

**МІНІСТЕРСТВО ОСВІТИ, НАУКИ, МОЛОДІ ТА СПОРТУ  
УКРАЇНИ**

**Запорізький національний технічний університет**

**МЕТОДИЧНІ ВКАЗІВКИ**

**до лабораторних робіт з дисципліни**

**Спеціальність 141 – Електроенергетика, електротехніка та електромеханіка, спеціалізація (освітня програма) - Електричні та електронні апарати; Електромеханічне обладнання енергосмних виробництв**

**2018**

Методичні вказівки до лабораторних робіт з дисципліни " **Методологія та інформатизація наукових досліджень електромеханічних пристроїв та систем.** для студентів усіх форм навчання спеціальності **141 – Електроенергетика, електротехніка та електромеханіка, спеціалізація (освітня програма) - Електричні та електронні апарати; Електромеханічне обладнання енергоємних виробництв** / Укл.: М.О. Поляков - Запоріжжя: ЗНТУ, 2018. - 28 с.

Укладач:

М.О. Поляков, доцент, к.т.н.

Рецензент:

В.М. Снігірьов, доцент, к.т.н.

Відповідальний за випуск:

О.В. Близняков, доцент, к.т.н.

Затверджено  
на засіданні кафедри  
«Електричні апарати»  
Протокол № 1  
від 21 серпня 2018 р

**ЗМІСТ**

Вступ .....	4
Лабораторна робота № 1 .....	4
Лабораторна робота № 2 .....	6
Лабораторна робота № 3 .....	7
Лабораторна робота № 4 .....	8
Лабораторна робота № 5 .....	11
Лабораторна робота № 6 .....	12
Лабораторна робота № 7 .....	14
Лабораторна робота № 8 .....	17
Лабораторна робота № 9 .....	19
Лабораторна робота № 10 .....	20
Лабораторна робота № 11 .....	23
Лабораторна робота № 12.....	26
Перелік посилань .....	27

## ВСТУП

У даному курсі лабораторних робіт вивчаються методи та засоби автоматизації наукових досліджень, такі як АЦП та фільтрація даних, нечіткий контролер, навчання нейронної мережи, кластерний аналіз, генетичні алгоритми, вейвлет аналіз, логічне програмування, комп'ютерне моделювання, пакети програмування промислових контролерів, програмування задач досліджень, пакети програмування SCADA системи та АСНД з видаленою лабораторією GOLDi

Використання комп'ютера та інформаційних технологій - це обов'язкова кваліфікаційна вимога до магістра, який виконує наукові дослідження у галузі електромеханічних пристроїв та систем. Тому якісне, творче виконання запропонованих лабораторних робіт буде сприяти закріпленню теоретичного матеріалу дисципліни.

Для успішного і своєчасного виконання кожної лабораторної роботи студент повинний якісно до неї підготуватись поза розкладом лабораторних занять: вивчити теоретичний матеріал [1-15], ознайомитись з методичними вказівками відповідно до теми лабораторної роботи, розробити тексти програми тощо. Після виконання роботи, студент повинен погодити з викладачем її результати, надати та захистити звіт, оформлений відповідно до СТП 1596.

Перед виконанням циклу лабораторних робіт студенти повинні пройти інструктаж з техніки безпеки і внутрішнього розпорядку комп'ютерного класу.

## ЛАБОРАТОРНА РОБОТА № 1

### АЦП та фільтрація даних

**Мета роботи:** навчитись вводити зібрані в результаті дослідження данні в комп'ютер и обробляти їх.

### Теоретичні відомості

Технологія введення зібраних даних в керуючий комп'ютер і їх обробки описана в [2, глава5]. Ключовими етапами обробки сигналів є їх цифрування (дискретизація, квантування) и цифрова фільтрація.

Інтервал дискретизації аналогового сигналу має бути достатнім для послідууючого встановлення його форми. Частота вибірки  $f_s$  - величина зворотна інтервалу дискретизації. Якщо  $f_s < 2f$ , де  $f$  - частота сигналу що досліджується, то форму цього сигналу неможливо встановити. Гранична частота дискретизації  $f_n = 2f$  зветься частотою Найквіста або Котельникова. Якщо аналоговий сигнал містить будь які частоти  $f$ , що перебільшують  $f_n/2$ , то ці високочастотні компоненти з'являються в послідовності даних вибірки як псевдочастоти  $f_0 = f_s - f$ .

### Практична частина

1. Дослідити дискретизацію синусоїдального сигналу с частотою  $f$ , вибірками с частотою  $f_s$  від  $1f$  до  $2f$ . Побудувати графік  $f_0 / f_s = \varphi(f / f_s)$ .

2. Дослідити дискретизацію синусоїдального сигналу с частотою  $f = 0,01f_c$  на фоні поміх с частотою  $f_c$  (наприклад,  $f_c = 50$ Гц), використовуючи вибірки с частотою  $f_s = 1,2f_c$ . Побудувати графіки залежності від часу сигналу разом з поміхами мережі, встановленого сигналу разом з  $f_0$ . Визначити  $f_0$  в встановленому сигналі.

3. Дослідити цифровий причинний фільтр низької частоти ковзного середнього сигналу. В якості сигналів, що вимірюються, обрати суміш шуму та синусоїдального сигналу або послідовності прямокутних імпульсів. Дослідити якість фільтрації при різній кількості вибірок що усереднюються. Привести осцилограми досліджень.

4. Дослідити цифровий експоненціальний фільтр низької частоти першого порядку. В якості сигналів, що вимірюються, обрати суміш шуму та послідовності прямокутних імпульсів. Дослідити якість фільтрації при різному значенні вагового коефіцієнту  $\alpha$ .

### Методичні вказівки

Дослідження процесів дискретизації та фільтрації сигналів проводити у середовищі Simulink.

### Контрольні запитання

1. Етапи введення зібраних даних в керуючий комп'ютер і їх обробки.
2. Вимоги до інтервалу дискретизації.
3. Що таке частота Найквіста.
4. Які фільтри звуться Moving Average - MA, AutoRegressive – AR, ARMA, causal, non-causal, exponential smoothing.
5. Операції обробки вимірюваної інформації.

## ЛАБОРАТОРНА РОБОТА № 2

### Нечіткий контролер

**Мета роботи:** вивчити принципи дослідження за допомогою нечітких контролерів

#### Теоретичні відомості

Нечіткий контролер використовують у випадках, коли неможливо створити модель об'єкту дослідження у формі диференціальних рівнянь. Тоді використовують досвід експертів, який є нечітким. Він втілюється в функції приналежності змінних моделі. Процес утворення нечіткого контролеру описано в [4].

#### Практична частина

1. Дослідити використання нечіткого контролеру як альтернативу PID- контролеру за допомогою демонстраційного прикладу ‘Water Tank with View Ruler’ у пакеті Fuzzy Logic Toolbox системи Matlab.
2. Дослідити використання нечіткого контролеру як експертної системи прогнозування технічного стану електромеханічного пристрою.

#### Методичні вказівки

Для створення нечіткого контролеру використовувати редактори структури, функцій приналежності та правил у пакеті Fuzzy Logic Toolbox системи Matlab.

### Контрольні запитання

1. Що таке нечітка множина.
2. Що таке лінгвістична зміна.

3. Етапи створення нечіткого контролеру.
4. Переваги нечіткого контролеру.
5. Приклади використання нечіткої логіки.

## ЛАБОРАТОРНА РОБОТА № 3

### Навчання нейронної мережі

**Мета роботи:** вивчити принципи побудови моделі явища або процесу за допомогою навчання нейронної мережі.

### Теоретичні відомості

Штучні нейронні мережі використовуються для побудови моделі явища або процесу у випадках коли не має досліду експертів. Навчання нейронної мережі виконується шляхом введення результатів догляду роботи об'єкту дослідження у форматі «вхід – вихід».

Процес утворення нечіткої моделі шляхом навчання нейронної мережі описано в [4].

### Практична частина

1. Дослідити процес утворення нечіткої моделі математичної функції шляхом навчання нейронної мережі. Для цього :
  - Зробити за допомогою текстового редактору файл із розширенням .txt який містить 10-20 пар строк значень аргумент-функція та завантажити його у програму **anfisedit** системи Matlab.
  - Здобуту модель зберегти як файл із розширенням .fis, відкрити та виконати у пакеті **Fuzzy Logic Toolbox** системи Matlab.
  - Розрахувати похибку обчислення значення математичної функції за допомогою створеної нечіткої моделі. При необхідності змінити параметри утворення моделі у середовищі **anfisedit** та розрахувати похибку.
2. Дослідити процес утворення нечіткої моделі випадкового процесу у якому функція процесу залежить від її значень у попередні моменти часу [4].

### Методичні вказівки

Для побудови нечіткої моделі використовувати функцію **anfisedit** а для дослідження пакет **Fuzzy Logict Toolbox** системи Matlab.

### Контрольні запитання

1. Що таке штучний нейрон.
2. У чому переваги побудови нечіткої моделі шляхом навчання нейронної мережі.
3. Приклади використання моделей здобутих шляхом навчання нейронної мережі.
4. Як змінити параметри отриманої моделі у середовищі програми **anfisedit**.

## ЛАБОРАТОРНА РОБОТА № 4

### Кластерний аналіз даних

**Мета роботи:** навчитися розбивати данні досліджень на множину кластерів.

### Теоретичні відомості

Кластерний аналіз (або автоматична класифікація, розпізнавання образів без учителя, численна таксономія) займає одно з центральних міст серед методів аналізу даних и представляє собою сукупність підходів, методів і алгоритмів призначених для знаходження деякого розбиття сукупності об'єктів, що досліджуються, на підмножини відносно, схожих між собою об'єктів - кластерів [4, глава 13].

Задачею нечіткої кластеризації є знаходження нечіткого розбиття або нечіткого покриття множин елементів, що досліджуються, які створюють структуру нечітких кластерів, присутніх у даних що розглядаються.

Для рішення задачі нечіткої кластеризації може бути використана функція командної строки **fcm** системи Matlab, алгоритм якої заснований на методі нечітких середніх [4, глава 13] або спеціальний графічний інтерфейс кластеризації функції **findcluster**. Вхідними аргументами функції **fcm** є: **data** - матриця первинних даних; **cluster\_n** – число первинних кластерів. Вихідні аргументи функції **fcm**: **center** – матриця центрів нечітких кластерів які шукаються; **U** - матриця зна-



чень функції належності розбиття що шукається:  $\mathbf{obj}_{f_{cm}}$  – значення цільової функції на кожній з ітерацій роботи алгоритму. Додаткові аргументи функції  $f_{cm}$  приведені в [4, глава 13]. Для рішення задачі визначення числа кластерів використовується функція **[C,S]= subclust (x, radii, x Bounds, options)**, де: **X** – матриця даних кластеризації, **radii** – вектор, який задає діапазон розрахунку центрів кластерів по кожній з ознак вимірювань (оптимальні значення **radii** = 0,2 – 0,5); **xBounds** – матриця відображення даних X в деякому одиничному гіперкубі, яка містить мінімальні та максимальні значення інтервалу вимірювань по кожній з ознак; **options** – вектор додаткових параметрів алгоритму кластеризації.

Компоненти вектора **options**:

- **options (1)** – визначає округу центра кластера;
- **options (2)** – встановлює потенціал як частину потенціалу першого кластера, вище якої друга точка даних може використовуватися в якості центра іншого кластера;
- **options (3)** – встановлює потенціал як частину потенціалу першого кластера, нижче якої друга точка даних може використовуватися в якості центра іншого кластера;
- **options (4)** – признак виведення інформації про виконання процесу кластеризації на монітор.

По умовчанняю **options** (1,25 0,5 0,15 0).

### Практична частина

1. Виконати нечітку класифікацію даних які містяться в файлі **fcmdata.dat**, який постачається с системою Matlab. Ці данні представляють матриці 140\*2. Для рішення задачі використовувати послідовності команд наведених в пунктах 13.1 и 13.2 [4, глава 13]. В звіті привести графіки координат крапок даних і центрів n кластерів, що шукаються, на площині відповідного вікна системи Matlab..

2. Виконати постановку задачі та нечітку класифікацію з використанням функції **fcm** даних що задані таблицею в пакеті Excel. У звіті привести графіки координат крапок початкової сукупності та центрів шуканих кластерів.

3. Виконати нечітку класифікацію даних з використанням графічного інтерфейсу функції **findcluster**. При цьому загрузити вихідні дані з зовнішнього файлу **Matlab/toolbox/fuzzy/furdemos/**

**fcmdata.dat.** У звіті привести графіки координат крапок вихідної сукупності та центрів шуканих кластерів.

4. Визначити, використовуючи функцію **subclust** командної строки, кількість кластерів в вихідних даних пункту 3. Дослідити вплив параметрів цієї функції на результат визначення кількості кластерів.

5. Визначити кількість кластерів у вихідних даних пункту 3 за допомогою графічного інтерфейсу. Для чого запустити у командній строчці засіб **findcluster** з методом кластеризації **subtractive** у вікні **Methods**.

### Контрольні запитання

1. Що таке численна таксономія.
2. Що таке кластерний аналіз.
3. Назвіть задачу нечіткої кластеризації.
4. Назвіть засоби нечіткої кластеризації в пакеті Fuzzy Logic Toolbox системи Matlab.
5. Суть методу субтрактивної кластеризації.

## ЛАБОРАТОРНА РОБОТА № 5

### Генетичні алгоритми

**Мета роботи:** навчитися вирішувати задачі оптимізації багато-параметричних функцій з використанням механізмів, що нагадують біологічну еволюцію.

### Теоретичні відомості

Генетичний алгоритм (ГА) – це алгоритм, який дозволяє знайти задовільне рішення аналітично нерозв'язних або складно розв'язних проблем через послідовний перебір і комбінування вихідних параметрів з використанням механізмів, що нагадують біологічну еволюцію [5].

Генетичні алгоритми оперують з сукупністю особей (популяцією), які є собою строки и кодують одно з рішень задачі.

Задача оптимізації розглядається ГА як задача знаходження максимуму деякої функції пристосованості. За допомогою функції прис-

тосованості з усіх особей популяції обирають найбільш пристосовані (більш підходящі рішення), які отримують можливість схрещуватися та давати потомство. Найгірші особи (погані рішення) видаляються з популяції і не дають потомства. Таким чином, пристосованість нового покоління у середньому вище попереднього.

Шаг ГА складається з трьох стадій: відбору, схрещування та мутації. В класичному ГА ймовірність кожної особи попасти в проміжну популяцію пропорційно її пристосованості (пропорціональний відбір).

При схрещуванні, особи проміжної популяції розбиваються на пари які схрещуватимуться, тобто. за допомогою кроссоверу формуються особи с новими функціями пристосованості. Оператор мутації необхідний для "вибивання" популяції з локального екстремуму. Під час мутації кожний біт кожної особи популяції с деякою ймовірністю (як правило меншої ніж 0,01) інвертується.

Критерієм зупинки еволюції є задана кількість поколінь, під час яких кроссовер практично не змінює популяції.

Існують різні моделі ГА, що відрізняються способами реалізації етапів алгоритму [5].

Засоби для реалізації ГА є у системі Matlab.

### Практична частина

1. У пакеті Matlab увійти у засіб Genetic Algorithm and Direct Search, ознайомитися с розділами допомоги Introducing Genetic Algorithm and Direct Search, Get Started with Genetic Algorithm

2. Знайти мінімальне значення функції

$$f(x_1, x_2) = x_1^2 - 2x_1x_2 + 6x_1 + x_2^2 - 6x_2. \quad (5.1)$$

З цією метою за зразком прикладу створити m-файл с фітнес функцією (5.1) и функцією генетичного алгоритму **ga**. Запустити цій файл у командній строчці та записати знайдене мінімальне значення функції та змінних  $x_1$  и  $x_2$ .

3. Знайти мінімальне значення функції (5.1) засобом **optitool**. З цією метою набрати в командній строчці **type ps\_example**, проглянути m-файл, потім створити за зразком m-файл функції (5.1). Далі відкрити вікно **optitool**, задати в ньому ім'я створеного файлу (**Objective**

**function**), початкові значення вхідних змінних (**Startpoint**), запустити оптимізацію (**Start**). Після завершення заданого числа ітерацій записати знайдене мінімальне значення функції та змінних  $x_1$  и  $x_2$ .

### Контрольні запитання

1. Що таке генетичний алгоритм.
2. Де застосовуються генетичні алгоритми.
3. Що таке функція пристосованості.
4. Стадії шага генетичного алгоритму.
5. Що таке кроссовер у генетичному алгоритмі.
6. Які засоби для реалізації генетичних алгоритмів є в системі Matlab.

## ЛАБОРАТОРНА РОБОТА № 6

### Вейвлет аналіз

**Мета роботи:** вивчити принципи дослідження и порівняти можливості перетворювань Фур'є та вейвлет - перетворювання.

### Теоретичні відомості

Як правило, сигнали що досліджуються представлені як функції часу(у часовій області). У багатьох випадках корисним є знання спектра частот, які містяться у вихідному сигналі - представлення сигналу в частотній області. Для переходу від часового представлення до частотного використовується перетворювання Фур'є. Його недоліки виявляються при аналізі нестационарних сигналів, - таких сигналів в яких окремі гармоніки присутні не весь час його спостереження. При аналізі таких сигналів ефективно використовувати вейвлет - перетворювання [3].

### Практична частина

1. Дослідити обмеження перетворювання Фур'є. Для чого в середовищі Simulink зібрати генератори сигналів:

$$x_1(t) = \cos(2\pi 10t) + \cos(2\pi 25t) + \cos(2\pi 50t) + \cos(2\pi 100t), \quad (6.1)$$

$$x_2(t) = \begin{cases} \cos(2\pi 10t), & 0 \leq t \leq 300 \text{ мс} \\ \cos(2\pi 25t), & 300t \leq 600 \text{ мс} \\ \cos(2\pi 50t), & 600 < t \leq 900 \text{ мс} \\ \cos(2\pi 100t), & 900 < t \leq 1200 \text{ мс} \end{cases} \quad (6.2)$$

2. Виконати моделювання на інтервалі [0,1200 мс] и отримати спектри сигналів  $x_1$  и  $x_2$  з використанням перетворювання Фур'є та блока FFT (Simulink/DSPBlockset/Estimation/PowerSpectrum Estimation). Графіки  $x_1(t)$  та  $x_2(t)$ , а також спектри сигналів відобразити у звіті.

3. Дослідити можливості вейвлет - перетворювання. Для чого використати сигнали  $x_1$  и  $x_2$  які задані рівняннями (6.1) и (6.2) та заціб Wavelet з пакету Toolboxes Matlab.

### Контрольні запитання

1. Переваги представлення сигналів у часовій та частотній області.
2. Комп'ютерні засоби для Фур'є та вейвлет - перетворювання.
3. Переваги вейвлет - перетворювання порівняно з перетворюванням Фур'є.
4. Поняття скейлет функції и материнського вейвлету.
5. Методика аналізу сигналів с використанням вейвлету Хаара.
6. Методика кратномасштабного аналізу сигналів з використанням вейвлетів с кратністю  $n=3$ .

## ЛАБОРАТОРНА РОБОТА № 7

### Логічне програмування

**Мета роботи:** навчитися синтезувати знання з фактів за допомогою логічного програмування мовою ПРОЛОГ.

### Теоретичні відомості

Логічне програмування – це використання дедуктивних процедур (процедур логічного виводу) як механізму обчислень [5]. Для ав-

томатизації обчислень використовуються програми на языках логічного програмування, наприклад ПРОЛОГ [6,7]. У таких програмах формулюються відомості про задачу у вигляді логічних аксіом та припущення, достатні для її рішення. Сама задача формулюється як цільове ствердження, що підлягає доказу.

Програма мовою ПРОЛОГ складається з речень. Кожне речення закінчується крапкою. Речення бувають трьох типів: факти, правила та питання. Факти містять правила, які є завжди безумовно, вірними. Правила містять ствердження, істинність яких залежить від деяких вимог. За допомогою питань користувач запитує систему, які ствердження є істинними.

Можливості логічного програмування виконувати логічний вивід із створенням нового знання на основі статичного опису ситуації (стану деякій прикладної області реального мира) широко застосовується у різноманітних "експертних системах" - системах підтримки прийняття рішень.

### Практична частина

1. Застосування логічного виводу для аналізу логічних схем. Нехай задана логічна схема напівсуматору (рис.7.1).

Ця схема описується мовою ПРОЛОГ як ствердження

**half – add (A, B, S, C) ← xor (A, B, S), nand (A,B,T1), not**

**(T1,C).**

Елементарні функції **xor**, **nand**, **not** описуються за допомогою фактів, що повторюють відповідні таблиці істинності:

**xor (x, x, 0)**

**xor (0, 1, 1)**

**xor (1, 0, 1)**

**nand (0, 0, 1)**

**nand (0, 1, 1)**

**nand (1, 0, 1)**

**nand (1, 1, 0)**

**not (0, 1)**

**not (1, 0).**

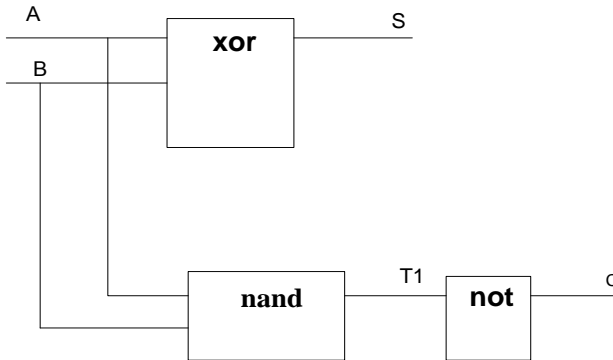


Рисунок 7.1 – Схема напівсуматору

Сформулювати мовою ПРОЛОГ питання:

- а) Які виходи S і C цієї схеми при входах 0,0?
- б) При яких входах, вихід S буде нульовим, а вихід C - одиничним?
- в) При яких входах схеми виходи S і C будуть збігатися?

Виконати ПРОЛОГ - програми с поставленими питаннями на комп'ютері. Тексти програм і відповіді привести в звіті.

2. Перевірити за допомогою логічного програмування чи є правильними слідуючи міркування:

"Якщо Джонс не зустрічав цією ніччю Сміта, то або Джонс був вбивцем, або Джонс брехун. Якщо Сміт не був вбивцем, то Джонс не зустрічав Сміта цією ніччю, і вбивство мало місце після полуночі. Якщо ж вбивство мало місце після полуночі, то або Сміт був вбивцем, або Джонс брехун. Отже, Сміт був вбивцем."

Тексти ПРОЛОГ - програми привести в звіті.

3. Перевірити за допомогою логічного програмування істинність стверджень:

- а) сепульки не хронічні тільки в случае відсутності у них властивості латентності;

б) латентність сепулєк не є не обхїдною вимогою їх хронїчностї або бїфуркальностї;

в) сепулькї бїфуркальнї тїльки в випадку їх хронїчностї або латентностї;

г) хронїчность сепулєк є достатньою вимогою їх латентностї або бїфуркальностї;

д) для того, щоб сепулькї були бїфуркальнї, достатньо тїльки щоби вони були хронїчнї;

е) для нехронїчностї сепулєк необхїдним є вїдсутнїсть у них як бїфуркальностї, так і латентностї.

Вїдомо, що "хронїчнї сепулькї завжди латентнї або бїфуркальнї".

Тексти ПРОЛОГ – програми привести в звїтї.

### Контрольнї запитання

1. Що таке логїчне програмування.
2. Структура програми мовою ПРОЛОГ.
3. Дайте визначення речень рїзних типїв, якї використовуються в ПРОЛОГ - програмах.
4. Трьох школярїв А, В и С було викликано до директору. В бесїдї з директором А стверджував, що В брехун, В стверджував, що бреше С, а С стверджував, що обидва А и В, брешуть. Що може заключить директор?
5. Методика аналізу логїчних схем с допомогою ПРОЛОГ - програм.
6. Синтез експертних систем с допомогою ПРОЛОГ – програм.

## ЛАБОРАТОРНА РОБОТА № 8

### Комп'ютерне моделювання

**Мета роботи:** навчитися проектувати та використовувати комп'ютерну модель системи що досліджується.

### Теоретичнї вїдомостї

Комп'ютерне моделювання – основний метод одержання інформації об об'єктї що досліджується при вїдсутностї реального об'єкта.



Модель системи досліджень включає підсистеми об'єкта дослідження, регулювання, зовнішнього середовища та обробки даних.

Вихідними даними для побудови підсистеми об'єкта досліджень є результати теоретичних досліджень в формі диференціальних рівнянь або передаточні функції елементів об'єкта та с в'язків між ними, що описують поведінку об'єкта; опит експертів, результати спостереження за реакціями на вихідні зовнішні дії аналогічного об'єкта.

Підсистема регулювання містить релейний, ПИД або нечіткий регулятори. Модель зовнішнього середовища представляє собою формул вачі залежних від часу зовнішніх дій на об'єкт дослідження. Підсистема обробки даних забезпечує розрахунок вторинних параметрів, візуалізацію та збереження результатів дослідження. В деяких випадках складовою частиною моделі об'єкта досліджень є моделі його дефектів и технічних станів.

Система досліджень може бути реалізована в програмному середовищі системи управління об'єктом як компонент програмного забезпечення програмованого логічного контролера, SCADA-системи або в універсальному пакеті моделювання, наприклад Simulink.

## Практична частина

1. Виконати аналіз вихідних даних для побудови системи досліджень теплових процесів в паперовій ізоляції обмоток у силовому ма-слонаповненому трансформаторі [8]. Трансформатор має одну ступень охолодження та працює в режимах ONAN, OFAF. Регулювання температури масла виконується релейним регулятором по значенням температури верхніх слоїв масла.

Для створення термодинамічної моделі трансформатора (ТДМТ) використати рівняння балансу енергій. Дефект системи охолодження, що приводить до зниження в процесі експлуатації трансформатора ефективності охолодження моделювати введенням залежності теплового опіру ділянки «трансформатор – зовнішнє середовище» від часу. Наявність дефектів приводить до відмов. Будемо розлічать катастрофічні и параметричні відмови системи охолодження. Для настройки ТДМТ використувати технічні характеристики трансформатора що заданий і результати його теплових випробувань.

Модель «зовнішнє середовище» різна на різних етапах дослідження. Зовнішні дії можуть бути незмінними синусоїдальними, імпульсними, випадковими, реальними даними, що отримані системою моніторингу параметрів трансформатора і др.

Підсистема обробки даних формується в відповідності до мети дослідження та може виконувати розрахунки температури найбільш нагрітої точки ізоляції, сумарного часу ввімкненого стану системи охолодження, відносної швидкості зносу ізоляції, витрат ресурсу ізоляції, затрат на функціонування системи охолодження за час дослідження та інші.

В якості середовища моделювання обрати засіб Simulink.

2. Виконати налагодження ТДМТ. Для чого зібрати цю модель у засобі Simulink, змодельовати зовнішні дії, при яких проводились теплові випробування, отримати в результаті моделювання залежність температури верхніх слоїв масла від часу та, шляхом підбору теплових параметрів моделі, добитися її максимально можливої відповідності результатам теплових випробувань у режимах ONAN, OFAF. Після виконання налагодження оцінити погрішність моделювання.

2. Виконати дослідження теплових процесів у трансформаторі у відсутності дефектів системи охолодження. З цією метою поєднати контур регулювання ТДМТ та релейний регулятор. Тепловий опір ділянки «трансформатор – зовнішнє середовище» задати константою, час моделювання роботи трансформатора – сутки. Параметри с уточненого коливання температури обрати по ДСТУ 14209-97 для найбільш жаркого дня року. Моделювання теплових процесів проводити при різних формах залежності струму навантаження від часу. В результаті досліджень встановити максимальну температуру верхніх слоїв масла, найбільш нагрітої точки ізоляції обмоток, сумарний час роботи системи охолодження в режимі OFAF, максимальну відносну швидкість термічного зносу ізоляції, витрату ресурсу ізоляції обмоток.

3. Дослідити теплові процеси в трансформаторі при наявності параметричних або катастрофічних дефектів в системі охолодження. Для чого, в модель системи, яка описана в пункті 2, вести дефект системи охолодження що виявляється в рості (лінійно або стрибком) починаючи з деякого моменту часу, значення теплового опору ділянки «трансформатор – зовнішнє середовище». Порівняти результати досліджень за пунктами 2 і 3.

### **Контрольні запитання**

1. Область застосування, переваги та недоліки метода дослідження шляхом комп'ютерного моделювання.
2. Основні елементи системи дослідження методом комп'ютерного моделювання.
3. Основні елементи системи дослідження теплових процесів у силовому маслонаповненому трансформаторі.
4. Методика налагодження ТДМТ.

### **ЛАБОРАТОРНА РОБОТА № 9**

#### **Пакети програмування промислових контролерів**

**Мета:** вивчити інтерфейс та основні види робіт у середовищі пакету програмування RSLogix при створенні додатку керування.

#### **Теоретичні відомості**

**Короткі теоретичні дані.** Методика та основні види робіт у середовищі пакету програмування RSLogix 500 описані у керівництві та в інтерактивній довідці до пакету.

Рекомендована література [2, 9-15).

#### **Практична частина**

1. Запустити пакети RSLogix500, RSLinx, RSEmulate 500.
2. Створити у пакеті RSLogix500 проект для процесора 1747-L542В вузла 1, драйвера EMUL500. У розділі I/O Configuration менеджера проекту додати до складу контролеру проекту один модуль введення та один модуль виведення.
3. Вивчити за допомогою довідки у меню HELP інструкції XIC та OTE.
4. Розробити та завантажити у емулятор контролера програму, яка містить інструкції XIC та OTE. Проглянути зміст файлів даних, які використовуються у програмі.
5. Виконати завантажену в емулятор програму для кількох значень даних. Записати здобуті результати.
6. Зберегти розроблений проект на диску комп'ютера.

### Контрольні запитання

1. Призначення та функціональні можливості пакетів RSLogix, RSLinx, RSEmulate.
2. Основні етапи створення проекту у пакеті RSLogix.
3. Як виконати конфігурування системи зв'язку з контролером.
4. Елементи менеджера проекту.
5. Як виконати конфігурування складу контролера.
6. Назвіть основні принципи введення у програму інструкцій, гілок та рангів.
7. Призначення команд XIC та OTE.

### ЛАБОРАТОРНА РОБОТА № 10

#### Програмування задач досліджень

**Мета:** Вивчити елементи мови програмування LD, інструкції контролера бітові, таймерів, лічильників та порівняння

#### Теоретичні відомості

Мова LD – це одна з п'яти стандартних мов програмування контролерів за стандартом IEC61131-3. LD програма складається з рангів. Ранг має умовну та виконавчу частини. Частини рангу складаються з гілок. Гілка, у свою чергу, має вхід, вихід та містить інструкцію контролера – умовну чи виконавчу. Дія умовної інструкції це перевірка істинності певної умови, а вихідної – у зміні значення програмних тегів. Кожна інструкція виконує певну функцію за якої її можна віднести до категорії базових, обробки даних тощо. Загальна кількість типів інструкцій від 60 до 150 в залежності від типу контролеру. Рекомендована література [2, 9-15].

Як правило, додатки керування є розподіленими, тобто містять частини, які виконуються у різних додатках, що виконуються на комп'ютері. Для обміну даними між контролером та іншими додатками існують різні способи, один з яких це механізм Dynamical Data Exchange (DDE). Для введення у додаток програми EXCEL значення змінної з контролера у клітинку листа EXCEL вводять формулу. Для виведення значення змінних з однієї або декількох клітинок листа EXCEL у контролер використовують VBA програму та

певну подію, наприклад подвійне натискування мишею на елемент керування на екрані EXCEL.

Рекомендована література [9,10].

## Практична частина

### 1. Вивчення інструкцій мови програмування LD

1.1. У середовищі програмування RSLogix500 використовуючи вбудовану допомогу (функціональне меню/Help/Instruction List) вивчити склад, умовні графічні позначення, призначення, параметри та прикладі використання бітових інструкцій, інструкцій таймерів, лічильників та порівняння.

1.2. Розробити та завантажити в емулятор контролеру програму, що реалізує логічну функцію  $y = \overline{(x_1 \wedge x_2)} \vee x_3$ , де  $x_1, x_2, x_3$  – вхідні змінні, що надходять до контролера через входи дискретного модуля введення;  $y$  – вихідна змінна, значення якої формується контролером на виході дискретного модуля виведення. Виконати програму для усіх комбінацій вхідних змінних. За результатами виконання скласти таблицю істинності логічної функції. Розробити блок-схему алгоритму програми, що реалізує логічну функцію.

1.3. Виконати попередній пункт щодо функції, яка задана викладачем.

Розробити та завантажити в емулятор контролеру програму що формує використовуючи інструкцію TON інтервали часу певної тривалості, підраховує кількість цих інтервалів за допомогою лічильнику STU, та виконує скидання лічильнику (інструкція RES ) у моменти часу, коли акумулятор лічильнику досягає певного значення. У звіті навести програму та графіки зміни основних змінних програми від часу.

1.4. Розробити та виконати за допомогою емулятору програму, що реалізує функції реверсивного лічильнику. Напряму зміни станів лічильнику задавати за допомогою вхідної змінної логічного типу. Виконати автоматичну зміну значення цієї змінної таким чином щоб значення акумулятору лічильнику не виходило за межі інтервалу  $[N_{min}, N_{max}]$ . У звіті навести програму та графіки зміни основних змінних програми від часу.

1.5. Модернізувати попередню програму додавши до неї керовані вихідні змінні логічного типу, які набувають значення «1»

якщо виконуються умови щодо значення акумулятору лічильнику  $N$  :

$$N > N_1, \text{ де } N_1 \in [N_{min}, N_{max}];$$

$$N < N_1, \text{ де } N_1 \in [N_{min}, N_{max}];$$

$$N_1 < N < N_2, \text{ де } N_1 \in [N_{min}, N_{max}], N_2 \in [N_{min}, N_{max}].$$

1.6. Виконання умов перевіряти за допомогою інструкцій порівняння EQU, GRT, LIM тощо. У звіті навести програму та графіки зміни основних змінних програми від часу.

2. Обміну даними між контролером та додатком користувача у середовищі програми EXCEL та механізму DDE.

2.1. Запустити пакети RSLogix500, RSLinx, RSEmulate 500, EXCEL.

2.2. Створити у пакеті RSLogix500 проект для процесора 1747- L542B вузла 1, драйвера EMUL500. Ввести у проект програму, що один раз у секунду додає одиницю до акумулятору лічильнику C5:0.ACC. Запустити цю програму на виконання за допомогою програми RSEmulate 500.

2.3. Створити новий топик у програмі RSLinx (Меню – DDE/OPC Topic Configuration – New – ввести ім'я топика “test” – визначити мережеву адресу контролера – Apply).

2.4. Для виведення значення слова **C5:0.acc** контролера з топиком “test” в клітинку A1 листа EXCEL ввести в цю клітинку формулу **=rslinx|test!'C5:0.acc'**. Побудувати діаграму для клітинки A1.

2.5. Для виведення значення клітинок A2-A4 листа EXCEL в слова N7:0 – N7:2 контролера з топиком “test” виконати послідовність дій: у програмі RSLogix500 збільшити кількість слів у файлі даних N7;

- ввести значення клітинок A2-A4 листа EXCEL;
- створити кнопку на листі EXCEL;
- ввести наступну програму реакції кнопки на подвійний клік мишки:

○ **A = DDEInitiate("RSLinx", "test")**

○ **DDEPoke A, "n7:0,L3", Range("A2", "A4") DDETerminate**

**(A);**

• у програмі EXCEL натиснути на кнопку, перейти до програми RSLogix500 та спостерігати зміни даних у словах N7:0 – N7:2 контролера.

### Контрольні запитання

1. Склад, призначення та параметри бітових інструкцій.

2. Склад, призначення та параметри інструкцій таймерів.
3. Склад, призначення та параметри інструкцій лічильників.
4. У чому переваги розподіленого додатку керування?
5. Що таке DDE.
6. Що таке «топiк».
7. Назвіть елементи формули, яка визначає дані контролера у програмі EXCEL.
8. Назвіть елементи програми, яка визначає дані EXCEL, що передаються до контролеру.

## ЛАБОРАТОРНА РОБОТА №11

### Пакети програмування SCADA системи

**Мета**, вивчити інтерфейс та основні види робіт у середовищі пакету людино – машинного інтерфейсу RSVIEW32 при створенні додатку керування, навчитися використовувати засоби пакету людино – машинного інтерфейсу для програмування SCADA системи.

### Теоретичні відомості

RSVIEW32 дозволяє:

- створювати та анімувати графічні об'єкти, що відображують елементи об'єкту керування;
- формувати сигнали тривоги щодо цифрових та аналогових тегів, обробляти послідовність сигналів тривоги за ступенем поважності та у хронологічному порядку;
- реєструвати та відстежувати у реальному часі інформацію про систему керування, включаючи виконання команд та макросів, системні події та помилки, а також помилки зв'язку та сигнали тривоги;
- зареєстровані дані зберігати у вигляді файлів та проглядати за допомогою трендів;
- налагодити додаток керування на автоматичний відгук на події у системі та процесах;
- за допомогою програми RSLinx відображувати теги контролерів.

RSVIEW32 існує в двох варіантах. Варіант **Rintime** дозволяє тільки виконувати існуючі додатки керування. Варіант **Works**

дозволяє як виконувати так і розробляти додатки керування.  
Рекомендована література [2,9-15].

### Практична частина

Знайомство з середовищем, створення графічної підсистеми

1. Запустити пакети RSLogix500, RSLinx, RSEmulate 500, RSView32.

2. Створити у пакеті RSLogix500 проект для процесора 1747-L542B вузла 1, драйвера EMUL500. Ввести у проект програму, що один раз у секунду додає одиницю до акумулятору лічильнику C5:0.ACC. Запустити цю програму на виконання за допомогою програми RSEmulate 500.

3. Створити у пакеті RSView32 новий проект. Вивчити склад папок та редакторів Менеджеру проекту.

4. Створити у проекті RSView32 тег Counter акумулятору лічильнику C5:0.ACC, який описано у пункті 2.

5. Створити у проекті RSView32 два нових графічний дисплея з іменами **Displ** та **Disp2**.

6. На дисплеї **Displ** створити новий графічний об'єкт з декількох графічних примітивів. Згрупувати ці примітиви. Згрупованому об'єкту додати керування обертанням.

7. На дисплеї **Displ** створити елемент керування типа «кнопка». Сконфігурувати відгук кнопки на подвійний клік мишею – виконання команди **display (Disp2)** переходу до іншого дисплею.

8. На дисплеї **Disp2** створити графічний об'єкт типу «тренд». Сконфігурувати пери тренду для відображення тегу Counter.

9. На дисплеї **Disp2** створити елемент керування типа «кнопка». Сконфігурувати відгук кнопки на подвійний клік мишею – виконання команди **display (Displ)** переходу до іншого дисплею.

10. Перейти до режиму Run виконання створеного додатку керування. Відкрити дисплей **Displ**. Спостерігати обертання графічного оберту. Зробити подвійний клік мишею на кнопці. Спостерігати перехід до дисплею **Disp2**. Спостерігати на тренді процес зміни значення тегу Counter у часі.

Конфігурування поведінки користувача на прикладі емуляції сигналів світлофору:



- створити наступні теги пам'яті: стану керуючого автомату **State**; кнопки перехід **Start**. Для тегу **State** встановити початковий стан 0;
- створити дисплей з графічними об'єктами ламп світлофору та кнопкою переходу. Задати керування кольором для ламп світлофору за допомогою тегу **State**;
- створити події вказані в таблиці 12.1.
- дозволити виконання подій у редакторі **Startup**;
- перейти до режиму виконання проекту, натиснути на кнопку **Start** та спостерігати процес зміни станів світлофору на дисплеї.

Таблиці 12.1 – Перелік подій

Опис	Дія	Вираз
Старт (перехід «жовтого»)	<b>State = State + 1</b>	<b>(State =0)&amp;(Start=0)</b>
Перехід до «зеленого»	<b>State = State + 1</b>	<b>(State =1)&amp;interval (5 sec)</b>
Перехід до «жовтого»	<b>State = State + 1</b>	<b>(State =2)&amp;interval (10 sec)</b>
Перехід до «червоного»	<b>State = 0</b>	<b>(State =3)&amp;interval (5 sec)</b>

### Контрольні запитання

1. Призначення та функції пакету людино – машинного інтерфейсу RSVIEW32.
2. Призначення та функції редакторів Менеджеру проекту.
3. Різновиди тегів та послідовність їх конфігурування у базі тегів проекту.
4. Послідовність дій при створенні графічного дисплею, графічного оберту, тренду, елемента керування.
5. Назвіть види керування графічними об'єктами та їх параметри.
6. Як задати стани керуючого автомату в проекті RSVIEW32?
7. З чого складається подія.

## ЛАБОРАТОРНА РОБОТА №12

### АСНД з видаленою лабораторією **GOLDi**

**Мета:** Вивчити методи дистанційного контролю та дистанційного спостереження реальних процесів

#### Теоретичні відомості

Гібридна лабораторія **GOLDi** (Grid of Online Lab Devices Ilmenau) розроблена департаментом інтегрованих комунікаційних систем (Department of Integrated Communication Systems) в університеті технологій Ільменау (Ilmenau University of Technology, Німеччина). Вона надає набір інструментів, що підтримують всі етапи проектування для складних завдань управління (наприклад, в області техніки керування, робототехніки, телемеханіки). Мета системи **GOLDi**, щоб показати сучасні способи і проблеми дистанційного контролю та дистанційного спостереження реальних процесів, з комплексним та інтерактивним використанням сучасних Інтернет і Інтранет технологій, таких як HTML5, JavaScript, і т.д. Вона пропонує різні функції, такі як візуалізація та анімація, яка дозволяє спостерігати і перевірити всі властивості конструкції. Зв'язок з формальними методами проектування, моделювання і прототипування використовується для створення основи для розвитку надійної конструкції системи.

Для використання лабораторії **GOLDi** необхідно мати підключений до мережі Internet персональний комп'ютер із встановленими на ньому програмами браузера і Java - машини. Для входу в лабораторію слід використати посилання <http://goldi-labs.net>. Після реєстрації можливо обрати зручну мову спілкування. Далі на домашній сторінці проглянути презентацію лабораторії, а на сторінці "Фізична система" ознайомитись з описом моделей та переліком їх давачів та виконавчих механізмів. На сторінці "Стартовий експеримент" обрати блок керування "Кінцевий автомат" та одну з запропонованих фізичних систем. Якщо експеримент не досяжний, то обрати віртуальний режим роботи та почати експеримент.

У вікні експерименту в блоці керування можливо завантажити свій файл або демонстраційний приклад, проглянути опис входів та

виходів моделі. У блоці керування потоком виконується старт/зупинка процесу виконання програми керування.

У основному вікні представлена анімація роботи віртуальної моделі об'єкту керування та є можливість корегування виразів файлу з програмою керування, які визначають поведінку виходів та машин моделі.

Приклад проекту можна зберегти на своєму комп'ютері і переглянути за допомогою стандартної програми Блокнот. У опису FSM - моделі використані наступні позначення:  $x$  - вхідна змінна,  $y$  - вихідна змінна,  $z$  - стан,  $a$  - автомат; # - логічна операція OR(АБО), & - логічна операція AND(І), ! - логічна операція NOT(НІ).

### Практична частина

1. На сайті лабораторії ознайомитись з структурою лабораторії та описом існуючих моделей. Виконати один із демоприкладів.

2. За допомогою інструменту GIFT створити власний FSM (скінчений автомат) керування та виконати його у віртуальному режимі.

### Контрольні питання

1. Структура видаленої лабораторії
2. Принципи роботи фізичних моделей.
3. Опис скінченого автомату керування фізичною моделлю.
4. Технології реалізації скінченого автомату керування фізичною моделлю

### ПЕРЕЛІК ПОСИЛАНЬ

1. Третяк О.В. Засоби та системи автоматизації наукових досліджень: підручник / О.В. Третяк, Ю.В. Бойко ; КНУТШ. - Київ: ВПЦ "Київський університет", 2007. - 320 с. - (Автоматизація наукових досліджень / ред. О.В. Третяк ).

2. Олсон Г., Пиани Дж. Цифровые системы автоматизации и управления. - СПб.: Невский диалект, 2001, 557с. (рос. мовою).

3. Петергеря Ю.С. Интеллектуальные системы забезпечення енергозбереження житлових будинків: навч. посіб. - К.: Медіа-Прес, 2008. - 255 с.

4. Леоненков А.В. Нечетное моделирование в среде Matlab и FuzzyTech. – СПб.: БХВ – Петербург, 2003. – 736 с. (рос. мовою).

5. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский; Пер. с польск. И.Д. Рудинского.- М.: Горячая линия - Телеком, 2006.- 452 с.- Парал. тит. на польск. яз.
6. Карпов Ю.Г. Теория автоматов. - СПб.: Питер. 2002. - 224 с.: ил. (рос. мовою).
7. Братко И. Программирование на языке ПРОЛОГ для искусственного интеллекта.: Пер. англ. - Мир, 1990. - 560 с., ил.
8. Поляков М.А., Мирошниченко А.Г. Исследование процессов идентификации тепловых параметров силового масляного трансформатора по данным мониторинга. // Вісн. Східноукр. нац. ун-ту. – №11 ч.1(117) 2007. с.160 - 167 (рос. мовою).
9. Поляков М.О., Ларіонова Т.Ю. Системи керування електричними машинами та апаратами. //В кн. Віддалений та віртуальний інструментарій в інжинірингу: монографія / за заг. ред. Карстена Хенке – Запоріжжя: Дике поле, 2015, с. 86-152.
10. Петров И.В. Программируемые контроллеры. Стандартные языки и инструменты / Под. Ред. Проф. В.П. Дьяконова.- М.: СОЛОН-Пресс,2003.-256с.,ил.
11. Системы диспетчерского управления и сбора данных (SCADA – системы). – Мир компьютерной автоматизации, №3, 1999, с.6-9.
12. RSView32. Руководство пользователя. Компания Rockwell Automation. Публикация. 9399-2SE32UG-FEB 97.

### **Література для груп з англійською мовою навчання**

13. Parr E.A. Programmable Controllers. An engineer's guide. Third edition. Oxford: Newnes, 2003, 429 p.
14. John, K.-H., Tiegelkamp, M. IEC 61131-3: Programming Industrial Automation Systems. Concepts and Programming Languages, Requirements for Programming Systems, Aids to Decision-Making Tools.- Шпрингер. 2001 VI, 376 p.
15. Poliakov M., Larionova T.. Control systems with programmable logic controllers, pp. 101-165. Remote and virtual tools in engineering: textbook / general editorship Dr.Ing.Karsten Henke. – Zaporizhzhya: Dike Pole, 2016. – 250 pp.