

Міністерство освіти і науки України  
Національний університет «Запорізька політехніка»

## **МЕТОДИЧНІ ВКАЗІВКИ**

до лабораторних робіт з дисципліни

**«ПРОМИСЛОВА ТА МОБІЛЬНА РОБОТОТЕХНІКА»**

для здобувачів освіти другого (магістерського) рівня вищої освіти

спеціальності 174 «Автоматизація, комп'ютерно-інтегровані

технології та робототехніка»

(освітня програма «Автоматизація, мехатроніка та робототехніка»)

усіх форм навчання

Методичні вказівки до лабораторних робіт з дисципліни «Промислова та мобільна робототехніка» для здобувачів освіти другого (магістерського) рівня вищої освіти спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» (освітня програма «Автоматизація, мехатроніка та робототехніка») усіх форм навчання / Уклад. : Н. О. Миронова, С.В. Шаптала. – Запоріжжя: НУ «Запорізька політехніка», 2024. – 20 с.

Укладачі:           Наталя МИРОНОВА, к.т.н., доцент каф. ІТЕЗ;  
                          Станіслав ШАПТАЛА, ст. викладач каф. ІТЕЗ;

Рецензент:         Микола ЄФИМЕНКО, д.т.н., доцент каф. ІТЕЗ;

Відповідальний  
за випуск:         Наталя МИРОНОВА, к.т.н., доцент каф. ІТЕЗ;

Затверджено на засіданні  
кафедри ІТЕЗ протокол № 1  
від 10.09.24 р.

Рекомендовано до видання на  
засіданні НМК ФІБЕК  
протокол № 2 від  
19.09.24р.

## ЗМІСТ

Вступ.....	4
Лабораторна робота №1 Дослідження та аналіз управління маніпулятором у середовищі симуляції Webots .....	5
Лабораторна робота №2 Розробка класу контролера для управління маніпулятором у Webots .....	7
Лабораторна робота №3 Розробка адаптивного контролера для маніпулятора з використанням сенсорних даних у Webots.....	10
Лабораторна робота №4 Ознайомлення з RoboDK для моделювання та симуляції рухів промислових роботів.....	13
Лабораторна робота №5 Розробка рухів промислових роботів з використанням RoboDK.....	15
Лабораторна робота №6 Експорт програм з RoboDK для використання на реальних промислових роботах .....	18
ЛІТЕРАТУРА .....	20

## ВСТУП

Метою лабораторних робіт є ознайомлення з основами проектування, управління та програмування роботизованих систем, зокрема маніпуляторів, в середовищах симуляції, таких як Webots та RoboDK; розвиток навичок розробки алгоритмів управління та контролерів, а також інтеграції сенсорних систем для адаптивного управління. Це дозволить здобувачам оволодіти практичними навичками у сфері промислової та мобільної робототехніки, підготуватися до роботи над реальними проектами та отримати глибше розуміння принципів функціонування роботизованих систем.

Кожному студенту при підготовці до виконання лабораторної роботи потрібно ознайомитись з методичними вказівками та конспектом лекцій по даному розділу. Виконання лабораторної роботи розбивається на два етапи. На першому етапі необхідно ознайомитись з конспектом лекцій. На другому етапі виконується індивідуальне завдання до лабораторної роботи. Наприкінці заняття результати роботи подаються викладачу для перевірки. Звіт з лабораторної роботи оформлюється на стандартних аркушах паперу формату А4 (297x210мм).

Лабораторна робота обов'язково захищається на наступному після виконання роботи навчальному занятті. За несвоєчасний захист роботи оцінка знижується. Захист лабораторних робіт проводиться під час навчальних занять. Студент не допускається до виконання наступної роботи, якщо має дві незахищені роботи.

# ЛАБОРАТОРНА РОБОТА №1 ДОСЛІДЖЕННЯ ТА АНАЛІЗ УПРАВЛІННЯ МАНІПУЛЯТОРОМ У СЕРЕДОВИЩІ СИМУЛЯЦІЇ WEBOTS

## 1.1 Мета роботи

Ознайомитися з моделями маніпуляторів у середовищі симуляції Webots, обрати модель для дослідження та провести аналіз управління обраним маніпулятором, включаючи розгляд наявних рішень і можливостей вдосконалення контролю руху.

## 1.2 Теоретичні відомості

Маніпулятори використовуються в автоматизації для точного маніпулювання об'єктами. Основні характеристики маніпуляторів включають кількість ступенів свободи (DoF), тип приводів, робочий діапазон, точність і стабільність виконання рухів.

Алгоритми управління маніпуляторами: існує кілька підходів до управління маніпуляторами, таких як зворотна кінематика, пряме та обернене управління, адаптивні та навчальні алгоритми.

Типи контролерів: поширені методи включають ПД-контролери, адаптивні контролери, нейронні мережі та інші методи, які забезпечують плавність, точність і стабільність рухів.

Сенсорні системи для маніпуляторів дозволяють отримувати зворотний зв'язок про позицію, орієнтацію та стан об'єктів, підвищуючи точність та адаптивність управління.

## 1.3 Порядок виконання лабораторної роботи

1. Ознайомитися з теоретичними відомостями та матеріалами лекції.
2. Затвердити індивідуальне завдання у викладача.
3. Відповісти на контрольні питання.
4. Оформити звіт та завантажити до «Системи дистанційного навчання» НУ «Запорізька політехніка».

## 1.4 Індивідуальні завдання

1. Ознайомитися з доступними моделями маніпуляторів у

середовищі симуляції Webots та обрати одну для подальшого дослідження.

2. Проаналізувати технічні характеристики обраного маніпулятора, включаючи кількість ступенів свободи, тип приводів, діапазон руху та можливі сценарії використання.

3. Вивчити готові рішення щодо керування рухом обраного маніпулятора: дослідити існуючі алгоритми управління та методи їх реалізації.

4. Ознайомитися з типами контролерів руху, що використовуються для управління маніпуляторами, та визначити, які підходи найкраще підходять для вашої моделі.

5. Дослідити можливості інтеграції сенсорних систем для підвищення точності та адаптивності руху маніпулятора.

6. Провести порівняльний аналіз знайдених рішень, оцінюючи їх ефективність, точність, стабільність, а також переваги та недоліки.

7. Підготувати звіт, який включатиме опис обраного маніпулятора, огляд готових рішень щодо його руху та можливі пропозиції для вдосконалення керування.

### 1.5 Контрольні запитання

1. Які моделі маніпуляторів доступні у середовищі Webots, та за якими критеріями обрана модель для подальшого дослідження?

2. Які технічні характеристики має обраний маніпулятор (ступені свободи, тип приводів, діапазон руху)?

3. Які існують алгоритми керування рухом маніпуляторів, та які методи найбільш застосовні для вашої моделі?

4. Які типи контролерів використовуються для управління маніпуляторами? Які переваги має обраний тип для вашої моделі?

5. Які сенсорні системи можуть бути інтегровані для підвищення точності маніпулятора, та які типи сенсорів є доцільними для вашого випадку?

6. Як оцінюється ефективність різних підходів до керування рухом маніпулятора (точність, стабільність, швидкість)?

7. Які можливі пропозиції щодо вдосконалення керування обраним маніпулятором ви можете запропонувати на основі проведеного аналізу?

## ЛАБОРАТОРНА РОБОТА №2 РОЗРОБКА КЛАСУ КОНТРОЛЕРА ДЛЯ УПРАВЛІННЯ МАНІПУЛЯТОРОМ У WEBOTS

### 2.1 Мета роботи

Розробити клас контролера рухів для управління маніпулятором, що забезпечить точне досягнення цільових позицій і плавне виконання рухів із використанням алгоритмів кінематики, адаптивного управління та зворотного зв'язку.

### 2.2 Теоретичні відомості

Кінематика – це розділ механіки, який вивчає рух без урахування сил, що його викликають. Для маніпуляторів кінематика визначає положення і орієнтацію кінцевого ефектора на основі заданих кутів суглобів (пряма кінематика) або знаходить кути суглобів для досягнення заданої точки (зворотна кінематика).

Пряма кінематика дозволяє обчислити положення кінцевого ефектора відомого маніпулятора на основі заданих кутів кожного суглоба. Це корисно для визначення положення робочого інструмента.

Зворотна кінематика розв'язує зворотну задачу: для заданого положення кінцевого ефектора знаходить кути кожного суглоба, що необхідні для досягнення цього положення. Це завдання є складнішим і потребує числових методів для його вирішення у більшості випадків.

ПД-контролер (пропорційно-інтегрально-диференційний контролер) ПІД-контролер є основним інструментом керування, що забезпечує коригування положення або швидкості за допомогою трьох компонентів:

– пропорційна компонента (P) – визначає реакцію контролера на поточну похибку. Вона пропорційна різниці між поточним і бажаним значенням, що дає основу для виправлення помилок;

– інтегральна компонента (I) – враховує накопичену помилку з часом, що дозволяє компенсувати постійні похибки, яких не може позбутися пропорційна компонента;

– диференційна компонента (D) – забезпечує реакцію на зміну похибки, допомагаючи стабілізувати систему і зменшити коливання. Налаштування коефіцієнтів цих трьох компонент є ключовим для забезпечення стабільності та точності контролера.

Програмна реалізація контролера рухів виконується у вигляді класу, який об'єднує всі необхідні методи для розрахунків кінематики та управління положенням маніпулятора. Такий клас дозволяє створити узагальнену модель для роботи з різними маніпуляторами і забезпечити адаптоване управління.

### 2.3 Порядок виконання лабораторної роботи

1. Ознайомитися з теоретичними відомостями та матеріалами лекції.
2. Затвердити індивідуальне завдання у викладача.
3. Відповісти на контрольні питання.
4. Оформити звіт та завантажити до «Системи дистанційного навчання» НУ «Запорізька політехніка».

### 2.4 Індивідуальні завдання

1. Розробити клас `MotionController`, який включатиме всі необхідні параметри для управління маніпулятором, такі як кількість ступенів свободи, допустимі межі руху, та константи контролера.
2. Визначити початкові значення для коефіцієнтів ПІД-контролера для руху маніпулятора.
3. Реалізувати методи для розрахунку прямої та зворотної кінематики, що дозволяють перетворювати цільові координати у відповідні кути для кожного суглоба маніпулятора.
4. Забезпечити можливість досягнення цільових позицій через використання кінематичних обчислень.
5. Створити метод `pid_control()`, який буде обчислювати керуючі сигнали для кожного суглоба, використовуючи алгоритм ПІД-контролю.
6. Додати можливість налаштування параметрів контролера для досягнення необхідної точності та стабільності рухів.
7. Розробити тестовий сценарій, який продемонструє роботу класу контролера при досягненні різних цільових позицій.
8. Оцінити точність і стабільність управління при різних налаштуваннях ПІД-параметрів.
9. Оформити звіт з описом розробленого класу, методів і параметрів.

## 2.5 Контрольні запитання

1. Поняття кінематики маніпулятора. Чим вона відрізняється від динаміки?
2. У чому різниця між прямою та зворотною кінематикою?
3. Як здійснюється обчислення положення кінцевого ефектора у прямій кінематиці?
4. Які математичні методи використовуються для розв'язання задачі зворотної кінематики?
5. Що таке ПД-контролер, і які його основні компоненти?
6. Як пропорційна компонента впливає на роботу контролера?
7. Яку роль відіграє інтегральна компонента у ПД-контролері?
8. Яка функція диференційної компоненти, і як вона впливає на стабільність системи?
9. Яким чином налаштування параметрів ПД-контролера впливає на ефективність управління?
10. Які методи управління можна використовувати для підвищення точності і стабільності роботи маніпулятора?
11. Як можна протестувати роботу класу контролера рухів у середовищі симуляції?
12. Чому важливо перевіряти межі допустимих кутів у маніпулятора, і як це враховується в класі контролера?

## ЛАБОРАТОРНА РОБОТА №3 РОЗРОБКА АДАПТИВНОГО КОНТРОЛЕРА ДЛЯ МАНІПУЛЯТОРА З ВИКОРИСТАННЯМ СЕНСОРНИХ ДАНИХ У WEBOTS

### 3.1 Мета роботи

Розробити клас адаптивного контролеру для управління маніпулятором у Webots, який забезпечуватиме автоматичне налаштування параметрів управління на основі сенсорних даних.

### 3.2 Теоретичні відомості

Адаптивний контроль – це підхід в управлінні, який дозволяє контролеру автоматично налаштовувати свої параметри залежно від змінних умов. На відміну від звичайного ПД-контролера, адаптивний контролер може змінювати свої налаштування у реальному часі для підвищення точності і стабільності системи.

Типи сенсорів для управління маніпулятором. Сенсори допомагають маніпулятору взаємодіяти з навколишнім середовищем і підвищують точність виконання завдань. Найбільш поширені сенсори для адаптивного контролю:

- датчики положення: вимірюють поточне положення суглобів маніпулятора;
- датчики сили або навантаження: визначають прикладену силу, що дозволяє контролеру коригувати рухи у випадку змінного навантаження;
- датчики наближення або об'єктів: виявляють наявність об'єктів поблизу, що допомагає уникати перешкод.

Адаптивний ПД-контролер може змінювати свої коефіцієнти ( $K_p$ ,  $K_i$ ,  $K_d$ ) залежно від даних сенсорів. Наприклад, при збільшенні навантаження на маніпулятор адаптивний контролер може збільшити пропорційний коефіцієнт для точнішого контролю руху. Це дозволяє системі працювати стабільніше в умовах непередбачених змін.

Алгоритми адаптивного налаштування. Існує кілька підходів до адаптивного налаштування параметрів:

- таблиці налаштувань: значення коефіцієнтів контролера змінюються в залежності від певних умов, визначених у таблиці;
- методи машинного навчання: для складних систем можуть

використовуватися алгоритми машинного навчання, які допомагають контролеру навчатися і коригувати параметри на основі даних;  
– евристичні методи: налаштування параметрів на основі правил, які враховують поведінку системи у відповідь на зміни.

### 3.3 Порядок виконання лабораторної роботи

1. Ознайомитися з теоретичними відомостями та матеріалами лекції.
2. Затвердити індивідуальне завдання у викладача.
3. Відповісти на контрольні питання.
4. Оформити звіт та завантажити до «Системи дистанційного навчання» НУ «Запорізька політехніка».

### 3.4 Індивідуальні завдання

1. Ознайомитися з можливими типами сенсорів у Webots, зокрема з датчиками положення, сили і наближення.
2. Визначити, які сенсори найкраще підходять для адаптивного управління об'єктом маніпулятором.
3. Розробити клас AdaptiveController, який включатиме базові методи контролю руху, зокрема ПД-контроль для досягнення заданих позицій.
4. Додати методи для автоматичного налаштування параметрів контролера (коефіцієнтів ПД) на основі зворотного зв'язку від сенсорів.
5. Додати можливість отримання даних із сенсорів у режимі реального часу.
6. Розробити алгоритм, який коригуватиме рухи маніпулятора залежно від показників датчиків (наприклад, уникнення перешкод або компенсація змінної маси об'єкта).
7. Налаштувати і провести тестові експерименти з адаптивним контролером для перевірки його працездатності.
8. Оцінити точність і стабільність рухів маніпулятора при роботі з різними навантаженнями та перешкодами.
9. Оформити звіт з описом роботи адаптивного контролера, обґрунтуванням вибору параметрів та алгоритму обробки сенсорних даних.

### 3.5 Контрольні запитання

1. Що таке адаптивний контроль, і чим він відрізняється від звичайного контролю?
2. Які типи сенсорів найкраще підходять для підвищення точності рухів маніпулятора?
3. Як працює адаптивний ПД-контролер, і які параметри він може змінювати?
4. Які задачі вирішує сенсорна система у процесі управління маніпулятором?
5. Які є основні методи адаптивного налаштування параметрів контролера?
6. Як сенсори наближення можуть впливати на рух маніпулятора?
7. Які переваги надає використання датчиків сили для управління рухом?
8. Як можна оцінити ефективність адаптивного контролера у порівнянні з традиційним контролером?

## **ЛАБОРАТОРНА РОБОТА №4 ОЗНАЙОМЛЕННЯ З ROBODK ДЛЯ МОДЕЛЮВАННЯ ТА СИМУЛЯЦІЇ РУХІВ ПРОМИСЛОВИХ РОБОТІВ**

### **4.1 Мета роботи**

Ознайомлення з функціональними можливостями програмного забезпечення RoboDK, його інтерфейсом та основними інструментами для моделювання, симуляції та програмування рухів промислових роботів.

### **4.2 Теоретичні відомості**

RoboDK – це програмне забезпечення для моделювання, симуляції та програмування промислових роботів. Воно надає інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам легко створювати траєкторії та програмувати роботи для виконання різноманітних завдань, таких як зварювання, складання, обробка та фарбування.

Моделювання рухів у RoboDK дозволяє розробникам тестувати свої програми у віртуальному середовищі перед їх реалізацією на реальних роботах. Це суттєво знижує ризик помилок і підвищує ефективність роботи.

RoboDK дозволяє експортувати тексти програм, у різних форматах, такі як G-code, RAPID, KRL, які можуть бути використані на різних типах промислових роботів. Це дозволяє інтегрувати програмне забезпечення RoboDK з уже існуючими роботизованими системами.

### **4.3 Порядок виконання лабораторної роботи**

1. Ознайомитися з теоретичними відомостями та матеріалами лекції.
2. Ознайомитися з функціональними можливостями RoboDK для симуляції рухів промислових роботів.
3. Відповісти на контрольні питання.
4. Оформити звіт та завантажити до «Системи дистанційного навчання» НУ «Запорізька політехніка».

#### 4.4 Індивідуальні завдання

1. Запустити програму RoboDK та вивчити основні елементи інтерфейсу (панелі інструментів, робоча область, меню).
2. Дослідити функції навігації у 3D-просторі та взаємодії з моделями роботів.
3. Імпортувати стандартну модель промислового робота з бібліотеки RoboDK (на вибір).
4. Налаштувати параметри робота (кількість ступенів свободи, робочі області тощо).
5. Створити прості траєкторії рухів для обраного маніпулятора.
6. Використати інструменти RoboDK для зміни параметрів траєкторій та експериментування з різними сценаріями.
7. Запустити симуляцію рухів маніпулятора та спостерігати за виконанням.
8. Вивчити можливості візуалізації (додати елементи середовища, наприклад, об'єкти для маніпуляції).
9. Ознайомитися з можливістю експорту програм для використання на реальних промислових роботах.
10. Вивчити формати файлів, які підтримує RoboDK для передачі програм у робототехнічні системи.
11. Оформити звіт про виконану роботу, звіт повинен містити скріншоти інтерфейсу та опис виконаних завдань.

#### 4.5 Контрольні запитання

1. Які його основні функціональні можливості RoboDK?
2. Які типи промислових роботів можна імплементувати в RoboDK?
3. Які основні етапи моделювання рухів у RoboDK?
4. Які формати експорту підтримує RoboDK для програмування роботів?
5. Які переваги надає симуляція рухів у віртуальному середовищі перед їх реалізацією на реальних роботах?
6. Як налаштувати параметри робота у RoboDK?
7. Яким чином можна візуалізувати рухи маніпулятора в RoboDK?

## ЛАБОРАТОРНА РОБОТА №5 РОЗРОБКА РУХІВ ПРОМИСЛОВИХ РОБОТІВ З ВИКОРИСТАННЯМ ROBODK

### 5.1 Мета роботи

Дослідити основи розробки рухів промислових роботів, використовуючи програмне забезпечення RoboDK. Реалізувати та протестувати алгоритми керування рухами для виконання промислових завдань, таких як маніпулювання об'єктами або збирання деталей.

### 5.2 Теоретичні відомості

RoboDK – це платформа для моделювання, програмування та симуляції промислових роботів, що дозволяє створювати моделі, розробляти траєкторії руху і навіть генерувати тексти програм для реальних роботів. У контексті розробки рухів промислових роботів важливо враховувати такі поняття:

- кінематика робота: кінематика вивчає рух механічної системи без урахування її маси та сил. Прямий розрахунок кінематики визначає положення і орієнтацію кінцевого ефектора на основі заданих суглобових кутів, тоді як зворотний розрахунок знаходить кути суглобів для досягнення певного положення кінцевого ефектора;

- планування траєкторії: робот має виконувати завдання, слідуючи заданій траєкторії, уникати перешкод та рухатися оптимальним шляхом для досягнення мети. RoboDK дозволяє легко налаштувати та оптимізувати траєкторії для конкретних завдань, наприклад, зварювання, пакування чи збирання;

- контроль руху: включає налаштування швидкості, точності й повторюваності рухів робота. У промислових системах важлива стабільність руху, щоб уникнути помилок під час виконання точних завдань;

- інтерфейс RoboDK: програма надає простий інтерфейс для симуляції, де користувач може вибрати різні моделі роботів, налаштувати їх траєкторії та виконувати симуляцію. Додатково RoboDK може генерувати тексти програм для роботів на різних мовах програмування C++, Python та ін., що робить його універсальним інструментом;

– синхронізація з фізичними роботами: RoboDK дозволяє експортувати розроблені алгоритми рухів у вигляді текстів програм, які можна завантажити на реальні роботи різних виробників, таких як ABB, KUKA, Fanuc та ін., що дозволить прискорити процес впровадження та тестування програм у реальних умовах.

### **5.3 Порядок виконання лабораторної роботи**

1. Ознайомитися з теоретичними відомостями та матеріалами лекції.
2. Обрати модель промислового робота в RoboDK та налаштувати його кінематичні параметри.
3. Розробити алгоритм планування траєкторії робота для виконання виробничої операції (наприклад, збирання об'єктів, маніпуляція деталями або зварювання).
4. Відповісти на контрольні питання.
5. Оформити звіт та завантажити до «Системи дистанційного навчання» НУ «Запорізька політехніка».

### **5.4 Індивідуальні завдання**

1. Ознайомитися з функціональними можливостями RoboDK для симуляції рухів промислових роботів.
2. Обрати модель промислового робота в RoboDK та налаштувати його кінематичні параметри.
3. Розробити алгоритм для прямого та зворотного розрахунку кінематики для керування рухом промислового робота.
4. Розробити алгоритм планування траєкторії робота для виконання однієї виробничої операції (наприклад, збирання об'єктів, маніпуляція деталями або зварювання).
5. Виконати симуляцію роботи робота та проаналізувати точність та плавність рухів у виконанні поставлених завдань.
6. Згенерувати текст програми для реального робота та дослідити можливості його використання у фізичних системах.
7. Оформити звіт, який включає опис алгоритму, результати експериментів, з графічними ілюстраціями результатів симуляції.

### 5.5 Контрольні запитання

1. Що таке кінематика робота і в чому полягає різниця між прямим та зворотним розрахунком кінематики?
2. Які методи планування траєкторій використовуються для оптимізації рухів промислових роботів?
3. Як RoboDK допомагає у плануванні та моделюванні рухів промислових роботів?
4. Які фактори впливають на точність і стабільність рухів робота під час виконання завдань?
5. Які особливості синхронізації алгоритмів, розроблених у RoboDK, з реальними роботами?
6. Які показники важливі для оцінки якості виконання промислового завдання роботом?

## **ЛАБОРАТОРНА РОБОТА №6 ЕКСПОРТ ПРОГРАМ З ROBODK ДЛЯ ВИКОРИСТАННЯ НА РЕАЛЬНИХ ПРОМИСЛОВИХ РОБОТАХ**

### **6.1 Мета роботи**

Ознайомлення здобувачів вищої освіти з процесом експорту програм з RoboDK у формати, сумісні з реальними промисловими роботами, а також підготовка їх до тестування на фізичних пристроях.

### **6.2 Теоретичні відомості**

У сучасній промисловій автоматизації важливу роль відіграє інтеграція роботизованих систем у виробничі процеси. RoboDK є одним з інструментів для симуляції та програмування промислових роботів. Після створення програми в RoboDK важливо експортувати її у формат, сумісний з реальними роботами. Серед основних форматів, які підтримує RoboDK, можна виділити G-code, RAPID (для ABB), KRL (для KUKA) та інші. Вибір формату залежить від типу робота та специфікацій виробника. Експорт програми включає налаштування параметрів, таких як швидкість, точність та безпека виконання завдань. Важливим етапом є тестування програм на реальних роботах, що дозволяє виявити можливі недоліки та внести корективи. Це тестування забезпечує не лише перевірку коректності виконання, але й підвищення ефективності виробничих процесів. Розуміння особливостей програмування та експорту є необхідним для успішного впровадження роботизованих рішень у виробництві.

### **6.3 Порядок виконання лабораторної роботи**

1. Ознайомитися з теоретичними відомостями та матеріалами лекції.
2. Обрати модель промислового робота в RoboDK.
3. Розробити програму в RoboDK для виконання відповідної виробничої операції.
4. Відповісти на контрольні питання.
5. Оформити звіт та завантажити до «Системи дистанційного навчання» НУ «Запорізька політехніка».

#### 6.4 Індивідуальні завдання

1. Ознайомитися з різними форматами програм, які підтримує RoboDK для експорту (G-code, RAPID, KRL тощо).
2. Розглянути специфікації для різних виробників роботів.
3. Розробити програму в RoboDK для конкретного завдання (наприклад, зварювання, обробка, фарбування).
4. Використати налаштування та параметри, які будуть необхідні для реального виконання.
5. Виконати експорт програми у форматі, сумісному з обраним промисловим роботом.
6. Ознайомитися з етапами підготовки та налаштування програми для реальної роботи.
7. Перевірити програму на фізичному промисловому роботі, вивчаючи реакцію та точність виконання завдань (за можливістю).
8. Оцінити продуктивність програми в реальних умовах(за можливістю).
9. Оформити звіт з описом етапів експорту програми та результатам тестування.

#### 6.5 Контрольні запитання

1. Які формати програм експортуються з RoboDK і для яких роботів вони призначені?
2. Які основні етапи експорту програми для промислового робота?
3. Які аспекти слід враховувати при тестуванні програми на реальному роботі?
4. Як можна оптимізувати програму для підвищення її продуктивності?
5. Чому важливо перевіряти програми у реальних умовах перед їх впровадженням?

## ЛІТЕРАТУРА

1. Robotis simulation services [Електроний ресурс]: режим доступу - <https://cyberbotics.com/>
2. Robot Modeling and Control [Режим доступу]: <http://www.coep.ufrj.br/~ramon/COE-841/robotics/book%202005%20-%20Robot%20Modeling%20and%20Control%20-%20Spong,%20Hutchinson%20&%20Vidyasagar.pdf>
3. Проць Я. І. Захоплювальні пристрої промислових роботів: Навчальний посібник. /Я. І. Проць — Т: Тернопільський державний технічний університет ім. І. Пулюя, 2008. — 232 с.
4. David J. C. MacKay[en]. Information Theory, Inference, and Learning Algorithms Cambridge: Cambridge University Press, 2003.
5. Довідник ROS, [Режим доступу]: <https://wiki.ros.org/>
6. Readme Robotics Toolbox for Python, [Режим доступу]: <https://github.com/petercorke/robotics-toolbox-python>
7. Peter Corke Robotics, Vision and Control: Fundamental Algorithms in Python (Springer Tracts in Advanced Robotics, 146) 3rd ed. 2023 Edition
8. Довідник «Robotics Toolbox for Python package», [Режим доступу]: [https://petercorke.github.io/robotics-toolboxpython/arm\\_erobot.html#erobot](https://petercorke.github.io/robotics-toolboxpython/arm_erobot.html#erobot)
9. Basic Guide <https://robodk.com/doc/en/Basic-Guide.html>
10. Poole C. P., Owens F. J. Introduction to Nanotechnology. — New Jersey: Wiley–Interscience, 2003. — 388 p.
11. RoboDK [Режим доступу]: <https://robodk.com/doc/en/Getting-Started.html>