



# ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Автоматики і комп'ютеризованих технологій  
Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки  
Рівень вищої освіти перший (бакалаврський)  
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології  
Тип програми освітньо-професійна  
Освітня програма Автоматизація та комп'ютерно-інтегровані технології  
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР \_\_\_\_\_  
(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студентові Радченка Семену Володимировичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка розподіленої системи автоматизації для управління мобільною платформою

затверджена наказом по університету від “ 03 ” червня 2024р. № 544 Ст.

2. Термін подання студентом роботи “ 27 ” червня 2024р.

3. Вихідні дані до роботи 3.1 Мобільна платформа, призначена для виконання різноманітних операцій з переміщення виробів;

3.2 Інформаційне моделювання;

3.3 Динамічне навколишнє середовище;

3.4 Поновлення карти середовища;

3.5 Модульність розподіленої системи;

3.6 Оформлення текстової документації згідно ДСТУ 3008 – 2015.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

4.1 Вступ;

4.2 Аналіз побудови узагальненої архітектури мобільної платформи;

4.3 Створення алгоритму управління переміщенням мобільної платформи на основі кінцево-автоматного підходу;

4.4 Функціонування та програмна реалізація розподіленої системи автоматизації для управління мобільної платформи;

4.5 Охорона праці;

4.6 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Демонстраційний матеріал представлений у форматі презентації PowerPoint (\*.ppt) – 12 с. формату А4

---

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Аналіз побудови узагальненої архітектури мобільної платформи</i>	13.05 – 20.05.24	виконано
2	<i>Створення алгоритму управління переміщенням мобільної платформи на основі кінцево-автоматного підходу</i>	21.05 – 29.05.24	виконано
3	<i>Функціонування та програмна реалізація розподіленої системи автоматизації для управління мобільної платформи</i>	30.05 – 09.06.24	виконано
4	<i>Охорона праці</i>	10.06 – 12.06.24	виконано
5	<i>Оформлення пояснювальної записки</i>	13.06 – 15.06.24	виконано
6	<i>Подання роботи на перевірку Інтернет-системою StrikePlagiarism</i>	16.06 – 19.06.24	виконано
7	<i>Подання роботи на рецензію</i>	20.06 – 22.06.24	виконано
8	<i>Подання роботи на підпис зав. кафедри</i>	23.06 – 25.06.24	виконано
9	<i>Подання кваліфікаційної роботи в ЕК</i>	26.06.24	виконано

Дата видачі завдання 13.05.2024р.

Студент \_\_\_\_\_ Радченко С. В.  
(підпис)

Керівник роботи \_\_\_\_\_ доц. Аллахверанов Р. Ю.  
(підпис) (посада, прізвище, ініціали)

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«16» червня 2024 р.



---

Радченко С. В.

## РЕФЕРАТ

Пояснювальна записка: 76 с., 1 табл., 17 рис., 2 дод., 12 джерел.

МОБІЛЬНА ПЛАТФОРМА, РОЗПОДІЛЕНА СИСТЕМА  
АВТОМАТИЗАЦІЇ, УПРАВЛІННЯ, НАВІГАЦІЙНА СИСТЕМА,  
МАШИННИЙ ЗІР, РОБОТИЗОВАНЕ ВИРОБНИЦТВО.

Об'єкт розробки – управління переміщенням всеспрямованої мобільної платформи.

Предмет розробки – програмне забезпечення.

Мета роботи – розроблення програмного забезпечення для розподіленої системи автоматизації управління всеспрямованої мобільної платформи у роботизованому виробництві.

Виконано моделювання функціонування мобільної платформи в роботизованому виробництві та організація тактико-технічного планування інформаційно-рухових дій.

Наведено узагальнена архітектура мобільної платформи і загальний принцип побудови його моделі. Розглянуто алгоритми планування рухових дій. Представлені інформаційні моделі багатокomпонентної мобільної платформи. Виконано вибір способу організації управління мобільної платформи. Наведено метод координації і планування управлінням мобільною платформою у складі роботизованого виробництва.

Проведено заходи і розрахунки для забезпечення безпечних умов праці в лабораторії де виконувалася кваліфікаційна робота.

## ABSTRACT

Explanatory note: 76 pp., 1 tab., 17 figs., 2 appendices, 12 sources.

MOBILE PLATFORM, DISTRIBUTED AUTOMATION SYSTEM, CONTROL, NAVIGATION SYSTEM, MACHINE VISION, ROBOTIC MANUFACTURING.

The object of development is the management of the movement of an omnidirectional mobile platform.

The subject of development is software.

The purpose of the work is to develop software for a distributed automation system for the control of an omnidirectional mobile platform in robotic production.

The simulation of the functioning of the mobile platform in robotic production and the organization of tactical and technical planning of informational and motor actions were carried out.

The generalized architecture of the mobile platform and the general principle of building its model are given. Algorithms for planning motor actions are considered. Information models of a multi-component mobile platform are presented. The choice of the method of organizing the management of the mobile platform has been made. The method of coordination and planning of the management of a mobile platform as part of robotic production is given.

Measures and calculations were carried out to ensure safe working conditions in the laboratory where qualification work was performed.

## ЗМІСТ

Перелік скорочень .....	9
Вступ... ..	10
1 Аналіз побудови узагальненої архітектури мобільної платформи .....	12
1.1 Узагальнена архітектура мобільної платформи .....	12
1.2 Загальний принцип побудови моделі.....	19
1.3 Алгоритми планування .....	24
2 Створення алгоритму управління переміщенням мобільної платформи на основі кінцево-автоматного підходу .....	40
2.1 Передумови створення алгоритму .....	40
2.2 Багатокомпонентна система .....	41
2.3 Вибір способу організації управління переміщенням мобільної платформи .....	43
2.4 Архітектура підсистеми багатокомпонентної робототехнічної системи .....	45
3 Функціонування та програмна реалізація розподіленої системи автоматизації для управління мобільної платформи .....	47
3.1 Координація виконання інформаційно-рухових дій .....	47
3.2 Планування виконання інформаційно-рухових дій .....	52
3.3 Функціонування системи підтримки і прийняття рішень .....	54
3.4 Рішення оберненої задачі кінематики .....	56
3.5 Програмна реалізація системи підтримки і прийняття рішень .....	60
4 Охорона праці .....	69
4.1 Аналіз умов праці в лабораторії .....	69
4.2 Промислова безпека в лабораторії .....	69
4.3 Виробнича санітарія і гігієна праці .....	70
4.4 Пожежна безпека лабораторії .....	72

Висновки .....	74
Перелік джерел посилання .....	75
Додаток А Фрагмент програми .....	77
Додаток Б Демонстраційний матеріал .....	78

## ПЕРЕЛІК СКОРОЧЕНЬ

АСК – абсолютна система координат;

БД – база даних;

БРС – багатокомпонентна робототехнічна система;

ЕОМ – електронно-обчислювальна машина;

ІРД – інформаційно-рухова дія;

КМП – коридор мінімального потенціалу;

МЗ – машинний зір;

МП – мобільна платформа;

ПК – персональний комп'ютер;

ПР – промисловий робот;

ТП – технологічний процес;

ЦСУ – центральна система управління.

## ВСТУП

Мобільні платформи (МП) у значенні об'єкта сприймаються як багатоканальні нелінійні динамічні системи. Хоча на сьогодні виконано цілу низку досліджень у галузі управління мобільними платформами, втім універсальні підходи щодо синтезу систем автоматичного управління мобільними платформами розроблені не на всі 100%.

Як правило, МП, перш за все, призначені для заміни людини, а точніше з метою полегшення її праці, чи роботи в небезпечних для людини зонах. З огляду на попередні визначення, створення систем управління, що не потребують втручання людини чи вони розраховані на мінімальну взаємодію в разі аварійних ситуацій – актуальне наукове завдання.

Об'єкт розробки – управління переміщенням всеспрямованої мобільної платформи.

Предмет розробки – програмне забезпечення.

Мета роботи – розроблення програмного забезпечення для розподіленої системи автоматизації управління всеспрямованої мобільної платформи у роботизованому виробництві.

У ході роботи слід розв'язати завдання щодо:

- аналізу методів управління МП;
- вивчення складової частини МП;
- огляду програмних середовищ;
- визначення елементів управління навігацією МП;
- створення алгоритму переміщення;
- побудови програми управління переміщенням МП;
- реалізації інформаційного моделювання функціонування МП;
- вивчення особливостей управління МП на базі кінцево-автоматного підходу;
- аналізу методу координації і планування дій.

Пояснювальну записку з кваліфікаційної роботи оформлено згідно з ДСТУ 3008:2015 [1], а також з рекомендаціями з підготовки і оформлення кваліфікаційної роботи здобувачами першого (бакалаврського) рівня вищої освіти [2-3].

# 1 АНАЛІЗ ПОБУДОВИ УЗАГАЛЬНЕНОЇ АРХІТЕКТУРИ МОБІЛЬНОЇ ПЛАТФОРМИ

## 1.1 Узагальнена архітектура мобільної платформи

Під час проведення аналізу побудови узагальненої архітектури МП розглянемо дефініцію інтелектуального агента, тому що конструкцію компонентів архітектури агентів (датчиків, виконавчих механізмів та процесорів) уже визначено раніше, отже, слід розробити програми агента. Втім успіхи зі створення реальних роботів також залежатимуть від грамотного проектування датчиків і виконавчих механізмів, необхідних для реалізації поставленої задачі [5].

Інтерфейс між роботами та середовищем, в якому вони безпосередньо діють, відповідно, забезпечують передачу результатів сприйняття, називають датчиком. Наприклад, пасивні датчики, на кшталт відеовідеокамери, в прямому значенні здійснюють функції спостерігача за середовищем: перехоплюють сигнали, що утворюються іншими джерелами сигналів у середовищі. Активні датчики, як от локатори, надсилають енергію до середовища, в результаті чого частина випромінюваної енергії відображається і, відповідно, надходить до датчика. Зауважимо, що активні датчики виробляють більше інформації, за пасивні, втім від джерела живлення спостерігається збільшення споживання енергії; крім того, недолік викликано й появою інтерференції, тобто в результаті одночасного використання багаточисельних активних датчиків. Фактично датчики активності та пасивності можна розподілити на три типи: перші реєструють відстань до об'єктів, другі створюють зображення середовища, а треті виконують контроль характеристик власне робота.

Більшість з мобільних платформ устаткована далекомірами (датчиками, що вимірюють відстань до найближчих об'єктів). Серед таких типів датчиків найчастіше затребуваними є звукові локатори, їх ще називають ультразвуковими вимірювальними перетворювачами, котрі випромінюють спрямовані звукові хвилі, що відображаються від об'єктів, в результаті чого частина цього звуку знову повертається до датчика. Звертаємо увагу, що інтенсивність та час надходження такого зворотного сигналу містять інформацію про відстань до найближчих об'єктів. У свою чергу, підводні апарати обладнані підводними гідролокаторами, а на землі найчастіше послуговуються звуковими, щоб запобігти зіткненням лише у найближчих околицях, бо для таких датчиків характерним є обмежений кутовий дозвіл. Іншими, альтернативними до звукових локаторів, пристроями є радари (діють на повітряних судах) та лазери. Лазерний далекомір продемонстровано на рисунку 1.1 [6].

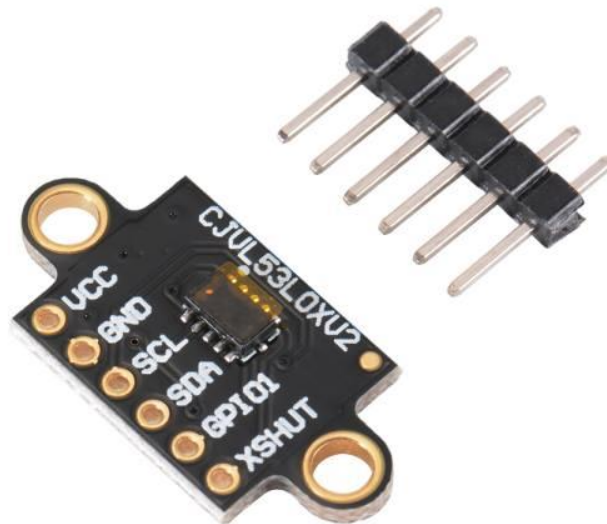


Рисунок 1.1 – Лазерний далекомір [6]

Певні датчики відстані актуальні для вимірювання як занадто короткої відстані, так і доволі довгої. Датчиками вимірювання короткої відстані назвемо тактильними, серед яких контактні вусики, контактні панелі та сенсорні покриття. На іншому кінці спектру – глобальна система позиціонування Global Positioning System (GPS), що ґрунтується на вимірюванні відстані до супутників випромінювальними імпульсними сигналами. На сьогодні на орбіті залучено понад два десятки супутників, кожен з яких передає сигнали із використанням двох різних частот. Приймачі GPS визначають відстань до цих супутників, спираючись на аналіз значень фазових зрушень, тобто шляхом тріангуляції сигналів від декількох супутників, приймачі GPS можуть обчислити свої абсолютні координати на Землі з точністю до кількох метрів. У диференційних системах GPS активізують другий наземний приймач з відомими координатами, тобто точність вимірювання координат за ідеальних умов обчислюється до міліметра. Втім системи GPS не активні всередині приміщення або під водою [6].

До другого важливого класу належать датчики зображення, тобто відеовідео-камери, з яких надходить зображення навколишнього середовища, крім того, вони спроможні моделювати та визначати характеристики середовища, залучаючи методи машинного зору (МЗ). У робототехніці особливого значення набуває стереоскопічний зір, бо розроблений для отримання інформації про глибину. Втім майбутнє цього напрямку знаходиться під загрозою, тому що розробники плідно працюють над новими активними технологіями для отримання просторових зображень.

Третій важливий клас містить пропріорецептивні датчики, котрі інформують робота про власний стан. Для вимірювання точної конфігурації робототехнічного шарніру, котрий відповідає за його дію, електродвигуни, як правило, містять дешифратори кута повороту валу. Іншими словами можна з'ясувати навіть незначні прирощення кута

повороту вала електродвигуна. Причому у маніпуляторах інформація є точною за будь-який період часу.

Крім того, дешифратори кута повороту вала мобільних роботів, які передають дані про кількість обертів колеса, можуть використовуватись в одометрії, тобто вимірювати пройдені відстані. Втім колеса часто зсуваються і прослизують, у зв'язку з цим точними результати одометра можуть бути тільки для дуже короткої відстані. До того ж, ще однією причиною помилок під час визначення позиції є зовнішні сили, наприклад, мають вплив на автономні підводні апарати течії чи вітри, котрі збивають з курсу автоматичні повітряні транспортні засоби. Покращити таку ситуацію можна шляхом устаткуванням інерційними датчиками – гіроскопами. Втім навіть їхнє застосування без інших додаткових коштів, не сприяють виключенню неминучого накопичення похибки під час визначення положення робота.

Інші ключові аспекти стану робота контролюються за допомогою датчиків зусиль, а також датчиків обертаючого моменту. Без них неможливо працювати з роботами, що призначені для роботи з крихкими об'єктами чи об'єктами з невідомим місцезнаходженням або точною формою. Наприклад, складно уявити, що робототехнічний маніпулятор з максимальним зусиллям стискання та в одну тонну закручує електричну лампочку в патрон. Утім дуже важко запобігти ситуації, в якій робот докладе занадто великі зусилля та розчавить лампочку. З огляду на це, датчики зусилля наділяють робота рисою зчутливлення, тобто можливістю визначати, як міцно він тримає лампочку, а датчики обертаючого моменту, в свою чергу, – з'ясувати, з яким зусиллям він повертає її. Крім того, датчиками можна визначати зусилля щодо трьох напрямків перенесення чи трьох напрямків обертання.

Виконавчими механізмами називають власне ті засоби, що допомагають роботу як пересуватися, так і змінювати форму свого тіла. Для того, щоб схарактеризувати особливості конструкції виконавчих

механізмів, перш за все розглянемо абстрактні поняття руху та форми, виходячи з концепції ступеня свободи. Як свобода розглядається кожний незалежний напрямок, за яким робот чи один з його виконавчих механізмів можуть рухатись. Розглянемо твердотільний робот, як автономний підводний апарат з вільними рухами з шістьма ступенями свободи, де три  $(x, y, z)$  з'ясовують положення робота в просторі, а інші три – його кутову орієнтацію за трьома осями обертання: хитанням, поворотом і нахилом. Названі ступені свободи відповідають за позу робота або кінематичний стан. Водночас динамічний стан робота для швидкості зміни кожного кінематичного виміру залучає по одному додатковому вимірюванню [5, 6].

На рисунку 1.2 наведено Стенфордський маніпулятор. Це один із перших маніпуляторів робота, котрий обладнаний п'ятьма поворотними шарнірами ( $R$ ) разом з одним призматичним зчленування ( $P$ ). У цілому зображено шість ступенів свободи.

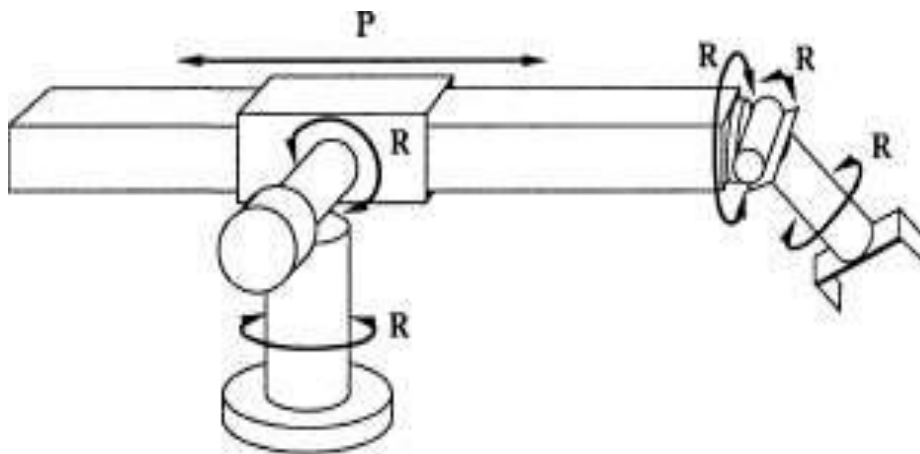


Рисунок 1.2 – Стенфордський маніпулятор

Нетвердотільні роботи застосовують додаткові ступені свободи всередині самих себе. Так, лікоть у руці людини має один ступінь свободи (згинається в одному напрямку), а кисть – три ступені (рухається вгору/вниз, з одного боку в інший, обертається). Також 1, 2 або 3 ступеня

свободи має і кожен із шарнірів робота. Для переміщення будь-якого об'єкта, наприклад, руки, до конкретної точки з конкретною орієнтацією слід володіти шістьма ступенями свободи. Щоб переконатися, що рука людини налічує більше шести ступенів свободи, проведемо елементарне дослідження: 1) покладіть кисть на стіл; 2) переконайтеся, що ви маєте ще можливість повертати руку в лікті, не змінюючи положення кисті на столі. Управляти маніпуляторами, що користуються більшою кількістю ступенів свободи, ніж потрібно для переведення кінцевого виконавчого механізму в цільове положення, легше, на відміну від роботів з мінімальною кількістю ступенів свободи.

Кількість ступенів свободи мобільної платформи не обов'язково має дорівнювати кількості залучених елементів. Так, звичний нам автомобіль може пересуватися вперед або назад, повертатися, відповідно задіяно два ступеня свободи. Втім кінематична конфігурація автомобіля – тривимірна: на відкритій плоскій поверхні можна легко змістити автомобіль до будь-якої точки  $(x, y)$ , з будь-якою орієнтацією. Отже, автомобіль налічує три ефективні ступеня свободи, але керованими є два з них. Робот, який характеризується як неголономний, володіє більшою кількістю ефективних ступенів свободи, ніж керованими, а голономним він стає, якщо ці два значення співпадають. Голономні роботи є простішими в управлінні (набагато простіше припаркувати автомобіль, який здатний рухатися не тільки вперед і назад, але і в різні боки), втім голономні роботи доволі складні механічно. Більшість серед маніпуляторів роботів є голономні, а серед мобільних платформ – неголономні [5-7].

У мобільних платформах реалізовано цілу низку механізмів для переміщення в просторі за допомогою коліс, гусениць або ніг. Роботи з диференційованим приводом обладнані незалежно керовальними колесами, що розташовані по обидва боки (або гусеницями, як в армійському танку). Якщо такі колеса обертаються з однаковою швидкістю, можна констатувати, що робот рухається по прямій. У разі

обертання коліс у протилежних напрямках, робот повертається на місці. Полярним варіантом назвемо застосування синхронного приводу, в котрому кожне колесо може обертатися та повертатися навколо вертикальної осі. Застосування такої системи приводу призвело б до хаотичного переміщення, якби не існували б обмеження, а саме: всі пари коліс повертаються в одному напрямку і обертаються з ідентичною швидкістю. Диференційний та синхронний приводи вважаються неголономними. Деякі більш вартісні роботи устатковані голономними приводами з трьома, як правило, чи більше коліс, спроможні повертатися та обертатися незалежно один від одного [5-7].

Ноги, на противагу колесам, актуальні під час пересування не плоскою поверхнею, а місцевістю з дуже грубим рельєфом. Утім на плоских поверхнях ноги, як засіб пересування, значно поступаються колесам. Крім того, завдання щодо створення механічної конструкції для них виявляється занадто складним. Науковці галузі робототехніки реалізовували спроби щодо розробки конструкцій з різною кількістю ніг: як з однією, так і кількома десятками. Було створено роботів, обладнаних ногами для ходіння, бігу і навіть стрибків.

Для мобільних платформ інших типів, спрямованих на пересування, використовують надзвичайно різноманітні інші механізми. Так, літальні апарати здебільшого устатковують пропелерами чи турбінами. Проте роботизовані дирижаблі тримаються в повітрі за допомогою теплових ефектів, а в автономних підводних транспортних засобах переважно використовують пристрої, котрі підрулюють, на кшталт тих, якими обладнані підводні човни.

Для повноцінної експлуатації робота недостатньо будувати його лише з датчиків і виконавчих механізмів. Повноцінний робот повинен містити джерело енергії для активізації своїх виконавчих механізмів. Для пересування та приведення в дію маніпулятора здебільшого залучають електродвигуни; певну область застосування охоплюють пневматичні

приводи, на базі стисненого газу, та гідравлічні з рідиною під високим тиском (рисунок 1.3).



Рисунок 1.3 – Мобільна платформа OMRON

Більшість роботів устатковані засобами цифрового зв'язку (бездротовою мережею). До того ж, корпус робота повинен бути жорстким, щоб всі ці прилади можна було б навісити на нього. Додатково мати при собі паяльник, на випадок, якщо його обладнання перестане працювати.

Підсумуючи викладене вище, можемо стверджувати, що фактично задачею для створення інтелектуальних робототехнічних систем є інформаційне моделювання МП як деякого активного агента середовища. Отже, одну із ключових задач проектування раціонального агента середовища визначимо як завдання розробки моделі планування його інформаційно-рухових дій.

## 1.2 Загальний принцип побудови моделі

З огляду на традиційну модель організації моделювання, в якій дослідник-проектувальник обмінюється інформаційними потоками з імітаційною моделлю, за результатами моделювання зворотний зв'язок

реалізується зовнішній ланцюг по відношенню до системи імітаційного моделювання – людина з додатковими допоміжними засобами та методами програмного забезпечення.

Разом з цим дослідник-проектувальник трансформує інформацію, котра складається з інтерпретації результатів і прийняття рішень щодо управління експериментами й узагальненням інформації до бази знань транспортного мобільного робота. Автоматизація управління експериментами має на меті створення замкненого програмно реалізованого контуру управління імітаційною моделлю в рамках засобів зовнішнього програмного забезпечення (рисунок 1.4).



Рисунок 1.4 – Структура інформаційно-управляючої моделі

Цілеспрямована низка експериментів відповідно до поставленої мети функціонування робота, а також врахування обмежень конфігураційних параметрів створюють модулі, що спеціально розподіляють до складу зовнішнього програмного забезпечення. У цілому дані модулі повинні задавати набори початкових даних, ініціювати прогони моделі загалом, опрацьовувати результати і приймати рішення про подальший хід експериментів згідно з реалізованим алгоритмом управління

моделюванням. Таким чином, алгоритм, спрямовуючи експерименти, в області припустимих значень параметрів виконує пошук такого об'єднання, що забезпечував би оптимальність заданого показника якості, іншими словами, власне реалізує завдання оптимізації:

$$f(\bar{x}) \xrightarrow{x \in X} \max, \quad (1.1)$$

де  $f$  – цільова функція, алгоритмічна імітаційна модель;

$\bar{x} = (x_1, x_2, \dots, x_n)$  – вектор параметрів об'єкта моделювання;

$X$  – множина припустимих значень вхідних параметрів.

Підсумуємо. Сукупність алгоритмічних та програмних засобів, яка втілює процес автоматизованого моделювання, реалізує систему автоматизації імітаційного моделювання (САІМ). До того ж, для чергового прогону імітаційної моделі користувач не вводить кожен набір початкових даних, а лише окреслює мету, критерій чи область варіювання параметрів, водночас за допомогою САІМ автоматично виконується пошук допустимих рішень задач моделювання. Отже, можемо говорити про дефініцію інтелектуальної системи моделювання. Так, функції САІМ у контурі управління гнучкої виробничої системи загалом ґрунтуються на аналізі альтернативних варіантів поведінки роботів після прийняття одного чи іншого можливого рішення щодо диспетчеризації та оперативного планування інформаційно-рухових дій.

Ефективність втілення цих функцій спрямовується на заміні жорсткої логіки управління інтелектуальним роботом. Мається на увазі використання гнучкого та динамічного механізму в певних ситуаціях заданих та фіксованих евристик. Отже, прийняття рішень здійснюється не тільки шляхом проведення аналізу щодо поточного стану робота, але й враховуються перспективи його розвитку. До того ж, існує об'єднання

високої вірогідності результатів та можливості автоматичного прийняття рішень у режимі «жорсткого» реального часу [7].

Проблема щодо автоматизації управління експериментами та синтезу знань, побудови бази знань можна подати як логічну структуру поетапного розв'язання окремих задач МП (рисунок 1.5).

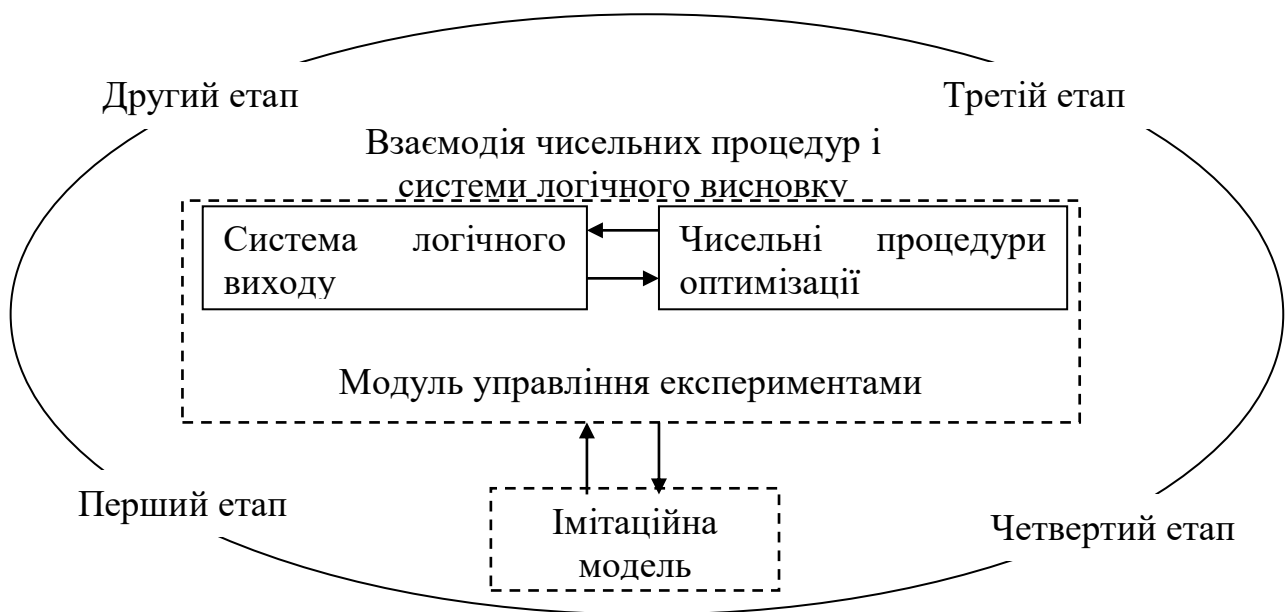


Рисунок 1.5 – Автоматизоване моделювання інтелектуального об'єкта

На початковому етапі автоматизації управління експериментами розв'язуються дві задачі: спершу, створюється структурно-алгоритмічна побудова власне модуля управління; далі окреслюються змістовні основи та формальні вимоги щодо організації інформаційного обміну з імітаційною моделлю. Задачу структурно-алгоритмічної побудови модуля управління експериментами та синтезу системи знань розв'язується шляхом з'ясування складу та структури модуля (визначаються умови взаємодії між його компонентами та положенням у загальній структурі система автоматизації імітаційного моделювання) [7].

Під час реалізації серії прогонів імітаційної моделі з'являється цілеспрямоване варіювання значення параметрів. Мова йде про можливий вплив на значення цільової функції. Він може бути прямим, спрямованим на показники функціонування об'єкта моделювання та побічним – на інші пов'язані елементи об'єкта (робота). В результаті чого змінюються складові та інші показники. Втім можна накласти обмеження на значення деяких параметрів, наприклад фіксувати їх частину – задавати декларативно. Це може виявитись під час продукування наборів вихідних даних та ініціалізації початкових станів процесу моделювання системи знань.

Якщо досліджувати імітаційну модель з точки зору засобу цілеспрямованого перетворення інформації згідно з деякою системою приписів, то доцільно розглянути алгоритм імітаційного моделювання. З огляду на це, формальну інтерпретацію розглянутих вимог подамо як:

$$\left. \begin{array}{l} S \in Q; \\ S \in C; \end{array} \right\} \quad (1.2)$$

де  $S$  – вхідне слово алгоритму імітаційного моделювання;

$Q$  – множина допустимих наборів значень параметрів робота;

$C$  – область окреслення алгоритму імітаційного моделювання.

Перше слово  $S$  задає набір початкових даних для конкретного набору даних, розтлумачимо,  $S = (i, \xi_j, \dots, \xi_z)$ , де кожна величина  $\xi_j$  збігається з деяким значенням певного параметра об'єкта моделювання. Область  $C$  можна встановити завдяки програмній реалізації алгоритму імітаційного моделювання, а утворити її – множиною  $D_{\text{вх}}$  наборів вхідного алфавіту. Всі величини  $\xi_j$ , мають допустиму реалізацію об'єкта моделювання  $U$ .

Вирази (1.2) окреслюють умови погодження  $S$  з імітаційною моделлю та алгоритмом. Постановка задачі міститься в розробленні апарату формального аналізу несуперечності змін семантично

взаємопов'язаних параметрів для забезпечення варіювання. Отже, умови (1.2) не порушуються, а зміни фіксованих параметрів не відбуваються.

Другий етап впроваджує автоматизації управління експериментами та побудови системи знань розв'язують задачі щодо:

- розроблення принципів, взаємодії системи логічного висновку та чисельних оптимізаційних процедур;
- розробляється чи обирається апарат для реалізації логічного висновку в комбінованих алгоритмах управління експериментами;
- розроблення ефективних алгоритмів висновку з урахуванням специфіки взаємодії з чисельними процедурами оптимізації (на базі запропонованого апарату).

Третій етап спрямований на розв'язання задач щодо розробки оптимізаційних процедур, які необхідні для планування екстремальних дій [6]. Власне питання розробки алгоритмічного забезпечення можна розглядати для специфічних аспектів моделювання об'єкта:

- автоматизації управління експериментами в умовах лінгвістичної невизначеності параметрів, як от «ступінь подоби/відзнаки ...» природного чисельного вимірювання не мають;
- утілення екстремальних експериментів у задачах однопараметричної оптимізації зважаючи на істотно нерівномірне розташування точок у середині інтервалу, на кшталт унімодальної функції відгуку, та фіксування відповідних станів у моделі системи знань;
- скорочення часу пошуку у разі значної тривалості прогону імітаційної моделі, що є ґрунтовними для багатоітераційних алгоритмів пошуку, повному переборі варіантів тощо.

### 1.3 Алгоритми планування

Паралельно розглядається випадок, коли робот устаткований досить точною навігаційною системою, щоб можна було знехтувати її помилками,

а системі управління відомі координати робота та вимірювального пристрою, а також орієнтація сектора огляду, і напрямок здійснюються вимірюваннями в деякій абсолютній системі координат (АСК). У всіх наведених прикладах робота є точкою з запропонованою вектором орієнтацією [7].

Детальніше розглянемо ідею методу. Уявімо, що ціль наділена деяким позитивним зарядом, а перешкоди – негативним; місця розташування цілі та перешкод є фіксованими. Припустимо, що також існує деяка негативно заряджена точка з можливістю пересування. Розмістимо її в початковій точці. Під дією сил рухома точка притягатиметься до цілі і відштовхуватиметься від перешкод. Закони руху можуть задаватися різними способами. Логічно, що у разі деяких обмежень на структуру місцевості і законів руху рухомої точки зазначена точка досягне цілі.

З огляду на спосіб завдання функцій можна створити траси з обходом перешкод з різними ступенями «ризик» (величини наближення до перешкод). Алгоритми, розглянуті нижче, забезпечують гарантію від зациклення, якщо контури перешкод опуклі. Крім того, метод буде актуальним у разі розбиття перешкод на групи, опуклі оболонки яких не перетинаються.

У кваліфікаційній роботі досліджено модифікації та напрямки модифікації вихідного алгоритму.

Розглянуто статичну та показову функції відштовхування від перешкод, а також їхній вплив на підсумковий шлях. Для оцінки ефективності шляху залучено функцію відхилення вектора напрямку руху від вектора напрямку на ціль.

Проаналізовано варіанти використання методу потенціалів для управління розподіленою мобільною платформою. Опрацьовано п'ять способів організації такого руху:

– рух за принципом «ланцюг», де сила тяжіння цілі діє на «лідера», таким чином, кожен «притягується» до попереднього;

– рух типу «гонка за лідером», де всі елементи мобільної платформи «притягуються» до «лідера», який, як правило, «притягується» до цільової точки;

– рух за принципом «розбіжності», де на розташовані компактною групою або ланцюгом мобільні платформи, починає впливати сила відштовхування від «лідера». МП «розбігаються», проте кожна досліджує свою ділянку;

– рух типу «сходження», де «лідер» збирає всі елементи РМС до компактної групи;

– організація руху за принципом «вільний пошук», де відсутня сила тяжіння до цілі, а кожен елемент МП рухається за вільним від перешкод напрямком.

Першими двома режимами послуговуються для організації пересування мобільної платформи, останніми трьома – для інформаційного дослідження середовища.

Спершу у дослідженнях інтелектуального переміщення МП перешкоди сприймалися за кола. Сила тяжіння до цілі обчислювалась як постійна за модулем і спрямована до точки цілі. Сила відштовхування від  $i$ -ї перешкоди  $f_i$  залежала від аргументу  $R_i/r_i$ , в якому  $R_i$  – радіус  $i$ -ої окружності,  $r_i$  – відстань від центру  $i$ -ої окружності до точки, що рухається.  $f_i$  визначалась спрямованою від центру кола.

Траєкторія (локомоція) окреслювалась в результаті інтегрування рівнянь руху другого порядку, бо прискорення, що діє на рухому точку, визначено сумою перелічених сил. У ході дослідження було з'ясовано, що наявна інерційність у зазначеній моделі, зумовлюється тим, що траєкторія руху стає малопрійнятною (перешкода дуже сильно «відкидає» рухому точку і траєкторія виходить надто «порізаною» [7]. Для того, щоб позбутися цього недоліку, а метод став доцільним для випадку

апроксимації контурів перешкод іншими способами, слід виконати певні процеси. Застосувати рівняння руху першого порядку (діючі сили визначають швидкість, тобто реалізовано аналог простого градієнтного спуску). Силу відштовхування визначити аргументом, який дорівнює відстані до перешкоди. До того ж, форма контуру перешкоди є довільною, а для поданого прикладу різниця існує  $r_i - R_i$ . Ця сила спрямована в бік від найближчої точки перешкоди. Крім того, у ході роботи зважаючи на наявності декількох перешкод було визнано доречним застосування функції від зазначеного аргументу  $x$  типу  $x-k$  або  $e-cx$ , які з відстанню швидко зменшуються. Зауважимо, що змінними параметрами можуть бути коефіцієнти  $k$  і  $c$ .

Підкреслимо, що, варіюючи параметри  $k$  і  $c$  під час обчислення сил відштовхування, можна визначити траєкторії для руху декількох мобільних платформ. Якщо до цих дій додати запізнення, то можна реалізувати режим «слідування один за одним» [7].

Зустрічаються випадки, в яких середовище з багатьма перешкодами «добре організовано». Так, у перешкодах, котрі розбиваються на групи, опуклі оболонки не присікаються. У наведеному прикладі силу відштовхування можна вирахувати відразу для всієї групи.

Підсумуємо, що метод потенціалів дозволяє створювати «розмиті» моделі середовища та отримувати розв'язання без урахування непотрібних подробиць.

Під час опрацювання різноманітних зарубіжних робіт щодо методу потенціалів, окреслюються два цікавих напрямки.

Першим напрямком назвемо спробу відповіді на питання: чи можна ефективно задавати силове поле, щоб в принципі були відсутні стійкі точки рівноваги. У цілому відповідь на це питання, як видно, є позитивною. Так, функція потенціалу в точці  $x$ , дорівнює мінімальній довжині допустимого шляху від  $x$  до  $g$  – точці цілі, задає таке поле. Щоправда у загальному випадку врахувати цю функцію вельми непросто.

Втім, було висунуто підхід до розв'язання цієї проблеми, який хоча й мав своєрідність, все одно, у результаті виявився, навряд чи набагато простішим за спосіб, наведений вище. Спершу розглянемо «Сферичний світ». Для площини він сприймається як окружності-перешкоди, оточені колом-рамкою. У цьому світі підсумкову силу визначено не як суму сил, які активізуються від різних перешкод, а як їхній добуток. Ці два положення дозволяють попередити виникнення точок рівноваги силового поля, зафіксованого теоретично.

Далі реалізовано послідовність дифкомп'ютерорфізмів. Насамперед між «сферичним світом» та «зоряним світом», де перешкоди фіксуються багатокутниками особливого виду. Потім втілено перехід від «зоряного світу» до все більш і більш «загальних світів» [7]. Зауважимо, що дифкомп'ютерорфізми зберігають «безособовість», утворену за допомогою поетапних переходів від «сферичного світу» силових полів. Утім підкреслимо, що приклади ілюстрації схожих полів для більш загальних постановок, ніж «сферичний світ», дають можливість впевнитись, що для них дуже характерні «яружні ефекти». Отже, за умови фіксованого значення кроку в найліпшому разі допустимі ефекти типу «інформаційного коливання», а також зациклення в помилковому екстремумі. Вірогідно, саме тому автори не наголошують на побудові траєкторій руху, а тільки – зображенні ліній рівня.

Другим, започаткованим на роботах дослідників Оксфордського університету, напрямком розглянемо візуалізацію власне робота не як крапку, а відрізок або прямокутник. Це дозволяє обчислити не тільки підсумкову силу, що діє на робота, але й момент сил, тобто управляти орієнтацією.

Розглянемо деякі, не опубліковані раніше, результати дослідження алгоритмів, які ґрунтуються на методі потенціалів.

Залежність траєкторії від показника  $k$  та кроку  $s$  для функції сили відштовхування  $F$  виду  $(R/r)^k$  для випадку кола:

(a):  $k=16, s=0.05, 0.1, 0.15, 0.2, 0.25,$

(b):  $k=4, 8, 12, 16, s=0.05,$

(c):  $F$  виду  $\exp[-(l - A \cdot r/R)^2]$ ,  $A=1.5,$

(d):  $F$  та сама, що і для (a),  $k=12, s=0.05, 0.1, 0.2, 0.3.$

Критичне відношення кроку до радіуса обхідної окружності для появи інформаційного брязкоту дорівнює 0,1 для степеневі функції й експоненти 0,3.

За допомогою методів потенціалів було досліджено й вплив інформаційного брязкоту для плавного контура (коло радіуса 1) за час обходу.

Розглянемо «метод потенційного наведення». Він передбачає устаткування робота дальномірами, що вимірюють відстань до об'єктів у робочій зоні. Обчислення напрямку руху робота постійно виконується за результатами складання трьох векторів, зважаючи на відповідні вагові функції: вектор, який характеризує напрямок, де найкраще проглядається цільова точка; вектор, який характеризує напрямок руху без зіткнень з перешкодами, і вектор, відповідного напрямку пересування робота до теперішнього часу. Разом з тим система управління роботом відповідає за інтелектуальний обхід невідомих перешкод. Управління реалізується з використанням карти за алгоритмом побудови в квазіреальному часі квазіоптимальної (за відстанню) траєкторії. Залучення ультразвукового далекоміра на роботі сприяє виявленню перешкод, утім процедура планування траєкторії знову повторюватиметься для фіксування необхідних корекцій [8].

Архітектура систем навігації реального часу, котра актуальна й для обходу перешкод МП окреслює:

– види архітектури та технології відповідних датчиків (сенсорів);

- активізацію різних моделей поведінки за допомогою сенсорно-керованих алгоритмів;
- метод штучного потенційного поля, іншими словами, алгоритм реального часу, котрий розроблено спеціально для навігації мобільних платформ під час руху.

У розглянутій системі дані інфрачервоних датчиків створюють образ середовища в деяких мобільних платформ, зважаючи на котрий потім будується траєкторія руху. З огляду на це з'являються суворі обмеження МП, пов'язані зі швидкістю отримання й обробки даних, швидкістю роботи навігаційного алгоритму і часом реакції системи управління мобільних платформ на зміну умов.

У такому виконанні метод потенціалів є точкою початку відліку в полярних координатах, де сенсором, що обертається, реалізується безперервне циклічне сканування місцевості з сектором огляду  $360^\circ$  [8].

Припустимо, що кут  $\gamma$  – кутовий крок сканування,  $d_i$  – результат  $i$ -ї (відносно поточного напрямку руху МП) виміру дальності від мобільної платформи до перешкоди. З огляду на це до кожного  $d_i$ , відповідно, відноситься вектор сили  $f_i$ , який обчислюється за допомогою рівнянь для штучного потенційного поля:

$$\begin{aligned} |f_i| &= \frac{A}{|d_i|^2}, \\ \arg(f_i) &= \arg(d_i) + \pi, \end{aligned} \quad (1.3)$$

де  $A$  – фіксована константа.

Після огляду всього сектору ( $360^\circ$ ) доречно визначити нові компоненти вектора швидкості  $V_x$  та  $V_y$ :

$$\begin{aligned}
 V_y &= \sum_{i=0}^{2\pi/\varphi} \beta^{(2(2\pi/\gamma)-i)} f_i \cos(i\gamma), \\
 V_x &= \sum_{i=0}^{2\pi/\varphi} \beta^{(2(2\pi/\gamma)-i)} f_i \sin(i\gamma),
 \end{aligned}
 \tag{1.4}$$

де  $\beta$  – з фізичної точки зору, є ваговий множник, який застосовується для того, щоб на компоненти швидкості більшого впливу впливали сили, що діють з боку мобільної платформи, ніж позаду.

Стверджується, що в загальному випадку  $\beta$  є функція  $\gamma$  та  $d_i$ :

$$\beta = \beta\left(\sum_{i=0}^{2\pi/\gamma} d_i \cos(i\gamma)\right),
 \tag{1.5}$$

при цьому  $\beta$  обчислюється щоразу за умови знаходження нового вектора швидкості [8].

Дослідимо алгоритм планування руху випуклого багатокутного об'єкта в середовищі з багатокутними перешкодами. У ньому присутня евристика, котра ґрунтується на опрацюванні моментів, які дозволяють розширити алгоритм і ввести до розгляду додатковий ступінь свободи мобільної платформи – кут повороту. Також продемонстровано результати побудови траси для МП, що рухається коридором.

У роботі спиратимемось на таке трактування. Нехай робоча область простору  $W$ , в якій активовано мобільну платформу, є підмножиною  $\mathcal{R}^n$ . Нехай  $O \subset W$  – це безліч перешкод у робочій області, у зв'язку з цим вільним простором у  $W$  буде множина  $F=W \setminus O$ ; задачею побудови шляху мобільною платформою, отже, стане завдання щодо пошуку набору точок у  $F$ , які визначають траєкторію руху МП з початкової точки до цільової.

По-перше, розглядається простий алгоритм, де багатокутний об'єкт  $M$  із двома ступенями свободи рухається в робочому просторі  $W$ , в якому з'являється кінцева множина перешкод  $O$ . Поточне положення об'єкта  $M$

окреслюється вектором  $x$ , початкове положення – вектором  $x_s$ , а цільова точка – вектором  $x_g$ . Таким чином, траса будується за алгоритмом:

```

 $x \leftarrow x_s$ 
repeat
   $\Delta_{min} \leftarrow \min (\Delta(M,o)) \forall o \in O$ 
   $F_{repulse} \leftarrow \Delta_{min} \cdot 1 / |\Delta|^2$ 
   $F_{attract} \leftarrow x_g - x$ 
   $F_{res} \leftarrow F_{attract} + a \cdot F_{repulse}$ 
   $x \leftarrow x + F_{res}$ 
until (  $x = x_g$  ) or (  $|F_{res}| = 0$  ).

```

Де  $\Delta$  – вектор, який подає мінімальну відстань між  $M$  і перешкодою  $o \in O$ . Константа  $a$  управляє впливом перешкоди на  $M$  зважаючи на відстані. Під час реалізації подібної потенційної функції зіткнень з перешкодами не відбувається, втім, алгоритм може зациклюватися за умови досягнення мобільною платформою локального мінімуму в потенційному полі [9]. З метою усунення такого явища можуть застосовуватись різні методи, зокрема, «бар'єр» з точок високого потенціалу навколо точки локального мінімуму чи метод Монте-Карло.

По-друге, для об'єкта  $M$  з'являється додатковий ступінь свободи – кут повороту  $\theta$ , початкова конфігурація об'єкта в даному разі –  $x_s, \theta_s$ . Планується рух у коридорі мінімального потенціалу (КМП). Якщо він орієнтований зважаючи на те, що момент обертання мобільної платформи в потенційному полі є мінімальним, то рух реалізується таким чином, що головна вісь направлена по дотичній до КМП.

Припустимо,  $c$  – центр мас  $M$ , а  $P$  – множина векторів, які описують стан деяких контрольних точок, рівномірно розподілених по межі  $M$  відносно  $c$ . Попередній алгоритм модифікуємо так:

```

 $x \leftarrow x_s$ 
 $\theta \leftarrow \theta_s$ 

```

*repeat*

$Frepulse \leftarrow (0, 0)$

$moment \leftarrow 0$

*for each*  $p \in P$

$\Delta_{min} \leftarrow \min (\Delta(c + p, o)) \forall o \in O$

$Frepulse \leftarrow Frepulse + \Delta_{min} \cdot 1 / |\Delta|/2$

$moment \leftarrow moment + (p \times \Delta_{min}) \cdot k$

*endfor*

$Fattract \leftarrow xg - x$

$Fres \leftarrow Fattract + a \cdot Frepulse$

$x \leftarrow x + Fres$

$\theta \leftarrow \theta + b \times moment$

*until*  $(x = xg)$  *or*  $(|Fres| = 0)$ .

Константа  $b$  керує величиною повороту й обчислюється емпірично, тому що математичне розв'язання є складним і залежить від багатьох чинників. Зважаючи на те, що під час застосування алгоритму на практиці, вибір  $c$  може бути неоднозначним. У проаналізованих прикладах для триколісної мобільної платформи використовували середину осі між двома задніми колесами [8].

Розглянемо метод обходу перешкод мобільною платформою, котрий також називається методом «гістограм векторних полів» (VHF-метод). Він спричиняє виявлення перешкод та можливість обходити їх під час руху. МП, котра підпадає під управління даним алгоритмом, швидко маневрує і навіть не зупиняється незважаючи на велику кількість неупорядкованих перешкод.

VHF-метод з метою здолання перешкод залучає сітку на двовимірній декартовій площині. Кожному осередку сітки відповідає своєрідне значення, котре позначає рівень «впевненості» алгоритму в існуванні

перешкоди в цій комірці. Метод ґрунтується на дворівневій системі представлення даних:

– на першому рівні запроваджується детальний опис навколишнього середовища робота із залученням декартової сітки  $S$ ;

– на другому рівні – полярна гістограма  $H$ , яка спирається на дані, котрі містяться в  $S$ , будується навколо центру мас мобільної платформи як набір значень з  $S$ , який співвідноситься з певним фіксованим секторам шириною  $\alpha$  кожен. Кожному сектору  $k$  протиставлено величину  $hk$ , яка трактується як полярна щільність перешкод у напрямку  $k$ .

За вихідні дані алгоритму беруться сигнали управління мобільною платформою.

Припустимо,  $S^*$ , у розумінні активної області, позначає область сітки  $S$  розміром  $w_s \cdot w_s$ , вибудованої навколо мобільної платформи; її елементи – активні осередки  $c_{ij}$ . Перетворення  $S$  на  $H$  реалізується таким чином: будуються вектори перешкод, напрямок яких відповідно до точки поточного становища мобільної платформи вираховується таким шляхом:

$$\beta_{ij} = \arctan \frac{y_i - y_0}{x_i - x_0}, \quad (1.6)$$

а модуль вектора:

$$m_{ij} = (c_{ij}^*)^2 [a - b d_{ij}], \quad (1.7)$$

де  $a, b = const > 0$ ;

$d_{ij}$  – відстань між активною ланкою та мобільною платформою;

$c_{ij}^*$  – середнє значення в активній клітинці  $(i, j)$ ;

$x_0, y_0$  – поточні координати МП;

$x_i, y_i$  – координати активної ланки  $(i, j)$ .

До кожного з  $k$  секторів добирається кут з ряду  $0, \alpha, 2\alpha, \dots, 360^\circ - \alpha$  [9].  
Таким чином, між  $k$  і  $c_{ij}^*$  виникає таке відношення:

$$k = \alpha \cdot INT(\beta / \alpha). \quad (1.8)$$

Для кожного сектора  $k$   $h_k$  вираховується як:

$$h_k = \sum_{i,j} m_{ij}. \quad (1.9)$$

Підсумуємо, що кожний з активних осередків розташовується в одному з секторів. Утім, зважаючи на дискретність сітки, внаслідок такого розподілу осередків можуть з'являтися «сходинки» в секторах. Це може спричинити появу помилок під час вибору напрямку. З метою попередження спотворення результату доцільно скористатись згладженою функцією:

$$h_k^{\wedge} = \frac{h_{k-l} + 2h_{k-l+1} + \dots + lh_k + \dots + 2h_{k+l-1} + h_{k+l}}{2l+1}, \quad l \in \mathbb{N}. \quad (1.10)$$

Далі вираховується напрямок руху в полярних координатах,  $\theta_{free}$ , і відповідний йому сектор  $k_{free}$  у  $H$ . Алгоритм надає перевагу більш «прохідному» напрямку і, водночас, як можна більш наближене до поточного напрямку на ціль  $\theta_{targ}$ .

У вихідній точці швидкість руху мобільної платформи встановлюється максимальною ( $S_{max}$ ), а потім на кожному кроці обчислюється згідно з формулою:

$$S^{\wedge} = S_{max} (1 - h_c^{\wedge} / h_m), \quad (1.11)$$

де  $h_c^{\wedge}$  – згладжена полярна щільність перешкод в обраному напрямку руху,

$$h_c^{\wedge} = \min(h_c^{\wedge}, h_m);$$

$h_m$  – емпірично встановлена константа.

До того ж відношення (2.11) гарантує  $S \geq 0$  при  $h_c^{\wedge} \leq h_m$ .

Звернемося до методу побудови гладких трас руху мобільних платформ, який ґрунтується на фізичній аналогії. Ключові переваги методу полягають у стійкому рішенні робота не тільки з двійковими (перешкода або вільний простір), але і з різномірними середовищами, чия поверхня може містити нерівні коефіцієнти тертя або кути нахилу на різних ділянках.

Метод ґрунтується на фізичних принципах гідродинаміки. Уявімо, все середовище заповнено рідиною, отже, потоки рідини дозволяють дістатися з вихідної точки до цільової. У такому разі оптимальним шляхом визначимо потік, спрямований уздовж градієнта тиску, в якому досягається стаціонарний рух рідини. З іншого боку, локальний мінімум не може бути досягнутий, зважаючи на те, що у всіх точках потоку задовільняється рівняння Лапласа. Для обліку неоднорідностей середовища залучається зовнішня сила, котра враховує силу тертя, а також вплив прохідних перешкод, разом з тим розглядаються потоки в'язкої рідини. Головним рівнянням руху в'язкої нестисливої рідини – рівняння Нав'є-Стокса:

$$\begin{aligned} \rho \frac{\partial \bar{v}}{\partial t} &= \bar{f} - \bar{\nabla} p + \mu \Delta \bar{v}, \\ \bar{\nabla} \bar{v} &= 0, \end{aligned} \quad (1.12)$$

де  $\rho$  – щільність рідини;

$v$  – вектор швидкості руху рідини;

$t$  – час;

$f$  – зовнішня сила;

$p$  – тиск;

$\mu$  – коефіцієнт в'язкості рідини.

Спрощене рівняння продемонстровано нижче:

$$\begin{aligned} \mu \Delta \bar{v}(\bar{x}) &= \bar{\nabla} p(\bar{x}) - \bar{f}(\bar{x}), \\ \bar{\nabla} \bar{v} &= 0. \end{aligned} \quad (1.13)$$

Утім невідомими є вектор швидкості  $v$  разом із абсолютною координатою  $x$ .

Граничні умови:

$$\begin{cases} \bar{v}(\bar{x}_F) = 0, & \bar{x}_F \in \partial\Omega, \\ \frac{\partial p(\bar{x}_F)}{\partial \bar{n}} = 0, & \bar{x}_F \in \partial\Omega, \end{cases} \quad (1.14)$$

де  $\partial\Omega$  – кордони перешкод;

$n$  – зовнішня нормаль до кордону перешкоди.

Початкові умови:

$$\begin{cases} p(\bar{x}_S) = 1, \\ p(\bar{x}_G) = 0, \end{cases} \quad (1.15)$$

де  $x_S$  – початкова точка;

$x_G$  – цільова точка.

З метою розв'язання рівняння в двовимірному просторі методом кінцевих різниць використаємо такий спосіб:

$$\begin{cases} 2\mu(v_{x_{i+1,j}} + v_{x_{i,j+1}} + v_{x_{i-1,j}} + v_{x_{i,j-1}} - 4v_{x_{i,j}}) = h(p_{i,j+1} - p_{i,j-1}) - 2hf_{x_{i,j}}, \\ 2\mu(v_{y_{i+1,j}} + v_{y_{i,j+1}} + v_{y_{i-1,j}} + v_{y_{i,j-1}} - 4v_{y_{i,j}}) = h(p_{i+1,j} - p_{i-1,j}) - 2hf_{y_{i,j}}, \\ v_{x_{i,j+1}} - v_{x_{i,j-1}} + v_{y_{i+1,j}} + v_{y_{i-1,j}} = 0, \end{cases} \quad (1.16)$$

де

$$\begin{aligned}\bar{v} &= v_x \bar{x} + v_y \bar{y}, \\ \bar{f} &= f_x \bar{x} + f_y \bar{y}.\end{aligned}\tag{1.17}$$

Якщо кількість точок сітки  $N$ , то слід розв'язати розріджену систему з  $3N$  лінійних рівнянь.

За результатами використання зазначеного алгоритму виявлено безліч так званих «коридорів». Кожен коридор бере початок в околі стартової точки і закінчується в околі цільової. Дотримання МП по осьовій лінії коридору гарантує її безпеку.

Проаналізуємо випадок, коли зовнішня сила не дорівнює нулю. Це дає можливість зважати на неоднорідність середовища.

Повна потенційна енергія частинки в потоці вираховується як:

$$\int_S^G \mu \Delta \bar{v} \cdot \bar{T} = \int_S^G \bar{\nabla} p \cdot \bar{T} = p_G - p_S,\tag{1.18}$$

де  $S$  – початкова точка;

$G$  – цільова точка;

$T$  – дотичний до траєкторії вектор;

$p_G - p_S$  – різниця тисків у  $x_S$  і  $x_G$ .

Зважаючи на присутність сили тертя  $F$ :

$$\begin{aligned}\int_S^G \mu \Delta \bar{v} \cdot \bar{T} + \int_S^G F \cdot \bar{T} &= \int_S^G \bar{\nabla} p \cdot \bar{T} = p_G - p_S, \\ \int_S^G \mu \Delta \bar{v} \cdot \bar{T} - L \cdot F &= \int_S^G \bar{\nabla} p \cdot \bar{T} = p_G - p_S.\end{aligned}\tag{1.19}$$

Механічна робота сили тертя  $L \cdot F$  залежить від довжини траєкторії  $L$ . У разі занадто великої величини  $F$ :

$$\int_S^G |\mu \Delta \bar{v} \cdot \bar{T}| \ll L \cdot F, \quad (1.20)$$

усі траєкторії мають обмежену довжину:

$$L \leq L_{\text{lim}} = \frac{P_S - P_G}{F}. \quad (1.21)$$

Фактично, встановлення занадто великої величини  $F$  на границі перешкод еквівалентна умові  $v = 0$ . У разі залучення  $F = \text{const}$  довжина потоків може бути обмеженою, таким чином, збільшуючи величину  $F$ , можна спричинити відсіювання шляхів більшої довжини, залишаючи лише шляхи з довжинами близькими до оптимальних.

Тестування даного методу було проведено на 4-колісній мобільній платформі на полігоні  $60 \text{ м} \times 100 \text{ м}$  з перешкодами. Зовнішню силу  $f$  було задано як:

$$f = mg(\sin \theta + K_f \cos \theta), \quad (1.22)$$

де  $m$  – маса МП;

$\theta$  – кут нахилу ділянки поверхні в напрямку руху;

$K_f$  – коефіцієнт тертя між колесами та поверхнею.

Не забуваємо також за напрямком, пов'язаний з доволі складним за своєю структурою завданням потенційної функції, котра позбавлена локальних мінімумів.

## 2 СТВОРЕННЯ АЛГОРИТМУ ПЕРЕМІЩЕННЯ МОБІЛЬНОЇ ПЛАТФОРМИ НА ОСНОВІ КІНЦЕВО-АВТОМАТНОГО ПІДХОДУ

### 2.1 Передумови створення алгоритму

Розподілені системи, на кшталт робототехнічних, останнім часом дедалі більше цікавлять дослідників. Такий попит, можна сказати, спричинений тим, що системами даного класу все частіше послуговуються як промислові, так і непромислові застосунки: одночасне об'єднання функціонуючих підсистем на практиці дозволяє реалізовувати такі завдання, котрі не під силу кожній з компонент складної системи [9]. На рисунку 2.1 продемонстровано багатокомпонентну систему, що входить до структури мобільних платформ різного призначення: навантажувачі, дослідники тощо.

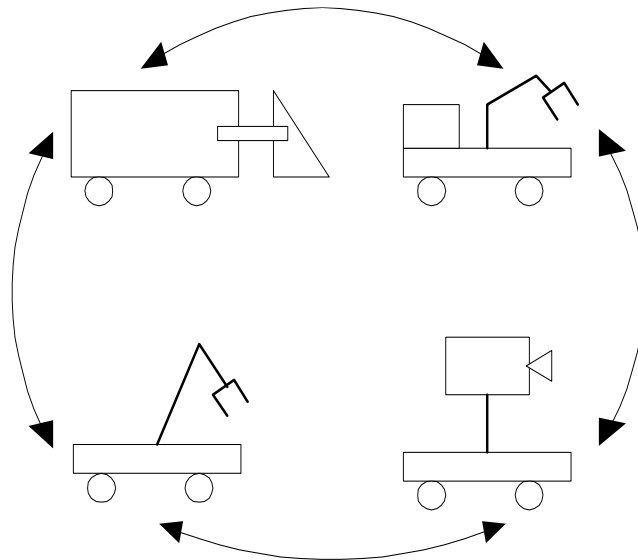


Рисунок 2.1 – Багатокомпонентні системи

Утім, управління системами такого класу є нелегким завданням: якщо управління кожної з підсистем, елементами складної системи, завдання, взагалі кажучи, розв'язане (наприклад, переведення маніпулятора з точки до точки заданою траєкторією), то управління скоординованою поведінкою групи роботів, пов'язаних спільною метою, є проблемою нелегкою.

Підкреслимо, що істотно ускладнюється завдання управління, якщо частина підсистем переслідує конфліктуючі цілі.

Пропонується підхід до розв'язання завдання координації взаємодії підсистем, які є частиною складної системи, що ґрунтується на описі всіх підсистем у значенні кінцевих автоматів і залучає мережу побудованих автоматів спеціальним чином для узгодження роботи підсистем під час виконання заздалегідь окресленого завдання. Разом з тим запропоновано способи планування поведінки складної системи зважаючи на методи штучного інтелекту.

## 2.2 Багатокомпонентна система

Багатокомпонентною системою позначаємо систему, що складається з множини підсистем, об'єднаних спільним завданням.

Розглянемо її більш детально.

Припустимо, завдання  $T$  реалізується як множина підзавдань  $t_i$ :

$$T = \{t_1, t_2, \dots, t_m\}. \quad (2.1)$$

Уведемо відношення слідування  $F$  на  $T$ , де  $t_i F t_j$  трактується таким чином, що задання  $t_i$  може реалізуватися тільки за умови втілення  $t_j$ . Якщо виконано окреслену структурування завдання  $T$ , тоді можна побудувати  $T$ -орієнтований граф:

$$G=(T,A), \quad (2.2)$$

де  $T=\{t_i\}$  – множина вершин графа  $G$ ;

$A=\{a_i\} \in T \times T$  – множина спрямованих дуг, у свою чергу, дуга  $a \in A$  з'єднує  $t_i$  з  $t_j$  тільки у разі, коли  $t_i F t_j$ .

Нехай система  $S$  формується з підсистем  $s_i$  (що позначають локальні системи чи термінали):

$$S=\{s_1, s_2, \dots, s_N\}, \quad (2.3)$$

до того ж  $T_i \subset T$  – завдання, котре здійснюється системою  $s_i$ .

Отже, систему:

$$CS=(S,T), \quad (2.4)$$

іменуватимемо  $T$  – складною системою (далі – складною системою), у разі дотримання таких умов:

$$\begin{aligned} C1: \bigcup_{i=1}^N T_i &= T, \\ C2: G. \end{aligned} \quad (2.5)$$

Умови  $C_1$ ,  $C_2$  можна трактувати як:

–  $C_1$ : складна система, що ґрунтується на наборі підсистем, яким під силу виконати завдання  $T$ ;

–  $C_2$ : дії всіх підсистем, тобто елементів складної системи, повинні узгоджуватись під час виконання завдання. Іншими словами, обмін даними (або командами) між підсистемами повинен бути скоординованим [10].

Якщо одна з підсистем  $s_i \in S$  виявляється роботом, то складну систему доцільно називати багатокомпонентною робототехнічною системою (БРС).

### 2.3 Вибір способу організації управління переміщенням мобільної платформи

Організація процедури планування та координації трактується як надзвичайно важлива, через те, що істотно впливає на залучені методи управління.

Метод централізованого управління організації вимагає присутність центральної системи управління, котра планує дії всіх підсистем, до того ж узгоджує їхню взаємодію під час виконання (рисунок 2.2) зважаючи на попередньо розроблений план.

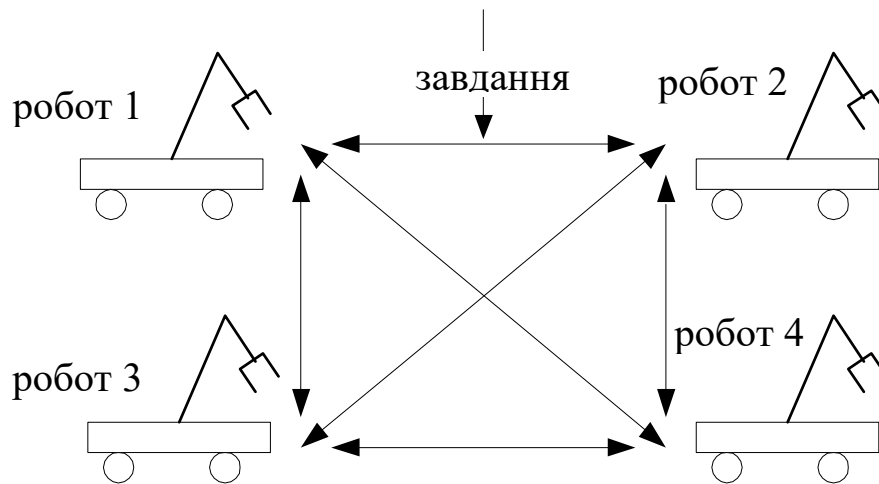


Рисунок 2.2 – Структура центральної системи управління

Центральна система управління (ЦСУ) адресує підсистемі завдання, розв'язання якого не вимагає будь-якого узгодження між підсистемами. Результат виконання повертається до ЦСУ. З огляду на отриманий результат ЦСУ направляє підсистемі наступне завдання, а потім процес повторюється. Підкреслимо, що фізично ЦСУ можна створити як окремий пристрій, або на базі системи управління однієї з компонент БРС.

За умови такого розподіленого управління відсутня центральна система – відсутня, зважаючи на це, процеси планування та узгодження завдань виконуються завдяки обміну повідомлень між підсистемами (рисунок 2.3).

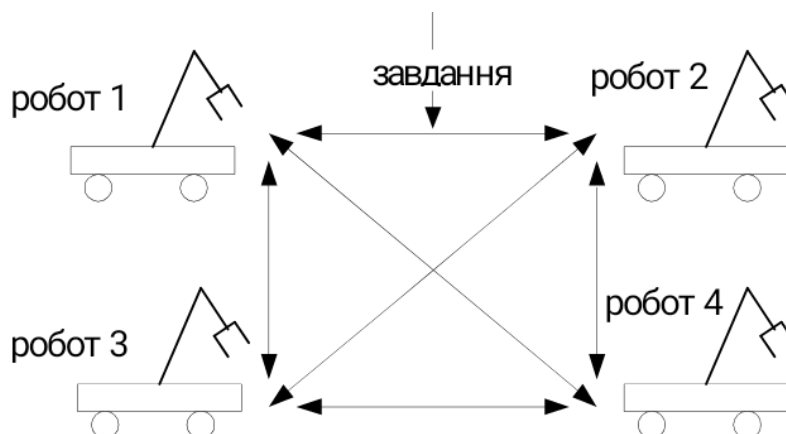


Рисунок 2.3 – Розподілене управління

На стадії планування послуговуються переговорним процесом, у наслідок якого скоординовано план розв'язання завдання. Втілення цього плану здійснюється у період реалізації і полягає у розв'язанні підзавдань та обміні отриманих даних. Зважаючи на сказане вище, не можна говорити про управління розподіленою системою, у зв'язку з тим, що, на відміну від централізованої організації, відкрито виражений відсутній носій управління, котрий забезпечує необхідну поведінку системи. Іншими словами, всі підсистеми є рівноправними як на стадії планування, так і виконання.

Таким чином, централізоване управління здійснюється на базі ієрархічного способу організації системи, котрий формулює, що управління є багаторівневим безумовним підпорядкуванням нижніх рівнів (локальних систем управління) верхнім (центральною системою управління). Всі обміни повідомленнями між підсистемами виконуються за допомогою

верхнього рівня. Розподілене управління здійснює гетерархічний спосіб організації, де всі компоненти системи є рівноправними, а процес планування реалізується всіма підсистемами водночас [10].

#### 2.4 Архітектура підсистеми багатокомпонентної робототехнічної системи

Дослідимо стисло архітектуру підсистем багатокомпонентної робототехнічної системи (БРС).

Архітектура кожної компоненти БРС окреслена таким набором функцій, котрі вона повинна реалізувати, зокрема:

–  $F_1$  – мати змогу обмінюватися даними з центральною системою управління (централізований спосіб управління) чи з локальними підсистемами (розподілений спосіб);

–  $F_2$  – гарантувати процес прийняття рішення або планування (децентралізоване управління);

–  $F_3$  – забезпечувати узгодження з іншими підсистемами на стадії реалізації спланованого завдання;

–  $F_4$  – розв'язувати відповідні підзавдання, котрі були сформовані на стадії планування.

У цілому функція  $F_1$  є суто технічною, зважаючи на це її реалізація полягає в підтримці відповідних мережевих протоколів. Для забезпечення функції  $F_2$  окреслюється вимога щодо наявності в складі підсистеми бази знань для прийняття рішень. Функція  $F_3$  вимагає присутність механізму, котрий активізує розв'язання системою відповідних підзавдань за сигналами, що надходять від інших підсистем. Утім такі сигнали не повинні потребувати перепланування завдання в цілому (як от поразка під час виконання підзавдання однією з підсистем). Реалізація функції  $F_4$  виконується на базі нижнього рівня локальної системи управління й апаратними засобами підсистем.

Підкреслимо, якщо локальна система є складною, то процес розв'язання підзавдання також може складатися з етапів планування й узгодження виконання. Так, для мобільної платформи, в якій додано шасі, маніпулятор та систему комп'ютерного зору, підзавдання типу «зрушити об'єкт  $A$  до позиції  $B$ » можна розпланувати таким чином:  $t_1$  – маніпулятор, «прибрати»;  $t_2$  – web-камера, «знайти  $A$ »;  $t_3$  – маніпулятор, «змінити схват»;  $t_4$  – маніпулятор, «взяти  $A$ »;  $t_5$  – маніпулятор, «перемістити до  $B$ »; з відповідним узгодженням між підсистемами «маніпулятор» і «web-камера» під час виконання.

Таким чином, підсистема повинна складатися принаймні з 4 таких функціонально різних компонент (рисунк 2.4): комунікатора ( $F_1$ ); планувальника ( $F_2$ ); координатора ( $F_3$ ); виконавця ( $F_4$ ).

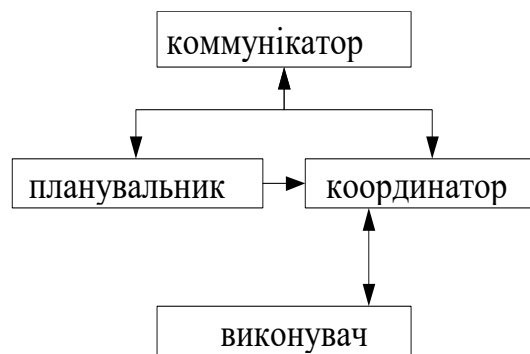


Рисунок 2.4 – Архітектура системи

### 3 ФУНКЦІОНУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗПОДІЛЕНОЇ СИСТЕМИ АВТОМАТИЗАЦІЇ ДЛЯ УПРАВЛІННЯ МОБІЛЬНОЇ ПЛАТФОРМИ

#### 3.1 Координація виконання інформаційно-рухових дій

Залучимо підхід, який реалізується в поданні цього рівня системи управління як мережі спеціальним чином побудованих кінцевих автоматів.

Нехай мережевий автомат  $NA$  з  $p$  входами і  $q$  виходами подається як:

$$NA = \langle I, O, U, X, Z, f, h \rangle, \quad (3.1)$$

де  $I = \{i_1, i_2, \dots, i_p\}$  – множина входів;

$O = \{o_1, o_2, \dots, o_q\}$  – множина виходів;

$U = \{u_1, u_2, \dots, u_m\}$  – вхідний алфавіт;

$X = \{x_1, x_2, \dots, x_n\}$  – множина станів;

$Z = \{z_1, z_2, \dots, z_k\}$  – вихідний алфавіт;

$f: X \times V \rightarrow X$  – однокрокова перехідна функція, в котрій  $V \subset U \times I$ ;

$h: X \times V \rightarrow W$  – вихідна функція, в котрій  $W \subset Z \times O$ .

Частини множин  $V$  та  $W$  трактуватимемо узагальненими вхідними та вихідними алфавітами відповідно.

Додатково введемо спеціальний символ  $\varepsilon$ , що є частиною як вхідного, так і вихідного алфавітів. Інтерпретуємо такий символ як порожній, котрий завжди наявний на виділеному вході автомата. Таким чином, якщо в описі переходу з певного стану наявний вхідний символ  $\varepsilon$ , тоді здійснюється відповідний перехід [11]. Виникнення символу  $\varepsilon$  в вихідному каналі сигналізує, що на вихід нічого не надходило. Звертаємо увагу, що отриманий у результаті автомат не буде автоматом Мілі, з огляду на те, що в ньому не зберігається довжина відображення.

Під час зображення графа мережевого автомата використаємо наступне тлумачення: через  $i.u$  позначимо символ вхідного алфавіту  $u \in U$ , котрий надійшов по вхідному каналу  $i \in I$ ; через  $z.o$  позначимо символ вихідного алфавіту  $z \in Z$ , котрий надійшов у вихідний канал  $o \in O$ .

Далі будемо трактувати поняття мережі автоматів як набір автоматів, котрі об'єднані своїми входами та виходами і взаємодіють за допомогою передачі/прийому символів своїх вихідних/вхідних алфавітів.

Нехай під дефініцією мережа автоматів будемо розуміти  $L$ -зв'язний мультиграф

$$L = (E, C), \quad (3.2)$$

де  $E = \{e_1, e_2, \dots, e_n\}$  – множина вершин графа;

$C = \{c_1, c_2, \dots, c_m\}$  – множина спрямованих дуг,  $c_i = (e_j, e_k)$ .

Вершину графів інтерпретуємо як мережевий автомат, а дугу – як канал зв'язку між автоматами, котрим послуговуються для обміну елементами вхідних/вихідних алфавітів.

Робота мережі ґрунтується на функціонуванні всіх складових її автоматів одночасно, поведінка кожного з них, вочевидь, визначається їхнім поточним станом, паралельно зі станом вхідних каналів [17].

Нехай тепер  $L = (E, C)$  – мережа автоматів, зважаючи на це розподілена система містить підсистеми, кожна з яких описується кінцевим автоматом, у зв'язку з цим  $M = \{M_i\}$  – множина моделей підсистем.

Якщо  $M \subset E$ , отримуємо:

$$L_c = \{E_c, C_c\}, \quad (3.3)$$

де  $E_c$  – структура управління для розподіленої системи, поданою моделями  $\{M_i\}$ , дорівнює  $E/M$ .

Певна річ, що система управління, котра з'єднана з дійсними підсистемами, забезпечує деяку поведінку підсистем, залежно від топології структури управління й атрибутів мережевих автоматів, які містяться у ній.

Наведемо елементарний приклад. Уявімо, що два робота  $Rb_1$  і  $Rb_2$  перекладають деталі з накопичувача  $S$  на два конвеєра  $C_1$  і  $C_2$  (рисунок 3.1).

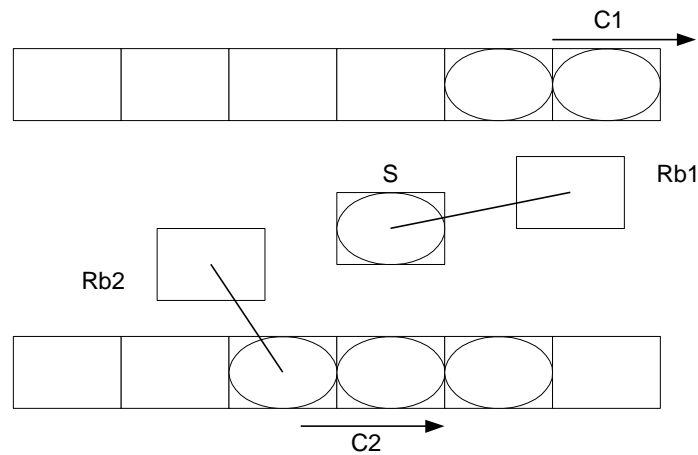


Рисунок 3.1 – Проста БРС

Зважаючи на зазначене вище, роботів  $Rb_1$ ,  $Rb_2$  окреслимо як кінцеві автомати:

$$Rb_i = (I_i, O_i, U_i, X_i, Z_i, f_i, h_i), i=1,2, \quad (3.4)$$

з наступними атрибутами:

$$I_i = \{i_0\}; O_i = \{i_0\}; U_i = \{g\}; x_i = \{0, 1, 2\}; Z_i = \{y, Y\}, i = 1,2. \quad (3.5)$$

Функції  $f_i$  і  $h_i$  продемонстровано на рисунку 3.2. Кожен з роботів  $Rb_i$  реагує лише на одну команду «g», після котрої він рухається до накопичувача  $S$ , захоплює деталь, переносить та встановлює її на конвеєр  $C_i$ . Сигнали «y» і «Y» на виході робота виникають під час звільнення ним

небезпечної зони  $i$ , в свою чергу, завершує виконання одного циклу. Представимо модель управління у вигляді трьох автоматів, серед яких два ( $Reg_1$  і  $Reg_2$ ) – логічні регулятори роботів, а третій ( $Res$ ) забезпечує стеження за ресурсом.

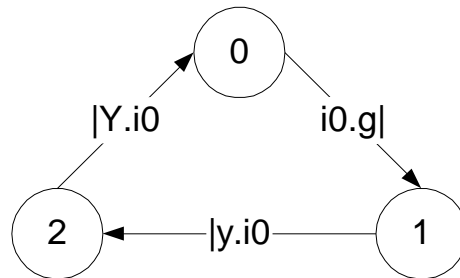
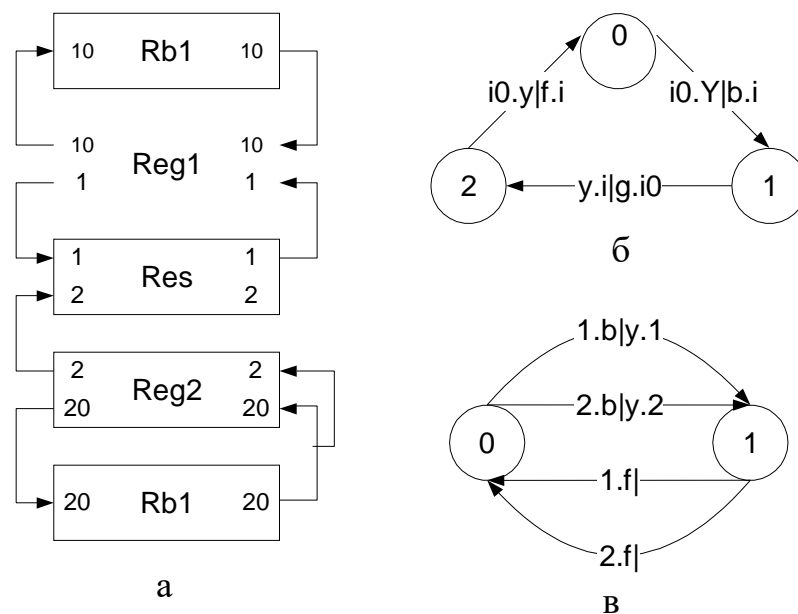


Рисунок 3.2 – Модель робота  $Rb_i$

На рисунку 3.3 продемонстровано топологію структури управління разом із діаграмами всіх вхідних до її складу автоматів.



а – топологія; б – діаграма регуляторів;  
в – діаграма автомата, котрий за ресурсом

Рисунок 3.3 – Структура управління

Структура функціонує за такою схемою: регулятор кожного робота надсилає запит на заволодіння небезпечної зони автомату-ресурсу. Якщо зона вільна, то такий запит задовольняється, а небезпечна зона переходить до стану «зайнято» і регулятор надсилає роботу команду «почати транспортну операцію». Після того, як робот залишить небезпечну зону, він інформує про це регулятор, який, зі свого боку, переводить небезпечну зону до стану «вільно» і чекає закінчення виконання операції. Потім цикл повторюється.

Розглянутий спосіб координації вважається доволі дієвим засобом для побудови тільки централізованих систем управління (рисунок 3.4).

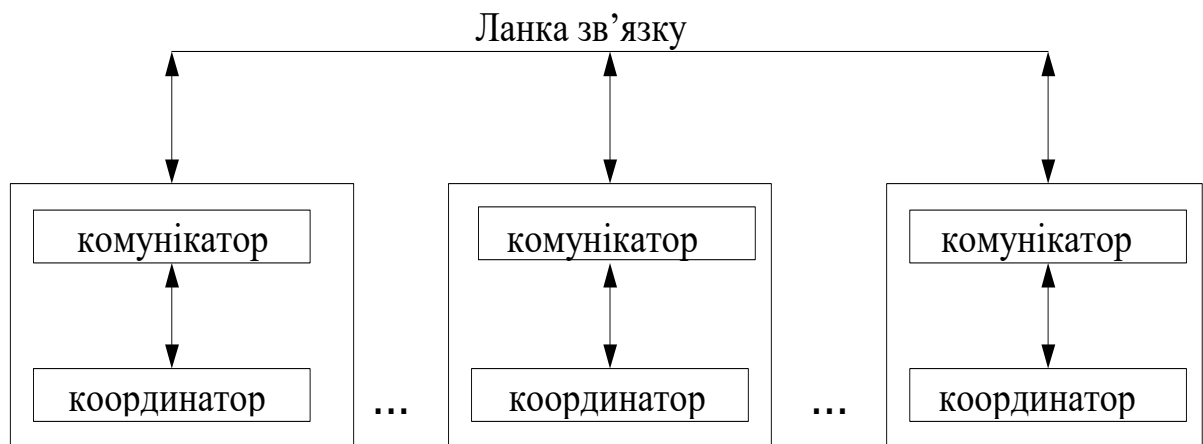


Рисунок 3.4 – Взаємодія підсистем в процесі координації

Дійсно, якщо нижній рівень структури управління є по суті справи логічним регулятором і може без роботи асоціюватися з локальними системами управління, то верхні рівні обмінюються даними з усіма підсистемами, котрі залучені для виконання завдання, таким чином, є фрагментом центральної системи управління [11].

### 3.2 Планування виконання інформаційно-рухових дій

Проаналізуємо метод планування завдання для централізованої системи, котрий полягає у реалізації двокрокової процедури.

Крок 1. Грунтуючись на методах штучного інтелекту визначити послідовність підзавдань, які реалізуються кожною з підсистем та сформовані в термінах вхідних алфавітів підсистем.

Крок 2. Перетворити отримані послідовності на структуру управління, що забезпечить узгодження виконання окресленого завдання.

Більш детально розглянемо цю процедуру. Як метод розв'язання задачі планування оберемо числення предикатів першого порядку. Вхідна інформація, котра необхідна для розв'язання завдання планування, складається з компонент, які мають такий опис:

- $K_1$  – спроможність кожною з підсистем;
- $K_2$  – початковий стан;
- $K_3$  – завдання;
- $K_4$  – універсальні властивості.

Компонент  $K_1$  містить набір пропозицій (правил), які формулюють спроможності кожною підсистемою змінювати стан зовнішнього середовища (переміщення об'єктів, отримання інформації про об'єкти тощо). Відмінна властивість цієї компоненти характеризується наявністю сколемівської функції, що належить до однієї з підсистем. Множина  $K_2$  охоплює факти, що описують початковий стан системи. Компонента  $K_3$  – формулювання завдання. Множина  $K_4$  формує універсальні особливості, що не залежать від підсистем, які є частиною БРС (ця множина може бути порожньою) [11]. Елементи всіх множин трактуються як правильно побудовані формули в численні предикатів 1-го порядку.

Завдання планування полягає в отриманні послідовності операторів, які забезпечують виконання задуму. Зважаючи на це для забезпечення

максимально можливого розпаралелювання процесу реалізації представимо множину  $K_1$  як:

$$K_1 = \bigcup_{i=1}^N K_{1i}, K_{1i} \cap K_{1i_2} = 0, i_1 \neq i_2, \quad (3.6)$$

де  $N$  – множина підсистем.

З огляду на зазначене вище, кожна підмножина  $K_{1i}$  містить тільки правильні формули, котрі окреслюють спроможності  $i$ -ої підсистеми. Після проведення такого розбиття побудуємо резолюції  $r_{ij}$ , і не-мети з елементами  $K_{1i}$  настільки довго, наскільки це можливо. Якщо даний процес неможливо продовжувати, залишаючись в  $K_{1i}$  (тобто ця стадія реалізації вимагає взаємодії між підсистемами), тоді будуємо резолюції з сусідніми підмножинами. Такий процес забезпечить (у разі успіху) отримання послідовності операторів, які наявні у кожній підсистемі, а також точок узгодження їхньої взаємодії. Отримані результати застосовують для перетворення на структуру управління, котра забезпечує виконання визначеного плану (рисунок 3.5).

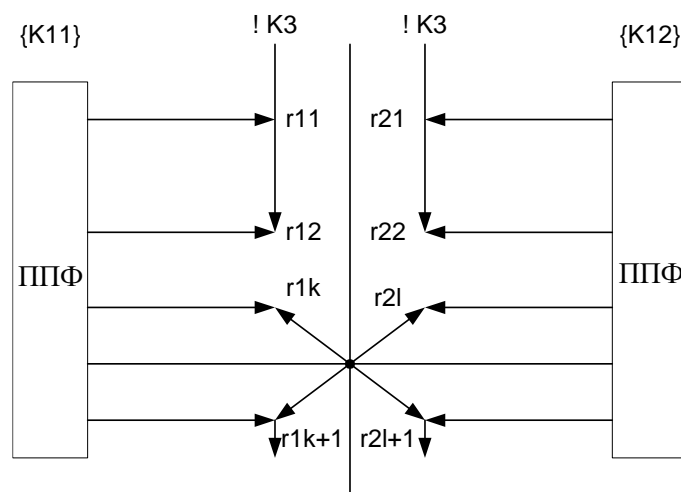


Рисунок 3.5 – Виконання визначеного плану

### 3.3 Функціонування розподіленої системи автоматизації

Розробка розподіленої системи автоматизації системи для МП – це розв'язання комплексу програмно-апаратних задач, яке повинно бути оптимальним не лише за часом виконання, а й за ресурсами, необхідними для роботи системи. Спершу, розглянемо взаємодію системи з зовнішніми по відношенню до неї функціональними елементами для окреслення можливостей, котрі можна реалізувати на їхній основі. Функціональну структуру взаємодії системи з системою машинного зору (МЗ) зображено на рисунку 3.6.

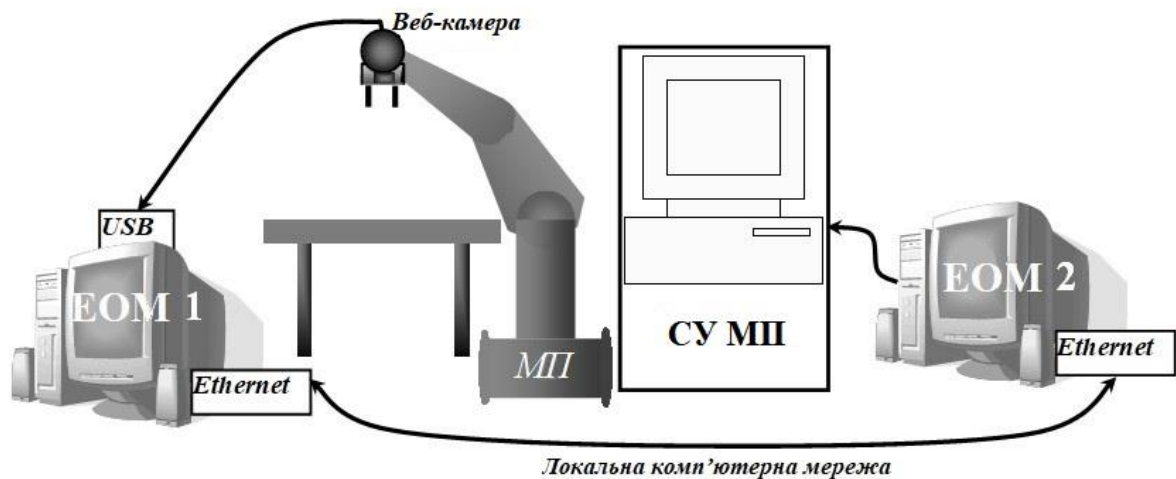


Рисунок 3.6 – Функціональна структура взаємодії системи

Під час роботи система повинна взаємодіяти з двома елементами: пультом управління МП за допомогою послідовного інтерфейсу; EOM1 (МЗ) реалізується за допомогою локальної комп'ютерної мережі (протокол UDP (User Datagram Protocol)). Програмна частина системи локалізована на EOM2 (рисунок 3.6).

На рисунку 3.7 продемонстровано структурну схему розробленої розподіленої системи автоматизації.

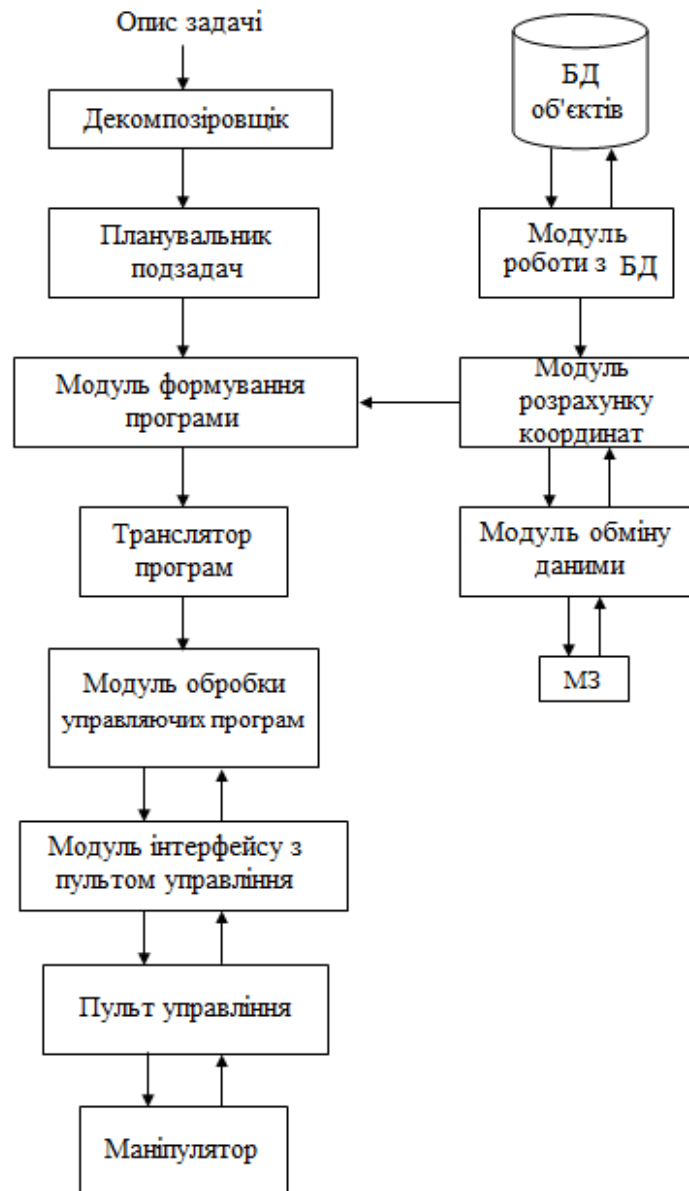


Рисунок 3.7 – Структурна схема системи

Опис задачі, котрий надходить до системи, обробляється декомпозиційником і розбивається на підзадачі. Так, задачу «перемістити об'єкт  $A$  до точки  $B$ » можна розподілити на підзадачі типу:

- відкрити схват;
- зрушити до зони об'єкта, щоб об'єкт опинився в схваті;
- закрити схват;
- переміститися до точки  $B$ ;
- відкрити схват;

– вийти з зони об'єкта.

Далі перелічені підзадачі надходять до планувальника, котрий визначає, яким чином краще їх розв'язати. Далі модуль формування програми отримує інформацію від системи МЗ про об'єкти, розташовані в робочому просторі МП. Після залучення модуля роботи з базою даних (БД) виконується завантаження параметрів, які окреслюють положення схвата в умовах захоплення кожного конкретного об'єкта. Ґрунтуючись на поточних координатах веб-камери та інформації, отриманої від системи МЗ та БД, відбувається обчислення реальних координат об'єктів. Далі на базі інформації сформованої планувальником підзадач проводиться створення програми управління з використанням отриманих координат об'єктів за результатами розрахунків.

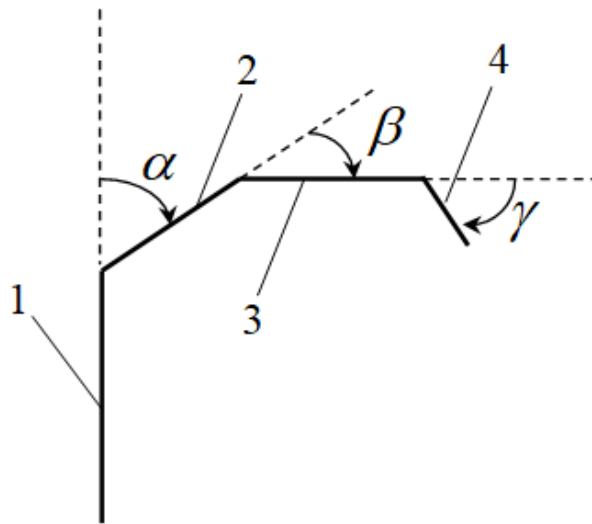
Після того, як програму створено, транслятор розв'язує зворотну задачу кінематики для кожного її кроку, формуючи, таким чином, програму в просторі приєднаних координат. Відтрансльована програма надходить до модуля відпрацювання програм управління, котрий спільно з модулем інтерфейсу забезпечує надсилання відповідних впливів на пульт управління МП.

### 3.4 Розв'язання оберненої задачі кінематики

Команди управління маніпулятором МП формуються в просторі приєднаних змінних, а координати об'єктів маніпулювання задаються в абсолютній системі координат. Для керування положенням та орієнтацією схвата роботизованої МП з метою реалізації необхідних операцій з об'єктом маніпулювання, необхідно вміти розв'язувати зворотну задачу кінематики [8]. Отже, треба вміти визначати за заданими координатами схвата та відомими параметрами його ланок і зчленувань приєднані параметри  $q = (q_1, \dots, q_6)^T$  маніпулятора, котрі забезпечують задане положення схвата.

У програмі UnitedRobotProject початково розв'язана зворотна задача кінематики для перших трьох зчленувань маніпулятора. Програма спроможна визначити за заданими декартовими координатами точки та кута  $q_5$  необхідні кути  $q_1, q_2, q_3$ . Для ефективного захоплення об'єктів та їхнього акуратного встановлення необхідно, щоб кисть у момент захоплення була повернута перпендикулярно до площини, на якій лежить об'єкт.

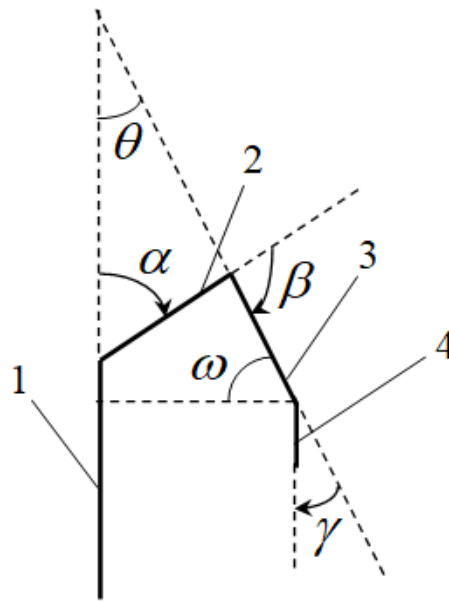
На рисунку 3.8 продемонстровано умовну кінематичну схему маніпулятора.



1 – колона; 2 – плече; 3 – лікоть; 4 – кисть

Рисунок 3.8 – Спрощена кінематична схема маніпулятора

Розшифруємо дефініції:  $\alpha$  – кут повороту плеча,  $\beta$  – кут повороту ліктя,  $\gamma$  – кут повороту кисті. Слід визначити такі кути, як  $\alpha, \beta, \gamma$ , щоб кисть була розташована перпендикулярно до робочого столу. Спершу визначимо, як кут  $\gamma$  від кутів  $\alpha$  та  $\beta$ , зважаючи на те, що кисть розташована перпендикулярно. Наведений варіант подано на рисунку 3.9.



1 – колона; 2 – плече; 3 – лікоть; 4 – кисть

Рисунок 3.9 – Кінематична схема маніпулятора у разі розташування кисті перпендикулярно робочого столу

З рисунка 3.9 окреслюється, що:

$$\angle\theta = 180 - \angle\alpha - \angle\beta, \quad (3.7)$$

$$\angle\omega = 90 - \angle\theta, \quad (3.8)$$

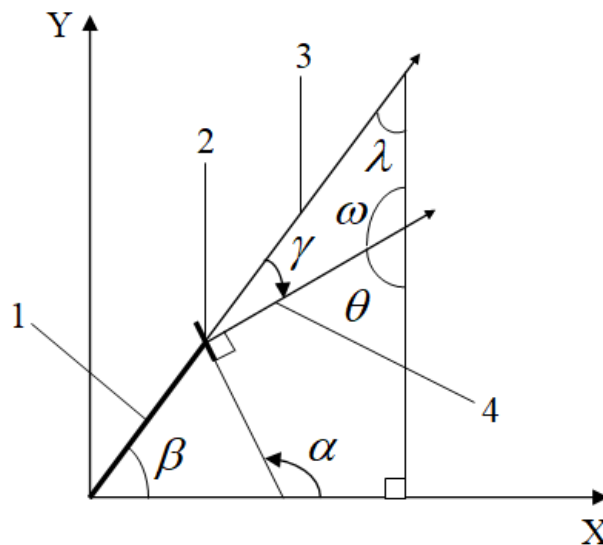
$$\angle\gamma = 90 - \angle\omega, \quad (3.9)$$

$$\angle\gamma = 90 - 90 + 180 - \angle\alpha - \angle\beta = 180 - \angle\alpha - \angle\beta. \quad (3.10)$$

Припустимо, що кисть буде розташовуватись перпендикулярно робочого столу, якщо рівність (3.10) буде дотримано. Програма UnitedRobotProject спроможна обчислювати кути  $\alpha$  та  $\beta$  за заданими координатами  $(x, y, z)$  і кута  $\gamma$ . Для отримання таких кутів, де кисть буде розташована перпендикулярно до робочого столу, необхідно змінювати

кут  $\gamma$  (від  $0^\circ$  до  $100^\circ$ ), доти, поки умову не буде виконано (3.10). Якщо на цьому відрізку точки, в якій виконано умову (3.10), не існує, то це стає підтвердженням, що або точка  $(x, y, z)$  знаходиться за межами робочої зони маніпулятора, або неможливо забезпечити перпендикулярну орієнтацію схвата для неї.

Разом з визначенням кутів повороту колони, плеча, ліктя актуальною задачею є визначення такого кута повороту схвата, котрий міг би забезпечити захоплення об'єкта, що знаходиться під певним кутом (цей кут надходить від системи МЗ). Зважаючи на те, що схват може повертатися в дуже широких межах (від  $-260^\circ$  до  $260^\circ$ ), необхідно забезпечити мінімальний кут повороту для досягнення необхідної орієнтації. На рисунку 3.10. зображено визначення кута повороту схвата.



1 – маніпулятор; 2 – об'єкт роботизації; 3 – вектор початкової орієнтації схвата; 4 – вектор необхідної орієнтації схвата

Рисунок 3.10 – Визначення кута повороту схвата

$$\gamma = 180^\circ - \angle \lambda - \angle \omega, \quad (3.11)$$

$$\angle \lambda = 90^\circ - \angle \beta, \quad (3.12)$$

$$\angle\theta = 180^\circ - \angle\alpha, \quad (3.13)$$

$$\angle\omega = 180^\circ - \angle\theta, \quad (3.14)$$

$$\angle\omega = 180^\circ - (180^\circ - \angle\alpha) = \angle\alpha, \quad (3.15)$$

$$\angle\gamma = 180^\circ - (90^\circ - \angle\beta) - \angle\alpha = 90^\circ + \angle\beta - \angle\alpha. \quad (3.16)$$

За результатами обчислень отримуємо кут  $\gamma$ , на який необхідно повернути схват МП.

### 3.5 Програмна реалізація розподіленої системи автоматизації

Управління МП здійснюється шляхом надсилання кодових посилок з асинхронного порту ЕОМ на порт ручного управління МП. Насправді, проводиться імітація натискання кнопок на пульті ручного управління в режимі JOINT. Повідомлення з пульта ручного управління надходять на СОМ-порт. У відповідь на таке повідомлення ЕОМ, ґрунтуючись на необхідних впливах управління, створює кодове повідомлення. Для прийому та надсилання кодових повідомлень у програмі створюється потік класу TTranslateThread, який є спадкоємцем стандартного класу TThread.

```
TTranslateThread = class(TThread)
protected
  procedure Execute; override;
  procedure Translate;
end;
```

Метод Execute активізується під час створення потоку та функціонує в циклі доти, поки потік не буде зупинено за допомогою методу Suspend. Метод Translate циклічно активізується з Execute. Процедура Translate отримує повідомлення з COM-порту до буфера RData та надсилає вміст буфера SData.

```

ReadFile(hCom,RData,BufSize,Bytes,nil);
ClearCommError(hCom,Errors,@Stat);
if TransMode=0 then
begin
WriteFile(hCom,RData,BufSize,Bytes,nil);
end
else
begin
WriteFile(hCom,SData,BufSize,Bytes,nil);
end;

```

До того ж, передбачено ретрансляцію повідомлень безпосередньо. Встановивши прапор TransMode на нулі, можна відключити подачу впливів управління з МП на маніпулятор. Це актуально для відключення управління в аварійних ситуаціях.

Водночас, метод Translate відстежує стан руху ланок маніпулятора і в разі, якщо якась ланка досягла заданого положення, він припиняє подачу на неї впливів управління. Для створення буфера впливів управління SData застосовується процедура GenMsg.

Перед запуском потоку, котрий відповідає за управління маніпулятором, здійснюється ініціалізація послідовного порту функцією InitSerialPort. А потім активізується процедура CaptureSignal, яка за умови появи сигналу з пульта ручного управління запускає потік класу TtranslateThread. Якщо сигнал з пульта ручного управління не подається, то з'являється повідомлення про помилку.

Для надходження інформації про об'єкти, що розташовані в робочому просторі робота, залучають систему МЗ. Так, веб-камеру встановлюють на руці робота та підключають її через USB (Universal serial bus) до ЕОМ. Зважаючи на те, що робот та камера підключені до різних ЕОМ, обмін інформацією доцільно проводити за допомогою локальної мережі. На обох машинах інсталиують мережевий протокол TCP/IP з виділенням IP-адрес. Для взаємодії з системою МЗ використовують протокол UDP (User Datagram Protocol). Надсилання й отримання повідомлень здійснюється через 7777 порт.

На початковій стадії роботи система управління МП перевіряє наявність системи МЗ. Для цього вона надсилає повідомлення «connect» та протягом трьох секунд очікує на відповідь «connected». Якщо відповідь відсутня, то фіксується помилка. Якщо ж відповідь отримано, то система розпочинає роботу.

Для отримання якісного зображення з ілюстрацією робочого простору МП, камеру необхідно устаткувати на фокусній відстані перпендикулярно робочій площині. Отже, спочатку система управління роботом забезпечує придатне для зйомки положення маніпулятора, а потім подає системі МЗ команду щодо сканування (повідомлення «start\_scan»). Отримавши цю команду, система МЗ робить знімок та аналізує його. Під час аналізу система МЗ здійснює розпізнавання й ідентифікацію об'єктів, які присутні у кадрі. Після закінчення ідентифікації система МЗ надсилає інформацію про виявлені об'єкти до розподіленої системи автоматизації. Таке повідомлення має вид:

“object: номер об'єкта: координата X: координата Y : кут нахилу ”

Після надходження інформації про всі виявлені об'єкти система МЗ надсилає повідомлення «end\_scan» до розподіленої системи автоматизації. Таке повідомлення трактується як сигнал про початок обчислення реальних координат об'єктів.

Розподілена система автоматизації для обміну повідомленнями застосовує об'єкт класу TNMUDP, який інкапсулює в собі всі необхідні поля та методи для роботи з UDP-пакетами.

Для відправлення повідомлень залучають метод `SendBuffer`:

```
procedure SendBuffer(Buff: array of char; length: integer);
```

Щоб прийняти інформацію від системи МЗ, програма обробляє повідомлення `OnDataReceived`, яке з'являється під час отримання UDP-пакета. Оброблювач повідомлення `OnDataReceived` зчитує дані та заносить їх у проміжний буфер за допомогою методу `ReadBuffer`.

```
procedure ReadBuffer(var Buff: array of char; var length: integer);
```

Зважаючи на те, що в пакеті, як правило, розміщено кілька інформаційних полів, то доцільно його роздробити.

```
iLength:=0;
MyNMUDP.ReadBuffer(szDataBuffer,iLength);
szDataBuffer[iLength]:=chr(0);
iCountParam:=1;
szParam[0]:=@szDataBuffer[0];
iStrLen:=StrLen(szParam[0]);
for i:=1 to iStrLen do
  if szDataBuffer[i]='!' then
    begin
      szDataBuffer[i]:=chr(0);
      szParam[iCountParam]:=@szDataBuffer[i];
      inc(szParam[iCountParam]);
      inc(iCountParam);
    end;
```

Алгоритм дроблення пакета полягає в тому, що інформація в буфері розташована у вигляді рядка з завершальним нулем. У циклі здійснюється заміна двокрапок, які є роздільниками, на символ '\0' – символ закінчення рядка. Крім того, створюється масив покажчиків `szParam`, де покажчики записуються на рядки, котрі з'явилися в наслідок поділу. Таким чином, у `szParam[0]` розташована перша частина пакета, `szParam` – друга.

Інформація щодо об'єктів, з якими співпрацює система, зберігається в базі даних. БД об'єктів створена в форматі Paradox 7 і містить одну таблицю. Поля в таблиці об'єктів можна розподілити на дві групи:

- інформація, корисна для правильного захоплення об'єктів;
- інформація, корисна для ідентифікації об'єктів.

Першою групою полів послуговується розподіленої системи автоматизації для розрахунку координат об'єктів. До неї належать такі поля:

- $X$  – координата  $X$  схвата в системі координат об'єкта;
- $Y$  – координата  $Y$  схвата в системі координат об'єкта;
- $Z$  – координата  $Z$  схвата в системі координат об'єкта;
- $X_x$  – координата  $x$  орта  $X$  системи координат схвата в системі координат об'єкта;
- $X_y$  – координата  $y$  орта  $X$  системи координат схвата в системі координат об'єкта;
- $X_z$  – координата  $z$  орта  $X$  системи координат схвата в системі координат об'єкта;
- $Y_x$  – координата  $x$  орта  $Y$  системи координат схвата в системі координат об'єкта;
- $Y_y$  – координата  $y$  орта  $Y$  системи координат схвата в системі координат об'єкта;
- $Y_z$  – координата  $z$  орта  $Y$  системи координат схвата в системі координат об'єкта;
- $Z_x$  – координата  $x$  орта  $Z$  системи координат схвата в системі координат об'єкта;

–  $Z_y$  – координата  $y$  орта  $Z$  системи координат схвата в системі координат об'єкта;

–  $Z_z$  – координата  $z$  орта  $Z$  системи координат схвата в системі координат об'єкта.

Поля другої групи розглядати докладно не будемо, з огляду на те, що вони призначені виключно для системи МЗ, тому розподіленою системою автоматизації не застосовуються.

Крім полів, які розглянули вище, в БД міститься назва об'єкта та його умовний код.

Для зберігання інформації про об'єкти, виявлені системою МЗ, застосовують масив записів типу TDetail.

```
TDetail = record
    MyX, MyY, MyZ    :single;
    RealX, RealY, RealZ :single;
    X, Y, Z          :single;
    Xx, Yx, Zx      :single;
    Xy, Yy, Zy      :single;
    Xz, Yz, Zz      :single;
    Code              :integer;
    Name              :string;
end;
```

Зважаючи на маніпулювання об'єктами згідно з завданням розподілена система автоматизації повинна спиратися на реальні координати об'єктів. Для розрахунку цих координат застосовуємо такі вихідні дані:

– поточні координати веб-камери в системі, котра пов'язана з основою МП;

– координати одиничних ортів системи координат веб-камери в системі, котра пов'язана з основою МП;

– координати центру мас об'єкта в екранній площині веб-камери;

- кут повороту об'єкта в екранній площині веб-камери;
- необхідні координати схвата в системі координат, котрі пов'язані з об'єктом;
- координати одиничних ортов системи координат схвата в системі, котрі пов'язані з об'єктом.

Для перетворення координат доцільно використовувати множення матриць переходу розмірністю  $4 \times 4$ . Множення двох матриць можна реалізувати в процедурі `Mult_Matrix`.

```
procedure Mult_Matrix(a,b:TMatr;var c:TMatr);
```

Для обчислення дійсних координат перш за все необхідно провести таку ініціалізацію, зокрема:

- матриці переходу з системи координат, пов'язаною зі схопом, до системи, пов'язаною з об'єктом;
- матриці переходу з системи координат, пов'язаною з об'єктом, до системи, пов'язаною з веб-камерою;
- матриці переходу з системи координат, пов'язаною з веб-камерою, до системи, пов'язаною з основою МП.

Власне визначення даних матриць переходу виконується за допомогою таких функцій, як `InitOz`, `InitKo`, `InitRk`. Кожна з функцій заповнює відповідну матрицю переходу. Для застосування цих функцій без труднощів більшість їхніх параметрів має значення за замовчуванням. Звертаємо увагу на умову для захоплення об'єкта, тобто схват має бути розташованим перпендикулярно до нього.

Докладно окреслимо призначення параметрів функцій ініціалізації. По-перше, опишемо параметри функції `InitOz`.

Координати ортів системи координат захоплення в системі координат об'єкта. Дані параметри задають, з огляду на положення захоплення згідно з об'єктом під час захоплення.

$(X_x, X_y, X_z)$  – орт  $X$

$(Y_x, Y_y, Y_z)$  – орт  $Y$

$(Z_x, Z_y, Z_z)$  – орт  $Z$

Орти задають орієнтацію схоплення відносно об'єкта (відповідно до пов'язаної з ним системи координат).

Координати початку системи координат схоплення в системі координат пов'язаної з об'єктом  $(OX, OY, OZ)$ .

Дані координати задають розташування схоплення відповідно до об'єкта. При  $OX=0$ ,  $OY=0$  схоплення знаходиться точно над центром об'єкта на висоті  $OZ$ .

По-друге, розглянемо параметри функції ініціалізації матриці переходу з системи координат, пов'язаної з об'єктом, до системи координат веб-камери (InitKo). Зазначена функція базується на таких параметрах, зокрема:

- координатах центру об'єкта в системі координат веб-камери  $(x, y)$ ;
- куті орієнтації об'єкта (alfa) в робочій площині (в системі координат веб-камери).

По-третє, окреслимо параметри функції ініціалізації матриці переходу з системи координат веб-камери до системи координат, яка пов'язана з основою МП (InitRk). Зазначена функція базується на параметрах, поданих нижче.

Координати ортов системи координат веб-камери в системі координат МП.

$(X_x, X_y, X_z)$  – орт  $X$

$(Y_x, Y_y, Y_z)$  – орт  $Y$

$(Z_x, Z_y, Z_z)$  – орт  $Z$

Ці координати задають орієнтацію веб-камери в просторі щодо системи координат МП.

Координати початку системи координат веб-камери в системі координат пов'язаної з підставою МП ( $OX, OY, OZ$ ).

Дані координати задають розташування веб-камери щодо основи МП. Підкреслимо, що  $OZ$  – висота робочого простору в системі координат, пов'язаної з основою МП, а не висота веб-камери.

Після завершення ініціалізації матриць переходу здійснюється перемноження за допомогою функції `Mult_Matrix`.

Матриця, котра з'явилась за результатами перемноження, містить необхідні координати захоплення в підматриці  $p_{3 \times 1}$ :

$$T = \left[ \begin{array}{c|c} R_{3 \times 3} & p_{3 \times 1} \\ \hline f_{1 \times 3} & 1 \times 1 \end{array} \right]. \quad (3.17)$$

За матрицею  $R_{3 \times 3}$  вираховується кут повороту об'єкта.

## 4 ОХОРОНА ПРАЦІ

### 4.1 Аналіз умов праці в лабораторії

Дослідження з кваліфікаційної роботи проводилися в лабораторії розміром: 4,2 м × 3,5 м × 3,2 м. У приміщенні працює 1 людина. Площа приміщення становить 14,7 м<sup>2</sup>, об'єм – 47,04 м<sup>3</sup>. На одне робоче місце припадає вся площа і об'єм приміщення, відповідає нормативному документу НПАОП 0.00-1.28-10 [12].

Було визначено, згідно з ДСТУ 2293-99, що в лабораторії існують небезпечні та шкідливі фактори:

- знижена рухливість повітря;
- недостатня освітленість робочої зони;
- підвищене значення напруги в електричному ланцюзі, замикання якого може відбутися через тіло людини (небезпечний фактор).

Домінуючий шкідливий виробничий фактор – недолік штучного освітлення.

### 4.2 Промислова безпека в лабораторії

В лабораторії використовується трифазна чотирьохдротова мережа змінного струму напругою 220/380 В, частота 50 Гц, режим нейтралі – глухозаземлена. Згідно НПАОП 40.1-1.21-98, приміщення належить до класу приміщень без підвищеної небезпеки. Умови, які створюють підвищену та особливу небезпеку (підвищена вологість, струмопровідна пил, струмопровідні підлоги, можливість одночасного дотику до заземленим металокопункцій будівлі і металевих поверхонь електроприладів), відсутні.

Для захисту працівників від ураження електричним струмом в лабораторії, згідно НПАОП 40.1-1.32-01, використовується система заземлення TN-C-S типу та захисне відключення.

Необхідно проводити контроль ізоляції відповідно до вимог ПУЕ - 2011. Контроль проводити між нульовим та фазним провідниками, а також між фазами. Опір ізоляції не менше 500 кОм на фазу. Контроль проводити не рідше 1 разу на рік при відключеному електроживленні.

Електропроводка в приміщенні виконана з можливістю заміни: прихована в каналах будівельних конструкцій, що відповідає вимогам НПАОП 40.1-1.32-01.

#### 4.3 Виробнича санітарія і гігієна праці

Роботи, що виконуються в приміщенні лабораторії, проводяться сидячи та не потребують систематичних важких фізичних навантажень, підняття та перенесення важких речей. В таких умовах енерговитрати працівників складають не більше 120 ккал/год. Отже, виконувані роботи відносяться до категорії 1а.

Згідно ДСН 3.3.6.042-99 та категорії виконуваних робіт, для приміщення встановлені наступні оптимальні норми мікроклімату: температура в літній період 23-25 °С, в зимовий період 22-24 °С, відносна вологість повітря 40 – 60 %, швидкість руху повітря  $\leq 0,1$  м/с.

Щоб забезпечити необхідні норми мікрокліматичних параметрів та чистоти повітря в лабораторії використовується кондиціонування. Підтримка зазначених параметрів в холодний період здійснюється системою опалення відповідно до СНиП 2.04.05-91.

Ще одним шкідливим фактором є розумове перенапруження. Основні заходи щодо захисту від розумового перенапруження людини:

– регулярно, через кожні 40-50 хв. робочого часу робити технологічні перерви, для розрядки розумового напруження;

– проведення фізичних вправ (зарядки, розминки) під час перерви.

Рівень загального штучного освітлення приміщення можна перевірити за допомогою методу питомої потужності.

Розрахунок штучного освітлення проводиться методом коефіцієнта використання світлового потоку. Мета перевірного розрахунку – визначення фактичної освітленості в приміщенні. Основна розрахункова формула методу коефіцієнта використання світлового потоку:

$$F = \frac{E_f k_z S_z}{n \eta \gamma N}, \quad (4.1)$$

де  $E_f$  – фактична освітленість, лк;

$S$  – площа освітлюваного приміщення, що залежить від розмірів приміщення;

$z$  – коефіцієнт нерівномірності освітленості ( $z=1,1$ );

$k_z$  – коефіцієнт запасу, що враховує запилення світильників та знос джерел запасу світла в процесі експлуатації;

$N$  – кількість світильників в ряду;

$\eta$  – коефіцієнт використання світлового потоку ламп;

$\gamma$  – коефіцієнт затінення ( $\gamma=0,8$ );

$n$  – кількість рядів світильників.

У таблиці 4.1 наведено вихідні дані.

Таблиця 4.1 – Вихідні дані

$F_l$	$n_l$	$N$	$\eta$	$N$	$k_z$
1600	2	2	0.5	4	1.5

Фактична освітленість складає:

$$E_f = \frac{F n \eta \gamma N}{k_z S_z}, \quad (4.2)$$

$$F = F_i n_i, \quad (4.3)$$

$$F = 1600 \cdot 2 = 3200 \text{ Лк},$$

$$E_f = \frac{3200 \cdot 2 \cdot 0,5 \cdot 4 \cdot 0,88}{1,5 \cdot 14,07 \cdot 1,1} = 441 \text{ Лк}.$$

За результатами розрахунку отримали освітленість складає 441 лк, що належить допустимому діапазону 200-500 лк. У приміщенні нестачі штучного освітлення немає.

#### 4.4 Пожежна безпека лабораторії

Приміщення лабораторії за вибухо- та пожежною небезпекою відноситься до категорії В за НАПБ Б.03.002-2007, оскільки в приміщеннях лабораторії знаходяться тверді горючі матеріали [12]. Основними причинами виникнення пожежі можна вважати:

- випадкове пошкодження ізоляції струмоведучих провідників;
- незадовільний стан вилок, розеток;
- перенавантаження дротів живлення.

Лабораторія розташована в будівлі II ступеня вогнестійкості відповідно до ДБН В.1.1.7-2016 .

За пожежонебезпекою, згідно з ПУЕ-2011, лабораторія належить зоні П – Па [12]. У ній присутні тверді горючі речовини, папір, меблі, одяг.

Заходи пожежної профілактики відповідно до ГОСТ 12.1.004-91. ССБТ, до яких входять: застосування системи запобігання пожежі, протипожежного захисту, виконання організаційних заходів.

Згідно НАПБ Б03.001-2004 в приміщеннях розміщені первинні засоби пожежогасіння – вуглекислотний вогнегасник ВВК-1,4 з розрахунку 1 вогнегасник на 3 ПК, але не менше 1 на приміщення [12].

Згідно ДБН В.2.5.56-2010 в приміщеннях встановлено точковий димової-пожежний сповіщувач ДП-1, який контролює площу до 86 м<sup>2</sup>.

## ВИСНОВКИ

Під час роботи над кваліфікаційною роботою створено інформаційне моделювання функціонування МП в виробничому середовищі на інформаційному рівні організації тактико-технічного планування інформаційно-рухових дій.

Продемонстровано узагальнену архітектуру МП, а також загальну схему побудови її моделі. Досліджено алгоритми планування.

Визначено інформаційні моделі багатокomпонентної МП. Обрано спосіб організації управління МП.

Розглянуто метод координації і планування інтелектуалізації управління МП як частини роботизованого виробництва.

Крім того, у ході роботи опрацьовано кілька підходів, пов'язаних з управлінням розподіленими робототехнічними системами. Така проблема є надзвичайно важливою, отже, перспективною, зважаючи на безумовну тенденцію об'єднання технічних систем. У результаті чого здійснюється суттєве підвищення функціональних можливостей складної системи, зокрема ЕОМ, під'єднаний до локальної обчислювальної мережі має незрівнянно більше можливостей за ізольований ЕОМ.

Розроблено метод координації та планування дій МП. Запропоновано підхід, який полягає у поданні цього рівня системи управління як мережі спеціальним чином побудованих кінцевих автоматів.

Розроблено систему підтримки та прийняття рішень для МП та програмну реалізацію розподіленої системи автоматизації.

При виконанні розділу «Охорона праці» були визначені небезпечні і шкідливі виробничі фактори. Таким чином, були розроблені заходи і технічні засоби щодо забезпечення безпеки праці працюючого персоналу. Застосування цих заходів дозволить максимально знизити ймовірність отримання травм при роботі, а також поліпшити умови роботи.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008:2015 Інформація та документація «Звіти у сфері науки і техніки». Структура та правила оформлювання. / В. Земцева; Ю. Поліщук, канд. фіз.-мат. наук; Р. Санченко, канд. техн. наук; Л. Шрамко; А. Ямчук (науковий керівник) ДП «УкрНДНЦ» від 22 червня 2015р. № 61 з 2017- 07-01.

2. Методичні вказівки з підготовки кваліфікаційної роботи бакалавра для студентів усіх форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» / Упоряд.: І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарева, С.П. Новоселов, О.В Сичова. – Харків: ХНУРЕ, 2022. – 55 с.

3. Навчальний посібник з підготовки кваліфікаційної роботи бакалавра для здобувачів вищої освіти денної і заочної форм навчання спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології» та 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» освітньої програми «Системна інженерія»: навчальний посібник / І. Ш. Невлюдов, О. В. Токарева, О. М. Цимбал, А. І. Бронніков ; М-во освіти і науки України, ХНУРЕ. – Харків : Видавництво Іванченка І. С., 2023. – 218 с.

4. Невлюдов І.Ш. Людино-машинний інтерфейс в технічних засобах автоматизації: Навчальний посібник / І.Ш. Невлюдов, О.І. Филипченко, Б.О. Шостак. – Харків : «ХТМТ», 2019. – 244 с.

5. Невлюдов І.Ш. Технічні засоби автоматизації: Підручник / І.Ш. Невлюдов, А.О. Андрусевич, О.І. Филипченко, Н.П. Демська, С.П. Новоселов. – Кривий Ріг : Криворізький коледж НАУ, 2019. – 366 с.

6. Костюк В. І. Робототехніка і мехатроніка / В. І. Костюк, Г. О. Спину. – К.: Вища школа, 1994. – 446 с.

7. Субботін С.О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: Навчальний посібник / С.О. Субботін. – Запоріжжя: ЗНТУ, 2008. – 341 с.

8. Платов І. М. Використання лазерного датчика відстані VL53L0X для систем зору мобільних роботів / І. М. Платов, О. М. Павловський // XIII Науково-практична конференція студентів, аспірантів та молодих вчених «Погляд у майбутнє приладобудування», 13-14 травня 2020 р., м. Київ, Україна: збірник праць конференції. – Київ: КПІ ім. Ігоря Сікорського, 2020. – С. 59-62.

9. Савченко, А. С. Синельников О.О. Методи та системи штучного інтелекту: навч. посібник– Київ : НАУ, 2017. – 176 с.

10. Лубко Д.В., Шаров С.В. Методи та системи штучного інтелекту: навч. посібник – Мелітополь: ФОП Однорог Т.В., 2019. – 264с

11. Діагностика та контроль робочих процесів: навч. посіб. для студентів спеціальності «Прикладна механіка» денної та дистанційної форм навчання / В. М. Доля – Харків: НТУ «ХПІ», 2019. – 129 с.

12. Організація керування умовами праці» підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету [Електронний ресурс] / ХНУРЕ; розроб.: Т.Є. Стиценко, Г.В. Пронюк, Н.М. Сердюк. – Харків, 2017. – 108 с.