

Міністерство освіти і науки України
Національний університет водного господарства
та природокористування
Кафедра автоматизації, електротехнічних та комп'ютерно-
інтегрованих технологій

04-03-442М

МЕТОДИЧНІ ВКАЗІВКИ
до проведення практичних занять з навчальної
дисципліни «Програмна інженерія в індустрії 4.0»

для здобувачів вищої освіти першого (бакалаврського) рівня
за освітньо-професійною програмою «Автоматизація,
комп'ютерно-інтегровані технології та робототехніка»
спеціальності 174 «Автоматизація, комп'ютерно-інтегровані
технології та робототехніка» всіх форм навчання

Рекомендовано науково-
методичною радою з якості ННІ
енергетики, автоматики та
водного господарства

Протокол № 6 від 28.01.2025 р.

Рівне – 2025

Методичні вказівки до проведення практичних занять з навчальної дисципліни «Програмна інженерія в індустрії 4.0» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» всіх форм навчання [Електронне видання] / Филипчук Л. В. – Рівне : НУВГП, 2025. – 18 с.

Укладач: Филипчук Л. В., канд. техн. наук, доцент, доцент кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій.

Відповідальний за випуск – Древецький В. В., док. техн. наук, професор, завідувач кафедри автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій.

Керівник групи забезпечення спеціальності
174 «Автоматизація,
комп'ютерно-інтегровані технології
та робототехніка»

Христюк А. О.

Вступ

Предметом вивчення навчальної дисципліни є формування теоретичних знань та практичних навичок, необхідних для розробки та впровадження програмного забезпечення, яке відповідає вимогам сучасних промислових підприємств у рамках концепції Індустрії 4.0. Курс охоплює ключові аспекти створення програмних рішень для автоматизації, роботизації, управління та моніторингу виробничих процесів, зокрема в умовах цифрових фабрик, інтеграції Інтернету речей (IoT), великої кількості даних (Big Data), штучного інтелекту та хмарних технологій.

Міждисциплінарні зв'язки: Дисципліна "Програмна інженерія в індустрії 4.0" має глибокі міждисциплінарні зв'язки з такими галузями, як інформатика, автоматизація, робототехніка, мехатроніка, штучний інтелект, великі дані, економіка та безпека інформаційних систем. Вона поєднує знання програмування і розробки програмного забезпечення з принципами автоматизації та роботизації виробничих процесів, а також з методами обробки великих даних та штучного інтелекту для аналізу і оптимізації роботи підприємств. Знання та навички при вивченні даної дисципліни також знадобляться під час проходження виробничих та переддипломної практик, допоможуть успішно написати та захистити випускову роботу.

Мета навчальної дисципліни "Програмна інженерія в індустрії 4.0" полягає в забезпеченні студентів знаннями та навичками, необхідними для розробки та впровадження програмного забезпечення в умовах Індустрії 4.0. Курс спрямований на ознайомлення студентів з основними принципами та концепціями, що визначають сучасні тенденції у галузі автоматизації та цифрових технологій. У межах дисципліни студенти вивчатимуть методології та інструменти програмної інженерії, які застосовуються для створення ефективних та гнучких систем управління виробництвом, збору та аналізу даних, а також інтеграції різноманітних технологій, таких як Інтернет речей (IoT), штучний інтелект та великі дані (Big Data).

Основними завданнями (цілями) вивчення дисципліни є ознайомлення студентів з методами розробки програмного забезпечення для автоматизованих виробничих систем, інтеграція новітніх технологій, таких як Інтернет речей, штучний інтелект і великі дані, а також набуття навичок розробки ефективних і безпечних програмних рішень для сучасних промислових підприємств у рамках Індустрії 4.0.

Вивчення навчальної дисципліни передбачає набуття здобувачам вищої освіти наступних **компетентностей**:

- здатність застосувати знання у практичних ситуаціях;
- здатність до пошуку, опрацювання та аналізу інформації з різних джерел;
- здатність використовувати для вирішення професійних завдань новітні технології у галузі автоматизації та комп'ютерно-інтегрованих технологій, зокрема, проектування багаторівневих систем керування, збору даних та їх архівування для формування бази даних параметрів процесу та їх візуалізації за допомогою засобів людино-машинного інтерфейсу;
- здатність вільно користуватись сучасними комп'ютерними та інформаційними технологіями для вирішення професійних завдань, програмувати та використовувати прикладні та спеціалізовані комп'ютерно-інтегровані середовища для вирішення задач автоматизації.

В результаті вивчення даної дисципліни здобувачі вищої освіти повинні набути такі **програмні результати навчання**:

- вміти застосовувати сучасні інформаційні технології та мати навички розробляти алгоритми та комп'ютерні програми з використання мов високого рівня та технологій об'єктно-орієнтованого програмування, створювати бази даних та використовувати інтернет-ресурси;
- вміти програмувати багаторівневі системи керування і збору даних для формування бази параметрів процесу та їх візуалізації за допомогою засобів людино-машинного інтерфейсу, використовуючи новітні комп'ютерно-інтегровані технології.

1. РЕКОМЕНДАЦІЇ ДО ВИВЧЕННЯ ДИСЦИПЛІНИ

Вивчати дисципліну рекомендується відповідно до даних методичних вказівок для здобувачів вищої освіти спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка».

У даній розробці наведені теми та плани виконання практичних занять, питання гарантованого рівня знань та список рекомендованої літератури.

Підготовка до кожного практичного заняття передбачає попереднє повторення відповідного теоретичного матеріалу з конспекту лекцій та літературних джерел або самостійне вивчення цього матеріалу з рекомендованих інформаційних джерел. Літературні джерела бажано використовувати при вивченні всіх тем курсу і вибірково переглядати при підготовці до кожного практичного заняття.

Лекційний матеріал та індивідуальні завдання до кожного практичного заняття, зразки їх виконання, перелік необхідної літератури наведено у практичних заняттях, викладених у навчальній платформі MOODLE НУВГП (<https://exam.nuwm.edu.ua/course/view.php?id=6787>), у відповідності до порядкового номера кожного студента у його списку в академічній групі.

Відповідно до навчального плану дисципліни передбачена форма підсумкового контролю – залік.

Оцінювання знань здійснюється за результатами поточного контролю: від 0 до 100 балів.

Теми практичних занять Змістовий модуль 1

Практичне заняття 1.

Розробка алгоритму для керування мікроконтролером.

Результат навчання. Вивчення основ програмування мікроконтролерів для автоматизованих систем.

Основні положення

Мікроконтролери є основними компонентами сучасних автоматизованих систем, які дозволяють інтегрувати програмну логіку в апаратне середовище. Вони містять такі основні елементи:

- Центральний процесор (CPU): Обробляє інструкції програми, управляє ресурсами мікроконтролера.
- Оперативна пам'ять (RAM): Зберігає тимчасові дані під час виконання програм.
- Постійна пам'ять (ROM/Flash): Містить програмний код.
- Периферійні модулі: GPIO, таймери, UART, SPI, I2C тощо.
- Тактовий генератор: Забезпечує синхронізацію всіх операцій.

Мікроконтролери використовуються для створення пристроїв "розумного" дому, медичних приладів, роботів і промислового обладнання. Наприклад, Arduino, ESP32 і STM32 є популярними платформами для навчання й розробки.

Програмування мікроконтролерів: Програмування виконується мовами високого рівня (C, C++). Для цього використовуються середовища розробки, наприклад Arduino IDE. Основна структура програми містить функції `setup()` (виконується один раз при запуску) та `loop()` (виконується циклічно).

```
void setup() {
    pinMode(2, INPUT); // Налаштування кнопки
    pinMode(13, OUTPUT); // Налаштування світлодіода
}

void loop() {
    if (digitalRead(2) == HIGH) {
        digitalWrite(13, HIGH);
    } else {
        digitalWrite(13, LOW);
    }
}
```

Приклад: Програма для ввімкнення світлодіода при натисканні кнопки

Література [3, 12, 14]

Практичне заняття 2.

Налаштування IoT-системи для моніторингу параметрів виробництва

Результат навчання. Опанувати принципи роботи IoT-систем, створити систему моніторингу за допомогою датчиків і хмарної платформи.

Основні положення

IoT-системи базуються на взаємодії датчиків, актуаторів і контролерів, які з'єднуються між собою через мережу Інтернет. Вони дозволяють створювати інтелектуальні виробничі системи з можливістю моніторингу та управління в реальному часі.

Компоненти IoT-системи:

1. Датчики: Зчитують фізичні параметри, такі як температура, тиск, вологість.
2. Контролери: Управляють датчиками й передають дані на сервер або хмару.
3. Хмарні платформи: AWS IoT, Blynk, Google Cloud IoT для зберігання та аналізу даних.
4. Комунікаційні модулі: Wi-Fi, Zigbee, Bluetooth, LoRa для передачі даних.

Протоколи зв'язку:

- MQTT: Легковаговий протокол, що працює за принципом публікації/підписки.
- HTTP: Використовується для передачі даних через веб-сервіси.

Приклад налаштування системи: Підключення датчика DHT11 до ESP32 із передачею даних у хмару через MQTT.

```

import paho.mqtt.client as mqtt
import Adafruit_DHT

sensor = Adafruit_DHT.DHT11
pin = 4

client = mqtt.Client()
client.connect("mqtt.example.com", 1883, 60)

def read_and_publish():
    humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
    if humidity is not None and temperature is not None:
        client.publish("iot/temperature", temperature)
        client.publish("iot/humidity", humidity)

while True:
    read_and_publish()

```

Література [3, 12, 14]

Практичне заняття 3.

Аналіз великих даних виробничих процесів

Результат навчання. Ознайомлення з методами аналізу великих даних для оптимізації виробництва.

Основні положення

Обробка великих даних включає етапи збору, очищення, аналізу та візуалізації. У виробництві великі дані дозволяють:

- Оптимізувати процеси.
- Виявляти аномалії.
- Прогнозувати поведінку систем.

Основні методи аналізу:

1. Машинне навчання: Використовується для побудови моделей прогнозування.
2. Статистичний аналіз: Аналіз кореляцій між змінними.
3. Візуалізація: Використовується для ілюстрації трендів і залежностей.

Приклад аналізу: Використання бібліотек Python для аналізу даних.


```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

data = pd.read_csv('production_data.csv')
sns.pairplot(data)
plt.show()
```

Література [3, 12, 14]

Практичне заняття 4.

Проектування цифрового двійника виробничого обладнання

Результат навчання. Ознайомлення з технологією цифрових двійників.

Основні положення

Цифровий двійник є комп'ютерною моделлю фізичного об'єкта, яка дозволяє прогнозувати поведінку системи в реальному часі та взаємодіяти з нею через цифрові інтерфейси. Він використовується для оптимізації виробничих процесів, зменшення витрат на експлуатацію та підвищення ефективності.

Основні складові цифрового двійника:

1. Збір даних: Включає використання сенсорів для збору інформації про фізичний об'єкт або систему. Приклади сенсорів:
 - Температурні датчики.
 - Датчики тиску.
 - Вібраційні аналізатори.
2. Створення математичної моделі:
 - Диференціальні рівняння.
 - Методи кінцевих елементів (Finite Element Method, FEM).
 - Моделі на основі штучного інтелекту.
3. Інтерфейс користувача:
 - Використання віртуальної реальності (VR) для візуалізації.
 - Інтеграція з платформами SCADA для реального часу.

Етапи створення цифрового двійника:

1. Визначення мети (наприклад, підвищення надійності або зменшення витрат).
2. Вибір необхідних параметрів для моніторингу.
3. Побудова моделі та її тестування.
4. Реалізація у виробничому середовищі.

Приклад: Створення цифрового двійника двигуна в MATLAB/Simulink:

```
% Параметри моделі
J = 0.01; % момент інерції
b = 0.1; % коефіцієнт тертя
K = 0.01; % коефіцієнт зворотного зв'язку
R = 1; % опір
L = 0.5; % індуктивність

% Побудова моделі
sim('motor_model')
```

Література [4, 6, 8]

Змістовий модуль 2

Практичне заняття 5.

Інтеграція робототехнічних систем із SCADA-системою.

Результат навчання. Навчитися інтегрувати робототехнічні системи в SCADA

Основні положення

SCADA (Supervisory Control and Data Acquisition) — це система управління виробничими процесами в реальному часі, яка дозволяє контролювати обладнання, збирати дані та створювати звіти.

Основні компоненти SCADA-системи:

1. Програмно-логічні контролери (PLC):
 - Основний елемент управління.
 - Використовує протоколи Modbus, OPC-UA.
2. RTU (Remote Terminal Unit):
 - Проміжний елемент між сенсорами і SCADA.
3. HMI (Human-Machine Interface):
 - Графічний інтерфейс для відображення інформації та управління.
4. База даних:

- Зберігання історичних даних.

Протоколи передачі даних:

- Modbus TCP/IP: Використовується для передачі даних між PLC і SCADA.
- OPC-UA: Стандартизований протокол для промислових додатків.

Приклад інтеграції: Підключення маніпулятора до SCADA через Ignition:

1. Створення тегів для контролю положення.
2. Конфігурація HMI для відображення.
3. Реалізація зв'язку через Modbus:

```
from pyModbusTCP.client import ModbusClient

client = ModbusClient(host="192.168.1.1", port=502)

if client.open():
    position = client.read_holding_registers(100,
```

Література [4, 5, 6, 8]

Практичне заняття 6.

Розробка алгоритмів управління для мобільного робота.

Результат навчання Опанувати принципи програмування та управління мобільними роботами.

Основні положення

1. Основи мобільної робототехніки:

Мобільні роботи є невід'ємною частиною сучасних автоматизованих систем. Вони здатні переміщуватись у фізичному просторі, взаємодіяти з об'єктами і виконувати складні завдання. Основними компонентами мобільного робота є:

- Шасі: Визначає тип руху робота (колісний, гусеничний, крокуючий).
- Датчики: Використовуються для збору інформації про навколишнє середовище (лідар, ультразвук, GPS, інерційні сенсори).
- Актори: Відповідають за фізичний рух (двигуни, сервоприводи).

- Контролер: Центральний модуль управління, який реалізує алгоритми прийняття рішень.

2. Програмне забезпечення:

Програмування мобільних роботів виконується у спеціалізованих середовищах, таких як:

- Robot Operating System (ROS): Платформа для розробки програмного забезпечення роботів.
- MATLAB: Використовується для моделювання та аналізу траєкторій.

3. Алгоритми навігації:

Навігація мобільного робота включає кілька ключових аспектів:

- Локалізація: Визначення поточного положення в просторі. Використовуються методи, такі як SLAM (Simultaneous Localization and Mapping).
- Планування траєкторії: Алгоритми, як-от A*, Dijkstra, забезпечують оптимальний маршрут до цілі, уникаючи перешкод.
- Управління рухом: Підсистема керує швидкістю та напрямком руху, враховуючи дані від датчиків.

4. Типи датчиків:

- Лідари (LIDAR): Створюють карту оточення за допомогою лазерних променів.
- Камери: Забезпечують візуальну інформацію для аналізу.
- Ультразвукові датчики: Використовуються для визначення відстаней до об'єктів.

5. Приклад алгоритму:

Алгоритм A* для знаходження оптимального шляху в середовищі з перешкодами:

```

1  from heapq import heappop, heappush
2
3  def a_star(grid, start, goal):
4      open_list = []
5      heappush(open_list, (0, start))
6      came_from = {}
7      g_score = {start: 0}
8
9      while open_list:
10         current = heappop(open_list)[1]
11
12         if current == goal:
13             return reconstruct_path(came_from, current)
14
15         for neighbor in get_neighbors(grid, current):
16             tentative_g_score = g_score[current] + 1
17             if tentative_g_score < g_score.get(neighbor, float('inf')):
18                 came_from[neighbor] = current
19                 g_score[neighbor] = tentative_g_score
20                 f_score = tentative_g_score + heuristic(neighbor, goal)
21                 heappush(open_list, (f_score, neighbor))
22

```

Література [4, 6, 8, 10]

Практичне заняття 7.

Створення системи технічного зору для робототехнічного комплексу.

Результат навчання Ознайомлення з принципами роботи систем технічного зору.

Основні положення

1. Технічний зір у робототехніці:

Системи технічного зору дозволяють роботам аналізувати візуальну інформацію для прийняття рішень. Вони знаходять застосування у промисловій збірці, сортуванні, навігації та розпізнаванні об'єктів.

2. Компоненти системи технічного зору:

- Камери: Високошвидкісні або стереокамери для отримання тривимірних зображень.
- Процесори обробки зображень: Використовуються для аналізу даних із камер.

- Освітлення: Забезпечує якість зображення в умовах недостатнього освітлення.

3. Алгоритми технічного зору:

- Фільтри зображення: Gaussian, Sobel для згладжування та виявлення країв.
- Розпізнавання об'єктів: Використовуються моделі машинного навчання (ResNet, YOLO).
- Сегментація зображення: Розділяє об'єкти на основі пікселів.

4. Приклад роботи:

Реалізація виявлення об'єкта за допомогою OpenCV:

```
import cv2

image = cv2.imread('object.jpg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
edges = cv2.Canny(gray, 50, 150)

cv2.imshow("Edges", edges)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

5. Використання нейронних мереж:

Для покращення точності розпізнавання застосовуються згорткові нейронні мережі (CNN). Наприклад, навчання мережі TensorFlow на основі набору даних COCO.

Результатом виконання практичної роботи є правильно оформлений, вчасно зданий та захищений звіт. Звіт вважається вчасно зданим і захищеним, якщо його здача/захист проводились не пізніше наступного практичного заняття. Звіт складається з титульного листа, мети роботи, основної частини та висновку. Звіт оформлюється на стандартному аркуші паперу формату А4 (210x297) з одного боку. Поля: праве – 10 мм, верхнє, нижнє, лівє - 20 мм. Звіт може бути рукописним або друкованим. Звіт може подаватись в електронному вигляді обумовленим викладачем способом (електронна пошта,

електронний носій, представлення звіту на наступному занятті в рукописному вигляді).

Невчасна здача/захист звіту відбувається у терміни, обумовлені викладачем.

3. ПИТАННЯ ГАРАНТОВАНОГО РІВНЯ ЗНАНЬ

ЗМІСТОВИЙ МОДУЛЬ 1

1. Що таке Індустрія 4.0, її основні концепції та роль в автоматизації?
2. Опишіть архітектуру та принципи роботи кіберфізичних систем.
3. Які основні протоколи використовуються в ІоТ?
4. Як великі дані застосовуються для оптимізації виробництва?
5. В чому полягає суть цифрових двійників і їх застосування?
6. Як працюють нейронні мережі в задачах управління роботами?
7. Які функції виконує SCADA-система?
8. Що таке ПоТ, і які відмінності між ІоТ та ПоТ?
9. Які основні методи захисту програмного забезпечення для автоматизованих систем?

ЗМІСТОВИЙ МОДУЛЬ 2.

1. Що таке мікроконтролер, і як його використовують в автоматизації?
2. Як відбувається інтеграція роботів у виробничі процеси?
3. Опишіть основи роботи хмарних обчислень у виробництві.
4. Які моделі використовуються для симуляції автоматизованих систем?
5. Як штучний інтелект допомагає виявляти несправності обладнання?
6. Опишіть етичні аспекти впровадження Індустрії 4.0.

4. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Базова

1. Білан, О. І., Коваленко, В. А. "Індустрія 4.0: нові виклики для автоматизації виробництва". Київ: Либідь, 2022. 312 с.
2. Schwab, К. "The Fourth Industrial Revolution". Geneva: World Economic Forum, 2020. 198 p.
3. Rojko, А. "Industrial IoT in the Context of Industry 4.0". Springer, 2021. 245 p.
4. Lee, J., Bagheri, B., & Kao, H.-A. "A Cyber-Physical Systems Architecture for Industry 4.0-based Manufacturing Systems". Elsevier, 2021. 278 p.
5. "Machine Learning for Industrial Applications", edited by Petkov, P., Springer, 2022. 362 p.
6. Shrouf, F., & Miragliotta, G. "Energy Management in Industry 4.0". Wiley, 2020. 223 p.
7. Шамота, В. Г., Петров, О. М. "Технології цифрового виробництва". Харків: Ранок, 2023. 267 с.
8. Vogl, G. W., Weiss, B. A., & Helu, M. "Digital Twin Applications for Manufacturing". Springer, 2023. 300 p.
9. "SCADA Systems and Industry 4.0: Integration Challenges", edited by Baker, T., Springer, 2021. 310 p.
10. Ковальчук, П. О. "Основи робототехніки та автоматизації". Львів: Світ, 2022. 289 с

Додаткова

1. "IoT and Edge Computing for Industry 4.0 Applications", edited by Jeschke, S., Springer, 2021. 320 p.
2. Ivanov, D., Dolgui, A., & Sokolov, B. "Digital Supply Chain Transformation and Resilience". Springer, 2020. 345 p.
3. "Cybersecurity for Smart Manufacturing", edited by Smith, R., Wiley, 2021. 288 p.
4. Троян, І. В. "Штучний інтелект у системах автоматизації". Дніпро: Наука, 2023. 276 с.

5. "Advanced Analytics in Manufacturing Systems", Springer, 2022. 245 p.

Інформаційні ресурси

1. Цифровий репозиторій Національного університету водного господарства та природокористування [Електронний ресурс]. <http://er3.nuwm.edu.ua/>

2. Каталог нормативних документів України [Електронний ресурс]. <http://uas.org.ua/ua/>

3. Наукова бібліотека НУВГП (м.Рівне, вул. Олекси Новака, 75) [Електронний ресурс]. <http://nuwm.edu.ua/naukova-biblioteka>

4. Національна бібліотека ім. В.І. Вернадського [Електронний ресурс]. <http://www.nbu.gov.ua/>