕМ ЦИВІЛЬНОЇ АВІАЦІЇ

Навчальний посібник

Редактор *М.П. Мухіна*
Технічний редактор *А.П. Козлов*
Коректор *Є.П. Бортін*

Підписано до друку 06.02.17. Формат 60×90 $^{1}/_{16}$.
Ум. друк. арк.0,9. Обл.-вид. арк. 0,9.
Наклад 100 прим. Замовлення № 311.
Віддруковано на різографі.
Видавництво: СПД Нестроєвий А.І.,
Свідоцтво ДК № 2187 від 17.05.2005 р.
04053 м. Київ, вул. Січових Стрільців, 26А
тел./факс 486-55-15, 277-40-16
http://www.printc/com.ua. E-mail printc@ukr.net