

Р. С. Белзецький, А. В. Козловський, Ю. М. Паночишин

**ПРАКТИКУМ З КОМП'ЮТЕРНОЇ
ІНЖЕНЕРІЇ ТА ОСНОВ
РОБОТОТЕХНІКИ**

Міністерство освіти і науки України
Вінницький національний технічний університет

Р. С. Белзецький, А. В. Козловський, Ю. М. Паночишин

ПРАКТИКУМ З КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ОСНОВ РОБОТОТЕХНІКИ

Електронний практикум

Вінниця
ВНТУ
2026

УДК 004.415 (076)

Б43

Рекомендовано до видання Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 11 від 26.03.2026 р.)

Рецензенти:

Н. Р. Веселовська, доктор технічних наук, професор

М. С. Юхимчук, доктор технічних наук, професор

Л. Г. Коваль, кандидат технічних наук, доцент

Белзецький, Р. С.

Б43 Практикум з комп'ютерної інженерії та основ робототехніки : практикум [Електронний ресурс] / Белзецький Р. С., Козловський А. В., Паночишин Ю. М. – Вінниця : ВНТУ, 2026. (PDF, 106 с.).

У виданні стисло викладено основні положення курсу «Комп'ютерна інженерія та основи робототехніки» відповідно до навчальної програми з тематик, за якими виконуються практичні роботи. Наведено приклади програмного коду та надано алгоритми виконання розрахункової та програмної частин практичних робіт. Наведено необхідні довідникові матеріали для виконання практичних робіт. Для самоконтролю готовності до практичної роботи у практикумі наприкінці кожної теми подано контрольні запитання. Видання розроблено відповідно до плану кафедри комп'ютерних наук та програми дисципліни «Комп'ютерна інженерія та основи робототехніки» для виконання практичних робіт здобувачами вищої освіти за спеціальністю ФЗ – «Комп'ютерні науки» (освітній ступінь – бакалавр). Практикум може бути використаний під час підготовки здобувачів інших спеціальностей.

УДК 004.415 (076)

© ВНТУ, 2026

ЗМІСТ

Вступ.....	4
Практична робота № 1. Побудова та аналіз роботи комбінаційних логічних пристроїв	5
Практична робота № 2. ВЕАМ робот на основі нейронних кіл М. Тільдена. Концепція. Будова. Принцип керування.....	16
Практична робота № 3. Персональна та локальна навігація мобільних роботів	39
Практична робота № 4. Будова маніпулятора. Структурно-кінематичний аналіз	63
Практична робота № 5. Методи розв’язання прямої задачі кінематики.....	80
Глосарій.....	97
Література	103

ВСТУП

Виконання практичних робіт з обов'язкової професійної навчальної дисципліни «Комп'ютерна інженерія та основи робототехніки» передбачено навчальним планом освітньої програми «Комп'ютерні науки» за спеціальністю F3 – «Комп'ютерні науки».

Дисципліна «Комп'ютерна інженерія та основи робототехніки» ґрунтується на опануванні базових знань з фізичних та логічних принципів побудови електронних схем цифрових елементів і функціональних вузлів комп'ютерної техніки, а також оволодінні основними принципами і методами побудови робототехнічних систем та їх програмування. Ця дисципліна безпосередньо пов'язана і доповнює такі базові дисципліни, як «Вступ до фаху», «Фізика», «Алгоритмізація та програмування» [1].

Практикум з дисципліни «Комп'ютерна інженерія та основи робототехніки» призначений для здобувачів освіти II курсу бакалаврату спеціальності F3 – «Комп'ютерні науки», але може стати у пригоді здобувачам інших спеціальностей.

Практична робота № 1

ПОБУДОВА ТА АНАЛІЗ РОБОТИ КОМБІНАЦІЙНИХ ЛОГІЧНИХ ПРИСТРОЇВ.

Мета роботи: набуття практичних навичок реалізації логічних функцій та аналізу роботи логічних комбінаційних пристроїв.

1.1 Теоретичні відомості [2, 3, 4, 5]

Логічний елемент – це електронний пристрій, що реалізує одну з логічних операцій. Логічні елементи являють собою електронні пристрої, у яких оброблювана інформація закодована у вигляді двійкових чисел, відображуваних напругою (потенціалом) високого і низького рівня. Термін «логічні» прийшов в електроніку з алгебри логіки, що оперує зі змінними величинами і їхніми функціями, які можуть набувати тільки двох значень: «істинно» або «хибно». Для їх позначення використовують відповідно символи 1 чи 0. Кожна логічна змінна може набувати тільки одного значення: логічна 1 або логічний 0. Ці двійкові змінні та функції від них називаються логічними змінними та логічними функціями. Пристрої, що реалізують логічні функції, називаються логічними чи цифровими пристроями. Основу сучасних пристроїв обробки інформації становлять цифрові (логічні) ІМС.

Аналіз роботи цифрових пристроїв базується на використанні апарату математичної логіки – алгебри Джорджа Буля (1815–1864 рр.). В її основі лежить поняття події, що оцінюється з погляду її настання: вона може статися або не статися.

Тоді кожному подію можна вважати істинною, що може моделюватися одиницею «1» (високий рівень напруги у разі електричного моделювання) або хибною, що моделюється нулем «0» (низький рівень напруги).

Обробка інформації, що подається у вигляді подій, ведеться у двійковій системі числення, яка має тільки дві цифри: 0 і 1. Величина, що може набувати тільки цих двох значень, називається двійковою (логічною) змінною.

Складна подія, що залежить від декількох двійкових змінних, називається двійковою (логічною, перемикальною) функцією:

$$y = f(x_1, x_2, \dots, x_n), \quad \text{де } x_i = \{1, 0\}.$$

Цифрові пристрої поділяються на два великі класи: *комбінаційні* і *последовні*.

Комбінаційні пристрої реалізують функції, що залежать тільки від комбінації змінних, у цей момент часу і не залежать від стану пристрою в попередній момент.

Послідовні пристрої реалізують функції, що залежать як від вхідних змінних у поточний момент, так і від попереднього стану пристрою.

В алгебрі логіки є три основних логічних операції:

- операція НЕ (інверсія) – логічне заперечення:

$$y = \bar{x}.$$

- операція АБО (диз'юнкція) – істина, якщо істина хоча б одна змінна:

$$y = x_1 + x_2$$

- операція І (кон'юнкція) – істина, якщо істинні обидві змінні:

$$y = x_1 \cdot x_2$$

Порядок дій в алгебрі логіки такий: НЕ $\rightarrow$ І $\rightarrow$ АБО. Для зміни порядку використовують дужки.

Для алгебри логіки справедливі закони:

- переставний (комутативний)

$$x_1 + x_2 = x_2 + x_1 \quad \text{та} \quad x_1 \cdot x_2 = x_2 \cdot x_1$$

- сполучний (асоціативний)

$$x_1 + x_2 + x_3 = x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3$$

$$x_1 \cdot x_2 \cdot x_3 = x_1 \cdot (x_2 \cdot x_3) = (x_1 \cdot x_2) \cdot x_3;$$

- розподільний (дистрибутивний)

$$x_1 \cdot (x_2 + x_3) = x_1 \cdot x_2 + x_1 \cdot x_3.$$

Тотожності алгебри логіки:

$$x + x = x \quad \text{та} \quad x \cdot x = x;$$

$$x + \bar{x} = 1 \quad \text{та} \quad x \cdot \bar{x} = 0;$$

$$x + 0 = x \quad \text{та} \quad x \cdot 0 = 0;$$

$$\bar{\bar{x}} = x;$$

$$x_1 + x_1 \cdot x_2 + x_1 \cdot x_3 = x_1;$$

$$x_1 + \bar{x}_1 \cdot x_2 = x_1 + x_2;$$

$$x_1 \cdot x_2 + x_1 \cdot \bar{x}_2 = x_1;$$

$$x_1 \cdot (x_1 \cdot x_2) = x_1;$$

$$x_1 \cdot (\bar{x}_1 + x_2) = x_1 \cdot x_2;$$

$$(x_1 + x_2) \cdot (x_1 + x_3) = x_1 + x_2 \cdot x_3.$$

До основних законів алгебри логіки також відносяться закони інверсії для логічних додавання та множення – правила де Моргана:

$$\overline{x_1 + x_2} = \bar{x}_1 \cdot \bar{x}_2 \quad \text{та} \quad \overline{x_1 \cdot x_2} = \bar{x}_1 + \bar{x}_2.$$

Застосування цих законів приводять до суттєвого спрощення логічних функцій, а значить і схем пристроїв, що їх реалізують.

Реалізація простих логічних функцій

Однією з найпростіших логічних функцій є функція заперечення НЕ.

Схему логічного елемента НЕ та її умовне позначення, що реалізує таку функцію, наведено на рисунку 1.1.

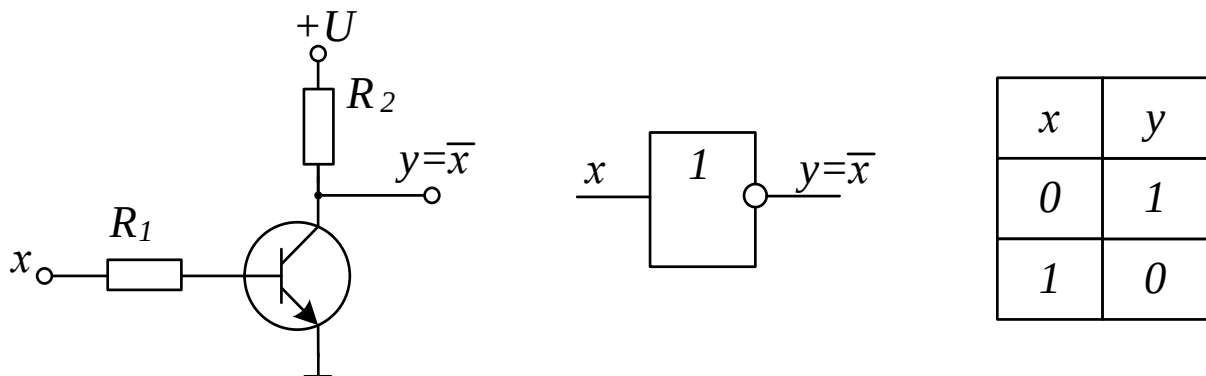


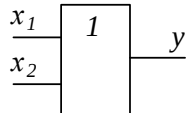
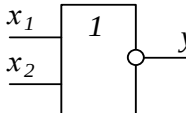
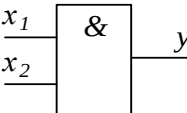
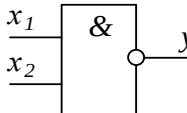
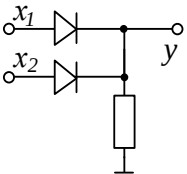
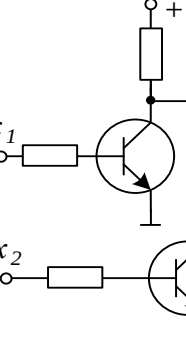
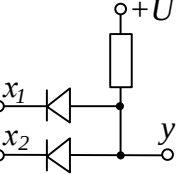
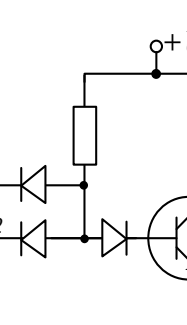
Рисунок 1.1 – Схема логічного елемента НЕ, її умовне позначення та таблиця істинності

Як такий елемент може бути використано, наприклад, транзисторний ключ – підсилювач зі спільним емітером (СЕ), що працює у ключовому режимі: за високого рівня напруги на вході на виході матимемо низький рівень і навпаки.

Функцію, яку виконує комбінаційний пристрій, для полегшення сприйняття часто подають у вигляді таблиці, яку називають таблицею істинності. Кількість стовпців цієї таблиці дорівнює числу змінних, що входять до функції, а також міститься ще один стовпець, у якому вказують значення функції для кожної з можливих комбінацій вхідних змінних. Кількість комбінацій, а отже й кількість рядків таблиці, у загальному випадку дорівнює 2^n , де n — число змінних.

Об'єднану таблицю істинності деяких основних логічних функцій, що залежать від двох змінних, а також математичний запис функцій, їхню назву, графічне позначення, та можливу реалізацію наведено в таблиці 1.1.

Таблиця 1.1 – Таблиця істинності деяких основних логічних функцій

Вхідні змінні		Функція y			
x_1	x_2	АБО	АБО-НЕ	I	I-НЕ
0	0	0	1	0	1
0	1	1	0	0	1
1	0	1	0	0	1
1	1	1	0	1	0
Математичний запис (формула)		$y = x_1 + x_2$	$y = \overline{x_1 + x_2}$	$y = x_1 \cdot x_2$	$y = \overline{x_1 \cdot x_2}$
Назва функції		Логічне додавання (диз'юнкція)	Заперечення логічного додавання (стрілка Пірса)	Логічне множення (кон'юнкція)	Заперечення логічного множення (штрих Шеффера)
Графічне позначення					
Схема					

З таблиці видно, що, наприклад, функція АБО є істинною, якщо істинною є хоча б одна зі змінних, що входять до неї, а функція I – лише у разі істинності обох змінних.

Однією з інтерпретацій булевих операцій є схеми, що складаються з ключів, джерела напруги E і лампочки L .

Для реалізації операції диз'юнкції двох змінних x_1 і x_2 використовують два паралельно з'єднаних нормально розімкнених ключі (рисунок 1.2, а).

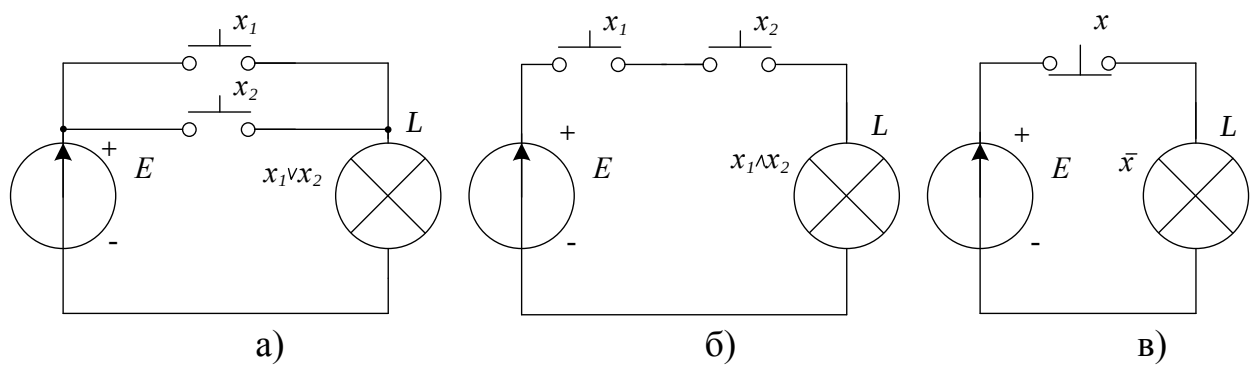


Рисунок 1.2 – Інтерпретація булевих операцій: а) – диз'юнкція;
б) – кон'юнкція; в) – заперечення

У разі натискання будь-якого ключа ($x_1 = 1$ або $x_2 = 1$) або обох разом лампочка світиться (значення 1). Для реалізації операції кон'юнкції двох змінних x_1 і x_2 застосовують два послідовно з'єднаних нормально розімкнутих ключі (рисунок 1.2, б). У разі натискання обох кнопок одночасно ($x_1 = x_2 = 1$) лампочка світиться (значення 1). Для реалізації операції заперечення застосовується нормально замкнутий ключ (рисунок 1.2, в). За умови $x = 0$ – ключ замкнутий, і лампочка світиться (значення 1); за $x = 1$ ключ розмикається, і лампочка не світиться (значення 0).

Якщо вхідні сигнали логічних елементів змінювати з часом, як це показано на рисунку 1.3, можна отримати часові діаграми роботи, що відповідають динамічному режиму роботи елементів.

Мінімальний набір логічних елементів, що реалізують деякі прості логічні функції і за наявності необмеженої кількості яких можна реалізувати як завгодно складну логічну функцію, називають функціонально повною системою логічних елементів або базисом. Найбільш відомими функціонально повними системами є:

1. елементи, що реалізують функції алгебри Буля — І, АБО, НЕ;
2. елемент, що реалізує функцію штрих Шеффера — І-НЕ;
3. елемент, що реалізує функцію стрілка Пірса — АБО-НЕ.

Як видно з таблиці істинності логічних елементів (таблиця 1.2), елемент І для одиниць є інверсією АБО для нулів.

Тобто, якщо для прямих значень

$$y = x_1 \cdot x_2,$$

то для інверсних

$$\bar{y} = \bar{x}_1 + \bar{x}_2.$$

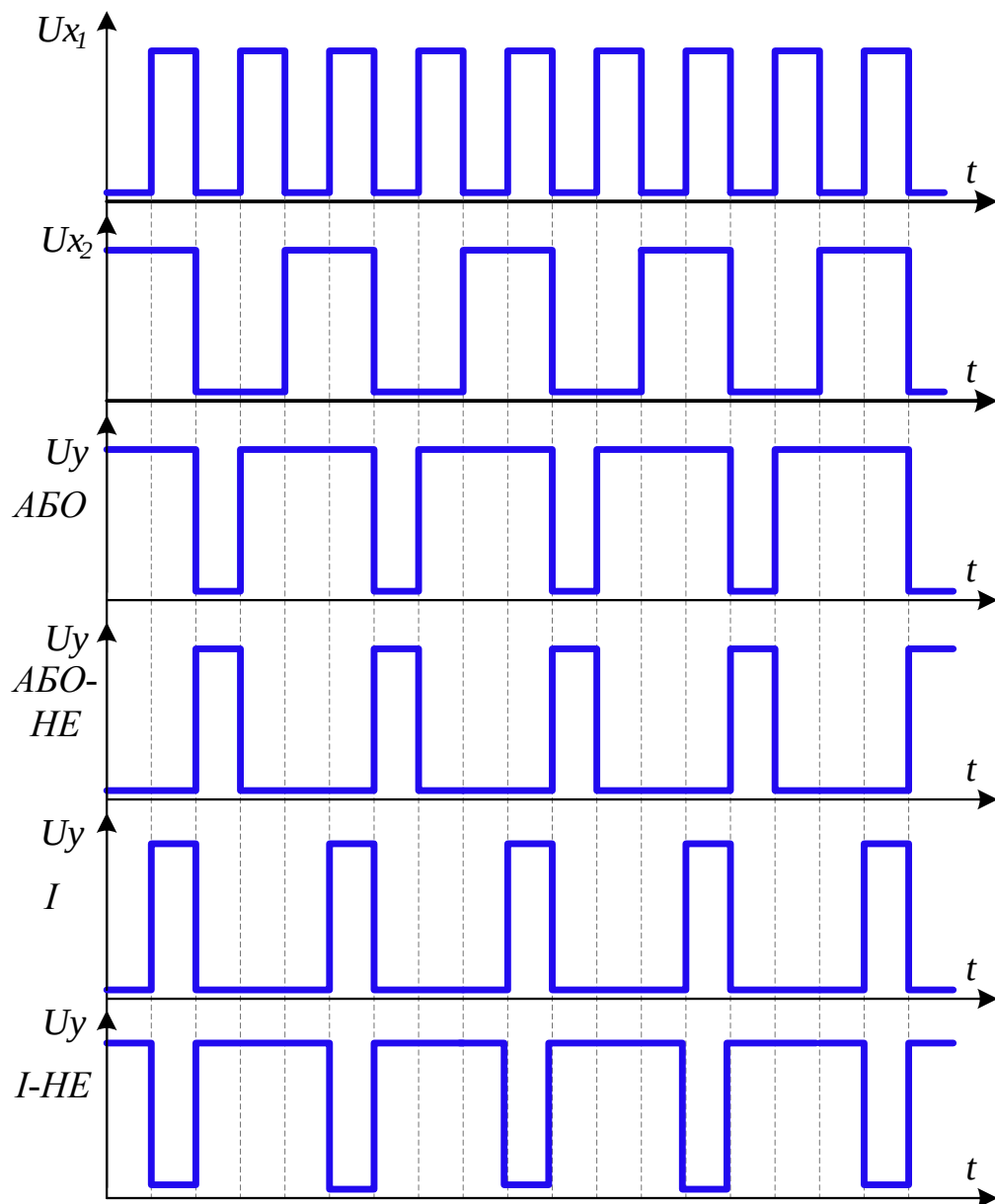


Рисунок 1.3 – Часові діаграми роботи деяких двовходових логічних елементів

Таблиця 1.2 – Таблиця істинності логічних елементів І та АБО

Вхідні змінні		Функція у	
x_1	x_2	І	АБО
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Для елемента І-НЕ можна записати

$$y = \overline{x_1 \cdot x_2} = \overline{x_1} + \overline{x_2},$$

а для елемента АБО-НЕ

$$= \overline{x_1 + x_2} = \overline{x_1} \cdot \overline{x_2}.$$

Наведені вище приклади називають правилами де Моргана.

Якщо задати, наприклад, відповідно $x_1 = 1$ або $x_2 = 0$, то матимемо

$$= \overline{x_1 \cdot x_2} = \overline{x_1 \cdot 1} = \overline{x_1},$$

або

$$= \overline{x_1 + x_2} = \overline{x_1 + 0} = \overline{x_1}.$$

Тобто, якщо на всі входи цих елементів подати одну й ту саму змінну, матимемо інвертори.

1.2 Послідовність виконання практичної роботи

1. Вибрати елементарні логічні функції та значення логічних змінних згідно з варіантом практичної роботи (таблиця 1.5). Останні дві цифри індивідуального навчального плану здобувача є варіантом практичної роботи.

2. Побудувати комбінаційний логічний пристрій, що реалізує логічну функцію: яка має наступний загальний вигляд:

$$= f_1[f_2(a_1, a_2), f_3(a_3, a_4), f_4(a_1, a_3)].$$

де f_1, f_2, f_3, f_4 – елементарні логічні функції; a_1, a_2, a_3 – логічні змінні;

3. Спростити (мінімізувати) запропоновану до реалізації функцію.

4. Побудувати логічну схему комбінаційного пристрою, що реалізує задану функцію.

5. Скласти таблиці істинності заданої функції.

6. Побудувати часові діаграми роботи комбінаційного пристрою.

1.3 Приклад виконання практичної роботи

1. Вибираємо варіант практичної роботи.

Вибір варіанта для номера індивідуального навчального плану здобувача вищої освіти №01-22-236. Дві останні цифри – 36, отже, відповідно до таблиці 1.3

Таблиця 1.3 – Приклад вибору варіантів практичної роботи

№ інд. навч. плану		0	1	2	3	4	5	6	7	8	9
дес.	од.										
a_1		x_1	1	$\overline{x_1}$	x_1	x_1	x_1	$\overline{x_1}$	x_1	1	$\overline{x_1}$
a_2		x_2	$\overline{x_2}$	0	x_2	x_2	1	x_2	$\overline{x_2}$	x_2	x_2
a_3		1	x_3	x_3	1	$\overline{x_3}$	x_3	1	x_3	$\overline{x_3}$	1
a_4		$\overline{x_4}$	x_4	x_4	$\overline{x_4}$	1	$\overline{x_4}$	x_4	0	x_4	x_4
	f_1	I		АБО		I-НЕ		АБО-НЕ		I	
	f_2	АБО	I-НЕ		АБО-НЕ		I		I-НЕ		I
	f_3	I			АБО			I-НЕ			

маємо:

зі стовпця 3: $a_1 = x_1$, $a_2 = x_2$, $a_3 = 1$, $a_4 = \overline{x_4}$.

зі стовпця 6: $f_1 = \text{АБО} - \text{НЕ}$, $f_2 = \text{I}$, $f_3 = \text{I} - \text{НЕ}$.

2. Будуємо комбінаційний логічний пристрій, який має загальний вигляд:

$$y = f_1[f_2(a_1, a_2), f_3(a_3, a_4)].$$

де f_1, f_2, f_3 – елементарні логічні функції;

a_1, a_2, a_3 – логічні змінні; (див. п. 1.).

Тоді логічна функція, яку необхідно реалізувати, має вигляд:

$$y = x_1 \cdot x_2 + 0 \cdot \overline{x_4}.$$

Спростуємо (мінімізуємо) отриману функцію. Оскільки $a_3 = 1$, то

$$f_3 = \overline{1 \cdot \overline{x_4}} = \overline{x_4} = x_4$$

Отже функція спрощується до

$$y = x_1 \cdot x_2 + x_4.$$

3. Будуємо логічну схему комбінаційного пристрою, що реалізує задану функцію. Для побудови пристрою можна застосовувати двовходові логічні елементи 2-I, 2-АБО, 2-АБО-НЕ, 2-I-НЕ та інвертор НЕ.

Припускаємо, що на входи пристрою надходять лише прямі сигнали вхідних змінних (тобто x_1, x_2, x_3, x_4); для отримання, наприклад, $\overline{x_1}$ необхідно застосувати інвертор НЕ.

З аналізу виразу видно, що для реалізації мінімізованої функції необхідно:

- двовходовий елемент I для реалізації функції $f_2 = x_1 \cdot x_2$.
- двовходовий елемент АБО-НЕ для реалізації функції $f_1 = \overline{f_2 + x_4}$.

Логічну схему комбінаційного пристрою, що реалізує задану функцію, наведено на рисунку 1.3.

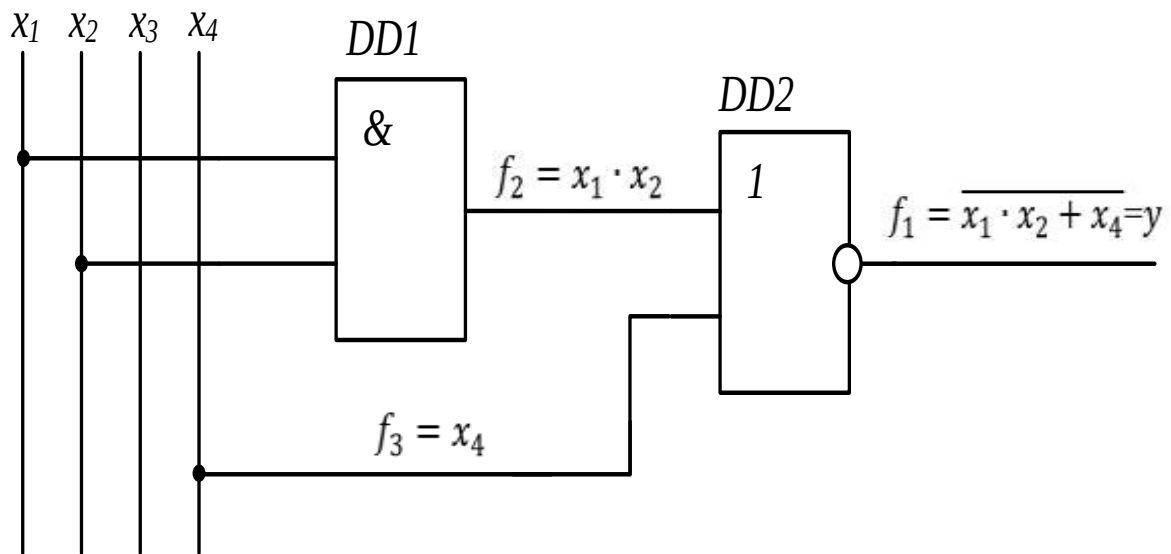


Рисунок 1.3 – Комбінаційний пристрій. Схема логічна

4. Складаємо таблиці істинності заданої функції

Таблиця 1.4 – Таблиця істинності логічної функції

Комбінація вхідних змінних	x_1	x_2	x_4	$f_2 = x_1 \cdot x_2$	$f_3 = x_4$	$f_1 = \overline{f_2 + f_3} = y$
1	0	0	0	0	0	1
2	0	0	1	0	1	0
3	0	1	0	0	0	1
4	0	1	1	0	1	0
5	1	0	0	0	0	1
6	1	0	1	0	1	0
7	1	1	0	1	0	0
8	1	1	1	1	1	0

5. Будуємо часові діаграми роботи комбінаційного пристрою (рис. 1.4).

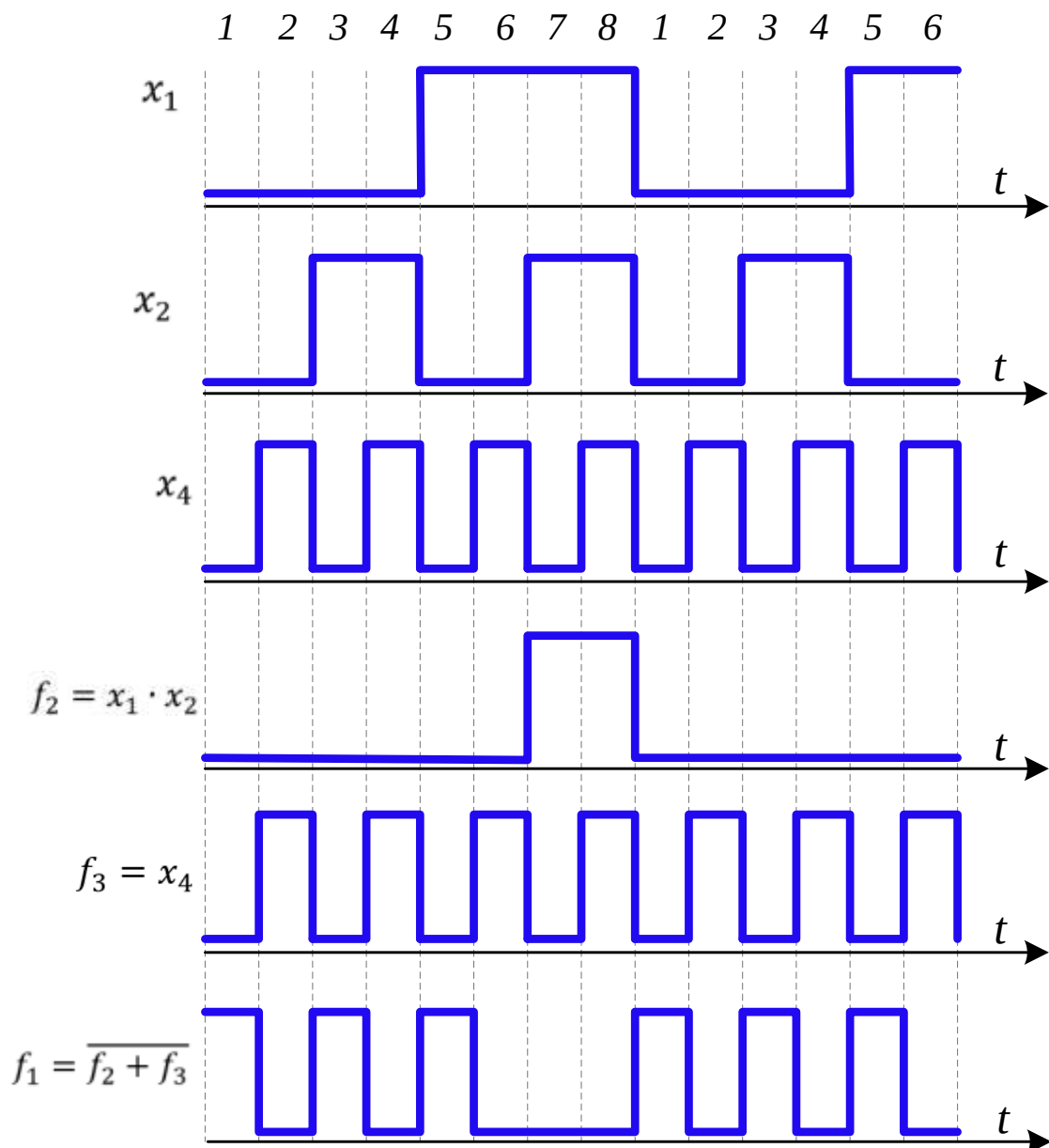


Рисунок 1.4 – Часові діаграми роботи комбінаційного пристрою

З аналізу таблиці істинності та часових діаграм видно, що логічна функція набуває значення «істинна» ($y = 1$) у випадку комбінацій вхідних змінних 1, 3 та 5.

1.4 Зміст звіту

1.4.1. Звіт з практичної роботи має бути виконаний на аркушах формату А4; відповідати варіанту практичної роботи.

1.4.2. Звіт має містити: назву практичної роботи, її мету та короткі теоретичні відомості.

1.4.3. В розділі «Результати виконання практичної роботи» здобувач має мінімізувати отриману логічну функцію, побудувати комбінаційний

логічний пристрій, скласти таблицю істинності заданої функції та побудувати часові діаграми роботи комбінаційного пристрою.

1.4.4. Написати висновки до практичної роботи.

Таблиця 1.5 – Варіанти завдань до практичної роботи

№ інд. навч. плану		0	1	2	3	4	5	6	7	8	9
Дес.	Од.										
a_1		x_1	1	x_1	$\overline{x_1}$	x_1	x_1	$\overline{x_1}$	x_1	1	$\overline{x_1}$
a_2		x_2	$\overline{x_2}$	0	x_2	x_2	1	x_2	$\overline{x_2}$	x_2	x_2
a_3		1	x_3	x_3	0	$\overline{x_3}$	x_3	0	x_3	$\overline{x_3}$	1
a_4		$\overline{x_4}$	x_4	$\overline{x_4}$	x_4	1	$\overline{x_4}$	x_4	0	x_4	x_4
	f_1	I		АБО		I-НЕ		АБО-НЕ		I	
	f_2	АБО	I-НЕ		АБО-НЕ		I		I-НЕ		I
	f_3	I			АБО			I-НЕ			
	f_4	I				АБО					

Контрольні запитання

1. Поясніть, на чому базується аналіз роботи цифрових пристроїв.
2. Вкажіть, як оцінюють подію в алгебрі логіки.
3. Поясніть, що таке логічна (двійкова) змінна та логічна функція.
4. Які найпростіші логічні функції ви знаєте?
5. Що таке штрих Шеффера?
6. Що таке стрілка Пірса?
7. Наведіть таблицю істинності для елемента НЕ.
8. Наведіть таблицю істинності для елемента АБО.
9. Наведіть контактно-релейну схему заміщення для елемента НЕ.
10. Наведіть контактно-релейну схему заміщення для елемента АБО.
11. Наведіть контактно-релейну схему заміщення для елемента I.

Практична робота № 2

ВЕАМ РОБОТ НА ОСНОВІ НЕЙРОННИХ КІЛ М. ТІЛЬДЕНА. КОНЦЕПЦІЯ. БУДОВА. ПРИНЦИП КЕРУВАННЯ

Мета роботи: ознайомитись з будовою ВЕАМ роботів, побудувати та дослідити схему керування рухом найпростішого робота.

2.1 Теоретичні відомості [6, 7, 8, 9, 10, 11]

На початку 90-х років ХХ століття Марк Тільден запропонував інноваційний підхід до конструювання роботів. Він розробив концепцію ВЕАМ роботів.

ВЕАМ є аббревіатурою від англійських слів Biology, Electronics, Aesthetics, Mechanics (біологія, електроніка, естетика, механіка) – категорія роботів, під час побудови яких використовують прості аналогові кола (наприклад, компаратори) замість мікропроцесорів з метою досягти простого дизайну, який жертвує гнучкістю заради надійності та ефективності виконання певного завдання. Однак існують винятки, що використовують не тільки аналогові кола. Роботи ВЕАМ зазвичай являють собою набір вищезгаданих аналогових кіл (які повторюють біологічні нейрони), що дозволяють простому роботу взаємодіяти з довкіллям.

Розглянемо ці чотири основні поняття:

- **БІОЛОГІЯ.** Багато аспектів дизайну та функцій ВЕАМ роботів запозичуються з живої природи, в них, як правило, не використовуються такі механічні винаходи, як підшипники, колеса. Мільярди років проектування, виробництва і польових випробувань біологічних об'єктів змушують поглянути на них з усією серйозністю і захопленням.

- **ЕЛЕКТРОНІКА.** Використовуються прості аналогові схеми, які дозволяють визначати їхню поведінку. Використання електронних компонентів абсолютно нестандартними способами. Чипи, призначені для посилення звуків, обробки сигналів тощо, підключаються по-новому, щоб керувати двигунами, обслуговувати датчики та ставати основою таймерів і нейронних кіл.

- **ЕСТЕТИКА.** Робот має презентабельно виглядати. Марк Тільден і його колеги – розробники ВЕАМ-роботів – завжди наголошували на тому, що елегантність у дизайні робота слугує еволюційній меті: елегантна конструкція надихає інших розробників, що часто сприяє реплікації та поліпшенню конструкції робота. Крім того, те, що зроблено красиво, працює краще.

• **МЕХАНІКА.** Механіка потрібна для переміщення робота. Механіка використовує той самий підхід, який використовується під час розроблення електронної частини. Часто одне оригінальне механічне рішення в дизайні робота може скоротити купу електроніки та зробити робота простішим і ефективнішим. Багато ВЕАМ-роботів використовують вільно сформовані схеми, в яких компоненти спаяні безпосередньо один з одним, а не на друкованій платі. Це дає змогу знизити вартість і вагу робота, а також демонструє технічну майстерність проєктанта.

Класифікація ВЕАМ-роботів

- аудіотропи (реагують на звуки);
- фототропи (реагують на світло);
- радіотропи (реагують на радіочастоти);
- термотропи (реагують на теплове випромінювання).

Роботи ВЕАМ мають безліч механізмів руху і позиціонування, у зв'язку з чим можуть бути виділені різні категорії за динамікою, наприклад:

- Сіттер: нерухомі роботи, мають пасивне призначення.
- Маяки: передають сигнал (зазвичай навігаційний) іншим роботам ВЕАМ.
- Паммери: відображають світлове шоу.
- Орнаменти: інші роботи.
- Сквірмери: нерухомі роботи, що виконують будь-яку дію (зазвичай рух кінцівок).
- Магботи: використовують магнітні поля для своєї роботи.
- Турботи: котяться з використанням кінцівок.
- Землеміри: переміщують частину тіла вперед, в той час, як інша залишається на місці.
- Гусеничні роботи: використовують гусениці.
- Джампери: роботи, що відштовхуються від поверхні для пересування.
- Віброботи: пересуваються за рахунок вібрації.
- Спрінгботи: рухаються стрибками в певному напрямку.
- Ролери: роботи, що рухаються перекатами.
- Мініболи: переміщують центр маси, за рахунок чого рухається сферичне тіло робота.
- Вокери: роботи, що рухаються з використанням ніг.
- Рухомий мотор: використовують мотори для руху ніг (зазвичай 3 або більше мотори).
- Рухомі м'язи: використовують дроти зі сплаву нікелю з титаном для руху ніг.

- Свіммер: роботи, що рухаються в рідині.

Концепція BEAM роботів. Основною відмінністю концепції, запропонованої Марком Тільденом, було створення роботів, здатних незалежно від людини, «жити самостійним життям».

У роботах BEAM не використовуються мікропроцесори, а використовуються прості аналогові кола. Їх не треба програмувати, що дозволяє знизити потребу в енергії та зводить до мінімуму можливість відмови. Ці роботи прості у дизайні, та мають низьку вартість.

BEAM роботи часто імітують поведінку живих істот: комах, рептилій, простих ссавців. Ідея Тільдена полягала в тому, щоб створити механічні тварини, здатні жити без допомоги людини, які шукають для свого існування джерела енергії, як це роблять живі істоти, коли шукають їжу.

Марк Тільден запропонував три закони для BEAM роботів:

1. Робот має захищати своє існування за будь-яку ціну.

Багато типів BEAM роботів мають різні датчики, що дозволяє їм виявляти різні перешкоди. Наприклад, такі датчики можуть бути виконані у вигляді механічних щупалець (як у комах або невеликих ссавців), щоб запобігти зіткненню робота з перешкодою та дозволити її обійти.

2. Робот має отримувати доступ до джерел енергії та підтримувати свою роботоздатність.

В одних випадках традиційне оснащення сучасних роботів акумуляторною батареєю дуже неефективне, тому що це збільшує в кілька разів його масу. В інших, часто використовуваних конструкціях роботів, застосовують потужніші двигуни, що ускладнює та здорожчує проекти. Це суперечить концепції BEAM, згідно з якою система має бути якомога простішою і дешевшою. Саме тому BEAM роботи часто оснащуються фотоелектричними модулями (сонячними батареями), які забезпечують незалежне та надійне джерело живлення, що дозволяє їм тривалий час автономно працювати.

3. Робот має знаходити ефективне джерело енергії.

Функція пошуку джерела енергії у робототехнічних конструкціях проектується за аналогією з поведінкою простих живих істот, які прагнуть поліпшення умови їхнього власного існування. У роботі використовуються фотоелементи для виявлення світла та рівня освітленості, що дозволяє керувати роботом та спрямовувати його у бік найбільшого освітлення, де він зможе отримати більше енергії для своєї системи живлення.

Крім того, на думку Марка Тільдена, розробники BEAM роботів мають дотримуватися трьох принципів:

- використовувати мінімальну кількість компонентів у робототехнічній конструкції та прагнути до спрощення моделі;
- використовувати елементи та обладнання, що зменшує вартість конструкції (наприклад, як сонячні батареї можна використовувати частини від старих непотрібних калькуляторів);
- використовувати поновлювані джерела енергії для живлення робототехнічних конструкцій (наприклад, можна використовувати сонячну енергію).

Штучні нейрони. Принципи ВЕАМ роботів базуються на здатності конструкційної моделі реагувати на зовнішні стимули. Механізм симуляції поведінки робота за допомогою штучних нейронів був винайдений Марком Тільденом, нейронне коло Тільдена – система управління рухом роботів. Для управління ВЕАМ роботами замість мікропроцесора використовується штучна «нервова мережа». Така мережа складається з безлічі штучних нейронів. У своїй найпростішій формі ВЕАМ нейрон є колом затримки імпульсу і складається з резистора, конденсатора та інвертора (прилад, в якому високий рівень вхідної напруги перетворюється на низький, і навпаки) (рис. 2.1).

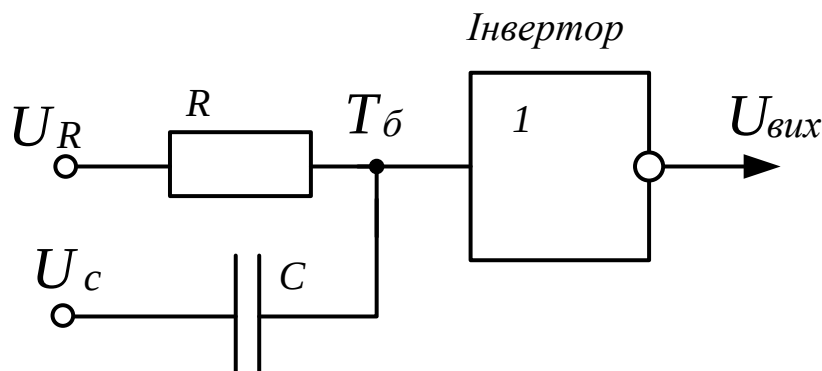


Рисунок 2.1 – ВЕАМ нейрон

ВЕАМ нейрон змінює стан вихідної напруги на протилежний вхідному, залежно від величини сигналів U_C та U_R .

Затримка у зміні стану $U_{вих}$ залежить від часу очікування T (тобто тривалості вихідних імпульсів), значення якого залежить від опору резистора, ємності конденсатора та порога інвертора. Ця система чутлива до перешкод на вході інвертора та коливань напруги живлення, що впливає на T .

Це описується таким рівнянням:

$$T = RC \log_e(1/((U_{th} - U_{min})/(U_{max} - U_{min}))),$$

де T – час затримки сигналу або тайм-аут (тривалість вихідних імпульсів), с;
 R – опір, МОм;
 C – ємність, мФ;
 U_{th} – порогове значення напруги перемикавання інвертора, В;
 U_{max} – максимальна напруга, В;
 U_{min} – мінімальна напруга, В;

Також можна використовувати простішу формулу:

$$T = RC.$$

М. Тільден визначив два типи нейронів:

- Nv нейрон («збудливий», М. Тільден назвав його «нерв»), використовується для управління двигунами, може конструюватися за схемою $Nv+$ та $Nv-$.
- Nu нейрон («гальмувальний, гальмівний»), використовується для обробки сигналів від сенсорів, може конструюватися за схемою $Nu+$ та $Nu-$.

Короткий опис можливих конфігурацій системи з сигналами напруги у вибраних точках подано на рисунках 2.2 – 2.5.

Нейрон $Nv-$.

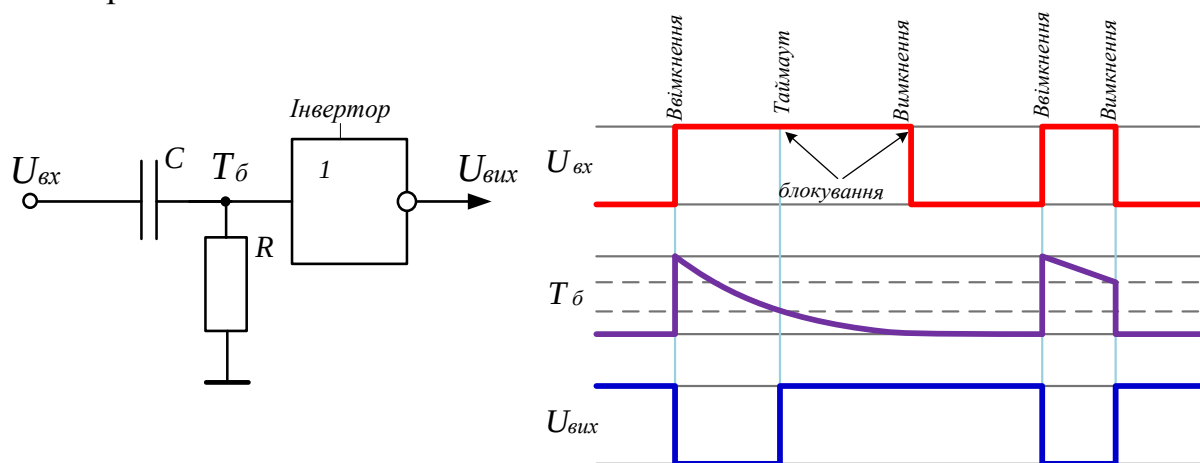


Рисунок 2.2 – Нейрон $Nv-$

Під час подачі позитивної напруги на вхід відбувається миттєва активація пристрою і на виході встановлюється 0 (земля). Для скидання необхідно припинити подавати позитивний сигнал на вхід, миттєва деактивація пристрою і на виході виходить позитивний сигнал.

Нейрон $Nv+$.

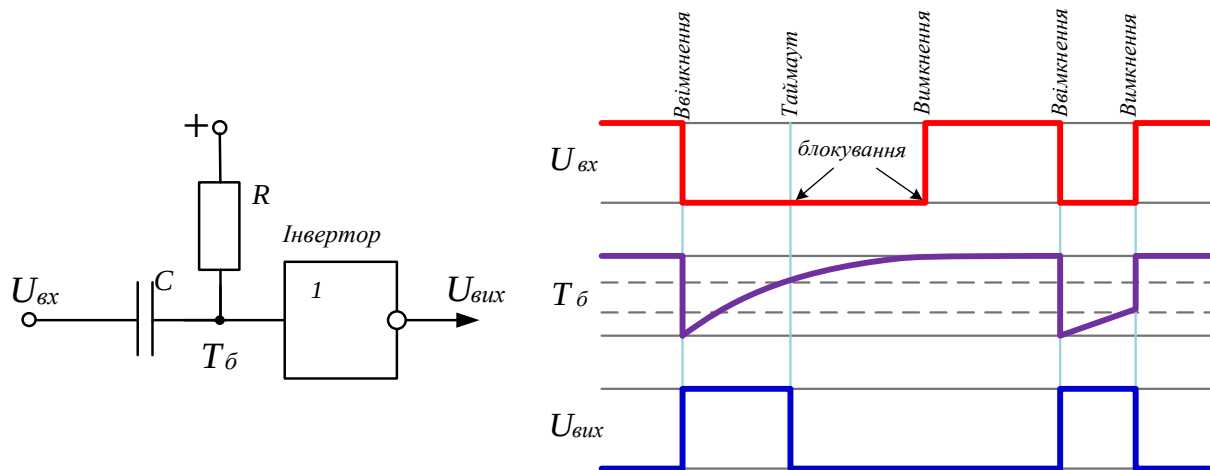


Рисунок 2.3 – Нейрон $Nv+$

Нейрон $Nu-$ («гальмувальний») – нейрон реагує на вхідні зміни після деякої затримки сигналу та перемикає вихід після затримки сигналу.

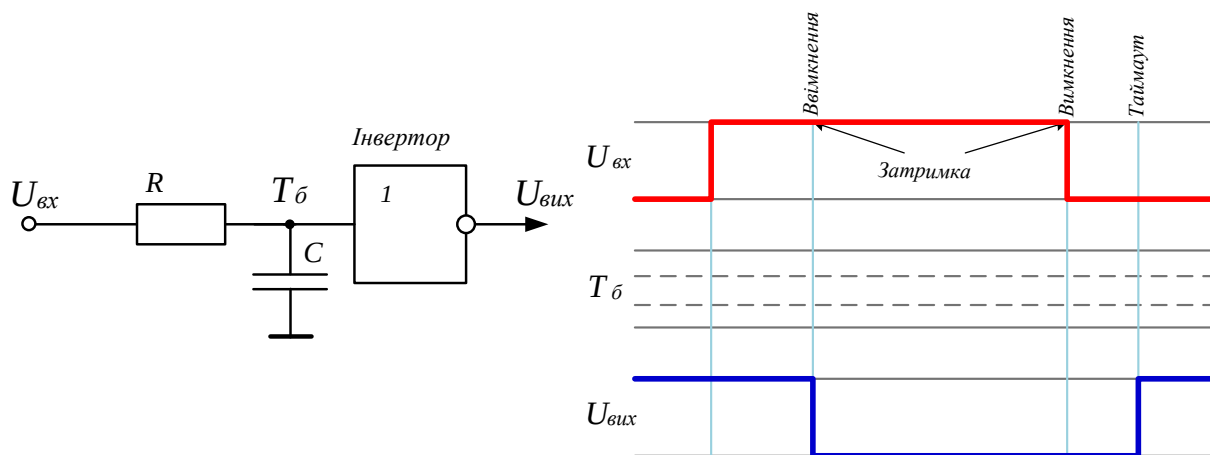


Рисунок 2.4 – Нейрон $Nu-$

Схема складається з інтегрувального RC кола (що подовжує імпульс) та інвертора. В такому випадку вихідною напругою $U_{вих}$ є напруга на конденсаторі C . У разі ненасиченого конденсатора C вихідна напруга $U_{вих}$ дорівнює нулю. Під час підвищення вхідної напруги $U_{вх}$ конденсатор C почне насичуватися, що відбуватиметься за експонентою, в цьому випадку, і напруга підвищуватиметься від нуля до максимуму за експонентою.

Нейрон Nu+.

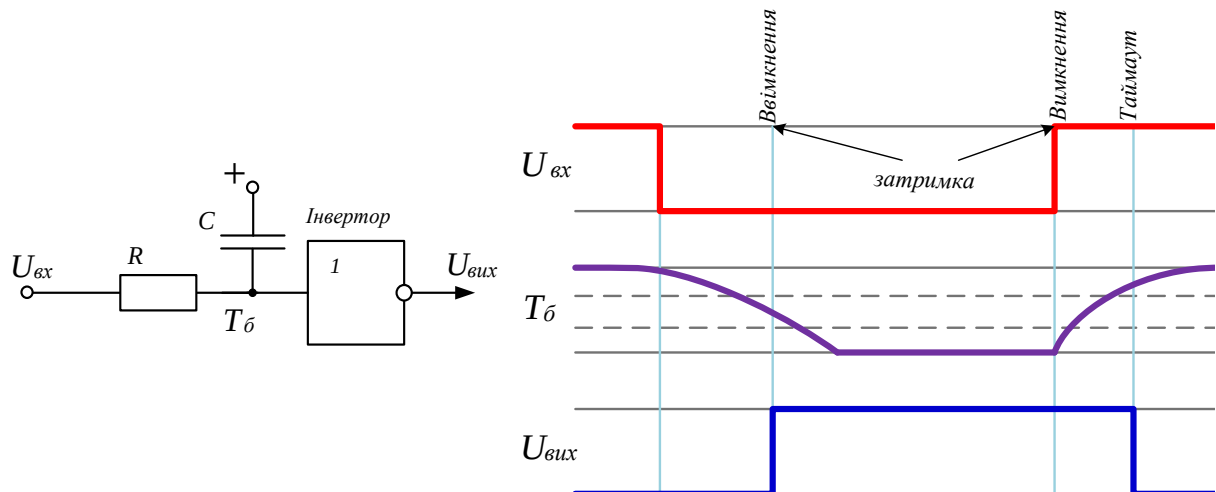


Рисунок 2.5 – Нейрон Nu+

2.2 Побудова найпростішого аналогового робота

Сьогодні одне з перших практичних завдань пов'язане з розробкою конструкції ВЕАМ робота, що буде здатний самостійно рухатись певною траєкторією (яка називається «чорною лінією»), є конструкція, електричну схему якої подано на рисунку 2.6.

Розглянемо процес створення такого робота. Необхідні матеріали:

Мікросхема L272M – 1 шт.

Інфрачервоний сенсор TCRT5000 – 2 шт.

Резистори: 510 Ом – 2 шт, 1 кОм – 3 шт, 10 кОм – 2 шт.

Змінний резистор: 10 кОм – 2 шт.

Конденсатор електролітичний: 470 мкФ – 1 шт.

Світлодіод: червоний – 1 шт, зелений – 2 шт.

Вимикач – 1 шт.

Елементи живлення – 3,7 В.

Мотор редуктор з колесом – 2 шт.

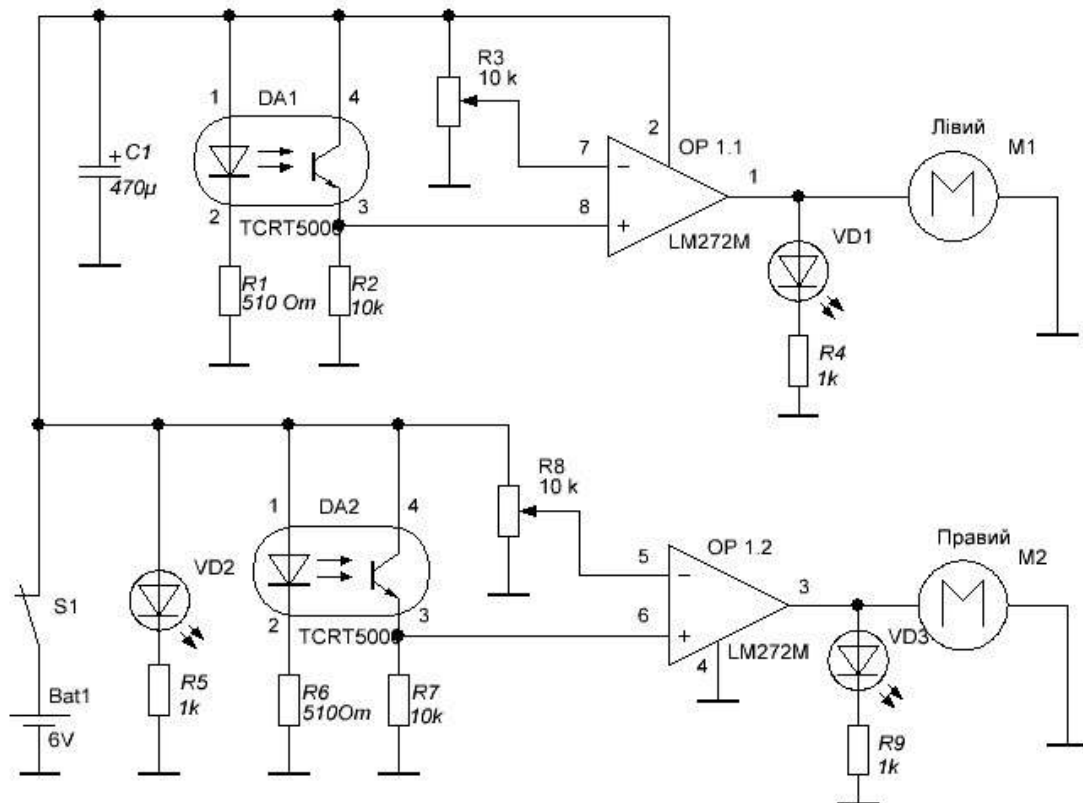


Рисунок 2.6 – Схема електрична принципова ВЕАМ робота

Схему електричну комутаційну усіх компонентів ВЕАМ робота подано на рисунку 2.7.

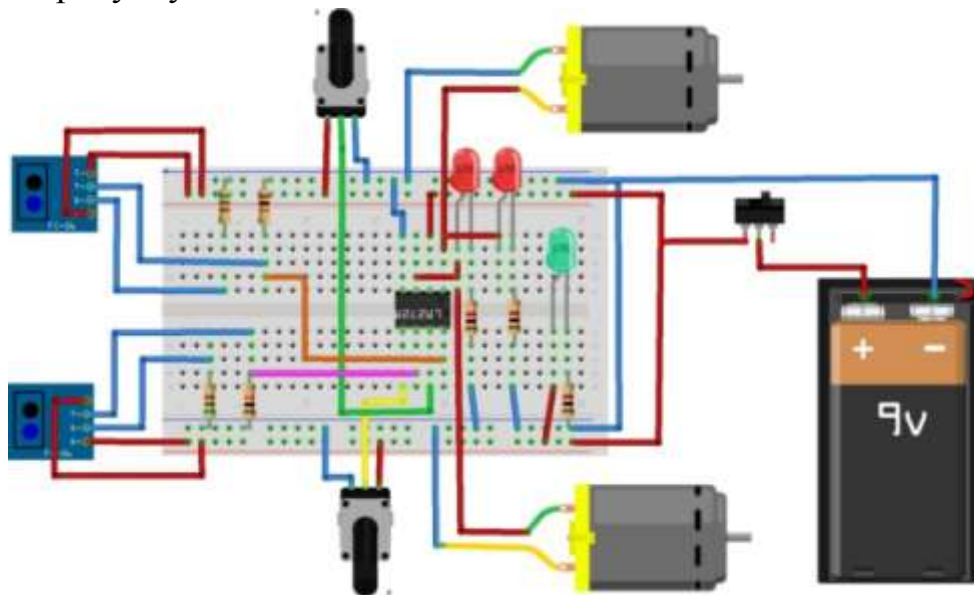


Рисунок 2.7 – Схема електрична комутаційна ВЕАМ робота

Нейромережа, зображена на рисунку 2.7, складена на мікросхемі LM272M. Прямий вхід кожної частини мікросхеми (+) підключається до

інфрачервоного відбивача DA1, DA2, який використовується як детектор «чорної лінії». Інверсний вхід (-) з'єднаний зі змінними резисторами R3, R8, які використовуються для налаштування чутливості датчиків DA1 та DA2.

Інфрачервоний сенсор TCRT5000 працює на принципі відбиття світла.

Цей сенсор відображення поєднує в одному корпусі інфрачервоний передавач (IR світлодіод 950 нм) та приймач (фототранзистор), які розділені перегородкою (рис. 2.8). Цей прилад випромінює світло в інфрачервоному спектрі та приймає відбиток цього інфрачервоного сигналу від поверхні. Залежно від типу та кольору поверхні змінюється інтенсивність відбитого сигналу.

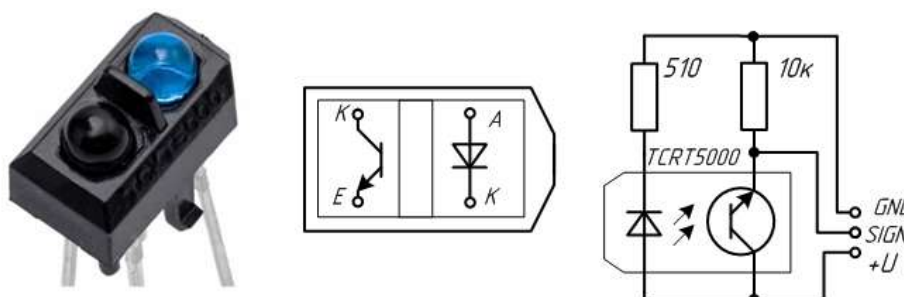


Рисунок 2.8 – Інфрачервоний датчик TCRT5000 та його схема підключення

Сенсор реагує на наявність перешкод, що відбивають світло, на відстані до 15 мм і зміну кольору поверхні, що відбиває промені. Може використовуватися як датчик швидкості обертання, датчик відстеження лінії в роботах, датчик наближення, оптичний кінцевик і т. д. TCRT5000 має вбудований фільтр денного світла, завдяки чому не реагує на стороннє засвітлення (рис. 2.9).

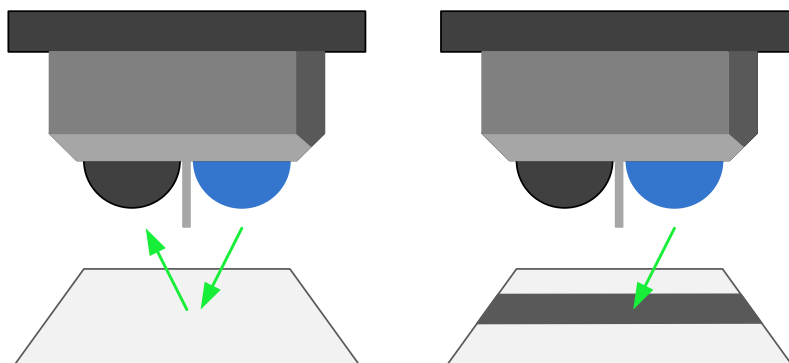


Рисунок 2.9 – Схема роботи TCRT5000

Оскільки TCRT5000 має транзисторний вихід, з датчика можна знімати аналоговий сигнал, що змінює показання залежно від відстані до поверхні перед ним та її кольору. Характеристики сенсора наведено в таблиці 2.1.

Таблиця 2.1 – Характеристики сенсора TCRT5000

Робочий діапазон відстаней	0.5...15 мм
Довжина хвилі ІЧ випромінювання	950 нм
Максимальний постійний струм випромінювача (світлодіода)	60 мА
Спад напруги на випромінювачі (світлодіоді)	1.5 В
Максимальний постійний струм через транзистор	100 мА
Діапазон робочих температур	-25...+85 °С

Коли напруга на прямому вході більша, ніж на інверсному, вихідна напруга на мікросхемі приблизно дорівнює напрузі живлення. Світлодіод світиться, двигун працює. Якщо напруга, що надходить із датчика DA на прямий вхід нижча, ніж опорна напруга на інверсному вході (яка встановлюється за допомогою змінних резисторів R3 або R8), двигун постійного струму на виході зупиняється, світлодіод не світиться.

Таким чином, ця схема працює як компаратор напруги для управління двигунами, використовуючи роботу датчиків відстеження лінії, щоб встановити умови руху.

Компаратор – це операційний підсилювач без зворотного зв'язку з великим коефіцієнтом підсилення. Тому, якщо подати на один його вхід (наприклад, інверсний) якийсь постійний рівень опорної напруги, а на інший (прямий) вхід сигнал, що змінюється – вихідна напруга у нього зміниться стрибком, від мінімальної до максимальної в той момент, коли рівень вхідного сигналу перевищить рівень сигналу опорної напруги, встановленого на іншому вході, і навпаки. Компаратори мають два входи, прямий та інверсний, і, залежно від бажаного результату, опорна та порівнювана напруга можуть підключатися до будь-якого входу.

Якщо вхідна напруга на прямому вході перевищить напругу інверсного входу, вихідний транзистор компаратора відкривається, якщо стане нижче – закривається. Тобто компаратор порівнює напругу.

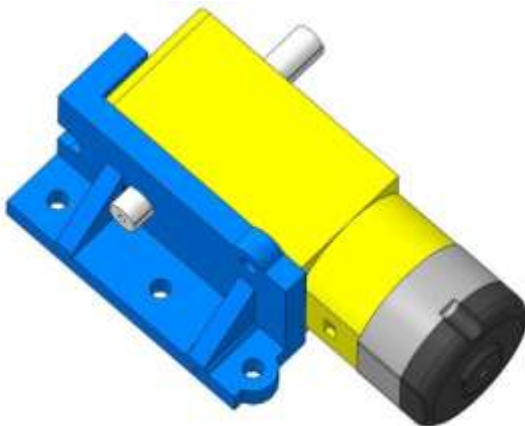
Суть призначення компаратора – порівнювати між собою дві напруги (сигнал), і видавати на виході напругу (сигнал) у тому випадку, коли сигнал на одному вході став більшим або меншим за рівень, встановлений опорною напругою іншого входу.

Ця схема керує двома двигунами постійного струму. Можна використовувати прості двигуни постійного струму з передатним відношенням 87:1, які можуть живитися від напруги живлення, що змінюється в широкому діапазоні від +3 В до +9 В (рисунок 2.10).



Рисунок 2.10 – Двигун постійного струму.

Схему можна змонтувати на друкованій платі, платформі для робота, використовуючи роздруковану на 3D принтері деталь (рисунок 2.11).



а)



б)

Рисунок 2.11 – Двигун та його кріплення: а) 3D модель; б) готова деталь

Провести калібрування датчиків освітлення за допомогою змінних резисторів. Запустити робота.

Розглянемо, як це працює.

Лівий та правий датчики, що відстежують чорну лінію, виявляють чорний колір (рис. 2.12).

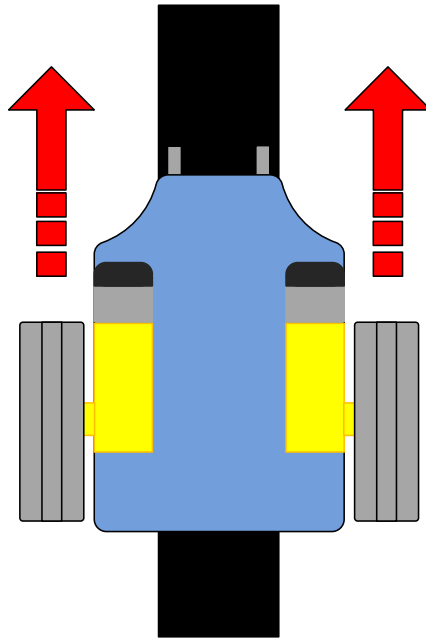


Рисунок 2.12 – Рух робота вперед

Це означає, що робот знаходиться в правильному положенні.

Обидва датчики видають високий рівень напруги (близької до напруги живлення). І тут напруга на прямих входах обох частин компаратора висока та обидва двигуни працюють, тобто робот рухатиметься вперед.

Правий датчик виявив біле поле. Щоразу, коли правий датчик виявляє біле поле, а лівий датчик виявляє чорну лінію, правий датчик видає низьку напругу (нижчу за опорну на одному із входів компаратора).

Це приводить до того, що один із компараторів видає нульову напругу на виході, лівий двигун припиняє роботу, а правий двигун продовжує працювати, оскільки лівий датчик все ще виявляє чорну лінію. Робот повернеться вліво (рис. 2.13).

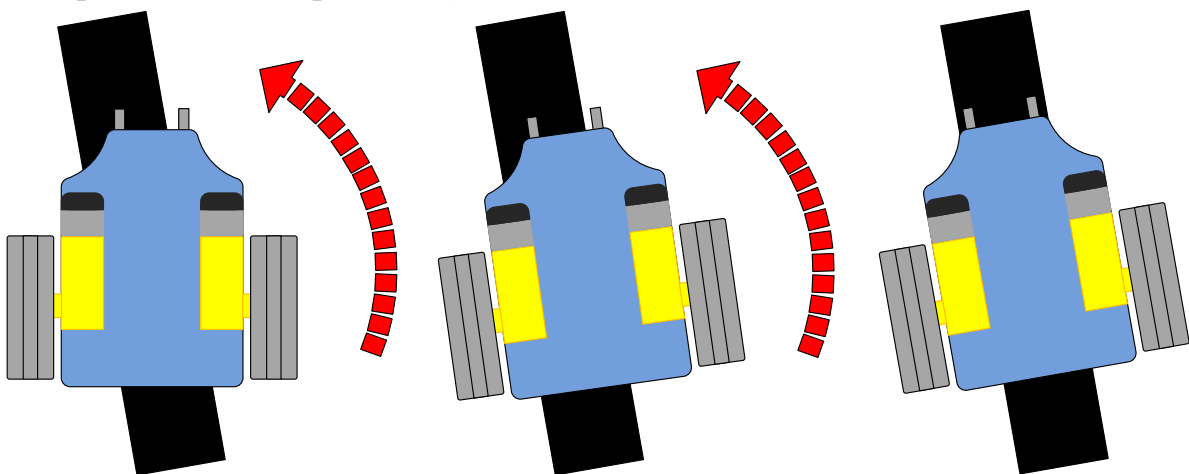


Рисунок 2.13 – Поворот робота, що переміщається вздовж лінії вліво

Лівий датчик виявляє білу поверхню, а правий датчик виявляє чорну лінію. Лівий датчик посилає низьку напругу на вихід. Двигун з правого боку зупиняється, а другий двигун все ще працює, тому що правий датчик ще виявляє чорну поверхню. Робот повернеться праворуч (рис. 2.14).

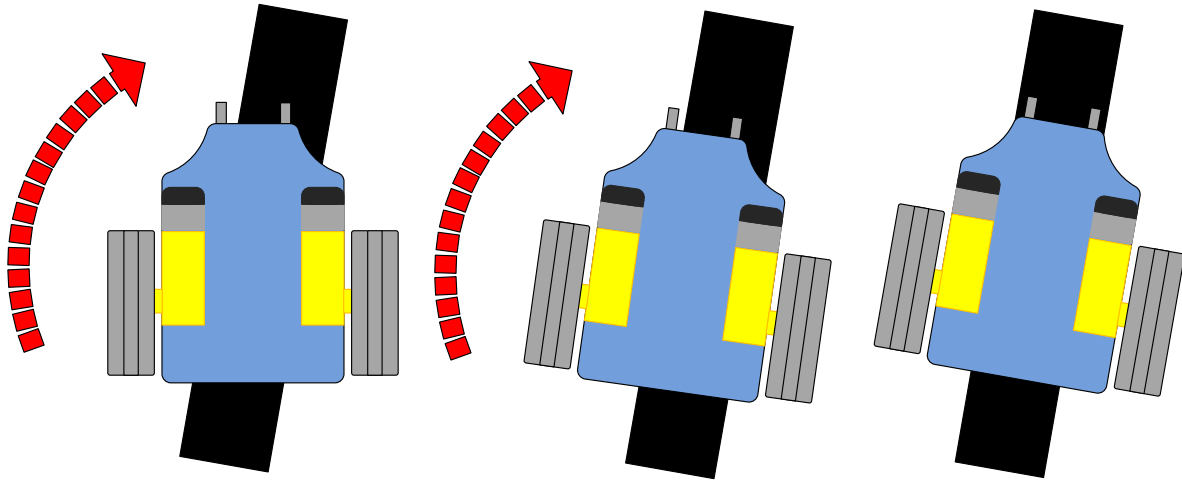


Рисунок 2.14 – Поворот робота, що переміщається вздовж лінії вправо

У разі, якщо обидва датчики відстеження лінії виявляють біле поле, це приводить до того, що обидва компаратори видають нульову напругу на виході, двигуни перестають працювати. Робот зупиниться.

2.3 Послідовність виконання практичної роботи

2.3.1 Відповідно до схеми, зображеної на рисунку 2.15, скласти схему електричну комутаційну на макетній платі.

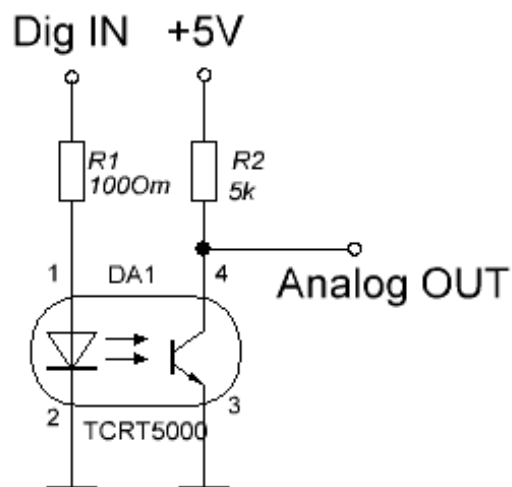


Рисунок 2.15 – Схема підключення TRCT5000

2.3.2 За допомогою інтегрованого середовища розробки Arduino IDE завантажити наведений нижче код, який видаляє статичний шум із даних, що надходять з ІЧ-датчика, та дослідити цей процес.

Для забезпечення статичного шуму використовуємо різні джерела освітленості, на кшталт денного світла, ліхтарика телефону тощо.

```
int a, b, c;
int Sensor_D = 3;
int Sensor_T = A3;

void setup() {
  Serial.begin(9600);
  pinMode(Sensor_D, OUTPUT);
}

void loop() {
  digitalWrite(Sensor_D, HIGH); // Вмикаємо ІЧ
світлодіод
  delayMicroseconds (500);      // Очікуємо 500 мкс.
  a = analogRead(Sensor_T);      // Зчитуємо значення
сигналу,
  digitalWrite(Sensor_D, LOW); // Вимикаємо ІЧ
світлодіод
  delayMicroseconds (500); // Очікуємо 500 мкс.
  b = analogRead(Sensor_T); // Зчитуємо значення
сигналу,
                                // який надходить на аналоговий
порт
  c = b - a;                    // Отримуємо значення
                                // (сигнал+шум) - шум
```

```

Serial.println("|||||||||||||||||||||||||||||||||||||");
);
    Serial.println("Signal+Shum || Shum ||
Signal");
    Serial.print("      ");
    Serial.print(b);
    Serial.print("      ");
    Serial.print(a);
    Serial.print("      ");
    Serial.println(c);    // Відображаємо значення в
                          // монітор порту

    delay(500);
}

```

2.3.3 Підключіть датчик TCRT5000 до відповідної плати згідно зі схемами, зображеними на рисунках 2.16 і 2.17.

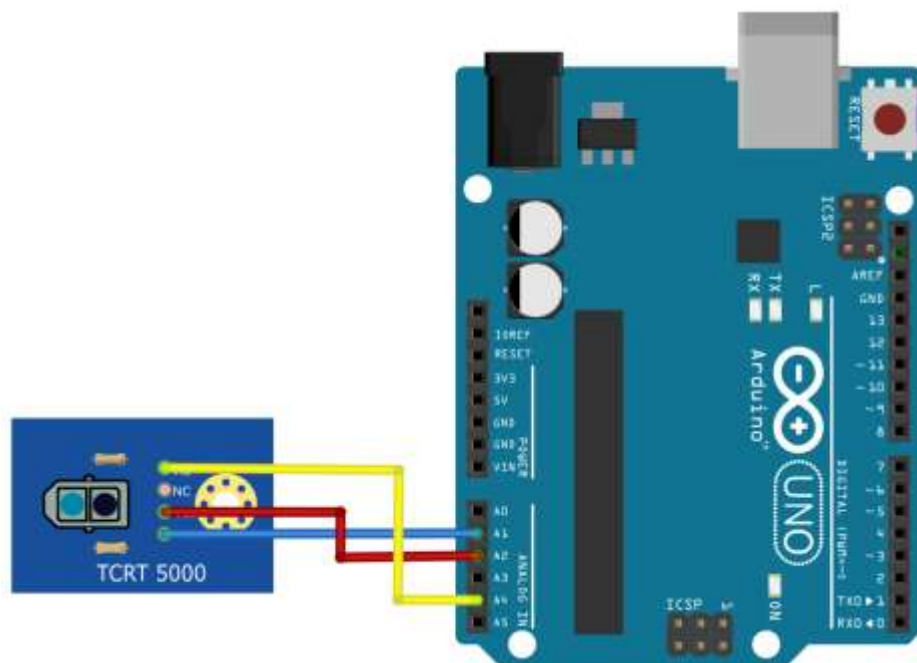
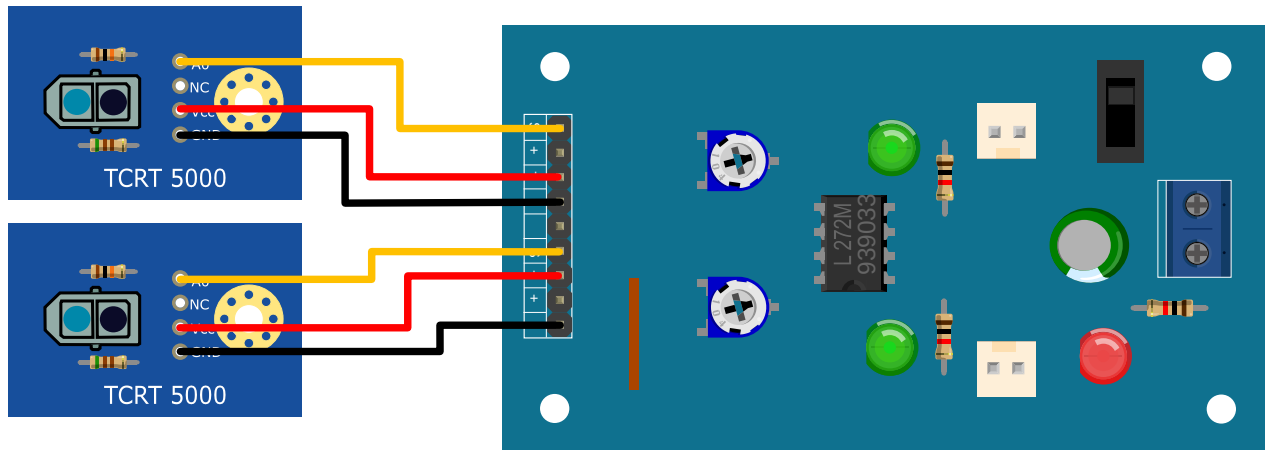


Рисунок 2.16 – Схема підключення модуля TRCT500 до плати Arduino
UNO



Рисунка 2.17 – Схема підключення модуля TRCT500 до плати керування BEAM-робота

2.3.4 За допомогою інтегрованого середовища розробки Arduino IDE завантажити код, наведений нижче, та дослідити роботу цього модуля. Проаналізувати дані, що надходять у монітор порту залежно від розміщення сенсора над чорною лінією та білому полі.

```
int Sensor_GND = 15;
int Sensor_POW = 17;
int Sensor_T = A4;
float volt = 0;

void setup() {
  Serial.begin(9600);
  pinMode(Sensor_POW, OUTPUT);
  pinMode(Sensor_GND, OUTPUT);
  pinMode(Sensor_T, INPUT);

  digitalWrite(Sensor_POW, HIGH);
  digitalWrite(Sensor_GND, LOW);
}

void loop() {
  int a = analogRead(Sensor_T);
  volt = map(a, 0, 1023, 0, 50);
  volt = volt / 10;
  Serial.print("Відносні одиниці ");
```

```

Serial.print(a);
Serial.print("  ||  ");
Serial.print("Напруга ");
Serial.print(volt);
Serial.println("  Вольт");
delay(100);
}

```

2.3.5 Відповідно до схеми електричної принципової та схеми електричної комутаційної, зображених на рис. 2.18, 2.19., (в деяких випадках використовуючи макетну плату) скласти конструкцію простого BEAM робота та дослідити принципи, що покладені в основу роботи такого аналогового робота.

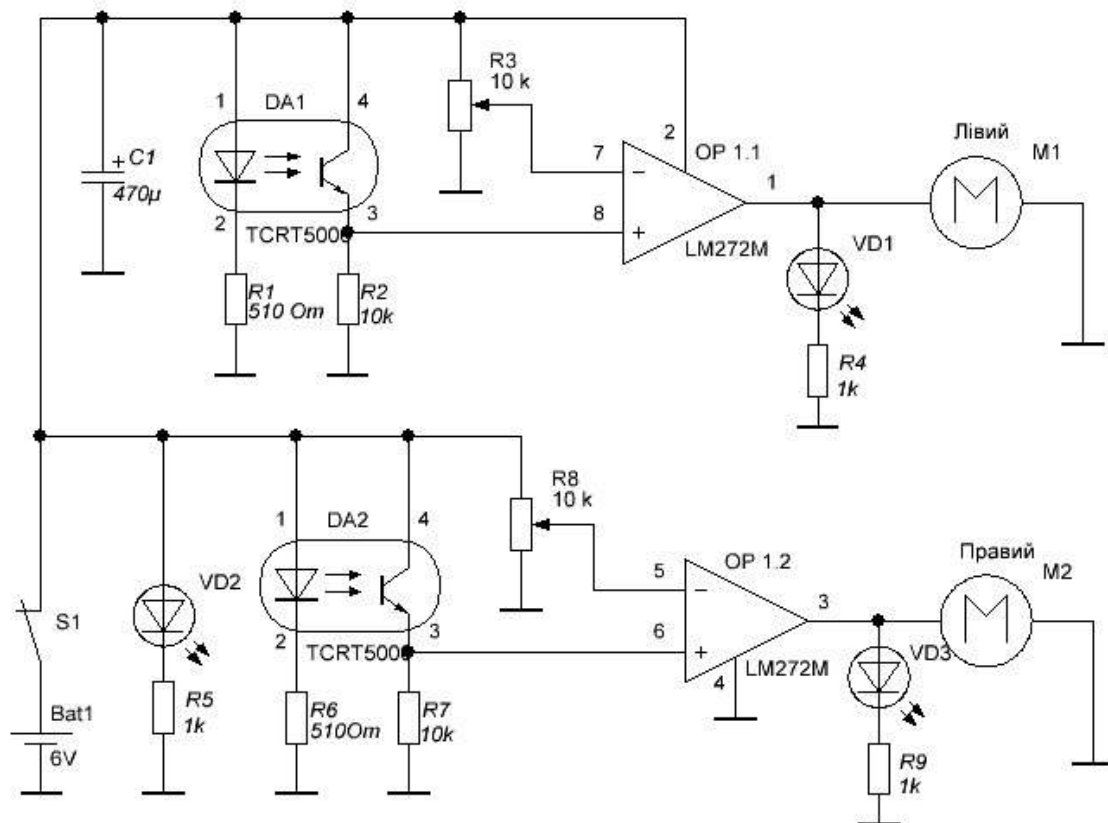


Рисунок 2.18 – Схема електрична принципова BEAM-робота

Схему електричну комутаційну усіх компонентів BEAM робота подано на рисунку 2. 19.

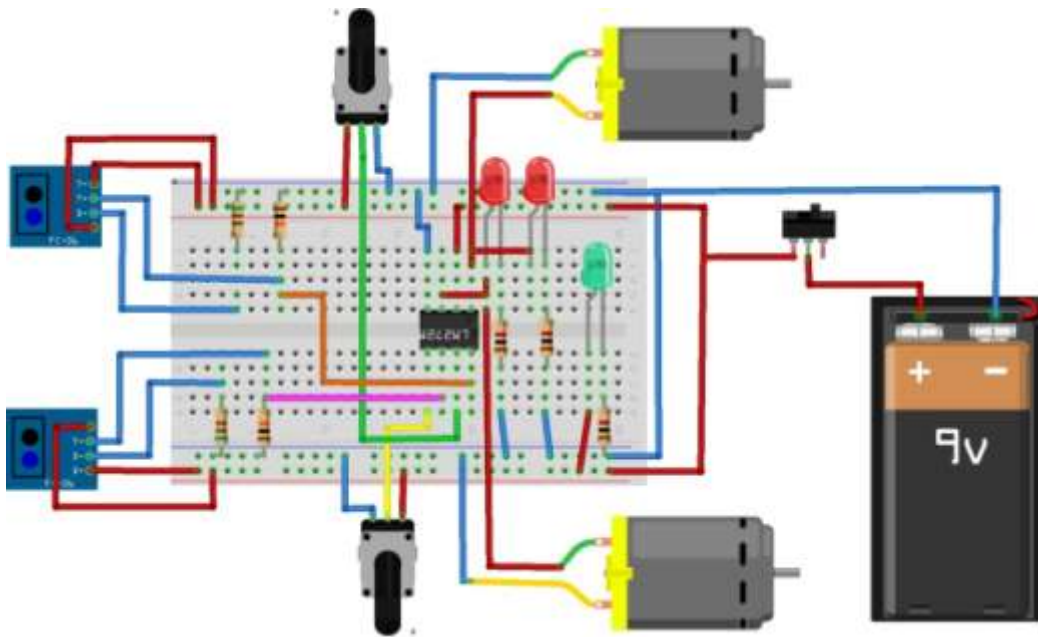


Рисунок 2.19 – Схема електрична комутаційна BEAM-робота на макетній платі

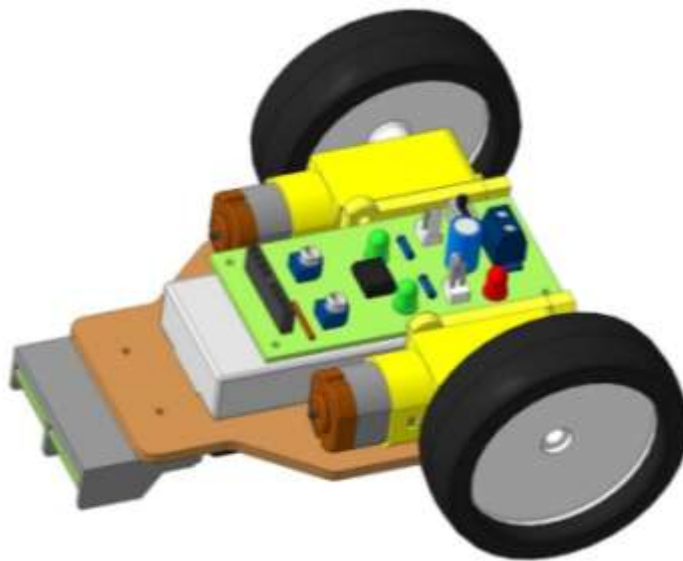


Рисунок 2.20 – 3D модель BEAM-робота

Змонтувати усі компоненти відповідно до рисунку 2.20.

2.3.6 Дослідити принцип руху такого аналогового робота по чорній лінії та описати принципи його поведінки під час руху.

2.4 Приклад виконання практичної роботи

Відповідно до схеми, зображеної на рисунку 2.15, складаємо схему електричну комутаційну на макетній платі.

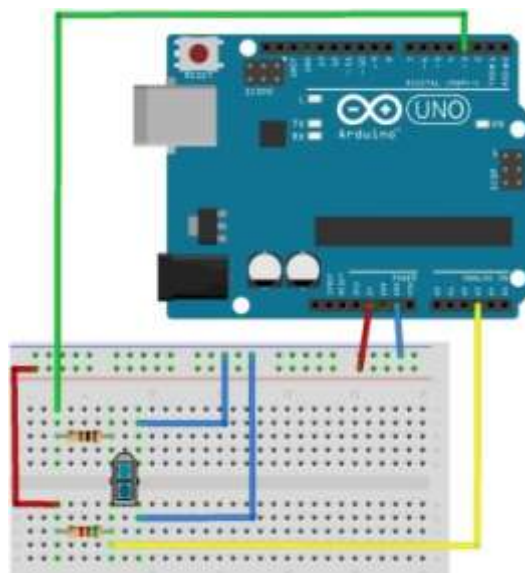


Рисунок 2.21 – Схема електрична комутаційна

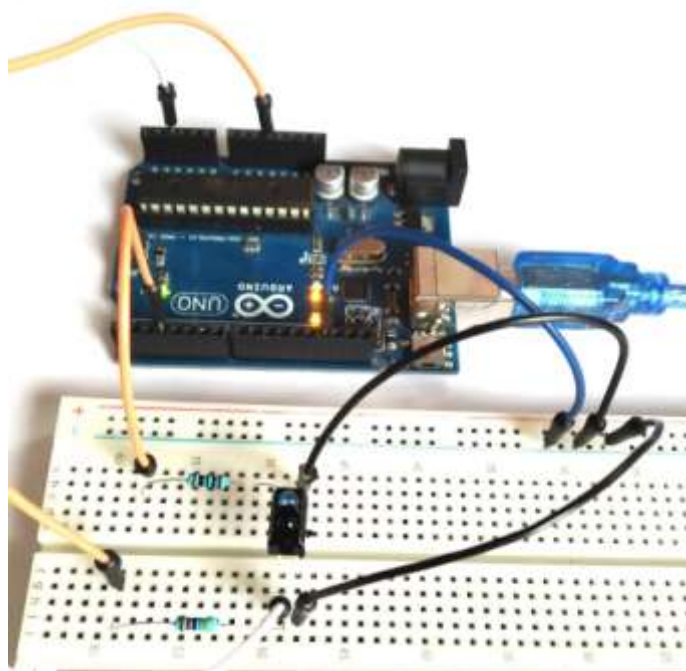


Рисунок 2.22 – Зовнішній вигляд досліджуваної схеми

Завантажуємо код, наведений в практичній роботі, який видаляє статичний шум із даних, що надходять з ІЧ-датчика.

Як бачимо з даних (рис. 2.23), що надходять в монітор порту, чим більша інтенсивність освітлення сенсора, за тієї самої відстані, тим більші значення фіксуються мікроконтролером.

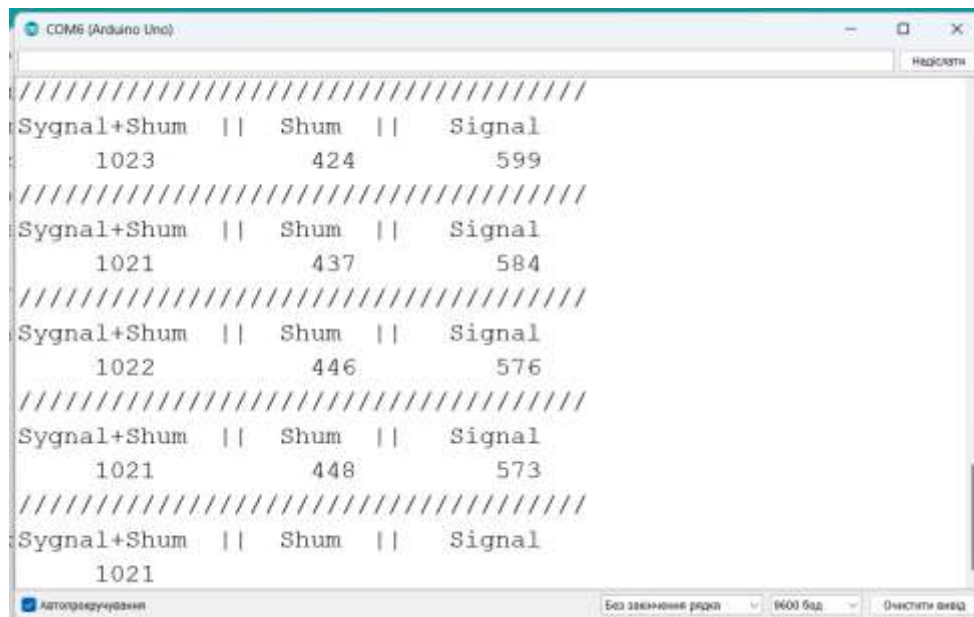


Рисунок 2.23 – Значення статичних шумів, отриманих за різної інтенсивності освітлення сенсора

Для отримання корисного сигналу сенсора необхідно здійснити програмне віднімання значень (сигнал + шум – шум). Однак таке віднімання шумів не дасть абсолютно точного результату, оскільки зв'язок між відстанню та вихідним сигналом датчика не є лінійним.

Відповідно до схеми електричної комутаційної, зображеної на рис. 2.22, складаємо схему та завантажуюмо відповідний програмний код.

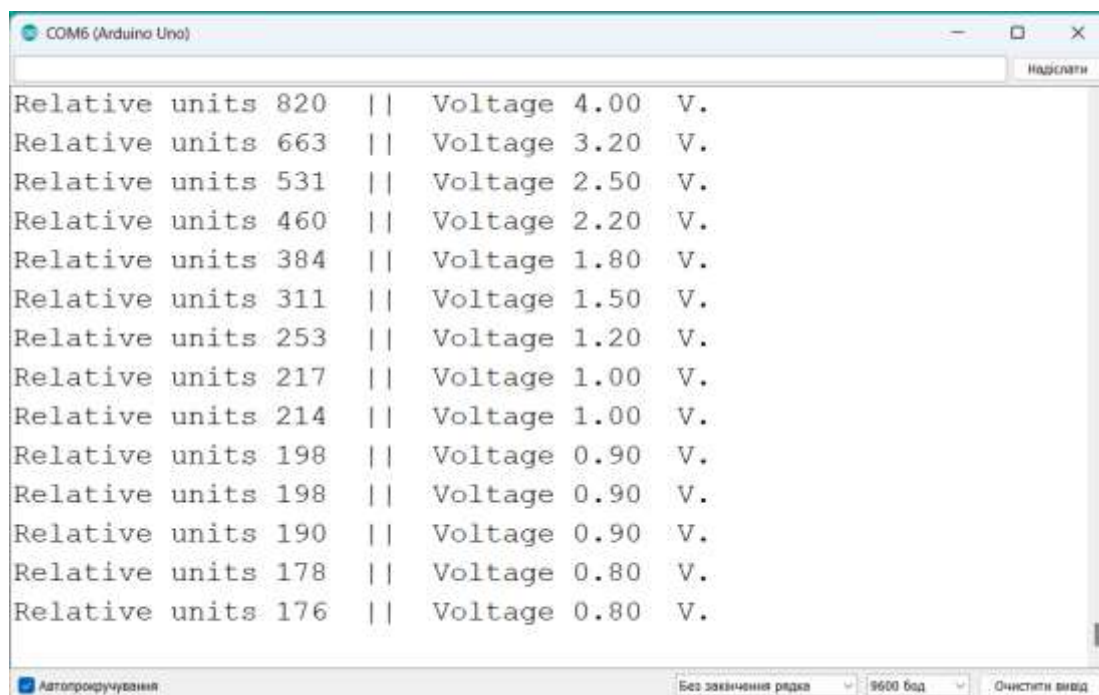


Рисунок 2.24 – Відносні одиниці та відповідне значення напруги, отримані за різної відстані сенсора

Як бачимо з даних (рис. 2.24), що надходять в монітор порту, чим більша відстань до сенсора, за тієї самої освітленості, тим більші значення фіксуються мікроконтролером.

Відповідно до схеми електричної принципової та схеми електричної комутаційної, поданих раніше, використовуючи макетну плату, складаємо конструкцію простого ВЕАМ робота (рис. 2.25 – 2.26).

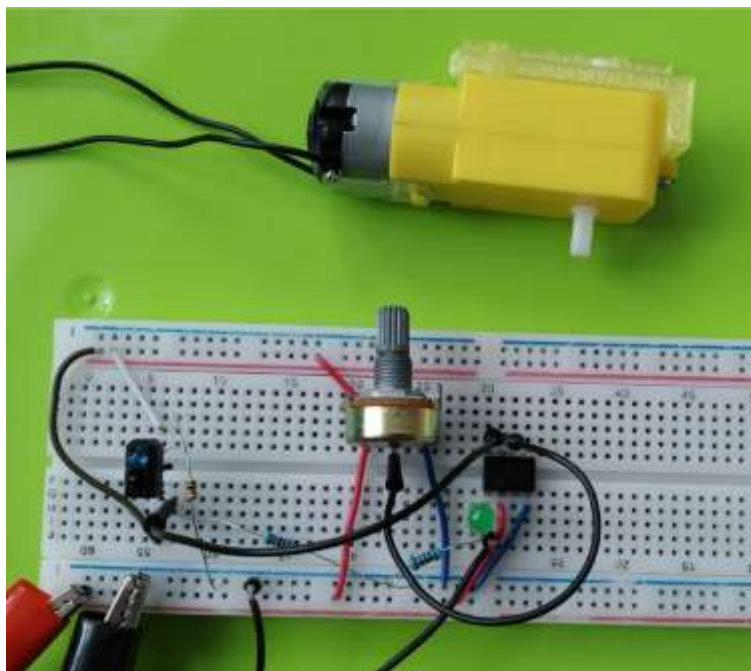


Рисунок 2.25 – Зовнішній вигляд складеного макета

Для встановлення рівня спрацювання компаратора, налаштувати схему за допомогою потенціометра 10 кОм.

За умови роботоздатності цієї схеми у разі відкритого сенсора TCRT 500 двигун починає працювати, приводячи в рух вал мотор-редуктора, зелений світлодіод буде світитись.

Відео роботи складеного макета подано за посиланням:
https://drive.google.com/file/d/1qnEOOrzisLrjlU720HlulYEETQj3JQOYy/view?usp=drive_link



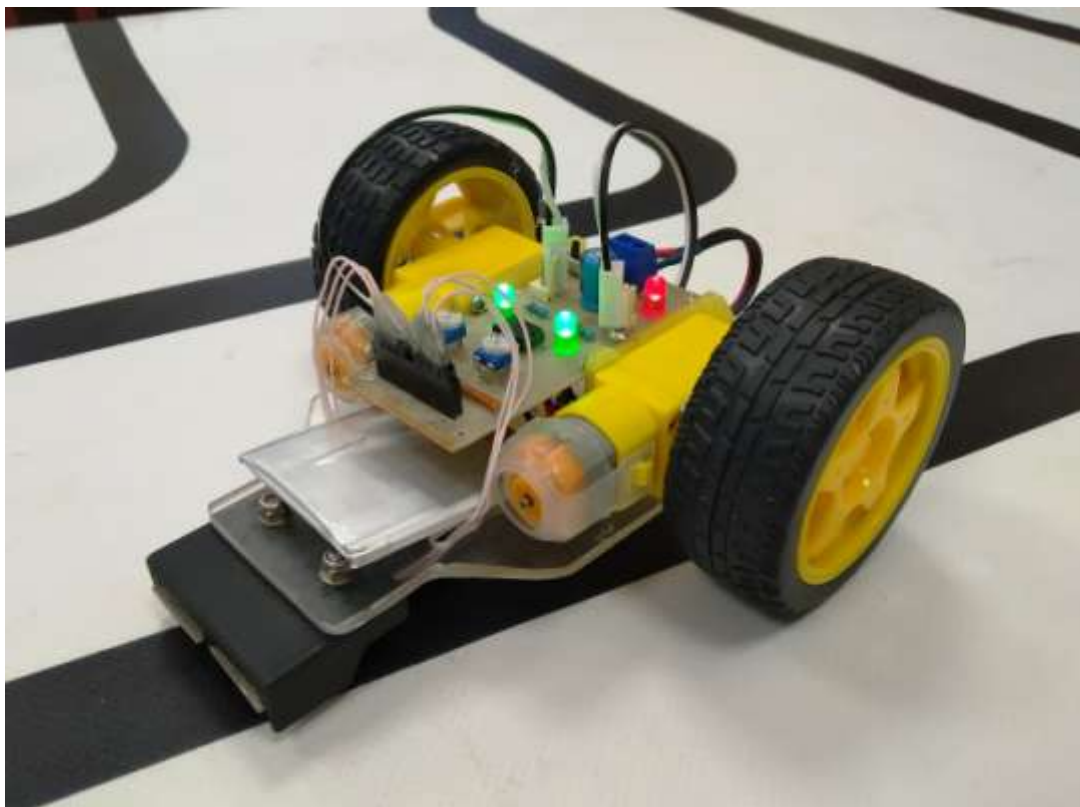


Рисунок 2.26 – Зовнішній вигляд складеного макета

2.5 Зміст звіту

1. Звіт з практичної роботи має бути виконаний на аркушах формату А4 та відповідати варіанту практичної роботи.

2. Звіт має містити: назву практичної роботи, її мету і короткі теоретичні відомості.

3. В розділі «Результати виконання практичної роботи» здобувач має навести структурні схеми та їх реалізацію на макетній платі та надати фото складеного під час практичної роботи робота, результати виконання роботи сенсора та отримані значення з монітора послідовного порту середовища програмної розробки Arduino IDE.

4. Написати висновки до практичної роботи.

Контрольні запитання

1. Які типи сенсорів зазвичай використовуються для створення найпростішої сенсорної системи в BEAM роботах?
2. Як можна реалізувати зв'язок між сенсорами та системою керування BEAM робота для збору та аналізу інформації?
3. Які конкретні завдання та функції можна доручити цій найпростішій сенсорній системі в BEAM роботі?
4. Як використання сенсорів сприяє підвищенню здатності BEAM робота реагувати на зміни в навколишньому середовищі?
5. Як працює схема руху по лінії на базі компаратора LM272M в BEAM робототехніці?
6. Які датчики або сенсори можуть бути використані для визначення лінії, вздовж якої має переміщуватися BEAM робот?
7. Яким чином схема руху по лінії на базі LM272M може бути оптимізована для досягнення точного і стабільного слідування за лінією?
8. Які можливі виклики або проблеми можуть виникнути у випадку використання цієї схеми у BEAM робототехніці і як їх можна вирішити?
9. Які типи електромоторів можуть бути задіяні в процесі проектування BEAM роботів, і які основні характеристики електромоторів потрібно враховувати під час підключення?
10. Як можна вдосконалити схему BEAM робота, що обходить перешкоди без використання датчиків, для підвищення ефективності та надійності?
11. Яким чином можна реалізувати схему BEAM робота, яка може обходити перешкоди без використання датчиків?
12. Які принципи роботи логічних елементів використовуються в цій схемі для визначення та управління рухом робота?
13. Які можливі обмеження та виклики можуть виникнути під час роботи BEAM робота без використання датчиків у режимі обходу перешкод?
14. Яким чином апаратне забезпечення робота працює у цій схемі для досягнення оптимальної продуктивності?

Практична робота № 3

ПЕРСОНАЛЬНА ТА ЛОКАЛЬНА НАВІГАЦІЯ МОБІЛЬНИХ РОБОТІВ

Мета роботи: ознайомитися з принципами персональної та локальної навігації мобільного робота, вивчити методи визначення положення та орієнтації в просторі, а також реалізувати алгоритми руху робота до заданої цілі.

3.1 Теоретичні відомості [12, 13, 14, 15, 16]

3.1.1 Типи та засоби персональної навігації мобільних роботів

Персональна навігація відповідає за здатність коректного взаємного розташування робота і його частин відносно об'єктів навколишнього середовища для переміщення по необхідній траєкторії, а також уникнення зіткнення шляхом відповідного керування приводами маніпулятора та засобів переміщення робота.

Водночас необхідно вирішувати такі завдання:

- управління мобільними роботами для переміщення по необхідній траєкторії;
- виявлення різних об'єктів (зокрема перешкод) навколо робота та їх координат відносно робота;
- уникнення зіткнень з небажаними зовнішніми об'єктами;
- коректна взаємодія (включно маніпулювання) із зовнішніми об'єктами.

Персональна навігація базується на отриманні інформації з датчиків, що знаходяться на самому роботі.

У цьому випадку використовуються датчики внутрішньої інформації для визначення положення та переміщення як окремих частин, так і самого робота (датчики лінійного та кутового переміщення), а також датчики зовнішньої інформації для визначення положення та переміщення як окремих частин, так і самого робота відносно зовнішніх об'єктів (датчики для визначення відстані та пристрої візуального спостереження – машинного зору).

Ці датчики забезпечують досить високу точність визначення положення окремих частин робота одна відносно одної та відносно зовнішніх об'єктів. Так, засоби одометрії дають можливість визначити шлях, траєкторію та швидкість переміщення за допомогою датчиків кута обертання коліс.

Недоліком таких засобів є накопичення помилок визначення траєкторії переміщення.

У разі використання засобів одометрії траєкторію переміщення можна задавати, виходячи зі шляху, який проходять колеса робота. Для визначення траєкторії переміщення найчастіше використовується лінійне переміщення та переміщення по колу з заданим радіусом.

На рисунках 3.1, 3.2 наведено визначення траєкторії за допомогою переміщення по дузі з заданим радіусом R та кутом повороту $\Delta\theta$, а також лінійного переміщення на відстань l_V , відповідно, для триколісного робота (трицикла) та робота з диференціальним приводом.

За умови використання оптичного датчика визначення шляху переміщення здійснюється за допомогою вимірювання кута переміщення колеса.

Шлях l , який проходить колесо під час обертання на кут φ , дорівнює:

$$l = d \cdot \varphi / 2 \text{ або } l = d \cdot \varphi^\circ / 360^\circ, \quad (3.1)$$

де d – діаметр колеса,

φ та φ° – кут обертання колеса у радіанах або градусах, відповідно.

Це дає можливість визначити зв'язок кута обертання колеса та шляху, який проходить робот під час лінійного переміщення на відстань l_V .

Зв'язок кута обертання колеса та шляху, який проходить робот під час кутового переміщення, залежить від типу привода, що здійснює переміщення.

Для триколісного робота (рис. 3.1), як було показано вище, радіус дуги повороту робота R , кут повороту робота $\Delta\theta$, кут повороту ведучого колеса відносно робота s , переміщення ведучого колеса $l_{\Delta\theta}$ та відстань між ведучим та опорними колесами L пов'язані такими залежностями:

$$R = \frac{L}{\tan s}, \quad \Delta\theta = \frac{l_{\Delta\theta}}{\tan s}, \quad \tan s = \frac{L}{R}. \quad (3.2)$$

Тому для переміщення по дузі із радіусом R та кутом повороту робота $\Delta\theta$ треба задати кут повороту ведучого колеса s та переміщення ведучого колеса $l_{\Delta\theta}$ таким чином:

$$s = \arctg \frac{L}{R}, \quad l_{\Delta\theta} = \frac{L \cdot \Delta\theta}{\sin s}. \quad (3.3)$$

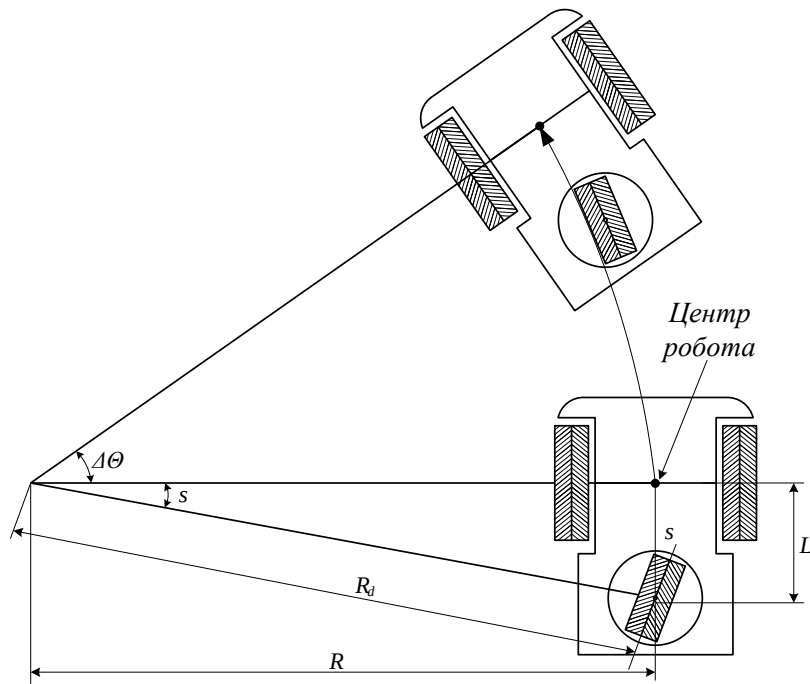


Рисунок 3.1 – Визначення траєкторії переміщення для триколісного робота за допомогою переміщення по дузі із заданим радіусом R та кутом повороту $\Delta\theta$, а також лінійного переміщення на відстань l_V

Для диференційного привода (рисунок 3.2), як було показано вище, радіус дуги повороту робота R , кут повороту робота $\Delta\theta$, швидкість переміщення робота V , швидкості переміщення лівого та правого коліс V_L , V_R , переміщення лівого та правого коліс l_L , l_R та відстань між колесами W пов'язані такими залежностями:

$$R = \frac{W \cdot (V_R + V_L)}{2(V_R - V_L)} \quad (3.4)$$

$$\Delta\theta = \frac{(V_R - V_L) \cdot \Delta t}{W} = \frac{l_R - l_L}{W} \quad (3.5)$$

Для переміщення по дузі з радіусом R швидкість переміщення робота V , швидкості переміщення коліс V_L та V_R пов'язані такими залежностями:

$$\frac{V_R}{V_L} = \frac{2 \cdot R + W}{2 \cdot R - W} \quad (3.6)$$

$$V = \frac{V_R + V_L}{2} \quad (3.7)$$

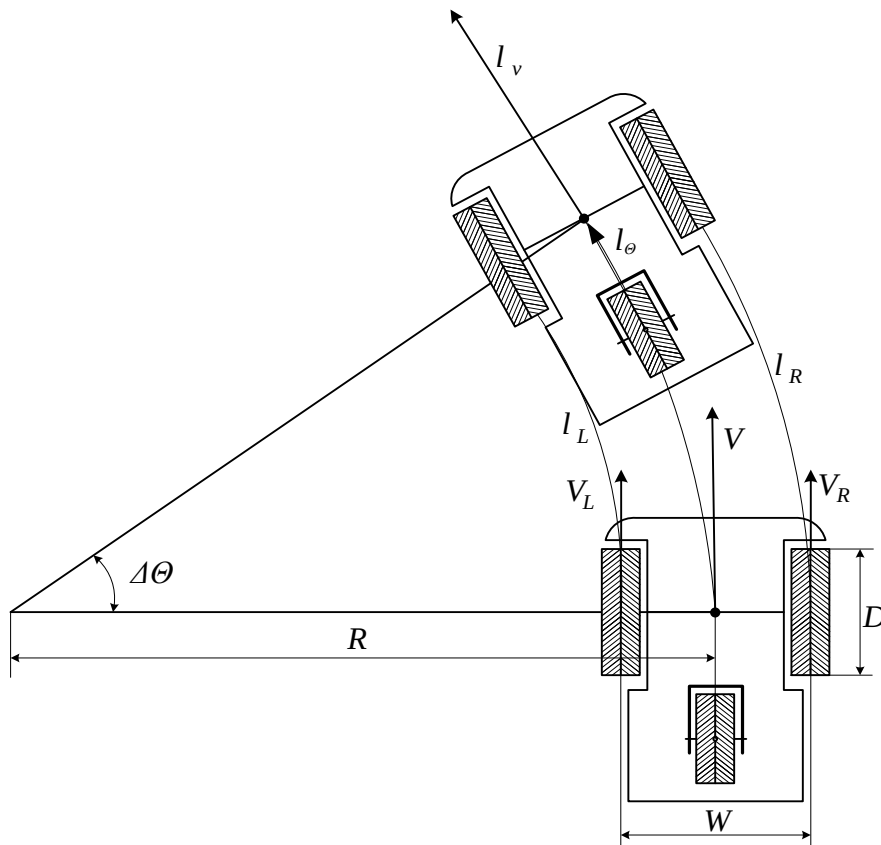


Рисунок 3.2 – Визначення траєкторії переміщення для диференційного привода за допомогою переміщення по дузі із заданим радіусом R та кутом повороту $\Delta\theta$, а також лінійного переміщення на відстань l_v

Шлях, який пройде ліве колесо по дузі з радіусом R та кутом повороту робота $\Delta\theta$ дорівнює:

$$l_L = V_L \cdot \Delta t = (R - W/2) \cdot \Delta\theta. \quad (3.8)$$

Шлях, який пройде праве колесо по дузі з радіусом R та кутом повороту робота $\Delta\theta$ дорівнює:

$$l_R = V_R \cdot \Delta t = (R + W/2) \cdot \Delta\theta. \quad (3.9)$$

Під час переміщення по заданій траєкторії можна використовувати різні засоби навігації, але для триколісного робота та робота з диференційним приводом, як було показано вище, досить просто описати цю траєкторію лінійною та коловою інтерполяцією.

Позиційне керування можна здійснити за допомогою одометра шляхом визначення переміщення за рахунок обертання колеса та регулювання швидкості привода. Шлях l , що проходить колесо під час обертання на кут φ , визначається формулою (3.1).

Ця формула може бути використана для визначення шляху переміщення робота по прямій лінії. У робота з диференційним приводом у цьому випадку $V_R = V_L$.

Для повороту робота по дузі з радіусом R на кут $\Delta\theta$ для триколісного робота потрібно здійснити переміщення $l_{\Delta\theta}$ та кут повороту у ведучого колеса s згідно з формулою (3.3).

Для повороту робота з диференційним приводом по дузі з радіусом R на кут $\Delta\theta$ потрібно здійснити переміщення лівого та правого коліс l_R та l_L , що визначаються формулами (3.8) та (3.9).

Водночас потрібно забезпечити відношення між швидкостями переміщення коліс V_R та V_L згідно з формулою (3.6).

Для уникнення зіткнень та визначення зовнішніх об'єктів застосовуються засоби, що дозволяють визначити наявність об'єктів та відстань до них. Ця задача вирішується за допомогою ультразвукових, інфрачервоних та лазерних датчиків, що дозволяють знайти відстань до об'єктів та їх положення відносно положення робота.

Такі датчики вимірювання відстані можуть використовуватися, наприклад, для переміщення вздовж перешкод або стін у приміщенні (див. рисунок 3.3).

Розглянемо можливість переміщення вздовж перешкоди на певній відстані з використанням ультразвукового датчика та пропорційного регулювання.

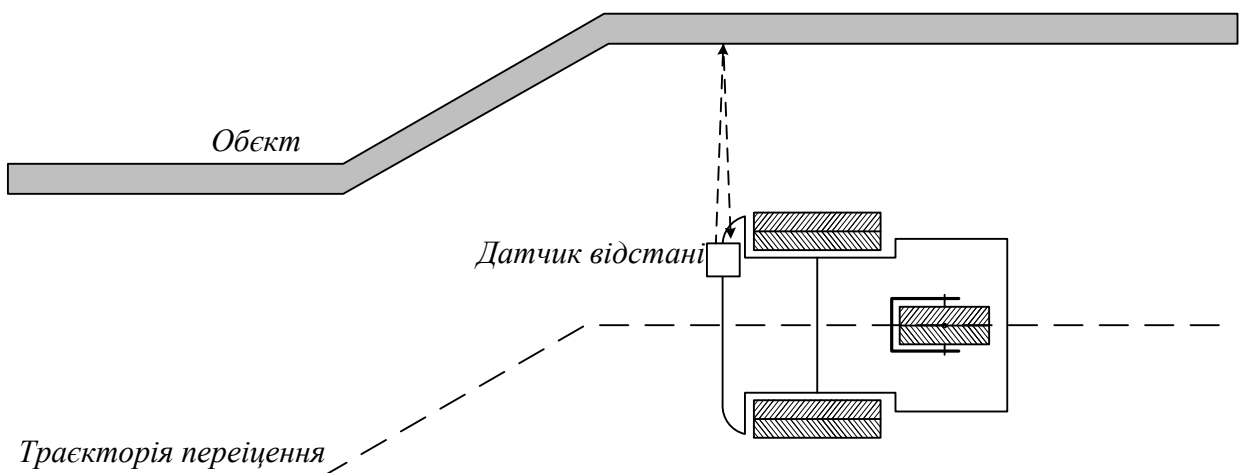


Рисунок 3.3 – Використання датчиків вимірювання відстані для переміщення вздовж перешкод або стін у приміщенні

Поточне значення відстані до перешкоди позначимо L . Визначимо максимально допустимі відстані наближення l_{max} і l_{min} , та знайдемо потрібну відстань до перешкоди як середнє значення наближення

$$l_{сер} = (l_{max} + l_{min})/2.$$

Позначимо відносну швидкість лівого двигуна V_L , а потужність правого двигуна V_R .

Для визначення швидкості V_L , V_R у програмі, як правило, використовуються відносні значення, які можуть змінюватись від 0 до 100.

Керування здійснюється за формулою:

відносна швидкість лівого двигуна:

$$V_L = V_{сер} - (l_{сер} - L) \cdot k; \quad (3.10)$$

відносна швидкість правого двигуна:

$$V_R = V_{сер} + (l_{сер} - L) \cdot k; \quad (3.11)$$

де k – коефіцієнт пропорційності, який вибирають таким чином, щоб значення величини $(l_{сер} - L) \cdot k$ не перевищувало значень $V_{сер}$ та $1 - V_{сер}$, а значення V_L та V_R були в межах від 0 до 100.

Якщо середнє значення наближення дорівнює $l_{сер}$, то множники в дужках рівні, і правий та лівий двигуни здійснюють рух в одному напрямку з однаковою швидкістю (потужністю $P_{сер}$).

Якщо відстань до об'єкта менше $l_{сер}$, то відповідно зменшується швидкість (потужність) лівого двигуна і збільшується швидкість правого двигуна, а якщо відстань до перешкоди більше $l_{сер}$, то навпаки. Таким чином радіус повороту буде залежати від ступеня відхилення від середнього значення наближення до перешкоди та переміщення буде більш плавним.

3.1.2 Локальна навігація мобільних роботів

Локальна навігація здійснює визначення координат відносно деякої (зазвичай стартової) точки, що актуально для переміщення у певних приміщеннях, наприклад, на технологічних ділянках або у складах.

Локальна навігація використовує датчики зовнішньої інформації, що дають можливість знайти положення робота шляхом визначення зовнішніх об'єктів або вказівників положення робота та засобів маршрутослідкування. Визначення зовнішніх об'єктів може здійснюватися

за допомогою ультразвукових, ультрафіолетових та лазерних датчиків, що дозволяють знайти відстань до об'єктів та їх положення відносно положення робота.

Системи маршрутослідування використовують різні вказівники маршруту та датчики, що визначають ці вказівники й забезпечують відповідне переміщення робота (рисунок 3.4).

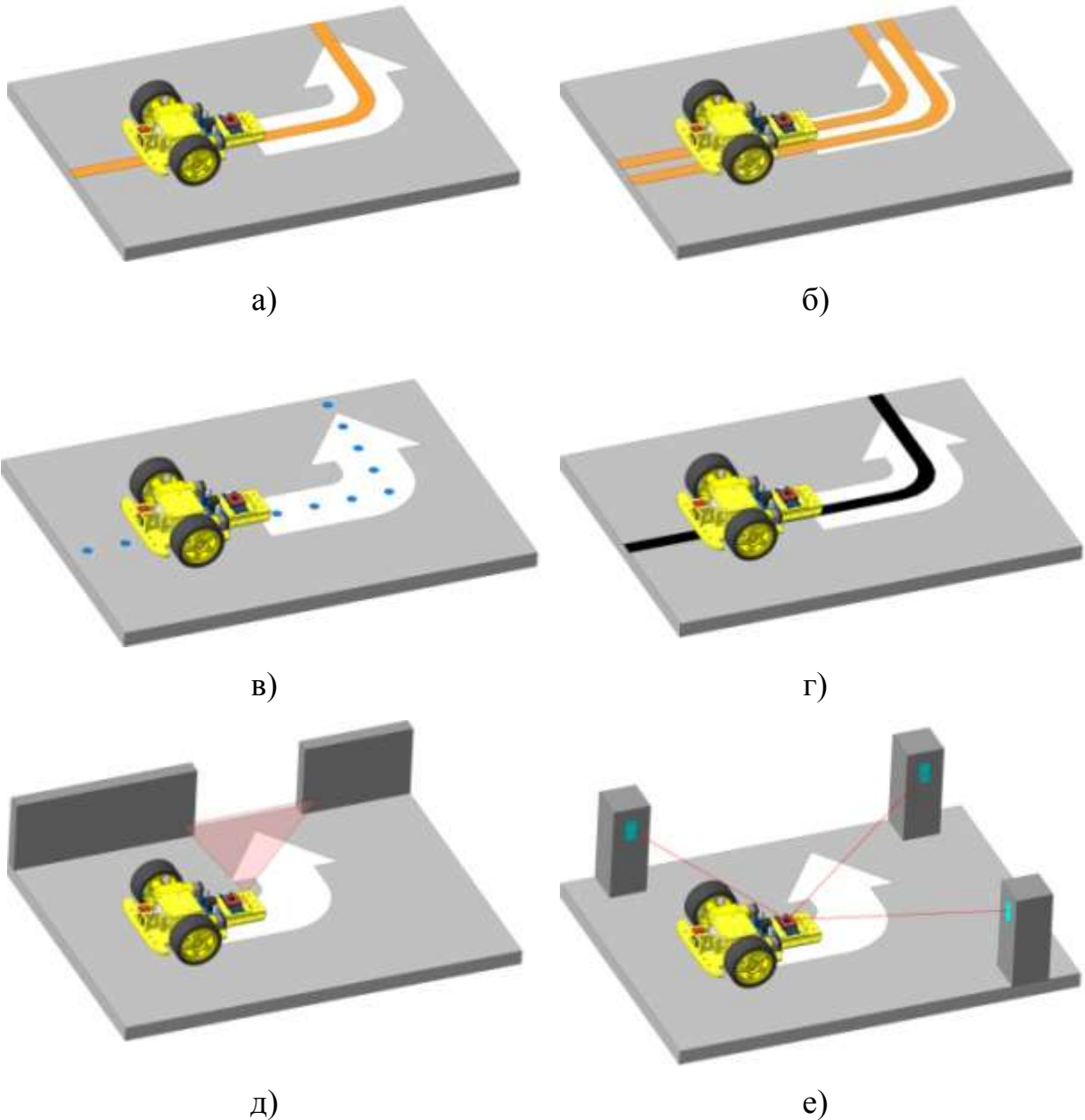


Рисунок 3.4 – Системи маршрутослідування

Найбільш поширеними є системи маршрутослідування на основі індуктивних датчиків (рисунок 3.4, а), б), в) (системи активного та пасивного маршрутослідування), системи маршрутослідування на основі оптичних датчиків (рисунок 3.4, г), системи маршрутослідування на

основі датчиків вимірювання відстані до поверхні, вздовж якої здійснюється переміщення (рисунок 3.4, д), системи маршрутослідування на основі лазерних датчиків, що здійснюють визначення положення робота за допомогою рефлекторів (рисунок 3.4, е).

Системи активного маршрутослідування на основі індуктивних датчиків використовують вказівники маршруту у вигляді дроту, по якому протікає змінний струм. На рухомій платформі встановлено дві котушки, в яких індукується змінний струм від дроту, що вказує маршрут. Різниця струму в котушках визначає відхилення від вказівника та використовується для керування двигунами робота для зменшення цього відхилення.

Системи пасивного маршрутослідування на основі індуктивних датчиків використовують магнітну металеву стрічку завширшки близько 5 – 10 см, яку розпізнають датчики магнітного поля. У разі встановлення двох – трьох таких датчиків можна визначити відхилення від маршруту на відстань до 30 мм.

Системи активного маршрутослідування на основі індуктивних датчиків можуть використовувати як вказівники маршруту вмонтовані в підлогу дроти, за допомогою яких здійснюється бездротова передача енергії (рисунок 3.4, б).

Послідовні позначки однозначно визначають маршрут пересування (рисунок 3.5, а), растрові позначки дають можливість віртуального встановлення маршруту за допомогою обчислювального пристрою (рисунок 3.5, б).

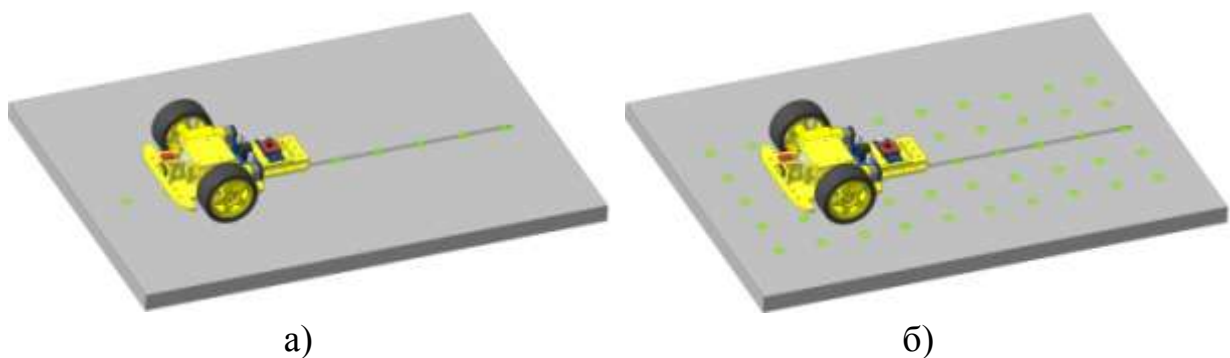


Рисунок 3.5. Навігація за допомогою послідовних а) та растрових б) магнітних позначок

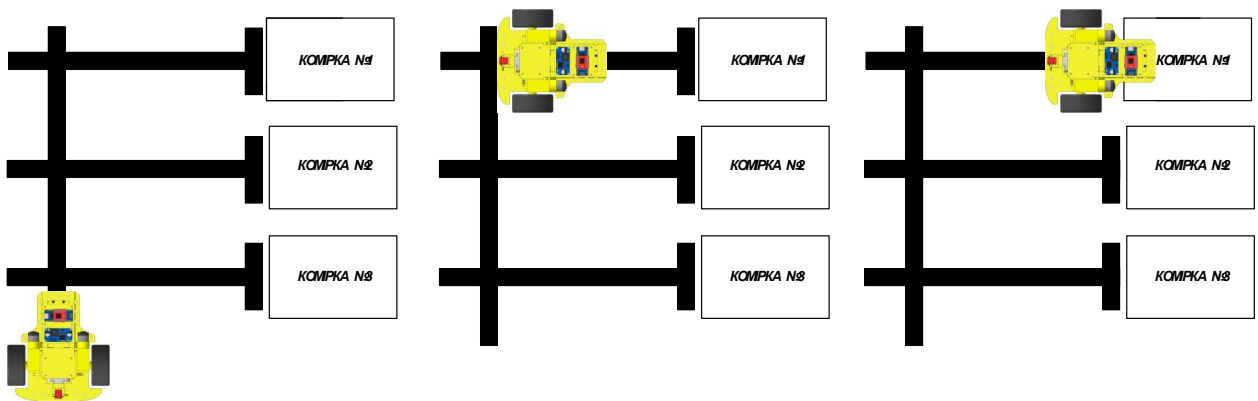
На рисунку 3.6 наведено мобільний робот, який здійснює маршрутослідування за допомогою трьох оптичних датчиків.

Робот має два приводи – правий та лівий, а також три оптичних датчики – праворуч та ліворуч від смуги вказівника маршруту. У вихідному положенні оптичні датчики знаходяться справа та зліва від

смуги вказівника маршруту (датчики не спрацьовують) і включені обидва двигуни (візок переміщується прямо).



а)



б)

Рисунок 3.6 – Робот mBot з маршрутослідуванням за допомогою трьох оптичних датчиків а) та приклад переміщення автоматичного навантажувача на складі для встановлення вантажу у комірку б)

Якщо на смугу потрапляє лівий датчик, то він спрацьовує та здійснюється поворот праворуч (зупиняється лівий двигун), а якщо на смугу потрапляє правий датчик, то здійснюється поворот ліворуч (зупиняється правий двигун).

Спрацювання обох датчиків на контрольній позначці визначає встановлену дію, наприклад: зупинку, точку повороту або точку підрахунку позицій зміни маршруту (рисунок 3.6, а).

На рисунку 3.6, б) наведено приклад переміщення автоматичного навантажувача на складі для встановлення вантажу у певну комірку стелажа.

Для переміщення вздовж лінії вказівника траєкторії можна також використати один датчик освітлення, який під час переходу, наприклад, з білої на чорну смугу буде змінювати значення освітлення.

Приклад програмної реалізації маршрутослідування по заданій траєкторії для робота mBoot наведено нижче (приклад 3.1).

Приклад 3.1

```
#include <GyverTM1637.h>
#define CLK 12
#define DIO 11
#include "GyverTM1637.h"
GyverTM1637 disp(CLK, DIO);

uint8_t Speed_motor = 185;      // Швидкість двигунів
рух вперед

int mas[11] = {1, 0, 0, 2, 2, 1, 1, 2, 0, 1, 3};
int i = 0;

const uint8_t PIN_LINE_Left = 0;
const uint8_t PIN_LINE_Right = 1;
int FLAG_Left = 0;
int FLAG_Right = 0;
const uint8_t PIN_LINE_Left_POV = 2;
const uint8_t PIN_LINE_Right_POV = 3;
int FLAG_Left_POV = 0;
int FLAG_Right_POV = 0;
int POVOROT = 0;
const int SENSOR_PIN = 10;
int lastState = LOW;
int currentState, Button_State ;

void setup() {
  Serial.begin(9600);
  disp.clear();
  disp.brightness(7);
  pinMode(4, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
```

```

pinMode(13, OUTPUT);
pinMode(SENSOR_PIN, INPUT);
disp.displayByte(0x00, 0x6, 0x76, 0x00);

label:
currentState = digitalRead(SENSOR_PIN);
if ((currentState) == LOW) {
    goto label;
}
else {
}
disp.displayByte(0x00, 0x7, 0x31, 0x00);
}

void loop() {
    FLAG_Left = analogRead(PIN_LINE_Left);
    FLAG_Right = analogRead(PIN_LINE_Right);
    FLAG_Left_POV = analogRead(PIN_LINE_Left_POV);
    FLAG_Right_POV = analogRead(PIN_LINE_Right_POV);

    drive();
    // Опитуємо сенсори бічні
    if ((FLAG_Left_POV < 800) or (FLAG_Right_POV <
800)) {
        // disp.displayInt(mas[i]);
        // delay(1);

        switch (mas[i]) {
            case 0:
                disp.displayByte(0x00, 0x7, 0x31, 0x00 );
                delay(1);
                break;
            case 1:
                disp.displayByte(0x00, 0x00, 0x00, 0x70);
                delay(1);
                break;
            case 2:
                disp.displayByte(0x46, 0x00, 0x00, 0x00);
                delay(1);
                break;
            case 3:
                disp.displayByte(0x5e, 0x3f, 0x47, 0x6d);
                delay(1);
                break;
        }
    }
}

```

```

    }

    //пропускаємо поворот
    if (mas[i] == 0) {
        //рух робота прямо
        digitalWrite(4, HIGH);                // Обертання
        правого двигуна за годинниковою стрілкою
        analogWrite (5, Speed_motor);        // швидкість
        обертання лівого двигуна (ШИМ=255)
        digitalWrite(7, LOW);                // Обертання
        лівого двигуна проти годинникової стрілки
        analogWrite (6, Speed_motor);
        delay(250);
        //стоп
        digitalWrite(4, HIGH);                // Обертання
        правого двигуна за годинниковою стрілкою
        analogWrite (5, 0);                  // швидкість
        обертання лівого двигуна (ШИМ=255)
        digitalWrite(7, LOW);                // Обертання
        лівого двигуна проти годинникової стрілки
        analogWrite (6, 0);
        delay(50);
        drive();
    }
    // поворот на ліво
    if (mas[i] == 1) {
        digitalWrite(4, HIGH);                // Обертання
        правого двигуна за годинниковою стрілкою
        analogWrite (5, Speed_motor);        // швидкість
        обертання лівого двигуна (ШИМ=255)
        digitalWrite(7, LOW);                // Обертання
        лівого двигуна проти годинникової стрілки
        analogWrite (6, Speed_motor);
        delay(180);

        //ПОВОРОТ
        digitalWrite(4, HIGH);                // Обертання
        правого двигуна за годинниковою стрілкою
        analogWrite (5, 255);                // швидкість
        обертання лівого двигуна (ШИМ=255)
        digitalWrite(7, LOW);                // Обертання
        лівого двигуна проти годинникової стрілки
        analogWrite (6, 0);
        delay(400);
    }

```

```

        digitalWrite(4, HIGH);                // Обертання
правого двигуна за годинниковою стрілкою
        analogWrite (5, 0);                    // швидкість
обертання лівого двигуна (ШИМ=255)
        digitalWrite(7, LOW);                  // Обертання
лівого двигуна проти годинникової стрілки
        analogWrite (6, 0);
        delay(50);
        drive();
    }
    // поворот праворуч
    if (mas[i] == 2) {
        digitalWrite(4, HIGH);                // Обертання
правого двигуна за годинниковою стрілкою
        analogWrite (5, Speed_motor);          //
швидкість обертання лівого двигуна (ШИМ=255)
        digitalWrite(7, LOW);                  // Обертання
лівого двигуна проти годинникової стрілки
        analogWrite (6, Speed_motor);
        delay(180);

        //ПОВОРОТ
        digitalWrite(4, HIGH);                // Обертання
правого двигуна за годинниковою стрілкою
        analogWrite (5, 0);                    // швидкість
обертання лівого двигуна (ШИМ=255)
        digitalWrite(7, LOW);                  // Обертання
лівого двигуна проти годинникової стрілки
        analogWrite (6, 255);
        delay(470);

        digitalWrite(4, HIGH);                // Обертання
правого двигуна за годинниковою стрілкою
        analogWrite (5, 0);                    // швидкість
обертання лівого двигуна (ШИМ=255)
        digitalWrite(7, LOW);                  // Обертання
лівого двигуна проти годинникової стрілки
        analogWrite (6, 0);
        delay(50);
        drive();
    }

    if (mas[i] == 3) {

```

```

        digitalWrite(4, HIGH);                // Обертання
правого двигуна за годинниковою стрілкою
        analogWrite (5, Speed_motor); // швидкість
обертання лівого двигуна (ШИМ=255)
        digitalWrite(7, LOW);                // Обертання
лівого двигуна проти годинникової стрілки
        analogWrite (6, Speed_motor);
        delay(750);

        digitalWrite(4, HIGH);                // Обертання
правого двигуна за годинниковою стрілкою
        analogWrite (5, 0);                // швидкість
обертання лівого двигуна (ШИМ=255)
        digitalWrite(7, LOW);                // Обертання
лівого двигуна проти годинникової стрілки
        analogWrite (6, 0);
        delay(1000000);
    }
    i++;
}
drive();
}
void drive() {

label1:
    FLAG_Left = analogRead(PIN_LINE_Left);
    FLAG_Right = analogRead(PIN_LINE_Right);

    // Умова білого поля "доповертання"
    if ((FLAG_Left > 800) & (FLAG_Right > 800)) {

        if (mas[i] == 1) {
            //ПОВОРОТ НАПРАВО
            digitalWrite(4, HIGH);                // Обертання
правого двигуна за годинниковою стрілкою
            analogWrite (5, 255);                // швидкість
обертання лівого двигуна (ШИМ=255)
            digitalWrite(7, LOW);                // Обертання
лівого двигуна проти годинникової стрілки
            analogWrite (6, 0);
            delay(100);

```

```

        digitalWrite(4, HIGH);                // Обертання
правого двигуна за годинниковою стрілкою
        analogWrite (5, 0);                  // швидкість
обертання лівого двигуна (ШИМ=255)
        digitalWrite(7, LOW);                // Обертання
лівого двигуна проти годинникової стрілки
        analogWrite (6, 0);
        delay(100);
    }
    if (mas[i] == 2) {
        //ПОВОРОТ НАЛІВО
        digitalWrite(4, HIGH);                // Обертання
правого двигуна за годинниковою стрілкою
        analogWrite (5, 0);                  // Швидкість
обертання лівого двигуна (ШИМ=255)
        digitalWrite(7, LOW);                // Обертання
лівого двигуна проти годинникової стрілки
        analogWrite (6, 255);
        delay(100);

        digitalWrite(4, HIGH);                // Обертання
правого двигуна за годинниковою стрілкою
        analogWrite (5, 0);                  // Швидкість
обертання лівого двигуна (ШИМ=255)
        digitalWrite(7, LOW);                // Обертання
лівого двигуна проти годинникової стрілки
        analogWrite (6, 0);
        delay(100);
    }
    goto label1;
}

if ((FLAG_Left <= 800) & (FLAG_Right <= 800)) {
    // Рух робота mBot вперед
    digitalWrite(4, HIGH);                // Обертання
правого двигуна за годинниковою стрілкою
    analogWrite (5, Speed_motor);          // швидкість
обертання лівого двигуна (ШИМ=255)
    digitalWrite(7, LOW);                // Обертання
лівого двигуна проти годинникової стрілки
    analogWrite (6, Speed_motor);          // швидкість
обертання правого двигуна (ШИМ=255)
}

```

```

    if ((FLAG_Left <= 800) & (FLAG_Right > 800)) {
        digitalWrite(4, HIGH);          // Обертання
        правого двигуна за годинниковою стрілкою
        analogWrite (5, Speed_motor);    // швидкість
        обертання лівого двигуна (ШИМ=255)
        digitalWrite(7, LOW);            // Обертання лівого
        двигуна проти годинникової стрілки
        analogWrite (6, 0);              // швидкість обертання
        правого двигуна (ШИМ=255)
    }

    if ((FLAG_Left > 800) & (FLAG_Right <= 800)) {
        digitalWrite(4, HIGH);          // Обертання
        правого двигуна за годинниковою стрілкою
        analogWrite (5, 0);              // швидкість обертання
        лівого двигуна (ШИМ=255)
        digitalWrite(7, LOW);            // Обертання
        лівого двигуна проти годинникової стрілки
        analogWrite (6, Speed_motor);
    }
}

```

Приклад роботи наведеного коду подано на відео за посиланням:
https://drive.google.com/file/d/1goe5y-2iqM7KgwmNK36yLN7nz94VdNb7/view?usp=drive_link



Використання відеокамери та цифрової обробки зображення дає можливість не тільки здійснити переміщення вздовж вказівника, але й оптимізувати шлях переміщення за допомогою попереднього визначення ділянок, де здійснюється поворот і потрібно зменшити швидкість. Закодований ідентифікатор дозволяє дати додаткову інформацію про переміщення (рис. 3.7).

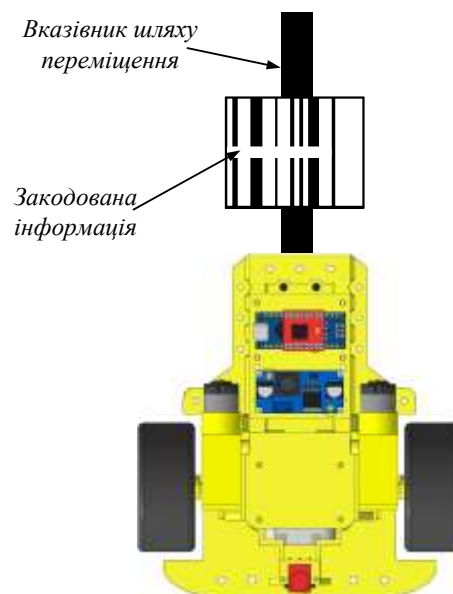
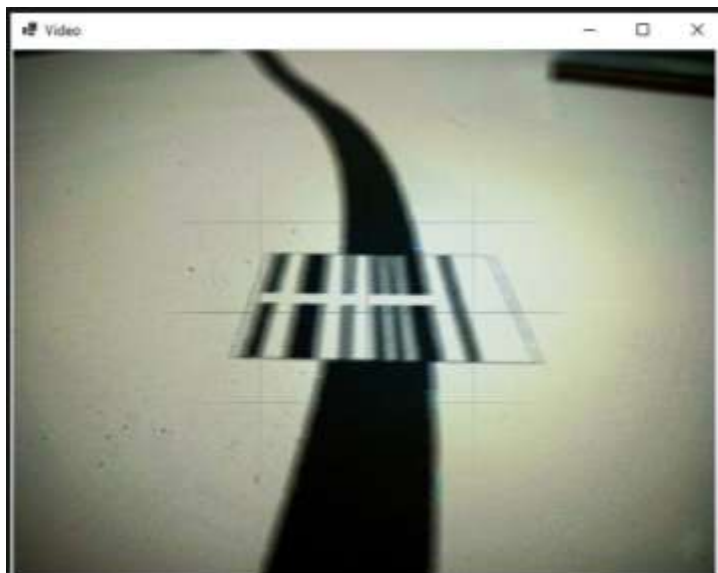


Рисунок 3.7 – Використання відеокамер для визначення траєкторії переміщення робота за допомогою вказівника та закодованої інформації

За допомогою лазерних датчиків можна визначати положення робота шляхом вимірювання відносного положення та відстані до рефлекторів, що встановлюють у певних місцях приміщення, де пересувається робот, а також до перешкод або стін у приміщенні. Для визначення взаємного положення об'єктів відносно положення робота використовують лазерні сканери, що здійснюють сканування шляхом обертання, з вимірюванням кута повороту датчика відносно положення робота. Приклади використання лазерних датчиків для визначення положення робота наведено на рисунку 3.8.

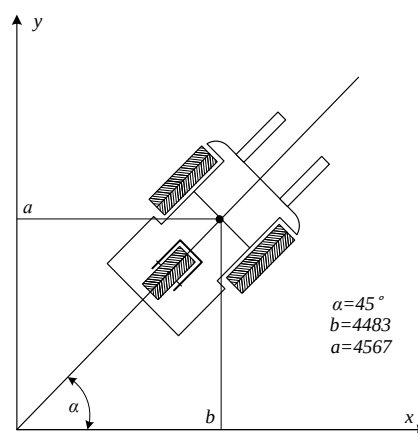
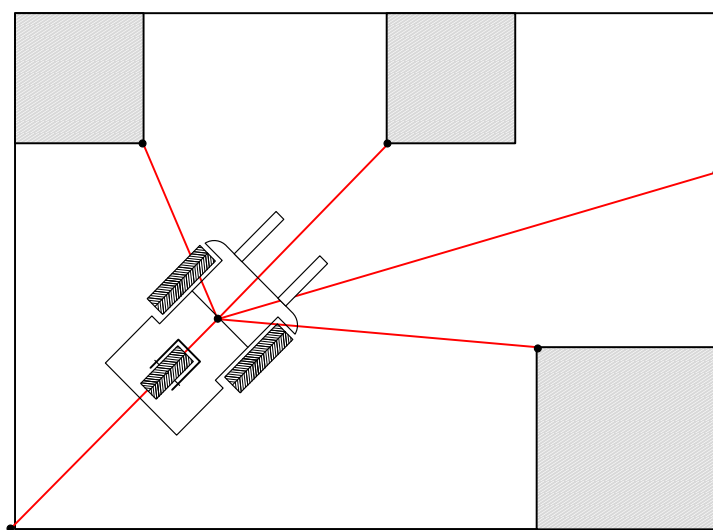


Рисунок 3.8 – Використання лазерних датчиків для визначення положення робота

Як показано на рисунку 3.9, при використанні лазерного сканера достатньо трьох рефлєкторів, щоб визначити положення та орієнтацію робота (x_p, y_p, α_p) , за допомогою визначених значень відстані (L_1, L_2, L_3) та кутів направлення $(\alpha_1, \alpha_2, \alpha_3)$ до рефлєкторів відносно положення та орієнтації робота.

Це дає можливість визначати положення мобільного робота з досить високою точністю, яка може складати декілька міліметрів.

У разі переміщення робота у приміщенні з великою кількістю об'єктів кількість рефлєкторів збільшується таким чином, щоб до кожної позиції робота надходили сигнали щонайменше від трьох рефлєкторів. Але у цьому випадку треба визначати положення робота на плані приміщення, що потребує засобів глобальної навігації, яка буде розглянута у наступному розділі.

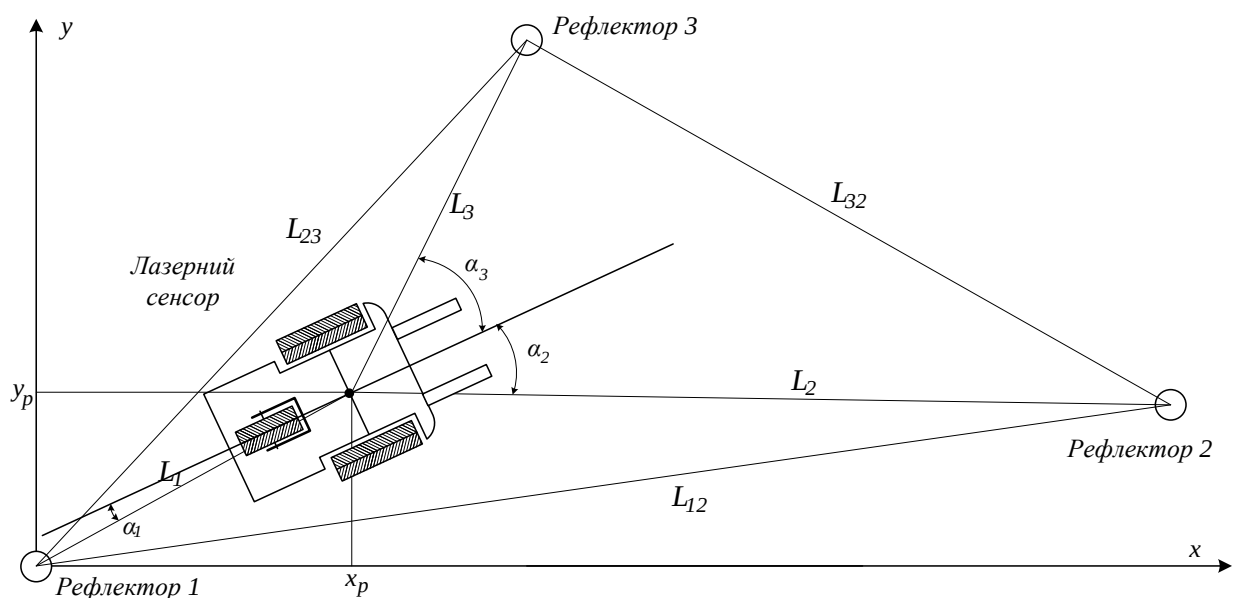


Рисунок 3.9 – Використання лазерного сканера та рефлєкторів для визначення положення робота

3.2 Послідовність виконання практичної роботи

3.2.1 Вибрати варіант практичної роботи відповідно до таблиці 3.4. Останніх дві цифри індивідуального навчального плану здобувача є варіантом практичної роботи.

3.2.2 Визначити початкову та кінцеву точки маршруту відповідно до вибраного варіанта.

3.2.3 Визначити траєкторію руху робота для локальної та персональної навігації.

3.2.4 Відповідно до варіанта в програмному коді визначити змінні, що відповідатимуть за швидкість двигунів у відносних одиницях та час, за який робот долає лінійну відстань у межах растра (LDR).

3.2.5 Побудувати мапу маршрутослідування локальної навігації.

3.2.6 Навести програмний код, що відповідає локальному маршруто-слідуванню мобільного робота по заданій траєкторії.

3.2.7 Навести програмний код, що відповідає персональному маршруто-слідуванню мобільного робота по заданій траєкторії.

3.3 Приклад виконання практичної роботи

1. Вибираємо варіант практичної роботи.

Вибір варіанта для номера індивідуального навчального плану здобувача №01-22-236. Дві останні цифри 36 отже, відповідно до таблиці 3.1.

Таблиця 3.1 – Приклад вибору варіантів практичної роботи

№ інд. навч. плану		0	1	2	3	4	5	6	7	8	9
Дес.	Од.										
Локальна навігація											
	1	↗↖	↘↖	↗↖	↘↖	↘↖	↗↖	↗↖	↘↖	↗↖	↘↖
	2	↗↖	↘↖	↘↖	↘↖	↗↖	↘↖	↗↖	↘↖	↘↖	↗↖
	3	↗↖	↘↖	↗↖	↘↖	↗↖	↘↖	↗↖	↘↖	↘↖	↗↖
	4	↘↖	↘↖	↗↖	↘↖	↘↖	↘↖	↗↖	↘↖	↘↖	↘↖
	5	↘↖	↗↖	↘↖	↘↖	↘↖	↗↖	↘↖	↗↖	↘↖	↗↖
	6	↗↖	↗↖	↘↖	↘↖	↘↖	↘↖	↗↖	↘↖	↗↖	↘↖
x		9	4	3	6	7	9	7	3	7	2
y		3	6	7	9	0	9	7	8	2	7
Персональна навігація											
	LDR, с.	1,5	1,8	2,1	1,35	1,65	1,93	2	2,14	1,7	1,4
	ш-ть, од.	255	185	200	190	240	220	210	230	250	205
Об.		↗↖	↘↖	↗↖	↘↖	↗↖	↘↖	↗↖	↘↖	↗↖	↘↖

Примітка: під час зміни напрямку руху триколісний робот має здійснювати:

↖↘ – обертання навколо своєї осі;

↖↗ – зміна траєкторії руху по дузі.

2. Визначаємо початкову та кінцеву точки маршруту а також траєкторію руху мобільного робота для локальної навігації.

Дві останні цифри 36, отже початкова точка руху мобільного робота у разі локальної навігації має координати A(x;y) 3,6:

- з колонки 3 маємо траєкторію руху: поворот праворуч → поворот праворуч → поворот ліворуч → поворот праворуч → поворот праворуч → поворот ліворуч;

- з колонки 6 маємо координати кінцевої точки **B** (x;y) 7, 7.

3. Відповідно до заданого варіанта час, за який рухома платформа (триколісний робот) долає лінійну відстань у межах растра (LDR), становить 2 секунди. Це значення зберігатимемо в числовій змінній `Speed_motor – uint8_t lineardistance = 2000`.

Швидкість двигунів у відносних одиницях становить – 210. Це значення будемо зберігати в числовій змінній `Speed_motor` – це зручно для корегування змінною під час подальшої роботи з кодом – `uint8_t motorSpeed = 210`.

4. Будуємо мапу маршрутослідування локальної навігації відповідно до отриманих даних з варіанта практичної роботи, яка має такий загальний вигляд:

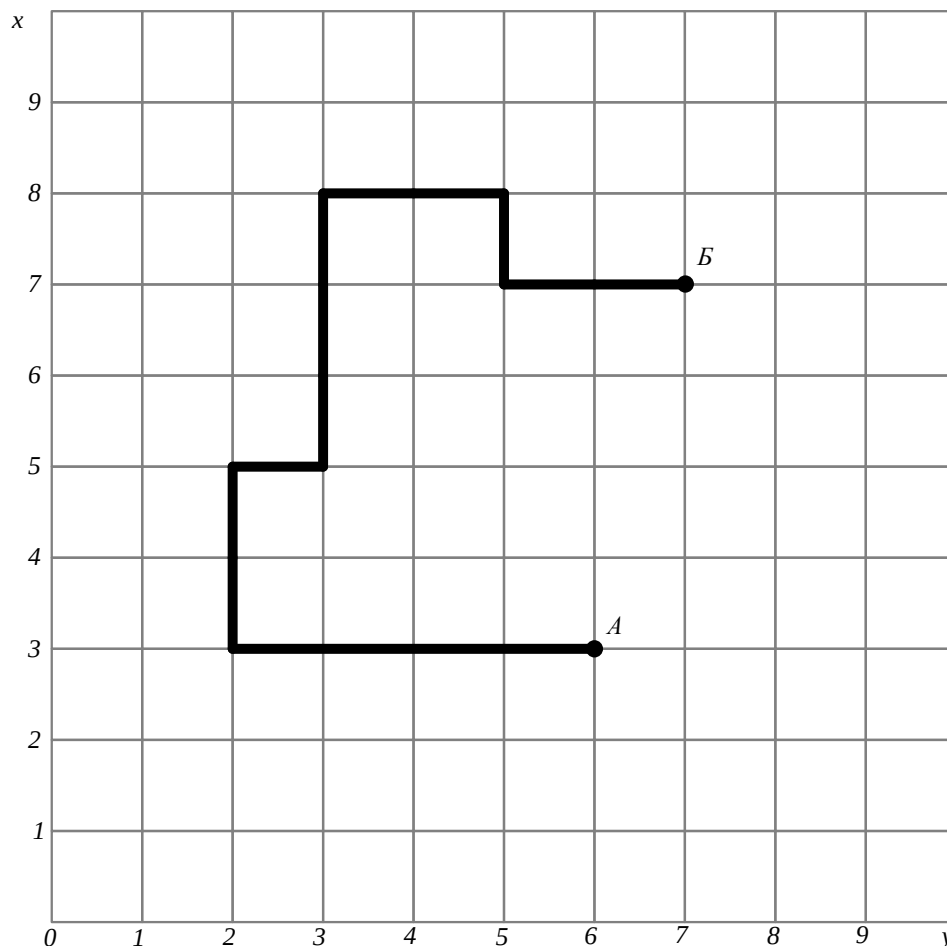


Рисунок 3.10 – Мапа локальної навігації маршрутослідування мобільного робота

5. Реалізуємо програмний код, що відповідає локальному маршрутослідуванню мобільного робота по заданому маршруту:

```

#include <GyverTM1637.h>
#define CLK 12
#define DIO 11
#include "GyverTM1637.h"
GyverTM1637 disp(CLK, DIO);

uint8_t Speed_motor = 205;
int mas[15] = {0,0,0,2,0,2,1,0,0,2,0,2,1,0,3};
int i = 0;
const uint8_t PIN_LINE_Left = 0;
const uint8_t PIN_LINE_Right = 1;
int FLAG_Left = 0;
int FLAG_Right = 0;
const uint8_t PIN_LINE_Left_POV = 2;
const uint8_t PIN_LINE_Right_POV = 3;
int FLAG_Left_POV = 0;
int FLAG_Right_POV = 0;
int POVOROT = 0;
const int SENSOR_PIN = 10;
int lastState = LOW;
int currentState, Button_State ;

void setup() {
  Serial.begin(9600);
  disp.clear();
  disp.brightness(7);
  .....

```

6. Реалізуємо програмний код, що відповідає персональному маршрутослідуванню мобільного робота по заданій траєкторії відповідно до варіанта.

Для здійснення лінійного руху робота створюємо функцію `Linear_motion`, яка має вигляд:

```

void Linear_motion () {
  digitalWrite(4, HIGH);
  analogWrite (5, Speed_motor);
  digitalWrite(7, LOW);
  analogWrite (6, Speed_motor);
}

```

Для здійснення повороту робота вліво та вправо створюємо функції `turn_left` та `turn_right`, відповідно:

```

void turn_left () {
  digitalWrite(4, HIGH);
  analogWrite (5, Speed_motor);
  digitalWrite(7, HIGH);
  analogWrite (6, Speed_motor);
}

```

```

}
void turn_right () {
    digitalWrite(4, HIGH);
    analogWrite (5, Speed_motor);
    digitalWrite(7, HIGH);
    analogWrite (6, Speed_motor);
}

```

Програмний код, що відповідає персональному маршрутослідуванню мобільного робота по заданому маршруту має такий вигляд:

```

uint8_t left_motor = 4;
uint8_t PWM_right_motor = 5;
uint8_t PWM_left_motor = 6;
uint8_t right_motor = 7;
uint8_t Speed_motor = 210; // Швидкість двигунів рух вперед
float turn_duration = 0.65; // тривалість повороту (с)
uint8_t lineardistance = 2; // тривалість руху прямо (с)
uint8_t SENSOR_PIN = 10;

void setup() {
    for (int i = 4; i < 8; i++) {
        pinMode(i, OUTPUT);
    }
    pinMode(SENSOR_PIN, INPUT);
label:
    bool currentState = digitalRead(SENSOR_PIN);
    if ((currentState) == LOW) {
        goto label;
    }
}

void loop() {
    linear_motion();
    delay (lineardistance*1000*4);
    turn_right ();
    linear_motion();
    delay (lineardistance*1000*2);
    turn_right();
    linear_motion();
    delay (lineardistance*1000);
    turn_left();
    linear_motion();
    delay (lineardistance*1000*3);
    turn_right();
    linear_motion();
    delay (lineardistance*1000*2);
    turn_right();
    linear_motion();
}

```

```

    delay (lineardistance*1000);
turn_left();
    linear_motion();
    delay (lineardistance*1000*2);
    stop_motion() ;
    delay(2000000);
}

void linear_motion() { // рух прямо
    digitalWrite(right_motor, HIGH);
    analogWrite (PWM_right_motor, Speed_motor);
    digitalWrite(left_motor, LOW);
    analogWrite (PWM_left_motor, Speed_motor);
}
void turn_right() { // поворот праворуч
    digitalWrite(right_motor, LOW);
    analogWrite(PWM_right_motor, Speed_motor);
    digitalWrite(left_motor, LOW);
    analogWrite(PWM_left_motor, Speed_motor);
    delay(turn_duration * 1000);
}
void turn_left() { // поворот ліворуч
    digitalWrite(right_motor, HIGH);
    analogWrite(PWM_right_motor, Speed_motor);
    digitalWrite(left_motor, HIGH);
    analogWrite(PWM_left_motor, Speed_motor);
    delay(turn_duration * 1000);
}
void stop_motion() { // зупинка руху
    analogWrite(PWM_right_motor, 0);
    analogWrite(PWM_left_motor, 0);
    delay(200);
}

```

3.3 Зміст звіту

1 Звіт з практичної роботи має бути виконаний на аркушах формату А4 та відповідати варіанту практичної роботи.

2 Звіт має містити: назву практичної роботи, її мету і короткі теоретичні відомості.

3 В розділі «Результати виконання практичної роботи» здобувач має покроково подати хід виконання практичної роботи відповідно до підрозділу 3.2 цієї практичної роботи.

4. Написати висновки до практичної роботи.

Таблиця 3.2 – Варіанти завдань до практичної роботи

№ інд. навч. плану		0	1	2	3	4	5	6	7	8	9
Дес.	Од.										
Локальна навігація											
	1	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗
	2	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗
	3	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗
	4	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗
	5	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗
	6	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗	↖↗
x		9	4	3	6	7	9	0	3	7	2
y		3	6	7	9	0	9	2	8	2	7
Персональна навігація											
	LDR, с.	1,5	1,8	2,1	1,35	1,65	1,93	1,8	2,14	1,7	1,4
	ш-ть, од.	255	185	200	190	240	220	190	230	250	205
Об.		↗↖	↗↖	↗↖	↗↖	↗↖	↗↖	↗↖	↗↖	↗↖	↗↖

Примітка: при зміні напрямку руху триколісний робот має здійснювати:

↗↖ – обертання навколо своєї осі;

↗ – зміна траєкторії руху по дузі.

Контрольні запитання

1. Які завдання вирішує персональна навігація мобільних роботів?
2. Які датчики використовують засоби персональної навігації мобільних роботів?
3. Як здійснюється персональна навігація за допомогою засобів одометрії?
4. Які параметри потрібно задавати для переміщення робота з диференціальним приводом по дузі з заданим радіусом та кутом повороту?
5. Як здійснити переміщення вздовж перешкод або стін у приміщенні?
6. Які завдання вирішує локальна навігація мобільних роботів?
7. Які вказівники маршруту та датчики використовують системи маршрутослідування?

Практична робота № 4

БУДОВА МАНІПУЛЯТОРА. СТРУКТУРНО-КІНЕМАТИЧНИЙ АНАЛІЗ

Мета роботи: ознайомитись з робочими органами маніпуляторів, провести структурно-кінематичний аналіз і ознайомитись з основними властивостями кінематичних ланок.

4. 1 Теоретичні відомості [17, 18, 19, 20, 21, 22, 23, 24]

Промисловий робот – автоматична машина, що складається з маніпулятора та пристрою програмного управління його рухом, призначена для заміни людини під час виконання основних та допоміжних операцій у виробничих процесах.

Маніпулятор – сукупність просторового важільного механізму та системи приводів, що здійснює певні дії (маніпуляції), під управлінням програмованого автоматичного пристрою або людини-оператора, аналогічні діям руки людини.

Призначення та область застосування

Промислові роботи-маніпулятори призначені для заміни людини під час виконання основних та допоміжних технологічних операцій у процесі промислового виробництва. Водночас вирішується важливе соціальне завдання – звільнення людини від робіт, пов'язаних з небезпеками для здоров'я або з тяжкою фізичною працею, а також від простих монотонних операцій, що не потребують високої кваліфікації. Гнучкі автоматизовані виробництва, що створюються на базі промислових роботів, дозволяють вирішувати завдання автоматизації на підприємствах з широкою номенклатурою продукції у разі дрібносерійного та поштучного виробництва. Копіювальні маніпулятори, керовані людиною-оператором, потрібні у випадку виконання різних робіт з радіоактивними матеріалами. Крім того, ці пристрої незамінні під час виконання робіт у космосі, під водою, у хімічно активних середовищах.

Класифікація промислових роботів-маніпуляторів

Промислові роботи та маніпулятори класифікуються за такими ознаками:

- *за характером виконуваних технологічних операцій:*

- основні;
- допоміжні;
- універсальні;
- *за видом виробництва:*
 - ливарні;
 - зварювальні;
 - ковальсько-пресові;
 - для механічної обробки;
 - складальні;
 - фарбувальні;
 - транспортно-складські;
- *за системою координат руки маніпулятора:*
 - прямокутна;
 - циліндрична;
 - сферична;
 - сферична кутова (ангулярна);
 - інші;
- *за вантажопідйомністю:*
 - надлегкі (до 10 Н);
 - легкі (до 100 Н);
 - середні (до 2000 Н);
 - важкі (до 10000 Н);
 - надважкі (понад 10000 Н);
- *за типом силового привода:*
 - електромеханічний;
 - пневматичний;
 - гідравлічний;
 - комбінований;
- *за рухливістю основи:*
 - мобільні;
 - стаціонарні;
- *за видом програми:*
 - з постійним програмним забезпеченням;
 - перепрограмовані;
 - адаптивні;
 - з елементами штучного інтелекту;

Основні технічні показники промислових роботів

Основні технічні показники промислових роботів (ПР) визначаються передбачуваною областю застосування й умовами виробництва, для яких призначений робот [1].

Вантажопідйомність промислових роботів визначається як сумарна вантажопідйомність його рук.

Вантажопідйомність руки промислових роботів – найбільша маса об'єктів маніпулювання (включно масу захоплювального пристрою), які можуть переміщатися рукою за заданих умов (за максимальної або мінімальної швидкості, за максимального вильоту руки й т. п.).

У разі виконання операцій декількома руками разом із сумарною вантажопідйомністю промислових роботів потрібно вказувати вантажопідйомність однієї руки.

Для деяких типів промислових роботів важливим показником є зусилля (або крутний момент), що розбудовується виконавчим механізмом за заданих умов. До таких показників можна віднести зусилля затискача (захоплювача, утримання) об'єкта маніпулювання захоплювальним пристроєм; робоче зусилля руки ПР уздовж її поздовжньої осі; крутний момент під час ротації захоплювального пристрою.

Число ступенів рухливості промислових роботів – сума можливих координатних рухів об'єктів маніпулювання відносно опорної системи (стояки, підставки).

Похибка позиціонування – відхилення заданої позиції виконавчого механізму від фактичної у випадку багаторазового позиціонування (повторення руху). Похибка позиціонування може оцінюватися в лінійних або кутових одиницях. Стосовно промислових роботів важливим показником є сумарна похибка позиціонування всіх виконавчих механізмів, наведена до фактичного положення об'єкта маніпулювання, що відрізняється від заданого у програмі роботи. Такий показник називають похибкою позиціонування робочого органа ПР, він визначається як величина відхилення робочого органа ПР від заданого керівною програмою.

Похибка відпрацьовування траєкторії робочого органа промислового робота або маніпулятора – відхилення фактичної траєкторії від заданої у програмі.

Робочий простір промислового робота (маніпулятора) – простір, у якому може перебувати виконавчий орган (пристрій) промислового робота або маніпулятора.

Робоча зона промислового робота або маніпулятора – простір, у якому може перебувати робочий орган (наприклад рука) під час функціонування промислового робота або маніпулятора.

Зона обслуговування промислового робота або маніпулятора – частина робочої зони, де повністю зберігаються задані (паспортні) значення технічних характеристик.

У випадку роботи декількох промислових роботів або маніпуляторів як характеристика робототехнічного комплексу наводиться **зона спільного обслуговування** – частина простору, у якому переміщення об'єкта маніпулювання можуть виконуватися декількома промисловими роботами або маніпуляторами.

Мобільність промислового робота визначається його можливістю робити рухи. За мобільністю роботів розділяють на дві групи: стаціонарні та пересувні.

Конструкція маніпулятора промислового робота

Маніпулятор промислового робота за своїм функціональним призначенням має забезпечувати рух вихідної ланки та закріпленого в ньому об'єкта маніпулювання у просторі по заданій траєкторії та із заданою орієнтацією. Для повного виконання цієї вимоги потрібно, щоб основний важільний механізм маніпулятора мав не менше шести ступенів рухливості, причому рух кожного з них має бути керованим. Промисловий робот-маніпулятор із шістьма ступенями рухливості є складною автоматичною системою. Ця система складна як у виготовленні, так і в експлуатації. Тому в реальних конструкціях промислових роботів часто використовуються механізми з кількістю ступенів рухливості, меншою шести. Найпростіші маніпулятори мають три, рідше два, ступеня рухливості.

Поєднання та взаємне розташування ланок та з'єднань визначає число ступенів рухливості, а також область дії маніпуляційної системи робота. Зазвичай припускають, що *три з'єднання* у виконавчому механізмі маніпулятора реалізують транспортні ступені рухливості (так звана «*РУКА*»), інші реалізують орієнтаційні ступені рухливості (відповідаючи за необхідну орієнтацію робочого органу) («*КИСТЬ*»).

Залежно від виду перших трьох з'єднань більшість роботів відносять до однієї з чотирьох категорій:

- роботи, що працюють у декартовій системі координат, у яких усі три початкових з'єднання є поступальними (рисунок 4.1);

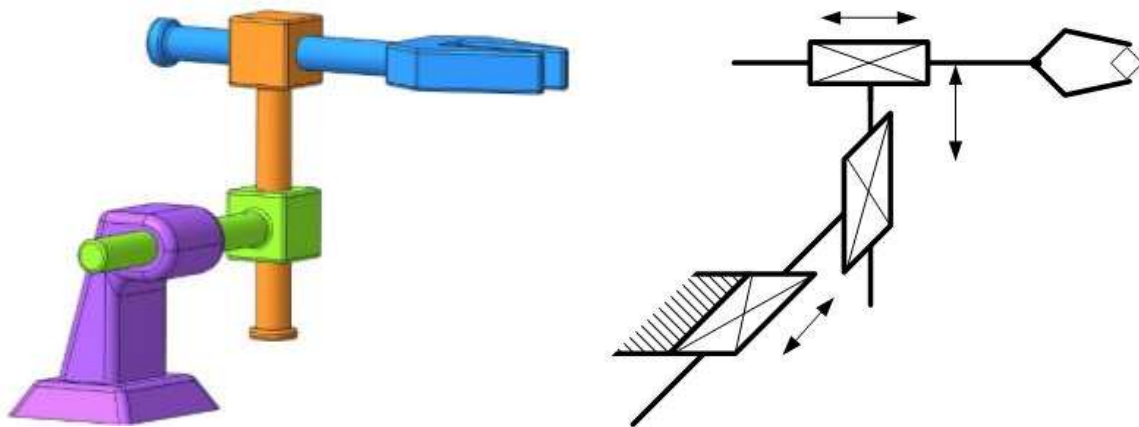


Рисунок 4.1 – Структурно-кінематична схема та її 3D модель робота-маніпулятора з декартовою системою координат

- роботи, що працюють у циліндричній системі координат, у яких серед початкових з'єднань два поступальних і одне оберতальне (рисунок 4.2);

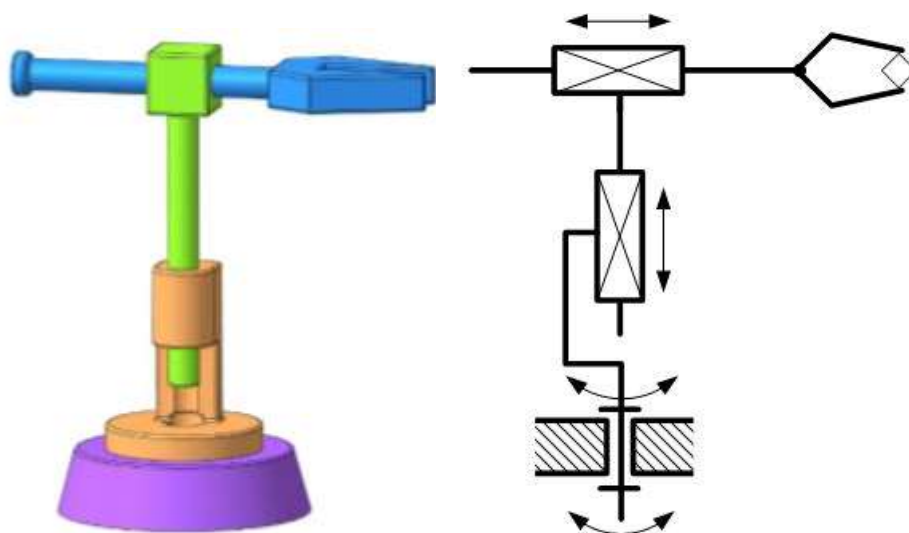


Рисунок 4.2 – Структурно-кінематична схема та її 3D модель робота-маніпулятора з циліндричною системою координат

- роботи, що працюють у сферичній системі координат, у яких серед початкових з'єднань одне поступальне і два оберতальних (рисунок 4.3);

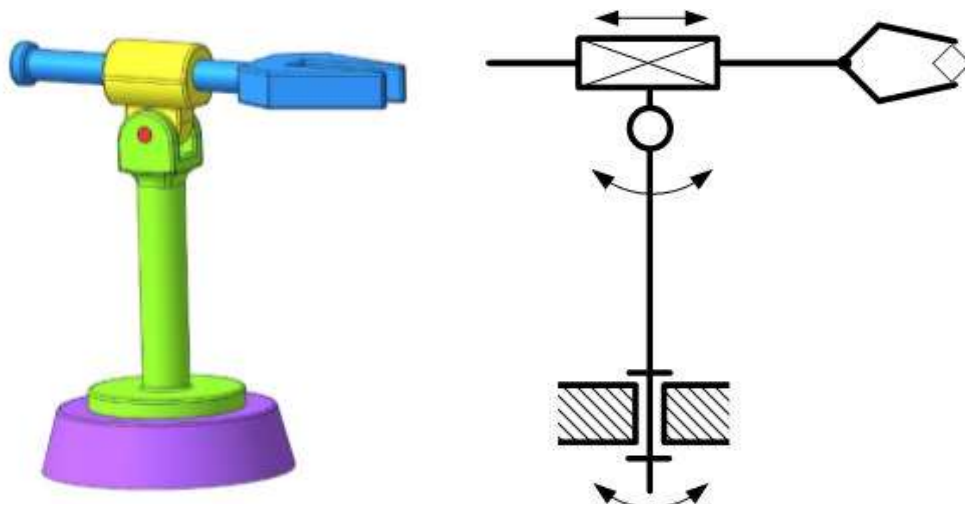


Рисунок 4.3 – Структурно-кінематична схема та її 3D модель робота-маніпулятора зі сферичною системою координат

- роботи, що працюють у кутовій, або обертальній, системі координат, у яких усі три початкових з'єднання є обертальними (рисунок 4.4).

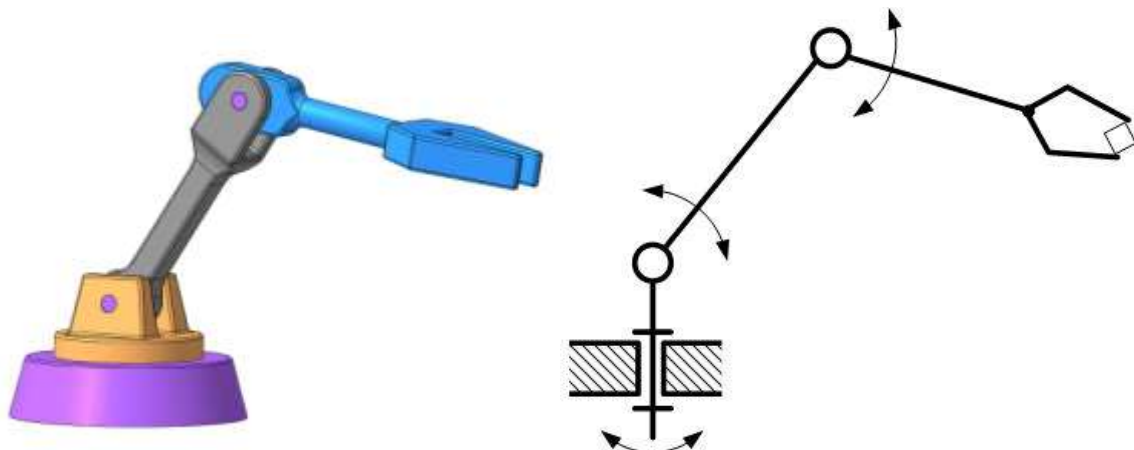


Рисунок 4.4 – Структурно-кінематична схема та її 3D модель робота-маніпулятора з кутовою системою координат

Розглянемо структурну та функціональну схеми маніпулятора

Розглянемо для прикладу структурну та функціональну схеми промислового робота-маніпулятора з числом рухливості – 5. Основний механізм руки маніпулятора складається з нерухомої ланки 0 та чотирьох рухомих ланок 1, 2, 3 та 4 (рисунок 4.5).

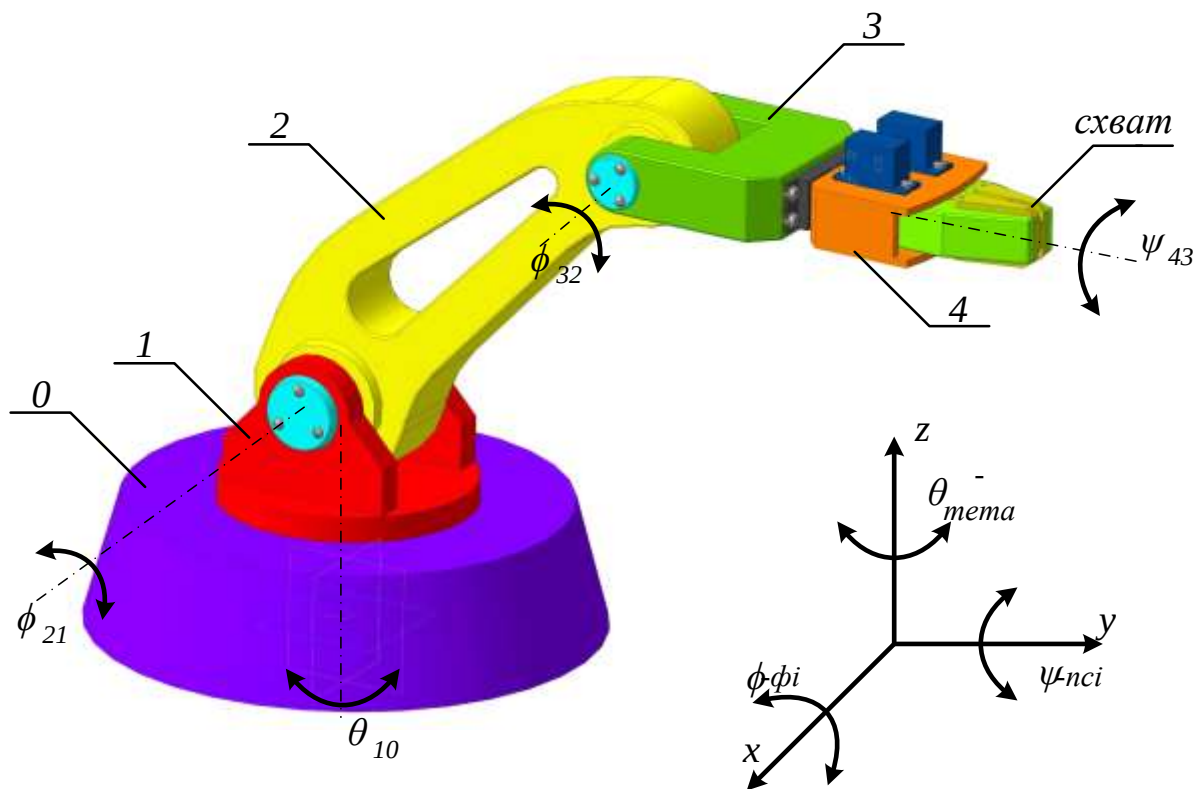


Рисунок 4.5 – Схема будови маніпулятора

Механізм цього маніпулятора відповідає кутовій системі координат.

У цій системі ланка 1 може обертатися відносно ланки 0 (відносно кутового зміщення θ_{10}), ланка 2 може обертатися відносно ланки 1 (відносно кутового зміщення ϕ_{21}), ланка 3 може обертатися відносно ланки 2 (відносно кутового зміщення ϕ_{32}), ланка 4 може обертатися відносно ланки 3 (відносно кутового зміщення ψ_{43}),

На кінці ланки 4 закріплено захоплювальний пристрій або захоплювач, призначений для захоплення та утримання об'єкта маніпулювання під час роботи маніпулятора. Ланки основного важільного механізму маніпулятора утворюють між собою 4 однорухливі кінематичні пари (чотири обертельних пари – А) і можуть забезпечити переміщення об'єкта в просторі з управлінням його орієнтацією. Для виконання кожного з відносних рухів маніпулятор має бути оснащений приводами, що складаються з двигунів з редуктором і системи датчиків зворотного зв'язку. Оскільки рух об'єкта здійснюється за заданим законом руху, то в системі мають бути пристрої, що зберігають і задають програму руху.

Рухи, що забезпечуються маніпулятором, поділяються на:

- *глобальні* (для робіт з рухомою основою) – рухи стояка маніпулятора, які істотно перевищують розміри механізму;
- *регіональні* (транспортні) – рухи, що забезпечуються першими трьома ланками маніпулятора або його «рукою», величину яких можна порівняти з розмірами механізму;
- *локальні* (орієнтувальні) – рухи, що забезпечуються ланками маніпулятора, які утворюють його «кисть», величина яких значно менша за розміри механізму.

Відповідно до цієї класифікації рухів, у маніпуляторі можна виділити дві ділянки кінематичного кола з різними функціями: механізм руки та механізм кисті. Під «рукою» розуміють ту частину маніпулятора, яка забезпечує переміщення центра захоплювача – точки М (регіональні рухи захоплювача); під «кистю» – ті ланки та пари, які забезпечують орієнтацію захоплювача (локальні рухи захоплювача).

Структурна схема механізму – його графічне зображення, у якому показані: стояк, рухливі ланки, кінематичні пари та їх взаємне розташування (рисунок 4.6).

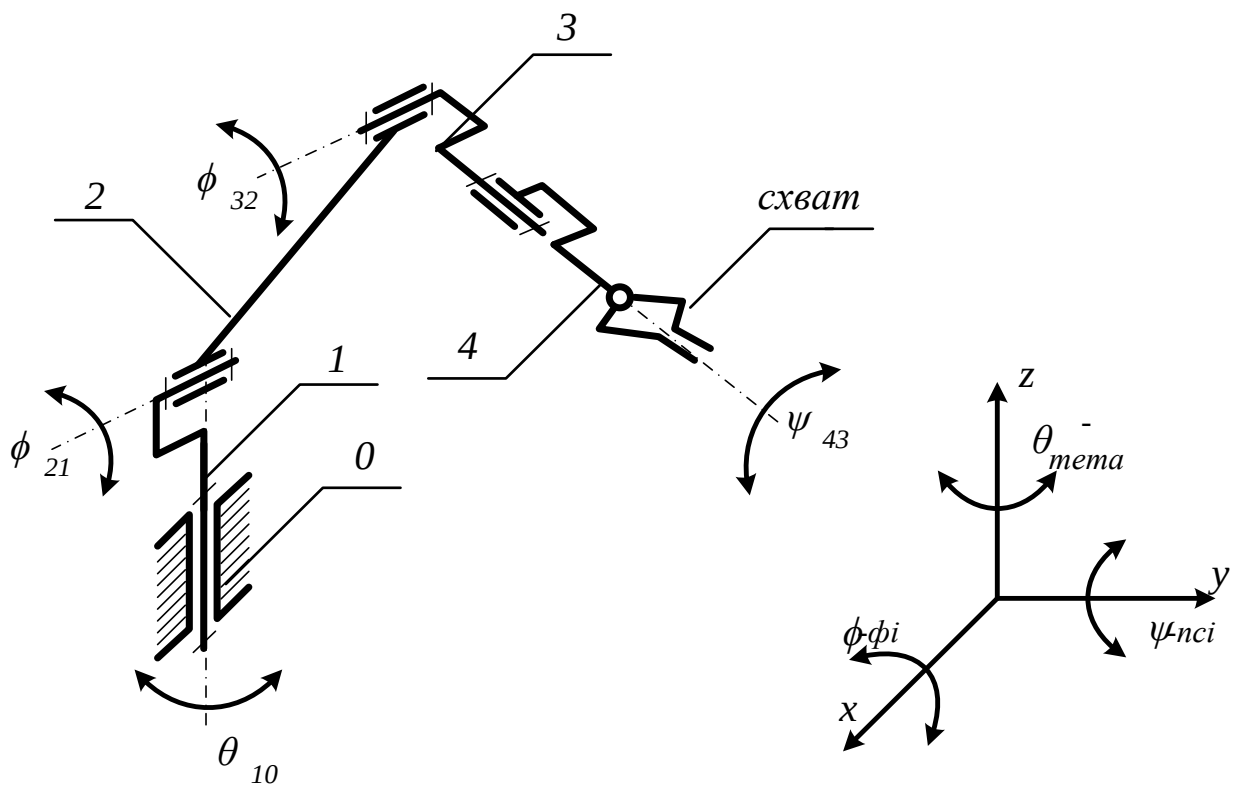


Рисунок 4.6 – Структурна схема маніпулятора

Цей механізм складається з 4 рухомих ланок і 4 кінематичних пар: чотирьох однорухливих обертальних пар $A_{1\ 06}$, $B_{1\ 06}$, $C_{1\ 06}$, $D_{1\ 06}$

Кінематичні пари маніпулятора характеризуються: іменем або позначенням КП – велика літера латинського алфавіту (А, В, С

тощо); ланками, що утворюють пару (0/1,1/2 тощо); відносним рухом ланок у парі (для однорухливих пар – обертальні, поступальні та гвинтові); рухливістю КП (для нижчих пар від 1 до 3, для вищих пар від 4 до 5); віссю орієнтації осі КП відносно осей базової чи локальної системи координат.

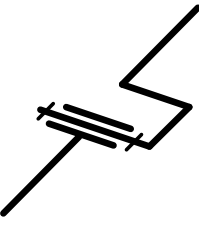
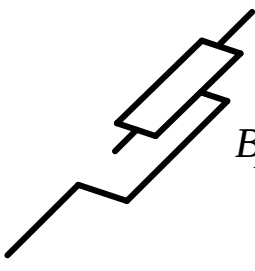
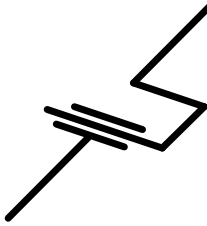
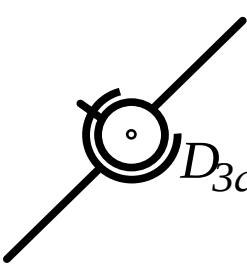
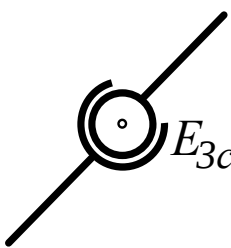
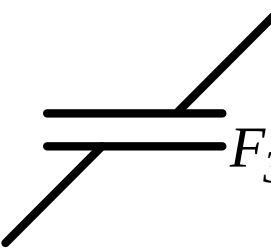
Кінематичним ланцюгом називається система ланок, що утворюють між собою кінематичні пари.

Проста кінематична пара. Ланцюг, в якому кожна ланка входить не більше ніж у дві кінематичні пари, називається простим.

Незамкненим називається такий *кінематичний ланцюг*, в якому є ланки, що входять тільки в одну кінематичну пару.

Графічне зображення елементів схеми виконується з урахуванням прийнятих умовних позначень. У таблиці 4.1 наведено умовні позначення кінематичних пар.

Таблиця 4.1 – Умовні позначення кінематичних пар

Однорухлива обертальна	Однорухлива поступальна	Дворухлива циліндрична
 $A_{1об.}$	 $B_{1п.}$	 $C_{2об.}$
Дворухлива сферична	Трирухлива сферична	Трирухлива плоска
 $D_{3сф.}$	 $E_{3сф.}$	 $F_{3п.}$

Кінематичні пари маніпуляторів забезпечуються приводами, що містять двигуни та гальмівні пристрої, тому в схемах маніпуляторів зазвичай використовуються однорухливі кінематичні пари: – обертальні або поступальні.

Формула будови маніпулятора – математичний запис структурної схеми маніпулятора, що містить інформацію про кількість його

рухливостей, вид кінематичних пар та їх орієнтацію відносно осей базової системи координат (системи, пов'язаної з нерухомою ланкою).

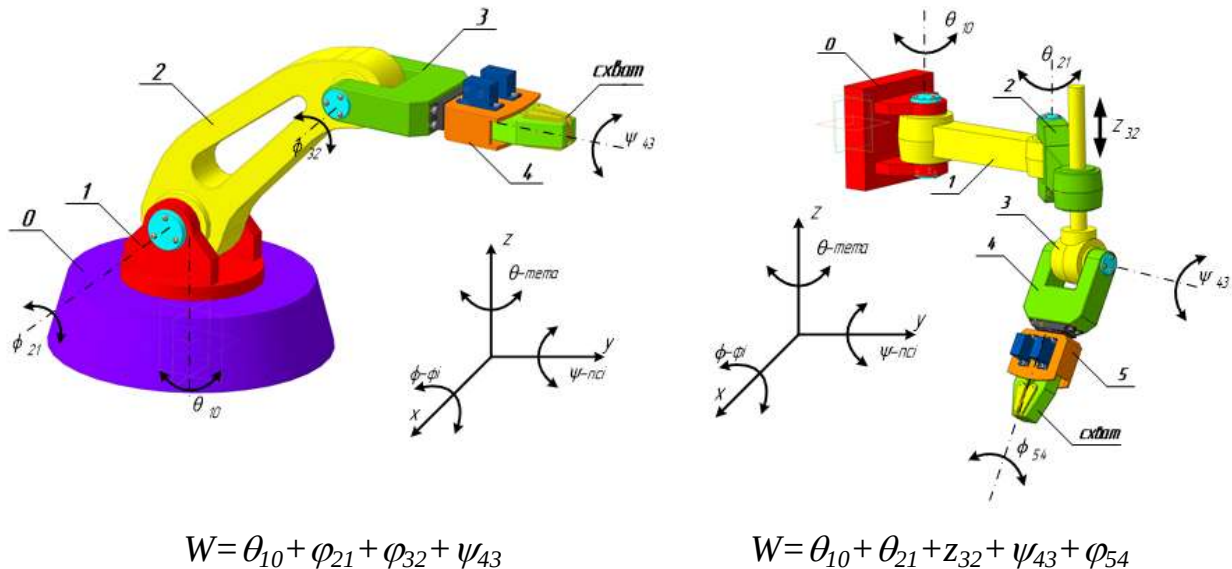


Рисунок 4.7 – Формула будови маніпулятора

Робочий простір маніпулятора – частина простору, обмежена поверхнями, які описуються можливим положенням його ланок.

Зона обслуговування маніпулятора – частина простору, відповідна можливим положенням центра захоплювача маніпулятора. Зона обслуговування є важливою характеристикою маніпулятора. Вона визначається структурою й системою координат руки маніпулятора, а також конструктивними обмеженнями, накладеними на відносні переміщення ланок у кінематичній парі.

Рухливість маніпулятора R – число незалежних узагальнених координат, що однозначно визначає положення захоплювача в просторі.

$$R = 6 * n - \sum_{i=1}^5 (6 - i) * p_i,$$

де n – кількість ланок;

i – клас кінематичної пари (1,2,...,5);

p_i – кількість кінематичних пар i -го класу.

Маневреність маніпулятора M – рухливість маніпулятора за умови зафіксованого (нерухомого) захоплювача.

$$M = R - 6.$$

Кут сервісу. Можливість зміни орієнтації захоплювача у випадку розміщення його центра в заданій точці зони обслуговування характеризується кутом сервісу – тілесним кутом Ψ , який може описати остання ланка маніпулятора (ланка, на якій закріплений захоплювач) за фіксації центра захоплювача в заданій точці зони обслуговування.

$$\psi = \frac{f_c}{l}.$$

де f_c – площа сферичної поверхні, описувана точною C ланки 3;
 l – довжина ланки 3 (рис. 4.8).

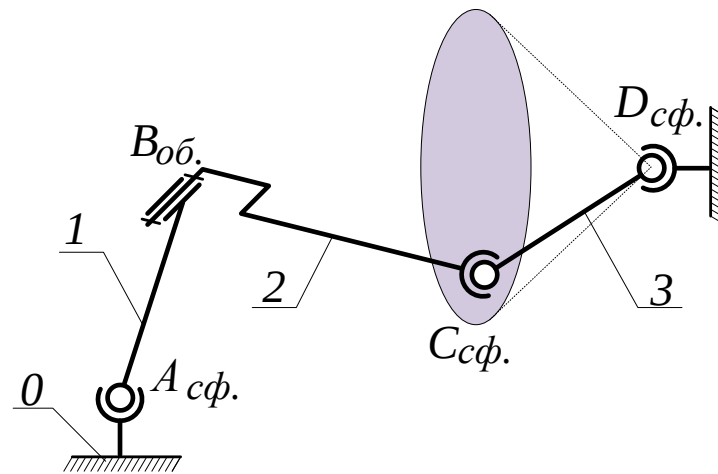


Рисунок 4.8 – Кут сервісу маніпулятора

4.2 Послідовність виконання практичної роботи

4.2.1 Визначити формулу будови маніпулятора та систему координат *оxyz* відповідно до варіанта практичної роботи (див. таблицю 4.5). Останніх дві цифри індивідуального навчального плану здобувача є варіантом практичної роботи.

4.2.2 Побудувати структурну схему маніпулятора, на якій осі кінематичних пар орієнтувати по осях заданої системи координат.

4.2.3 Позначити на структурній схемі ланки – цифрами, а кінематичні пари – латинськими літерами (в позначенні КП індексами вказати рухливість і вид КП: однорухлива обертальна – 1об., однорухлива поступальна – 1п., дворухлива циліндрична – 2ц., трирухлива сферична – 3сф. і т. д.).

4.2.4 Визначити ланки, що утворюють відповідно «Руку» і «Кисть» маніпулятора.

4.2.5 Визначити систему координат «руки» маніпулятора.

4.2.6 Визначити рухливість і маневреність маніпулятора.

4.2.7 Визначити основні параметри структурної схеми маніпулятора і параметри його кінематичних пар та занести їх в таблицю 4.2.

Таблиця 4.2 – Основні параметри структурної схеми маніпулятора

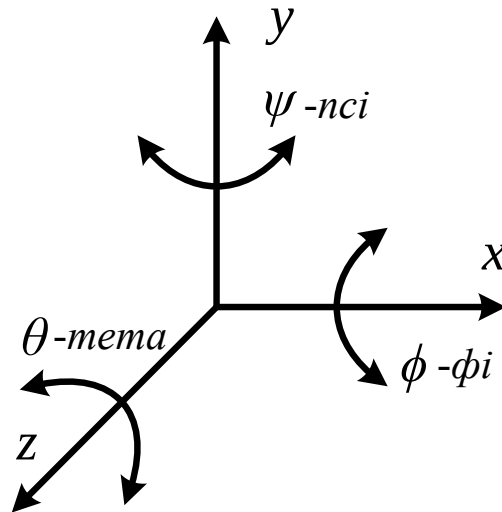
1	Формула будови					
2	Рухомі ланки, що утворюють в маніпуляторі	«руку»				
		«кисть»				
3	Система координат «руки» маніпулятора					
4	Число ланок	загальне	k			
		рухомих	n			
5	Число однорухливих кінематичних пар	обертальних	p _{1об}			
		поступальних	p _{1п}			
6	Рухливість маніпулятора	R				
7	Маневреність маніпулятора	M				
8	Кінематичні пари маніпулятора					
Позначення КП						
Ланки, що утворюють КП						
рухливість						
Рухливість в КП						
Вісь орієнтації КП						

4.3 Приклад виконання практичної роботи

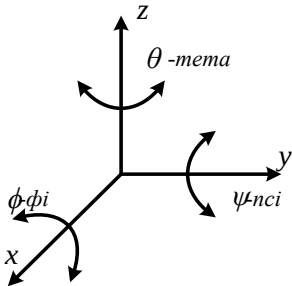
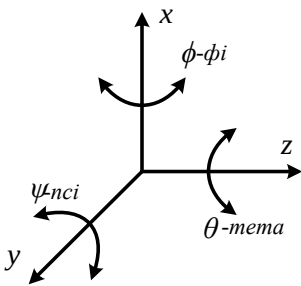
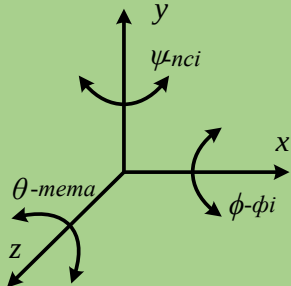
1. Вибираємо варіант практичної роботи.

Вибір варіанту для номера індивідуального навчального плану здобувача №01-22-236. Дві останні цифри 36 отже, відповідно до таблиці 4.3 маємо:

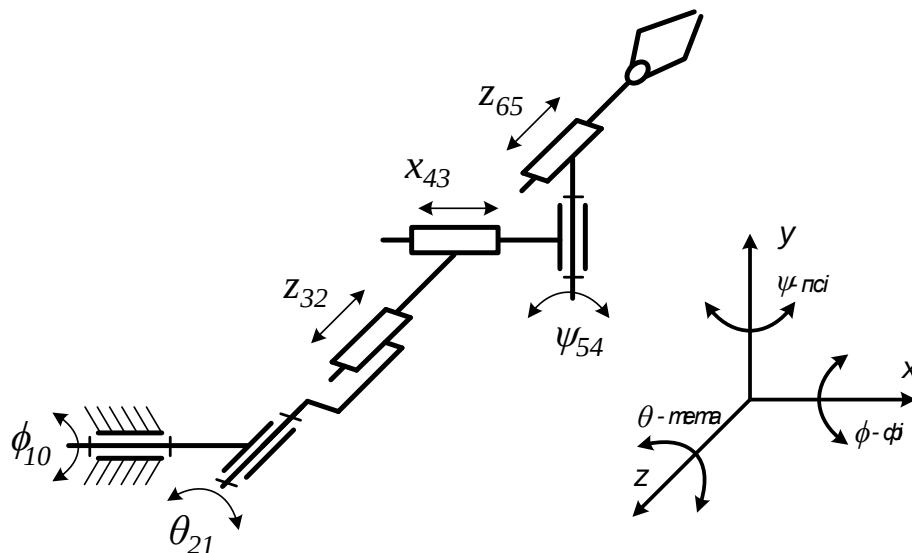
- формула будови маніпулятора - $W = \varphi_{10} + \theta_{21} + z_{32} + x_{43} + \psi_{54} + z_{65}$
- систему координат:



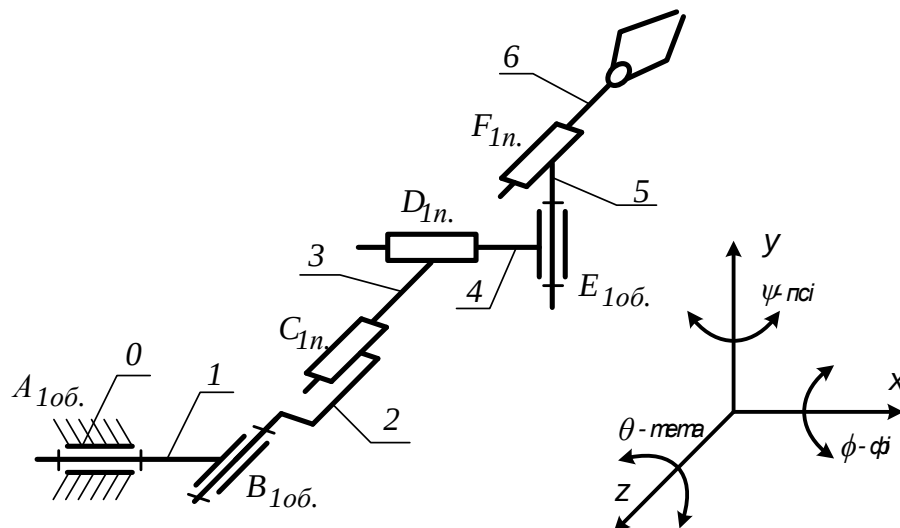
Таблиця 4.3 – Приклад вибору варіантів практичної роботи

№ інд. навч. плану		0	1	2	3	4	5	6	7	8	9
Дес.	Од.										
A_{10}		θ	φ	Ψ	φ	y	z	θ	θ	z	y
B_{21}		x	z	θ	θ	φ	x	x	φ	φ	Ψ
C_{32}		Ψ	θ	φ	z	x	θ	Ψ	y	θ	x
	D_{43}	x	Ψ	x	y	x	Ψ	x	θ	y	θ
	E_{54}	y	θ	y	θ	y	θ	Ψ	x	φ	x
	F_{65}	θ	φ	z	φ	z	φ	z	Ψ	y	z
систему координат (обирається по одиницях)											

2. Будуємо структурну схему маніпулятора на якій осі КП орієнтуємо по осях заданої системи координат



3. Позначаємо на структурній схемі ланки – цифрами, а кінематичні пари – латинськими літерами:



Ми отримаємо: кількість ланок $n=7$

Цей механізм складається з шести рухомих ланок ($n=7$) і 6 кінематичних пар: трьох однорухливих обертальних пар $A_{10б}$, $B_{10б}$, $E_{10б}$ та трьох однорухливих поступальних кінематичних пар $C_{1н}$, $D_{1н}$, $F_{1н}$.

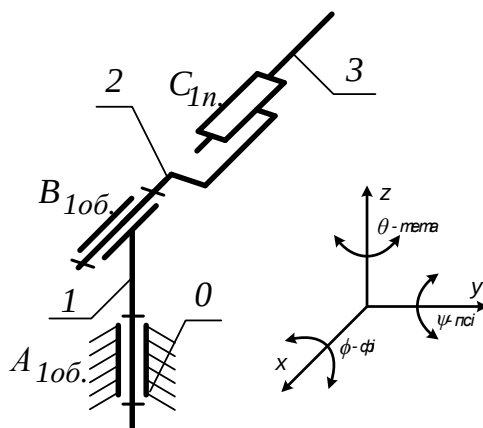
Також кінематичні ланки, що утворюють КП: 1/0; 2/1; 3/2; 4/3; 5/4; 6/5;

4. Визначаємо ланки, що утворюють відповідно:

- «Руку» ланки 1, 2, 3;
- «Кисть» ланки 4, 5, 6.

5. Визначаємо систему координат «руки» маніпулятора.

У нашому випадку дві кінематичних пари обертальні та одна поступальна, що відповідає сферичній системі координат маніпулятора.



6. Визначаємо рухливість маніпулятора R – число незалежних узагальнених координат, що однозначно визначає положення захоплювача в просторі.

$$R = 6 * n - \sum_{i=1}^5 (6 - i) * p_i$$

де $n=6$;

клас кінематичної пари ($i=1$);

$p_1 = 6$ – кількість кінематичних пар 1-го класу.

$$R = 6 * 6 - (6 - 1) * 6 = 36 - (5 * 6) = 6$$

7. Визначаємо маневреність маніпулятора M – рухливість маніпулятора за зафіксованого (нерухомого) захвату

$$M = 6 - 6 = 0.$$

8. Заносимо до таблиці основні параметри структурної схеми маніпулятора.

Таблиця 4.4 – Основні параметри структурної схеми маніпулятора

1.	Формула будови	$W = \varphi_{10} + \theta_{21} + z_{32} + x_{43} + \psi_{54} + z_{65}$					
2.	Рухомі ланки, що утворюють в маніпуляторі	«Руку»	1	2	3		
		«Кисть»	4	5	6		
3.	Система координат «Руки» маніпулятора		сферична				
4.	Число ланок	загальне	k	7			
		рухомих	n	6			
5.	Число однорухливих кінематичних пар	обертальних	p _{1об.}	3			
		поступальних	p _{1п.}	3			
6.	Рухливість маніпулятора	R	6				
7.	Маневреність маніпулятора	M	0				
8.	Кінематичні пари маніпулятора						
Позначення КП		A	B	C	D	E	F
Ланки, що утворюють КП		1-0	2-1	3-2	4-3	5-4	6-5
Вид КП		оберт.	оберт.	пост.	пост.	оберт.	пост.
Ступінь рухливості КП		1	1	1	1	1	1
Вісь орієнтації КП		x	z	z	x	y	z

4.4 Зміст звіту

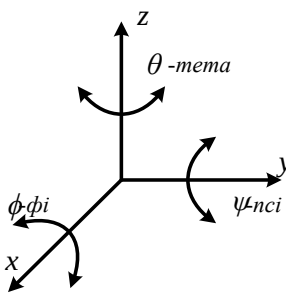
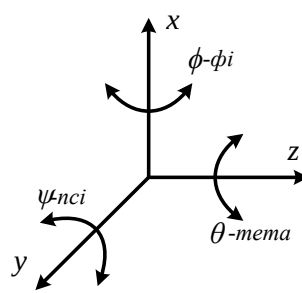
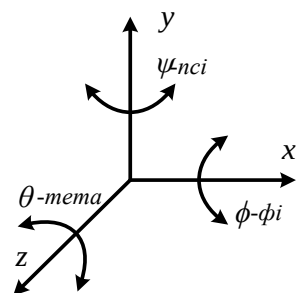
1. Звіт з практичної роботи має бути виконаний на аркушах формату А4 та відповідати варіанту практичної роботи.

2. Звіт має містити: назву практичної роботи, її мету і короткі теоретичні відомості.

3. В розділі «Результати виконання практичної роботи» здобувач має покроково описати хід виконання практичної роботи відповідно до підрозділу 4.2 цієї практичної роботи.

4. Написати висновки до практичної роботи.

Таблиця 4.5 – Варіанти завдань практичної роботи

№ інд. навч. плану		0	1	2	3	4	5	6	7	8	9
Дес.	Од.										
A ₀₁		θ	φ	Ψ	x	y	z	θ	θ	z	y
B ₁₂		x	z	θ	θ	φ	x	x	φ	φ	Ψ
C ₃₂		Ψ	θ	φ	z	x	θ	Ψ	y	θ	x
	D ₄₃	x	Ψ	x	y	x	Ψ	x	θ	y	θ
	E ₅₄	y	θ	y	θ	y	θ	y	x	φ	x
	F ₆₅	θ	φ	z	φ	z	φ	z	Ψ	y	z
систему координат (обирається по Од.)											

Контрольні запитання

1. Наведіть основні технічні показники промислових роботів-маніпуляторів.
2. Що таке вантажопідйомність промислового робота-маніпулятора?
3. Що таке число ступенів рухливості промислового робота-маніпулятора?
4. Що таке похибка позиціонування промислового робота-маніпулятора?
5. Що таке похибка відпрацьовування траєкторії промислового робота-маніпулятора?
6. Що таке робочий простір промислового робота-маніпулятора?
7. Що таке робоча зона промислового робота-маніпулятора?
8. Що таке зона обслуговування промислового робота-маніпулятора?
9. Наведіть приклад робота-маніпулятора з декартовою системою координат.
10. Наведіть приклад робота-маніпулятора з циліндричною системою координат
11. Наведіть приклад робота-маніпулятора з сферичною системою координат.
12. Наведіть приклад робота-маніпулятора з кутовою системою координат.
13. Як називається графічне зображення, у якому показані стійка, рухливі ланки, кінематичні пари та їх взаємне розташування?
14. Як називається система ланок, що утворюють між собою кінематичні пари?
15. Що таке проста кінематична пара?
16. Що називається робочим простором промислового робота-маніпулятора?
17. Що таке зона обслуговування промислового робота-маніпулятора?
18. Наведіть формулу розрахунку рухливості промислового робота-маніпулятора.
19. Що таке кут сервісу промислового робота-маніпулятора?

Практична робота № 5

МЕТОДИ РОЗВ'ЯЗАННЯ ПРЯМОЇ ЗАДАЧІ КІНЕМАТИКИ

Мета роботи: ознайомлення з різними кінематичними параметрами, що використовуються для опису руху маніпулятора. Розв'язання прямої задачі кінематики на прикладі маніпуляторів з поступальними та обертальними ступенями рухливості.

5.1 Теоретичні відомості [19, 20, 21, 22]

Проектування роботів-маніпуляторів потребує розв'язання таких основних задач кінематики:

- *пряма задача* – полягає у визначенні положення та орієнтації робочого органу маніпулятора, а в деяких випадках й інших ланок, за заданими узагальненими координатами;
- *пряма задача швидкостей (прискорень)* – полягає у визначенні швидкості (прискорення) руху робочого органу маніпулятора за заданими узагальненими швидкостями (прискореннями) в кінематичних парах;
- *обернена задача* – полягає у визначенні відносних координат ланок маніпулятора за заданим положенням об'єкта або робочого органу маніпулятора, жорстко з'єднаного з ним;
- *обернена задача швидкостей (прискорень)* – полягає у визначенні необхідних узагальнених швидкостей (прискорень) у кінематичних парах за заданими швидкістю (прискоренням) вихідної ланки.

Пряму задачу, зазвичай, багаторазово використовують під час проектування маніпулятора. За її допомогою можна визначити характеристики робочої області маніпулятора зі складною кінематичною схемою, за наявності обмежень на узагальнені координати, а також визначити характеристики точності, наприклад, похибки положення та орієнтації робочого органу маніпулятора, які зумовлені неточністю виготовлення ланок маніпулятора або неточністю обчислень тих чи інших координат.

Існує багато методів розв'язання цих задач, кожен з яких має свої переваги та недоліки. Вибір того чи іншого методу, як правило, пов'язаний зі специфікою поставленої задачі, а також з особливостями конструкції маніпулятора.

Відомо, що пряма задача кінематики полягає у визначенні координат кінцевого положення робочого органу маніпулятора за значеннями узагальнених координат. Це положення визначають за допомогою

координат, які характеризують відносні переміщення окремих кінематичних пар.

Здебільшого вони ґрунтуються на методах Денавіта-Хартенберга і Лагранжа-Ейлера та використовуються для маніпуляторів з невеликою кількістю зчленувань та ступенів свободи. Для випадків, коли частини суміжних зчленувань маніпулятора мають перетинатись в одній точці або бути паралельними чи перпендикулярними, відомі застосування сучасних підходів на основі кватерніонної кінематичної теорії, ітеративних методів розв'язування задачі, теорії нейронних мереж. Однак, які б методи не застосовувались, для створення ефективної системи керування переміщенням маніпулятора спочатку потрібно розв'язати пряму задачу кінематики, адже від отриманих даних залежить результат розв'язання оберненої задачі [1].

Метод Денавіта-Хартенберга

Для опису обертальних і поступальних зв'язків між сусідніми ланками Денавіт і Хартенберг запропонували матричний метод послідовної побудови систем координат, пов'язаних з кожною ланкою кінематичного ланцюга. Суть подання Денавіта-Хартенберга (ДХ-подання) полягає у формуванні однорідної матриці перетворення розміром 4×4 , яка описує положення системи координат кожної ланки відносно системи координат попередньої ланки. Це дає можливість послідовно перетворювати координати захоплювача маніпулятора із системи відліку, пов'язаної з останньою ланкою, в базову систему відліку, що є інерціальною системою координат для розглядуваної динамічної системи.

Окрім базової системи координат, для кожної ланки на осі її зчленування визначається ортонормована декартова система координат (x_i, y_i, z_i) , де $i = 1, 2, \dots, n$ та дорівнює числу ступенів вільності маніпулятора. Оскільки обертове зчленування має лише один ступінь вільності, кожна система координат (x_i, y_i, z_i) маніпулятора відповідає $(i+1)$ -му зчленуванню і пов'язана з i -ю ланкою. Коли силовий привод збуджує рух в i -му зчленуванні, i -а ланка починає рухатися відносно $(i-1)$ -ї ланки. Оскільки i -а система координат пов'язана з i -ю ланкою, вона рухається разом із нею. Таким чином, n -а система координат рухається разом з останньою n -ю ланкою маніпулятора. Базовою є нульова система координат (x_0, y_0, z_0) – інерціальна система координат маніпулятора.

Отже, для чотириланкового навчального маніпулятора мають бути визначені п'ять систем координат, а саме: (x_0, y_0, z_0) , (x_1, y_1, z_1) , ..., (x_4, y_4, z_4) .

Кожна система координат формується на основі таких трьох правил:

1. Вісь z_{i-1} спрямована вздовж осі i -го зчленування.
2. Вісь x_i перпендикулярна до осі z_{i-1} і спрямована від неї.
3. Вісь y_i доповнює осі x_i та z_i до правої декартової системи координат.

Ці правила залишають свободу у виборі 0-ї системи координат за умови, що вісь z_0 направлена вздовж осі першого зчленування. Остання n -а система координат також може бути вибрана в довільній точці n -ї ланки за умови, що вісь x_n перпендикулярна осі z_{n-1} .

ДХ-подання жорстких ланок залежить від чотирьох геометричних параметрів, що відповідають кожній ланці. Ці чотири параметри повністю описують будь-який обертальний чи поступальний рух і визначаються відповідно до рисунка 5.1 таким чином:

θ_i – приєднаний кут – кут, на який треба повернути вісь x_{i-1} навколо осі z_{i-1} , щоб вона стала співнаправлена з віссю: x_i (знак визначається відповідно до правила правої руки);

d_i – відстань між перетином осі z_{i-1} з віссю x_i і початком $(i-1)$ -ї системи координат, що відраховується вздовж осі;

a_i – лінійне зміщення – відстань між перетином осі z_{i-1} з віссю x_i і початком i -ї системи координат, що відраховується вздовж осі x_i , тобто найкоротша відстань між осями z_{i-1} та z_i ;

α_i – кутове зміщення – кут, на який потрібно повернути вісь z_{i-1} навколо осі x_i , щоб вона стала співспрямованою з віссю z_i (знак визначається відповідно до правила правої руки).

Для обертальних зчленувань параметри d_i , a_i і α_i є характеристиками зчленування, постійними для цього типу робота. Водночас θ_i є змінною величиною, що змінюється під час руху (обертання) i -ї ланки відносно $(i-1)$ -ї. Для поступальних зчленувань θ_i , a_i і α_i – характеристики зчленування, незмінні для цього робота, а d_i – змінна величина.

Нижче наведено опис процедури формування послідовності узгоджених систем координат, пов'язаних з ланками маніпулятора. Алгоритм базується на поданих вище трьох основних правилах побудови ортонормованих систем координат ланок і враховує геометричний зміст параметрів зчленувань і ланок.

Формування системи координат ланок маніпулятора

Для кожної ланки маніпулятора з n ступенями вільності, алгоритм формує ортонормовану систему координат. Вибір систем координат проводиться з урахуванням конфігурації маніпулятора, подібної до конфігурації людської руки. Системи координат нумеруються в порядку зростання – від основи до захоплювача маніпулятора. Взаємне розташування сусідніх ланок описується однорідною матрицею перетворення розмірністю 4×4 .

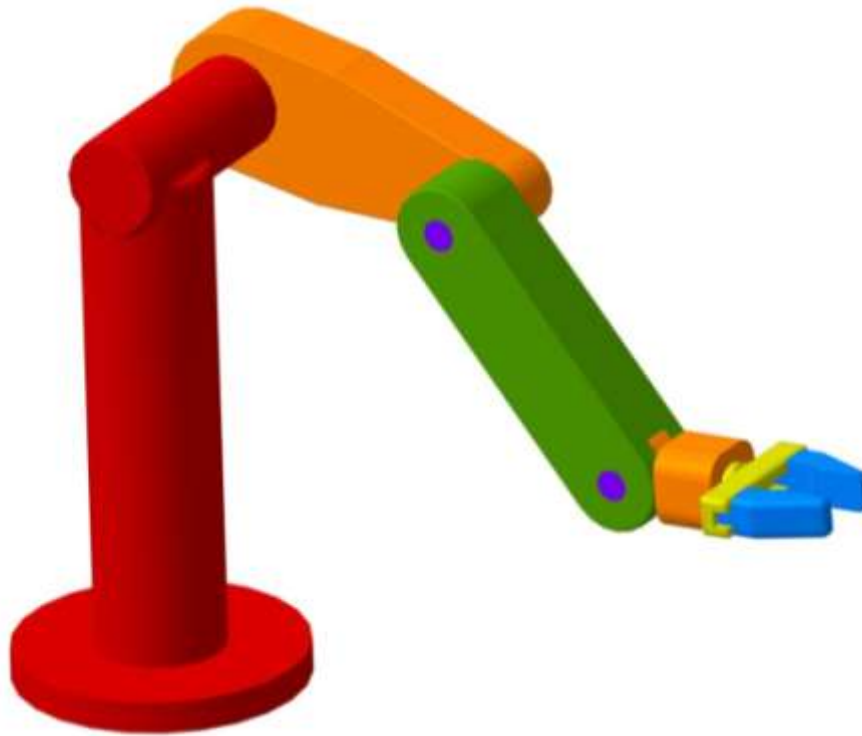


Рисунок 5.1 – 3D модель маніпулятора Пума

Таблиця 5.1 – Параметри систем координат ланок маніпулятора Пума

Зчленування i	θ_i	α_i	a_i	d_i	Межі
1	90	-90	0	0	-160 – 160
2	0	0	430	150	-225 – 45
3	90	90	-20	0	-45 – -225
4	0	-90	0	430	-110 – 170
5	0	90	0	0	-100 – 100
6	0	0	0	55	-266 – 266

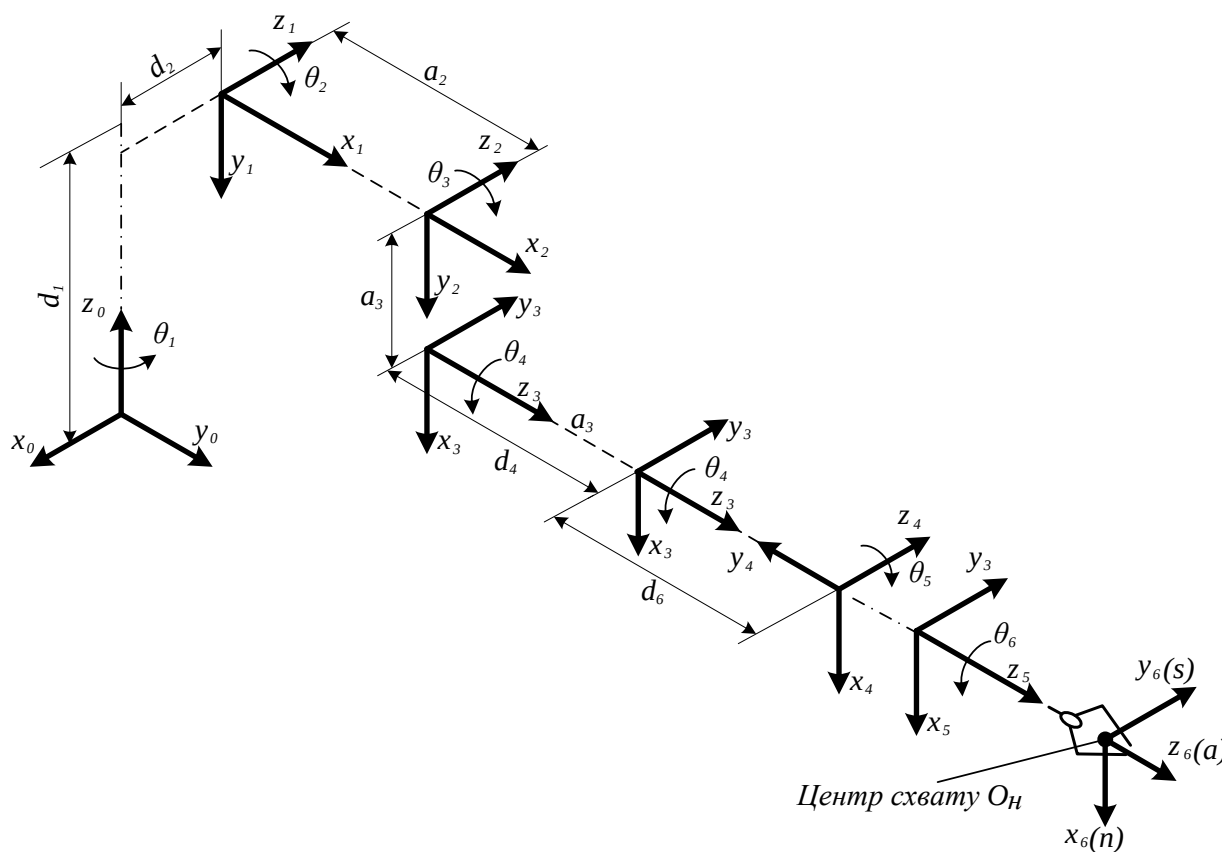


Рисунок 5.2 – Формування систем координат ланок для маніпулятора Пума

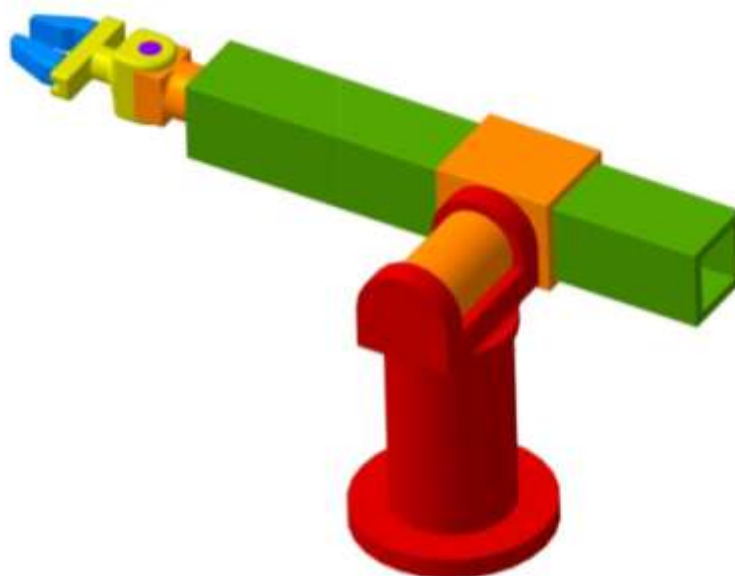


Рисунок 5.3 – 3D модель станфордського маніпулятора

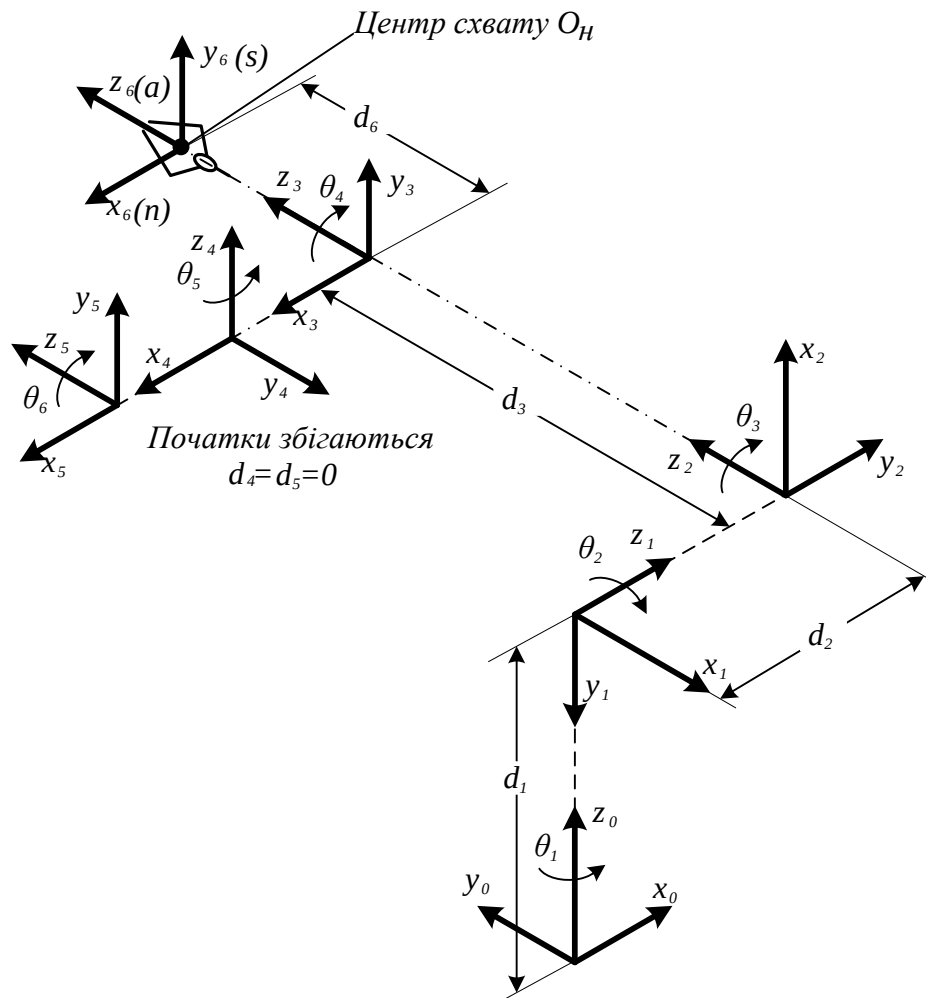


Рисунок 5.4 – Формування систем координат ланок для станфордського маніпулятора

Таблиця 5.2 – Параметри систем координат ланок станфордського маніпулятора

Зчленування i	θ_i	α_i	a_i	d_i
1	-90	-90	0	d_1
2	-90	90	0	d_2
3	-90	0	0	d_3
4	0	-90	0	0
5	0	90	0	0
6	0	0	0	d_6

Крок 1. Формування базової системи координат. Сформувати правобічну ортонормовану систему координат (x_0, y_0, z_0) , пов'язану з основою, спрямувавши вісь z_0 уздовж осі першого зчленування в напрямку до «плеча» маніпулятора. Осі x_0 та y_0 можна вибрати довільно за умови їх перпендикулярності до осі z_0 .

Крок 2. Початок. Для всіх $i (i = 1, 2, 3, \dots, n - 1)$ виконати кроки 3–6.

Крок 3. Формування осей зчленувань. Спрямувати вісь z_i уздовж осі руху (обертального або поступального) $(i+1)$ -го зчленування. Для маніпуляторів, які мають конфігурацію лівої та правої руки, осі z_1 і z_2 спрямовані від плеча і в загальному напрямку маніпулятора.

Крок 4. Формування початку i -ї системи координат. Розмістити початок i -ї системи координат на перетині осей z_i та z_{i-1} або на перетині загальної нормалі до осей z_i і z_{i-1} з віссю z_i .

Крок 5. Формування осі x_i . Вибрати одиничний вектор x_i таким чином: $x_i = \pm(z_{i-1} \times z_i) / \|z_{i-1} \times z_i\|$ або вздовж спільного перпендикуляра до осей z_{i-1} і z_i , якщо вони паралельні.

Крок 6. Формування осі y_i . Покласти $x_i = \pm(z_i \times z_{i-1}) / \|z_i \times z_{i-1}\|$, щоб утворити правобічну систему координат. (Продовжити осі z_i і x_i , якщо це потрібно для кроків 9–12.)

Крок 7. Формування системи координат захвата. Як правило, n -е зчленування є обертальним. Спрямувати вісь z_n уздовж осі руху та від робота. Вісь x_n вибрати так, щоб вона була перпендикулярна до осей z_{n-1} і z_n . Вісь y_n доповнює систему до правобічної.

Крок 8. Визначення параметрів ланок і зчленувань. Для кожного $i (i = 1, \dots, n)$ виконати кроки 9–12.

Крок 9. Визначення d_i . Це відстань від початку $(i-1)$ -ї системи координат до перетину осі z_{i-1} з віссю x_i , що визначається вздовж осі z_{i-1} . Якщо i -те зчленування поступальне, то d_i – приєднана змінна.

Крок 10. Визначення a_i . Це відстань між перетином осі z_{i-1} з віссю x_i і початком i -ї системи координат, що визначається вздовж осі x_i .

Крок 11. Визначення θ_i . Це кут, на який потрібно повернути вісь x_{i-1} навколо осі z_{i-1} , щоб вона збігалася з віссю x_i . Якщо i -те зчленування обертове, то θ_i – приєднана змінна.

Крок 12. Визначення α_i . Це кут, на який потрібно повернути вісь z_{i-1} навколо осі x_i , щоб вона збігалася з віссю z_i .

Як тільки системи координат сформовані для всіх ланок, можемо побудувати однорідні матриці перетворення, що пов'язують i -у та $(i-1)$ -у системи координат. Координати точки r_i , задані в i -й системі координат, можна перетворити в координати r_{i-1} цієї самої точки відносно $(i-1)$ -ї системи координат, виконавши послідовність таких операцій:

1. Поворот навколо осі z_{i-1} на кут θ_i , щоб вісь x_{i-1} стала співнаправленою з віссю x_i .
2. Зсув уздовж осі z_{i-1} на відстань d_i , щоб поєднати осі x_{i-1} і x_i .
3. Зсув уздовж осі x_i на відстань a_i , щоб поєднати початки координат.
4. Поворот навколо осі x_i на кут α_i , внаслідок чого досягається повний збіг систем координат.

Кожну з цих чотирьох операцій можна описати однорідною матрицею елементарного повороту-зсуву, а добуток таких матриць дає однорідну матрицю складного перетворення ${}^{i-1}A_i$, яку називають ДХ-матрицею перетворення для сусідніх систем координат з номерами i та $i-1$. Таким чином, отримуємо:

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

де α_i, a_i, d_i – константи,

θ_i – приєднана змінна, якщо це зчленування – обертальне.

Для поступального зчленування приєднаної змінної є d_i , а α_i, a_i і θ_i – константи. У цьому випадку ${}^{i-1}A_i$, набуває вигляду:

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & 0 \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

Використовуючи матрицю ${}^{i-1}A_i$, можна пов'язати однорідні координати p_i точки p , задані у i -й системі координат (точка p розташована в i -й системі координат), з однорідними координатами цієї самої точки відносно $(i-1)$ системи відліку, пов'язаної з $(i-1)$ -ю ланкою. Цей зв'язок встановлюється рівнянням:

$$p_{i-1} = {}^{i-1}A_i \cdot p_i \quad (3)$$

де $p_{i-1} = (x_{i-1}, y_{i-1}, z_{i-1}, 1)^T$ та $p_i = (x_i, y_i, z_i, 1)^T$.

5.2 Послідовність виконання практичної роботи

5.2.1 Вибрати варіант практичної роботи відповідно до таблиці 5.4.

5.2.2 Сформулювати системи координат ланок маніпулятора, дотримуючись правила свободи вибору 0-ї системи координат, за дотримання умови – вісь z_0 направлена вздовж осі першого зчленування.

5.2.3 Визначити DH-параметри маніпулятора та подати у вигляді таблиці, яка описує кожне з'єднання маніпулятора.

Для кожної ланки визначаються чотири параметри: θ_i – кут повороту навколо осі z_{i-1} (кутове зміщення); d_i – зміщення уздовж осі z_{i-1} (лінійне зміщення, постійна величина для обертальних пар); a_i – довжина ланки, відстань між осями z_{i-1} і z_i вздовж осі x_i ; α_i – кут нахилу між осями z_{i-1} і z_i (постійна величина).

5.2.4 Задати значення параметрів маніпулятора, за якими здійснюється рух для початкової та кінцевої точки у відповідній комп'ютерній програмі.

5.2.4.1 У вкладці «Пряма задача кінематики» за допомогою кнопки «Додати кінематичну пару» задаємо кількість кінематичних пар та заповнюємо їх відповідними значеннями.

5.2.4.2 За допомогою кнопки «Додати положення на графіку» отримуємо результати програмних обчислень матриць перетворень для початкового положення захоплювача маніпулятора у просторі.

5.2.4.3 Задаємо значення параметрів для кінцевого положення захоплювача маніпулятора.

5.2.4.4 За допомогою кнопки «Додати положення на графіку» отримуємо результати програмних обчислень матриць перетворень для кінцевого положення захоплювача маніпулятора у просторі.

5.2.4.5 За допомогою кнопки «Визначити траєкторію» отримуємо траєкторію руху захоплювача маніпулятора з точки **A**, з координатами x_0, y_0, z_0 , в точку **B** з координатами x_1, y_1, z_1 .

5.2.4.6 У вкладці «3D графік» будується тривимірна траєкторія руху захоплювача маніпулятора з точки **A** в точку **B**, в координатах x, y, z .

Примітка! У випадку некоректного відображення графіка необхідно змінити масштаб дисплея до значення 100 %, клацнувши правою кнопкою миші → вкладка «Параметри дисплея» → «Масштаб».

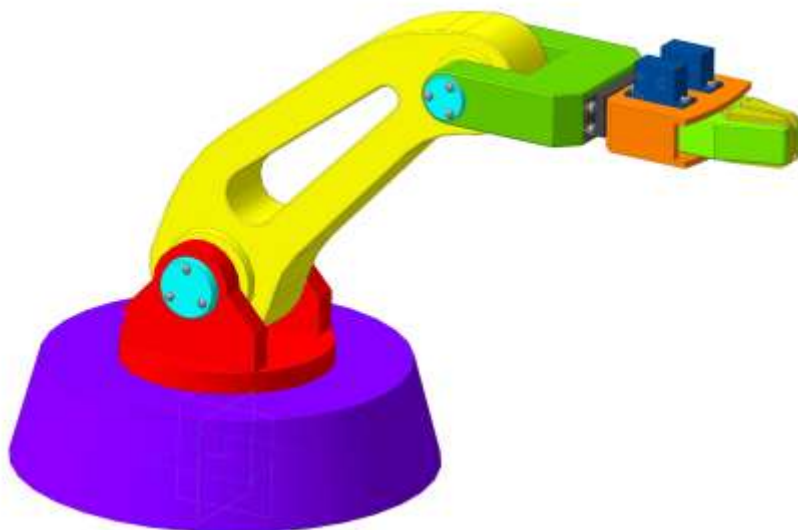
5.3 Приклад виконання практичної роботи

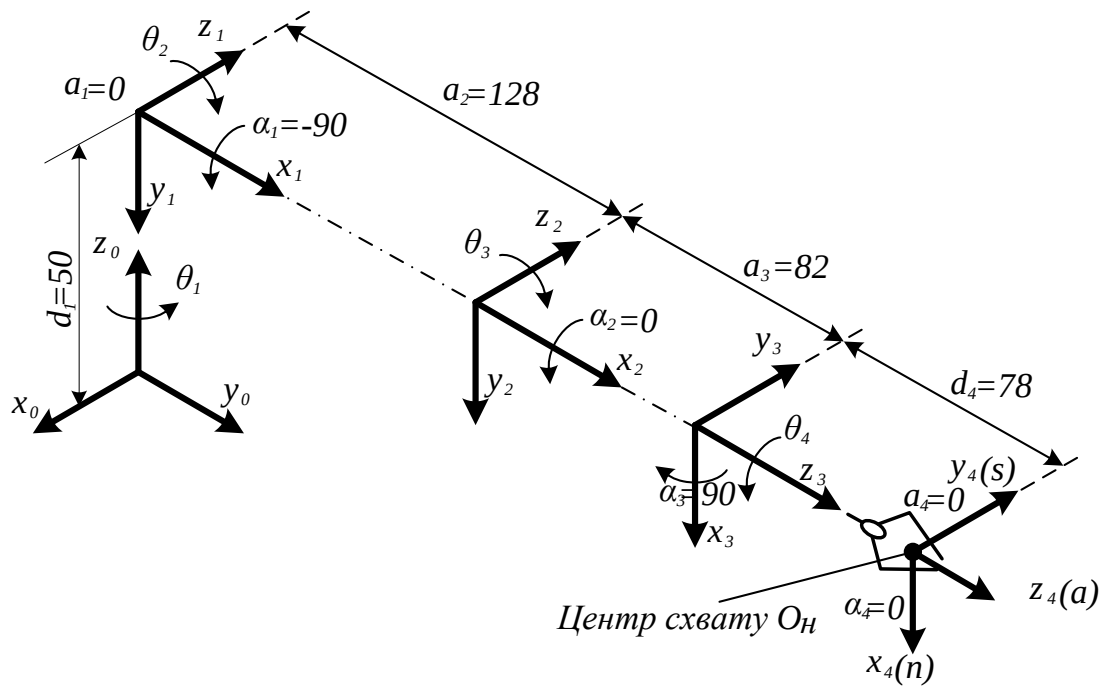
1. Вибір варіанта для номера індивідуального навчального плану здобувача №01-22-236. Дві останні цифри 36, отже, відповідно до таблиці 5.3 маємо такі параметри:

Таблиця 5.3 – Приклад вибору варіантів практичної роботи

№ інд. навч. плану			0	1	2	3	4	5	6	7	8	9
Дес.	Од.											
	θ_i	1	-90 ...160	-60 ...120	-45 ...180	30 ...180	-60 ...120	-45 ...180	-30 ...60	-160 ...45	-145 ...180	-95 ...180
		2	-60 ...60	-100 ...60	-35 ...50	- 80...50	-120 ...40	-110 ...55	60...20	0...60	-60 ...60	0...45
		3	-100 ...120	50 ...110	0...90	-15 ...110	25 ...110	-45 ...90	15...90	12...85	35 ...115	33...92
		4	0...90	30...60	15...90	-90...0	-45 ...45	-80 ...30	0...90	12...94	9...68	-15 ...20
	d_i	1	0	53	42	73	65	55	50	15	42	42
		2	30	0	0	23	0	0	0	47	0	0
		3	0	20	0	0	0	17	0	0	21	0
		4	65	76	71	50	69	0	78	71	0	71
a_i		1	0	25	0	0	0	25	0	25	0	25
		2	231	89	200	128	231	89	231	89	231	76
		3	46	65	65	82	46	65	46	65	46	49
		4	0	0	0	0	0	0	0	0	0	10
α_i		1	90	90	-90	-90	90	0	0	90	90	-90
		2	0	0	0	0	0	90	0	90	0	20
		3	-90	0	90	90	-90	0	-90	0	-90	0
		4	0	90	0	0	0	90	0	90	0	-90
			Маніпулятор Пума			4 DOF маніпулятор			Станфордський маніпулятор			

2. Формуємо системи координат ланок для чотирьохосового маніпулятора відповідно до варіанта практичної роботи.





3. Відповідно до сформованої системи координат маніпулятора визначаємо ДН-параметри маніпулятора та наводимо їх у вигляді таблиці, яка описує кожне з'єднання маніпулятора

ДН-параметри системи координат маніпулятора

Зчленування (i)	θ_i	d_i	a_i	α_i
1	θ_1	50	0	-90
2	θ_2	0	128	0
3	θ_3	0	82	90
4	θ_4	78	0	0

4. Запускаємо програму «Розрахунок прямої задачі кінематики» у вкладці «Пряма задача кінематики» задаємо значення параметрів маніпулятора, що відповідають початковій точці відповідно до отриманих ДН-параметрів, наведених у таблиці вище, та варіанта практичної роботи.

Розрахунок прямої задачі кінематики

Пряма задача кінематики 3D Графік

N°	θ (°)	d (мм)	a (мм)	Alpha (°)
1	-30	50	0	-90
2	60	0	128	0
3	15	0	82	90
4	0	78	0	0

Додати положення на графіки

Очистити

Визначити траєкторію

Зберегти розрахунки

Задаємо значення параметрів маніпулятора, що відповідають кінцевій точці.

Розрахунок прямої задачі кінематики

Пряма задача кінематики 3D Графік

N°	θ (°)	d (мм)	a (мм)	Alpha (°)
1	60	50	0	-90
2	20	0	128	0
3	90	0	82	90
4	90	78	0	0

Додати положення на графіки

Очистити

Визначити траєкторію

Зберегти розрахунки

Результати програмних обчислень матриць перетворень для початкової точки маніпулятора.

--- Ланка 1 ---

$$\theta = -30,00^\circ = -0,5236 \text{ рад}$$

$$\alpha = -90,00^\circ = -1,5708 \text{ рад}$$

Значення матриці перетворень:

[0,866 0,000 0,500 0,000]
[-0,500 0,000 0,866 0,000]
[0,000 -1,000 0,000 0,050]
[0,000 0,000 0,000 1,000]

Позиція кінематичної пари: $X = 0,00$ мм, $Y = 0,00$ мм, $Z = 50,00$ мм

--- Ланка 2 ---

$\theta = 60,00^\circ = 1,0472$ рад

$\alpha = 0,00^\circ = 0,0000$ рад

Значення матриці перетворень:

[0,500 -0,866 0,000 0,064]
[0,866 0,500 0,000 0,111]
[0,000 0,000 1,000 0,000]
[0,000 0,000 0,000 1,000]

Позиція кінематичної пари: $X = 55,43$ мм, $Y = -32,00$ мм, $Z = -60,85$ мм

--- Ланка 3 ---

$\theta = 15,00^\circ = 0,2618$ рад

$\alpha = 90,00^\circ = 1,5708$ рад

Значення матриці перетворень:

[0,966 0,000 0,259 0,079]
[0,259 0,000 -0,966 0,021]
[0,000 1,000 0,000 0,000]
[0,000 0,000 0,000 1,000]

Позиція кінематичної пари: $X = 73,81$ мм, $Y = -42,61$ мм, $Z = -140,06$ мм

--- Ланка 4 ---

$\theta = 0,00^\circ = 0,0000$ рад

$\alpha = 0,00^\circ = 0,0000$ рад

Значення матриці перетворень:

[1,000 0,000 0,000 0,000]
[0,000 1,000 0,000 0,000]
[0,000 0,000 1,000 0,078]
[0,000 0,000 0,000 1,000]

Позиція кінематичної пари: $X = 139,05$ мм, $Y = -80,28$ мм, $Z = -119,87$ мм

--- Всі позиції кінематичних пар (X, Y, Z) ---

Кінематична пара 0: $X = 0,00$ мм, $Y = 0,00$ мм, $Z = 0,00$ мм

Кінематична пара 1: $X = 0,00$ мм, $Y = 0,00$ мм, $Z = 50,00$ мм

Кінематична пара 2: $X = 55,43$ мм, $Y = -32,00$ мм, $Z = -60,85$ мм

Кінематична пара 3: $X = 73,81$ мм, $Y = -42,61$ мм, $Z = -140,06$ мм

Кінематична пара 4: $X = 139,05$ мм, $Y = -80,28$ мм, $Z = -119,87$ мм

Результати програмних обчислень матриць перетворень для кінцевої точки маніпулятора.

--- Ланка 1 ---

$\theta = 60,00^\circ = 1,0472$ рад

$\alpha = -90,00^\circ = -1,5708$ рад

Значення матриці перетворень:

[0,500 0,000 -0,866 0,000]

[0,866 0,000 0,500 0,000]

[0,000 -1,000 0,000 0,050]

[0,000 0,000 0,000 1,000]

Позиція кінематичної пари: $X = 0,00$ мм, $Y = 0,00$ мм, $Z = 50,00$ мм

--- Ланка 2 ---

$\theta = 20,00^\circ = 0,3491$ рад

$\alpha = 0,00^\circ = 0,0000$ рад

Значення матриці перетворень:

[0,940 -0,342 0,000 0,120]

[0,342 0,940 0,000 0,044]

[0,000 0,000 1,000 0,000]

[0,000 0,000 0,000 1,000]

Позиція кінематичної пари: $X = 60,14$ мм, $Y = 104,17$ мм, $Z = 6,22$ мм

--- Ланка 3 ---

$\theta = 90,00^\circ = 1,5708$ рад

$\alpha = 90,00^\circ = 1,5708$ рад

Значення матриці перетворень:

[0,000 0,000 1,000 0,000]

[1,000 0,000 0,000 0,082]

[0,000 1,000 0,000 0,000]

[0,000 0,000 0,000 1,000]

Позиція кінематичної пари: $X = 46,12$ мм, $Y = 79,88$ мм, $Z = -70,83$ мм

--- Ланка 4 ---

$\theta = 90,00^\circ = 1,5708$ рад

$\alpha = 0,00^\circ = 0,0000$ рад

Значення матриці перетворень:

[0,000 -1,000 0,000 0,000]

[1,000 0,000 0,000 0,000]

[0,000 0,000 1,000 0,078]

[0,000 0,000 0,000 1,000]

Позиція кінематичної пари: $X = 82,77$ мм, $Y = 143,35$ мм, $Z = -97,51$ мм

--- Всі позиції кінематичних пар (X, Y, Z) ---

Кінематична пара 0: $X = 0,00$ мм, $Y = 0,00$ мм, $Z = 0,00$ мм

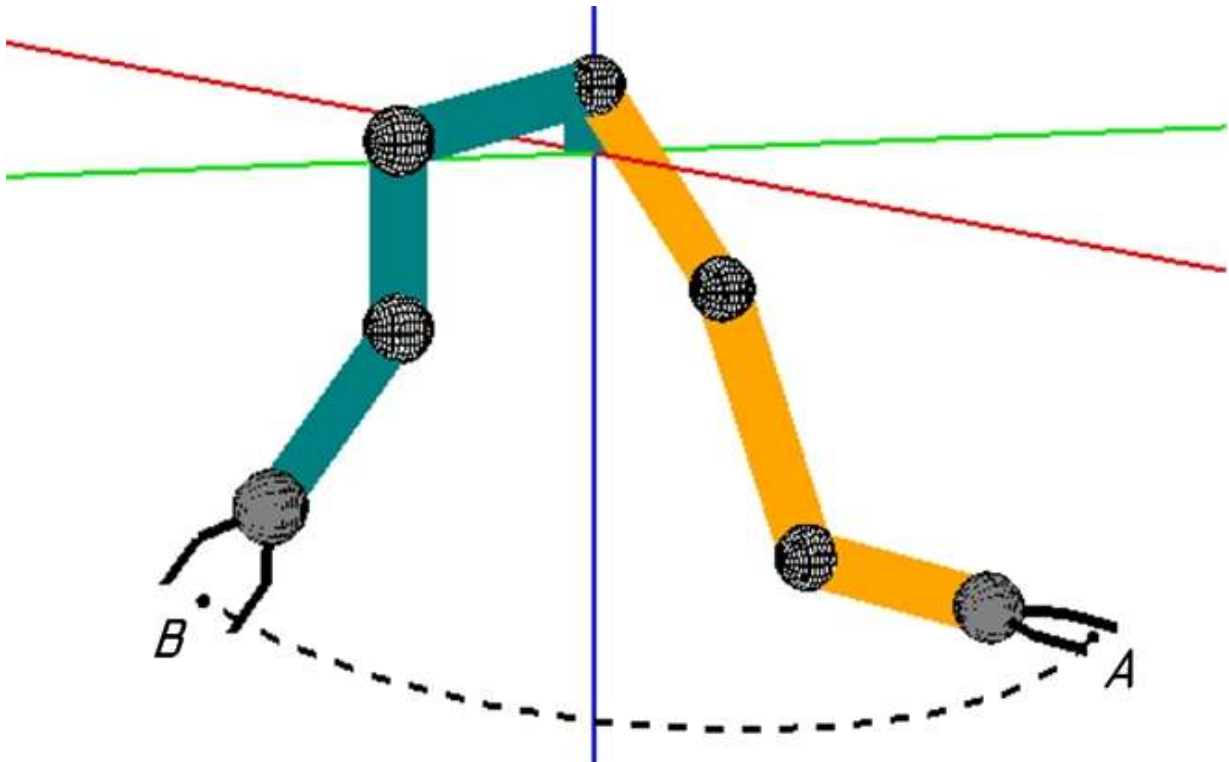
Кінематична пара 1: $X = 0,00$ мм, $Y = 0,00$ мм, $Z = 50,00$ мм

Кінематична пара 2: $X = 60,14$ мм, $Y = 104,17$ мм, $Z = 6,22$ мм

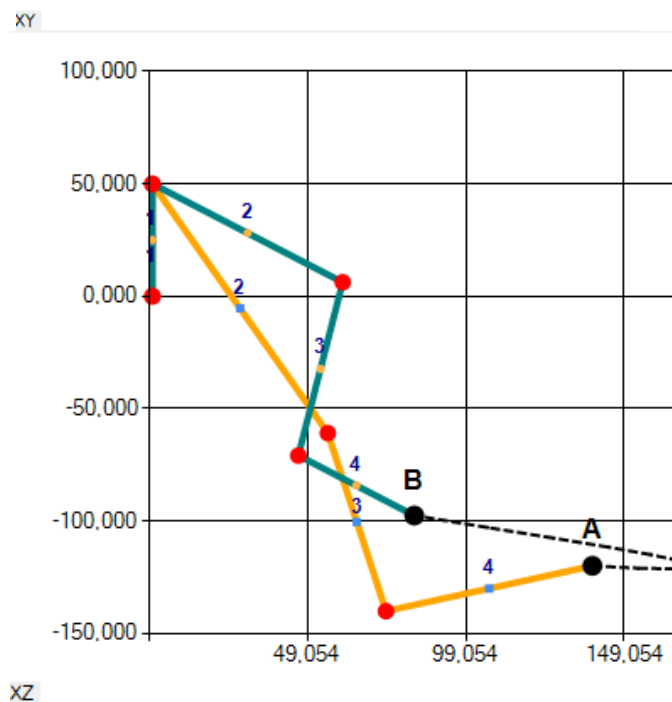
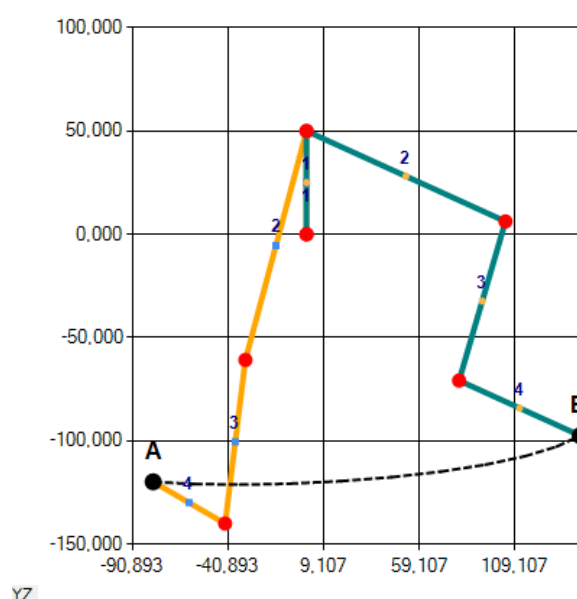
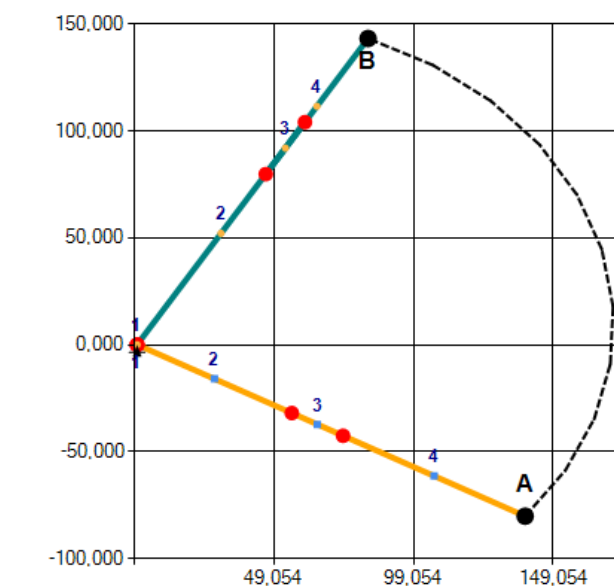
Кінематична пара 3: $X = 46,12$ мм, $Y = 79,88$ мм, $Z = -70,83$ мм

Кінематична пара 4: $X = 82,77$ мм, $Y = 143,35$ мм, $Z = -97,51$ мм

5. У вкладці «3D графік» отримуємо траєкторія руху захоплювача маніпулятора з точки A в точку B , в координатах x, y, z .



Траєкторія руху захоплювача маніпулятора з точки A в точку B , в різних площинах.



5.4 Зміст звіту

1. Звіт з практичної роботи має бути виконаний на аркушах формату A4 та відповідати варіанту практичної роботи.
2. Звіт має містити: назву практичної роботи, її мету і короткі теоретичні відомості.
3. В розділі «Результати виконання практичної роботи» здобувач має покроково описати хід виконання практичної роботи відповідно до підрозділу 5.2 цієї практичної роботи.
4. Написати висновки до практичної роботи.

Таблиця 5.4 – Варіанти завдань до практичної роботи

№ інд. навч. плану			0	1	2	3	4	5	6	7	8	9
Дес.	Од.											
	θ_i	1	-90 ...160	-60 ...120	-45 ...180	30 ...180	-60 ...120	-45 ...180	-30 ...60	-160 ...45	-145 ...180	-95 ...180
		2	-60 ...60	-100 ...60	-35 ...50	- 80...50	-120 ...40	-110 ...55	60...20	0...60	-60 ...60	0...45
		3	-100 ...120	50 ...110	0...90	-15 ...110	25 ...110	-45 ...90	15...90	12...85	35 ...115	33...92
		4	0...90	30...60	15...90	-90...0	-45 ...45	-80 ...30	0...90	12...94	9...68	-15 ...20
	d_i	1	35	53	42	73	65	55	54	75	42	62
		2	0	0	0	0	0	0	0	0	0	0
		3	0	0	0	0	0	0	0	0	0	0
		4	85	76	71	50	69	80	87	71	90	71
a_i		1	0	0	0	0	0	0	0	0	0	0
		2	231	89	200	128	231	89	231	89	231	76
		3	46	65	65	82	46	65	46	65	46	49
		4	0	0	0	0	0	0	0	0	0	0
α_i		1	90	90	-90	-90	90	90	-90	90	90	-90
		2	0	0	0	0	0	0	0	0	0	0
		3	-90	45	90	90	-90	-45	-90	45	-90	-45
		4	0	0	0	0	0	0	0	0	0	0

Контрольні запитання

1. Наведіть основні задачі кінематики під час проєктування роботів-маніпуляторів.
2. У чому полягає пряма задача кінематики?
3. У чому полягає пряма задача прискорень?
4. У чому полягає обернена задача кінематики?
5. У чому полягає обернена задача прискорень?
6. Що таке базова система координат ?
7. Що таке ортонормована Декартова система координат?
8. Назвіть три правила, на основі яких формується система координат ланки маніпулятора.
9. У чому полягає метод Денавіта–Хартенберга?
10. Що таке ДН-параметри, та в чому полягає їх суть?

ГЛОСАРІЙ

BEAM (Biology, Electronics, Aesthetics, Mechanics) – аббревіатура від слів «біологія, електроніка, естетика, механіка». Це категорія роботів, побудованих на простих аналогових електричних колах замість мікропроцесорів з метою досягнення надійності та ефективності.

LDR (Linear Distance within Raster) – параметр, що позначає час, за який робот долає лінійну відстань у межах одного елемента растра (клітинки мапи).

Nu-нейрон – «гальмівний» нейрон, який використовується для обробки сигналів від сенсорів.

Nv-нейрон – «збуджувальний» нейрон («нерв»), який використовується для керування двигунами.

Асоціативний закон – спосіб групування операндів не впливає на результат.

Ардуіно – апаратна обчислювальна платформа для аматорського конструювання, основними компонентами якої є плата мікроконтролера з елементами введення/виведення та середовище розробки Processing/Wiring мовою програмування, що є спрощеною підмножиною C/C++.

Аудіотропи – роботи, що реагують на звукові сигнали.

Булева алгебра – математична система для опису логічних операцій.

BEAM-нейрон – найпростіша форма кола затримки імпульсу, що складається з резистора, конденсатора та інвертора. Він змінює стан вихідної напруги на протилежний до вхідного із певною затримкою.

Виключне АБО (XOR) – логічна операція, істинна лише тоді, коли одна змінна істинна, а інша – хибна.

Віброти – роботи, що пересуваються за рахунок вібрації.

Вокери – роботи, що пересуваються за допомогою ніг.

Датчики внутрішньої інформації – сенсори, що визначають положення та переміщення самого робота або його частин (наприклад, енкодера, датчики кута обертання).

Датчики зовнішньої інформації – сенсори, що визначають положення робота відносно зовнішніх об'єктів (далекоміри, камери).

Двійкова система числення – система числення, що використовує цифри 0 і 1.

Джампери – роботи, що пересуваються шляхом стрибків, відштовхуючись від поверхні.

Диз'юнкція (АБО) – логічне додавання; істинне, якщо хоча б одна зі змінних істинна.

Дистрибутивний закон – закон, що визначає розподіл логічних операцій І та АБО.

Диференційний привод – система переміщення, у якій рух і поворот здійснюються за рахунок різниці швидкостей (V_L та V_R) лівого і правого коліс.

Драйвер – комп'ютерна програма, за допомогою якої операційна система отримує доступ до певного приладу чи частини апаратного забезпечення.

ДН-подання (Denavit–Hartenberg representation) – спосіб опису геометрії жорстких ланок та їх з'єднань за допомогою чотирьох параметрів ($\theta_i, d_i, a_i, \alpha_i$).

Еквівалентність – логічна операція, істинна тоді, коли обидві змінні мають однакове значення.

Закони де Моргана – правила перетворення інверсій складних логічних виразів.

Закони робототехніки BEAM – три правила, запропоновані Марком Тільденом:

- робот має захищати своє існування;
- робот має отримувати доступ до джерел енергії;
- робот має шукати ефективніші джерела енергії.

Захоплювач (End-effector) – робочий орган, закріплений на останній ланці маніпулятора. Його система координат ($x \square, y \square, z \square$) будується так, що вісь $z \square$ спрямована вздовж напрямку дії робочого органу.

Землеміри – роботи, що переміщують частину тіла вперед, тоді як інша частина залишається нерухомою.

Змінна – Іменована комірка в пам'яті, яка використовується для зберігання значень даних у програмі С. Змінні мають бути оголошені з певним типом даних.

Зона обслуговування – частина робочої зони, у межах якої повністю забезпечуються задані (паспортні) технічні характеристики.

Зона спільного обслуговування – частина простору, у якій переміщення об'єкта маніпулювання можуть виконуватися кількома промисловими роботами.

Зчленування (Joint) – з'єднання, що забезпечує рухливість ланок. Може бути обертальним (один ступінь вільності, змінна θ) або поступальним (змінна d_i).

Інверсія (HI) – логічне заперечення, результат якого є протилежним вхідному значенню.

Інвертор – електронний пристрій, у якому високий рівень вхідної напруги перетворюється на низький і навпаки.

Індуктивні датчики – використовуються в системах маршруто-слідкування. Поділяються на:

- активні, що реагують на змінний струм у дроті-маркері маршруту;
- пасивні, що реагують на магнітне поле металевої стрічки або магнітних міток.

Інтегроване середовище розробки (IDE) – комплексне програмне рішення для розробки програмного забезпечення. Зазвичай складається з редактора початкового коду, інструментів для автоматизації складання та налагодження програм.

Кисть – з'єднання у виконавчому механізмі маніпулятора (крім трьох початкових), які забезпечують орієнтаційні ступені рухливості робочого органа.

Комбінаційний цифровий пристрій – пристрій, вихід якого залежить лише від поточних вхідних сигналів.

Компаратор – операційний підсилювач без зворотного зв'язку, який порівнює дві напруги та формує вихідний сигнал залежно від того, яка з них більша.

Комутативний закон – закон, за яким перестановка операндів не змінює результату.

Кон'юнкція ($\wedge$) – логічне множення; істинне лише тоді, коли істинні обидві змінні.

Кут сервісу (Ψ) – тілесний кут, який може описати остання ланка маніпулятора за умови фіксованого положення центра захоплювача в заданій точці зони обслуговування.

Лазерний сканер (лідар) – пристрій, що дозволяє визначати положення робота (x , y , α) шляхом вимірювання відстаней та кутів до спеціальних рефлекторів або перешкод.

Ланка (*Link*) – жорстка деталь маніпулятора, що з'єднує два зчленування. Характеризується постійними параметрами довжини (a_i) та кручення (α_i).

Логічна змінна – величина, що може набувати значення 0 або 1.

Логічна функція – залежність між вхідними логічними змінними та результатом.

Логічний елемент – електронний пристрій, що реалізує певну логічну операцію.

Локальна навігація – вид навігації, що забезпечує визначення координат робота відносно деякої (зазвичай стартової) точки, що є актуальним під час переміщення у певних приміщеннях (складах, цехах). Вона використовує датчики зовнішньої інформації та засоби маршрутослідкування.

Магботи – роботи, що використовують магнітні поля для своєї роботи.

Маніпулятор – сукупність просторового важільного механізму та системи приводів, що здійснює певні дії (маніпуляції), аналогічні діям руки людини, під керуванням програмованого автоматичного пристрою або людини-оператора.

Марк Тільден – розробник концепції BEAM-роботів та винахідник механізму симуляції поведінки робота за допомогою штучних нейронів.

Маршрутослідкування (*Line Tracking*) – процес руху робота вздовж певного маршруту за допомогою спеціальних вказівників (ліній, дротів, магнітних стрічок) та датчиків, що їх розпізнають.

Метод Денавіта–Хартенберга – матричний метод опису кінематики, що полягає у послідовній побудові систем координат, пов'язаних з кожною ланкою, та використанні однорідних матриць перетворення 4×4 для опису їх взаємного розташування.

Мініболи – роботи зі сферичним тілом, що рухаються за рахунок зміщення центру мас.

Обернена задача кінематики – задача визначення необхідних відносних координат ланок (кутів повороту або лінійних переміщень) для досягнення робочим органом заданого положення у просторі.

Обернена задача швидкостей (прискорень) – розрахунок необхідних швидкостей або прискорень у зчленуваннях для забезпечення заданої швидкості руху вихідної ланки.

Одометрія – метод визначення шляху, траєкторії та швидкості переміщення за допомогою датчиків кута обертання коліс. Основним недоліком є накопичення помилок з часом.

Оптичні датчики – використовуються для відстеження лінії (наприклад, чорної смуги на білому тлі) шляхом реєстрації зміни відбитого світла.

Персональна навігація – здатність мобільного робота визначати коректне взаємне розташування себе та своїх частин відносно об'єктів навколишнього середовища для переміщення за заданою траєкторією й уникнення зіткнень. Вона базується на отриманні інформації з датчиків, що знаходяться безпосередньо на роботі.

Подія – логічний стан, який може бути істинним або хибним.

Послідовний цифровий пристрій – пристрій, вихід якого залежить від поточних сигналів і попередніх станів.

Похибка відпрацьовування траєкторії – відхилення фактичної траєкторії робочого органа від траєкторії, заданої програмою.

Похибка позиціонування – відхилення заданої позиції виконавчого механізму від фактичної під час багаторазового повторення руху.

Промисловий робот (ПР) – автоматична машина, що складається з маніпулятора та пристрою програмного керування його рухом, призначена для заміни людини під час виконання основних та допоміжних операцій у виробничих процесах.

Пропорційне регулювання – метод керування рухом (наприклад, вздовж стіни), під час якого швидкість двигунів змінюється пропорційно до різниці між поточною відстанню до об'єкта та бажаною (середньою) відстанню.

Пряма задача кінематики – задача, що полягає у визначенні положення та орієнтації робочого органу маніпулятора (та інших ланок) за заданими узагальненими координатами. Вона використовується для визначення характеристик робочої області, а також точності позиціонування маніпулятора.

Пряма задача швидкостей (прискорень) – визначення швидкості або прискорення робочого органу за відомими швидкостями (прискореннями) у кінематичних парах.

Прямокутна система координат (декартова) – система, в якій всі три початкові з'єднання (у «руці» маніпулятора) є поступальними.

Радіотропи – роботи, що реагують на радіочастоти.

Рефлектори – пасивні маяки, що встановлюються у приміщенні для роботи з лазерними сканерами. Для визначення позиції потрібно щонайменше три рефлектори.

Рівень напруги високий (1) – стан, що відповідає істині.

Рівень напруги низький (0) – стан, що відповідає хибності.

Робоча зона – простір, у якому може перебувати робочий орган (наприклад, рука) під час функціонування промислового робота або маніпулятора.

Робочий простір – простір, у якому може перебувати виконавчий орган (пристрій) промислового робота або маніпулятора.

Ролери – роботи, що рухаються перекатами.

Рука – три з'єднання у виконавчому механізмі маніпулятора, які реалізують транспортні ступені рухливості (тобто переміщення в просторі).

Свіммери – роботи, що рухаються в рідині.

Сіттери – нерухомі роботи з пасивним призначенням.

Сквірмери – роботи, що виконують обмежені рухи (наприклад, рух кінцівок).

Спрінгботи – роботи, що рухаються стрибками у певному напрямку.

Сферична кутова (ангулярна) система координат – система, в якій всі три початкові з'єднання є обертальними.

Сферична система координат – система, в якій серед початкових з'єднань два є обертальними, а одне — поступальним.

Таблиця істинності – таблиця всіх комбінацій вхідних сигналів і відповідних результатів.

Термотропи – роботи, що реагують на теплове випромінювання.

Трицикл (триколісний робот) – робот, траєкторія руху якого визначається кутом повороту ведучого колеса відносно корпусу та відстанню між ведучим і опорними колесами.

Турботи – роботи, що котяться з використанням кінцівок.

Фотоелектричні модулі – джерело живлення, яким часто оснащують BEAM-роботів для забезпечення автономності.

Фототропи – роботи, що реагують на світло.

Циліндрична система координат – система, в якій серед початкових з'єднань два є поступальними, а одне – обертальним.

Часова діаграма – графічне зображення зміни логічних сигналів у часі.

Число ступенів рухливості – сума можливих координатних рухів об'єктів маніпулювання відносно опорної системи (стояки, підставки).

Чорна лінія – траєкторія руху робота, яку він відстежує за допомогою оптичних датчиків.

ЛІТЕРАТУРА

1. Белзецький Р. С. Робоча програма навчальної дисципліни «Комп'ютерна інженерія та основи робототехніки». Вінниця : ВНТУ, 2025. 14 с.
2. Минайленко Р. М., Коноплицька-Слободенюк О. К., Гермак В. С. Комп'ютерна схемотехніка : навч. посіб. Кропивницький : Видавець Лисенко В. Ф., 2022. 153 с.
3. Сосков А. Г., Колонтаєвський Ю. П. Промислова електроніка : підручник / за ред. А. Г. Соскова. К. : Каравела, 2015. 536 с.
4. Барабан С. В., Белзецький Р. С., Арсенюк І. Р. Комп'ютерна інженерія та основи робототехніки : електронний навчальний посібник комбінованого (локального та мережного) використання. Вінниця : ВНТУ, 2024. 155 с. – URL:: <https://iq.vntu.edu.ua/repository/getfile.php/8226.pdf> (дата звернення: 12.12.2025).
5. Матвієнко М. П. Комп'ютерна логіка : підручник. Вид. 2-ге перероб. та доп. Київ : Видавництво Ліра-К, 2017. 324 с.
6. Contributors to Wikimedia projects. BEAM robotics - Wikipedia. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/BEAM_robotics (дата звернення: 12.12.2025).
7. BEAM Reference Library. BEAMbot Circuits. Solarbotics.net. BEAM News and Articles. URL: <http://solarbotics.net/library/circuits/default.htm> (дата звернення: 12.12.2025).
8. Tilden M. W., Hasslacher B. Living Machines. Los Alamos : Los Alamos National Laboratory, 1995. URL: http://solarbotics.net/library/pdflib/pdf/living_machines.pdf (дата звернення: 12.12.2025).
9. Tilden M. W., Hasslacher B. The design of “living” biomech machines: how low can one go? Los Alamos : Los Alamos National Laboratory, 1995. URL: http://downloads.solarbotics.net/pdf/living_biomech_machines.pdf (дата звернення: 18.12.2025)
10. TCRT5000 — датчик оптичний (Datasheet). URL: <http://mini-tech.com.ua/download/datasheet/sensors/tcrt5000.pdf> (дата звернення: 12.12.2025).
11. Белзецький Р. С., Чернілевський М. О., Мельник Т. С. Реалізація найпростішого BEAM робота, що здатний рухатись вздовж чорної лінії // *Матеріали LIV науково-технічної конференції підрозділів ВНТУ*, Вінниця, 24-27 березня 2025 р.

12. Трет'як А. В., Кльон А. М. Основи робототехніки : навч. посіб. для студ. спец. 133 «Галузеве машинобудування», 141 «Електроенергетика, електротехніка та електромеханіка», 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка». Полтава : Вид-во Нац. ун-ту «Полтавська політехніка імені Юрія Кондратюка», 2024. 135 с.

13. Основи робототехніки : конспект лекцій для студентів першого (бакалаврського) рівня вищої освіти денної та заочної форм навчання, спец. 141 «Електроенергетика, електротехніка та електромеханіка» / Держ. біотехнологічний ун-т; упоряд. М. С. Сорокін. Харків : [б. в.], 2024. 94 с.

URL: <https://repo.btu.kharkov.ua/handle/123456789/51001>

14. Михайлов Є. П. Мобільні роботи : навч. посіб. для студентів спец. 131 «Прикладна механіка», спец. «Мехатроніка та промислові роботи». Одеса : ОНПУ, 2016. 239 с.

15. Белзецький Р. С., Іванчук Я. В. Комп'ютерна програма «Керування рухом триколісного навчального робота по лабіринту» : свідоцтво про реєстрацію авторського права на твір (комп'ютерна програма) № 138486 від 05.08.2025. Україна, 2025.

16. Міщук Д. О., Рашківський В. П. Програмування робототехнічних інформаційних систем : конспект лекцій. Київ : КНУБА, 2024. 220 с.

17. Дорохов М. Ю. Роботи та маніпулятори : конспект лекцій Краматорськ, 2019. URL: http://www.dgma.donetsk.ua/docs/kafedry/ptm/meto_d/rm/Роботи%20та%20Маніп_Конспект.pdf (дата звернення: 12.12.2025).

18. Промислові роботи. Класифікація промислових роботів. URL: https://kempfi.in.ua/articles/promochlenie_roboti.htm (дата звернення: 12.12.2025).

19. Белзецький Р. С., Чернілевський М., Янчук К. Methods for solving the direct kinematics problem for determining the grip position of the robot-manipulator in space // *Матеріали LIII науково-технічної конференції підрозділів ВНТУ*, Вінниця, 20–22 березня 2024 р. [Електронний ресурс]. — Вінниця, 2024. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/20356> (дата звернення: 12.12.2025).

20. М. І. Вігуро, А. В. Маляр, «Розв'язання прямої задачі кінематики зварювального маніпулятора з шістьма ступенями свободи», *Електроенергетичні та електромеханічні системи*, **Випуск 3, № 2**, с. 27 — 34, 2021, <https://doi.org/10.23939/sepes2021.01.027>.

21. Ceccarelli M. *Robot Manipulators* / ed. by M. Ceccarelli. Vienna : I-Tech Education and Publishing KG, 2008. 556 p. ISBN 978-953-7619-06-0.

22. K. S. Fu, R.C. Gonzalez, C. S. G. Lee, *ROBOTICS: Control, Sensing, Vision, and Intelligence*. New York, USA: McGraw-Hill Book Company, 1987, 580 p.

23. Белзецький Р. С. Розв'язання прямої задачі кінематики робота-маніпулятора з чотирма ступенями вільності. *Вісник Вінницького політехнічного інституту*. 2025. № 5. С. 61–68.
<https://doi.org/10.31649/1997-9266-2025-182-5-61-68>

24. Белзецький Р. С. Комп'ютерна програма «Розрахунок прямої задачі кінематики робота-маніпулятора з чотирма ступенями вільності» : свідоцтво про реєстрацію авторського права на твір (комп'ютерна програма) № 1385868 від 05.01.2026. Україна, 2026.

Електронне навчальне видання

**Руслан Станіславович Белзецький
Андрій Володимирович Козловський
Юрій Миколайович Паночишин**

ПРАКТИКУМ З КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ОСНОВ РОБОТОТЕХНІКИ

Практикум

Рукопис оформив *Р. Белзецький*

Редактор *Т. Старічек*

Оригінал-макет виготовила *Т. Старічек*

Підписано до видання 21.09.2026 р.

Гарнітура Times New Roman.

Зам. № P2026-121.

Видавець та виготовлювач

Вінницький національний технічний університет,

Редакційно-видавничий відділ.

ВНТУ, ГНК, к. 114.

Хмельницьке шосе, 95,

м. Вінниця, 21021.

press.vntu.edu.ua;

Email: rvv.vntu@gmail.com

Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.