

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
імені ВОЛОДИМИРА ДАЛЯ

Факультет комп'ютерних технологій  
Кафедра комп'ютерно-інтегрованих систем управління

**МЕТОДИЧНІ ВКАЗІВКИ**

до виконання лабораторних робіт з дисципліни  
**«ОСНОВИ КОМП'ЮТЕРНО-ІНТЕГРОВАНОГО УПРАВЛІННЯ»**  
для студентів денної та заочної форми навчання за напрямом підготовки  
6.050202 «Автоматизація та комп'ютерно-інтегровані технології»  
Тема: «Інформаційне забезпечення АСКТП»  
**(електронне видання)**

ЗАТВЕРДЖЕНО  
на засіданні кафедри КІСУ  
Протокол №9 від 16.06.2015 р.

Сєверодонецьк, 2015

УДК 681.518

Методичні вказівки до виконання лабораторних робіт з курсу «Основи комп'ютерно-інтегрованого управління» для студентів денної та заочної форми навчання за напрямом підготовки 6.050202 «Автоматизація та комп'ютерно-інтегровані технології». Тема: «Інформаційне забезпечення АСКТП» / Укладачі: Проказа О.І., Кузнецова О.В. – Сєверодонецьк: Вид-во СНУ ім. В.Даля, 2015.- 36 с.

У методичних вказівках викладені матеріали, необхідні для виконання лабораторних робіт з дисципліни «Основи комп'ютерно-інтегрованого управління». Методичні вказівки містять у собі необхідні теоретичні відомості по темам лабораторних робіт, основні матеріали та пояснення про зміст і хід виконання роботи, варіанти завдань до виконання лабораторних робіт, приклади рішення типових завдань у середовищі TPWin, що дозволяють одержати поглиблене уявлення про суть розглянутого питання, а також список літератури.

Укладачі:

О.І. Проказа, доцент

О.В. Кузнецова, ст. викладач

Рецензент:

Й.І. Стенцель, д.т.н., проф.

## ЗМІСТ

ВСТУП.....	4
ЗАГАЛЬНІ ВКАЗІВКИ ДО ВИКОНАННЯ РОБІТ.....	6
ЛАБОРАТОРНА РОБОТА №1.....	7
ЛАБОРАТОРНА РОБОТА №2.....	13
ЛАБОРАТОРНА РОБОТА №3.....	17
ЛАБОРАТОРНА РОБОТА №4.....	24
ДОДАТКИ.....	29
СПИСОК ЛІТЕРАТУРИ.....	35

## ВСТУП

Методичні вказівки складені відповідно до робочої програми з курсу “Основи комп’ютерно-інтегрованого управління” за напрямом підготовки 6.050202 „Автоматизація та комп’ютерно-інтегровані технології”.

Метою даних методичних вказівок є допомога студентам у формуванні знань й умінь, необхідних для вибору, проектування та експлуатації інформаційного забезпечення АСКТП.

Функції АСКТП поділяються на керуючі, інформаційні та допоміжні. Інформаційні функції АСКТП – це функції системи, змістом яких є збір, обробка і представлення інформації про стан автоматизованого технологічного комплексу (АТК) оперативному персоналу, або передача цієї інформації для наступної обробки. До інформаційних функцій АСКТП відносяться:

- централізований контроль і вимір технологічних параметрів;
- непрямий вимір технологічних параметрів;
- обчислення параметрів процесу;
- формування і видача даних оперативному персоналу;
- підготовка і передача інформації в суміжні системи керування;
- узагальнена оцінка і перевірка стану АТК і його устаткування.

Лабораторні роботи охоплюють тільки невелику частину інформаційних функцій. Але реалізація навіть обмеженої кількості функцій дозволить студентам одержати навички не тільки розробки та налагодження окремих блоків, а також забезпечення їх взаємодії та сумісної роботи.

В результаті вивчення дисципліни студент зобов’язаний знати:

- роль інформаційного забезпечення в складі АСКТП;
- підстави й принципи його побудови та функціонування;
- основні застосовані алгоритми;
- методи розробки;

вміти :

- формулювати завдання на розробку інформаційного забезпечення;

- вибирати необхідні алгоритми;
- підбирати початкові дані та обмеження на розробку;
- будувати блок-схем алгоритмів;
- реалізовувати алгоритми на одній з мов програмування.

## **ЗАГАЛЬНІ ВКАЗІВКИ ДО ВИКОНАННЯ РОБІТ**

### **1. Порядок виконання лабораторних робіт**

Лабораторні роботи складаються з:

- одержання індивідуального завдання;
- одержання шуканих результатів;
- оформлення й захист звіту з лабораторної роботи.

Для виконання лабораторних робіт необхідно уважно ознайомитися з теоретичними основами кожної роботи, приведеними в методичних вказівках. Для більш глибокого ознайомлення з матеріалом, необхідно звертатися до конспекту лекцій, а також до технічної літератури, на яку приводяться посилання у вказівках.

Якщо під час вивчення дисципліни з'являться запитання, на які не знаходиться відповідь у конспекті лекцій або технічній літературі, необхідно звертатись за консультацією до ведучого лектора.

Лабораторні роботи необхідно виконувати за допомогою комп'ютера в інтегрованому середовищу TPWin.

### **2. Правила оформлення звітів з лабораторних робіт**

Звіт повинен містити:

- титульний аркуш із найменуванням лабораторної роботи й даними виконавця;
- мета роботи;
- постановка задачі відповідно до варіанту;
- блок - схему алгоритму;
- текст програми, що реалізує заданий алгоритм;
- висновки з виконаної роботи;
- список літератури.

Всі матеріали лабораторної роботи необхідно роздрукувати на листах формату A4, зброшурувати, сторінки пронумерувати.

# ЛАБОРАТОРНА РОБОТА №1

## СТВОРЕННЯ ФАЙЛУ ІСТОРІЇ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ

### 1. Мета

Ознайомлення з методами програмної імітації стандартних дискретних сигналів та збереження структурованої інформації на диску. Придбання навичок програмної реалізації розроблених алгоритмів.

### 2. Методичні вказівки до організації самостійної роботи студентів

При підготовці до даної роботи необхідно повторити такий матеріал:

- Правила складання блок - схем і опису алгоритмів.
- Правила програмування з використанням алгоритмічної мови ПАСКАЛЬ.

Необхідно вивчити методи модуляції безперервних сигналів і організації збереження історії технологічного процесу.

### 3. Теоретичні відомості

Оскільки АСКТП має багато інформаційно-вимірювальних каналів (ІВК), їхнє опитування розділене в часі, кожен канал періодично з періодом  $t_0$  підключається на короткий час до входу ЕОМ. У результаті безперервна функція вхідного сигналу  $g(t)$  перетвориться в послідовність імпульсів, модульованих по амплітуді функцією  $g(t)$ . Отримана дискретна функція піддається подальшій обробці відповідно до алгоритмів, передбачених в конкретній АСКТП. В ході виконання лабораторних робіт надходження вхідного дискретного сигналу в систему буде імітуватись програмне.

Однією з важливих функцій АСКТП є функція ведення історії технологічного процесу. Усі дії оператора, які будь-яким чином впливають на функціонування системи керування, реєструються у файлах історії процесу поряд із записами про знайдені порушення та значеннями технологічних параметрів. Накопичена інформація далі використовується для:

- Підготовки та видачі відповідних звітів про хід технологічного процесу;
- Аналізу ходу технологічного процесу;
- Підготовки даних для оптимізації.

Підсистема історії технологічного процесу викликається періодично. Частота застосування залежить від важливості параметра, можливостей системи накопичення інформації (ємність пристроїв накопичення, пропускна спроможність ІВК) та мети її подальшого використання.

У даній роботі організується структурований файл послідовного доступу. Такі файли не дуже зручні при модернізації інформації, але легко формуються й не потребують великих ресурсів. При цьому вони цілком задовільно демонструють функції інформаційного архіву.

Для пошуку, вилучення, модифікації та використання інформації застосовують системи керування базами даних (СКБД). Головна роль СКБД міститься в забезпеченні користувача приладом для роботи з даними, незалежно від способу їх зберігання в ЕОМ.

#### **4. Індивідуальні завдання до виконання роботи**

Розробити алгоритм і скласти програму, що імітує вхідний дискретний сигнал. Сформувати структуру файлу історії технологічного процесу. Одержати вхідний дискретний сигнал із параметрами, заданими відповідно до таблиці варіантів (додаток №1). Записати отриману інформацію у файл історії. Для можливості перевірки сформувати копію файлу історії у вигляді текстового файлу.

#### **5. Порядок виконання і методичні вказівки до виконання роботи**

5.1. Скласти блок-схему алгоритму, що реалізує імітацію вхідного дискретного сигналу.

5.2. Скласти програму алгоритмічною мовою ПАСКАЛЬ, що реалізує даний алгоритм, використовуючи модуль користувача.

Один із можливих алгоритмів створення бібліотеки функцій імітації вхідного дискретного сигналу:

```
Unit Source;  
Interface  
Function Step (left,right:real):real;  
Function Signal (LastSignal,stp:real):real;  
Implementation  
Function Step (left,right:real):real;  
    Begin  
        Randomize;  
        Step := (right – left)/2;
```



```

End;
Function Signal (LastSignal,stp:real):real;
Begin
    Signal:= LastSignal + (Random – 0.5) * stp;
End;
End.

```

***Бібліотечні функції, які використані у програмі:***

**Function Step (left, right: real): real;**

Функція використовується для розрахунку початкового максимального діапазону змін вхідного сигналу. Розраховується один раз на початку дії програми.

*Параметри:*

left - нижня межа заданого стандартного вхідного сигналу,  
right - верхня межа заданого стандартного вхідного сигналу.

**Function Signal (LastSignal, stp : real): real;**

Функція використовується для імітації вхідного дискретного сигналу.

*Параметри:*

LastSignal - діюче значення вхідного сигналу.

Stp - діапазон змін вхідного сигналу, розрахований функцією

Step.

5.3. Виконати збереження файлу з модулем у поточну папку, причому найменування файлу та ім'я програми повинні бути однаковими.

5.4. В меню **Compile** натиснути на опцію **Destination Memory**, яка після цього зміниться на **Destination Disk**.

5.5. Виконати компіляцію модулю. Для компіляції програми необхідно вибрати з меню **Compile** команду **Compile**. Турбо Паскаль відкомпілює вашу програму, перевівши її з мови Паскаль (яка зрозуміла вам) у машинні коди для мікропроцесора (які ваш персональний комп'ютер може виконати). У поточній папці з'явиться файл **Source.tpu** .

Після успішної компіляції закрити файл з модулем.

5.6. Скласти блок-схему формування файлу історії технологічного процесу.

5.7. Скласти програму алгоритмічною мовою ПАСКАЛЬ, що реалізує даний алгоритм. У ролі імітатора дискретного вхідного сигналу використовуються функції з бібліотеки **Source**.

***Підключення бібліотеки виконується виразом***

Uses Source;

Для запису даних на диск, необхідно сформувати файл відповідної структури. Як приклад розглянемо файл історії технологічного процесу такої структури:

```
< Найменування типу для файлів історії > = Record
    Numb: integer; { номер опитування }
    DataInput: real; { дійсне значення сигналу }
End;
```

Структура файлу складається з декількох полів. Номер опитування після множення його на період опитування дає час одержання інформації. У друге поле записується одержане значення вхідного сигналу.

В ході одержання інформації, необхідно записувати її у завчасно відкритий файл історії технологічного процесу і записати його на диск. Крім того, для більш зручного контролю роботи програми, по завершенні її роботи додатково організується вивід даних на екран монітора й у текстовий файл.

### **Приклад застосування імітатора:**

Program Name;

Uses Source;

Const

```
    StartSignal = 10; { початкове значення вхідного сигналу }
    ParamCycle = 30; { кількість вхідних значень }
    DownSignal = 4; { нижня межа вхідного нормованого сигналу }
    UpSignal = 20; { верхня межа вхідного нормованого сигналу }
```

Type{ Описання інформаційної структури }

StructureArch = record

```
    Numb: integer; { номер опитування }
    DataInput: real; { дійсне значення сигналу }
```

End;

Var

```
    Arch: file of StructureArch; { файлова змінна для бінарного
файлу }
```

```
    a: StructureArch;
```

```
    t: text; { файлова змінна для текстового
```

```
файлу }
```

```
    i: integer;
```

```
    Stp, LastSignal, CurrentSignal: real;
```

```

Begin
{ Назначаємо файлову змінну та відкриваємо файл історії для запису }
  Assign (arch, '<ім'я файлу даних>');           { напр., файл на ім'я 'FILE.DAT' }

  Rewrite (arch);
  Stp := Step (DownSignal, UpSignal);
  CurrentSignal := Signal (StartSignal, Stp);
{ При першому застосуванні функції Signal задається початкове значення
вхідного сигналу. Одержане значення треба перевірити на відповідність
допуску. }
  For i := 1 to ParamCycle do
  Begin
    LastSignal := CurrentSignal; { зберігання попереднього
                                  поточного значення сигналу }
    CurrentSignal := Signal (LastSignal, Stp); { одержання імітації
                                                  поточного сигналу }
{ Присвоювання конкретних значень відповідним полям структурованої змінної
та запис їх на диск. }
    a.numb := i; { номер опитування }
    a.dataInput := CurrentSignal; { дійсне значення сигналу }
    write(arch,a); { Запис даних на диск у бінарний файл історії }

  End;
  Close(arch);
{ Читання даних із файлу історії та вивід їх для контролю на екран монітора та у
текстовий файл }
  Reset (arch);
  Assign (t, '<ім'я текстового файлу даних>');   { напр., файл на ім'я
                                                  'FILE.TXT' }

  Rewrite (t);
  Writeln ('№ ', Digsne);
  Writeln (t, '№ ', Digsne);
  While not eof(arch) do
  Begin
    Read(arch,a);
    Writeln(a.numb:2,' ',a.datainput:5:1);
    Writeln(t,a.numb:2,' ',a.datainput:5:1);

  End;
  Close (t);
End.

```

## 5.8. Виконати збереження файлу у поточну папку

5.9. Виконати компіляцію програми.

5.10. Після усунення всіх виявлених помилок потрібно повернутися в основне меню і для запуску програми вибрати команду **Run→Run**.

5.11. Результат роботи програми можна побачити, натиснувши комбінацію клавіш: **Alt+F5** та проглянувши текстовий файл з розширенням **.txt**, який був створений у результаті роботи програми.

## **6. Контрольні питання**

1. Поняття алгоритму.
2. Графічні схеми алгоритмів.
3. Програмна реалізація алгоритмів.
4. Структурне програмування.
5. Організація циклів у програмах.
6. Структуроване зберігання інформації у базах даних.
7. Використання складних змінних типу “record” для організації бази даних.

## **ЛАБОРАТОРНА РОБОТА №2 РЕАЛІЗАЦІЯ АЛГОРИТМІВ ДОПУСКОВОГО КОНТРОЛЮ**

### **1. Мета роботи**

Ознайомлення з алгоритмами допускового контролю технологічних параметрів. Придбання навичок програмної реалізації розроблених алгоритмів.

### **2. Методичні вказівки до організації самостійної роботи студентів**

При підготовці до даної роботи необхідно повторити наступний матеріал:

- Правила складання блок - схем і опису алгоритмів.
- Алгоритм і програма формування й імітації вхідного дискретного сигналу (лабораторна робота №1).
- Алгоритм і програма ведення історії технологічного процесу (лабораторна робота №1).
- Правила програмування з використанням алгоритмічної мови ПАСКАЛЬ.

Необхідно ретельно розібратись з алгоритмами допускового контролю технологічних процесів.

### **3. Теоретичні відомості**

Інформація про поточний стан об'єкта керування надходить в ЕОМ по багатьох інформаційно-вимірювальних каналах (ІВК). Із збільшенням їх кількості, зростає ризик потрапляння в систему невірної інформації, тому однією з найважливіших функцій первинної обробки інформації в АСКТП є контроль її вірогідності.

Невірною інформація з'являється в разі відмов ІВК, які поділяються на повні та часткові (метрологічні). Повна відмова настає при виході з ладу вимірювального приладу (ВП), або в разі пошкодження лінії зв'язку ВП з ЕОМ. При частковій відмові технічні засоби зберігають працездатність, але похибка вимірювання відповідного параметра перевищує допустиме значення.

Розглянемо алгоритми контролю вірогідності початкової інформації, які дають змогу знайти лише повну відмову ІВК.

#### **Алгоритм допускового контролю параметра.**

Алгоритм ґрунтується на тому, що при нормальній роботі об'єкта значення кожного з контролюючих технологічних параметрів  $x$ , не повинні виходити за деякі межі:

$$x_i' \leq x_i \leq x_i'' .$$

Відповідно при справному ІВК має бути обмежений також сигнал вимірюваної інформації  $y_i$ , який надходить в ЕОМ по цьому каналу:

$$y_i' \leq y_i \leq y_i'' . \quad (*)$$

Контроль вірогідності за цим алгоритмом полягає в перевірці виконання умови (\*) для кожного значення сигналу вимірюваної інформації, який надходить при черговому опитуванні ІВК.

Після введення початкових даних організується цикл перевірки ІВК, кожному з яких присвоюється свій номер  $i$ . При виконанні умови (\*) перевіряється умова закінчення роботи алгоритму: якщо  $i = m$ , де  $m$  – кількість контрольованих ІВК, робота алгоритму припиняється. Інакше номер ІВК збільшується на одиницю й цикл повторюється. Якщо при перевірці умова (\*) не виконується, то невірогідне значення  $y_i[n t_0]$  замінюється вірогідною оцінкою, у ролі якої може бути використане попереднє вірогідне значення цього параметра  $y_i[(n-1)t_0]$  або його значення, усереднене за деякий інтервал часу, що випереджає момент визначення відмови. Після цього формується повідомлення про відмову  $i$ -го ІВК і продовжується опитування каналів.

#### **Алгоритм допускового контролю швидкості зміни сигналу.**

Алгоритм ґрунтується на тому, що швидкість зміни кожного з контролюючих технологічних параметрів  $x_i$  обмежується умовою:

$$\left| \frac{dx_i}{dt} \right| \leq \omega_C \left| x_i^* - \bar{x}_i \right| ,$$

де  $\omega_C$  – частота зрізу функції  $x_i(t)$ ;  $\bar{x}_i$  – середнє значення цієї функції.

Відповідно має бути обмежена також швидкість зміни сигналу вимірюваної інформації:

$$\left| \frac{dy_i}{dt} \right| \leq v_i^* ,$$

де  $v_i^*$  – максимально можливе значення швидкості.

Контроль вірогідності за даним алгоритмом полягає в перевірці виконання умови (\*\*), при цьому оцінку похідної розраховують за формулою:

$$\frac{dy_i}{dt} \approx \frac{y_i[nt_0] - y_i[(n-1)t_0]}{t_0},$$

або, після переходу до відносного часу  $\tau = \frac{t}{t_0}$ , будемо мати

$$\frac{dy_i}{d\tau} \approx y_i[j] - y_i[j-1].$$

#### 4. Індивідуальні завдання до виконання роботи

Розробити алгоритм і скласти програму, що здійснює допусканий контроль вхідного дискретного сигналу. Метод і параметри допускного контролю задаються відповідно до таблиці варіантів (додаток №2). Для імітації дискретного вхідного сигналу використовувати програми й параметри відповідного варіанта лабораторної роботи №1.

#### 5. Порядок виконання і методичні вказівки до виконання роботи

5.1. Ознайомитися з основними методами допускного контролю параметрів.

5.2. Скласти блок - схему й опис алгоритму фільтрації вхідного сигналу від перешкод по методу і з параметрами, що відповідають заданому варіанту.

5.3. Скласти програму, що реалізує даний алгоритм.

5.4. Налаштувати програму допускного контролю технологічних параметрів у комплексі з програмами імітації вхідного дискретного сигналу й ведення файлу історії (лабораторна робота №1).

5.5. Значення вхідного дискретного сигналу після допускного контролю записуються на диск у файл історії. Для цього необхідно внести зміни в опис структурованої змінної StructureArch, внівши додаткові поля Sign і DataReal. Після цього змінна набуває вигляду:

StructureArch = record

Numb	: integer;	{ номер опитування }
DataInput	: real;	{ дійсне значення сигналу }
Sign	: char;	{ ознака невірогідного значення }
DataReal	: real;	{ вірогідне значення сигналу }

End;

У випадку надходження невірогідного сигналу в символному полі Sign проставляється умовний символ, а в поле DataReal заноситься його вірогідне значення. Вірогідне значення знаходять або підстановкою останнього вірогідного значення, або усередненням декількох вірогідних значень (відповідно до варіанта завдання). У випадку надходження вірогідного значення, воно без змін переноситься в поле DataReal. Після чого вся структурована змінна записується на диск у файл історії.

5.6. Після запису інформації на диск у файл історії, формується його текстова копія (див. лабораторну роботу №1).

### **6. Контрольні питання.**

1. Квантування інформації.
2. Поняття про решітчасті функції й різницеві рівняння.
3. Види відказів інформаційне - вимірювальних каналів.
4. Відносний час у дискретних системах.
5. Методи допускового контролю вірогідності.
6. Інші методи контролю вірогідності.



# ЛАБОРАТОРНА РОБОТА №3 РЕАЛІЗАЦІЯ АЛГОРИТМІВ ФІЛЬТРАЦІЇ ВИМІРЮВАЛЬНИХ ВЕЛИЧИН

## 1. Мета роботи

Ознайомлення з алгоритмами фільтрації вимірюваних величин від перешкод, придбання навичок програмної реалізації алгоритмів.

## 2. Методичні вказівки по організації самостійної роботи студентів

При підготовці до даної роботи необхідно повторити наступний матеріал:

- Правила складання блок - схем і опису алгоритмів.
- Алгоритм і програма формування й імітації вхідного дискретного сигналу (лабораторна робота №1).
- Алгоритм і програма допускового контролю вхідного дискретного сигналу (лабораторна робота №2).
- Алгоритм і програма ведення історії технологічного процесу (лабораторна робота № 1).
- Правила програмування з використанням алгоритмічної мови ПАСКАЛЬ .

Необхідно ретельно розібратись з методами фільтрації вхідних параметрів від перешкод.

## 3. Теоретичні відомості

Крім контрольованих впливів на вимірюваний сигнал накладається ціла низка неконтрольованих випадкових перешкод.

$$g(t) = y(t) + e(t),$$

де  $g(t)$  – вимірюваний сигнал,  $y(t)$  – корисний сигнал,  $e(t)$  – випадкова перешкода.

У загальному випадку під фільтрацією розуміють вилучення корисного сигналу вимірюваної інформації  $y(t)$  із його суми з перешкодою  $e(t)$ .

Звичайно методи фільтрації побудовані на відміні частотних спектрів функцій  $y(t)$  та  $e(t)$ . Як правило, перешкода є більш високочастотною.

Прийемо наступні допущення:

1) Функція  $y(t)$  є стаціонарним випадковим процесом з відомими статистичними характеристиками:

- математичним сподіванням  $M_y$  ;
- дисперсією;
- автокореляційною функцією

$$R_y(\tau) = D_y e^{-\alpha|\tau|}, \quad \text{де } \alpha = \text{const.}$$

2) Перешкода  $e(t)$  також є стаціонарним випадковим процесом, некорельованим з корисним сигналом ; для неї відомі статистичні характеристики:

- математичне сподіванням  $M_e = 0$  ;
- дисперсією  $D_e = kD_y$  ;
- автокореляційною функцією

$$R_e(\tau) = kD_y e^{-m\alpha|\tau|}, \quad \text{де } k, m = \text{const} \quad k < 1, m > 1.$$

3) В результаті фільтрації одержуємо оцінку  $y_\phi(t)$  сигналу вимірювальної інформації, до якої пред'являють наступні вимоги:

- вона повинна бути незміщеною, тобто повинна задовольняти умові

$$M\{y_\phi(t)\} = M_y ;$$

- середньоквадратична погрішність оцінки повинна бути мінімальний, тобто

$$M\{[y_\phi(t) - y(t)]^2\} \rightarrow \min .$$

На практиці найбільше поширення одержали декілька відносно простих алгоритмів фільтрації.

### 3.1. Експоненціальний фільтр

В аналоговому варіанті експоненціальний фільтр являє собою аперіодичну ланку й описується диференціальним рівнянням

$$\frac{1}{\gamma} \frac{dy_{\phi}}{dt} + y_{\phi}(t) = k_{\phi} g(i),$$

де  $\gamma$  і  $k_{\phi}$  – параметри настроювання фільтру.

$$\gamma = \frac{1}{T_{\phi}}, \text{ де } T_{\phi} \text{ – постійна часу фільтру.}$$

Оптимальне значення  $\gamma$  одержують з умови мінімуму дисперсії погрішності експоненціального фільтру. На практиці такі розрахунки проводяться не завжди й у якості оптимального вибирають найбільше припустиме значення параметра  $\gamma$ .

Для статичного режиму оптимальне значення коефіцієнта підсилення  $k_{\phi}$  дорівнює

$$k_{\phi}^{\circ} = 1.$$

При програмній реалізації експоненціального фільтру його диференціальне рівняння заміняють різницеvim рівнянням вигляду

$$\frac{1}{\gamma} [y_{\phi}(j) - y_{\phi}(j-1)] + y_{\phi}(j-1) = q(j),$$

де  $j$  – номер циклу розрахунку.

З нього одержуємо наступне рекурентне співвідношення для обчислення згладженого значення  $y_{\phi}(j)$  в черговому  $j$ -тому циклі розрахунку:

$$y_{\phi}(j) = \gamma g(j) + (1 - \gamma) y_{\phi}(j-1).$$

До переваг алгоритму експоненціальної фільтрації відноситься мала трудомісткість розрахунків і малий обсяг пам'яті, у якій повинна зберігатися величина  $\gamma$  і обновлювана в кожному циклі розрахунку величина  $y_{\phi}(j-1)$ .

### 3.2. Фільтр змінного середнього

В аналоговому варіанті алгоритм реалізує обчислення середнього значення функції  $g(t)$  на інтервалі часу від  $(t - T_{\phi})$  до  $t$ :

$$y_{\phi}(j) = \frac{1}{T_{\phi}} \int_{t-T_{\phi}}^t g(\theta) d\theta,$$

де  $T_{\phi}$  – параметр настроювання фільтру (час усереднення).

При програмній реалізації фільтру змінного середнього розрахунок згладженого значення  $y_{\phi}(j)$  в черговому  $j$ -тому циклі розрахунку проводиться за формулою:

$$y_{\phi}(j) = \frac{1}{N+1} \sum_{k=0}^N g(j-k),$$

де  $N = \frac{T_{\phi}}{t_0}$  – параметр настроювання фільтру.

Для розрахунку по цій формулі потрібно зберігати в пам'яті  $(N+1)$  значень функції  $g(j)$ .

### 3.3. Статистичні фільтри

Статистичними називають фільтри, що в аналоговому варіанті являють собою паралельне з'єднання  $(N+1)$  ланцюжків, що складаються з підсилювальної ланки й ланки чистого запізнювання. Передаточна функція такого фільтру має вигляд:

$$W_{\phi}(s) = \sum_{j=0}^N b_j e^{-js\tau}, \quad b_j = \text{const}, \quad j = 0, 1, \dots, N.$$

Статистичний фільтр нульового порядку. Це найпростіший серед фільтрів даної групи. Його передаточна функція виходить із загальної формули при  $N = 0$ , тобто це просто підсилювальна ланка, вихідний сигнал якої дорівнює:

$$y_{\phi}(j) = b_0 g(t).$$

При безпосереднім використанні даної формули, згладжена функція  $y_{\phi}(t)$  буде зміщеною оцінкою корисного сигналу  $y(t)$ . Для одержання незміщеної оцінки до правої частини необхідно додати постійний член  $a$ , що задовольняє умові:

$$b_0 m_y + a = m_y,$$

звідки

$$a = m_y(1 - b_0).$$

Таким чином, первісна формула набуває вигляду:

$$y_\phi(t) = b_0 g(t) + m_y(1 - b_0),$$

де  $b_0$  – параметр настроювання фільтру.

При програмній реалізації статистичного фільтру нульового порядку розрахунок згладжених значень робиться за формулою:

$$y_\phi(j) = b_0 g(j) + m_y(1 - b_0).$$

Статистичний фільтр першого порядку. Його передаточну функцію одержують із загальної формули при  $N = 1$ :

$$W_\phi(s) = b_0 + b_1 e^{-s\tau},$$

де  $b_0, b_1, \tau$  – константи.

В часовій області рівняння цього фільтру має вигляд:

$$y_\phi(t) = b_0 g(t) + b_1 g(1 - \tau).$$

Для виконання умови незміщеності оцінки  $y_\phi(t)$ , коефіцієнти  $b_0$  і  $b_1$  повинні задовольняти співвідношенню  $b_1 = 1 - b_0$ . З урахуванням цього, формула приводиться до вигляду:

$$y_\phi(t) = b_0 g(t) + (1 - b_0)g(1 - \tau),$$

де  $b_0$  і  $\tau$  – параметри настроювання статистичного фільтру першого порядку.

#### 4. Індивідуальні завдання до виконання роботи

Розробити алгоритм і скласти програму, що здійснює фільтрацію від перешкод вхідного дискретного сигналу. Метод фільтрації й параметри фільтру задаються відповідно до таблиці варіантів (додаток №3). Побудувати графік вірогідного вхідного дискретного сигналу до і після фільтрації. Для імітації вірогідного вхідного сигналу використовувати програми й параметри відповідних варіантів лабораторних робіт №1 і №2.

## 5. Порядок виконання і методичні вказівки до виконання роботи

5.1. Ознайомитися з теоретичними основами методів фільтрації вимірюваних величин від перешкод.

5.2. Скласти блок - схему й опис алгоритму фільтрації вхідного сигналу від перешкод по методу і з параметрами, що відповідають заданому варіанту.

5.3. Скласти програму, що реалізує даний алгоритм.

5.4. Налагодити програму фільтрації у комплексі з програмами імітації вхідного дискретного сигналу (лабораторна робота №1), допускового контролю (лабораторна робота №2) і ведення файлу історії (лабораторна робота №1). У якості вхідного дискретного сигналу параметричного фільтру використовувати значення, що генерує імітатором дискретного сигналу й які пройшли допусковий контроль.

5.5. Значення вхідного дискретного сигналу після фільтрації записуються на диск у файл історії. Для цього необхідно внести зміни в опис структурованої змінної StructureArch, внівши додаткове поле DataFilter. Після цього змінна набуває вигляду:

StructureArch = record

Numb	:	integer;	{ номер опитування }
DataInput	:	real;	{ дійсне значення сигналу }
Sign	:	char;	{ ознака невірнісного значення }
DataReal	:	real;	{ вірогідне значення сигналу }
DataFilter	:	real;	{ фільтроване значення сигналу }

End;

Фільтроване значення вірогідного вхідного сигналу надається полю **DataFilter**, після чого вся структурована змінна записується на диск у файл історії.

5.6. Після запису інформації на диск у файл історії, формується його текстова копія (див. лабораторну роботу №1).

## 6. Контрольні питання

1. Що називають фільтрацію.

2. На яких властивостях вхідних сигналів побудовані методи фільтрації.
3. Які припущення приймають при застосуванні методів фільтрації.
4. Метод експоненціальної фільтрації.
5. Фільтрація за методом змінного середнього.
6. Статистичні фільтри нульового й першого порядків.

## **ЛАБОРАТОРНА РОБОТА №4 РЕАЛІЗАЦІЯ АЛГОРИТМІВ ОБЧИСЛЕННЯ УЗАГАЛЬНЕНИХ ПОКАЗНИКІВ ПРОЦЕСУ**

### **1. Мета роботи**

Ознайомлення з алгоритмами обчислення узагальнених показників процесу: обчислення інтегральних і усереднених значень вимірюваних величин; придбання навичок програмної реалізації алгоритмів.

### **2. Методичні вказівки по організації самостійної роботи студентів**

При підготовці до даної роботи необхідно повторити наступний матеріал:

- Правила складання блок - схем і опису алгоритмів.
- Алгоритм і програма формування й імітації вхідного дискретного сигналу (лабораторна робота №1).
- Алгоритм і програма допускового контролю вхідного дискретного сигналу (лабораторна робота №2).
- Алгоритм і програма фільтрації вхідного дискретного сигналу (лабораторна робота №3).
- Алгоритм і програма ведення історії технологічного процесу (лабораторна робота № 1).
- Правила програмування з використанням алгоритмічної мови ПАСКАЛЬ.

Необхідно ретельно розібратись з методами обчислення узагальнених показників процесу.

### **3. Теоретичні відомості**

Основою для оцінки якості функціонування виробництва в цілому є техніко-економічні показники. Розрахунок цих показників – це важлива задача, розв'язання якої дає інформацію для прийняття рішень по керуванню виробництвом. З формальної точки зору, техніко-економічними показниками є функції двох або більше змінних. До того ж, особливістю таких функцій є те, що вони розраховуються за певний проміжок часу. Задача вимагає обчислення



інтегралів від поточного значення вимірюваних величин на заданому інтервалі часу, а також усереднених значень показників.

Обчислення інтеграла від поточного значення вимірюваної величини звичайно зв'язано з визначенням сумарної кількості деякої речовини чи енергії, що надходять у об'єкт чи виходять з нього за визначений заданий інтервал часу  $T$ . При цьому вимірюваною величиною  $x(t)$  є витрата (потужність) відповідного потоку, а шукана (сумарна) величина  $S_x$  визначається співвідношенням:

$$S_x = \int_0^T x(t) dt.$$

До обчислення величини  $S_x(T)$  зводиться також задача розрахунку середнього значення  $\bar{x}$  вимірюваної величини  $x(t)$  на інтервалі часу  $0 \leq t \leq T$ :

$$\bar{x}(T) = \frac{1}{T} \int_0^T x(t) dt = \frac{1}{T} S_x(T).$$

Особливість обчислення величини  $S_x(T)$  у даному випадку зв'язана з тим, що вихідні значення представлені у вигляді дискретних значень функції  $x(t)$ . Тому необхідно застосовувати методи дискретного інтегрування й усереднення.

При обчисленні інтеграла найчастіше використовують методи прямокутників і трапецій.

### 3.1. Метод прямокутників

Метод заснований на апроксимації безперервної функції  $x(t)$  кусочно-сталою функцією. При цьому оцінку величини  $S_x(T)$  з урахуванням погрішностей виміру визначають за формулою:

$$\tilde{S}_n(T) = t_0 \sum_{j=0}^{n-1} \tilde{x}_j,$$

де  $\tilde{x}_j = x_j + \Delta x_j$ ;  $x_j = x(jt_0)$  – значення параметра  $x$  у момент  $t_j = jt_0$ ,  $j \in \overline{0, n-1}$ ;  $\Delta x$  – погрішність виміру параметра  $x$  у момент  $t_j$ ;  $n = \frac{T}{t_0}$  – число періодів відліку величини  $x(t)$  на відрізку часу інтегрування  $T$ .

Рекурентна формула для обчислення оцінки  $\tilde{S}_n$  в темпі з надходженням значень  $\tilde{x}_j$  має вигляд:

$$\tilde{S}_n(j) = \tilde{S}_n(j-1) + \tilde{x}_j t_0, \quad 0 \leq j \leq n-1, \quad \tilde{S}_n(0) = 0.$$

### 3.2. Метод трапецій

Метод заснований на апроксимації безперервної функції  $x(t)$  кусочно-лінійною функцією. При цьому оцінку величини  $S_x(T)$  по цьому методу розраховують по формулі:

$$\tilde{S}_m(T) = t_0 \left( \sum_{j=1}^{n-1} \tilde{x}_j + \frac{\tilde{x}_0 + \tilde{x}_n}{2} \right).$$

У рекурентній формі алгоритм обчислення оцінки  $\tilde{S}_m$  в темпі з надходженням вимірюваних значень  $\tilde{x}_j$  описується формулою:

$$\tilde{S}_m(j) = \tilde{S}_m(j-1) + t_0 \left( \frac{\tilde{x}_{j-1} + \tilde{x}_j}{2} \right); \quad 1 \leq j \leq n, \quad \tilde{S}_m(0) = 0.$$

### 3.3. Розрахунок середнього значення

Розрахунок середнього значення вимірюваного параметра  $\bar{x}(t)$  на заданому інтервалі  $0 \leq t \leq T$ , зводиться до обчислення інтегрального значення  $S_x(T)$ , для чого може бути використаний кожний з описаних вище методів чисельного інтегрування:

$$\bar{x}(T) = \frac{1}{T} S_x(T).$$

В рекурентної формі:

$$\bar{x}(j) = \frac{1}{j} \tilde{S}(j).$$

## 4. Індивідуальні завдання до виконання роботи

Розробити алгоритм і скласти програму, що здійснює розрахунок інтегральної й усередненої оцінки вхідного дискретного сигналу. Метод інтегрування параметра задається викладачем. Для

імітації достовірного вхідного сигналу використовувати програми й параметри відповідних варіантів лабораторних робіт №1, №2 і №3.

## 5. Порядок виконання і методичні вказівки до виконання роботи

5.1. Ознайомитися з теоретичними основами методів чисельного інтегрування вимірюваних величин.

5.2. Скласти блок - схему й опис алгоритму інтегрування й усереднення вхідного сигналу відповідно до заданого варіанта.

5.3. Скласти програму, що реалізує даний алгоритм.

5.4. Налаштувати програму інтегрування й усереднення в комплексі з програмами імітації вхідного дискретного сигналу (лабораторна робота №1), допускового контролю (лабораторна робота №2), фільтрації (лабораторна робота №3) і ведення файлу історії (лабораторна робота №1). У якості вхідного дискретного сигналу параметричного фільтру використовувати значення, що генерує імітатор дискретного сигналу й які пройшли допусковий контроль і фільтрацію.

5.5. Інтегральна й усереднена оцінки вхідного дискретного сигналу записуються на диск у файл історії. Для цього необхідно внести зміни в опис структурованої змінної StructureArch, внівши додаткові поля IntegralData і AverageData. Після цього змінна набуде вигляду:

```
StructureArch = record
Numb          : integer;   { номер опитування }
DataInput     : real;      { дійсне значення сигналу }
Sign          : char;      { ознака невірогідного значення }
DataReal      : real;      { вірогідне значення сигналу }
DataFilter    : real;      { фільтроване значення сигналу }
IntegralData  : real;      { інтегральне значення сигналу }
AverageData   : real;      { усереднене значення сигналу }
End;
```

Розраховані оцінки вхідного дискретного сигналу надаються відповідним полям, після чого вся структурована змінна записується на диск у файл історії.

5.6. Після запису інформації на диск у файл історії, формується його текстова копія (див. лабораторну роботу №1).

## **6. Контрольні питання**

1. Прикладні функції АСКТП.
2. Математичне забезпечення АСКТП.
3. Розрахунки узагальнюючих показників.
4. Методи дискретного інтегрування.
5. Розрахунок усереднених значень.

## Додаток 1

<b>№ вар</b>	<b>Діапазон вхідного сигналу</b>	<b>Початкове значення</b>	<b>Циклів опитування</b>
1	0 ... 5	1	30
2	0 ... 20	5	35
3	4 ... 20	8	40
4	0 ... 5	2	45
5	0 ... 20	10	50
6	4 ... 20	12	30
7	0 ... 5	3	35
8	0 ... 20	15	40
9	4 ... 20	16	45
10	0 ... 5	4	50
11	0 ... 20	5	30
12	4 ... 20	8	35
13	0 ... 5	1	40
14	0 ... 20	10	45
15	4 ... 20	12	50
16	0 ... 5	2	30
17	0 ... 20	15	35
18	4 ... 20	16	40
19	0 ... 5	3	45
20	0 ... 20	5	50
21	4 ... 20	8	30
22	0 ... 5	4	35

23	0 ... 20	10	40
24	4 ... 20	12	45
25	0 ... 5	1	50
26	0 ... 20	15	30
27	4 ... 20	16	35
28	0 ... 5	2	40
29	0 ... 20	5	45
30	4 ... 20	8	50

## Додаток 2

<b>№ вар</b>	<b>Діапазон вірогідності</b>	<b>Допустима швидкість (% від діапазону)</b>	<b>Кількість значень для усереднення</b>
1	0 ... 3		1
2		5	1
3	4 ... 12		1
4		7	2
5	5 ... 15		2
6		10	2
7	2 ... 5		3
8		10	3
9	10 ... 18		3
10		7	1
11	2 ... 12		1
12		5	1
13	0 ... 4		2
14		5	2
15	8 ... 16		2
16		7	3
17	10 ... 18		3
18		10	3
19	2 ... 5		1
20		10	1

21	4 ... 12		1
22		7	2
23	5 ... 15		2
24		5	2
25	0 ... 3		3
26		5	3
27	10 ... 20		3
28		7	1
29	2 ... 10		1
30		10	1



Метод фільтрації	Рівняння фільтра	№ вар	Параметри фільтра
Експоненціальний фільтр	$y_{\phi}(j) = \gamma g(j) + (1 - \gamma)y_{\phi}(j-1)$	1	$\gamma = 0.3$
		2	$\gamma = 0.4$
		3	$\gamma = 0.5$
		4	$\gamma = 0.6$
		5	$\gamma = 0.7$
		6	$\gamma = 0.8$
Фільтр змінного середнього	$y_{\phi} = \frac{1}{N+1} \sum_{k=0}^N g(j-k)$	7	$N = 1$
		8	$N = 2$
		9	$N = 3$
Статистичний фільтр нульового порядку	$y_{\phi}(j) = b_0 g(j) + (1 - b_0)m_y$ , де $m_y$ - середнє значення діапазону вірогідності.	10	$b_0 = 0.2$
		11	$b_0 = 0.3$
		12	$b_0 = 0.4$
		13	$b_0 = 0.5$
		14	$b_0 = 0.6$
		15	$b_0 = 0.7$
		16	$b_0 = 0.8$
Статистичний фільтр першого порядку	$y_{\phi}(j) = b_0 g(j) + (1 - b_0)g(j - \tau)$	17	$b_0 = 0.2 \quad \tau = 1$
		18	$b_0 = 0.4 \quad \tau = 1$
		19	$b_0 = 0.5 \quad \tau = 1$
		20	$b_0 = 0.6 \quad \tau = 1$
		21	$b_0 = 0.8 \quad \tau = 1$
		22	$b_0 = 0.2 \quad \tau = 2$

		23	$b_0 = 0.4 \quad \tau = 2$
		24	$b_0 = 0.5 \quad \tau = 2$
		25	$b_0 = 0.6 \quad \tau = 2$
		26	$b_0 = 0.8 \quad \tau = 2$
		27	$b_0 = 0.2 \quad \tau = 3$
		28	$b_0 = 0.4 \quad \tau = 3$
		29	$b_0 = 0.5 \quad \tau = 3$
		30	$b_0 = 0.6 \quad \tau = 3$
		31	$b_0 = 0.8 \quad \tau = 3$

## СПИСОК ЛІТЕРАТУРИ

1. Стенцель Й.І. Автоматизація технологічних процесів хімічних виробництв: Навч. посібник. -К.ПСДО, 1995. – 360с.
2. Единая система программной документации. Схемы алгоритмов и программ. Правила выполнения. ГОСТ 19.002-80.
3. Единая система программной документации. Схемы алгоритмов и программ. Обозначения условные графические. 19.003-80.
4. Боон К. ПАСКАЛЬ для всех: Пер. с гол. – М.: Энергоатомиздат, 1988. – 190 с.
5. Джексон Г. Проектирование реляционных баз данных для использования с микроЭВМ: Пер. с англ. – М.: Мир, 1991. – 252 с.

