

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ  
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова праця  
на правах рукопису

**БРОННІКОВ АРТЕМ ІГОРОВИЧ**

УДК 004.896:007.52]:004.416.3

**ДИСЕРТАЦІЯ**

«МОДЕЛІ ТА МЕТОДИ АДАПТИВНОГО ВІЗУАЛЬНОГО  
КЕРУВАННЯ РОБОТАМИ»

05.13.07 – автоматизація процесів керування

технічні науки

Подається на здобуття наукового ступеня кандидата технічних наук

Дисертація містить результати власних досліджень. Використання ідей,  
результатів і текстів інших авторів мають посилання на відповідне джерело  
\_\_\_\_\_ А.І. Бронніков

Науковий керівник: Цимбал Олександр Михайлович, доктор технічних наук,  
доцент

Цей примірник дисертаційної роботи  
ідентичний за змістом з іншими,  
поданими до спеціалізованої вченої ради Д 64.052.08

Вчений секретар спецради Д 64.052.08

Плісс І. П.

Харків 2021

## АНОТАЦІЯ

*Бронніков А.І.* Моделі та методи адаптивного візуального керування роботами – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.07 – автоматизація процесів керування. Харківський національний університет радіоелектроніки, Міністерство освіти і науки України, Харків, 2021.

Дисертаційна робота присвячена вирішенню актуального завдання розробки моделей та методів адаптивного візуального керування мобільними роботизованими засобами.

Об'єктом дослідження в роботі є процеси керування гнучким інтегрованим роботизованим виробництвом.

Предметом дослідження є моделі та методи автоматизованого керування мобільними транспортними роботами у складі гнучкого інтегрованого виробництва.

Методи дослідження засновані на використанні аналізу результатів досліджень сучасних теоретичних і прикладних розробок вітчизняних і зарубіжних вчених в області моделей і методів адаптивного візуального керування. Для вирішення поставлених задач використано методи теорії множин – для розробки інформаційно-логічної моделі робочого простору гнучкї інтегрованої виробничої системи (ГІВС), функціональної моделі ГІВС, а також для удосконалення моделей розпізнавання та ідентифікації об'єктів, керування мобільним роботом у просторі ГІВС; методи теорії автоматичного керування та нечітких множин – для моделювання роботи адаптивної системи керування роботом; методи теорії обробки зображень – для програмної реалізації завдань, що входять у склад системи адаптивного візуального керування. При проведенні експериментальних досліджень, для практичної перевірки результатів і розробки програмного забезпечення

використаний мови програмування C/C++ і Python, програмні бібліотеки MFC, OpenCV, NumPy, середовища розробки Microsoft Visual Studio, PyCharm та пакет прикладних програм GNU Octave для моделювання роботи системи адаптивного візуального керування.

Наукова новизна отриманих результатів:

– вперше розроблена інформаційно-логічна модель робочого простору, що відображає об'єкти ГІВС, характерні для завдань інтелектуального керування роботами та зв'язки між об'єктами, і яка, на відміну від існуючих враховує дискретність та заповненість виробничого робочого простору; ґрунтується на інформації, отриманій від об'єктової системи комп'ютерного зору; визначає та враховує властивості об'єктів, що розміщуються в робочому просторі; враховує взаємодію, упорядкованість та сумісність об'єктів;

– вперше розроблена функціональна модель ГІВС, яка забезпечує опис взаємодії об'єктів на рівні виконання поставлених перед ГІВС виробничих, транспортувальних, сервісних, моніторингових, управлінських та інших завдань, та, на відміну від існуючих, ґрунтується на інформаційно-логічній моделі робочого простору враховує особливості технологічних операцій, містить детальний опис властивостей об'єктів та їх взаємодії, включає можливість кількісної оцінки вартості та якості робіт;

– вдосконалено метод адаптивного візуального керування, що описує сукупність прийомів та операцій, які слід застосувати у інтелектуальних системах керування роботів для планування завдання переміщень на основі отримання зорової інформації від систем комп'ютерного (технічного) зору, та, на відміну від існуючих методів, орієнтований на використання в мобільних робототехнічних системах;

– отримала подальший розвиток модель керування мобільним роботом у просторі ГІВС, що здійснює керування шасі двоколісного робота, та, на відміну від існуючих, функціонує на основі інформації від об'єктових/локальних систем комп'ютерного зору та виконує функції

розпізнавання та ідентифікації об'єктів ГІВС, забезпечує розрахунок координат та швидкостей роботизованої платформи та інших об'єктів ГІВС, розрахунок лінійних та криволінійних траєкторій переміщень у дискретному робочому просторі, подання маршруту у вигляді послідовностей переміщень, забезпечує організацію керування транспортувальними роботами за їх візуальним положенням; містить нечітку складову з розгорнутим описом процедур формування лінгвістичних змінних, пов'язаних з динамікою руху мобільної роботизованої платформи та формалізує набір продукційних правил нечіткого регулятора.

Практичне значення отриманих результатів полягає у розробці системи керування мобільним транспортувальним роботом, яке реалізує запропоновані моделі і метод, практично забезпечує аналіз робочого простору за допомогою системи комп'ютерного/технічного зору, розпізнавання та ідентифікацію об'єктів робочого простору та робототехнічних засобів, отримання їх просторових координат, підтримку візуального цілевказання маршруту мобільного робота та функціонування інтелектуальної системи підтримки прийняття рішень.

Результати досліджень можуть бути використані в дослідженнях та розробках, що виконуються у ДП «Науково-дослідний технологічний інститут приладобудування» (м. Харків), АТ «Хартрон» (м. Харків), ПАТ «ФЕД» (м. Харків), ПАВ «Вертикаль» (м. Харків), ТОВ «Укрінтех» (м. Харків) та інших організаціях та підприємствах, що вирішують проблеми створення гнучких інтегрованих систем, автоматизованих та роботизованих пристроїв різного призначення.

Результати розробок і досліджень впроваджені в освітній процес у Харківському національному університеті радіоелектроніки на кафедрі комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології при викладанні дисциплін «Комп'ютерні системи керування рухомими об'єктами» та «Автоматизовані системи керування роботизованим виробництвом» (акт

від 6 лютого 2021 року). Також інформаційно-логічна модель робочого простору, функціональна модель взаємодії об'єктів, метод адаптивного візуального керування впроваджені у наукову діяльність у ННЦ «Інститут судових експертиз ім. Засл. проф. М.С. Бокаріуса» (акт від 26 червня 2020 року).

За результатами дисертаційних досліджень опубліковано 38 наукових праць, з них 19 статей, у тому числі – 1 стаття у науковому фаховому виданні у Турецькій республіці (входить у Організацію економічного співробітництва та розвитку (ОЕСР)); 16 статей в наукових фахових виданнях України з технічних наук; 2 статті у наукових виданнях інших держав; 19 тез доповідей, що опубліковані в матеріалах міжнародних наукових конференцій, з яких 2 включено до наукометричної бази даних Scopus.

Ключові слова: Industry 4.0, Flexible Integrated System, автоматизація процесів керування, робототехніка, моделі, методи, адаптація, адаптивне візуальне керування.

#### *Список публікацій здобувача*

##### **Статті без співавторства**

1. Бронніков А.І., Методи пошуку оптимального керування / Технологія приборостроения. Харьков, 2015, № 2 , С. 53 - 55.

2. Бронніков А.І., Адаптивное визуальное управление производственным агентом / Технологія приборостроения. Харьков, 2016, № 3, С. 3 - 6.

3. Бронніков А.І., Сучасний стан та перспективи розвитку гнучких інтегрованих систем / Технологія приборостроения. Харьков, 2018, № 1, С. 54 - 57.

##### **Статті у співавторстві**

4. Artem Bronnikov, Nevliudov Igor, Oleksandr Tsymbal. Flexible manufacturing tendencies and improvements with visual sensing / Eskisehir

Technical University Journal of Science and Technology. Applied Sciences and Engineering, 2019. Vol. 20, ICONAT issue, P. 77-83. (фахове періодичне видання за кордоном, Туреччина)

5. Цымбал А.М., Бронников А.И., Литвинова А.Е., Чернышенко О.Е. Технологии программирования и робототехника / Восточно-Европейский журнал передовых технологий, Харьков, 2009, № 3/2 (39), С. 56-60.

6. Цимбал О.М., Бронников А.И. Адаптивність у прийнятті рішень роботів / Восточно-Европейский журнал передовых технологий, Харьков, 2011, № 4/4 (52), С. 40 - 43.

7. Цимбал О.М., Бронников А.И. Адаптивні процеси у завданнях робототехніки // Системи обробки інформації, Харків. – 2012. – Вип. 3 (101), том 1, С. 68 -73.

8. Цимбал О.М., Бронников А.И., Куценко О.И., Шеін Є.С. Концепція інтелектуальних виробничих агентів та особливості її реалізації // Восточно-Европейский журнал передовых технологий, Харьков, 2014, № 1/2 (67), С. 9 - 13.

9. Бронников А.И., Цимбал О.М., Фомовський Ф.М., Інтелектуальні засоби в системі керування виробничим агентом / Технологія приборостроения. Харьков, 2014, № 2 , С. 26 - 30.

10. Цимбал О.М., Бронников А.И., Нгуєн Т.М.Л., Бекметова А.О. Інтелектуальні виробничі системи та перспективи їх реалізації / Технологія приборостроения. Харьков, 2016, № 1, С. 29 - 34.

11. Бронников А.И., Цимбал О.М., Сіноутін А.М. Мехатроніка у освітньому та загальнонауковому контексті / Системи керування, навігації та зв'язку. Полтава, 2017, № 2 (42) , С. 30 - 32.

12. Цимбал О.М., Бронников А.И. Прямий метод адаптації для керування робото технічною платформою. / Технологія приборостроения. Харків, 2016, № 3, С. 19 - 21.

13. Nevlyudov I., Tsymbal O., Bronnikov A. Intelligent means in the system of managing a manufacturing agent / Сучасний стан наукових досліджень та

технологій в промисловості. 2018. № 1 (3). С. 33-47.

14. Tsymbal O., Bronnikov A. Decision-making information technology for flexible integrated manufacturing / Сучасний стан наукових досліджень та технологій в промисловості. 2019. № 2 (8). С. 105-112.

15. Nevliudov I., Tsymbal O., Bronnikov A., Mordyk O. Internet of things for robotic projects / Сучасний стан наукових досліджень та технологій в промисловості. 2020. № 3 (13). С. 58-64.

16. Цимбал О.М., Замірець М.В., Мордик О.О., Бронніков А.І., Корнілова Ю.Б. Інтелектуальні методи відеоспостереження: аналіз методів реалізації / Технологія приборостроєння. Харків, 2020, № 1, С. 35 - 40.

17. Igor Nevliudov, Vladyslav Yevsieiev, Murad Omarov, Artem Bronnikov and Viacheslav Liashenko. Method of Algorithms for Cyber-Physical Production Systems Functioning Synthesis //International Journal of Emerging Trends in Engineering Research (IJETER), Volume 8 No.10 (October 2020), pp: 7465 – 7473. ISSN 2347 - 3983 (DOI:10.30534/ijeter/2020/1278102020) (фахове видання за кордоном, Індія )

18. Цымбал А.М., Бронников А.И. Моделирование адаптивного принятия решений в ИСУ роботом // Вестник БГТУ им. В.Г. Шухова. – 2013. – №4, С. 173-176. (фахове видання за кордоном).

19. V. Yevsieiev, A. Bronnikov. Development of databases interconnection “essences” information model for cyber-physical production systems additive cyber design creation automation //Збірник наукових праць національного університету кораблебудування ім. адмірала Макарова, №3 (481), 2020р. pp.: 56- 62 ISSN 2311-3405 (Print), ISSN 2313-0415 (Online) (DOI: 10.15589/znp2020.3(481).7)

### **Конференції**

20. Oleksandr Tsymbal, Artem Bronnikov, Andriy Yerokhin. Adaptive Decision-making for Robotic tasks // Proceeding of IEEE 8th International Conference on Advanced Optoelectronics and Lasers, CAOL\*2019,

DSMOLE\*2019: Scientific Workshop “Data Science in Modern Optoelectronics and Laser Engineering”, September 6-8, Sozopol, Bulgaria. – P. 594-597. (Scopus)

21. Oleksandr Tsymbal, Artem Bronnikov, Paolo Mercorelli. Decision-making models for Robotic Warehouse // Proceedings of 2020 International Symposium on Power Electronics, Electrical Drives, Automation and Motion, Virtual Meeting, June 24-26, P. 546-551. (Scopus)

22. Цымбал А.М. Программное моделирование системы управления мобильным роботом / А.М. Цымбал, А.И. Бронников // Автоматизация: проблемы, идеи, решения: Материалы международной научно-технической конференции 6 – 10 сентября 2010 г.: тезисы докл. – Севастополь, 2010. С. 224 – 226.

23. Цимбал О.М., Бронников А.І. Особливості адаптивного візуального керування мобільними роботами // Перша Всеукраїнська науково-практична конференція «Актуальні проблеми створення електронних засобів промислових автоматизованих систем»: матеріали конференції. – Сєвєродонецьк: Технол. Ін-т Східноукр. Нац. Ун-ту ім. В. Даля, 2011. С. 26 – 28.

24. Цимбал О.М. Адаптивні методи та їх реалізація в робототехніці / О.М. Цимбал, А.І. Бронников // Системний аналіз та інформаційні технології: Матеріали міжнародної науково-технічної конференції 23–28 травня 2011 р.: тези доп.– Київ, 2011.– С. 335.

25. Tsymbal A.M. Decision-making in Robotics and adaptive tasks / A.M. Tsymbal, A. I. Bronnikov // Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2012), Kharkov, Sept. 14-17, 2012. – P. 417-420.

26. Бронников А.І., Цимбал О.М. Інтелектуальне керування мобільним роботом в обмеженому робочому просторі // Матеріали 5-ї міжнародної науково-технічної конференції «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів керування». 21-22 квітня 2016 р. – Харків: ДП «ХНДІТП», 2016. – С. 33 – 34.

27. Цимбал О.М., Сінотін А.М., Бронников А.І. Інтервальні засоби у



моделі керування роботом // Матеріали першої міжнародної науково-практичної конференції «Computer and informational systems and technologies». 20-22 квітня 2017 р. – Харків: ХНУРЕ, 2017. С. 43-44.

28. Цимбал О.М., Бронніков А.І. Інтелектуальне керування гнучкою роботизованою системою // Матеріали першої міжнародної науково-практичної конференції «Computer and informational systems and technologies». 20-22 квітня 2017 р. – Харків: ХНУРЕ, 2017. С. 45-46.

29. О.М. Цимбал, А.І. Бронніков. Формування стратегій прийняття рішень в завданнях робототехніки // Матеріали V Міжнародної науково-технічної Інтернет-конференції Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами», 2018, С. 213-214.

30. Цимбал О.М., Бронніков А.І. Інтернет роботизованих речей: огляд концепції // Матеріали 3-ї міжнародної науково-практичної конференції «Computer and informational systems and technologies». 23-24 квітня 2019 р. – Харків: ХНУРЕ, 2019. – С. 95-96.

31. Nevliudov Igor, Oleksandr Tsymbal, Artem Bronnikov. The Implementation of Manufacturing Agent's Concept for Flexible Production Systems // 1st International Scientific-Practical Conference - Modern Information, Measurement and Control Systems: Problems and Perspectives 2019, 1-2 July 2019, Baku. – P. 187.

32. Vladyslav Yevsieiev, Artem Bronnikov, Analysis of architectural models for representing the integration of cyber-physical production systems hierarchical levels // Manufacturing & Mechatronic Systems 2020: Proceedings of IVth International Conference, Kharkiv, October 22-23, 2020: Theses of Reports / [Ed. I.Sh. Nevlyudov (chief editor).] - Kharkiv .: [electronic version], 2020., pp 17-19.

33. Vladyslav Yevsieiev, Artem Bronnikov Analysis of the cyber-physical production systems implementation impact to achieve the goals of lean production //The II th International scientific and practical conference «Development of

scientific and practical approaches in the era of globalization» (September 28-30, 2020). Boston, USA 2020. pp. 221-226. ISBN - 978-1-64945-867-4. (DOI: 10.46299/ISG.2020.II.II) (Google Scholar, Crossref)

34. Yevsieiev V., Bronnikov A. Analysis of the CMMI model application for solving the tasks of CPPS control processes automation development // The IV th International scientific and practical conference «Actual Trends of Modern Scientific Research» (11-13 October 2020) MDPC Publishing, Munich, Germany, pp.128-132. ISBN 978-3-954753-02-4

35. Yevsieiev V., Bronnikov A. Information systems development methodologies application analysis for cyber-physical production systems development // III International scientific-practical conference “Theory, science and practice” (October 05-08, 2020), Tokyo, Japan, pp. 399-401, 2020. ISBN - 978-1-64945-868-1 (DOI: 10.46299/ISG.2020.II.III) (Google Scholar, Crossref)

36. Yevsieiev V., Bronnikov A. Analysis of the multi-agent systems application to solve the problem of cyberphysical production systems development // The IV th International scientific and practical conference «Integration of scientific bases into practice», (October 12-16, 2020), Stockholm, Sweden, pp. 459-462. ISBN - 978-1-64945-864-3 (DOI:10.46299/ISG.2020.IV) (Google Scholar, Crossref)

37. Yevsieiev V., Bronnikov A. Structural model of a cyber-physical production system based on multi-agent systems analysis // Матеріали VII Міжнародної науково-технічної Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами», 26 листопада 2020. [Електронний ресурс] – К: НУХТ, 2020, с. 312-313.

38. Yevsieiev V., Bronnikov A. (2020) Complexity development analysis of cyber-physical production systems for smart manufacturing // The X th International scientific and practical conference «Trends in the development of modern scientific thought» November 23-26, 2020 Vancouver, Canada. pp. 699-703 (ISBN - 978-1-63649-921-5) (DOI - 10.46299/ISG.2020.II.X) (Google Scholar, Crossref)

## ABSTRACT

*Bronnikov A.I.* Models and methods of robots adaptive visual control. – Qualifying scientific work on the rights of the manuscript.

Thesis for the degree of technical sciences candidate in specialty 05.13.07 – Automation of control processes. – Kharkiv National University of Radio Electronics, Kharkiv, 2021.

The thesis is devoted to solving the urgent problem of developing models and methods of adaptive visual control of mobile robotic means..

The object of research is the control processes of flexible integrated robotic radio electronic manufacturing.

The subject of the research is the models and methods of flexible integrated robotic radio electronic manufacturing automated control.

The research methods are based on the usage of the research results analysis of modern theoretical and applied developments of native and foreign scientists in the field of adaptive visual control models and methods. To solve the tasks there are used set of theory methods – for the development of a declarative model of the FIS workspace, a functional FIS model, as well as for improving models of objects recognition and identification, control of a mobile robot in the FIS workspace; methods of the automatic control theory and fuzzy sets – for modeling the work of an adaptive control robot system; methods of image processing theory – for tasks software implementation that are part of the adaptive visual control system. During experimental research, for practical verification of the results and software development, the programming languages C/C++ and Python, MFC, OpenCV, NumPy software libraries, Microsoft Visual C++ 2015 and 2019 development environments, PyCharm and the GNU Octave application package were used to simulate the work of the adaptive visual control system.

There are scientific novelty of the results obtained:

– an information-logical model of the workspace that reflects FIS objects characteristic of the intelligent work control tasks and communication between

objects, and which, in contrast to the existing ones, takes into account the discreteness and fullness of the production workspace; based on information received from object computer vision systems; defines and takes into account the properties of objects placed in the workspace; takes into account the interaction, orderliness and compatibility of objects was firstly developed;

- a functional FIS model, which provides a description of objects interaction at the level of production performance, transport, service, monitoring, control and other tasks assigned to FIS, and, unlike existing ones, is based on a declarative model of the workspace takes into account the features of technological operations, contains a detailed description of objects properties and their interactions, including the possibility of quantifying the cost and quality of work was firstly developed;

- a model for controlling a mobile robot in the FIS space, controls the chassis of a two-wheeled operation, and, unlike existing ones, operates on the basis of information from global / local computer vision systems and performs the functions of FIS objects recognition and identification, provides the calculation of coordinates and speeds of a robotic platform and other FIS objects, calculation of linear and curvilinear movements trajectories in a discrete workspace, presentation of the route in the movements sequences form, ensures the organization of transport operations control by their visual position was improved;

- the method of adaptive visual control has been improved, it describes a set of techniques and operations that should be applied in intelligent control systems of robots for planning a movement task based on obtaining visual information from computer (technical) vision systems, and, unlike existing methods, is focused on using mobile robotic systems;

- the model for controlling a mobile robot in the FIS space was further developed, it controls the chassis of a two-wheeled operation, and, unlike existing ones, operates on the basis of information from global / local computer vision systems and performs the functions of recognition and identification of GIS objects, provides the calculation of coordinates and velocities a robotic platform and other GIS objects, calculation of linear and curvilinear trajectories of movements in a

discrete workspace, presentation of a route in the form of sequences of movements, ensures the organization of control of transporting works according to their visual position; the fuzzy component of the model contains a detailed description of the procedures for the formation of linguistic variables associated with the dynamics of the movement of the mobile robotic platform and formalizes the set of products of the fuzzy controller.

The practical significance of the results obtained lies in the fact that information and software have been developed for an adaptive visual control system for a mobile transport robot, which implements the proposed models and method, practically provides an analysis of the workspace using a computer / technical vision system, recognition and identification of objects in the workspace and robotic means, obtaining their spatial coordinates, supporting visual target designation of the route of the mobile robot and the functioning of an intelligent decision support system.

The results of development and research have been introduced at the Kharkiv National University of Radio Electronics in the educational process in the specialty 151 Automation and computer-integrated technologies in the disciplines Computer control systems for moving objects and Automated control systems for robotic production (act of February 15, 2021). in scientific activity at the NSC "Institute of Forensic Expertise named after Honored prof. M.S. Bocarius " (act of June 26, 2020).

Based on the results of thesis, 38 publications on the topic were published among them 19 articles, of which – 1 an article in a scientific professional publication in the Republic of Turkey (part of the Organization for Economic Cooperation and Development (OECD)) 16 articles in scientific publications of Ukraine on technical sciences; 2 articles in publications of other states; 19 abstracts published in the materials of international scientific conferences, of which 2 are included in the scientometric Scopus database.

Key words: Industry 4.0, Flexible Integrated System, automation of control processes, robotics, models, methods, adaptation, adaptive visual control.

*List of applicant's publications***Articles without co-authorship**

1. Bronnikov A.I., Metody poshuku optimal'noho upravlinnya / Tekhnolohyya pryborostroenyia. Khar'kov, 2015, № 2 , S. 53 - 55.
2. Bronnikov A.I., Adaptivnoe vyzual'noe upravlenye proyzvodstvennym ahentom / Tekhnolohyya pryborostroenyia. Khar'kov, 2016, № 3 , S. 3 - 6.
3. Bronnikov A.I. Suchasnyy stan ta perspektyvy rozvytku hnuchkykh intehrovanykh system / Tekhnolohyya pryborostroenyia. Khar'kov, 2018, № 1, S. 54 - 57.

**Articles with co-authorship**

4. Artem Bronnikov, Nevliudov Igor, Oleksandr Tsymbal. Flexible manufacturing tendencies and improvements with visual sensing / Eskisehir Technical University Journal of Science and Technology. Applied Sciences and Engineering, 2019. Vol. 20, ICONAT issue, S. 77-83. (фахове періодичне видання за кордоном, Туреччина)
5. Tsymbal A.M., Bronnikov A.I., Litvinova A.Ye., Chernyshenko O.Ye. Tekhnologii programmirovaniya i robototekhnika / Vostochno-Yevropeyskiy zhurnal peredovykh tekhnologiy, Khar'kov, 2009, № 3/2 (39), S. 56-60.
6. Tsymbal O.M., Bronnikov A.I. Adaptivnist' u pryunyatti rishen' robotiv / Vostochno-Evropeyskiy zhurnal peredovykh tekhnolohyy, Khar'kov, 2011, № 4/4 (52), S. 40 - 43.
7. Tsymbal O.M., Bronnikov A.I. Adaptivni protsesy u zavdanniyakh robototekhniki // Systemy obrobky informatsiyi, Kharkiv. – 2012. – Vyp. 3 (101), tom 1, S. 68 -73.
8. Tsymbal O.M., Bronnikov A.I., Kutsenko O.I., Shein YE.S. Kontsepsiya intelektual'nykh vyrobnychykh ahentiv ta osoblyvosti yiyi realizatsiyi // Vostochno-Evropeyskiy zhurnal peredovykh tekhnolohyy, Khar'kov, 2014, № 1/2 (67),

S. 9 - 13.

9. Bronnikov A.I., Tsymbal O.M., Fomovs'kyi F.M., Intelektual'ni zasoby v systemi keruvannya vyrobnychym ahentom / Tekhnolohyya pryborostroenyya. Khar'kov, 2014, № 2 , S. 26 - 30.

10. Tsymbal O.M., Bronnikov A.I., Nhuyen T.M.L., Bekmetova A.O. Intelektual'ni vyrobnychi systemy ta perspektyvy yikh realizatsiyi / Tekhnolohyya pryborostroenyya. Khar'kov, 2016, № 1 , S. 29 - 34.

11. Bronnikov A.I., Tsymbal O.M., Sinotin A.M. Mekhatronika u osvitr'omu ta zahal'nonaukovomu konteksti / Systemy upravlinnya, navihatsiyi ta zv'yazku. Poltava, 2017, № 2 (42) , S. 30 - 32.

12. Tsymbal O.M., Bronnikov A.I. Pryamyi metod adaptatsiyi dlya keruvannya roboto tekhnichnoyu platformoyu. / Tekhnolohyya pryborostroenyya. Kharkiv, 2016, № 3, S. 19 - 21.

13. Nevlyudov I., Tsymbal O., Bronnikov A. Intelligent means in the system of managing a manufacturing agent / Suchasnyy stan naukovykh doslidzhen' ta tekhnolohiy v promyslovosti. 2018. № 1 (3). S. 33-47.

14. Tsymbal O., Bronnikov A. Decision-making information technology for flexible integrated manufacturing / Suchasnyy stan naukovykh doslidzhen' ta tekhnolohiy v promyslovosti. 2019. № 2 (8). S. 105-112.

15. Nevliudov I., Tsymbal O., Bronnikov A., Mordyk O. Internet of things for robotic projects / Suchasnyy stan naukovykh doslidzhen' ta tekhnolohiy v promyslovosti. 2020. № 3 (13). S. 58-64.

16. Tsymbal O.M., Zamirets' M.V., Mordyk O.O., Bronnikov A.I., Kornilova YU.B. Intelektual'ni metody videosposterezhennya: analiz metodiv realizatsiyi / Tekhnolohyya pryborostroenyya. Kharkiv, 2020, № 1, S. 35 - 40.

17. Igor Nevliudov, Vladyslav Yevsieiev, Murad Omarov, Artem Bronnikov and Viacheslav Liashenko. Method of Algorithms for Cyber-Physical Production Systems Functioning Synthesis //International Journal of Emerging Trends in Engineering Research (IJETER), Volume 8 No.10 (October 2020), pp: 7465 – 7473. ISSN 2347 - 3983 (DOI:10.30534/ijeter/2020/1278102020) (professional

publication abroad, India)

18. Tsymbal AM, Bronnikov AI Modeling of adaptive decision-making in ISU by a robot // Bulletin of BSTU. V.G. Shukhova. - 2013. - №4, S. 173-176. (professional publication abroad).

19. V. Yevsieiev, A. Bronnikov. Development of databases interconnection “essences” information model for cyber-physical production systems additive cyber design creation automation // Zbirnyk naukovykh prats' natsional'noho universytetu korablebuduvannya im. admirala Makarova, №3 (481), 2020p. S.: 56- 62 ISSN 2311-3405 (Print), ISSN 2313-0415 (Online) (DOI: 10.15589/znp2020.3(481).7)

### Conferences

20. Oleksandr Tsymbal, Artem Bronnikov, Andriy Yerokhin. Adaptive Decision-making for Robotic tasks // Proceeding of IEEE 8th International Conference on Advanced Optoelectronics and Lasers, CAOL\*2019, DSMOLE\*2019: Scientific Workshop “Data Science in Modern Optoelectronics and Laser Engineering”, September 6-8, Sozopol, Bulgaria. – S. 594-597. (Scopus)

21. Oleksandr Tsymbal, Artem Bronnikov, Paolo Mercorelli. Decision-making models for Robotic Warehouse // Proceedings of 2020 International Symposium on Power Electronics, Electrical Drives, Automation and Motion, Virtual Meeting, June 24-26, S. 546-551. (Scopus)

22. Tsymbal A.M. Programmnoye modelirovaniye sistemy upravleniya mobil'nym robotom / A.M. Tsymbal, A.I. Bronnikov // Avtomatizatsiya: problemy, idei, resheniya: Materialy mezhdunarodnoy nauchno-tekhnicheskoy konferentsii 6 – 10 sentyabrya 2010 g.: tezisy dokl. – Sevastopol', 2010. – S. 224 – 226.

23. Tsymbal O.M., Bronnikov A.I. Osoblyvosti adaptivnoho vizual'noho keruvannya mobil'nymy robotamy // Persha Vseukrayins'ka naukovo-praktychna konferentsiya «Aktual'ni problemy stvorenniya elektronnykh zasobiv promyslovykh avtomatyzovanykh system»: materialy konferentsiyi. – Syevyerodonets'k: Tekhnol. In-t Skhidnoukr. Nats. Un-tu im. V. Dalya, 2011. – S. 26 – 28.

24. Tsymbal O.M. Adaptivni metody ta yikh realizatsiya v robototekhnitsi



/ O.M. Tsymbal, A.I. Bronnikov // Systemnyy analiz ta informatsiyi tekhnolohiyi: Materialy mizhnarodnoyi naukovo-tekhnichnoyi konferentsiyi 23–28 travnya 2011 r.: tezy dop.– Kyiv, 2011.– S. 335.

25. Tsymbal A.M. Decision-making in Robotics and adaptive tasks / A.M. Tsymbal, A. I. Bronnikov // Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2012), Kharkov, Sept. 14-17, 2012. – P. 417-420.

26. Bronnikov A.I., Tsymbal O.M. Intelktual'ne keruvannya mobil'nym robotom v obmezhenomu robochomu prostori» // Materialy 5-yi mizhnarodnoyi naukovo-tekhnichnoyi konferentsiyi «Suchasni napryamy rozvytku informatsiyno-komunikatsiynykh tekhnolohiy ta zasobiv upravlinnya». 21-22 kvitnya 2016 r. – Kharkiv: DP «KHNDITP», 2016. – S. 33 – 34.

27. Tsymbal O.M., Sinotin A.M., Bronnikov A.I. Interval'ni zasoby u modeli keruvannya robotom // Materialy pershoi mizhnarodnoyi naukovo-praktychnoyi konferentsiyi «Computer and informational systems and technologies». 20-22 kvitnya 2017 r. – Kharkiv: KHNURE, 2017. – S. 43-44.

28. Tsymbal O.M., Bronnikov A.I. Intelktual'ne keruvannya hnuchkoyu robotyzovano systemoyu // Materialy pershoi mizhnarodnoyi naukovo-praktychnoyi konferentsiyi «Computer and informational systems and technologies». 20-22 kvitnya 2017 r. – Kharkiv: KHNURE, 2017. – S. 45-46.

29. O.M. Tsymbal, A.I. Bronnikov. Formuvannya stratehiy pryynyattya rishen' v zavdanniyakh robototekhniky // Materialy V Mizhnarodnoyi naukovo-tekhnichnoyi Internet-konferentsiyi Suchasni metody, informatsiyne, prohramne ta tekhnichne zabezpechennya system keruvannya orhanizatsiyno-tekhnichnymy ta tekhnolohichnymy kompleksamy», 22 lystopada 2018 S. 213-214.

30. Tsymbal O.M., Bronnikov A.I. Internet robotyzovanykh rechey: ohlyad kontseptsiyi // Materialy 3-yi mizhnarodnoyi naukovo-praktychnoyi konferentsiyi «Computer and informational systems and technologies». 23-24 kvitnya 2019 r. – Kharkiv: KHNURE, 2019. – S. 95-96.

31. Nevliudov Igor, Oleksandr Tsymbal, Artem Bronnikov. The Implementation of Manufacturing Agent's Concept for Flexible Production Systems

// 1st International Scientific-Practical Conference - Modern Information, Measurement and Control Systems: Problems and Perspectives 2019, 1-2 July 2019, Baku. – P. 187.

32. Vladyslav Yevsieiev, Artem Bronnikov, Analysis of architectural models for representing the integration of cyber-physical production systems hierarchical levels // Manufacturing & Mechatronic Systems 2020: Proceedings of IVth International Conference, Kharkiv, October 22-23, 2020: Theses of Reports / [Ed. I.Sh. Nevlyudov (chief editor).] .- Kharkiv .: [electronic version], 2020. - 146 p, pp:17-19.

33. Vladyslav Yevsieiev, Artem Bronnikov Analysis of the cyber-physical production systems implementation impact to achieve the goals of lean production //The II th International scientific and practical conference «Development of scientific and practical approaches in the era of globalization» (September 28-30, 2020). Boston, USA 2020. p.241. pp. 221-226. ISBN - 978-1-64945-867-4. (DOI: 10.46299/ISG.2020.II.II) (Google Scholar, Crossref)

34. Yevsieiev V., Bronnikov A. Analysis of the CMMI model application for solving the tasks of CPPS control processes automation development // The IV th International scientific and practical conference «Actual Trends of Modern Scientific Research» (11-13 October 2020) MDPC Publishing, Munich, Germany, S.128- 132. ISBN 978-3-954753-02-4

35. Yevsieiev V., Bronnikov A. Information systems development methodologies application analysis for cyber-physical production systems development // III International scientific-practical conference “Theory, science and practice” (October 05-08, 2020), Tokyo, Japan, S. 399-401, 2020. ISBN - 978-1-64945-868-1 (DOI: 10.46299/ISG.2020.II.III) (Google Scholar, Crossref)

36. Yevsieiev V., Bronnikov A. Analysis of the multi-agent systems application to solve the problem of cyberphysical production systems development // The IV th International scientific and practical conference «Integration of scientific bases into practice», (2020), Stockholm, Sweden, pp. 459 – 462. ISBN - 978-1-64945-864-3 (DOI:10.46299/ISG.2020.IV) (Google Scholar, Crossref)

37. Yevsieiev V., Bronnikov A. Structural model of a cyber-physical production system based on multi-agent systems analysis // Materialy VII Mizhnarodnoyi naukovo-tekhnichnoyi Internet-konferentsiyi «Suchasni metody, informatsiyne, prohramne ta tekhnichne zabezpechennya system keruvannya orhanizatsiyno-tekhnichnymy ta tekhnolohichnymy kompleksamy», 26 lystopada 2020. [Elektronnyy resurs] – K: NUKHT, 2020 . S. 312-313.

38. Yevsieiev V., Bronnikov A. (2020) Comlexity development analysis of cyber-physical production systems for smart manufacturing // The X<sup>th</sup> International scientific and practical conference «Trends in the development of modern scientific thought» November 23-26, 2020 Vancouver, Canada. S. 699-703 (ISBN - 978-1-63649-921-5) (DOI - 10.46299/ISG.2020.II.X) (Google Scholar, Crossref)

## ЗМІСТ

Перелік скорочень .....	22
Вступ.....	23
Розділ 1 Аналіз методів інтелектуального керування роботами.....	31
1.1 Аналіз сучасних гнучких інтегрованих систем .....	31
1.2 Адаптивне керування в робототехніці.....	44
1.3 Інтелектуальні системи керування в робототехніці .....	49
1.4 Аналіз моделей та методів адаптивного візуального керування .....	63
1.5 Висновки за розділом 1 та постановка завдань дисертаційного дослідження .....	67
Розділ 2 Моделювання робочого простору та функціональних зв'язків гнучких інтегрованих виробничих систем .....	71
2.1 Інформаційно-логічна модель робочого простору .....	71
2.2 Функціональна модель гнучкої інтегрованої системи.....	81
2.3 Висновки до розділу 2 .....	92
Розділ 3 Розробка моделей та методу адаптивного візуального керування мобільними транспортувальними роботами в ГІВС .....	93
3.1 Визначення основних вимог до моделей адаптивного візуального керування мобільним роботом .....	93
3.2 Модель розпізнавання та ідентифікації об'єктів .....	109
3.3 Модель керування мобільним роботом у просторі гнучкої інтегрованої системи на основі інформації об'єктової системи комп'ютерного зору .....	113
3.4 Метод адаптивного візуального керування роботом .....	121
3.5 Модель нечіткого адаптивного керування роботом.....	124
3.6 Висновки за розділом 3 .....	135
Розділ 4 Моделювання адаптивних систем керування засобами GNU Octave.....	136
4.1 Знаходження передавальних функцій двигуна.....	136

	21
4.2 Моделювання адаптивного режиму роботи одного двигуна .....	139
4.3 Моделювання адаптивного режиму роботи двох двигунів .....	144
4.4 Моделювання розрахунку швидкостей та крутного моменту переміщення робототехнічної платформи .....	147
4.5 Моделювання нечіткого адаптивного керування роботом.....	149
4.6 Висновки за розділом 4 .....	156
Розділ 5 Розробка програмного забезпечення для адаптивного візуального керування роботами .....	158
5.1 Розробка програмного забезпечення для розрахунку швидкостей робота з використанням GNU Octave .....	158
5.2 Розробка програмного забезпечення для розпізнавання об'єктів робочого простору .....	159
5.3 Розробка програмного забезпечення для розпізнавання об'єктів робочого простору методом SIFT .....	164
5.4 Розробка програмного забезпечення для візуального контролю кута повороту об'єкту на зображенні.....	166
5.5 Розробка програмного забезпечення для адаптивного візуального керування мобільним роботом .....	168
5.6 Висновки за розділом 5 .....	178
Висновки .....	179
Перелік джерел посилання .....	182
Додаток А Список публікацій.....	195
Додаток Б Код програми, що реалізує запропоновані моделі та метод адаптивного візуального керування мобільними роботами .....	202
Додаток В_Акти про впровадження результатів дисертаційної роботи.....	219

## ПЕРЕЛІК СКОРОЧЕНЬ

ГІВС – гнучка інтегрована виробнича система,  
ККД – коефіцієнт корисної дії,  
КФС – кібер-фізичні системи,  
КФВС – кібер-фізичні виробничі системи,  
ЛСКЗ – локальна система комп'ютерного зору,  
МР – мобільний робот,  
ОСКЗ – об'єктова система комп'ютерного зору,  
ОЦ – обчислювальні центри,  
ПІД-регулятор – пропорційно-інтегрально-диференційний регулятор,  
ПЛК – програмований логічний контролер,  
РП – робочий простір,  
СД – складальні ділянки,  
СКЗ – система комп'ютерного зору,  
СНС – самоналагоджувальна система,  
СППР – системи підтримки прийняття рішень,  
ЧПК – числове програмне керування,  
ШІ – штучний інтелект,  
ШНМ – штучна нейронна мережа,  
AGV – Automated Guided Vehicle (автоматизованих керований пристрій, мобільний робот),  
SLAM – Simultaneous Localization And Mapping (одночасна локалізація і відображення).

## ВСТУП

**Актуальність теми.** Сучасні гнучкі інтегровані виробничі системи (ГВС), порівняно з виробництвом попередніх періодів, характеризуються, з одного боку здатністю швидкого переналаштування і перепрограмування до випуску нових типів продукції, з іншого складаються з більш складних, і, відповідно, більш коштовних компонентів програмно-апаратного забезпечення. Введення роботів до складу ГВС відкриває нові можливості у вигляді часткової або повної автоматизації певних типів робіт, особливо таких, що є важкими або небезпечними для виконання людиною. Ступінь складності виконання роботами завдань ГВС суттєво залежить від технічних характеристик транспортувальних та маніпуляційних систем роботів, здатності сприймати інформацію про стан робочого простору ГВС, наявності баз знань роботизованих систем, зокрема, у вигляді сценаріїв поведінки робота, що відображають специфіку виконання практичних завдань. При цьому, отримання роботом інформації про стан робочого простору забезпечується сенсорними системами, що складаються з датчиків, які створено на основі різних фізичних принципів і орієнтованих на конкретні умови експлуатації.

Використання зорової інформації в системах керування роботів практично привертає найбільшу увагу розробників та дослідників за аналогією з зором людини, особливо з точки зору розв'язання завдань розпізнавання та ідентифікації. Проблемою створення адаптивних систем керування присвячені наукові роботи таких вчених як: A. Pugh, J. Canny, D. Forsyth, J. Ponce, S. Thrun, R. Lozano, K. Åström, A. Тимофєєв, Є. Юревич, А. Ющенко, А. Ладанюк, В. Лисенко, А. Александров, Є. Бодянський, І. Невлюдов, О. Филипенко, А. Купін, В. Філатов та ін. Проте завдання інтеграції засобів комп'ютерного та технічного зору у системи керування, візуальної корекції руху мобільних платформ та маніпуляторів, адаптації роботи засобів керування на основі

візуальної інформації є до сих пір такими, що не отримали свого повного розв'язання. Очевидним є протиріччя між сучасним рівнем розробки технології і методів комп'ютерного зору та рівнем їх впровадження у системи інтелектуального керування мобільними та маніпуляційними роботами. Таким чином, тематика дисертаційного дослідження, спрямованого на розробку моделей та методів адаптивного візуального керування мобільними роботизованими засобами, є своєчасною та актуальною.

**Зв'язок роботи з науковими програмами, планами, темами.**

Дисертаційна робота за своїм спрямуванням і змістом відповідає паспорту спеціальності 05.13.07 – автоматизація процесів керування.

Дисертаційна робота виконана відповідно до плану науково-дослідних робіт кафедри Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки (КІТАМ) Харківського національного університету радіоелектроніки в рамках НДР за держбюджетними темами: «Створення мікрооптоелектромеханічних засобів для інтелектуальних технологічних систем промислового обладнання та робототехніки» (ДР №0115U002433), «Безскладальні гнучко-жорсткі конструкції зі змінною конфігурацією для мікросистемної техніки та інтелектуальних роботів» (ДР №0117U002529), «Інтелектуальна багатоцільова робототехнічна платформа з удосконаленими маніпуляційними можливостями» (ДР №0121U109909). Здобувачем, в якості виконавця, розроблені метод та окремі моделі адаптивного візуального керування, що застосовуються у інтелектуальних системах керування роботів, відповідне інформаційне та програмне забезпечення.

**Мета роботи і завдання досліджень.** Метою дисертаційної роботи є розробка нових та удосконалення існуючих моделей та методів адаптивного візуального керування, що забезпечують підвищення продуктивності, швидкодії та надійності робототехнічних засобів гнучких інтегрованих виробничих систем.

Для досягнення мети необхідно вирішити такі завдання:



- дослідити сучасний стан гнучких інтегрованих роботизованих систем і застосування засобів адаптивного та інтелектуального керування;
- розробити інформаційно-логічну модель подання інформації про робочий простір ГІВС;
- розробити функціональну модель, що забезпечує опис взаємодії об'єктів ГІВС під час виконання виробничих завдань;
- вдосконалити метод адаптивного візуального керування;
- вдосконалити моделі керування мобільним роботом у просторі ГІВС на основі впровадження візуального цілевказання та нечіткого опису параметрів керування, модель розпізнавання та ідентифікації об'єктів;
- практично реалізувати розроблені моделі і метод, розробити інформаційне та програмне забезпечення, що забезпечує адаптивне візуальне керування мобільними роботами, які функціонують у складі ГІВС.

**Об'єкт дослідження** – процеси керування гнучким інтегрованим роботизованим виробництвом.

**Предмет дослідження** – моделі і методи автоматизованого керування мобільними транспортними роботами у складі гнучкого інтегрованого виробництва.

**Методи дослідження.** Для вирішення поставлених завдань використані методи теорії множин – для розробки декларативної моделі робочого простору ГІВС, функціональної моделі ГІВС, а також для удосконалення моделей розпізнавання та ідентифікації об'єктів, керування мобільним роботом у просторі ГІВС; методи теорії автоматичного керування та нечітких множин – для моделювання роботи адаптивної системи керування роботом; методи теорії обробки зображень – для програмної реалізації завдань, що входять у склад системи адаптивного візуального керування. При проведенні експериментальних досліджень, для практичної перевірки результатів і розробки програмного забезпечення використані мови програмування C/C++, Python, програмні бібліотеки MFC, OpenGL, OpenCV, NumPy, середовища

розробки Microsoft Visual Studio, PyCharm та пакет прикладних програм GNU Octave для моделювання роботи системи адаптивного візуального керування.

**Наукова новизна отриманих результатів.** До нових, отриманих особисто автором, слід віднести наступні результати:

– вперше розроблено інформаційно-логічну модель робочого простору, що відображає об'єкти ГІВС, характерні для завдань інтелектуального керування роботами та зв'язки між об'єктами, і яка, на відміну від існуючих враховує дискретність та заповненість виробничого робочого простору; ґрунтується на інформації, отриманій від об'єктової системи комп'ютерного зору; визначає та враховує властивості об'єктів, що розміщуються в робочому просторі; враховує взаємодію, упорядкованість та сумісність об'єктів;

– вперше розроблено функціональну модель ГІВС, яка забезпечує опис взаємодії об'єктів на рівні виконання поставлених перед ГІВС виробничих, транспортувальних, сервісних, моніторингових, управлінських та інших завдань, та, на відміну від існуючих, ґрунтується на інформаційно-логічній моделі робочого простору, враховує особливості технологічних операцій, містить детальний опис властивостей об'єктів та їх взаємодії, включає можливість кількісної оцінки вартості та якості робіт;;

– вдосконалено метод адаптивного візуального керування, що описує сукупність прийомів та операцій, які слід застосувати у інтелектуальних системах керування роботів для планування завдання переміщень на основі отримання зорової інформації від систем комп'ютерного (технічного) зору, та, на відміну від існуючих методів, орієнтований на використання в мобільних робототехнічних системах;

– отримала подальший розвиток модель керування мобільним роботом у просторі ГІВС, що здійснює керування шасі двоколісного робота, та, на відміну від існуючих, функціонує на основі інформації від об'єктових/локальних систем комп'ютерного зору та виконання функцій розпізнавання та ідентифікації об'єктів ГІВС, забезпечує розрахунок

координат та швидкостей роботизованої платформи та інших об'єктів ГВС, розрахунок лінійних та криволінійних траєкторій переміщень у дискретному робочому просторі, подання маршруту у вигляді послідовностей переміщень, забезпечує організацію керування транспортувальними роботами за їх візуальним положенням; нечітка складова моделі містить розгорнутий опис процедур формування лінгвістичних змінних, пов'язаних з динамікою руху мобільної роботизованої платформи та формалізує набір продукцій нечіткого регулятора.

### **Практичне значення отриманих результатів дисертації.**

Розроблено інформаційне та програмне забезпечення адаптивної візуальної системи керування мобільним транспортувальним роботом, яке реалізує запропоновані моделі і метод, практично забезпечує аналіз робочого простору за допомогою системи комп'ютерного/технічного зору, розпізнавання та ідентифікацію об'єктів робочого простору та робототехнічних засобів, отримання їх просторових координат, підтримку візуального цілевказання маршруту мобільного робота та функціонування інтелектуальної системи підтримки прийняття рішень.

Результати досліджень можуть бути використані в дослідженнях та розробках, що виконуються у ДП «Науково-дослідний технологічний інститут приладобудування» (м. Харків), АТ «Хартрон» (м. Харків), ПАТ «ФЕД» (м. Харків), ПАВ «Вертикаль» (м. Харків), ТОВ «Укрінтех» (м. Харків) та інших організаціях та підприємствах, що вирішують проблеми створення гнучких інтегрованих систем, автоматизованих та роботизованих пристроїв різного призначення.

Результати розробок і досліджень впроваджені у Харківському національному університеті радіоелектроніки у навчальний процес на кафедрі комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології за дисциплінами Комп'ютерні системи керування рухомими об'єктами та Автоматизовані системи керування роботизованим виробництвом (акт

від 6 лютого 2021 року), у наукову діяльність у ННЦ «Інститут судових експертиз ім. Засл. проф. М.С. Бокаріуса» (акт від 26 червня 2020 року).

**Особистий внесок здобувача.** Наукові результати дисертаційної роботи сформовані і одержані автором особисто. Всі основні результати опубліковані в роботах [63-100]. Роботи [69-70], [75] опубліковані одноосібно. В роботах, виконаних у співавторстві, особистий внесок здобувача полягає в наступному: [63] – визначено основні властивості візуального цілевказання, визначено основний метод зшивки зображень для багатозонової системи комп'ютерного зору, визначено основні алгоритми перетворення координат; [64] – розроблено математичне, алгоритмічне та програмне забезпечення для інформаційно-логічної моделі робочого простору; [65] – вдосконалено структурну схему адаптивного візуального керування; [66] – сформульовано основні вимоги до моделі та методу адаптивного візуального керування; [67] – запропоновано інформаційно-логічну модель агентно-орієнтованого виробництва; [68] – розроблено програмні методи ідентифікації та розпізнавання виробничого агента в робочому просторі гнучкої інтегрованої системи; [71] – сформульовано основні вимоги до побудови функціональної моделі ГІВС для виробничого роботизованого агента; [72] – визначено окремі складові функціональної моделі ГІВС для проектування мехатронних систем; [73] – вдосконалено метод адаптивного керування мобільною платформою та проведено комп'ютерне моделювання; [74] – удосконалено модель керування мобільним роботом у просторі ГІВС; [76] – вдосконалені інформаційно-логічна модель та модель адаптивного керування мобільним роботом у просторі гнучкої інтегрованої системи, в тому числі на основі нечіткого підходу; [77] – вдосконалено інформаційно-логічну модель робочого простору за рахунок використання концепції інтернету роботизованих речей; [78] – вдосконалено модель розпізнавання та ідентифікації об'єктів у робочому просторі за рахунок впровадження програмних методів визначення рухів динамічних об'єктів; [79] – запропоновано використання інформаційно-логічної моделі для кібер-фізичного виробництва; [80] – вдосконалюється

інформаційно-логічна модель у складі інтелектуальної системи управління роботом, [81] – запропоновано використання інформаційно-логічної моделі для завдань автоматизованого проектування. Усі співавтори із задекларованим особистим внеском Броннікова А.І. згодні. Задекларований особистий внесок здобувача в роботах, виконаних у співавторстві, відповідає темі та змістові дисертації.

**Апробація матеріалів дисертації.** Основні теоретичні і практичні результати дисертаційної роботи доповідалися і отримали схвалення на таких конференціях: 8th International Conference on Advanced Optoelectronics and Lasers, CAOL\*2019, DSMOLE\*2019 (Созополь, Болгарія); Proceedings of 2020 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (Virtual Meeting, Італія); Міжнародній науково-практичній конференції «Автоматизация: проблемы, идеи, решения» (Севастополь, 2010); Першій Всеукраїнській науково-практичній конференції «Актуальні проблеми створення електронних засобів промислових автоматизованих систем» (Севєродонецьк, 2011); Міжнародній науково-технічній конференції «Системний аналіз та інформаційні технології» (Київ, 2011); Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2012) (Харків, 2012); 5-ій міжнародній науково-технічній конференції «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів керування» (Харків, 2016); першій міжнародній науково-технічній конференції «Computer and information systems and technologies» (Харків, 2017); V Міжнародній науково-технічній Інтернет-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами» (Київ, 2018); 3-ій міжнародній науково-практичній конференції «Computer and informational systems and technologies» (Харків, 2019); 1-ій міжнародній науково-технічній конференції «Modern Information, Measurement and Control Systems: Problems and Perspectives» (Баку, 2019); 4-ій міжнародній конференції «Manufacturing & Mechatronic Systems» (Харків, 2020); міжнародній науково-технічній конференції «Development of

scientific and practical approaches in the era of globalization» (Бостон, 2020); міжнародній науково-технічній конференції «Actual Trends of Modern Scientific Research» (Мюнхен, 2020); III International scientific-practical conference «Theory, science and practice» (Токіо, 2020); IV th International scientific and practical conference «Integration of scientific bases into practice» (Стокгольм, 2020); Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами» (Київ, 2020); The X th International scientific and practical conference «Trends in the development of modern scientific thought» (Ванкувер, Канада, 2020).

**Публікації.** За результатами дисертаційних досліджень опубліковано 38 наукових праць, з них 19 статей, у тому числі – 1 стаття у науковому фаховому виданні у Турецькій республіці (входить у Організацію економічного співробітництва та розвитку (ОЕСР)); 16 статей в наукових фахових виданнях України з технічних наук; 2 статті у наукових виданнях інших держав; 19 тез доповідей, що опубліковані в матеріалах міжнародних наукових конференцій, з яких 2 включено до наукометричної бази даних Scopus.

**Структура і обсяг дисертації.** Дисертація складається зі вступу, п'яти розділів, висновків, що містять основні результати роботи, списку використаних джерел і додатків. Загальний обсяг дисертації складає 222 сторінки, що включає 180 сторінок основного тексту, 77 рисунків (з них 60 рисунків займають площу на 30 сторінках), 6 таблиць, список використаних джерел зі 107 найменувань (на 13 сторінках) та 3 додатки (на 27 сторінках).

## РОЗДІЛ 1

### АНАЛІЗ МЕТОДІВ ІНТЕЛЕКТУАЛЬНОГО КЕРУВАННЯ РОБОТАМИ

#### 1.1 Аналіз сучасних гнучких інтегрованих систем

##### 1.1.1 Аналіз тенденцій розвитку сучасного виробництва

Завданням керування сучасною виробничою системою називають переведення її з одного стану в інший, що характеризується збільшенням стійкості та ефективності [1].

Аналіз розвитку виробничих систем як класу складних систем, демонструє еволюцію систем від простих до складних (рис.1.1) [2]. З одного боку, чим вище складність таких систем, тим вище їх ефективність. З іншого боку, зростання різноманітності систем призводить до збільшення факторів невизначеності виробництва.

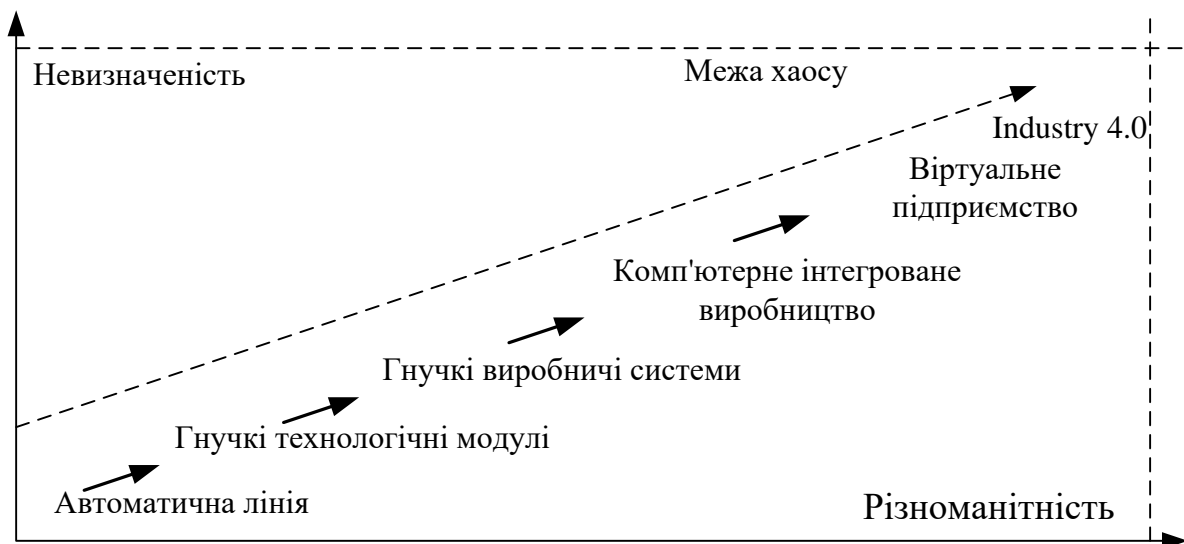


Рисунок 1.1 – Еволюція гнучких виробничих систем

Модернізація гнучких виробничих систем збільшує число невизначених ситуацій і появу непередбачених ситуацій, що можуть призвести до стану

хаосу в системі. Вирішити ці проблеми мають допомогти автоматизовані інтегровані виробничі системи.

Під гнучкістю системи розуміють її здатність швидко перебудовуватися на виконання нових завдань, що визначаються технічними можливостями обладнання та технологією обробки [3]. Перевагами гнучкості виробництва є такі властивості системи, як:

- відповідність вимогам виробництва;
- швидкий перехід від випуску одного виду продукції до іншого;
- збереження дрібносерійного виробництва;
- автоматизація виробництва за рахунок комп'ютеризації;
- зниження витрат на незавершене виробництво.

Гнучке інтегроване виробництво має володіти такими ознаками [4]:

- здатність коректно функціонувати при різних зовнішніх і внутрішніх змінах;
- забезпечення можливості легко включати в систему нове обладнання для збільшення потужностей у зв'язку зі збільшенням обсягу виробництва;
- можливість розширення обробки різного виду деталей;
- здатність системи враховувати зміни послідовностей виконання технологічних операцій;
- здатність системи функціонувати при переналаштуванні в верстатах;
- здатність до безперебійного і оптимального завантаження металорізального обладнання до певної стратегії керування;
- здатність системи до гнучкої зміни інструменту;
- раціональна побудова маршрутів обробки і транспортувальних потоків, відповідно до критеріїв оптимальності;
- можливість простого і негайного переходу на обробку будь-якої з освоєних системою деталей.

Гнучка інтегрована виробнича система (ГІВС) у наш час є високоавтоматизованою і інтегрованою виробничою системою, в якій



використовують набір промислових роботів, багатофункціональних верстатів з числовим програмним керуванням, автоматизовані системи обробки матеріалів і інтелектуальні склади. ГІВС працює під керуванням промислових комп'ютерів [5]. На відміну від системи масового виробництва, яка в основному орієнтована на економію на масштабі, у основі використання ГІВС закладено баланс між продуктивністю системи і її гнучкістю за рахунок застосування комп'ютерних технологій для досягнення високого рівня конкурентоспроможності на поточному ринку з швидкими змінами, тобто використання такого поняття як Industry 4.0.

Авторами Сетхі і Сетхі [6] було додано критерії гнучкості матеріалів, програмного забезпечення і ринкової гнучкості до раніше існуючих.

Концепція ГІВС полягає в тому, що вона об'єднує ідеологію одиничного виробництва і виробничу систему серійного виробництва, тобто за допомогою ГІВС одночасно можуть вироблятися різні продукти з різним розміром партії [7].

Інтеграція інтелектуальних верстатів і промислових роботів в ГІВС, а також розробка і використання нового програмного забезпечення для моделювання, моніторингу та керування виробничими процесами вимагають глибоких перетворень в галузі [8-10]. З огляду на складність цих систем, спрямованих на збільшення виробничих потужностей в контексті диверсифікації виробництва, скорочення життєвого циклу продуктів і підвищення конкурентоспроможності, стратегічне рішення про впровадження ГІВС в інтелектуальну виробничу систему в даний час є важливою метою [11-14].

Кібер-фізичні системи (КФС) – це системи взаємодіючих обчислювальних об'єктів, які знаходяться в інтенсивному зв'язку з навколишнім фізичним світом і його поточними процесами, надаючи і одночасно використовуючи послуги доступу до даних і обробки даних [15]. Іншими словами, КФС можна в цілому охарактеризувати як фізичні та

інженерні системи, операції яких відслідковуються, контролюються, координуються і інтегруються обчислювальним і комунікаційним ядром.

Взаємодія між фізичними і кібер-елементами у цьому випадку має ключове значення.

Для виробництва було введено поняття кібер-фізичних виробничих систем (КФВС), що складаються з автономних і кооперативних елементів і підсистем, які пов'язані в залежності від контексту всередині і на всіх рівнях виробництва, від процесів через машини до виробничих і логістичних мереж. Слід виділити три основні характеристики КФВС:

- наявність інтелекту, тобто елементи можуть отримувати інформацію зі свого оточення і діяти автономно;
- можливості підключення, тобто здатність встановлювати і використовувати зв'язки з іншими елементами системи, включаючи людей, для співпраці та співробітництва, а також зі знаннями і послугами, доступними в Інтернеті;
- чутливість до внутрішніх і зовнішніх змін.

Моделювання роботи, а також прогнозування нової поведінки цих систем піднімають ряд основних і орієнтованих на додатки дослідницьких завдань, не кажучи вже про контроль будь-якого рівня цих систем. Фундаментальне питання полягає в тому, як досліджувати такі фактори автономності, співпраця, оптимізація та оперативність.

Очікується, що інтеграція аналітичних підходів і підходів на основі моделювання стане більш значущою, ніж будь-коли. Все більш вагомими постають проблеми експлуатації сенсорних мереж, обробки великих обсягів і швидкостей даних, а також питання пошуку інформації, подання та інтерпретація з особливим наголосом на аспекти безпеки. Нові способи людино-машинного спілкування повинні бути досягнуті в процесі створення КФВС.

З використанням КФВС з'являються нові методи: одночасно з існуванням типових рівнів керування на основі звичайних ПЛК, на інших,

більш високих рівнях ієрархії виробничих систем вводиться більш децентралізований спосіб функціонування, характерний для КФВС.

КФВС є складним організаційно-технічним об'єктом в яких закладено використання концепції Digital Twins у реальному виробництві, тому розробка подібного роду систем є актуальною задачею. Для цього потрібно забезпечити такі властивості як контроль цілісності переданих даних, повнота їх представлення, зв'язки між змінними і контейнерами рішень, які містять фрагменти програмного коду; гнучкість структури даних для адаптивності і масштабованості системи [16].

Архітектура 5C (Configuration, Cognition, Cyber, Conversion, Connection), представлена на рис 1.2, складається з 5 рівнів виробничих процесів і ілюструє, як за допомогою аналітики будується КФВС від початкового збору даних до кінцевого створення продукту [17], [94-100]. У підході КФВС, рівень інтелектуального підключення (рівень I) представляє фізичний простір, рівні II-IV – кіберпростір, а рівень конфігурації (рівень V) реалізує зворотний зв'язок від кіберпростору в фізичний простір.



Рисунок 1.2 – Архітектура 5C кібер-фізичних виробничих систем

Отже, на сучасному виробництві впровадження кібер-фізичних систем стає одним з ключових факторів для досягнення більших результатів.

### 1.1.2 Аналіз реальних виробничих систем в галузі машино- та приладобудування і радіоелектронного виробництва

Сучасне виробництво в наш час включає в себе виробниче обладнання, гнучкі автоматизовані лінії і ділянки (рис. 1.3–1.4), верстати з ЧПК, інструменти, системи моніторингу, тощо:



Рисунок 1.3 – Виробнича ділянка компанії Jabil



Рисунок 1.4 – Транспортвальна система компанії Jabil

Хоча більшість з приведених підприємств і мають у своєму складі сучасне обладнання та маніпуляційних роботів, але одночасно, частина обладнання має транспортувальне обслуговування за допомогою ручних засобів (рис. 1.5–1.6).



Рисунок 1.5 – Ділянка збірки автомобілей



Рисунок 1.6 – Транспортувально-складська система виробництва автомобілей



Поряд з цим існують компанії, що використовують у своїх виробничих потужностях для транспортувальних операцій мобільних роботів – так звані AGV (automated guided vehicle, автоматизований керований транспортувальний засіб). Серед таких компаній є Amazon, Airbus, Skoda, Volkswagen, Ford, Shell, Henkel, FIAT, Bosch та ін.

Зовнішній вигляд цих засобів представлено на рис. 1.7.



Рисунок 1.7 – AGV Omron

Провівши критичний аналіз виробництва таких сучасних підприємств як Dacia-Renault, Toyota, McLaren, Mercedes, заводів FEIDAL та Атлантик, виробничих кампаній LG та Jabil можна зробити висновки про те, що їх транспортувальні система ще й досі знаходиться на недостатньому рівні автоматизації, оскільки на операціях перевезення вантажів використовується ручна праця, а засоби роботизації мають обмежене застосування. Водночас, є компанії, що використовують мобільні роботи у своїх виробничих приміщеннях для транспортувальних операцій, і, як наслідок, мають більшу швидкодію автоматизованих систем.

### 1.1.3 Технології навігації AGV

Технології навігації AGV – це лазерна навігація, магнітна навігація, природна або вільна навігація (включаючи SLAM-навігацію з датчиками

LiDAR), магнітну точкову навігацію, індуктивну дротову навігацію, оптичну навігацію, візуальну навігацію [18].

Вибір правильної технології наведення транспортувального засобу дуже важливий, тому що він вплине на роботу автоматизованої системи керування транспортувальним засобом. Характеристики AGV складні і залежать від багатьох ключових елементів.

Найбільш поширеними навігаційними системами AGV є:

- лазерна навігація;
- магнітна навігація;
- AGV з природною навігацією (SLAM або LiDAR navigation);
- магнітна точкова навігація;
- проводова навігація;
- оптична навігація;
- навігації за допомогою комп'ютерного зору.

Засіб з лазерним наведенням (навігацією) – це AGV, що переміщується за допомогою лазерної системи позиціонування [19].

Кожний транспортувальний засіб з лазерним наведенням оснащено двовимірним навігаційним лазером (навігаційним пристроєм або лазерним випромінювачем).

Навігаційний лазер випромінює безперервний сигнал модульованого лазерного світла на 360 градусів, взаємодіючи з відбивачами, розташованими в робочій зоні AGV. Залежно від виробника пристрою такі засоби обчислюють і коректують своє місце розташування від 30 до 40 разів на секунду. Ці пристрої неймовірно точні.

Відбивачі відправляють на навігаційний пристрій безперервне відображення. Кут повороту лазера визначає опорну радіальну координату, яка дозволяє обчислити координати X, Y відбивачів.

Центр відбивачів повинен бути розташований на тій же висоті навігаційного пристрою і на відстані не більше 30 метрів від AGV.

Магнітна навігація полягає в наступному. Маршрут з інструкціями становить магнітна стрічка, яка наклеюється на поверхню підлоги [20].

Магнітний датчик AGV визначає магнітне поле стрічки і приводить в рух AGV по шляху.

Магнітна стрічка проста в установці. Для укладання магнітних доріжок використовується клей з високою адгезією. Стандартні розміри – 1 мм в товщину і 5 см в ширину, тому стрічка не є інвазивною.

Система керування AGV може вказувати, як і куди рухатися, за допомогою міток (RFID) або магнітних індикаторів, розміщених на підлозі.

У цих заздалегідь визначених положеннях AGV може вирішити, що робити: повернути ліворуч, зупинитися, задіяти візок тощо.

Природня навігація. Найбільш важливою є SLAM-навігація або одночасна локалізація і відображення (SLAM). Це просто означає, що AGV з SLAM Navigation може відображати своє оточення і визначати його місцезнаходження завдяки інформації, отриманої з навколишнього середовища [21].

AGV можуть відображати навколишнє середовище за допомогою різних датчиків AGV, таких як камери технічного зору, лідарні датчики або навіть ті ж лазери, які використовуються в цілях безпеки – вся ця інформація об'єднується з внутрішнім інерціальним вимірювальним блоком для визначення і перерахунку реальне позиціонування AGV.

Всі ці обчислення проводяться дуже складним алгоритмом SLAM.

SLAM Navigation замінить інші види навігації, такі як магнітна навігація, оптична навігація і тощо.

Ця технологія є найкращим рішенням для тягачів. Велика кількість основних виробників AGV розробляють і включають цю технологію в свої AGV. Основне занепокоєння з приводу технології природньої навігації (SLAM, LIDAR тощо) полягає в її надійності в різних середовищах, таких як виробничі лінії, де постійно переміщаються люди, предмети, ящики, піддони тощо.



У цих умовах AGV може не знайти, де він знаходиться. З цієї причини SLAM з LiDAR – відмінне рішення для AGV, де у вас є чітко визначені профілі і середовища з фіксованими конструкціями, такими як стіни і колони. Природню навігацію можна використовувати на складах і в лікарнях.

Магнітна точкова навігація. AGV можуть переміщатися за невеликими циліндричними магнітними точками [22], вбудованими в підлогу. Магнітні точки зазвичай являють собою циліндричні магніти з розмірами близько 20x10 мм. Вони встановлюються кожні 250-500 мм, створюючи віртуальний шлях.

AGV переміщається з одного місця в інше, використовуючи датчики і керуючі, такі як датчики на ефекті Холла, енкодери, лічильники, гіроскопічні датчики і інші типи енкодерів для калібрування з урахуванням помилок кута повороту.

AGV слідує кресленнями САПР, раніше завантаженим в систему керування AGV, а магнітні плями служать орієнтиром відповідно до карти.

Допуски на позиціонування відносно гарні і складають близько 2,5 мм. Цей допуск краще, ніж допуски, отримані іншими «безкоштовними» навігаційними системами, такими як SLAM, LiDAR тощо.

Установка полягає у тому, що у місці розташування магніту потрібно зробити невеликий отвір в підлозі і після залити його епоксидною смолою.

Технологія індуктивного наведення (проводова навігація) полягає у наступному [23]:

- підлогу розрізають, і закладається дріт, щоб позначити напрямну;
- датчик направляючого шляху встановлюють на автоматичному засобі;
- на підлозі позначають доріжки;
- шляхи є безперервними;
- шляхи є фіксованими, і в разі переналагодження їх нелегко змінити;
- переналагодження об'єкта не так гнучко, як деякі інші навігаційні технології, і може бути неповним через обмеження.

Оптична навігація – це гнучка навігаційна система для операцій в далекому космосі, яка не потребує вимірювань GPS [24-26]. Рішення для навігації розраховується з використанням фільтра Калмана, який може приймати будь-яку комбінацію дальності, ширини хорди планети, орієнтирів і кутів з використанням будь-якого небесного об'єкта. Використовує повну нелінійну динамічну модель орбіти, включаючи гравітаційні моделі і моделі збурень, а також в якості динамічної основи для алгоритмів визначення орієнтації. Тобто поширюється модель гіроскопа, а не модель космічного корабля. Це робить датчик більш функціональною заміною зоряного трекера, що знижує витрати на інтеграцію і тестування при додаванні цього датчика в космічний корабель. Лінійні акселерометри використовуються для вимірювання сил на космічному кораблі. Це дозволяє точно вимірювати прискорення, прикладаються підрулюючими пристроями під час маневрів. Точність навігації обмежена знанням ефемерид цілей вимірювання, але достатня для цілей маневрування на орбіті.

Навігація на основі систем комп'ютерного зору. Системи AGV в цілому складаються з комп'ютерного програмного забезпечення і технологій, які є керуючим пристроєм [27]. Розвиток комп'ютерних технологій, застосування систем технічного зору надали дуже значний вплив на розробку гнучких, ефективних і недорогих систем з низькими витратами на обслуговування, особливо в мобільному обладнанні.

З іншого боку, система AGV, в якій використовувався датчик технічного зору і комп'ютерна обробка зображень для навігації, може зробити систему більш гнучкою, обробляти більший обсяг інформації і підвищити надійність роботи системи. Наприклад, одне зображення, отримане датчиком камери на верхній частині AGV, можна використовувати для визначення орієнтиру, вивіски, орієнтирів і перешкод.

Для того, що реалізувати цей метод керування потрібно, по-перше, визначити параметри камери, що можна використовувати. Вибір датчика

технічного/комп'ютерного зору залежить від вимог дослідження і придатності самої системи.

По-друге, це пов'язано з виявленням і розпізнаванням об'єктів, а також ліній, за якими роботи рухаються.

По-третє, це стосується виявлення і обминання перешкод. Зазвичай є дві категорії перешкод; рухомі і статичні перешкоди. Прикладами рухомих перешкод є людина і інше AGV, в той час як статичну перешкода являє собою якийсь об'єкт (коробку). Зокрема, оператору AGV необхідно усувати перешкоди, що заважають його маршруту, а іноді і вміти їх уникати, щоб запобігти зіткненню.

По-четверте, мова йде про виявлення знаків або розпізнаванні орієнтирів. Це дуже важливо, оскільки маршрути AGV не програмуються, а визначені оператором/комп'ютером. Напрямки засновані на спеціальних знаках або орієнтирах, розроблених уздовж маршрутів AGV. Ці знаки розташовані на кожному перехресті маршруту, щоб вказати, в якому напрямку рухатися.

В п'яте, питання керування навігацією, що поєднує в собі всі згадані проблеми з правильним і реалізованим методом. Звичайна навігація без перешкод відома як основна навігація, в той час як навігація для уникнення перешкод називається навігацією без зіткнень або безконфліктної навігацією. Цю навігацію можна віднести до категорії спеціальної навігації.

Наведені AGV мають складні сенсорні системи (10 і більше датчиків), але, в основному, керуючі впливи формуються на основі інформації з датчиків різного роду. Основним недоліком цих систем є їх дорожнеча.

Використання систем комп'ютерного/технічного зору для керування транспортувальними системами, у яких використовуються AGV, може значно здешевити впровадження мобільних роботів у сучасне виробництво [27].

Також всі наведені методи навігації можуть використовуватись на сучасному виробництві, але саме методи керування на основі інформації з

систем комп'ютерного/технічного зору ще не знайшли повного впровадження у виробництво.

Основними параметрами, за якими є можливість порівняти методи навігації є:

- складність встановлення;
- гнучкість до переналагодження;
- вартість встановлення;
- вартість AGV;
- надійність використання;
- точність;
- ремонтпридатність.

Порівняння кожного з методів представлено у табл. 1.1.

Як видно з табл. 1.1, навігація на основі комп'ютерного зору є однією з найкращих за всіма параметрами, що свідчить про актуальність її використання в сучасному автоматизованому виробництві.

## 1.2 Адаптивне керування в робототехніці

Як відомо, керуюча система, що автоматично визначає потрібний закон керування за допомогою аналізу поведінки об'єкта при поточному управлінні, називається адаптивною [28-30].

Адаптивні системи ділять на два класи: системи, здатні до самоорганізації і які самостійно налаштовуються [31-33].

Системи, здатні до самоорганізації, характеризуються формуванням такого алгоритму керування (його структури і параметрів), який дозволить оптимізувати систему з боку поставленої перед нею мети керування. Виникає в умовах зміни структури і параметрів об'єкта керування в залежності від режиму функціонування (нестача апіорної інформації для визначення поточного режиму).

Таблиця 1.1 – Порівняння методів навігації AGV

Методи Параметри	Лазерна	Магнітна	SLAM, LiDAR	Магнітна точкова	Проводова	Оптична	Комп'ютерний зір
Складність встановлення	Низька	Низька	Дуже низька	Низька	Велика	Велика	Дуже низька
Гнучкість до переналагодження	Середня	Середня	Дуже велика	Низька	Низька	Низька	Дуже велика
Вартість встановлення	Середня	Середня	Дуже велика	Середня	Дуже велика	Велика	Низька
Вартість AGV	Дуже Велика	Дуже Велика	Низька	Велика	Дуже велика	Велика	Низька
Надійність використання	Дуже велика	Велика	Дуже велика	Низька	Велика	Велика	Велика
Точність	Дуже велика	Велика	Велика	Середня	Середня	Велика	Велика
Ремонтопридатність	Середня	Середня	Дуже велика	Низька	Велика	Велика	Дуже велика

При широкому класі можливих структур об'єкта важко сподіватися на вибір єдиною структури алгоритму керування, здатної забезпечити замкнутій системі досягнення ЦУ у всіх режимах функціонування. Таким чином, виникає необхідність в синтезі при вільній структурі регулятора, а, отже в ускладненні алгоритмів рішення і на широке впровадження таких систем в практику.

У разі, якщо структура об'єкта керування відома і постійна, а поведінка залежить від ряду невідомих параметрів, виникає необхідність в застосуванні самоналагоджувальних систем (СНС), в яких структура регулятора задана (заздалегідь обрана) і потрібно визначити лише алгоритм настройки його коефіцієнтів (алгоритм адаптації) [34].

Один з способів зменшення цих параметрів – додавання можливості самоадаптації. Самоналагоджувальна система (СНС) може автоматично змінювати себе у відповідності до змін в своєму складі. Модифікація виконується шляхом налаштування атрибутів (параметрів) або артефактів системи у відповідь на зміни в самій системі або в її середовищі [35].

Ієрархія систем, здатних до самоадаптації показана на рис. 1.8.

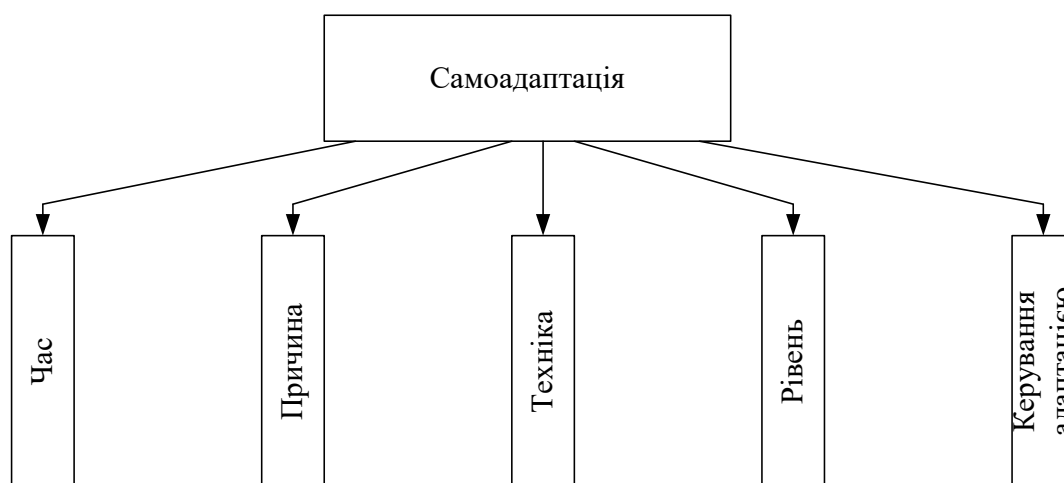


Рисунок 1.8 – Принципи самоадаптації

У системах, здатних до самоадаптації, фактором є зміна одного або декількох елементів системи:

- зміна технічних ресурсів, наприклад, дефект апаратного компонента, програмний збій або доступність альтернативного мережевого підключення;
- зміна середовища, наприклад, стан змінної контексту;
- зміна у складі користувача, наприклад зміна складу групи користувачів або уподобань користувача.

Адаптація реалізується на різних рівнях системи, зокрема на рівні об'єкта керування і логіки адаптації [36]. Логіка адаптації реалізується у блоці керування технічними ресурсами. Можна адаптувати середовище та користувачів. Адаптація логіки адаптації дозволяє поліпшити характеристики адаптації у порівнянні з адаптацією часу.

СНС діляться на два підкласи: пошукові та безпошукові [37]:

- в яких мінімальні (або максимальні) показники якості знаходяться за допомогою спеціально організованих пошукових сигналів (наприклад, екстремальні системи, в яких недолік апріорної інформації заповнюється за рахунок поточної інформації, одержуваної у вигляді реакції об'єкта на штучно вводяться пошукові дії);

- в яких є модель з бажаними динамічними характеристиками, а завдання алгоритму адаптації полягає в налаштуванні коефіцієнтів регулятора таким чином, щоб звести неузгодженість між об'єктом керування і моделлю до нуля (еталонна модель може бути реалізована у вигляді реальної динамічної ланки або бути присутньою у вигляді деякого еталонного керування, що зв'язує регульовані змінні і їх похідні.

Оптимальне керування – вид керування, який забезпечує найкращий в певному сенсі результат. У найпростішому випадку показником, що оцінюється, є стійкість (або запас стійкості), точність в усталеному режимі, ступінь зменшення впливу зовнішніх впливів, якість перехідних процесів.

Порівняльна оцінка якості керування, що включає в себе максимальну кількість окремих показників якості керування, називається критерієм оптимізації.

На рисунку 1.9 представлена структурна схема системи оптимального керування.

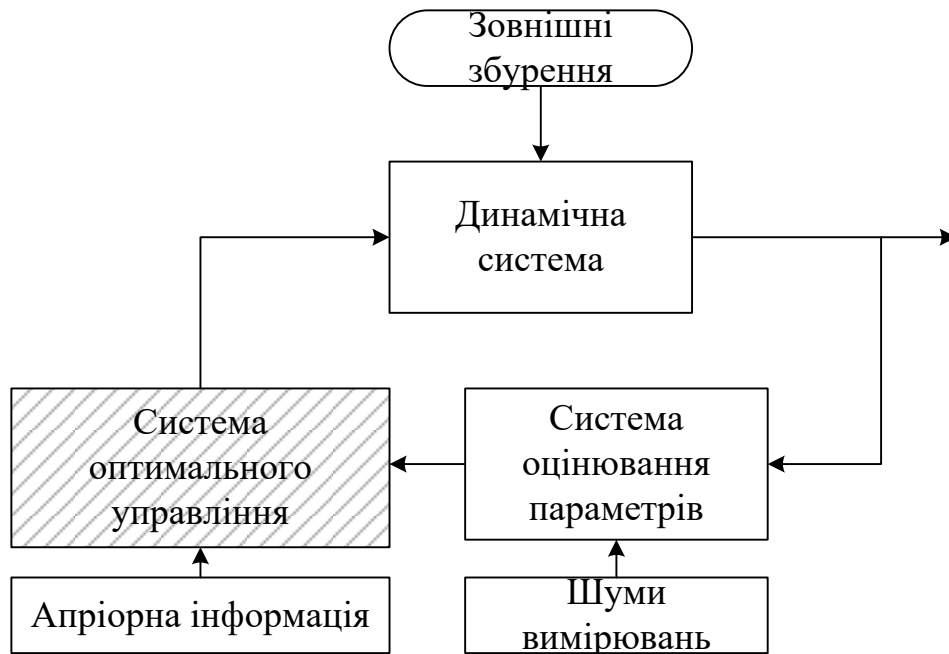


Рисунок 1.9 – Структурна схема системи оптимального керування

Блок системи оптимального керування служить для вибору оптимального режиму керування на основі вибору спеціальних критеріїв.

Прикладами критерію оптимальності для транспортувальної задачі можуть служити:

- критерій мінімуму шляху (придатний тільки для вирішення транспортувальних завдань в межах одного виду роботи);
- критерій мінімуму енергетичних витрат (сума експлуатаційних витрат, вкладень в сам засіб, вартості перевезення вантажів, різного роду витрат);
- критерій посилення пропускної спроможності ліній в залежності від розміщення на них оптимальних вантажопотоків;
- критерій суми витрат виробництва і тарифних плат за перевезення (методи динамічного програмування);
- критерій мінімуму витраченого часу.



Отже, використання адаптивного керування в сучасному автоматизованому виробництві є актуальним напрямом. Цей вид керування компенсує збурення та проводить самоналагоджування системи. Адаптивні системи дозволяють більш гнучко налаштовувати об'єкт керування та його компоненти.

### 1.3 Інтелектуальні системи керування в робототехніці

#### 1.3.1 Завдання керування двоколісним мобільним роботом

Одночасно з розв'язанням завдань прийняття рішень в гнучких інтегрованих роботизованих системах вирішуються практичні завдання керування верстатами з ЧПК, промисловими та мобільними роботами. На даний час виконано велику кількість досліджень, пов'язаних з розробкою алгоритмів керування розв'язання низки завдань мобільних роботів: уточнення карти місцевості, планування траєкторій обходу перешкод, проникнення в утруднені зони, прецизійне керування на складних криволінійних траєкторіях, тощо [38].

Автономний колісний робот є складною керованою електромеханічною системою, яка складається з ходової частини та багаторівневої системи керування рухами. Для нього характерним є одночасне здійснення механічних, пов'язаних з рухом, і інформаційних процесів, пов'язаних з обробкою сигналів вимірювальних пристроїв та формуванням керуючих сигналів. Моделлю механічної частини мобільного робота є система абсолютно твердих тіл, пов'язаних циліндричними шарнірами.

Основне тіло робота, до якого приєднані колеса, є платформою. Нехай на платформу встановлено два електродвигуни постійного струму, що забезпечують обертання ведучих коліс (рис. 1.10).

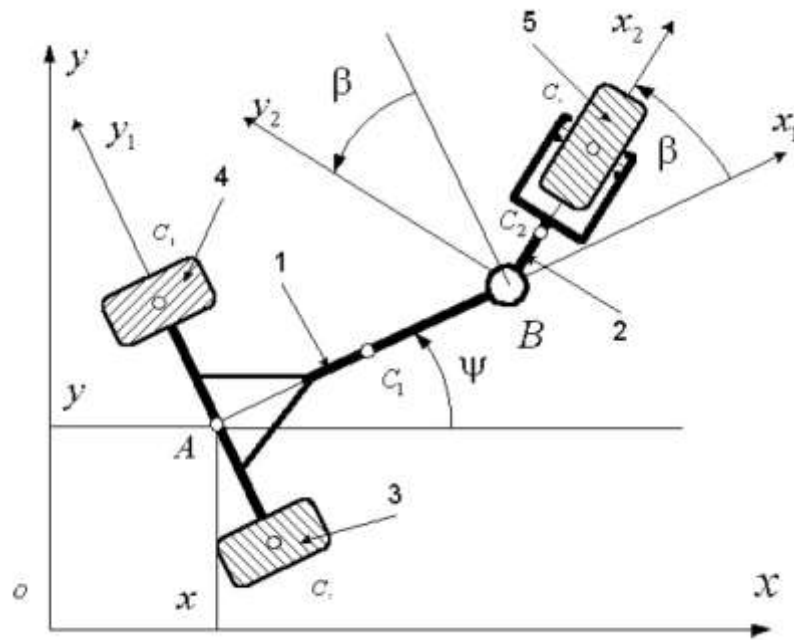


Рисунок 1.10 – Мобільний робот з двома ведучими колесами (1 – шасі, 2 – вилка стернового колеса, 3,4 – колеса, 5,6 – стернове колесо)

Для зменшення громіздкості моделі, впливом інерційності стернового колеса та його виделки на динаміку робота зневажають, а опора у точці  $B$  вважається абсолютно гладкою. Електрична підсистема містить два контури зі струмом  $i$ , таким чином, вектор узагальнених координат електромеханічної підсистеми:

$$q = |x \ y \ z \ \psi \ \phi_3 \ \phi_4 \ e_1 \ e_2|^T, \quad (1.1)$$

яка містить заряди  $e_1, e_2$ , що вводяться у рівняння, виходячи з  $i_1 = \dot{e}_1, i_2 = \dot{e}_2$  – струмів зовнішніх кіл живлення відповідних двигунів.

Матриця  $B$ , що описує рівняння неголономних зв'язків має вигляд:

$$B = \begin{bmatrix} -\sin \psi & \cos \psi & 0 & 0 & 0 & 0 & 0 \\ \cos \psi & \sin \psi & l & -r & 0 & 0 & 0 \\ \cos \psi & \sin \psi & -l & 0 & -r & 0 & 0 \end{bmatrix}. \quad (1.2)$$

Через те, що вектор узагальнених координат  $q$  містить вісім компонентів, запропонована електромеханічна система має чотири ступені свободи і для компоненти вектору псевдо швидкостей  $\dot{\pi} = (V \ \Omega \ i_1 \ i_2)^T$  зручно обирати величину  $V = \dot{x} \cos \psi + \dot{y} \sin \psi$  для швидкості точки  $A$ ,  $\Omega = \dot{\psi}$  – для кутової швидкості платформи, струми  $i_1 = \dot{e}_1, i_2 = \dot{e}_2$  – для зовнішніх кіл електродвигунів. При цьому зв'язок між узагальненими і псевдо-швидкостями визначається матрицею  $H$ :

$$H = \frac{1}{r} \begin{bmatrix} r \cos \psi & 0 & 0 & 0 \\ r \sin \psi & 0 & 0 & 0 \\ 0 & r & 0 & 0 \\ 1 & l & 0 & 0 \\ 1 & -l & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & r \end{bmatrix}, \quad \dot{x} = H \dot{\pi}. \quad (1.3)$$

Дані рівняння у скалярній формі мають вигляд:

$$\begin{aligned} \dot{x} &= V \cos \psi, & \dot{y} &= V \sin \psi, & \dot{\psi} &= \Omega, \\ \dot{\phi}_1 &= \frac{V + l\Omega}{r}, & \dot{\phi}_2 &= \frac{V - l\Omega}{r}, & \dot{e}_1 &= i_1, & \dot{e}_2 &= i_2. \end{aligned} \quad (1.4)$$

Далі на основі [38] і враховуючи, що масо-інерційні характеристики ведучих коліс є ідентичними, підрахуємо кінетичну енергію системи:

$$\begin{aligned} 2T &= (m_1 + 2m_k)(\dot{x}^2 + \dot{y}^2) - 2m_1 a(\dot{x} \sin \psi - \dot{y} \cos \psi) \dot{\psi} + \\ &+ (m_1 a^2 + J_1 + 2m_k l^2 + 2J_{kz}) \dot{\psi}^2 + J_y (\dot{\phi}_3^2 + \dot{\phi}_4^2), \end{aligned} \quad (1.5)$$

де  $m_1$  – маса платформи робота  $m_k$  – сумарна маса ведучого колеса і робота електродвигуна,  $J_1$  – момент інерції платформи відносно вертикальної осі, що

проходить крізь центр мас  $C$ ,  $J_{kz}$  – момент інерції ведучого колеса відносно вертикальної осі,  $J_y = J_{ky} + n^2$  – приведений момент інерції колеса,  $J_{ky}$  – момент інерції колеса відносно горизонтальної осі,  $J_{ry}$  – момент інерції робота електродвигуна,  $J_{ry}$  – передавальне число редуктора [38].

Магнітну енергію і дисипативну функцію задають у вигляді:

$$\begin{aligned} 2W_m &= L(\dot{e}_1^2 + \dot{e}_2^2) + 2cn(\phi_3\dot{e}_1 + \phi_4\dot{e}_2), \\ \Psi &= \frac{1}{2}R(\dot{e}_1^2 + \dot{e}_2^2), \end{aligned} \quad (1.6)$$

де  $L$  – узагальнена індуктивність кола електродвигуна,  $c$  – коефіцієнт електромагнітної взаємодії,  $R$  – опір кола робота, сили тертя не враховуються.

Потенційна та електрична енергія рівні 0, тому ланганжіан електромеханічної системи, що розглядається, визначається сумою кінетичної та магнітної енергій:

$$\begin{aligned} 2L &= (m_1 + 2m_k)(\dot{x}^2 + \dot{y}^2) - 2m_1a(\dot{x}\sin\psi - \dot{y}\cos\psi)\dot{\psi} + J_y(\dot{\phi}_3^2 + \dot{\phi}_4^2) + \\ &+ (m_1a^2 + J_1 + 2m_kl^2 + 2J_{kz})\dot{\psi}^2 + L(i_1^2 + i_2^2) + 2cn\phi_3i_1 + 2cn\phi_4i_2. \end{aligned} \quad (1.7)$$

Вектор узагальнених сил містить сторонні електрорушійні сили  $U_1$ ,  $U_2$ , що застосовуються до електродвигунів:

$$Q = (0 \ 0 \ 0 \ 0 \ 0 \ U_1 \ U_2)^T. \quad (1.8)$$

З отриманих таким чином рівнянь виводяться [39] динамічні рівняння руху робота:

$$\begin{aligned}
m\dot{V} - m_1 a \Omega^2 - \frac{nc}{r}(i_1 + i_2) &= 0, \\
J\dot{\Omega} + m_1 a V \Omega - \frac{ncl}{r}(i_1 - i_2) &= 0, \\
L \frac{di_1}{dt} + R i_1 + \frac{nc}{r}(V + l\Omega) &= U_1, \\
L \frac{di_2}{dt} + R i_2 + \frac{nc}{r}(V - l\Omega) &= U_2,
\end{aligned} \tag{1.9}$$

де  $m = m_1 + 2m_k + 2\frac{J_y}{r^2}$ ,  $J = J_1 + 2J_{kz} + (m - m_1)l^2 + m_1 a^2$ .

Кути поворотів ведучих коліс  $\varphi_3, \varphi_4$  і заряди  $i_1, i_2$  не входять в рівняння, тому з кінематичних виразів найбільш суттєвими є лише три:

$$\dot{x} = V \cos \psi, \quad \dot{y} = V \sin \psi, \quad \dot{\psi} = \Omega. \tag{1.10}$$

Останні два рівняння описують перехідні процеси в колах електродвигуна, причому постійну часу перехідного процесу  $\tau = L/R$  розглядають [38] як час запізнення у колах керування ведучими колесами. Для випадку, коли час запізнення нульовий ( $L = 0$ ), з двох останніх рівнянь отримуються струми в електродвигунах:

$$i_1 = \frac{U_1}{R} - \frac{nc}{rR}(V + l\Omega), \quad i_2 = \frac{U_2}{R} - \frac{nc}{rR}(V - l\Omega). \tag{1.11}$$

Якщо внести струми з одного рівняння в інше, отримаємо диференціальні рівняння:

$$\begin{aligned}
m\dot{V} - m_0 a \Omega^2 + \frac{2n^2 c^2}{rR} V &= \frac{nc}{rR}(U_1 + U_2), \\
J\dot{\Omega} + m_0 a V \Omega + \frac{2n^2 c^2 l^2}{r^2 R} \Omega &= \frac{ncl}{rR}(U_1 - U_2).
\end{aligned} \tag{1.12}$$

Для практичного використання вирази можуть спрощуватися із введенням практичних параметрів електродвигунів мобільних платформ робота.

### 1.3.2 Інтелектуальне керування та його особливості

Термін інтелектуальне керування з'явився відносно нещодавно і пов'язується із застосуванням методів штучного інтелекту під час реалізації керування складними об'єктами [39-40].

Інтелектуальне керування належить до класу технологій, що використовують різні методи штучного інтелекту, зокрема наступні з них:

- керування за допомогою ШНМ;
- байєсівське керування;
- нечітке (фаззі) керування;
- експертні системи;
- генетичне керування;
- інтелектуальні агенти (когнітивне/усвідомлене керування).

Контролери на основі ШНМ використовуються для розв'язання завдань в багатьох сферах науки та технології, зокрема для ідентифікації та керування.

Байєсівські контролери також використовуються у системах керування, наприклад у завданнях оцінки змінних станів систем. На основі підходу Байєса будуються фільтри Калмана та частковий фільтр [41-42].

Одним за завдань, що вирішуються на рівні усвідомленого керування, є завдання вибору дії. У системах штучного інтелекту і у обчислювальній когнітивній науці таке завдання напряму пов'язують з концепціями інтелектуальних агентів та аніматів (анімат – штучна тварина) – штучними системами, що виконують складні завдання у агентному середовищі [43]. Сама дія може відображати різні рівні абстракції (від дій одноклітинного організму, до активності цілого суспільства). Для простих систем набір можливих дій є наперед визначеним і сталим. До агентів, що функціонують

відповідно принципам інтелектуального керування висуваються такі вимоги [44]:

- діючий агент має функціонувати у динамічному середовищі;
- агент має приймати рішення і діяти у реальному часі;
- агенти можуть виконувати набір (іноді конфліктний) різних завдань;
- середовище агенту може включати людей та інших агентів, які можуть втручатися у роботу;
- агенти можуть імітувати дії людей або тварин.

Основною проблемою вибору дії є складність, особливо якщо агент має розв'язувати завдання в умовах обмеженого часу (простору) та в умовах надлишку інформації про цей простір. Для розв'язання завдань подібного типу слід використати спеціальні підсистеми (вирішувачі) і в низці випадків розподіляти завдання між агентами, що виконують одні завдання.

Механізм вибору (прийняття) рішення має визначати не лише зміст та порядок дій агенту, а й спрямовувати його увагу на властивості середовища і оновлювати пам'ять, реалізувати навчання. Ідеальною вважається ситуація, коли механізм прийняття рішення буде здатним навчатися і адаптуватися [44].

Інтелектуальні механізми прийняття рішень ґрунтуються на класичних методах, які є основою для символічних систем; розподілених методах; методах з динамічним та реактивним плануванням.

Символічні системи виходять з того, що під час роботи агент має знайти оптимальний шлях розв'язання поставленого завдання і виконати його. При цьому має використатися опис символічної системи [45], що є необхідним та достатнім для моделювання інтелектуальних здібностей агенту. Символічна система має отримувати дані про стан сенсорів, об'єктів навколишнього середовища, набір своїх можливих дій і цілей і подавати їх у вигляді логіки предикатів. Недоліками символічного підходу вважають недостатню пристосованість для завдань планування у реальному часі і нездатності генерувати оптимальні плани з причини нездатності зведення реальної

системи до логічного опису. Однією з концепцій символічних систем є концепція задовільного рішення – адекватного і близького до оптимального.

Властивості агенту, відповідного символічній системі, описуються набором цілей, кожна з яких досягається в результаті виконання процесу, описаного планом. Завдання агенту полягатиме у виборі найкращого плану, що складається з підцілей, розв'язання може бути рекурсивним.

Відміною розподілених систем вибору дій є, насамперед, відсутність одного модуля, відповідного за прийняття рішення й наявність багатьох розподілених елементів, що функціонують паралельно і взаємодіють у виконанні поставленої перед агентом мети. До речі, ШНМ є засобом реалізації саме розподілених систем. Проблемним у такому підході залишається визначення характеру взаємодії і повсякчас необхідність у централізованому керуванні процесом. Вочевидь, біологічний мозок діє саме таким способом, оцінюючи декілька конкуруючих варіантів і обираючи найбільш бажаний з них.

Розподілені системи на практиці важко реалізувати, тому символічні системи отримали новий розвиток у підході динамічного або реактивного планування. За умов динамічного планування кожен наступний крок рішення, яке виконується за розробленим планом, виходить з поточної ситуації і має відповідати їй. На відміну від класичних методів, динамічний підхід не страждає проблемою комбінаторного вибуху, хоча і вважається негнучким з точки зору створення біологічного інтелекту (сильний ШІ) з точки зору попереднього складання планів діяльності. На основі даної концепції розробляються СПР для комп'ютерних ігор [46].

Таким чином, якщо в якості об'єкта інтелектуального керування розглядати робототехнічні пристрій, можна зауважити на доречність концепції інтелектуального керування як засобу поєднання, з одного боку – методів теорії автоматичного керування (наприклад для розв'язання завдань керування окремими двигунами мобільного або маніпуляційного робота), з другого – методів штучного інтелекту. Такий підхід цілком вписується у



концепцію інтелектуального агента, який приймає рішення на рівні інтелектуальної складової керування і здійснює заплановані дії за допомогою генерації необхідних сигналів двигунам виконавчої системи роботи.

Системи підтримки прийняття рішень (СППР) виявляють потенційні рішення і альтернативи, що слід розглянути для вирішення неструктурованих проблем. Ці системи надають і визначають альтернативи, що найкраще узгоджувались би з цілями і завданнями організації. У процесі прийняття рішень в організаціях беруть участь різні етапи [46-48].

Виявлення проблем є першим кроком у процесі прийняття рішення, оскільки саме тут визначається проблема, визначаються зацікавлені сторони і визнаються особи, які приймають рішення в організації .

В загальному вигляді процес прийняття рішень, який складається з 6 фаз, представлено на рис. 1.11.

Визначення проблем та вимог (фази 1 і 2), як частини процесу прийняття рішення, початковим кроком якої є правильне визначення проблеми, як тільки вона була виявлена [49-50].

Завдання фази 1, в процесі прийняття рішення, полягає в тому, щоб задокументувати проблему і визначити вимоги, які необхідно виконати, перш ніж цілі будуть чітко сформульовані [51-53]. Дії по визначенню вимог, які необхідно вирішити, підкреслюють аспекти рішень, які будуть вирішувати проблеми.

Наступна фаза (фаза 2) – створення альтернатив, які надають окремим особам і організаціям варіанти, які слід враховувати для мінімізації двозначності, яка існує в процесі прийняття рішень. Важливим фактором, який слід враховувати, те, що створені варіанти повинні відповідати заявленим вимогам при вирішенні проблем. Ця фаза є етапом виявлення проблем і вироблення альтернатив. Результати, отримані на цьому етапі, будуть вважатися авторитетними в контексті і можуть використовуватися при формуванні заяви про рішення.

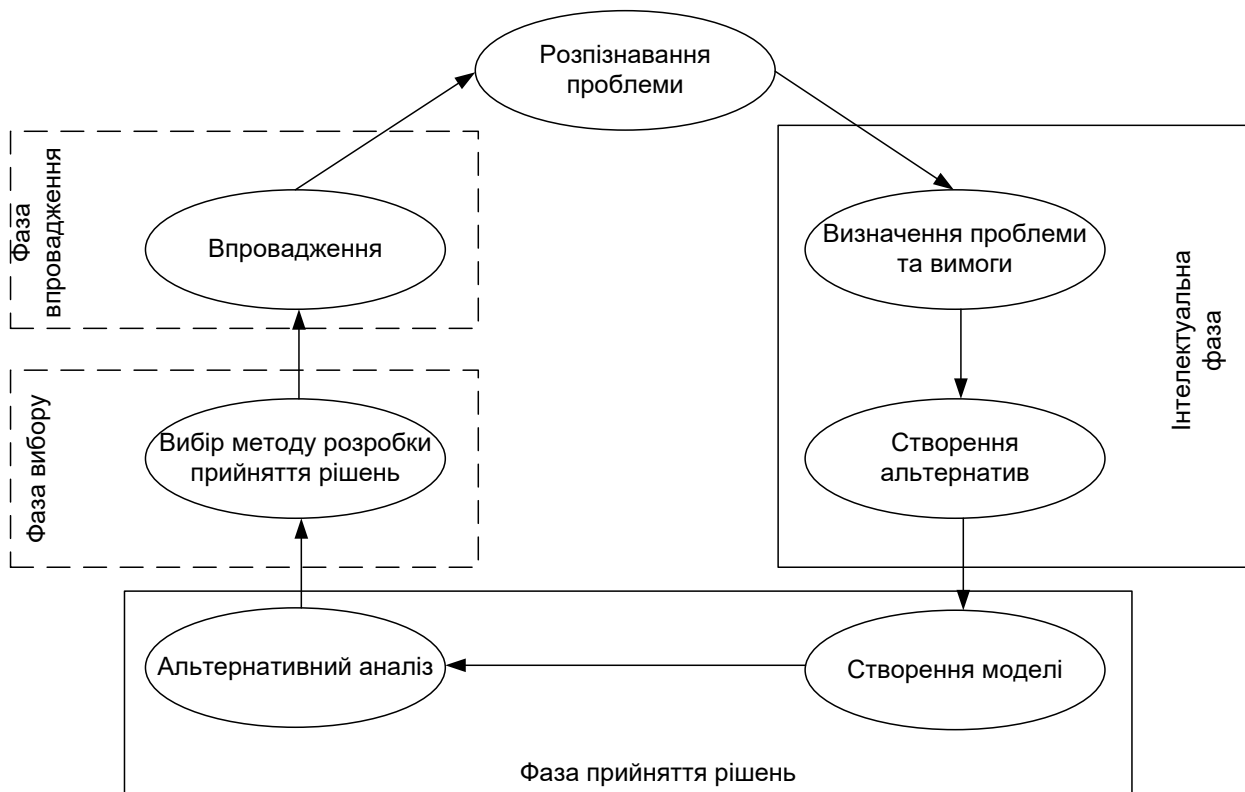


Рисунок 1.11 – Процес прийняття рішень

Створення моделі (фаза 3). Наступним кроком після створення альтернативних рішень і вибору варіанта, який, здавалося б, відповідав цілям організації і був би найбільш гідною кандидатурою, є створення моделі. У процесі прийняття рішення моделі використовуються при аналізі згенерованих альтернатив. При аналізі альтернатив необхідно встановити вимір, щоб забезпечити об'єктивну оцінку ефективності обраної альтернативи. Найкращою моделлю буде та, яка може аналізувати альтернативи швидко і відповідно до цілей організації [54-56].

Альтернативний аналіз (фаза 4) – після розробки моделі наступним етапом в процесі прийняття рішення є аналіз виявлених варіантів, і це буде шлях, який може привести до вибору найкращого варіанту на основі оцінки варіанти проти певних заходів з використанням допоміжних інструментів для прийняття рішень. Існує безліч інструментів підтримки процесу прийняття рішень, таких як процес аналітичної ієрархії, аналіз теорії корисності з декількома атрибутами, аналіз плюсів і мінусів і аналіз рішень Кепнера-Трего

(КТ). В останні роки були проведені дослідження, присвячені оцінці доречності інструментів підтримки прийняття рішень, переваг і недоліків кожного з них, а також застосування цих інструментів в ситуаціях, коли рішення було змінено після використання конкретних методів.

Нарешті, знайдене рішення повинно бути оцінено з точки зору того, якою мірою вирішена первісна виявлена проблема. Ця частина процесу відома як етап проектування, коли встановлюються різні варіанти вирішення, а цілі організації інтегруються в моделі.

Вибір методу розробки (фаза 5): на цьому етапі фактичне рішення, вибране з набору альтернативних рішень, згенерованих на етапі проектування, готується до виконання як частина «етапу реалізації».

Впровадження (фаза 6): на цьому етапі реалізується обраний для виконання варіант. Ключовим моментом є те, що варіант, обраний в якості рішення, повинен відповідати всім визначеним вимогам і вирішувати заявлені організаційні цілі в контексті процесу прийняття рішень [56].

Фабрики майбутнього в епоху Industry 4.0 вимагають інтегрованого керування всіма поставками, прогнозування попиту і інтеграції технологій протягом усього процесу керування – створення модельно-керованих СППР. Датчики в поєднанні з кількісними моделями можуть знизити витрати і виявити несправності в електромережі, такі як відмова датчика і його погіршення. В ідеалі моделі будуть оптимізувати частоту і час вимірювання датчиків і усувати або скорочувати затримки в мережі ланцюжка поставок [57].

Також, створення інтелектуальної системи керування зі зворотним зв'язком може поліпшити якість продукції і забезпечити зворотний зв'язок для керування системою, яку можна використовувати для поліпшення планування виробництва, для підтримки машини та поліпшити профілактичне обслуговування.

Процеси прийняття рішень відбуваються і у робототехніці. Зокрема у різних процесах виконання завдання. На рис. 1.12 представлений алгоритм прийняття рішень про завдання траєкторії руху роботів [59].

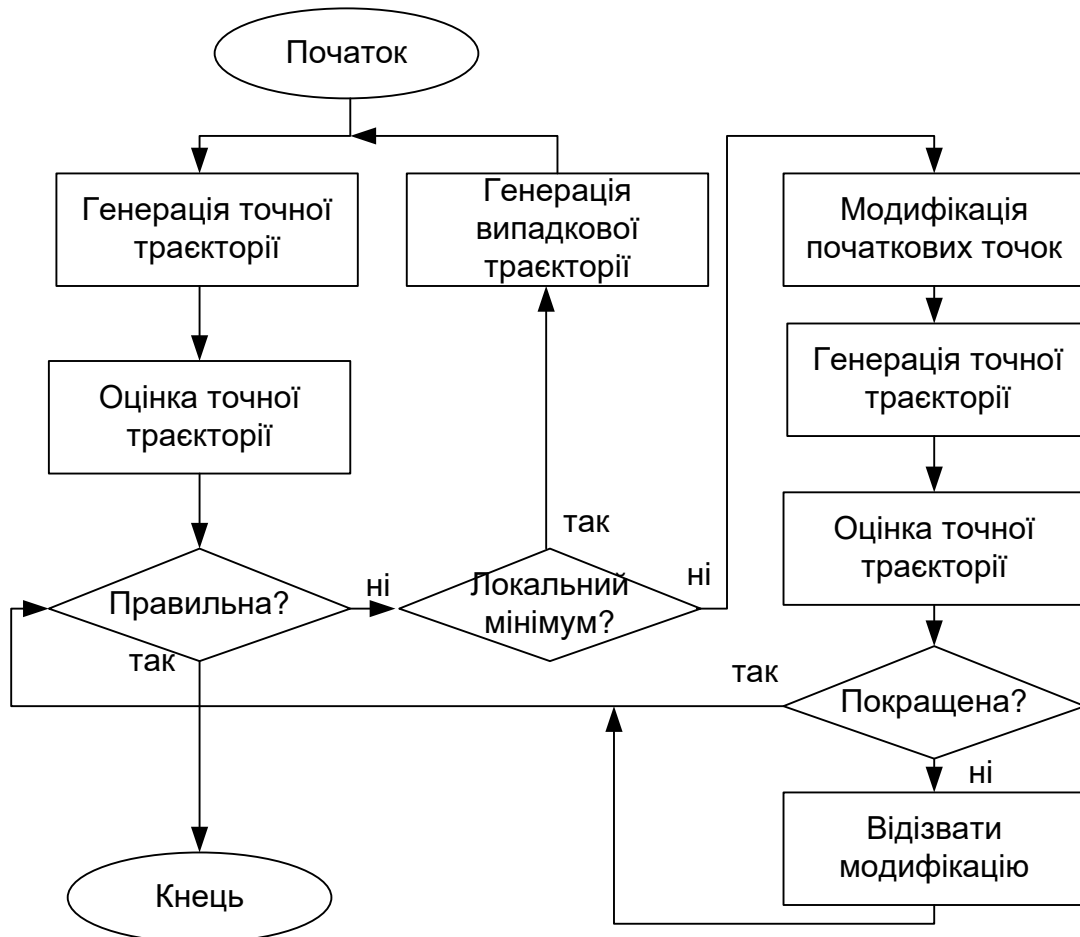


Рисунок 1.12 – Алгоритм прийняття рішень для отримання траєкторії руху робота

У [58] автор генерує точну траєкторію на основі базової точки і обчислює оцінку її якості. Потім це повторюється до тих пір, поки не буде знайдена точна траєкторія (тобто траєкторія без зіткнень, яка дотримується всіх кінематичні і динамічні обмеження робота).

Знаходиться і вибирається відрізок поточної траєкторії з найгіршим показником якості – оцінка траєкторії проводиться кусочно. Особливу увагу приділено парі наступних один за одним базових точок. Потім ця частина

змінюється локально, поки не знайдеться модифікацію траєкторії, що дозволяє згенерувати кращу траєкторію, ніж попередня. Можливими модифікаціями є переміщення або вставка базової точки. Якщо якість не поліпшено, виконується випадкова генерація точок. Робот випадково переміщується в просторі конфігурації і відбувається корегування сусідніх базових точок, щоб досягти траєкторії базової точки.

Потім знову обирається частина поточної траєкторії з найгіршим показником якості і повторюються всі кроки.

Оскільки нема можливості повністю дослідити робочий простір, потрібно досягти мети переміщення, багаторазово покращуючи задану траєкторію, тобто виконуючи локальний пошук. На кожному етапі потрібно покращити результати, за винятком тих випадків, коли застосовується випадкова генерація. Намагаючись поліпшити ці показники, особливу увагу приділено окремому відрізку поточної траєкторії, а саме відрізку з гіршим показником якості, оскільки такий відрізок є хорошим кандидатом на зміну, яке, ймовірно, призведе до формування траєкторії з найкращим рейтингом.

Метод, описаний у [59] вимагає створення дорожньої карти, яка побудована для статичної частини сцени, тобто її набір вершин і ребер у робочому просторі не повинен мати зіткнень щодо статичних перешкод. З цього можна зробити висновок, що дорожню карту можна створити на етапі попередньої обробки зображення. Передбачається, що старт та цілі переміщення присутні у дорожній карті як вершини. Якщо ні, їх можна підключити до дорожньої карти на етапі запиту. Метод застосовний як до спрямованих, так і до ненаправлених дорожніх карт.

Ідея використання попередньо обробленої дорожньої карти полягає в тому, що під час фази запиту стаціонарні перешкоди не потрібно враховувати під час перевірки зіткнень, що економить велику кількість часу. Більше того, проблеми вузького проходу, викликані стаціонарними перешкодами, вирішуються на фазі попередньої обробки, що істотно полегшує фазу виконання, а також скорочується пошук можливих шляхів.

Використання дорожньої карти може мати також практичні переваги. У багатьох реальних середовищах, таких як фабричні підлоги, морські та аеропортові тощо, мобільні роботи змушені рухатися уздовж заданих мереж шляхів (наприклад, уздовж ліній, намальованих на підлозі). Їх можна ідеально змоделювати в дорожню карт.

Аналогічні принципи опубліковано і у [60]. Але крім дорожньої карти, автор пропонує використання навчання з підкріпленням. При навчанні з підкріпленням змодельований агент (в нашому випадку, мобільний робот) виконує дії і отримує винагороду за певні результати.

Замість того, щоб компілювати план в потенційну функцію, формується послідовність керування в певний момент часу, але виконується тільки перша команда в послідовності керування. Після виконання команди відстежується новий стан робота і складається нова оптимальна послідовність керування.

Замість того, щоб попередньо обчислювати план і перетворювати його в потенційну функцію, інший варіант полягає в тому, щоб одночасно керувати агентом, використовуючи інформацію, отриману в результаті цієї дії, для поліпшення політики керування. Для цього можна використовувати методи динамічного програмування в реальному часі, адаптувати ці алгоритми до роботи в режимі реального часу.

У динамічному програмуванні в реальному часі використовується марковський процес прийняття рішень (МППР) для моделювання основного поведінки системи. Це також передбачає, що структура цієї моделі відома. Адаптивне динамічне програмування в реальному часі є розширенням цього підходу до ситуацій, коли структура моделі невідома.

Провівши аналіз СППР, можна зробити висновки про їх використання:

- у експертних системах виробничого призначення;
- у системах комунікації між роботами та їх операторами;
- у системах планування маршрутів переміщення мобільних роботів;
- у будь-яких системах інтелектуального керування для більш швидкого прийняття рішень.

#### 1.4 Аналіз моделей та методів адаптивного візуального керування

Розглянутий вище розвиток гнучких інтегрованих систем, робототехнічних засобів та інтелектуальних систем керування демонструє збільшення ролі і значення сенсорних систем у прийнятті рішень щодо розв'язання технологічних завдань на конкретних робочих місцях для маніпуляційних роботів, на шляхах переміщення транспортувальних роботів та поєднанні маніпуляційних та транспортувальних завдань [62-63].

Основою для отримання інформації про навколишнє середовище у людини виступає його зорова система. Завдання моделювання цієї системи людини в системах інтелектуального керування досить успішно вирішується створенням систем технічного та комп'ютерного зору [64-65]. Проте, проблема полягає у визначенні ступеню інтеграції штучних зорових функцій технічних систем з виконавчими механізмами автоматизованого обладнання, маніпуляційних та транспортувальних роботів. Інакше кажучи, системи технічного та комп'ютерного зору мають бути інтегровані в інтелектуальні системи керування таким чином, щоб забезпечити супроводження процесів автоматизованого керування у режимі реального часу, із одночасним моніторингом виконання процесів та, в разі необхідності, корекцією керуючих впливів [66-67].

Один з підходів до вирішення проблем, що викликаються невизначеними або мінливими параметрами системи – використання методів адаптивного керування. Адаптація – процес цілеспрямованої зміни параметрів і структури системи, який полягає у визначенні критеріїв її функціонування і виконання цих критеріїв.

Адаптивне візуальне керування – керування, засноване на адаптації системи до змін параметрів робочого середовища роботизованої системи, на основі інформації, що надходить від системи технічного зору (візуальної підсистеми) в процесі функціонування системи [68-69].

Основною особливістю адаптивних систем візуального керування [70] є можливість отримання зорової інформації в процесі функціонування і використання цієї інформації для реалізації функцій керування.

Прикладом застосування адаптивної системи візуального керування може бути мобільний робот (МР) [71-72], обладнаний бортовою системою технічного або комп'ютерного зору (або система технічного/комп'ютерного зору може бути об'єктовою – розташованою над робочим пр/остором). Під час виконання завдання із переміщення, на шляху руху робота можуть виникати динамічні перешкоди – інші роботи, люди, транспортувальні засоби, переміщене обладнання. Система технічного/комп'ютерного зору повинна в момент їх появи відразу ж посилати команду роботу про наявність перешкод, а система інтелектуального керування – вирішити, як продовжити виконання поставленого перед роботом завдання, надсилаючи йому керуючі впливи, які адаптуватимуть систему керування до змін робочого простору [73-74].

Розглянемо класичну задачу адаптивного візуального керування, запропоновану в [61]. Структурна схема системи візуального керування роботом представлена на рис. 1.13.

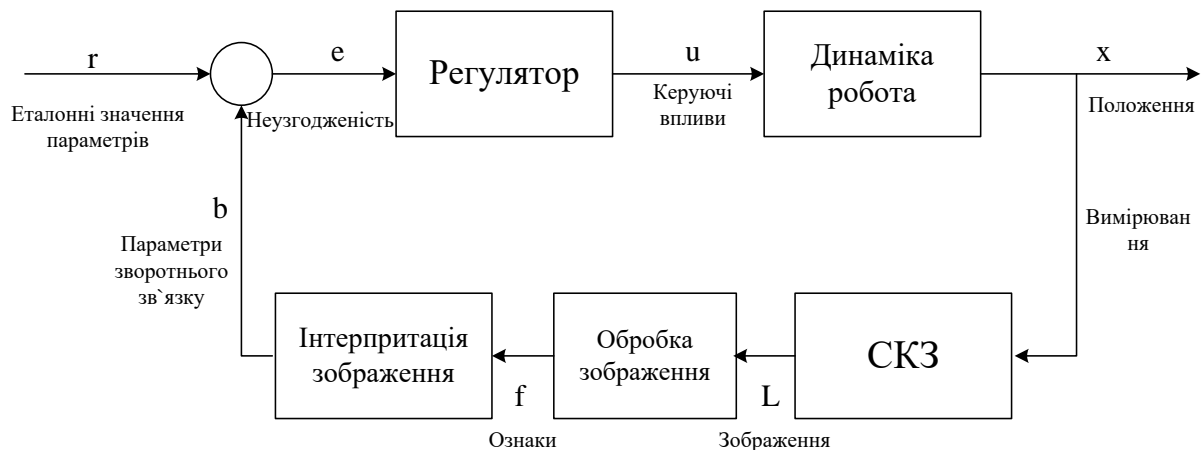


Рисунок 1.13 – Класична задача адаптивного візуального керування



Схема складається з регулятора, динамічної складової для керування мобільним роботом, системи комп'ютерного зору (СКЗ), блоку обробки зображень, блоку інтерпретації зображення.

Даний вид схеми (рис. 1.13) підходить для ідеального випадку, коли вплив зовнішніх факторів настільки малий, що ними можна знехтувати.

Але для більш складних систем, зокрема, адаптивних, ця схема має певні недоліки, що зумовлені суттєвим удосконаленням програмно-технічних засобів СКЗ та їх доступності широкому колу користувачів та розробників. Крім того, з'явилися ефективні програмні бібліотеки обробки візуальної інформації, серед яких слід вирізнити OpenCV [74]. Також оновилися методи обробки та інтерпретації зображень, які мали досить низьку якість розпізнавання при зміні масштабу зображення цільового об'єкту на кадрі та його повороту.

Для завдання обробки зображень і їх інтерпретації використовували лише класичні підходи до обробки інформації комп'ютерного зору, і це в наш час потребує оновлення, в тому числі, шляхом впровадження засобів машинного навчання та нейронних мереж. Вирішення завдань розпізнавання та ідентифікації об'єктів робочого простору надає можливість побудувати модель опису, зокрема, на основі декларативного подання. У свою чергу, декларативна модель здатна бути підґрунтям для побудови функціональної моделі робочого простору і реалізації методу адаптивного візуального керування. Впровадження вказаних моделей і методу дозволяє представити схему адаптивного візуального керування у вигляді, показаному на рис. 1.14.

Відповідно до динамічного характеру робочого простору робота, візуальна система керування повинна забезпечити аналіз робочого простору для переміщення транспортувальної роботизованої платформи в межах гнучкої інтегрованої системи приладобудівного виробництва. Динамічний характер робочого простору визначається умовами виробництва, висуває вимоги до адаптації, які повинні забезпечити підвищення стабільності роботи гнучкої інтегрованої виробничої системи [76-78].

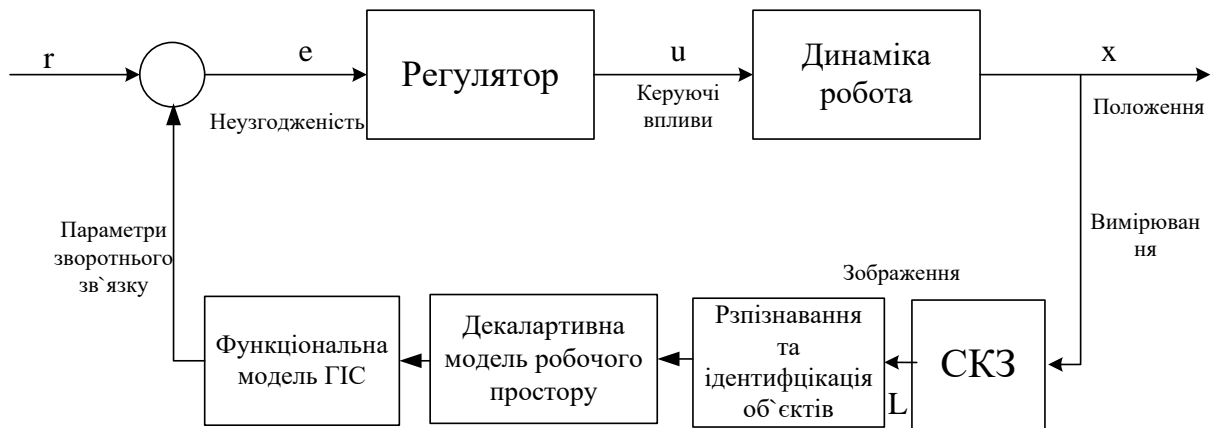


Рисунок 1.14 – Вдосконалений підхід адаптивного візуального керування

Таким чином, актуальною проблемою [77-78] сучасних гнучких інтегрованих систем залишається забезпечення виконання виробничих функцій, спрямованих на підвищення ефективності виробництва за рахунок безперервності функціонування всієї системи.

Істотним їх недоліком є відсутність можливостей систем керування робототехнічними платформами адаптуватися до змін робочого простору, що можуть наступати внаслідок динаміки виконання виробничих завдань, впливу людського фактору, потреби у зміні технології виробництва, виникнення різного роду нештатних ситуацій тощо. Система керування робототехнічною платформою, яка буде враховувати зміни робочого середовища і станів гнучкої інтегрованої системи, повинна здійснювати спостереження за умовами виконання технологічних завдань, спостерігати за пересуванням персоналу та інших мобільних платформ, слідкувати за виникненням нештатних ситуацій і при необхідності, адаптувати процес виконання виробничих функцій у ГВС в цілому. В якості такої системи пропонується система адаптивного візуального керування [79-81].

Отже, адаптивне візуальне керування є перспективним напрямком, що може бути використано на сучасному автоматизованому виробництві для керування мобільними роботами, що виконують транспортувальні операції та

промисловими роботами для виконання маніпуляцій. Головною перевагою такої системи є її гнучкість, що полягає у високій пристосованості та здатності до швидкого переналагодження виробничих ділянок.

Для того, щоб створити адаптивну візуальну систему керування, необхідно провести певні функціональні дії:

- провести розпізнавання та ідентифікацію усіх об'єктів робочого простору;
- провести моделювання робочого простору з описом характеристик об'єктів та наявними між ними зв'язками (декларативна модель);
- оцінити функції, що виконуються об'єктами і їх взаємозв'язок.

#### 1.5 Висновки за розділом 1 та постановка завдань дисертаційного дослідження

У першому розділі розглянуто проблему адаптивного керування мобільними та маніпуляційними роботами, що функціонують у складі гнучких інтегрованих систем приладобудівного виробництва.

Сучасні виробничі системи організуються у такий спосіб, що технологічний процес механічної обробки і складання має здійснюватися в одному або декількох цехах, що містять обчислювальні центри (ОЦ), окремі верстати з ЧПУ, складальні ділянки (СД) і транспортувальну систему, технологічне обладнання з автоматизованим складом.

До недоліків організації виробничих процесів, зокрема належать:

- низький рівень автоматизації транспортувальних, завантажувально-розвантажувальних операцій, в тому числі для верстатів з ЧПК;
- відсутність автоматизованих засобів усунення нештатних або нестандартних виробничих ситуацій;

– недостатній рівень забезпечення засобами моніторингу станів технологічного обладнання, транспортувальних, складських та обслуговуючих систем.

З метою подолання зазначених недоліків пропонується:

– ввести до складу гнучких інтегрованих систем засоби моніторингу технологічного обладнання, транспортувальних, складських та обслуговуючих систем на основі систем комп'ютерного/технічного зору, що встановлюватимуться для спостереження ділянок робочого простору гнучких інтегрованих систем та на борту мобільних роботизованих платформ;

– розробити математичне, алгоритмічне та програмне забезпечення адаптивних систем керування мобільних роботизованих платформ на основі візуальної інформації засобів комп'ютерного/технічного зору.

Мобільна роботизована платформа має відповідати наступним вимогам:

– вільне переміщення в межах цехів поза робочих просторів окремих одиниць технологічного обладнання;

– доставку заготовок та інших матеріалів до робочої зони оброблювальних центрів і верстатів з ЧПК;

– доставку необхідного інструменту або оснащення за регулярними або терміновими запитами;

– спостереження за технологічним обладнанням та іншим обладнанням цехів та виробничих ділянок;

– контроль роботи окремих одиниць технологічного обладнання.

Для забезпечення своєї функціональності роботизована платформа має відповідати наступним конструктивним вимогам:

– наявність шасі;

– наявність маніпулятора (або декількох маніпуляторів);

– наявність вантажного відсіку для перевезення заготовок, деталей, інструменту та оснастки;

– наявність візуальної системи керування;

- наявність системи керування на основі бортової ЕОМ;
- наявність сенсорної системи шасі і маніпулятора.

Роботизована платформа має обиратися на основі існуючих моделей транспортувальних роботів і маніпуляторів.

Відповідно до динамічного характеру робочого простору роботизованої платформи, візуальна система керування має забезпечити аналіз робочого простору для подальшого переміщення. Динамічний характер робочого простору платформи визначається умовами виробництва, висуває вимоги до адаптації, які повинні забезпечити підвищення стабільності роботи гнучких інтегрованих виробничих систем.

Таким чином, актуальним завданням сучасних гнучких інтегрованих виробничих систем залишається забезпечення виконання виробничих функцій, спрямованих на підвищення ефективності виробництва за рахунок безперервності функціонування всієї системи.

Аналіз існуючих методів адаптивного керування роботами підтверджує доцільність застосування засобів комп'ютерного/технічного зору в якості одного з основних елементів систем керування мобільними платформами та маніпуляторами. При цьому необхідно вирішити ряд теоретичних і практичних завдань, пов'язаних з розробкою інтелектуальних моделей ідентифікації робочого простору гнучких інтегрованих робототехнічних систем, моделей та метода адаптивного візуального керування, що забезпечують удосконалення експлуатаційних характеристик гнучких інтегрованих робототехнічних систем.

Метою дисертаційної роботи є розробка нових та удосконалення існуючих моделей та методів адаптивного візуального керування, що забезпечують підвищення продуктивності, швидкодії та надійності робототехнічних засобів гнучких інтегрованих виробничих систем.

Для досягнення мети необхідно вирішити такі завдання:

- дослідити сучасний стан гнучких інтегрованих роботизованих систем і застосування засобів адаптивного та інтелектуального керування;
- розробити інформаційно-логічну модель подання інформації про робочий простір ГІВС;
- розробити функціональну модель, що забезпечує опис взаємодії об'єктів ГІВС під час виконання виробничих завдань;
- вдосконалити метод адаптивного візуального керування;
- вдосконалити моделі керування мобільним роботом у просторі ГІВС на основі впровадження візуального цілевказання та нечіткого опису параметрів керування, модель розпізнавання та ідентифікації об'єктів;
- практично реалізувати розроблені моделі і метод, шляхом розробки інформаційного та програмного забезпечення, що забезпечує адаптивне візуальне керування мобільними роботами, які функціонують у складі ГІВС.

Основні результати цього розділу опубліковано в роботах [63], [66], [71], [75], [79], [81].

## РОЗДІЛ 2

### МОДЕЛЮВАННЯ РОБОЧОГО ПРОСТОРУ ТА ФУНКЦІОНАЛЬНИХ ЗВ'ЯЗКІВ ГНУЧКИХ ІНТЕГРОВАНИХ ВИРОБНИЧИХ СИСТЕМ

#### 2.1 Інформаційно-логічна модель робочого простору

Створення будь-якої моделі об'єктів керування неможливе без моделі робочого простору, в якому присутні такі об'єкти, що виконують поставлені перед ними завдання [67], [71-72].

Розглянемо побудову моделі робочого простору гнучкої інтегрованої виробничої системи (ГІВС), в якій ставляться завдання керування мобільними роботами.

Нехай існує робочий простір (РП)  $W_s$  робота  $Rb$ . Простір описується властивостями:

- геометричні розміри  $D(x, y, z)$ ;
- набір приналежних простору об'єктів  $Obj$ ;
- інтервал часу  $T_{param}$  існування РП;

Тоді простір можна записати наступним чином:

$$W_s = \langle D(x, y, z), Obj, T_{param} \rangle \quad (2.1)$$

Кожен з об'єктів множини (набору) об'єктів  $Obj$  робочого простору має унікальний ідентифікатор  $ID$ , що означає можливість ідентифікації об'єкта, зокрема за допомогою штрих-, QR-кодів, тощо.

Необхідно враховувати головну властивість простору – його дискретність і скінченність (обмеженість). Під скінченністю будемо розуміти межі робочого простору камери. Випадок відкритого (необмеженого) простору, в принципі, є окремим завданням.

Під дискретністю будемо розуміти поділ простору на осередки, однакові по своїй довжині і ширині. В залежності від рівня дискретизації РП, можна ставити завдання переміщення (або маніпуляції) об'єктів керування різної точності.

Дискретний характер робочого простору означає наявність координат об'єктів, що знаходяться в РП та коефіцієнту  $K_{FL}$  заповненості ділянки РП:

$$D(z, y, z) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l d(x_i, y_j, z_k), K_{FL} \in [0,1] \quad (2.2)$$

де  $d(x_i, y_j, z_k)$  – геометричні параметри осередку дискретного простору,  $K_{FL}$  – коефіцієнт заповнення осередку. Необхідно також враховувати те, що всі параметри осередків повинні бути однакового розміру.

$$K_{FL} = \frac{S_{FL}(d(x, y, z))}{S(d(x, y, z))}, K_{FL} \leq 0.25 \quad (2.3)$$

де  $S(d(x, y, z))$  – площа осередку дискретного простору,  $S_{FL}(d(x, y, z))$  – заповнена частина осередку дискретного простору  $d(x, y, z)$ .

У робочому просторі ГІВС передбачається існування певних об'єктів *Obj* – верстати (*Vr*), інструмент (*Ins*), оснащення (*Osn*), людина (*Hum*), робот (*Rb*), накопичувачі (*Storage*), конвеєри (*Conv*), пристрої моніторингу робочого простору (*Mon*):

$$\begin{aligned} \exists Vr \in W_s; \exists Ins \in W_s; \exists Osn \in W_s; \exists Hum \in W_s; \exists Rb \in W_s; \\ \exists Storage \in W_s; \exists Conv \in W_s; \exists Mon \in W_s \end{aligned} \quad (2.4)$$

$$Obj = \langle Vr, Ins, Osn, Hum, Rb, Storage, Conv, Mon \rangle \quad (2.5)$$



З цього виходить, що з точки зору оголошення властивостей вся ГІВС може бути виражена виразом:

$$FIS = \langle W_s, Rb, Vr, Ins, Osn, Hum, Storage, Conv, Mon \rangle \quad (2.6)$$

Кожен з об'єктів володіє набором властивостей. Ці властивості мають певні значення, входять в множини імен і значень властивостей. У табл. 2.1 наведені основні характеристики об'єктів в символічному поданні.

Між об'єктами (предметами) і їх властивостями існують відносини приналежності, тобто певному об'єкту належать певні властивості. Приклад властивостей предметів наведено в табл. 2.1.

Приклад схеми робочого простору і об'єктів, розташованих в ньому, наведений на рис. 2.1, що показує необхідність врахування людського фактору під час планування транспортувальних операцій для мобільних робототехнічних засобів.

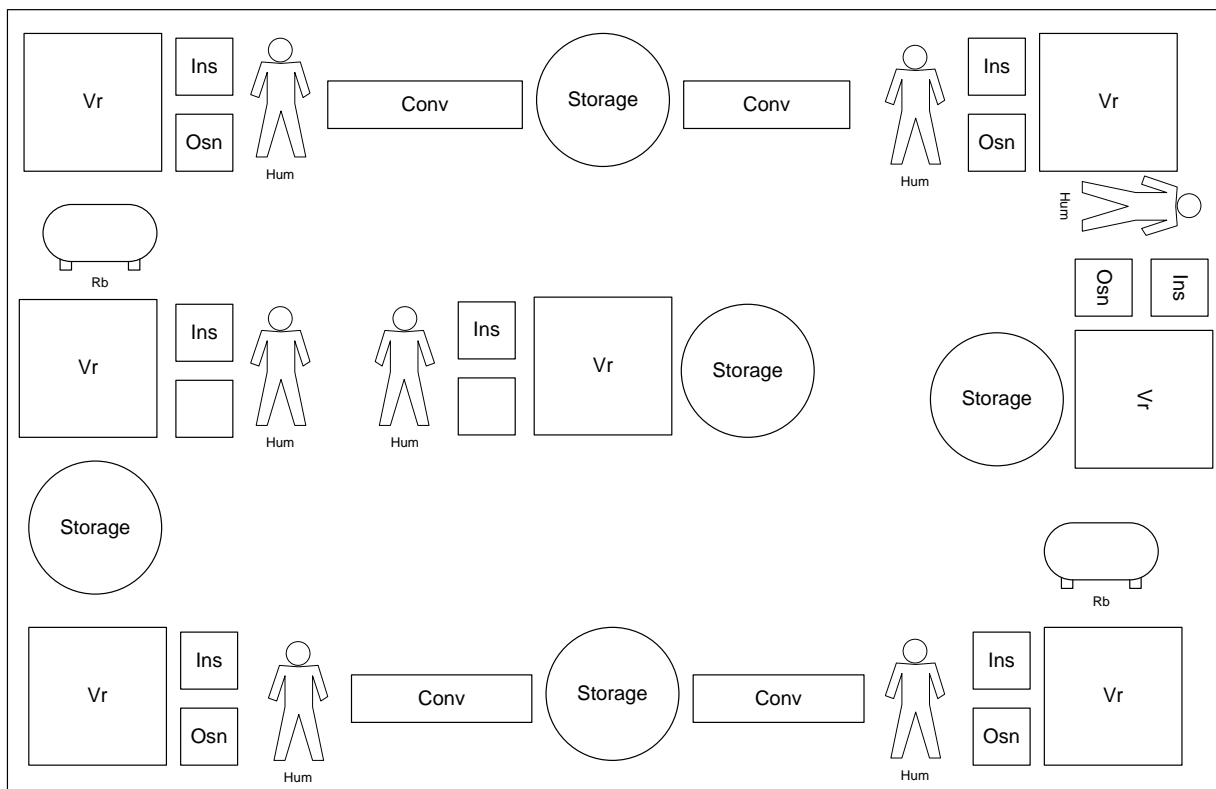


Рисунок 2.1 – Приклад розташування об'єктів у робочому просторі

Об'єкти робочого простору  $W_s$  існують як статично, так і динамічно.

До статичних об'єктів, що не змінюють своє положення і не впливають на стани переміщення робота, можна віднести такі об'єкти: верстати ( $Vr$ ), конвеєри ( $Conv$ ) та накопичувачі ( $Storage$ ).

Динамічними об'єктами, що можуть змінити своє положення і тим самим вплинути на стан переміщення робота у робочому просторі є: інструмент ( $Ins$ ), оснащення ( $Osn$ ), людина ( $Hum$ ).

Спостереження динаміки робочого простору забезпечується пристроями моніторингу, до яких належать об'єктові системи комп'ютерного/технічного зору (ОСКЗ) –  $Camera_{Glob}$  та локальні системи комп'ютерного/технічного зору (ОСКЗ) –  $Camera_{Loc}$ , інші датчики стану робочого простору ( $Sens$ ).

Таблиця 2.1 – Перелік властивостей предметів  $Obj$

Предмет	Властивість предмета $Param_{Obj}$	Позначення імені властивості
$Vr$	Вид верстата	$Type_{Vr}$
	Геометричні параметри	$D_{Vr}(x_{Vr}, y_{Vr}, z_{Vr})$
	Методи обробки	$PM_{Vr}$
	Особливі умови	$Sc_{Vr}$
	Унікальний ідентифікатор	$ID_{Vr}$
$Ins$	Вид інструмента	$Type_{Ins}$
	Геометричні параметри	$D_{Ins}(x_{Ins}, y_{Ins}, z_{Ins})$
	Методи обробки	$PM_{Ins}$
	Унікальний ідентифікатор	$ID_{Ins}$
$Osn$	Вид оснащення	$Type_{Osn}$
	Геометричні параметри	$D_{Osn}(x_{Osn}, y_{Osn}, z_{Osn})$
	Унікальний ідентифікатор	$ID_{Osn}$

Продовження таблиці 2.1

Предмет	Властивість предмета $Param_{Obj}$	Позначення імені властивості
<i>Mon</i>	ОСКЗ	$Camera_{Glob}$
	ЛСКЗ	$Camera_{Loc}$
	Інші датчики	$Sens$
<i>Hum</i>	Геометричні параметри	$D_{Hum}(x_{Hum}, y_{Hum}, z_{Hum})$
	Параметри пересування в робочій зоні	$Mv_{Hum}(x_{HumMv}, y_{HumMv}, z_{HumMv})$
	Приналежність до персоналу	$Per_{Hum}$
	Досвід	$Exp_{Hum}$
	Вік	$Age_{Hum}$
	Кваліфікація	$Qual_{Hum}$
	Унікальний ідентифікатор	$ID_{Hum}$
<i>Rb</i>	Геометричні параметри	$D_{Rb}(x_{Rb}, y_{Rb}, z_{Rb})$
	Параметри пересування в робочій зоні	$Mv_{Rb}(x_{Rb}, y_{Rb}, z_{Rb})$
	Швидкість пересування	$Speed_{Rb}$
	Поточне місце розташування	$Cp_{Rb}(x_{cpRb}, y_{cpRb}, z_{cpRb})$
	Унікальний ідентифікатор	$ID_{Rb}$
<i>Storage</i>	Геометричні параметри	$D_{Storage}(x_{Storage}, y_{Storage}, z_{Storage})$
	Тип	$Type_{Storage}$
	Максимальна кількість деталей	$Quan_{Storage}$
	Унікальний ідентифікатор	$ID_{Storage}$
<i>Conv</i>	Геометричні параметри	$D_{Conv}(x_{Conv}, y_{Conv}, z_{Conv})$
	Тип	$Type_{Conv}$
	Максимальна кількість деталей	$Quan_{Conv}$
	Швидкість	$Speed_{Conv}$
	Унікальний ідентифікатор	$ID_{Conv}$

Згідно таблиці 2.1, всі властивості предметів можна записати у вигляді кортежів параметрів.

$$\forall v \in Vr, \exists v = \langle Type_{Vr}, D_{Vr}(x_{Vr}, y_{Vr}, z_{Vr}), PM_{Vr}, Sc_{Vr}, ID_{Vr} \rangle \quad (2.7)$$

$$\forall ins \in Ins, \exists ins = \langle Type_{Ins}, D_{Ins}(x_{Ins}, y_{Ins}, z_{Ins}), PM_{Vr}, ID_{Ins} \rangle \quad (2.8)$$

$$\forall o \in Osn, \exists o = \langle Type_{Osn}, D_{Osn}(x_{Osn}, y_{Osn}, z_{Osn}), ID_{Osn} \rangle \quad (2.9)$$

$$\forall h \in Hum, \exists h = \langle D_{Vr}(x_{Vr}, y_{Vr}, z_{Vr}), Mv_{Hum}(x_{HumMv}, y_{HumMv}, z_{HumMv}), \quad (2.10)$$

$$Per_{Hum}, Exp_{Hum}, Age_{Hum}, Qual_{Hum}, ID_{Hum} \rangle$$

$$\forall rb \in Rb, \exists rb = \langle D_{Rb}(x_{Rb}, y_{Rb}, z_{Rb}), Mv_{Rb}(x_{Rb}, y_{Rb}, z_{Rb}), Speed_{Rb}, \quad (2.11)$$

$$Cp_{Rb}(x_{cpRb}, y_{cpRb}, z_{cpRb}), ID_{Rb} \rangle$$

$$\forall storage \in Storage, \exists storage = \langle D_{Storage}(x_{Storage}, y_{Storage}, z_{Storage}), Type_{Storage}, \quad (2.12)$$

$$Quan_{Storage}, ID_{Storage} \rangle$$

$$\forall conv \in Conv, \exists conv = \langle D_{Conv}(x_{Conv}, y_{Conv}, z_{Conv}), Type_{Conv}, Quan_{Conv}, \quad (2.13)$$

$$Speed_{Conv}, ID_{Conv} \rangle$$

$$\forall mon \in Mon, \exists mon = \langle Camera_{Glob}, Camera_{Loc}, Sens \rangle. \quad (2.14)$$

При цьому, система моніторингу складається з камер спостереження та сенсорів різного типу та призначення.

Зокрема, камери мають наступні властивості:

$$\forall cam \in Cam, \exists cam = \langle Inst_{pt}(x_{pt}, y_{pt}, z_{pt}), Angle_{View}, Resolution \rangle. \quad (2.15)$$

Всі предмети робочої зони взаємодіють між собою різного роду залежностями, що дозволяє ввести визначення.

Визначення 1. Кожен об'єкт робочої зони робота має хоча б одну властивість.

$$\forall x(x \in Obj) \exists param(param \in Param_{Obj})[x(param)]. \quad (2.16)$$

де  $\forall$  – квантор спільності,  $\exists$  – квантор існування,  $x$  – певний об'єкт робочої зони,  $Obj$  – множина об'єктів робочої зони,  $Param_{Obj}$  – множина властивостей об'єктів.

Множина  $Param_{Obj}$  властивостей предметів включає в себе такі параметри як:  $Param_{Vr}$  – множина властивостей верстатів,  $Param_{Ins}$  – множина властивостей інструментів,  $Param_{Osn}$  – множина властивостей оснащення,  $Param_{Hum}$  – множина властивостей людей,  $Param_{Rb}$  – множина властивостей робіт,  $Param_{Storage}$  – множина властивостей накопичувачів,  $Param_{Conv}$  – множина властивостей конвеєрів, які описані у таблиці 2.1.

Звідси:

$$Param_{Obj} = Param_{Vr} \cup Param_{Ins} \cup Param_{Osn} \cup Param_{Hum} \cup Param_{Rb} \cup Param_{Storage} \cup Param_{Conv}. \quad (2.17)$$

З цих визначень випливає, що

$$\forall x \exists Param_{Obj} [Param_{Vr}(x)], \quad (2.18)$$

$$\forall x \exists Param_{Obj} [Param_{Ins}(x)], \quad (2.19)$$

$$\forall x \exists Param_{Obj} [Param_{Osn}(x)], \quad (2.20)$$

$$\forall x \exists Param_{Obj} [Param_{Hum}(x)], \quad (2.21)$$

$$\forall x \exists Param_{Obj} [Param_{Rb}(x)], \quad (2.22)$$

$$\forall x \exists Param_{Obj} [Param_{Storage}(x)], \quad (2.23)$$

$$\forall x \exists Param_{Obj} [Param_{Conv}(x)]. \quad (2.24)$$

Визначення 2. Кожен предмет простору пов'язаний будь-яким відношенням з іншим предметом:

$$\forall x \exists y (x \leftrightarrow y) \quad (2.25)$$

Визначення є вірним, так як кожен з предметів створений для взаємодії з іншими предметами.

Визначення 3. Всі об'єкти робочого простору ГІВС є впорядкованими по відношенню до інших:

$$\forall x \exists y, \{x, y\} \quad (2.26)$$

Визначення 4. Для кожного предмета робочої зони знайдеться інший предмет, який сумісний з першим в процесі роботи:

$$\forall x \exists y, x \cap y \quad (2.27)$$

Визначення є істинним, оскільки кожен з предметів роботизованої ділянки існує з метою участі в технологічному процесі.

Визначення 5. Існують такі предмети, які зумовлюють один одного в технологічному процесі.

$$\forall x \exists y, x \rightarrow y \quad (2.28)$$

Це відношення є окремим випадком виразу з визначення 4, тому воно існує між тими ж об'єктами.

Визначення 6. Існують одні й ті ж предмети.

$$\forall x \exists y \in x \equiv y \quad (2.29)$$

Визначення 7. Для кожного значення властивості  $param_i$  знайдеться рівне йому. У символічному вигляді цей вислів можна записати так:

$$\forall param_i \exists param_j ([param_i] = [param_j]) \quad (2.30)$$

Визначення 8. Для кожного числового значення властивості (крім максимального на кінцевій множині значень  $N$ ) можна знайти значення більше поточного.

$$\forall param_i \exists param_j ([param_i] < [param_j], param_i \in N, param_j \in N) \quad (2.31)$$

Також окрім робочого простору ГІВС, де працює робот, існує і інший – простір складського устаткування *Warehouse*, який взаємодіє з ГІВС і має набір характеристик, зокрема: геометричні параметри  $D_{Warehouse}(x_{Warehouse}, y_{Warehouse}, z_{Warehouse})$ , матеріали (заготовки, певні компоненти складання, тощо) *Material* та готові продукти *Product*, а також параметри транспортування *Transport*.

$$Warehouse = \langle D_{Warehouse}(x_{Warehouse}, y_{Warehouse}, z_{Warehouse}), Material, Product, Transport \rangle \quad (2.32)$$

В даній моделі параметр *Material* можна виразити набором матеріалів:

$$Material = (material_1, material_2, \dots, material_n). \quad (2.33)$$

Аналогічно і для продуктів та параметрів транспортування:

$$Product = (product_1, product_2, \dots, product_n), \quad (2.34)$$

$$Transport = (transport_1, transport_2, \dots, transport_n). \quad (2.35)$$

Таким чином, усі введені позначення об'єктів РП ГІВС та приналежної транспортувальної системи формують інформаційно-логічну модель роботизованої системи. Основним призначенням запропонованої моделі є збереження поточної інформації про стан об'єктів ГІВС та суміжних систем,

також підтримка виконання запитів щодо властивостей об'єктів, які входять у опис інформаційно-логічної моделі.

Наприклад, для робочого простору, що показаний на рис. 1.5, побудова інформаційно-логічної моделі буде формуватися на основі нижченаведених вхідних даних.

Існує робочий простір, що має геометричні розміри 70 x 25 x 10 м. У ньому присутні об'єкти, такі як люди, верстати та оснащення з різного роду інструментами, накопичувачі, підвісний кран-штабелер в якості системи переміщення вантажів між двома верстатами, або верстатом та шафою оснащення.

Верстати, що використовуються мають певні габаритні розміри (1.7 x 1.906 x 2.26 м для верстатів фірми ANS, 1.78 x 2.2 x 1.675 м для верстатів фірми ASM, що використовуються на виробництві).

Шафи оснащення має геометричні розміри 1.5 x 0.7 x 1.7 м. На них розташовані інструменти як загального призначення, так і спеціалізовані, що використовуються на верстатах.

Якщо інструмент важкий, то використовують кран-штабелер, що дозволяє переміщувати його між шафою оснащення і верстатом (10 м).

Операції по зміні вантажу, його переміщення, а також слідкування за верстатами виконує людина.

Після складання виробів, вони переміщуються на накопичувачі, звідки вже людина за допомогою візка доставляє їх на склад устаткування.

Слід вказати, що проміжки між об'єктами робочого простору є дуже широкими і доцільно використовувати для транспортувальних операцій мобільних роботів, що дозволять значно прискорити процес.

Гнучка виробнича ділянка на основі розробленої моделі представлена на рис. 2.2.



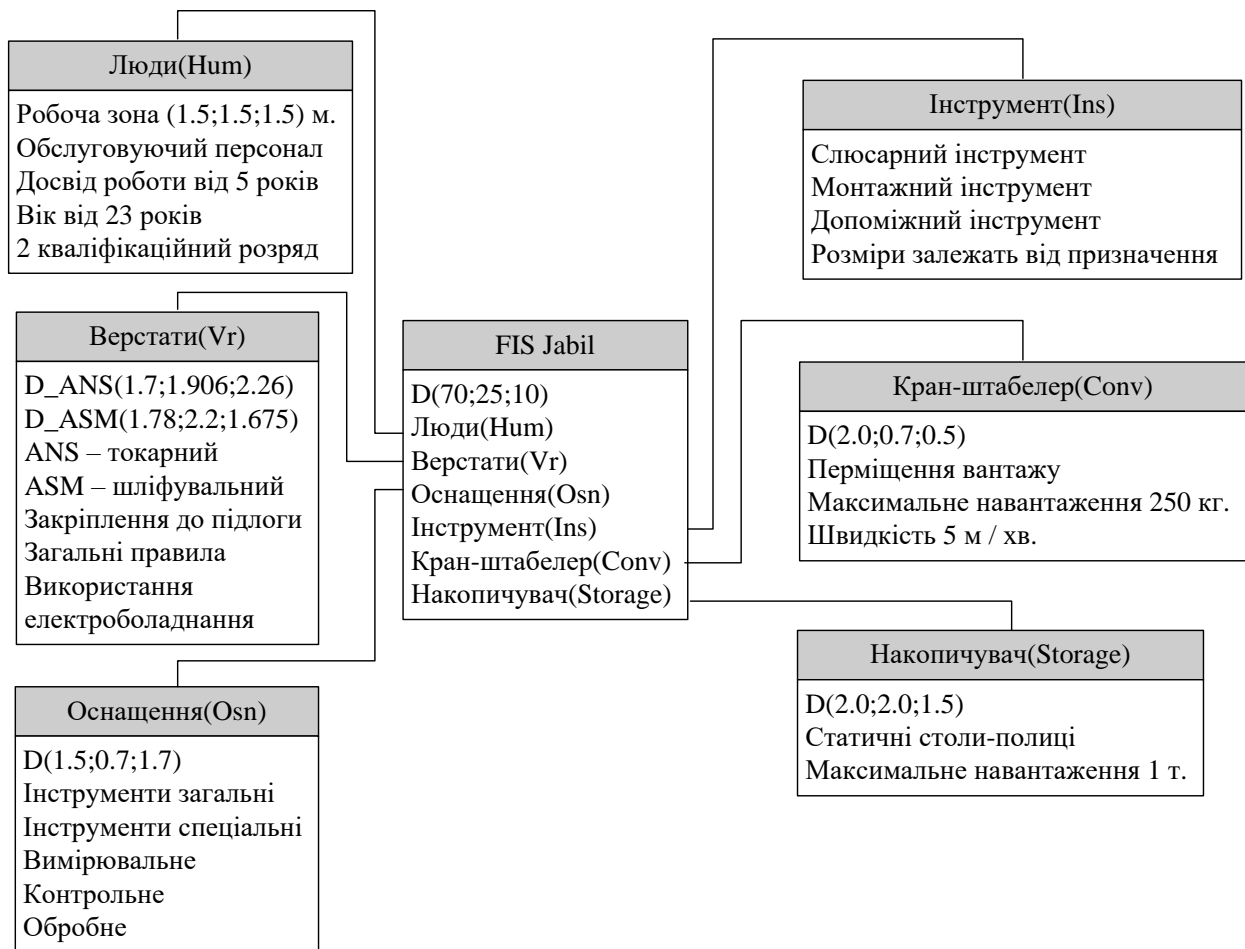


Рисунок 2.2 – Гнучка виробнича ділянка на основі розробленої моделі

## 2.2 Функціональна модель гнучкої інтегрованої системи

Крім оголошення властивостей об'єктів ГІВС важливим є виокремлення характеру взаємодії об'єктів на рівні виконання поставлених перед ГІВС виробничих, транспортувальних та інших завдань [76-77].

Між усіма об'єктами, описаними в п 2.1 існують певні функціональні залежності  $F$ , що забезпечують виконання:

- виробничих функцій  $F_{prod}$  ;
- транспортувальних функцій  $F_{trans}$  ;
- функцій керування  $F_{control}$  ;
- сервісних функцій  $F_{service}$  ;
- функцій моніторингу  $F_{Mon}$  .

Виробничі функції, що виконуються у *FIS*, можна представити у вигляді:

$$Product_i = F_{prod}(Material_i, Rb_i, Vr_i, Osn_i, Ins_i, Hum_i, Conv_i, Storage_i, t). \quad (2.36)$$

Тобто, для того, щоб зробити необхідний готовий продукт  $Product_i$  треба виконати набір функцій, наприклад:

- за допомогою робота  $Rb_i$  потрібно доставити матеріал  $Material_i$  на верстат  $Vr_i$ ;
- людина  $Hum_i$  за допомогою оснащення  $Osn_i$  та інструменту  $Ins_i$  складає продукт  $Product_i$ ;
- готовий продукт  $Product_i$  за допомогою конвеєра  $Conv_i$  переміщується на накопичувач  $Storage_i$ .

Схематично цю функцію можна представити у вигляді *idef0*-діаграм (рис. 2.3 – 2.5).

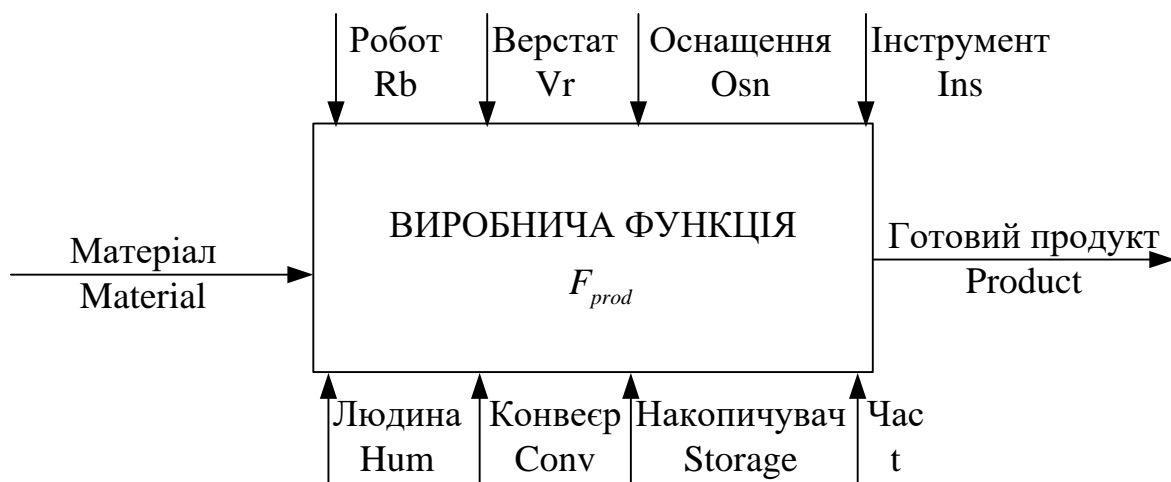


Рисунок 2.3 – Виробнича функція ГІВС

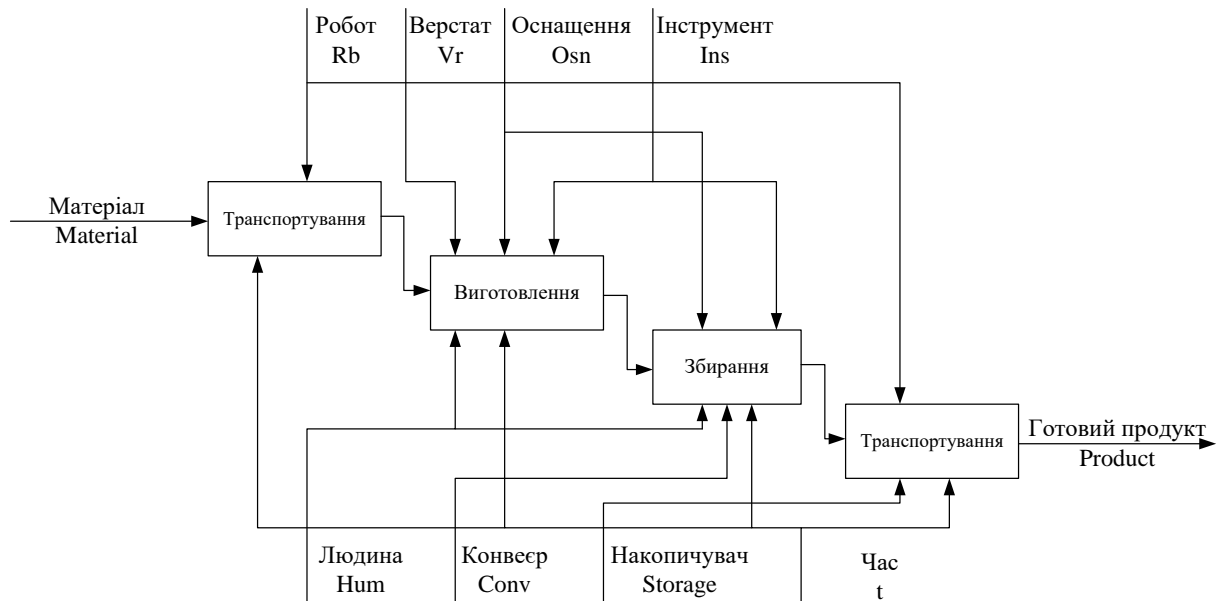


Рисунок 2.4 – Детальний опис виробничої функції ГІВС

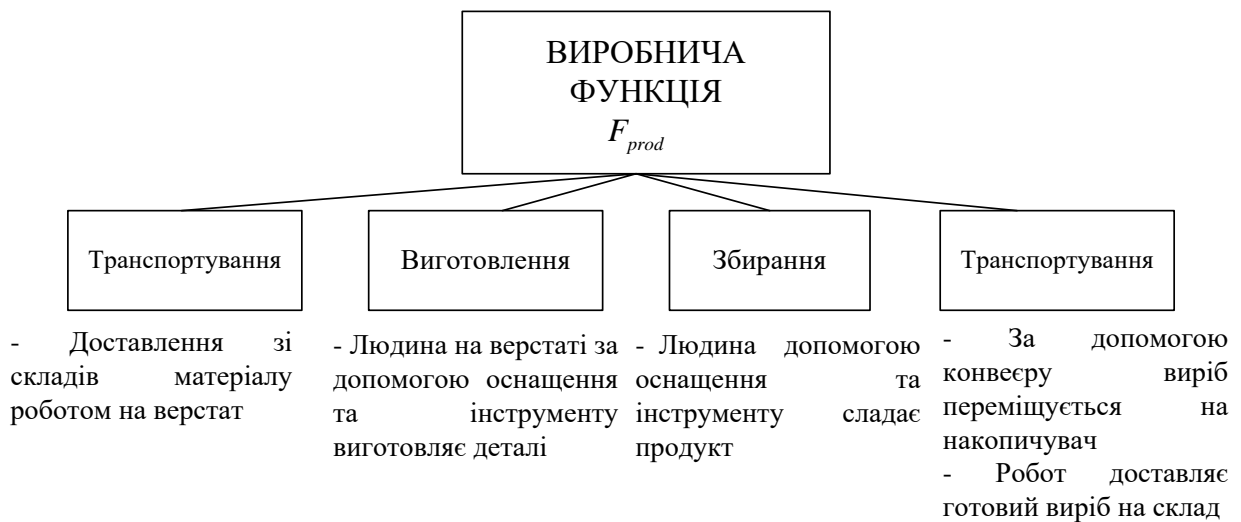


Рисунок 2.5 – Декомпозиція виробничої функції ГІВС

Транспортувальні функції можна представити у вигляді:

$$Transport_i = F_{trans}(Product_i, Rb_i, Material_i, Conv_i, Storage_i, t, vlc) \quad (2.37)$$

Основними елементами, що задіяні у транспортувальній функції є робот  $Rb_i$ , що виконує переміщення матеріалів  $Material_i$  та готових виробів  $Product_i$  з накопичувачів  $Storage_i$ , а також конвеєр  $Conv_i$ , який доставляє готові вироби

$Product_i$  на ці накопичувачі  $Storage_i$ . Важливими характеристиками транспортувальної функції є також час  $t$  і швидкість  $v/c$ .

Схематично транспортувальну функцію можна представити наступним чином (рис. 2.6–2.7).

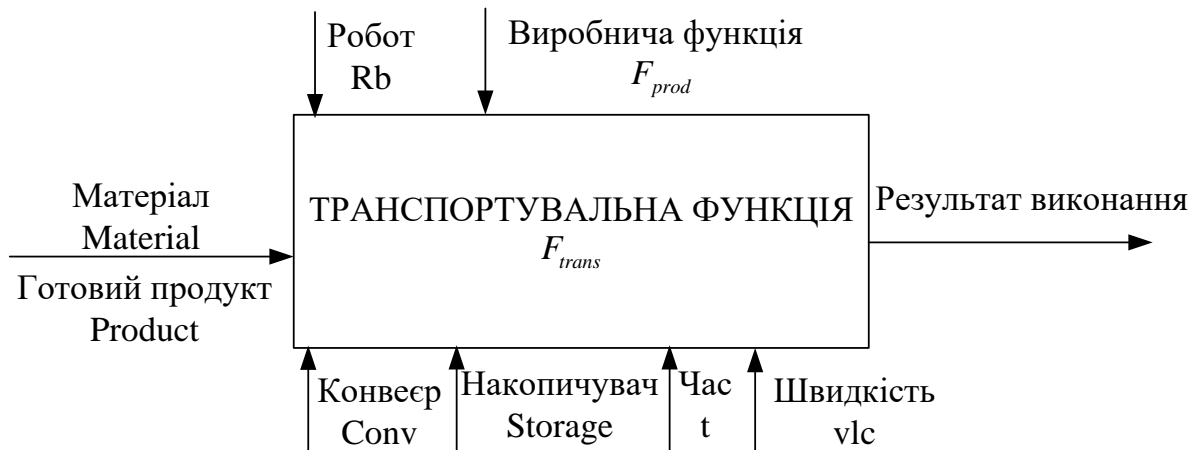


Рисунок 2.6 – Транспортувальна функція ГІВС



- залежить від виробничого завдання
- робот переміщує матеріали та готові вироби між складом та накопичувачами
- конвеєр доставляє готові вироби на накопичувачі
- зменшення часу переміщення
- збільшення швидкості переміщення

Рисунок 2.7 – Декомпозиція транспортувальної функції ГІВС

Параметри керування  $Control$  можна записати у вигляді:

$$Control_i = F_{control}(Product_i, Transport_i, Rb_i, Material_i, Conv_i, Storage_i, Hum_i, Ins_i, Osn_i, Warehouse_i, t) \quad (2.38)$$

Тобто параметри керування будуть залежати від усіх параметрів, що використовуються у загальній системі. Схематично функцію керування можна представити у наступному вигляді (рис. 2.8–2.9).

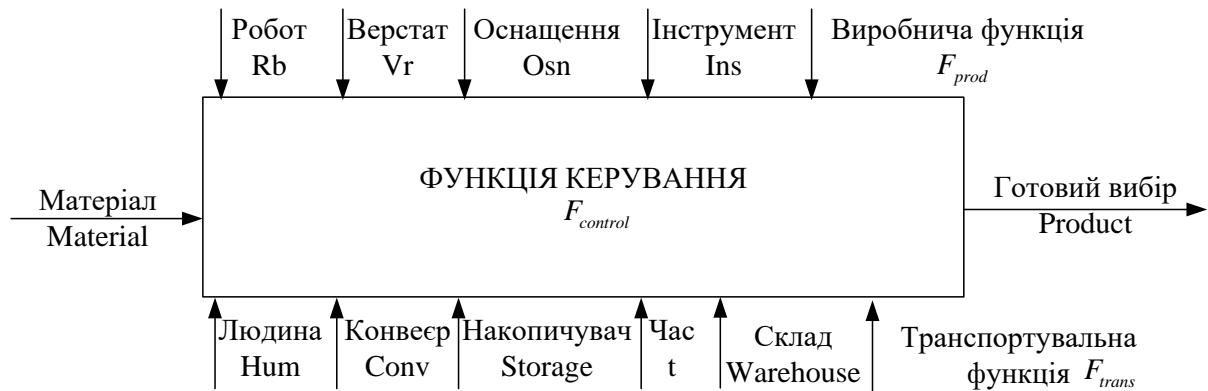


Рисунок 2.8 – Функція керування ГІВС



- залежить від виробничого завдання
- залежить від транспортувальних завдань
- залежить від усіх елементів, що присутні у ГІС
- формує керуючі впливи на основі інформації від заданих функцій системи

Рисунок 2.9 – Декомпозиція функції керування ГІВС

Параметри сервісної функції мають такий вигляд:

$$Service_i = F_{service}(Rb_i, Conv_i, Storage_i, Hum_i, Osn_i, t) \quad (2.39)$$

Сервісна функція означає виконання операцій обслуговування для таких об'єктів, як роботи  $Rb$ , конвеєри  $Conv$ , накопичувачі  $Storage$ , люди  $Hum$ , оснащення  $Osn$ , який схематично матиме наступний вигляд (рис. 2.10–2.11).

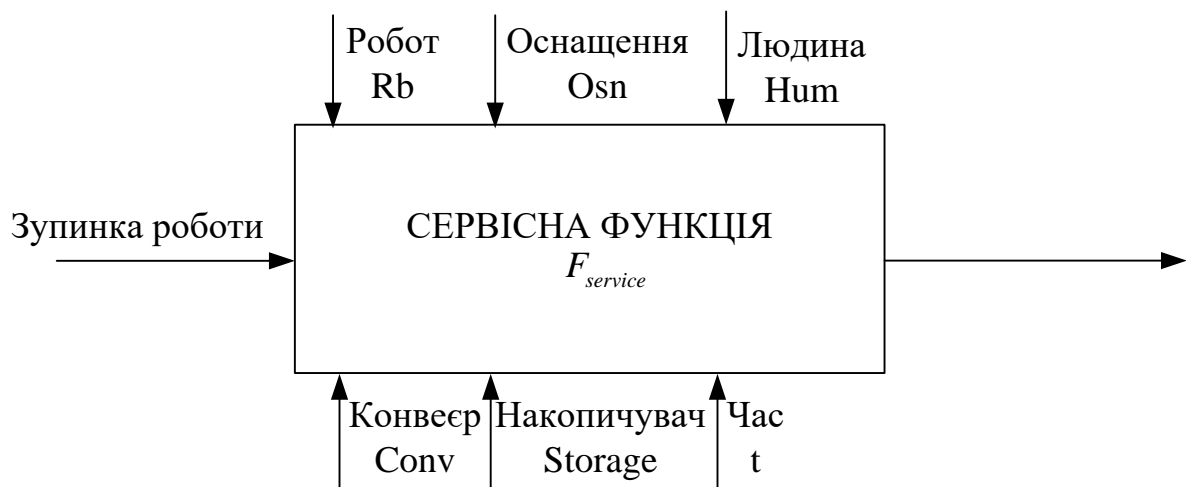


Рисунок 2.10 – Сервісна функція ГІВС



- виконується у разі відмови виробничого обладнання/ мобільного роботу
- виконується у разі травмування людини
- виконується з найменшими втратами часу

Рисунок 2.11 – Декомпозиція сервісної функції ГІВС

Аргументами функції моніторингу виступають:

$$Mon_i = F_{Mon}(\forall Obj, Obj \in W_s, t) \quad (2.40)$$

Схематично можна представити у вигляді (рис. 2.12–2.13).

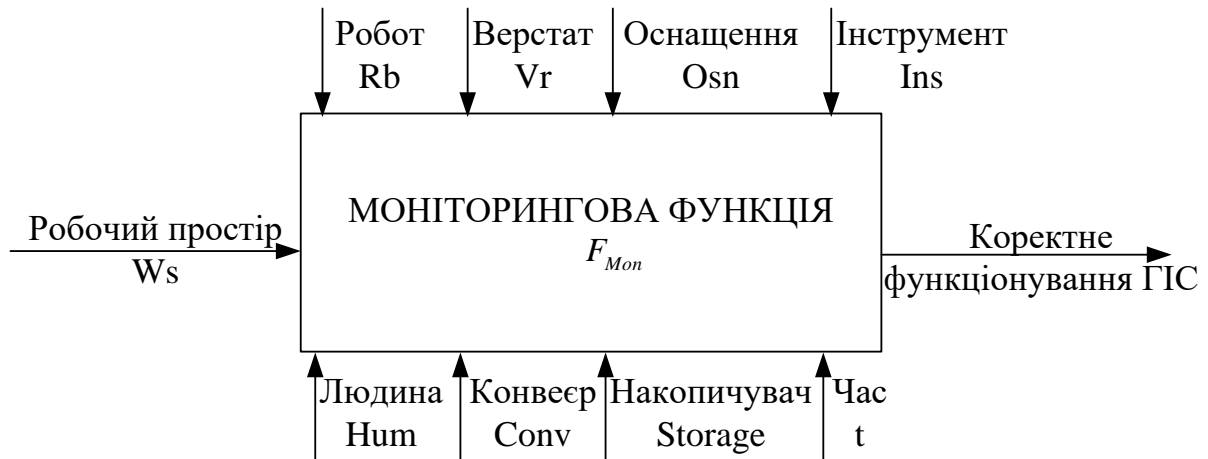


Рисунок 2.12 – Сервісна функція ГІВС



- спостереження за положенням, виконанням виробничих, транспортувальних, сервісних функцій, функцій керування
- спостереження за працездатністю технологічного обладнання
- спостереження за працездатністю роботів, інструментів
- спостереження за діями людей у робочому просторі ГІС

Рисунок 2.13 – Декомпозиція сервісної функції ГІВС

Виконання функції моніторингу означає спостереження за положенням, виконанням виробничих, транспортувальних, сервісних функцій, функцій керування, спостереження за працездатністю технологічного обладнання, роботів, інструментів та діями людей у робочому просторі ГІВС.

Використовуючи параметри двох розроблених моделей (інформаційно-логічної та функціональної), ці вирази можна записати наступним чином.

Виробнича функція:

$$\begin{aligned}
 Product_i = F_{prod} & ((material_1^i, material_2^i, material_3^i, \dots, material_n^i), (D_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), \\
 Mv_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), Speed_{Rb}^i, Cp_{Rb}^i(x_{cpRb}^i, y_{cpRb}^i, z_{cpRb}^i), ID_{Rb}^i), (Type_{Vr}^i, D_{Vr}^i(x_{Vr}^i, y_{Vr}^i, z_{Vr}^i), \\
 PM_{Vr}^i, Sc_{Vr}^i, ID_{Vr}^i), (Type_{Ins}^i, D_{Ins}^i(x_{Ins}^i, y_{Ins}^i, z_{Ins}^i), PM_{Ins}^i, ID_{Ins}^i), (Type_{Osn}^i, \\
 D_{Osn}^i(x_{Osn}^i, y_{Osn}^i, z_{Osn}^i), ID_{Osn}^i), (D_{Hum}^i(x_{Hum}^i, y_{Hum}^i, z_{Hum}^i), Mv_{Hum}^i(x_{HumMv}^i, y_{HumMv}^i, z_{HumMv}^i), \\
 Per_{Hum}^i, Exp_{Hum}^i, Age_{Hum}^i, Qual_{Hum}^i, ID_{Hum}^i), (D_{Conv}^i(x_{Conv}^i, y_{Conv}^i, z_{Conv}^i), Type_{Conv}^i, Quan_{Conv}^i, \\
 Speed_{Conv}^i, ID_{Conv}^i), (D_{Storage}^i(x_{Storage}^i, y_{Storage}^i, z_{Storage}^i), Type_{Storage}^i, Quan_{Storage}^i, ID_{Storage}^i), t)
 \end{aligned} \quad (2.41)$$

Транспортна функція:

$$\begin{aligned}
 Transport_i = F_{trans} & ((material_1^i, material_2^i, material_3^i, \dots, material_n^i), (D_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), \\
 Mv_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), Speed_{Rb}^i, Cp_{Rb}^i(x_{cpRb}^i, y_{cpRb}^i, z_{cpRb}^i), ID_{Rb}^i), (Type_{Vr}^i, D_{Vr}^i(x_{Vr}^i, y_{Vr}^i, z_{Vr}^i), \\
 PM_{Vr}^i, Sc_{Vr}^i, ID_{Vr}^i), (Type_{Ins}^i, D_{Ins}^i(x_{Ins}^i, y_{Ins}^i, z_{Ins}^i), PM_{Ins}^i, ID_{Ins}^i), (Type_{Osn}^i, \\
 D_{Osn}^i(x_{Osn}^i, y_{Osn}^i, z_{Osn}^i), ID_{Osn}^i), (D_{Hum}^i(x_{Hum}^i, y_{Hum}^i, z_{Hum}^i), Mv_{Hum}^i(x_{HumMv}^i, y_{HumMv}^i, z_{HumMv}^i), \\
 Per_{Hum}^i, Exp_{Hum}^i, Age_{Hum}^i, Qual_{Hum}^i, ID_{Hum}^i), (D_{Conv}^i(x_{Conv}^i, y_{Conv}^i, z_{Conv}^i), Type_{Conv}^i, Quan_{Conv}^i, \\
 Speed_{Conv}^i, ID_{Conv}^i), (D_{Storage}^i(x_{Storage}^i, y_{Storage}^i, z_{Storage}^i), Type_{Storage}^i, Quan_{Storage}^i, ID_{Storage}^i), \\
 (D_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), Mv_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), Speed_{Rb}^i, Cp_{Rb}^i(x_{cpRb}^i, y_{cpRb}^i, z_{cpRb}^i), ID_{Rb}^i), \\
 (material_1^i, material_2^i, material_3^i, \dots, material_n^i), (D_{Conv}^i(x_{Conv}^i, y_{Conv}^i, z_{Conv}^i), Type_{Conv}^i, \\
 Quan_{Conv}^i, Speed_{Conv}^i, ID_{Conv}^i), (D_{Storage}^i(x_{Storage}^i, y_{Storage}^i, z_{Storage}^i), Type_{Storage}^i, Quan_{Storage}^i, \\
 ID_{Storage}^i), t, vlc)
 \end{aligned} \quad (2.42)$$



### Функція керування:

$$\begin{aligned}
Control_i = & F_{control}((material_1^i, material_2^i, material_3^i, \dots, material_n^i), (D_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), \\
& Mv_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), Speed_{Rb}^i, Cp_{Rb}^i(x_{cpRb}^i, y_{cpRb}^i, z_{cpRb}^i), ID_{Rb}^i), (Type_{Vr}^i, D_{Vr}^i(x_{Vr}^i, y_{Vr}^i, z_{Vr}^i), \\
& PM_{Vr}^i, Sc_{Vr}^i, ID_{Vr}^i), (Type_{Ins}^i, D_{Ins}^i(x_{Ins}^i, y_{Ins}^i, z_{Ins}^i), PM_{Ins}^i, ID_{Ins}^i), (Type_{Osn}^i, \\
& D_{Osn}^i(x_{Osn}^i, y_{Osn}^i, z_{Osn}^i), ID_{Osn}^i), (D_{Hum}^i(x_{Hum}^i, y_{Hum}^i, z_{Hum}^i), Mv_{Hum}^i(x_{HumMv}^i, y_{HumMv}^i, z_{HumMv}^i), \\
& Per_{Hum}^i, Exp_{Hum}^i, Age_{Hum}^i, Qual_{Hum}^i, ID_{Hum}^i), D_{Conv}^i(x_{Conv}^i, y_{Conv}^i, z_{Conv}^i), Type_{Conv}^i, Quan_{Conv}^i, \\
& Speed_{Conv}^i, ID_{Conv}^i), (D_{Storage}^i(x_{Storage}^i, y_{Storage}^i, z_{Storage}^i), Type_{Storage}^i, Quan_{Storage}^i, ID_{Storage}^i), \\
& (material_1^i, material_2^i, material_3^i, \dots, material_n^i), (D_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), \\
& Mv_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), Speed_{Rb}^i, Cp_{Rb}^i(x_{cpRb}^i, y_{cpRb}^i, z_{cpRb}^i), ID_{Rb}^i), (Type_{Vr}^i, D_{Vr}^i(x_{Vr}^i, y_{Vr}^i, z_{Vr}^i), \\
& PM_{Vr}^i, Sc_{Vr}^i, ID_{Vr}^i), (Type_{Ins}^i, D_{Ins}^i(x_{Ins}^i, y_{Ins}^i, z_{Ins}^i), PM_{Ins}^i, ID_{Ins}^i), (Type_{Osn}^i, \\
& D_{Osn}^i(x_{Osn}^i, y_{Osn}^i, z_{Osn}^i), ID_{Osn}^i), (D_{Hum}^i(x_{Hum}^i, y_{Hum}^i, z_{Hum}^i), Mv_{Hum}^i(x_{HumMv}^i, y_{HumMv}^i, z_{HumMv}^i), \\
& Per_{Hum}^i, Exp_{Hum}^i, Age_{Hum}^i, Qual_{Hum}^i, ID_{Hum}^i), (D_{Conv}^i(x_{Conv}^i, y_{Conv}^i, z_{Conv}^i), Type_{Conv}^i, Quan_{Conv}^i, \\
& Speed_{Conv}^i, ID_{Conv}^i), (D_{Storage}^i(x_{Storage}^i, y_{Storage}^i, z_{Storage}^i), Type_{Storage}^i, Quan_{Storage}^i, ID_{Storage}^i), \\
& (D_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), Mv_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), Speed_{Rb}^i, Cp_{Rb}^i(x_{cpRb}^i, y_{cpRb}^i, z_{cpRb}^i), ID_{Rb}^i), \\
& (material_1^i, material_2^i, material_3^i, \dots, material_n^i), (D_{Conv}^i(x_{Conv}^i, y_{Conv}^i, z_{Conv}^i), Type_{Conv}^i, \\
& Quan_{Conv}^i, Speed_{Conv}^i, ID_{Conv}^i), (D_{Storage}^i(x_{Storage}^i, y_{Storage}^i, z_{Storage}^i), Type_{Storage}^i, Quan_{Storage}^i, \\
& ID_{Storage}^i), (D_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), Mv_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), Speed_{Rb}^i, Cp_{Rb}^i(x_{cpRb}^i, y_{cpRb}^i, z_{cpRb}^i), ID_{Rb}^i), \\
& (Type_{Vr}^i, D_{Vr}^i(x_{Vr}^i, y_{Vr}^i, z_{Vr}^i), PM_{Vr}^i, Sc_{Vr}^i, ID_{Vr}^i), (Type_{Ins}^i, D_{Ins}^i(x_{Ins}^i, y_{Ins}^i, z_{Ins}^i), PM_{Ins}^i, ID_{Ins}^i), \\
& (Type_{Osn}^i, D_{Osn}^i(x_{Osn}^i, y_{Osn}^i, z_{Osn}^i), ID_{Osn}^i), (D_{Hum}^i(x_{Hum}^i, y_{Hum}^i, z_{Hum}^i), Mv_{Hum}^i(x_{HumMv}^i, y_{HumMv}^i, z_{HumMv}^i), \\
& Per_{Hum}^i, Exp_{Hum}^i, Age_{Hum}^i, Qual_{Hum}^i, ID_{Hum}^i), D_{Conv}^i(x_{Conv}^i, y_{Conv}^i, z_{Conv}^i), Type_{Conv}^i, Quan_{Conv}^i, \\
& Speed_{Conv}^i, ID_{Conv}^i), (D_{Storage}^i(x_{Storage}^i, y_{Storage}^i, z_{Storage}^i), Type_{Storage}^i, Quan_{Storage}^i, ID_{Storage}^i), \\
& (material_1^i, material_2^i, material_3^i, \dots, material_n^i), t, vlc)
\end{aligned} \quad (2.43)$$

### Сервісна функція:

$$\begin{aligned}
Service_i = & F_{service}((D_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), Mv_{Rb}^i(x_{Rb}^i, y_{Rb}^i, z_{Rb}^i), Speed_{Rb}^i, \\
& Cp_{Rb}^i(x_{cpRb}^i, y_{cpRb}^i, z_{cpRb}^i), ID_{Rb}^i), Conv_i, Storage_i, Hum_i, Osn_i, t), \\
& (D_{Conv}^i(x_{Conv}^i, y_{Conv}^i, z_{Conv}^i), Type_{Conv}^i, Quan_{Conv}^i, Speed_{Conv}^i, ID_{Conv}^i), \\
& (D_{Storage}^i(x_{Storage}^i, y_{Storage}^i, z_{Storage}^i), Type_{Storage}^i, Quan_{Storage}^i, ID_{Storage}^i), \\
& (Type_{Osn}^i, D_{Osn}^i(x_{Osn}^i, y_{Osn}^i, z_{Osn}^i), ID_{Osn}^i), (D_{Hum}^i(x_{Hum}^i, y_{Hum}^i, z_{Hum}^i), \\
& Mv_{Hum}^i(x_{HumMv}^i, y_{HumMv}^i, z_{HumMv}^i), Per_{Hum}^i, Exp_{Hum}^i, Age_{Hum}^i, Qual_{Hum}^i, \\
& ID_{Hum}^i), t)
\end{aligned} \quad (2.44)$$

Моніторингова функція:

$$\begin{aligned}
 Mon_i = F_{Mon}(\{ & Camera_1, Camera_2, \dots, Camera_n \} \times \{ Vr, Ins, Osn, Hum, Rb, Storage, \\
 Conv \} = (\{ & Camera_0 \times Vr, Camera_0 \times Ins, Camera_0 \times Osn, Camera_0 \times Hum, \\
 Camera_0 \times Hum, & Camera_0 \times Rb, Camera_0 \times Storage, Camera_0 \times Conv, \\
 Camera_1 \times Vr, & Camera_1 \times Ins, Camera_1 \times Osn, Camera_1 \times Hum, Camera_1 \times Hum, \\
 Camera_1 \times Rb, & Camera_1 \times Storage, Camera_1 \times Conv, Camera_2 \times Vr, Camera_2 \times Ins, \cdot \\
 Camera_2 \times Osn, & Camera_2 \times Hum, Camera_2 \times Hum, Camera_2 \times Rb, \\
 Camera_2 \times Storage, & Camera_2 \times Conv, \dots, Camera_n \times Vr, Camera_n \times Ins, \\
 Camera_n \times Osn, & Camera_n \times Hum, Camera_n \times Hum, Camera_n \times Rb, \\
 Camera_n \times Storage, & Camera_n \times Conv, \})
 \end{aligned} \quad (2.45)$$

Виконання ГІВС своїх функцій означає виконання усіх завдань керування  $F_{FIS}$ , виробничих, транспортувальних, моніторингових, сервісних функцій та функцій керування:

$$F_{FIS} = \langle F_{prod}, F_{trans}, F_{control}, F_{Service}, F_{Mon} \rangle \quad (2.46)$$

При цьому вартість  $C_{FIS}$  та час виконання кожної функції ГІВС мають бути мінімальними за умови максимізації якості  $Q_{FIS}$  виконання завдань:

$$C_{FIS} = \min[C(F_{prod})] \cup \min[C(F_{trans})] \cup \min[C(F_{control})] \cup \min[C(F_{Service})] \cup \min[C(F_{Mon})] \quad (2.47)$$

$$\begin{aligned}
 Q_{FIS} = \max[Q(F_{prod})] \cup \max[Q(F_{trans})] \cup \max[Q(F_{control})] \\
 \cup \max[C(F_{Service})] \cup \max[C(F_{Mon})]
 \end{aligned} \quad (2.48)$$

Таким чином, функціональна модель ГІВС забезпечує опис виконання необхідних виробничих та транспортувальних функцій, також пов'язаних з ними функцій керування об'єктами ГІВС. Виходячи з наявного переліку функцій на кожному кроці роботи системи визначатимуться завдання

інтелектуальної системи керування, порядок виконання яких визначатиметься загальною метою ГВС та закономірностями її роботи [76].

Приклад реального виробництва компанії Jabil представлено на рис. 2.14.



Рисунок 2.14 – Приклад реального виробництва

На основі аналізу реального виробництва та базуючись на запропонованій моделі можна визначити наступні функції.

Виробнича функція полягає у наступному. Робот переміщує матеріали та заготовки (в даному випадку для транспортних засобів) на верстати ЧПК / оброблювальні центри, що керуються людиною, де за допомогою необхідного оснащення та інструменту виготовляються або складаються деталі та готові вироби. Далі готові вироби за допомогою мобільного робота відправляються на склад. Основним критерієм тут є час виконання цих дій і він повинен бути найнижчим.

Транспортувальна функція полягає у тому, що робот переміщує матеріали та готові вироби між складом та робочим простором, а також між об'єктами робочого простору. Головні критерії тут – мінімальний час переміщення і максимальна швидкість доставки.

Функція керування полягає в тому, що усі об'єкти в робочій зоні керуються відповідно до завдань, що повинні бути виконані відповідно до

графіку (виробнича функція). Вона залежить від того, з якою швидкістю мобільний робот доставить матеріали/готові вироби (транспортувальна функція). У випадках, коли елементи виробничої системи виходять з ладу або виникають нештатні ситуації, обумовлені людським фактором, має активуватися сервісна функція, виконання якої полягатиме у виклику сервісної або медичної бригад для швидкого відновлення роботи ГІВС [77].

Також, однією з найголовніших функцій є моніторингова, яка технічно забезпечується наявністю камер спостереження і дозволяє здійснювати керування виробництвом на рівні виконання процедур спостереження за виконанням виробничих, транспортувальних та сервісних функцій.

### 2.3 Висновки до розділу 2

У другому розділі було розроблено інформаційно-логічну та функціональну моделі робочого простору.

Основним призначенням інформаційно-логічної моделі є збереження поточної інформації про стан об'єктів ГІВС та суміжних систем, також підтримка виконання запитів щодо властивостей об'єктів, які входять у опис моделі, що на відміну від існуючих моделей враховує дискретність та заповненість виробничого робочого простору; ґрунтується на інформації, отриманій від об'єктової системи комп'ютерного зору; визначає та враховує властивості об'єктів, що розміщуються в робочому просторі; враховує взаємодію, упорядкованість та сумісність об'єктів.

На основі інформаційно-логічної моделі створено функціональну модель гнучкої інтегрованої системи, що відображає зв'язки об'єктів ГІВС та їх взаємодію і складається з виробничих, транспортувальних, керуючих, сервісних та моніторингових функцій [78].

Основні результати цього розділу опубліковано в роботах [67], [71], [72], [76-77], [78].

## РОЗДІЛ 3

### РОЗРОБКА МОДЕЛЕЙ ТА МЕТОДУ АДАПТИВНОГО ВІЗУАЛЬНОГО КЕРУВАННЯ МОБІЛЬНИМИ ТРАНСПОРТУВАЛЬНИМИ РОБОТАМИ В ГІВС

3.1 Визначення основних вимог до моделей адаптивного візуального керування мобільним роботом

#### 3.1.1 Візуальне цілевказання

Система керування, яка буде враховувати зміни робочого середовища і станів гнучкою виробничої інтегрованої системи повинна здійснювати контроль за умовами виконання завдання і, при необхідності, адаптувати процес виконання виробничих функцій РТС [65], [70].

В якості такої системи може виступати система адаптивного візуального керування. Впровадження такої системи адаптації має істотно поліпшити характеристики систем керування робототехнічними системами, що входять до складу ГІВС.

Головними проблемами будь-якого гнучкого виробництва в наш час є автоматизовані транспортні операції, як усередині одного цеху (однозонова система), так і між цехами (багатозонові системи). Ці цехи мають досить широкі відстані між виробничим обладнанням і широкі шляхи для переміщення вантажів. З цього випливає можливість інтеграції для транспортних операцій гнучких комп'ютеризованих виробничих модулів - інтелектуальних робототехнічних об'єктів.

Такі модулі будуть складатися з мобільних робототехнічних платформ і систем сприйняття, встановлених як об'єктові (системи комп'ютерного зору), так і локально – на самій робототехнічній платформі (GPS, далекоміри і ін.). Застосування даного роду систем дозволить значно збільшити швидкість

виконання транспортних операцій, що позитивно відіб'ється і на самому виробництві теж.

Переваги застосування систем комп'ютерного зору полягають в дешевизні використовуваного обладнання – відсутності необхідності використання для орієнтування в робочому просторі величезної кількості дорогих датчиків.

Відповідно до динамічного характеру робочого простору інтелектуального робототехнічного об'єкта (робота), система візуального керування повинна забезпечити аналіз робочого простору для переміщення транспортно-складального робота, що визначається умовами механоскладального виробництва, висуває вимоги до адаптації, які повинні забезпечити підвищення стабільності роботи гнучких інтегрованих виробничих систем.

Система візуального керування інтелектуального робототехнічного об'єкта повинна забезпечувати наступні можливості [73-74]:

- роботу з камерою / камерами (підключення та налаштування);
- аналіз та обробку зображень;
- класифікацію об'єктів робочої зони з результатів аналізу зображень;
- формування моделі руху об'єктів робочої зони;
- прогнозування подальшої траєкторії руху об'єктів;
- виділення непрохідних ділянок;
- формування маршруту;
- подачу керуючих впливів;
- оцінку помилок навігації;
- усунення помилок навігації.

Візуальне цілевказання буде складатися з:

- отримання інформації про робочий простір (геометричні параметри);
- отримання інформації про робочий простір камери (параметри зображення, зчитує за допомогою СКЗ);

- перетворення просторових координат в координати простору камери (СКЗ);
- розпізнавання та ідентифікацію робота в просторі (як в робочому просторі, так і в просторі камери);
- вказівки початкової точки пересування на зображенні, отриманому за допомогою СКЗ;
- вказівки кінцевої точки пересування на зображенні, отриманому за допомогою СКЗ.

Розглянемо найпростіший випадок, коли інформацію про параметри робочого простору отримуємо за допомогою вимірювальних пристроїв.

У робочому просторі існує одна статична СКЗ (в даному випадку в якості СКЗ виступає звичайна Web-камера), розташована об'єктивно (камера фіксується над робочим простором робота).

З цього можна зробити наступні висновки:

- площина зображення робота паралельна площині руху;
- камера може отримувати зображення з усієї робочої області;
- камера може обчислювати центри мас мобільного робота на основі розпізнавання деяких фізичних характеристик;
- камера може визначати орієнтацію мобільного робота, і таким чином, напрямок переміщення МР за допомогою розпізнавання додаткових характеристик.

Під параметрами зображення розуміємо його розподільну здатність (наприклад, 800x600 пікселів).

Перетворення просторових координат в координати простору камери (СКЗ) полягатиме в математичному зіставленні геометричних параметрів робочого простору з розмірами зображення (наприклад 1 метру робочого простору буде відповідати 150 пікселів зображення).

Далі необхідно провести розпізнавання та ідентифікацію робота.

Загалом процес розпізнавання та ідентифікації буде полягати в наступному.

Процес буде розділятися на декілька етапів: первинне розпізнавання, збір статистики, кластеризація зібраної статистики, формування еталонів (бази характеристик), дорозпізнавання.

Визначимо кожен з названих етапів:

- первинне розпізнавання означає розпізнавання всієї робочої зони.
- збір статистики – процес відбору надійно розпізнаних об'єктів, які згодом складуть навчальну вибірку для адаптивного алгоритму;
- кластеризація – розбиття навчальної вибірки на кластери (класи), за допомогою такого розбиття уточнюються результати розпізнавання, отримані на етапі первинного розпізнавання, буде виявлена статистична структура робочої зони;
- формування еталонів це створення остаточних, довічних наборів даних (бази характеристик), за якими буде проводитися дорозпізнавання;
- дорозпізнавання – другий прохід розпізнавання по всій робочій зоні з метою уточнити результати первинного розпізнавання, виставити адекватні оцінки точності, дорозпізнати то, що було не розпізнано раніше, відзначити ненадійно розпізнані об'єкти.

Розпізнавання і ідентифікацію робота можна провести, використовуючи, наприклад, каскадний класифікатор Хаара, побудований за методом Віюлі і Джонса. Класифікатор Хаара має формат, відмінний від решти бібліотеки машинного навчання, оскільки він був розроблений раніше в якості повноцінного додатка розпізнавання особи. Для розробки детектора робота слід розглянути класифікатор в деталях і показати, як можна навчитися розпізнавати об'єкти, що знаходяться в робочому просторі робота.

Проводиться навчання на кількох сотнях видів одного зразка конкретного об'єкта (мобільний робот), які називаються позитивними зразками [101].



Дані зображення приводяться до одного стандартного вигляду (параметри освітленості, положення об'єкта і розмір). Окремо вибираються негативні зразки: довільні зображення того ж розміру, які не містять цільовий об'єкт.

По завершенню попереднього навчання класифікатора, можливе застосування його до області інтересу (розмір відповідає розміру при навчанні) у вхідному образі.

Класифікатор приймає значення 1 якщо область, з певною ймовірністю, є цільовим об'єктом (мобільним роботом), 0 – в іншому випадку. Перевагою такого класифікатора є можливість застосування його до об'єктів різного розміру без попереднього масштабування. Для цього процедура сканування повинна бути проведена кілька разів з різними ваговими коефіцієнтами.

Класифікатор складається з певної кількості простих класифікаторів (попередні етапи створення класифікатора), які послідовно застосовуються до області інтересу, до того моменту, поки всі етапи не будуть закінчені успішно. Основні класифікатори є деревом рішень, не менше ніж з 2 рівнями. Даний алгоритм використовує такі Хаара-подібні характеристики (рис. 3.1).

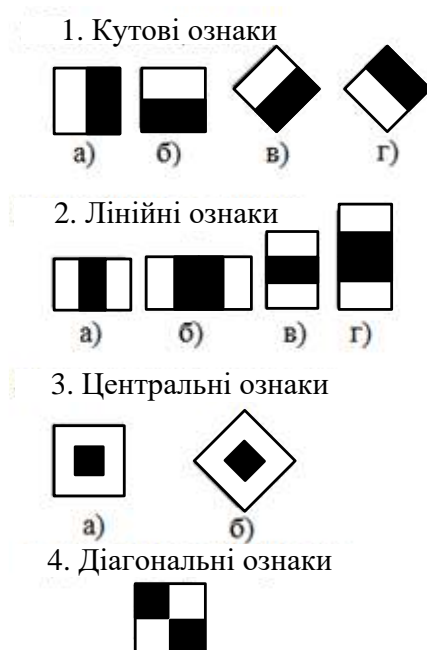


Рисунок 3.1– Хаар-ознаки виділення об'єктів

Характеристика, використана в класифікаторі, визначається формою (1а або 2б тощо), позицією в межах області інтересу і коефіцієнтом.

Наприклад, в разі третьої характеристики рядки (2в) відповіддю є відмінність між сумою пікселів образу під прямокутником, що покриває цілу характеристику (включаючи дві білих смуги і чорну смугу в середині) і сумою пікселів образу під чорною смугою помноженої на 3 (компенсація розміру області).

Нормалізація зображення складається з масштабування об'єкта і приведення його до конкретного розміру (розміром фотографії на яких проводилося навчання системи). В якості алгоритму масштабування найчастіше вибирають білінійну інтерполяцію.

Таким чином, даний алгоритм дає хорошу якість зображення при масштабуванні і вимагає мінімальної кількості ресурсів, що важливо при вирішенні даного завдання.

Ще одним алгоритмом є виділення об'єктів певної форми на зображенні і стеження за ними [78].

Після отримання відеопослідовності з СКЗ виконується захоплення зображень (послідовність кадрів). Ця процедура здійснюється з пристроєм відеозахоплення, в ході якого повинна бути можливість контролю параметрів запису, а саме число фреймів і частота.

Далі відбуваються операції, пов'язані з обробкою зображень, морфологічні операції, аналіз і покращення зображень, відновлення зображень і інші. Після зчитування, зображення повинне проходити морфологічну обробку. В ході аналізу повинні бути виділені координати центру об'єкта (центри мас). На основі центру мас відбувається побудова мінімальної рамки, що обмежує об'єкт (мобільний робот).

На першому кадрі користувачем вказується об'єкт. Маючи значення пікселів вибраного об'єкту аналізуються наступні кадри і знаходиться об'єкт – виділяється активна область і обчислюється координати центру мас. Для подальшої роботи зображення переводиться в бінарне, далі виконується

заповнення пустот і видалення тих об'єктів, які не відповідають умові пошуку. В результаті отримуємо зображення з виділеним об'єктом, всі інші об'єкти стають фоном.

Далі вказується початкова точка переміщення  $(x, y, z)$  на зображенні, отриманому з СКЗ, потім кінцева точка з координатами  $(x_r, y_r, z_r)$ .

Після того, як робот отримає координати переміщення, вони надходять на вхід моделі.

### 3.1.2 Обробка зображень

При зміні положення мобільного робота за допомогою системи комп'ютерного зору (СКЗ) отримуємо зображення, яке необхідно буде обробити для подальшої інтерпретації.

Під обробкою зображення з його подальшою інтерпретацією можна розуміти такі визначення.

Основними перетвореннями, що використовувались для виділення об'єктів, є:

- перетворення об'єктів в півтонове;
- заповнення отворів на вихідному кольоровому зображенні;
- апроксимація зображення структурними елементами;
- відстеження зовнішніх кордонів об'єктів на зображенні;
- вимір певного набору характеристик для кожної області;
- обчислення центрів мас (центроїд) елементів;
- формування мінімальних обмежують прямокутників / кіл.

Для більш коректного виділення об'єктів на тлі необхідно перетворити зображення з кольорового в 8-канальний – знаходження вибірки певних областей зображення, які відрізняються від інших яскравістю. Найпростіший спосіб кодування точки зображення це бінарний  $(0,1)$ , тобто точка може знаходитися в двох станах чорна або біла.

При перетворенні об'єктів в півтонове можливе виникнення помилок – поява дірок на самих об'єктах, отже, потрібно їх заповнити. Це здійснюється за допомогою заливки фоновими пікселями вихідного зображення, починаючи з точок, визначених параметром. Діра являє собою набір фонових пікселів, які не можуть бути отримані шляхом заповнення фону від краю зображення.

Виділення меж об'єктів на зображенні можливо алгоритмом Кенні [102], який складається з п'яти окремих кроків:

- згладжування – розмиття зображення для видалення шуму;
- пошук градієнтів (межі відзначаються там, де градієнт зображення набуває максимальне значення);
- придушення немаксимумів (тільки локальні максимуми відзначаються як кордони);
- подвійна порогова фільтрація (потенційні межі визначаються порогоми);
- трасування області неоднозначності (підсумкові межі визначаються шляхом придушення всіх країв, незв'язаних з визначеними межами).

Далі проводиться створення плоских, дископодібних структурних елементів, де параметром є радіус. Він має бути невід'ємним дійсним числом. Параметр для апроксимації повинен приймати значення 0, 4, 6 або 8. Коли він більше 0, тоді структурний елемент апроксимується послідовністю цього числа періодично-лінійних структурних елементів. Коли параметр дорівнює 0, апроксимація не використовується і структурні елементи формуються з усіх пікселів, віддалених не більше ніж на від величину радіусу від центру. Порогові функції застосовуються при виділенні об'єктів для визначення порогу яскравості на зображенні.

Деякі завдання обробки зображення пов'язані з перетворенням півтонового в бінарне. Для скорочення інформаційної надмірності зображення, залишається тільки та інформація, яка потрібна для вирішення конкретного завдання. На бінарному зображенні повинні бути збережені

деталі, що цікавлять (наприклад, форми зображених об'єктів) і виключені несуттєві особливості (фон).

Апроксимація контуру дозволяє без істотної втрати інформації в рази зменшити кількість точок в контурі і значно прискорити контурний аналіз. Апроксимація може бути здійснена наступним алгоритмом:

- знаходяться дві найвіддаленіші один від одного точки контуру і дві частини контуру між ними розглядаються незалежно;
- для кожного отриманого ділянки контуру знаходиться точка найбільш віддалена від прямої з'єднує кінцеві точки ділянки;
- якщо відстань від точки, знайденої на кроці 2 до прямої більше заданого порогу, то ця частина контуру розбивається цією точкою на два менших ділянки;
- якщо знайдена відстань менше порогового значення, то замість цієї частини контуру в результуючий контур зберігається відрізок, що з'єднує кінцеві точки ділянки.

Порогова обробка півтонових зображень полягає в розбитті всіх елементів зображення на два класи за ознакою яскравості, тобто у виконанні поелементного перетворення.

Одним з методів порогової обробки є метод Оцу [103]. Цей метод дозволяє обчислити таке порогове значення, де дисперсія між об'єктом і фоном результатів буде мінімальною.

На основі вже отриманих контурів зображень можливо визначити їх контури – з нескінченності виділених крайових точок сформувати криві.

Після визначення контурів знаходимо центри мас об'єктів (центроїди) на зображенні – геометричні точки, що характеризують рух тіла або системи частинок як цілого.

Останнім етапом є виділення отриманого об'єкта таким прямокутником або окружністю, які будуть мінімально обмежувати об'єкт (об'єкти будуть вписані в прямокутник / коло).

### 3.1.3 Зшивка зображень

Розглянемо випадок, коли робочий простір є дуже великим і інформація про нього отримується з декількох камер [67].

Нехай  $\epsilon$  робочий простір інтелектуального робототехнічного об'єкта, оснащений набором відеокамер (систем комп'ютерного зору), причому робочі простори всіх камер перетинаються на деяких ділянках (рис. 3.2).

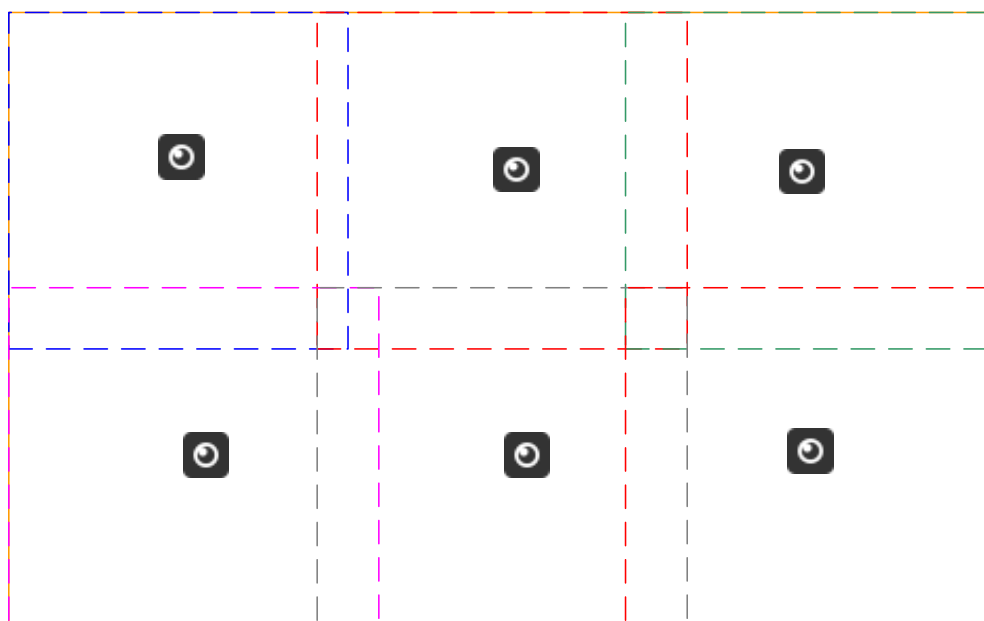


Рисунок 3.2– Робочий простір з розташованими в ньому камерами

На рисунку 3.2 кольорами позначені робочі простори, з яких камери отримують зображення.

Зображень, які одержуються з камери, зшиваються в панорамне.

У робочому просторі присутні кілька камер, зображення кожної з них може відрізнитись за кількома параметрами, таким як:

- розподільна здатність матриці;
- тип матриці;
- значення яскравості і контрастності;
- чутливість;
- кут огляду;
- частота кадрів тощо.

Звідси випливає, що зображення або потокове відео з камер може мати різні параметри, що ускладнює його зшивання в одне зображення для подальшої роботи.

В якості алгоритму зшивання (побудови панорами) обрано алгоритм SIFT (Scale Invariant Feature Transform) [104].

Алгоритм полягає в знаходженні особливих точок на зображенні і їх дескрипторів.

Особливі точки – ті точки, які з великою ймовірністю зустрінуться на іншому зображенні.

Дескриптори – параметри особливих точок, які виділяють їх серед інших, так звану унікальність кожної з точок.

Для знаходження особливих точок необхідно обчислення гауссіанів (застосування розмиття Гаусса до зображення) і їх різниць.

Знаходження гауссіана здійснюється за такою формулою:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

де  $L$  – значення гауссіана в точці з координатами  $(x, y)$ ,  $\sigma$  – радіус розмиття.  $G$  – гауссове ядро,  $I$  – значення вихідного зображення,  $*$  – операція згортки.

Різниця гауссіанів – зображення, отримане шляхом попіксельного віднімання одного гауссіана вихідного зображення з гауссіана з іншим радіусом розмиття і віднімається по формулі:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma)$$

Далі необхідно перевірити, чи підходить знайдена точка на роль ключової. Для цього необхідно провести апроксимацію різниці гауссіанів многочленом Тейлора 2 порядку, обчисленого в точці екстремума.

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x$$

де  $D$  – функція різниці гауссіанов,  $x = (x, y, \sigma)$  – вектор зміщення відносно точки розкладання,  $\frac{\partial}{\partial x}$  – градієнт,  $\frac{\partial^2}{\partial x^2}$  – матриця Гессе.

Екстремум многочлена Тейлора – обчислення похідної та порівнювання її до нуля. Результатом є зміщення точки обчисленого екстремуму щодо точного.

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x}$$

У разі, якщо одна з компонент вектору  $\hat{x}$  більше, ніж  $0,5 \times \text{крок\_сітки\_уу\_цьому\_напрямку}$ , то точка екстремуму була обчислена невірно і потрібно перейти до сусідньої точки в напрямку зазначених компонент. Аналогічні дії і для сусідньої точки. Якщо таким чином ми вийшли за межі октави, то слід виключити дану точку з розгляду.

У разі успішного знаходження положення точки екстремуму, перевіряється саме значення різниці гауссіанів в цій точці за формулою:

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x}$$

Якщо це значення не задовольняє, то точка вважається з малим контрастом і виключається.

У разі, коли особлива точка лежить на межі якогось об'єкта або погано освітлена, то таку точку можна виключити з розгляду. Ці точки мають великий вигин (одна з компонент другої похідної) уздовж кордону і малий в



перпендикулярному напрямку. Цей великий вигин визначається матрицею Гессе. Для перевірки можна використовувати матрицю розміру  $2 \times 2$ .

$$H = \begin{vmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{vmatrix}$$

Позначивши слід матриці як  $Tr(H)$ , визначник як  $Det(H)$ , можна записати наступне:

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$$

Якщо позначимо  $r$ , як відношення більшого вигину до меншого  $\alpha = r\beta$ :

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}$$

Тоді:

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r}$$

Після цього знайдемо орієнтацію в просторі ключовий точки. Величина  $i$  напрямок градієнта в точці визначається за формулами:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y)))$$

де  $m$  – величина цього градієнта,  $\theta$  – напрямок

Визначаємо околицю ключової точки для згортки з гаусовим ядром, в якому будуть розглянуті градієнти, коло з радіусом розмиття ( $\sigma$ ) яке буде дорівнює  $1.5 \times \text{масштаб\_ключової\_точки}$ . Для гауссова ядра діє так зване правило «трьох сигм» – значення цього ядра дуже близько до нуля на відстані, що перевищує  $3 \times \sigma$ .

Напрямок ключовий точки знайдемо з гістограми напрямків, що складається з 36 компонент, які рівномірно покривають проміжок в 360 градусів – кожна точка вікна  $(x, y)$  вносить вклад, що дорівнює  $m \times G(x, y, \sigma)$ , в ту компоненту гістограми, яка покриває проміжок, що містить напрям градієнта  $\theta(x, y)$ .

Напрямок ключовий точки лежить в проміжку, що покривається максимальною компонентою гістограми. Значення максимальної компоненти ( $\max$ ) і двох сусідніх з нею інтерполюються параболою, і точка максимуму цієї параболи береться в якості напрямку ключовий точки. Якщо в гістограмі є ще компоненти з величинами не менш  $0.8 \times \max$ , то вони аналогічно інтерполюються і додаткові напрямки приписуються до ключової точки.

Далі необхідно створити самі дескриптори на основі інформації про околиці ключових точок.

Для методу SIFT дескриптором є вектор, який вираховується на гауссіані, найближчому за масштабом до ключовій точці, і, виходячи з градієнтів, в деякому вікні ключової точки. Перед обчисленням дескриптора це вікно повертають на кут напрямку ключової точки, чим і досягається інваріантність щодо повороту (рис. 3.3).

У лівій частині зображені пікселі, позначені маленькими квадратиками, які беруться з вікна дескриптора, яке в поділено на чотири рівні частини (регіони). Стрілка в центрі кожного пікселя позначає градієнт цього пікселя.

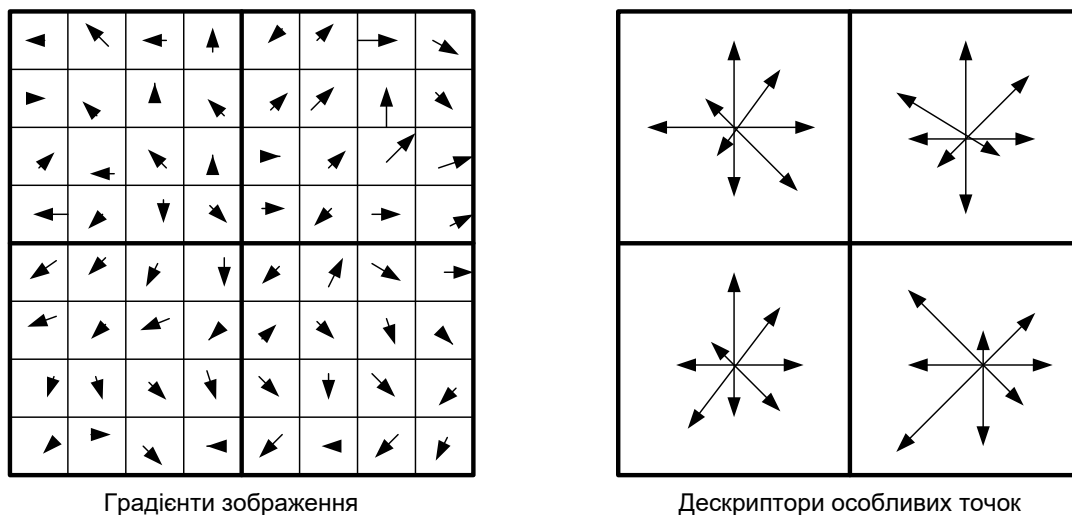


Рисунок 3.3– Зображення та отриманий на його основі дескриптор

Центр вікна знаходиться між пікселями. Він вибирається якомога ближче до точних координат ключовий точки. Окружністю позначено вікно згортки з гаусовим ядром (аналогічно вікну для обчислення напрямку ключовий точки). Для цього ядра визначається  $\sigma$ , рівна половині ширини вікна дескриптора. Надалі значення кожної точки вікна дескриптора буде помножуватись на значення гауссова ядра в цій точці, як на ваговий коефіцієнт.

У правій частині зображення розташований дескриптор особливої точки, розмірності  $2 \times 2 \times 8$ , де  $2 \times 2$  – кількість регіонів по горизонталі і вертикалі, а 8 - означає кількість компонент гiстограми цих регіонів. Гiстограми регіонів обчислюються аналогічно гiстограмі напрямків з трьома умовами:

- кожна гiстограма покриває ділянку в 360 градусів і ділить його на 8 частин;
- в якості вагового коефіцієнта береться значення гауссові ядра, загального для всього дескриптора;
- в якості додаткових вагових коефіцієнтів беруться коефіцієнти трилінійної інтерполяції.

Кожному градієнту в вікні дескриптора можна приписати три речові координати  $(x, y, n)$ , де  $x$  і  $y$  – відстані до градієнта по горизонталі і вертикалі,  $n$  – відстань до напрямку градієнта в гістограмі.

Точкою відліку називається лівий нижній кут вікна дескриптора і початкове значення гістограми. Одиничними відрізками беруться розміри регіонів по горизонталі і вертикалі для  $x$  і  $y$  відповідно, і кількість градусів в компоненті гістограми для  $n$ . Коефіцієнт трилінійної інтерполяції визначається для кожної координати  $(x, y, n)$  градієнта як  $1-d$ , де  $d$  – відстань від координати градієнта до середини того одиничного проміжку в якому знаходиться ця координата. Кожне входження градієнта в гістограму множиться на всі три вагові коефіцієнта трилінійної інтерполяції.

Дескриптор ключовий точки складається з усіх отриманих гістограм. Отриманий дескриптор нормалізується, після чого всі його компоненти, значення яких більше 0,2, урізаються до значення 0,2 і потім дескриптор нормалізується ще раз. У такому вигляді дескриптори готові до використання.

Після того, як отримані дескриптори ключових точок, зображення може бути зшите в панорамне.

Для цього використовуємо метод RANSAC [105] – метод підгонки моделей до даних, що містить точки, що не укладаються в модель. Після визначення точок, що входять в модель (гомографію), необхідно накласти все зображення на загальну площину. Наприклад, створюємо велике зображення, заповнене нулями, в площині, паралельній центральному зображенню накладаємо на нього все зображення. Центральне зображення заповнюється нулями праворуч і ліворуч. Використовуючи гомографію, визначаємо, з якого боку необхідно доповнювати зображення.

Розглянуті методи обробки та зшивки зображень будуть використані для створення моделей розпізнавання та ідентифікації, а також моделі керування мобільним роботом у просторі гнучкої інтегрованої системи.

### 3.2 Модель розпізнавання та ідентифікації об'єктів

Розглянемо гнучку інтегровану робототехнічну систему.

Кожен з предметів робочої простору (верстати, інструмент, оснащення, люди тощо) входять до складу робочої зони робота. Всі властивості об'єктів робочого простору в базі даних властивостей предметів *Param*, так званої матриці ознак об'єктів. Дану матрицю представимо в такий спосіб:

$$Param = \begin{bmatrix} param_{11} & param_{12} & \dots & param_{1n} \\ param_{21} & param_{22} & \dots & param_{2n} \\ \dots & \dots & \dots & \dots \\ param_{n1} & param_{n2} & \dots & param_{nm} \end{bmatrix} \quad (3.1)$$

Системі комп'ютерного зору робота необхідно провести операцію ідентифікації і розпізнавання об'єктів робочої зони для подальшого виконання роботом поставлених завдань за допомогою блоку адаптивного керування.

Процес ідентифікації буде полягати в тому, що певний об'єкт  $Obj_i$  робочого простору робота може бути розпізнаний за підмножиною ознак (властивостей) об'єктів  $Param'$ .

Для всієї множини об'єктів  $Obj$  набір підмножин  $param'_i$  формує множину  $Param'$ :

$$Param = \begin{bmatrix} param_{11} & param_{12} & \dots & param_{1n} \\ param_{21} & param_{22} & \dots & param_{2n} \\ \dots & \dots & \dots & \dots \\ param_{n1} & param_{n2} & \dots & param_{nm} \end{bmatrix} \quad (3.2)$$

Загальний вигляд процедури розпізнавання об'єктів формально можна записати у вигляді:

$$Rb_{sensor} \times Obj \Rightarrow Obj' \quad (3.3)$$

де  $Rb_{sensor}$  – робот, оснащений системою датчиків  $sensor$ ;  $Obj$  – множина об'єктів;  $Obj'$  – множина розпізнаних об'єктів.

З цього можна записати процедуру розпізнавання та ідентифікації об'єктів у наступному вигляді:

$$Rb_{sensor} \times Obj \equiv \begin{bmatrix} sensor_{11} & sensor_{12} & \dots & sensor_{1m} \\ sensor_{21} & sensor_{22} & \dots & sensor_{2m} \\ \dots & \dots & \dots & \dots \\ sensor_{n1} & sensor_{n2} & \dots & sensor_{nm} \end{bmatrix} \times \begin{bmatrix} param_{11} & param_{12} & \dots & param_{1k} \\ param_{21} & param_{22} & \dots & param_{2k} \\ \dots & \dots & \dots & \dots \\ param_{m1} & param_{m2} & \dots & param_{mk} \end{bmatrix} \Rightarrow \begin{bmatrix} param'_{11} & param'_{12} & \dots & param'_{1k} \\ param'_{21} & param'_{22} & \dots & param'_{2k} \\ \dots & \dots & \dots & \dots \\ param'_{n1} & param'_{n2} & \dots & param'_{nk} \end{bmatrix} \Rightarrow Obj' \quad (3.4)$$

де  $\begin{bmatrix} sensor_{11} & sensor_{12} & \dots & sensor_{1m} \\ sensor_{21} & sensor_{22} & \dots & sensor_{2m} \\ \dots & \dots & \dots & \dots \\ sensor_{n1} & sensor_{n2} & \dots & sensor_{nm} \end{bmatrix}$  – множина сенсорів;

$\begin{bmatrix} param_{11} & param_{12} & \dots & param_{1k} \\ param_{21} & param_{22} & \dots & param_{2k} \\ \dots & \dots & \dots & \dots \\ param_{m1} & param_{m2} & \dots & param_{mk} \end{bmatrix}$  – множина ознак;

$Param' = \begin{bmatrix} param'_{11} & param'_{12} & \dots & param'_{1k} \\ param'_{21} & param'_{22} & \dots & param'_{2k} \\ \dots & \dots & \dots & \dots \\ param'_{n1} & param'_{n2} & \dots & param'_{nk} \end{bmatrix}$  – множина розпізнаних ознак;

$Obj'$  – множина розпізнаних об'єктів.

Таким чином, процедура ідентифікації буде полягати у визначенні в матриці розпізнаних ознак тих з них, які є необхідними і достатніми для ідентифікації об'єктів [78]:

$$I \times \begin{bmatrix} param'_{11} & param'_{12} & \dots & param'_{1n} \\ param'_{21} & param'_{22} & \dots & param'_{2n} \\ \dots & \dots & \dots & \dots \\ param'_{n1} & param'_{n2} & \dots & param'_{nm} \end{bmatrix} \Rightarrow \begin{bmatrix} param''_{11} & param''_{12} & \dots & param''_{1n} \\ param''_{21} & param''_{22} & \dots & param''_{2n} \\ \dots & \dots & \dots & \dots \\ param''_{n1} & param''_{n2} & \dots & param''_{nm} \end{bmatrix} \Rightarrow Obj'' . \quad (3.5)$$

де  $\begin{bmatrix} param''_{11} & param''_{12} & \dots & param''_{1n} \\ param''_{21} & param''_{22} & \dots & param''_{2n} \\ \dots & \dots & \dots & \dots \\ param''_{n1} & param''_{n2} & \dots & param''_{nm} \end{bmatrix}$  – матриця ідентифікованих ознак, що

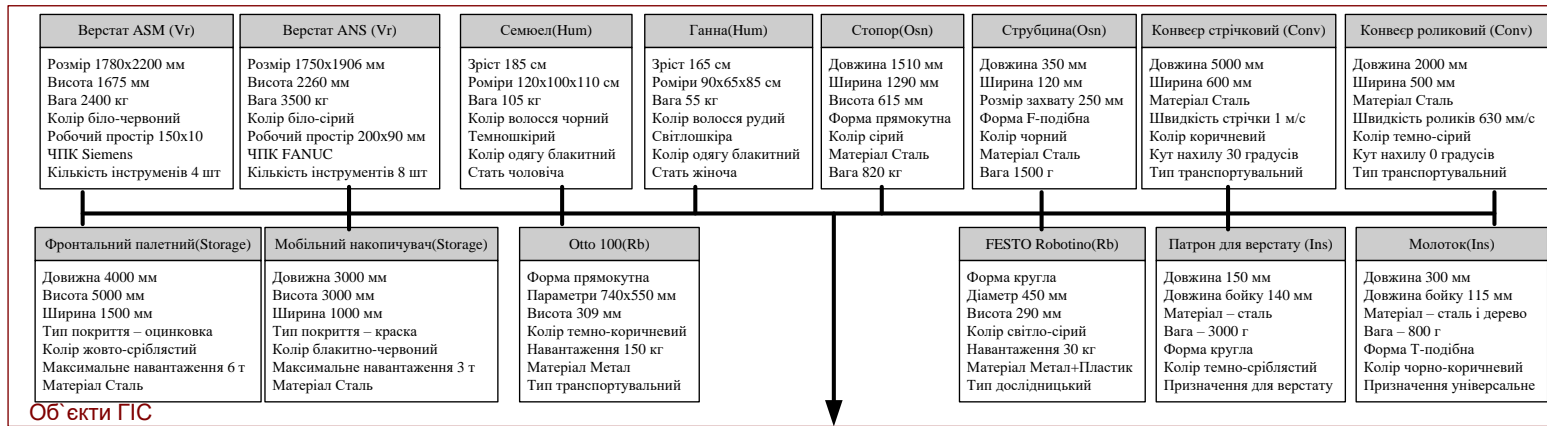
характеризує множину ідентифікованих об'єктів  $Obj''$ .

З цієї матриці буде сформована матриця розпізнаних і ідентифікованих ознак робота:

$$Param_{Rb} = \begin{bmatrix} param_{Rb1} \\ param_{Rb2} \\ \dots \\ param_{Rbn} \end{bmatrix} . \quad (3.6)$$

Аналогічним чином буде формуватися і матриця ідентифікованих ознак інших об'єктів адаптивного роботизованого виробництва.

Приклад використання моделі розпізнавання і ідентифікації об'єктів, що присутні у робочому просторі мобільного робота, можна представити у вигляді схеми, зображеної на рис. 3.4. Зі схеми видно, що СКЗ здійснює ідентифікацію та розпізнавання об'єктів у робочому просторі ГІВС на основі порівняння візуальної інформації з наявними у базі даних характеристиками технічних об'єктів ГІВС.



Об'єкти ГІС

База даних властивостей об'єктів ГІС

1780x2200	1675	2400	біло-червоний	150x10	ЧПК Siemens	4 інструменти
1750x1906	2260	3500	біло-сірий	200x90	ЧПК FANUC	8 інструментів
185	120x100x110	105	волося чорне	темношкіра	одяг блакитний	стать чоловіча
165	90x65x85	55	волося руде	світлошкіра	одяг блакитний	стать жіноча
1510	1290	615	прямокутна	сірий	сталь	820 кг
350	120	290	F-подібна	чорний	сталь	1500 г
5000	600	сталь	1 м/с	коричневий	30 градусів	транспортувальний
2000	500	сталь	630 мм/с	темно-сірий	0 градусів	транспортувальний
4000	5000	1500	оцинковка	жовто-сріблястий	6 т	сталь
3000	3000	1000	краска	блакитно-червоний	3 т	сталь
300	115	сталь і дерево	800 г	T-подібна	чорно-коричневий	універсальне призначення
150	140	сталь	3000 г	кругла	темно-сріблястий	для верстату
прямокутна	740x550	309	темно-коричневий	150 кг	метал	транспортувальний
кругла	450	290	світло-сірий	30 кг	метал+пластик	дослідницький

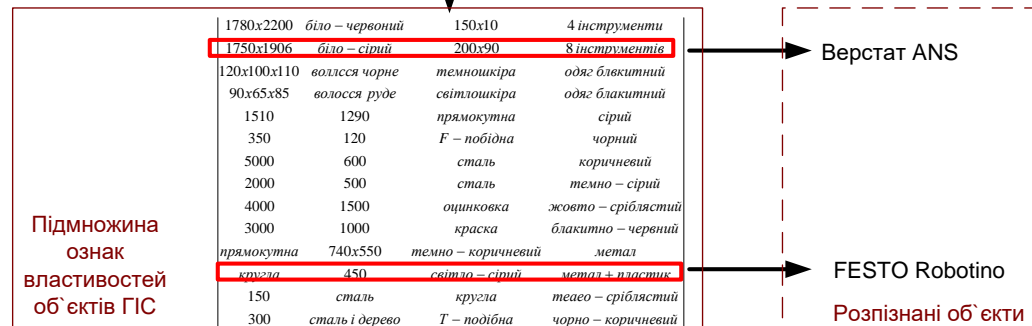


Рисунок 3.4 – Приклад виконання моделі розпізнавання



### 3.3 Модель керування мобільним роботом у просторі гнучкої інтегрованої системи на основі інформації об'єктові системи комп'ютерного зору

Побудова моделі керування мобільним роботом має виходити з наявності технічних особливостей конструкції шасі МР, наявності сенсорної системи певного рівня, здатності виконувати маніпуляції з об'єктами РП. Оскільки передбачається, що ключовим елементом сенсорної системи має бути система комп'ютерного зору, запропонована модель має зворотні зв'язки, що забезпечуються на основі отримання візуальної інформації [74-76], [80], [82].

Для опису функціонування системи керування мобільного робота за допомогою об'єктові системи комп'ютерного зору (ОСКЗ) пропонується схема керування, що показана на рис. 3.5 та для локальної системи комп'ютерного зору (ЛСКЗ) на рис. 3.6.

Головна розбіжність між системами полягає у тому, що для керування ОСКЗ використовується комп'ютер, а робот лише виконує команди керування від нього на основі інформації з об'єктові камери.

ЛСКЗ виконує всі дії за допомогою процесору плати керування, встановленої на ній. Це може бути плата Raspberry Pi, ASUS Tinkerboard та подібні їм.

Аналіз інформації та її інтерпретацію, всі розрахунки і формування керуючих впливів виконує комп'ютер *Computer*, а робот *Rb* – лише виконує ці впливи.

Вихідну інформацію для функціонування системи керування МР має надавати система підтримки прийняття рішень *DMS*. Вона має попередньо генерувати маршрут робота як послідовність цільових точок  $Sp_i(x_i, y_i, z_i)$ , або у випадку 2D-проекції –  $Sp_i(x_i, y_i, z_i = 0)$ .

Весь шлях буде складатись з суми відрізків, які потрібно пройти роботу:

$$Cp_i(x_i, y_i, z_i) = \sum_{j=0}^N Cp_i^j(x_i^j, y_i^j, z_i^j) \quad (3.7)$$

Формування різниці координат  $\varepsilon_x, \varepsilon_y$  формує перехід з точки  $Cp_i^0$  в точку  $Cp_i$  наступним чином (рис 3.7).

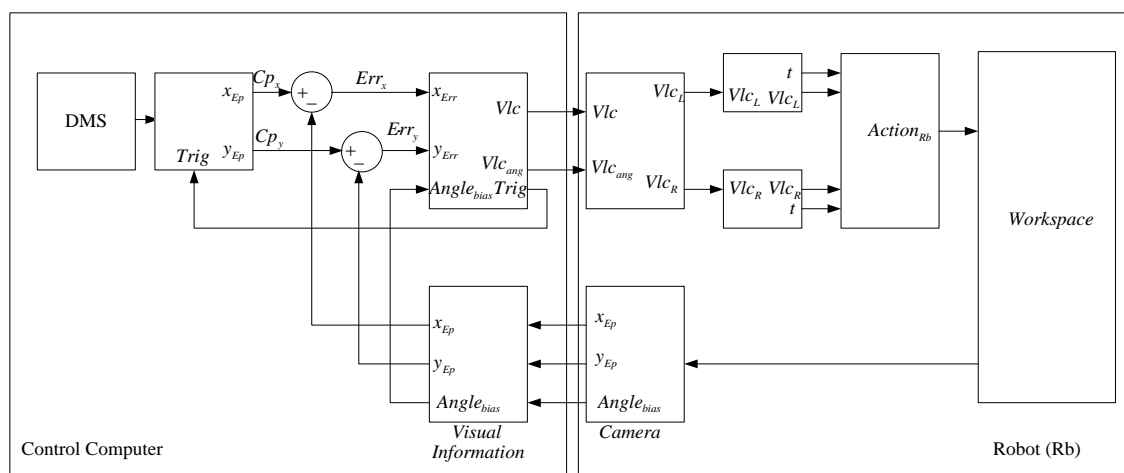


Рисунок 3.5 – Схема керування мобільним транспортувальним колісним роботом із зворотнім зв'язком за візуальним положенням з використанням ОСКЗ

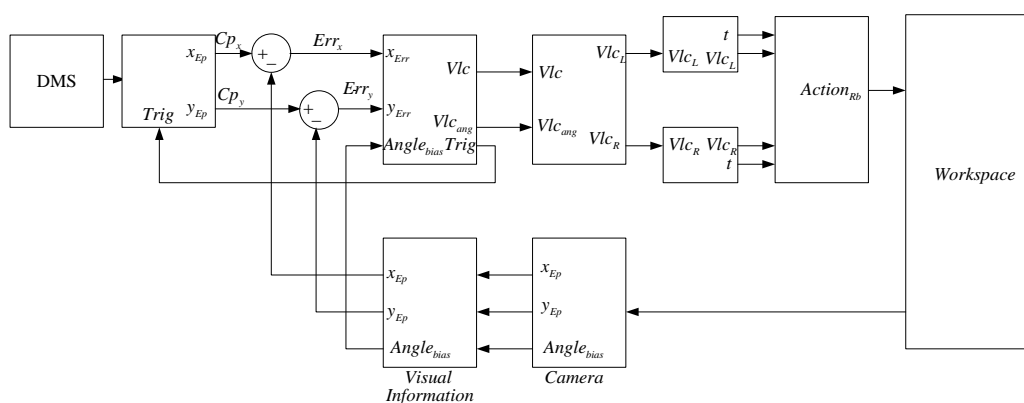


Рисунок 3.6 – Схема керування мобільним транспортувальним колісним роботом із зворотнім зв'язком за візуальним положенням з використанням ЛСКЗ

Система позиціонування робота порівнює поточні координати робота  $Cp_i^0(x_i^0, y_i^0)$  з  $Cp_i^1(x_i^1, y_i^1)$  і знаходить різницю між ними, як  $Cp_i^1 Cp_i^0(x_i^1 - x_i^0, y_i^1 - y_i^0)$ . Таким чином знаходиться різниця координат  $\varepsilon_x = (x_i^1 - x_i^0)$ ,  $\varepsilon_y = (y_i^1 - y_i^0)$ .

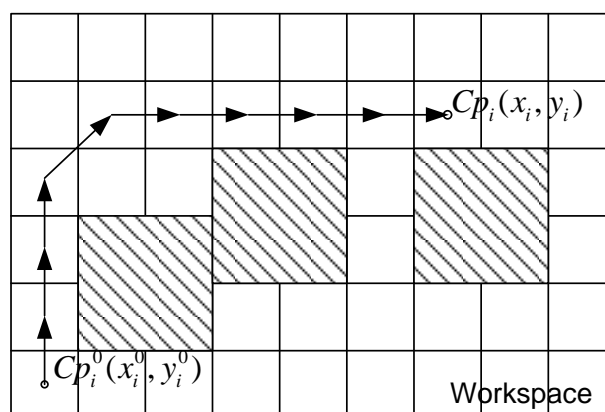


Рисунок 3.7 – Схема переходу у точку дискретного РП

Виходячи з того, що камера розташована об'єктивно, початкова точка  $Sp_{Rb}$  має координати  $(x_0; y_0)$ . Оскільки весь шлях є дискретним, то між початковою та кінцевою точками визначаються проміжні точки  $Cp_i^0$ .

Далі задається кінцева точка  $Ep_{Rb}$  маршруту з координатами  $(x_{Ep}; y_{Ep})$ .

Під час реалізації маршруту практично завжди виникають похибки по осях –  $Err_x$  та  $Err_y$ , тому координати кінцевої точки з помилкою  $Ep_{Rb}^{Err}$  будуть мати координати:

$$(Err_x, Err_y) = (Cp_{ix}^n - Ep_x, Cp_{iy}^n - Ep_y). \quad (3.8)$$

Наявність похибок не має перевищувати певний рівень дискретності робочого простору, тобто:

$$(Err_x < k \times D_x; Err_y < k \times D_y), \quad (3.9)$$

де  $k \in [0, 0.25]$ .

Знаючи загальну швидкість робота, отримаємо час  $t$ , який потрібен роботу, щоб переміститись у кінцеву точку переміщення.

На основі кінематичної моделі двоколісного розраховуються величини лінійної та кутової швидкостей:

$$\dot{x} = \cos \alpha \times Vlc, \quad (3.10)$$

$$\dot{y} = \sin \alpha \times Vlc, \quad (3.11)$$

$$\dot{\alpha} = Vlc_{Ang}, \quad (3.12)$$

$$Vlc = \frac{K}{2}(Vlc_1 + Vlc_2), \quad (3.13)$$

$$Vlc_{Ang} = -\frac{K}{2}(Vlc_1 - Vlc_2), \quad (3.14)$$

де  $(x, y)$  – координати колісного робота на площині,  $\alpha$  – кут повороту робота на площині,  $Vlc_{Ang}$  – кутова швидкість,  $Vlc$  – лінійна швидкість робота,  $Vlc_1$  та  $Vlc_2$  – лінійні швидкості правого та лівого коліс,  $K$  – коефіцієнт передавання приводу.

Виконання мобільним роботом переміщень із заданими швидкостями  $Vlc_L$  та  $Vlc_R$  на інтервалі часу  $t$  означає виконання дій  $Action_{Rb}$ , що взаємодіють з командами керування роботом і виконують дії у робочому просторі *Workspace*.

Система комп'ютерного зору *Camera* отримує інформацію з робочого простору *Workspace*, а саме положення роботу по осям  $(x_{Ep}; y_{Ep})$ , а також кут його повороту  $Angle_{bias}$  відносно осей координат.

Блок візуального керування *Visual Information* отримує ці координати та аналізує положення робота.

Ці координати порівнюються з необхідними кінцевими і вираховується наступна точка переміщення в дискретному робочому просторі. Якщо вони

різні, тоді поточні координати  $(x_{cur}; y_{cur})$  стають початковими координатами переміщення і весь процес повторюється знову, доки робот не прибуде в кінцеву точку переміщення.

Далі відбувається переміщення у наступну точку дискретного простору.

Якщо помилки немає, або вона компенсується і робот досяг цільової точки, встановлюється булевий тригер  $Trig$ , що сигналізує ( $true$ ) про виконання завдання.

Шлях, що потрібно проти роботу між двома точками дискретного простору дорівнює  $L_{Rb}^i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$ .

Тоді весь шлях матиме наступний вигляд:

$$L_{Rb} = \sum_{i=0}^n L_{Rb}^i. \quad (3.15)$$

Маршрут від точки  $Cp_i^0(x_i^0, y_i^0)$  до  $Cp_i(x_i, y_i)$  може бути визначений двома способами:

- визначення шляху, який повинен пройти робот та кут, на який йому потрібно повернутись;
- прямолінійний рух, потім рух за окружністю.

Перший спосіб є найлегшим за відсутності розрахунку швидкостей і кутових швидкостей, а тільки поворотів колеса.

Основними параметрами тут є радіус колеса робота  $R_w$ , а також відстань між колесами, що буде радіусом повороту робота  $R_{L_w}$ .

Колесо за один повний оберт проходить відстань  $L_w = 2 \times \pi \times R_w$ .

Робот за один повний оберт (розворот на 360 градусів), використовуючи одне колесо, проходить  $L_{L_w} = 2 \times \pi \times R_{L_w}$ .

Кількість обертів коліс  $Q_{rot}$ , яку потрібно пройти роботу для подолання цієї дистанції буде дорівнювати:

$$Q_{rot} = \frac{L_{Rb}}{L_w}. \quad (3.16)$$

Кут повороту робота  $Angle_{bias}$  можна визначити з використанням теореми косинусів. Він може бути таким:

$$Angle_{bias} = \begin{cases} 0 - \arccos \left( \frac{(x_{Sp} - x_{Ep})^2 + (\sqrt{(x_{Sp} - x_{Ep})^2 + (y_{Sp} - y_{Ep})^2})^2 - (y_{Sp} - y_{Ep})^2}{2(x_{Sp} - x_{Ep})\sqrt{(x_{Sp} - x_{Ep})^2 + (y_{Sp} - y_{Ep})^2}} \right); & \text{якщо } x_{Ep} > x_{Sp}, y_{Ep} < y_{Sp} \\ -90 - \arccos \left( \frac{(x_{Ep} - x_{Sp})^2 + (\sqrt{(x_{Ep} - x_{Sp})^2 + (y_{Ep} - y_{Sp})^2})^2 - (y_{Ep} - y_{Sp})^2}{2(x_{Ep} - x_{Sp})\sqrt{(x_{Ep} - x_{Sp})^2 + (y_{Ep} - y_{Sp})^2}} \right); & \text{якщо } x_{Ep} > x_{Sp}, y_{Ep} > y_{Sp} \\ 180 + \arccos \left( \frac{(x_{Sp} - x_{Ep})^2 + (\sqrt{(x_{Sp} - x_{Ep})^2 + (y_{Ep} - y_{Sp})^2})^2 - (y_{Ep} - y_{Sp})^2}{2(x_{Sp} - x_{Ep})\sqrt{(x_{Sp} - x_{Ep})^2 + (y_{Ep} - y_{Sp})^2}} \right); & \text{якщо } x_{Ep} < x_{Sp}, y_{Ep} > y_{Sp} \\ 90 + \arccos \left( \frac{(x_{Ep} - x_{Sp})^2 + (\sqrt{(x_{Sp} - x_{Ep})^2 + (y_{Sp} - y_{Ep})^2})^2 - (y_{Sp} - y_{Ep})^2}{2(x_{Sp} - x_{Ep})\sqrt{(x_{Sp} - x_{Ep})^2 + (y_{Sp} - y_{Ep})^2}} \right); & \text{якщо } x_{Ep} < x_{Sp}, y_{Ep} < y_{Sp} \end{cases} \quad (3.17)$$

Тоді для того, щоб дізнатися, який шлях слід пройти роботу для повороту одним колесом  $L_{Angle}$  на кут  $Angle_{bias}$  потрібно скористатись формулою:

$$L_{Angle} = \frac{L_w}{360} \times Angle_{Bias}. \quad (3.18)$$

Для другого випадку додається проміжна точка  $Ip_{Rb}(x_{Ip} = x_i^0, y_{Ip} = y_i)$ , до якої робот здійснює прямолінійний рух, а потім по колу (рис. 3.8).

Для того, щоб отримати швидкості для першого і другого мотору (для ідеального випадку, коли траєкторія проста) потрібно виконати математичні операції, які проілюстровано на рис. 3.9.

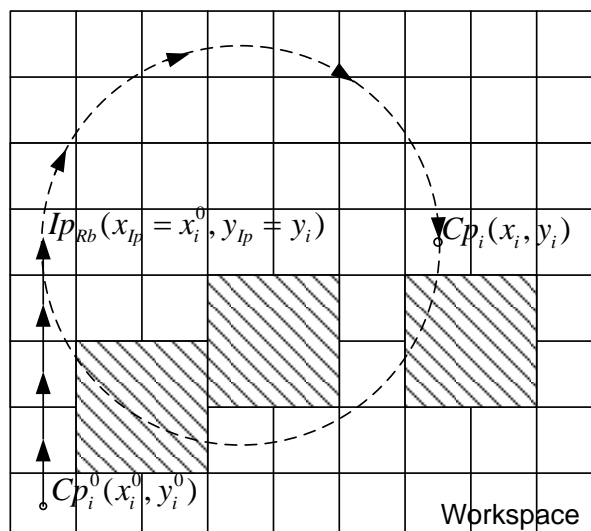


Рисунок 3.8 – Варіант з прямолінійним, а потім круговим рухом

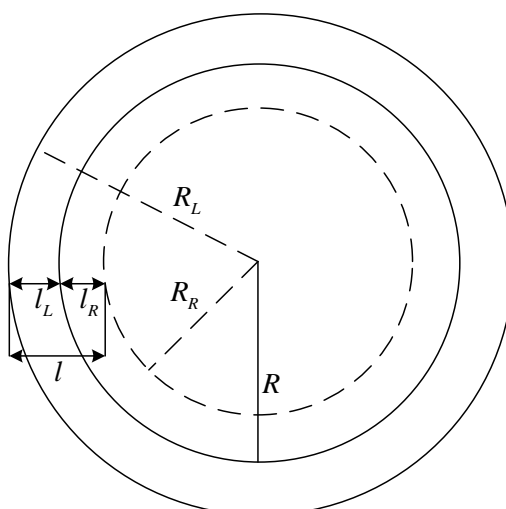


Рисунок 3.9– Ілюстрація знаходження швидкостей коліс робота для поворотів

На рис. 3.9 відстань між колесами робота дорівнює  $l$ , відстань від лівого та правого колеса до центру робота  $l_L$  та  $l_R$  відповідно, при умові, що  $l_L = l_R$ . З цієї умови можна знайти радіуси поворотів лівого  $R_L$  та правого  $R_R$  коліс. Радіус кругового руху  $R$  буде дорівнювати:

$$R = \frac{x_i - x_i^0}{2}. \quad (3.19)$$

Радіуси повороту лівого та правого коліс відповідно:

$$R_L = R - \frac{l}{2}; R_R = R + \frac{l}{2}. \quad (3.20)$$

Шлях, який проходить робот за одне повне коло, а також шляхи лівого та правого колеса будуть дорівнювати:

$$\left\{ \begin{array}{l} L = 2 \times \pi \times R \\ L_1 = 2 \times \pi \times R_L = 2 \times \pi \times \left( R - \frac{l}{2} \right) \\ L_2 = 2 \times \pi \times R_R = 2 \times \pi \times \left( R + \frac{l}{2} \right) \end{array} \right. \quad (3.21)$$

Швидкості будуть дорівнювати:

$$\left\{ \begin{array}{l} Vlc = \frac{L}{t} = \frac{2 \times \pi \times R}{t} \\ Vlc_L = \frac{L_L}{t} = \frac{2 \times \pi \times R_L}{t} = \frac{2 \times \pi \times \left( R - \frac{l}{2} \right)}{t} \\ Vlc_R = \frac{L_R}{t} = \frac{2 \times \pi \times R_R}{t} = \frac{2 \times \pi \times \left( R + \frac{l}{2} \right)}{t} \end{array} \right. \quad (3.22)$$

Запропонована модель керування роботом дозволяє на основі інформації об'єктової системи комп'ютерного зору розрахувати маршрут переміщення мобільної платформи у просторі ГІВС.



### 3.4 Метод адаптивного візуального керування роботом

Метод адаптивного візуального керування є описом сукупності прийомів та операцій, що у інтелектуальних системах керування роботами у зв'язку із розв'язанням завдання переміщення мобільної платформи або маніпулятора на основі отримання зорової інформації від систем комп'ютерного (технічного) зору [66], [93]. Розглянемо основні положення такого методу.

1. За допомогою систем комп'ютерного (технічного) зору *Camera* відбувається розпізнавання наявності та положення робота *Rb* у робочому просторі  $W_s$ ;

2. Початковою точкою маршруту робота *Rb* вважатиметься поточне розміщення робота  $Sp_{Rb}$  у робочому просторі  $W_s$ ;

3. Відбувається перетворення системи координат  $Trans(Rb, Camera)$  камери *Camera* у систему координат робота *Rb*;

4. Відбувається цілевказання кінцевої точки  $Ep_{Rb}$  переміщення робота;

5. Відбувається перетворення  $Trans(Camera, Rb)$  системи координат камери у систему координат робота;

6. Проводиться розрахунок шляху переміщення  $L_{Rb} = \sum_{i=0}^n L_{Rb}^i$ ;

7. Проводиться поділ шляху на відрізки  $L_{Rb}^i$  у відповідності до дискретності робочого простору;

8. Отримуються кути поворотів робота  $Angle_{Bias}$ ;

9. Отримуються лінійні  $Vlc$  швидкості і кутової  $Vlc_{Ang}$  швидкості робота;

10. Проводиться розрахунок керуючих впливів *Control*;

11. Відбувається виконання керуючих впливів *Control*;

12. Проводиться ідентифікація *I* робота *Rb* у робочому просторі;

13. Забезпечується отримання поточних координат робота у просторі камери  $Cp(x_{cp}, y_{cp})$ ;

14. Відбувається перетворення  $Trans(Rb, Camera)$  системи координат камери у систему координат робота;

15. Відбувається порівняння поточних координат робота  $Cp(x_{cp}, y_{cp})$  та цільових координат  $Ep_{Rb}(x_{Rb}, y_{Rb})$ ;

16. Розрахунок похибки. Якщо похибка переміщення перевищує максимальну допустиму, перехід на крок 3;

$$\begin{aligned} Err_x &= Cp_{ix}^n - Ep_x, Err_x > \varepsilon \\ Err_y &= Cp_{iy}^n - Ep_y, Err_y > \varepsilon \end{aligned} \quad (3.24)$$

17. Якщо похибка не перевищує максимальну допустиму, то робот досяг кінцевої точки маршруту;

$$\begin{aligned} Err_x &= Cp_{ix}^n - Ep_x, Err_x < \varepsilon \\ Err_y &= Cp_{iy}^n - Ep_y, Err_y < \varepsilon \end{aligned} \quad (3.25)$$

Представлення методу у вигляді блок-схеми наведено на рис. 3.10.

Запропонований метод може використовуватись як засіб постановки завдання мобільним транспортувальним роботам для виконання операцій в робочому просторі ГІВС та автоматизованих складів [82-83]; промисловим роботам – для виконання маніпуляцій з об'єктами на окремому робочому місці пов'язаних з завантаженням/розвантаженням технологічного обладнання та його обслуговуванням, а також у інших сферах, наприклад, візуальне цілевказання для наземних та повітряних дронів, систем автоматичного паркування транспортувальних засобів.

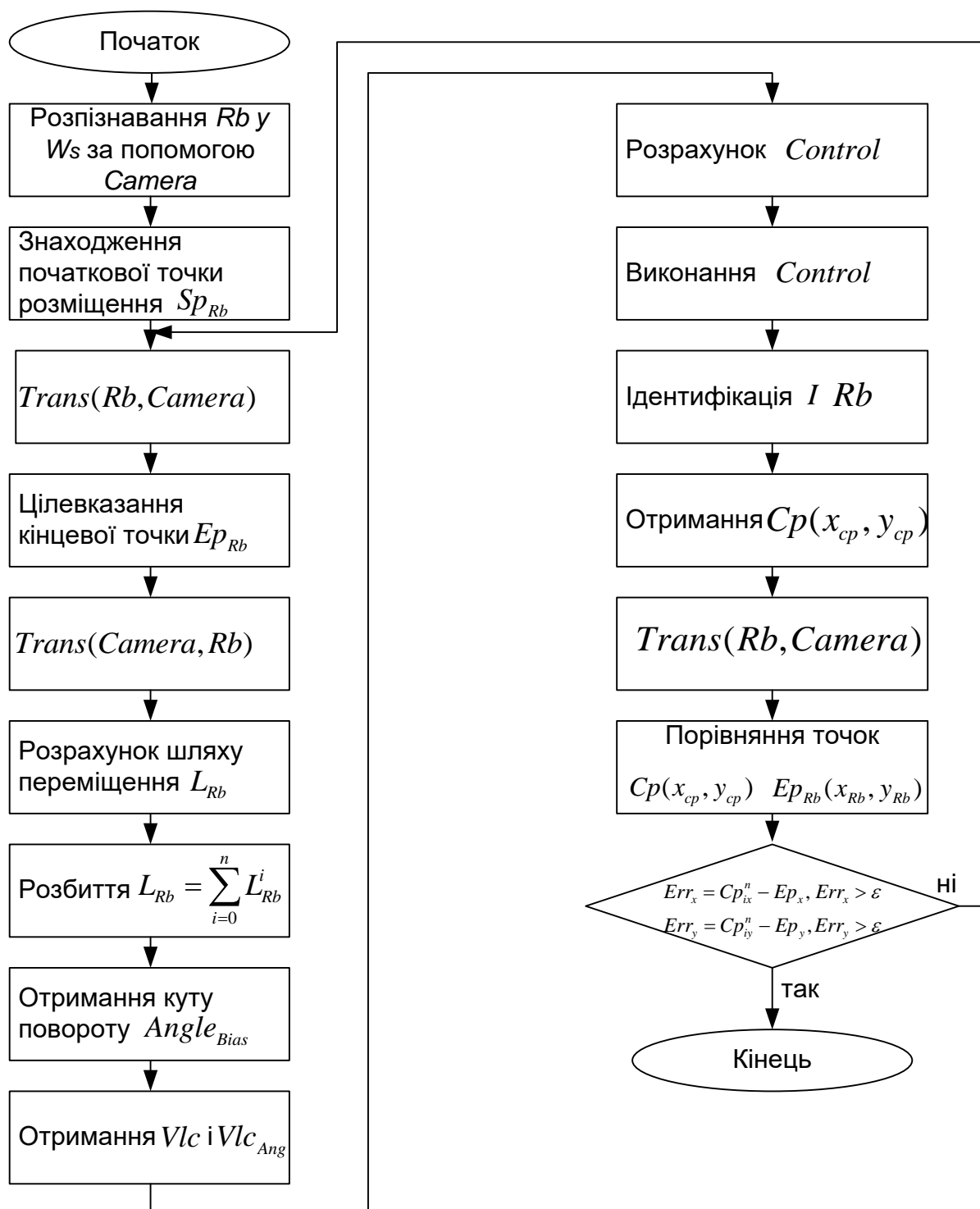


Рисунок 3.10 – Представлення методу у вигляді блок-схеми

### 3.5 Модель нечіткого адаптивного керування роботом

Для того, щоб створити модель нечіткого адаптивного керування, потрібно розробити його схему (рис. 3.11).

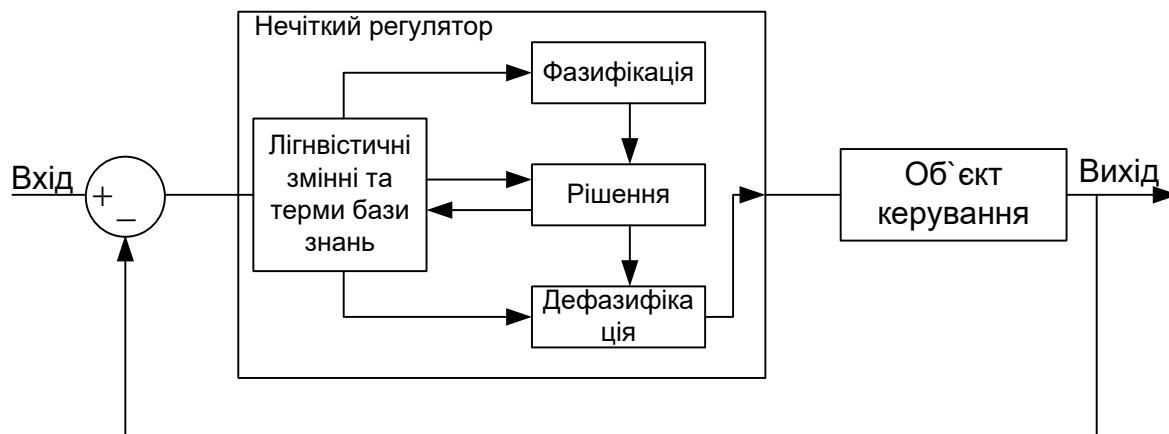


Рисунок 3.11 – Схема нечіткого адаптивного візуального керування

В якості алгоритму для нечіткого регулятора використано алгоритм Мамдані, який складається з таких пунктів:

- створення правил нечіткого виводу;
- формування бази, що зберігає правила нечіткого виводу;
- фазифікація вхідних змінних;
- об'єднання підумов у нечітких правилах;
- створення підрішень для цих нечітких правил;
- збирання рішень з підрішень нечітких правил;
- дефазицфікація вихідних змінних.

Створимо базу правил нечіткого виводу.

Першою лінгвістичною змінною, що буде використовуватись на вході нечіткого регулятора є *Length* – довжина маршруту, що потрібно пройти роботу для досягнення кінцевої точки.

Термами для цієї змінної є:

- ShortLen – мала відстань;

- MiddleLen – середня відстань;
- LargeLen – велика відстань.

Другою лінгвістичною змінною, що буде використовуватись на вході нечіткого регулятора є *Angle* – кут повороту, на який потрібно повернутись роботу.

Термами для цієї змінної є:

- LargeNeg – великий негативний;
- MiddleNeg – середній негативний;
- SmallNeg – малий негативний;
- Zero – нульовий;
- SmallPos – малий позитивний;
- MiddlePos – середній позитивний;
- LargePos – великий позитивний.

Першою і другою лінгвістичною змінною, що буде використовуватись на виході нечіткого регулятора є *VeloLeft* та *VeloRight* – швидкість лівого та правого коліс робота.

Термами для цієї змінної є:

- Zero – нульова швидкість;
- LowSmallPos – низька мала позитивна;
- SmallPos – мала позитивна;
- LowMiddlePos – низька середня позитивна;
- MiddlePos – середня позитивна;
- LowLargePos – низька велика позитивна;
- LargePos – велика позитивна;
- VeryLargePos – дуже велика позитивна.

Розглянемо детальний опис визначення списку необхідних лінгвістичних змінних, на основі яких формуються нечіткі правила виведення під час формування керуючих впливів.

Множина лінгвістичних змінних може бути представлена у наступному вигляді:

$$\beta_1 = \langle \text{Length}, \{\text{ShortLen}, \text{MiddleLen}, \text{LargeLen}\}, [0 \dots 100], G_1, M_1 \rangle; \quad (3.26)$$

$$\beta_2 = \langle \text{Angle}, \{\text{LargeNeg}, \text{MiddleNeg}, \text{SmallNeg}, \text{Zero}, \text{SmallPos}, \text{MiddlePos}, \text{LargePos}\}, [-180 \dots +180], G_2, M_2 \rangle; \quad (3.27)$$

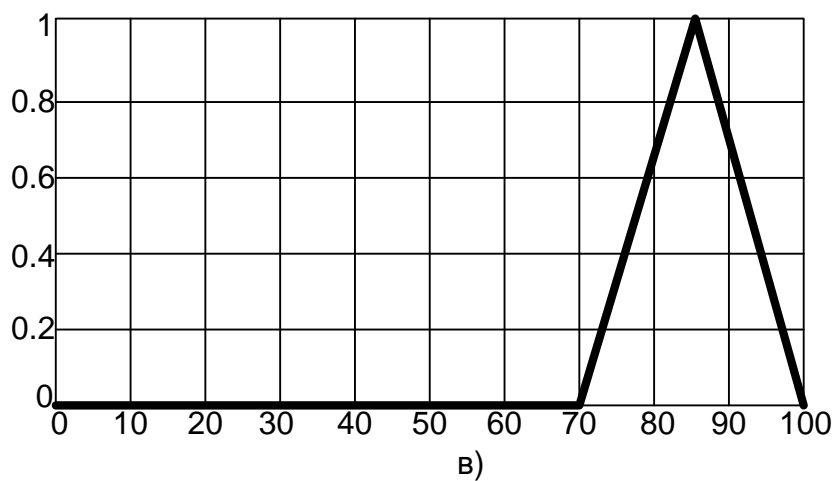
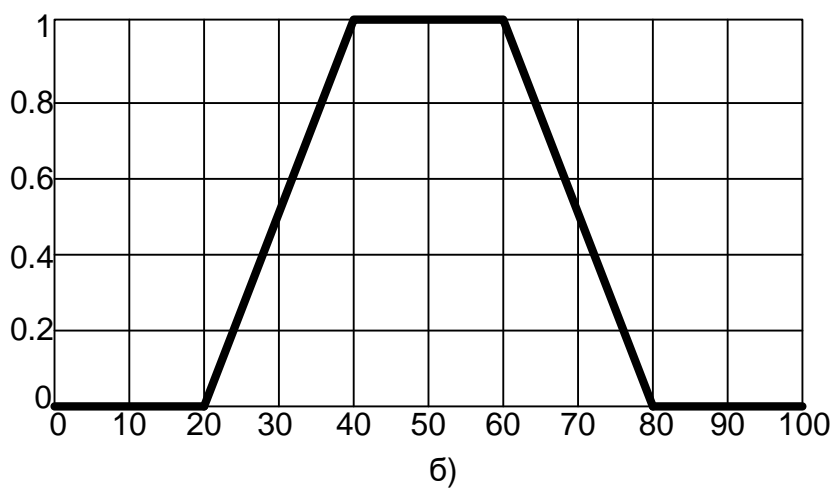
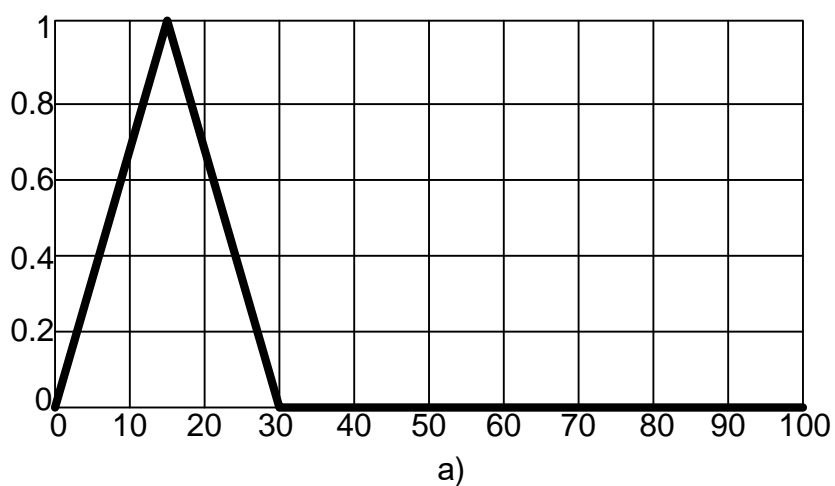
$$\beta_3 = \langle \text{VeloLeft}, \{\text{Zero}, \text{LowSmallPos}, \text{SmallPos}, \text{LowMiddlePos}, \text{MiddlePos}, \text{LowLargePos}, \text{LargePos}, \text{VeryLargePos}\}, [0 \dots 100], G_3, M_3 \rangle; \quad (3.28)$$

$$\beta_4 = \langle \text{VeloLeft}, \{\text{Zero}, \text{LowSmallPos}, \text{SmallPos}, \text{LowMiddlePos}, \text{MiddlePos}, \text{LowLargePos}, \text{LargePos}, \text{VeryLargePos}\}, [0 \dots 100], G_4, M_4 \rangle; \quad (3.29)$$

Для цих лінгвістичних змінних потрібно розробити функції приналежності.

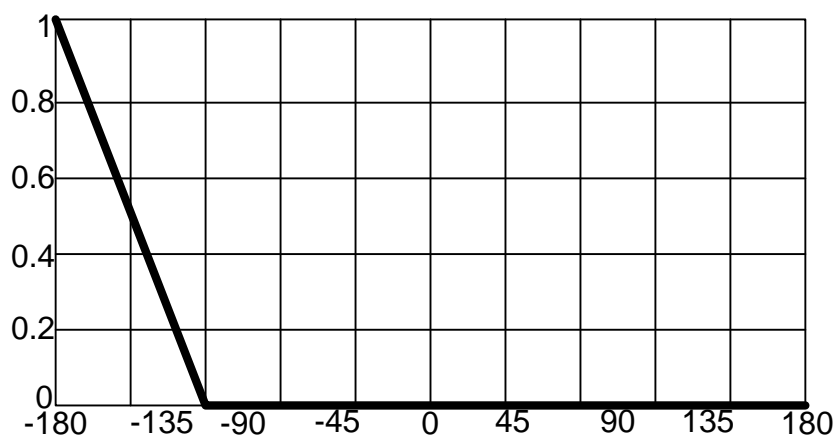
Розглянемо створення для кожної з лінгвістичних змінних, що використовуються.

На рис. 3.12 можна ознайомитись з цими функціями для вхідної змінної *Length*, на рис. 3.13-3.14 – для вхідної змінної *Angle*, на рис. 3.15-3.17 – для вихідних змінних *VeloLeft* та *VeloRight*.

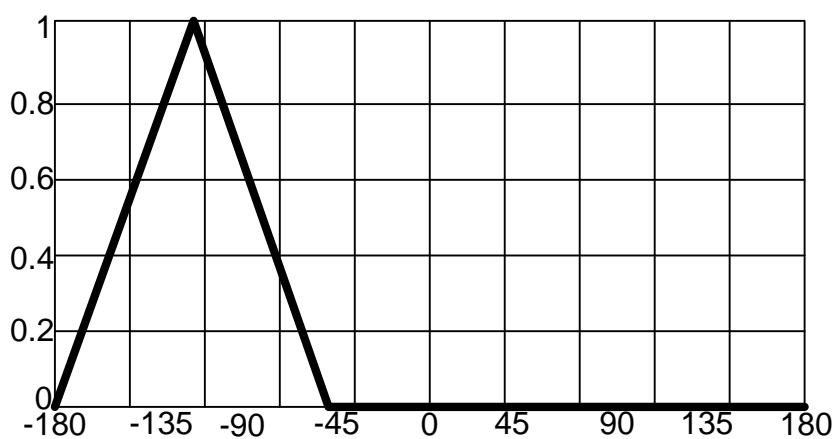


а) ShortLen (мала відстань), б) MiddleLen (середня відстань), в) LargeLen (велика відстань).

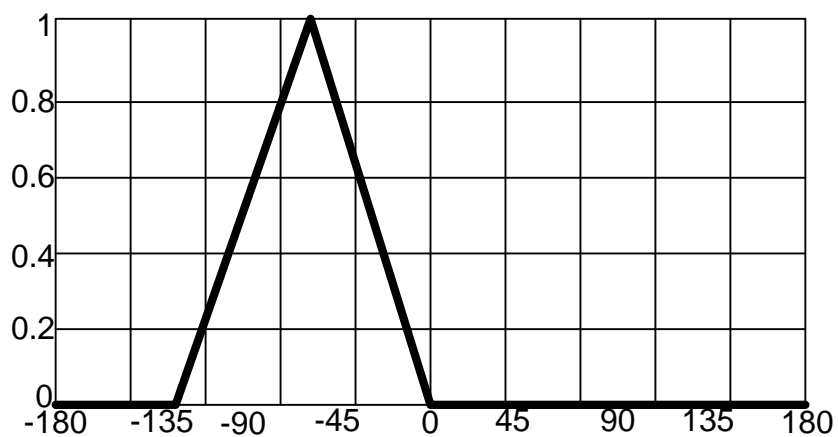
Рисунок 3.12 – Функції приналежності для вхідної змінної *Length*



а)



б)

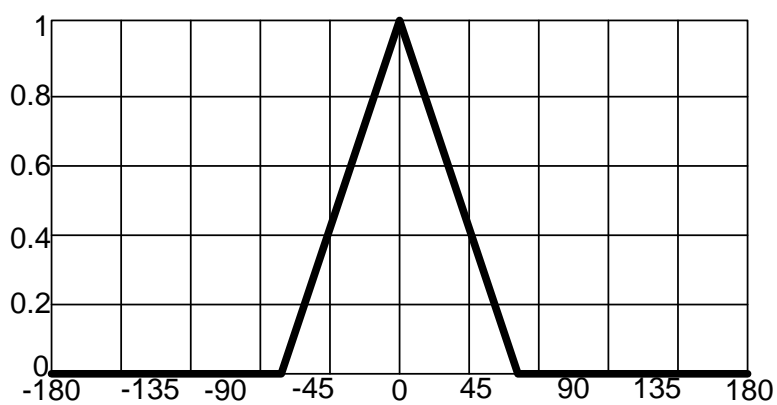


в)

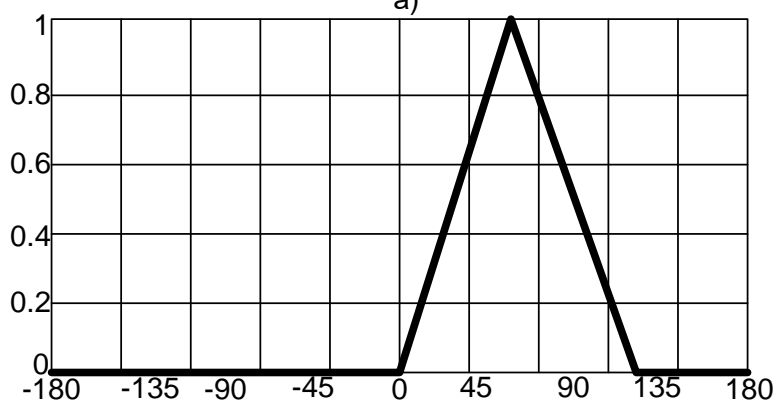
а) LargeNeg (великий негативний), б) MiddleNeg (середній негативний),  
в) SmallNeg (малий негативний).

Рисунок 3.13 – Функції приналежності для вхідної змінної *Angle*

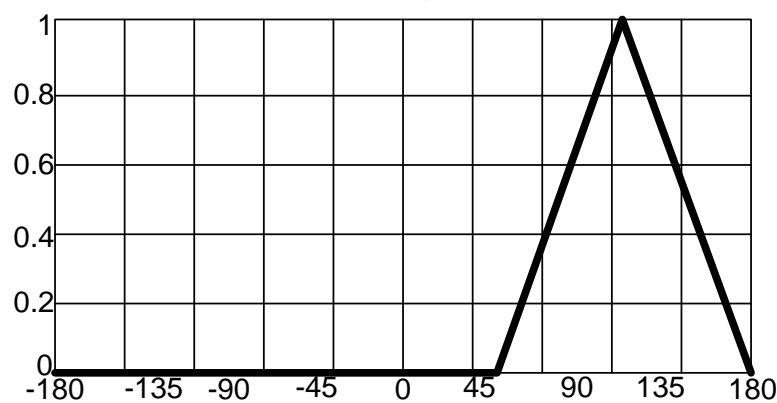




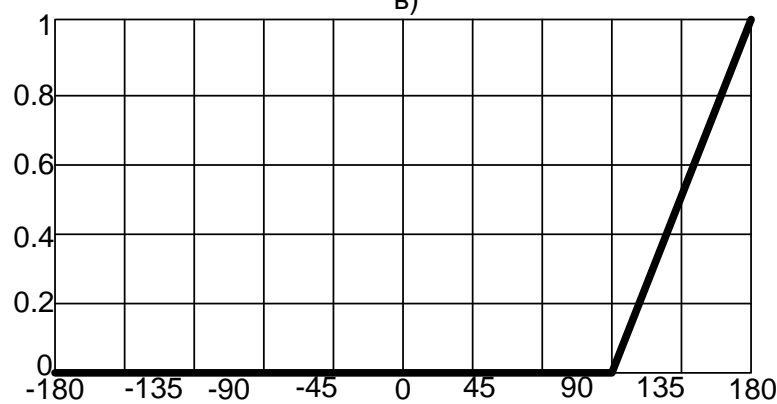
a)



б)



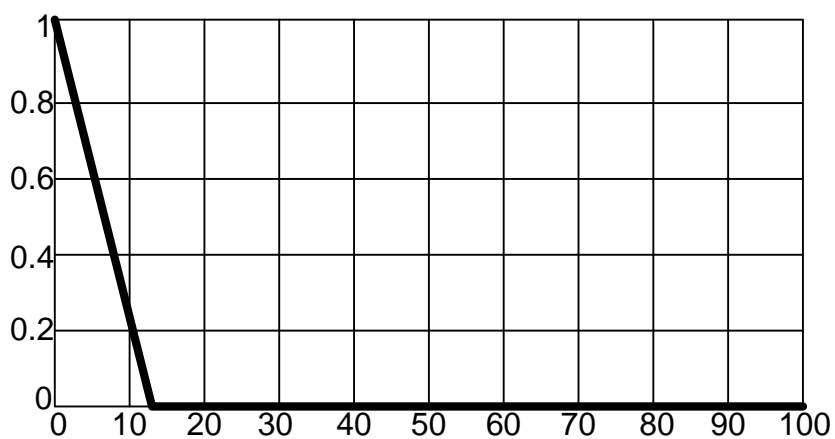
в)



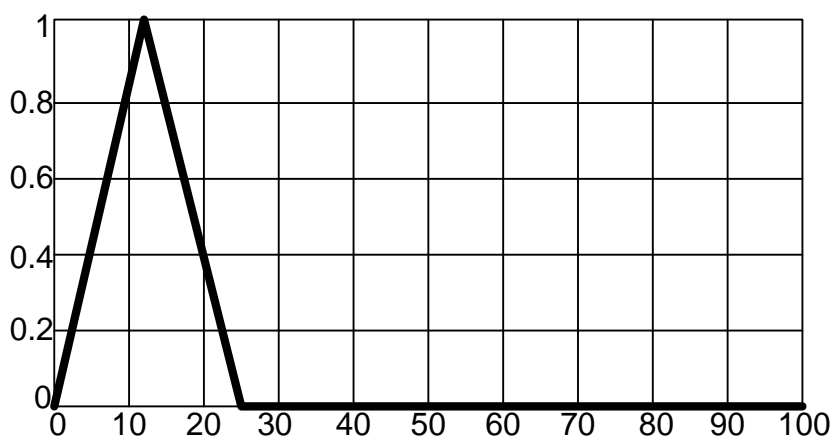
г)

a) Zero, б) SmallPos, в) MiddlePos, г) LargePos

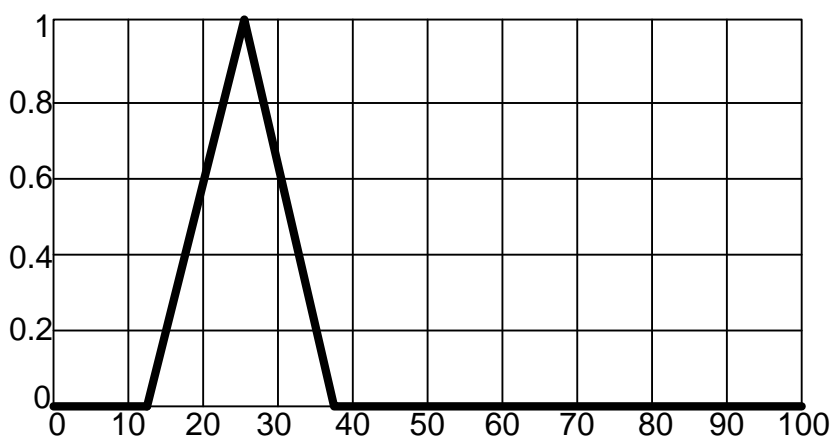
Рисунку 3.14 – Функції приналежності для вхідної змінної *Angle*



а)



б)



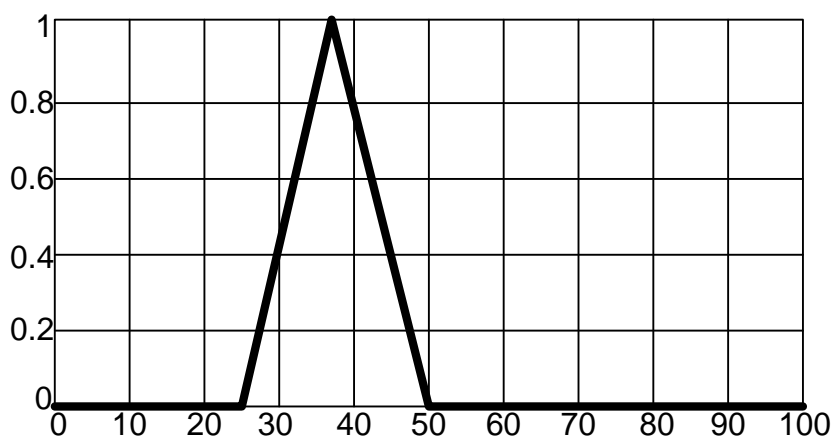
в)

а) Zero (нульова швидкість), б) LowSmallPos (низька мала швидкість),

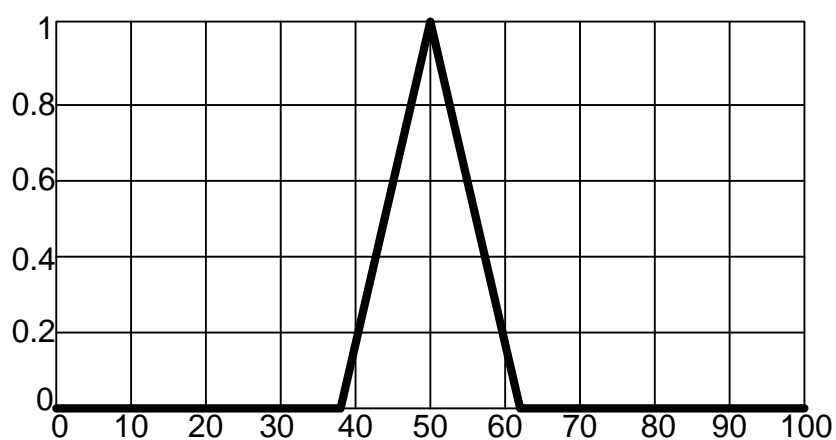
в) SmallPos (мала швидкість).

Рисунок 3.15 – Функції приналежності для вихідних змінних *VeloLeft* та

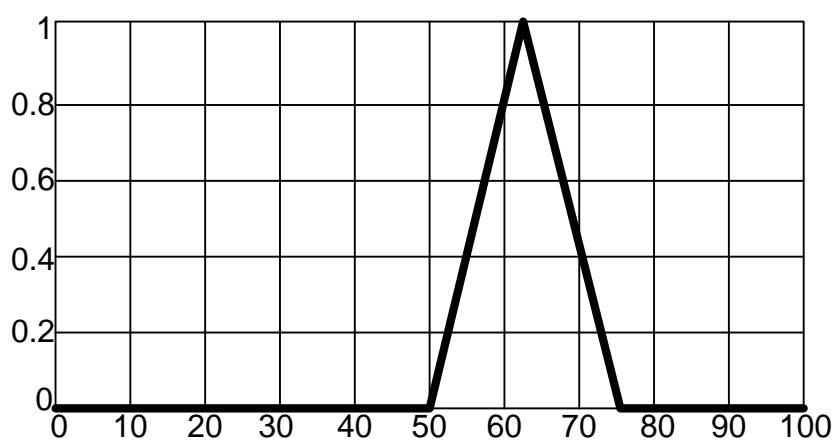
*VeloRight*



а)



б)

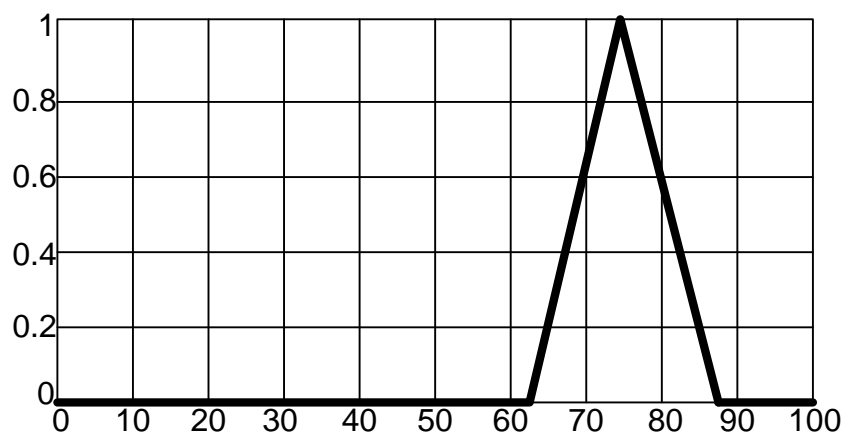


в)

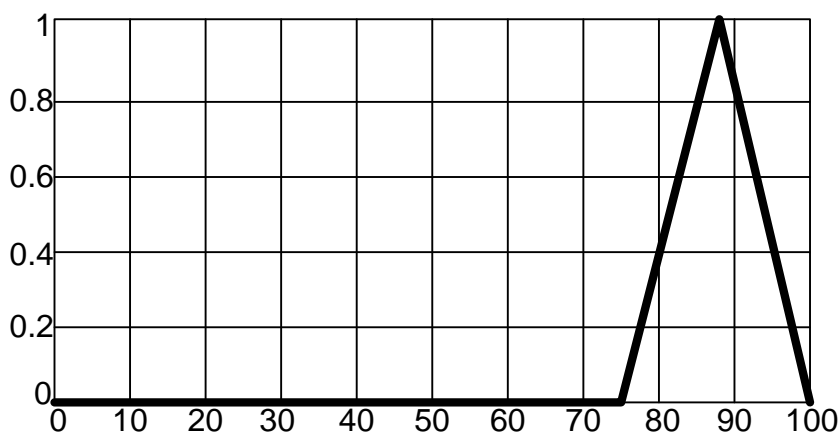
а) LowMiddlePos (низька середня позитивна), б) MiddlePos (середня позитивна), в) LowLargePos (низька велика позитивна).

Рисунок 3.16 – Функції приналежності для вихідних змінних *VeloLeft* та

*VeloRight*



а)



б)

а) LargePos (велика позитивна),

б) VeryLargePos (дуже велика позитивна).

Рисунок 3.17 – Функції приналежності для вихідних змінних *VeloLeft* та *VeloRight*

Розробимо правила *Rules* зв'язку для цих лінгвістичних змінних. Ці правила будуть складатись з кортежу усіх правил для об'єкту:

$$Rules = \langle Rule_1, Rule_2, \dots, Rule_N \rangle. \quad (3.30)$$

Тоді розробимо правила, які будуть входити в кортеж правил:

$$\text{Rule}_1 : \text{if } (Length == ShortLen) \text{ and } (Angle == Zero) \text{ then} \\ (VeloLeft == SmallPos) \text{ and } (VeloRight == SmallPos), \quad (3.31)$$

$$\text{Rule}_2 : \text{if } (Length == MiddleLen) \text{ and } (Angle == Zero) \text{ then} \\ (VeloLeft == MiddlePos) \text{ and } (VeloRight == MiddlePos), \quad (3.32)$$

$$\text{Rule}_3 : \text{if } (Length == LargeLen) \text{ and } (Angle == Zero) \text{ then} \\ (VeloLeft == LargePos) \text{ and } (VeloRight == LargePos), \quad (3.33)$$

$$\text{Rule}_4 : \text{if } (Length == ShortLen) \text{ and } (Angle == SmallPos) \text{ then} \\ (VeloLeft == SmallPos) \text{ and } (VeloRight == LowSmallPos), \quad (3.34)$$

$$\text{Rule}_5 : \text{if } (Length == ShortLen) \text{ and } (Angle == MiddlePos) \text{ then} \\ (VeloLeft == MiddlePos) \text{ and } (VeloRight == LowSmallPos), \quad (3.35)$$

$$\text{Rule}_6 : \text{if } (Length == ShortLen) \text{ and } (Angle == LargePos) \text{ then} \\ (VeloLeft == LargePos) \text{ and } (VeloRight == LowSmallPos), \quad (3.36)$$

$$\text{Rule}_7 : \text{if } (Length == ShortLen) \text{ and } (Angle == SmallNeg) \text{ then} \\ (VeloLeft == LowSmallPos) \text{ and } (VeloRight == SmallPos), \quad (3.37)$$

$$\text{Rule}_8 : \text{if } (Length == ShortLen) \text{ and } (Angle == MiddleNeg) \text{ then} \\ (VeloLeft == LowSmallPos) \text{ and } (VeloRight == MiddlePos), \quad (3.38)$$

$$\text{Rule}_9 : \text{if } (Length == ShortLen) \text{ and } (Angle == LargeNeg) \text{ then} \\ (VeloLeft == LowSmallPos) \text{ and } (VeloRight == LargePos), \quad (3.39)$$

$$\text{Rule}_{10} : \text{if } (Length == MiddleLen) \text{ and } (Angle == SmallPos) \text{ then} \\ (VeloLeft == MiddlePos) \text{ and } (VeloRight == LowMiddlePos), \quad (3.40)$$

$$\text{Rule}_{11} : \text{if } (Length == MiddleLen) \text{ and } (Angle == MiddlePos) \text{ then} \\ (VeloLeft == MiddlePos) \text{ and } (VeloRight == SmallPos), \quad (3.41)$$

$$\text{Rule}_{12} : \text{if } (Length == MiddleLen) \text{ and } (Angle == LargePos) \text{ then} \\ (VeloLeft == LargePos) \text{ and } (VeloRight == MiddlePos), \quad (3.42)$$

$$\text{Rule}_{13} : \text{if } (Length == ShortLen) \text{ and } (Angle == SmallNeg) \text{ then} \\ (VeloLeft == LowMiddlePos) \text{ and } (VeloRight == MiddlePos), \quad (3.43)$$

$$\text{Rule}_{14} : \text{if } (Length == MiddleLen) \text{ and } (Angle == MiddleNeg) \text{ then} \\ (VeloLeft == SmallPos) \text{ and } (VeloRight == MiddlePos), \quad (3.44)$$

$$\text{Rule}_{15} : \text{if } (Length == MiddleLen) \text{ and } (Angle == LargeNeg) \text{ then} \\ (VeloLeft == MiddlePos) \text{ and } (VeloRight == LargePos), \quad (3.45)$$

$$\text{Rule}_{16} : \text{if } (Length == LongLen) \text{ and } (Angle == SmallPos) \text{ then} \\ (VeloLeft == LargePos) \text{ and } (VeloRight == LowLargePos), \quad (3.46)$$

$$\text{Rule}_{17} : \text{if } (Length == LongLen) \text{ and } (Angle == MiddlePos) \text{ then} \\ (VeloLeft == LargePos) \text{ and } (VeloRight == MiddlePos), \quad (3.47)$$

$$\text{Rule}_{18} : \text{if } (Length == LongLen) \text{ and } (Angle == LargePos) \text{ then} \\ (VeloLeft == LargePos) \text{ and } (VeloRight == LowMiddlePos), \quad (3.48)$$

$$\text{Rule}_{19} : \text{if } (Length == LongLen) \text{ and } (Angle == SmallNeg) \text{ then} \\ (VeloLeft == LowLargePos) \text{ and } (VeloRight == LargePos), \quad (3.49)$$

$$\text{Rule}_{20} : \text{if } (Length == LongLen) \text{ and } (Angle == MiddleNeg) \text{ then} \\ (VeloLeft == MiddlePos) \text{ and } (VeloRight == LargePos), \quad (3.50)$$

$$\text{Rule}_{21} : \text{if } (Length == LargeLen) \text{ and } (Angle == LargeNeg) \text{ then} \\ (VeloLeft == LowMiddlePos) \text{ and } (VeloRight == LargePos). \quad (3.51)$$

На вхід моделі подаються довжина шляху та кут повороту робота, що отримуються, на виході отримуються швидкості лівого та правого коліс для того, щоб робот зміг дістатись заданої точки маршруту.

Докладніше про створення нечіткого регулятора у розділі 3.

Оскільки вхідна інформація в системі керування може бути визначена у вигляді елементів терм-множин лінгвістичних змінних, то доречним є використання нечітких регуляторів в системах керування мобільними роботами [76].

### 3.6 Висновки за розділом 3

В ході виконання розділу 3 було проведено моделювання засобів інтелектуального керування роботами.

Вдосконалено модель розпізнавання та ідентифікації об'єктів ГІВС за допомогою методів комп'ютерного зору за рахунок визначення процедури розпізнавання та ідентифікації об'єктів РП ГІВС.

Отримала подальший розвиток модель керування мобільним роботом у просторі ГІВС, що здійснює керування шасі двоколісного робота, та, на відміну від існуючих, функціонує на основі інформації від об'єктових/локальних систем комп'ютерного зору та виконання функцій розпізнавання та ідентифікації об'єктів ГІВС, забезпечує розрахунок координат та швидкостей роботизованої платформи та інших об'єктів ГІВС, розрахунок лінійних та криволінійних траєкторій переміщень у дискретному робочому просторі, подання маршруту у вигляді послідовностей переміщень, забезпечує організацію керування транспортувальними роботами за їх візуальним положенням; нечітка складова моделі містить розгорнутий опис процедур формування лінгвістичних змінних, пов'язаних з динамікою руху мобільної роботизованої платформи та формалізує набір продукційних правил нечіткого регулятора.

Також у розділі 3 вдосконалено метод адаптивного візуального керування, що описує сукупність прийомів та операцій, які слід застосувати у інтелектуальних системах керування роботів для планування завдання переміщень на основі отримання зорової інформації від систем комп'ютерного (технічного) зору, та, на відміну від існуючих методів, орієнтований на використання в мобільних робототехнічних системах.

Основні результати цього розділу опубліковано в роботах [65], [70], [73], [74], [78].

## РОЗДІЛ 4

### МОДЕЛЮВАННЯ АДАПТИВНИХ СИСТЕМ КЕРУВАННЯ ЗАСОБАМИ GNU OCTAVE

#### 4.1 Знаходження передавальних функцій двигуна

Для дослідження системи адаптивного керування використовується пакет програм GNU Octave, який дозволяє провести її комп'ютерне моделювання.

В побудованому роботі використовуються мотори з редуктором Arduino, що представлено на рис. 4.1.



Рисунок 4.1 – Мотор Arduino

Він має такі технічні характеристики [62]:

- діапазон напруг живлення 3В – 8В;
- номінальний струм споживання при напрузі 3,6 В – 240 мА;
- передаточне число редуктора – 1/48;
- швидкість обертання при напрузі 3,6 В без навантаження – 170 об / хв.
- момент, що крутить при напрузі 6В – 800 г / см;
- діаметр вала – 5.4 мм;
- габарити (для прямої модифікації) – 64мм x 20мм x 20мм;
- маса 26 грам
- ККД = 65-70%.



Тоді можна провести розрахунки. Номінальна кутова швидкість мотора складатиме:

$$\omega_n = \frac{\pi n_n}{30} = \frac{3.14 \times 170}{30} = 17.8 \text{ об / хв}$$

де  $n_n$  – швидкість обертання в об/хв.

Потужність двигуна визначається наступним чином:

$$P = I \times U \times \text{ККД} = 3.6 \times 0.240 \times 0.65 = 0.561 \text{ Вт}$$

де  $I$  та  $U$  – номінальний струм та напруга.

Момент двигуна можна розрахувати за наступною формулою:

$$M = 9.55 \frac{P}{n_n} = 9.55 \frac{0.561}{170} = 0.031$$

де  $P_n$  – номінальна потужність [Вт].

Опір  $R$  двигуна буде дорівнювати:

$$R = \frac{U}{I} = \frac{0.240}{3.6} = 0.066 \text{ Ом}$$

Втрати на проводах становлять приблизно 20%, тому:  
 $R \times 0.8 = 0.066 \times 0.8 = 0.053 \text{ Ом}$

Момент інерції  $J$  буде визначатись таким чином:

$$J = \frac{m \times g \times r}{2} = \frac{0.378 \times 9.8 \times 0.035}{2} = 0.065 \text{ кг} \cdot \text{м}^2$$

Передавальна функція двигуна має наступний вигляд:

$$W_{\delta} = \frac{K_{\delta}}{p(T_{\delta}p + 1)}$$

де  $K_{\delta}$  – коефіцієнт підсилення двигуна,  $T_{\delta}$  – електромеханічна постійна часу.

Ці параметри знаходяться за формулами:

$$K_{\delta} = \frac{\omega_n}{U} = \frac{17.8}{0.24} = 74.1$$

$$T_{\delta} = \frac{1.2 \times J \times R \times I \times \omega_n}{M(U - 1.2 \times R \times I)}$$

$$T_{\delta} = \frac{1.2 \times 0.065 \times 0.053 \times 3.6 \times 17.8}{0.031(0.240 - 1.2 \times 0.053 \times 3.6)} = \frac{0.265}{0.00034} = 779.4$$

Отже передавальна функція двигуна матиме вигляд:

$$W_{\delta} = \frac{74.1}{p(779.4p + 1)}$$

Також слід ураховувати передавальну функцію редуктора:

$$W_{ред}(p) = k_{ред} = \frac{1}{n_n} = \frac{1}{170} = 0.006.$$

Наведені розрахунки будуть використані для моделювання адаптивного режиму роботи двигунів мобільних роботів.

## 4.2 Моделювання адаптивного режиму роботи одного двигуна

Загальна структурна схема системи двигуном буде мати вигляд (рис. 4.2) і буде складатись з послідовного розташування передавальних функцій, розрахованих раніше і матиме зворотній зв'язок:

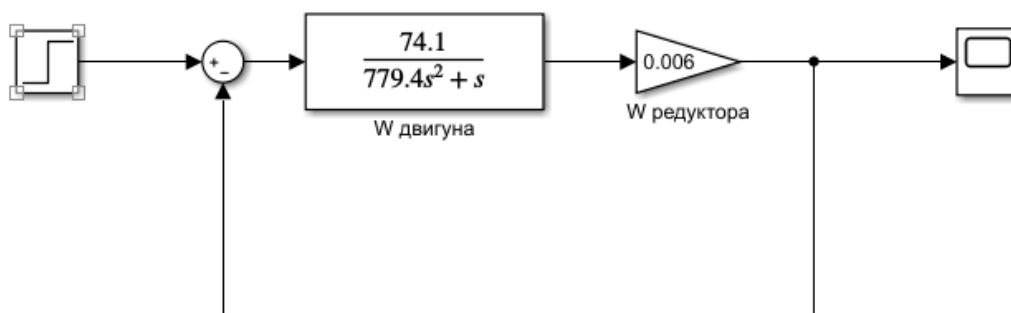


Рисунок 4.2 – Структурна схема системи двигуна

Зробивши перетворення структурної схеми за правилами перетворення отримали наступну (рис. 4.3):

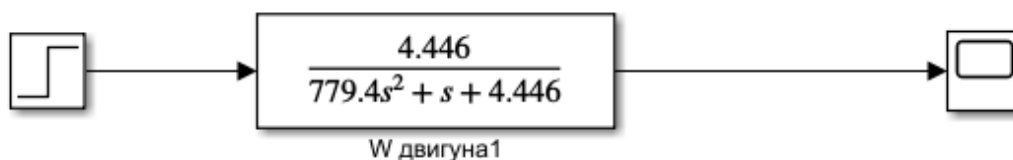


Рисунок 4.3 – Результат перетворення структурної схеми двигуна

Розробимо структурну схему адаптивної системи керування. Вона буде складатись з:

- об'єкту керування  $Obj_{Con}$  ;
- регулятора  $regulator$  ;
- еталонної моделі  $E_{mod}$  ;
- сигналу помилки  $Error$  ;

–контуру самоналаштування *STuning* .

Загальна схема адаптивної системи представлена на рис. 4.4.

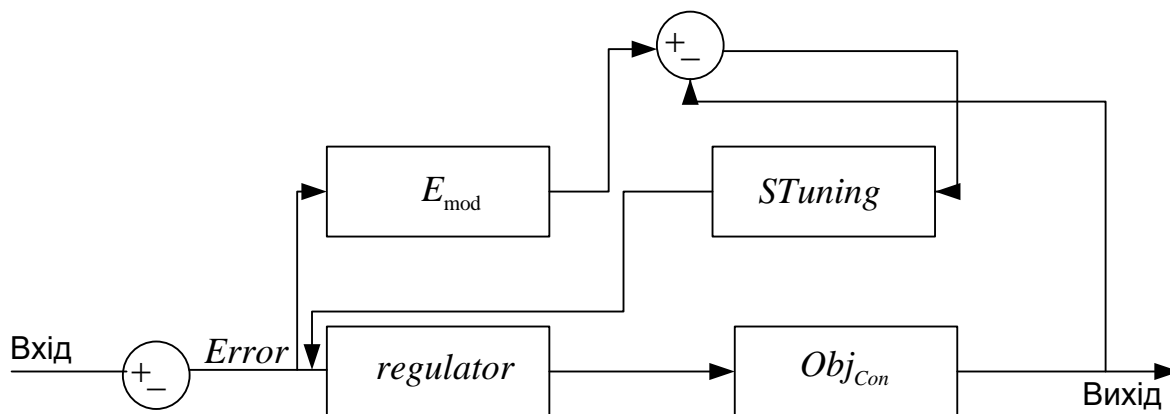


Рисунок 4.4 – Загальна схема адаптивної системи

У схемі є вхід і вихід, які позначимо через  $x$  та  $y$ . Аналогічно, у еталонної моделі теж є вихід, який позначимо через  $y_{E_{mod}}$  .

Якщо відбувається зміна параметрів об'єкта керування при взаємодії з еталонною моделлю відбувається налаштування регулятора за допомогою контуру самоналаштування [69]. З цього виходить, що головною метою адаптації системи є наступний вираз мінімізації помилки:

$$Error = y_{E_{mod}} - y = 0$$

Щоб провести настроювання параметрів блоку *STuning* , можна використати функцію Ляпунова для зміни параметрів регулятора:

$$reg_{param} = \frac{1}{k} \int disag \times Error$$

де  $reg_{param}$  – параметр, необхідний для настроювання регулятора;

$disag$  – параметр, який відповідає за непогодженість;

$k$  – коефіцієнт підсилення.

На жаль, для того, щоб знайти бажані показники якості, потрібно змінювати деякі параметри системи [74], [80]. Було обрано в якості еталонної моделі таку передавальну функцію:

$$W_{E_{\text{mod}}}(s) = \frac{0.4446}{0.001s^2 + s}.$$

Також значення коефіцієнту буде дорівнювати  $\frac{1}{0.4666}$ .

Графік перехідної характеристики еталонної моделі представлено на рис. 4.5.

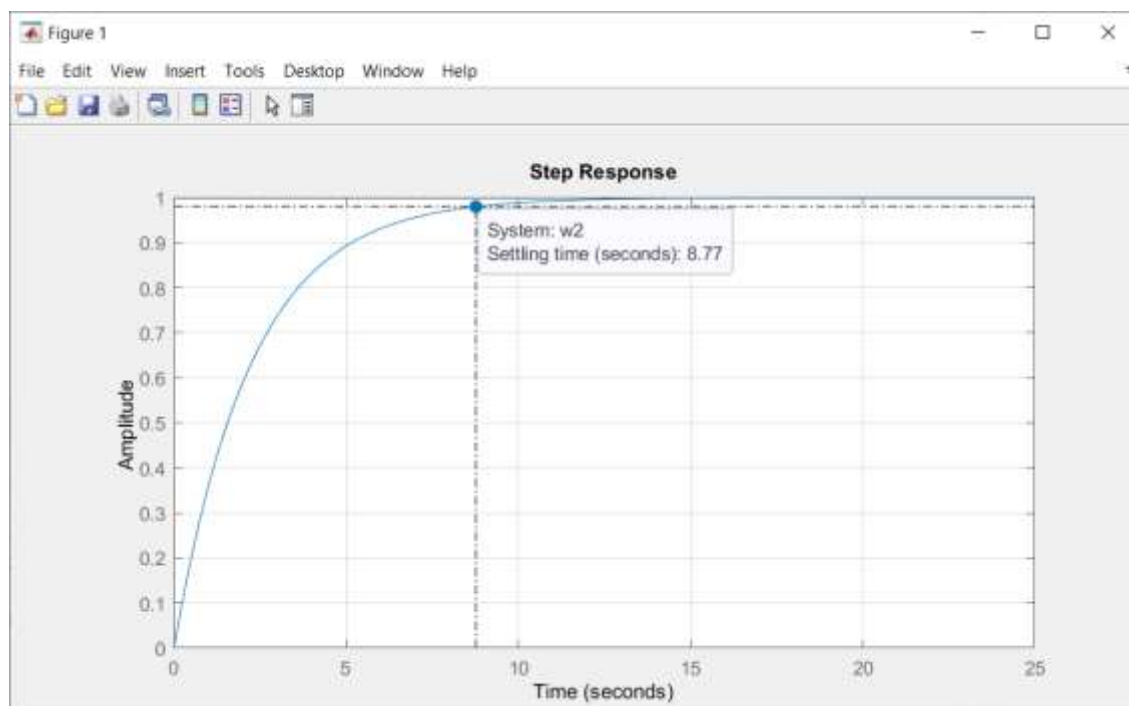


Рисунок 4.5 – Перехідна характеристика еталонної моделі

Час перехідного процесу = 8.77 сек, перерегулювання та похибки дорівнюють нулю, що є бажаними показниками якості.



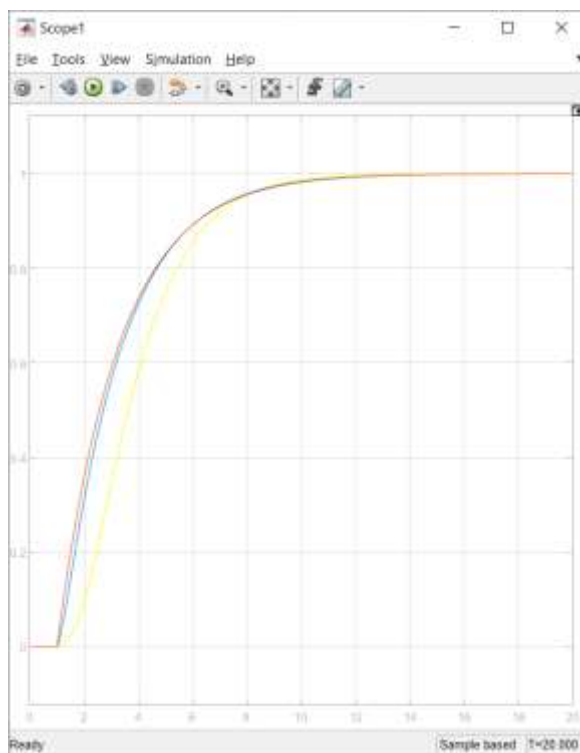


Рисунок 4.7 – Графіки перехідних характеристик адаптивної системи для різних показників коефіцієнтів підсилення

На 4 секунді помилка складає близько 14%, але вже на 7 – 1%, на 10 секунді – 0.1%.

Проведемо експеримент щодо оцінки якості адаптації за різних значень коефіцієнту підсилення  $k$  (від 0 до 50) і при значеннях показника часу регулювання в 10 секунд.

Таблиця 4.1 – Оцінка якості адаптації на 10 секундах для різних показниках коефіцієнту підсилення для одного двигуна

Різниця у коефіцієнтах підсилення	Помилка адаптації до зміни, %	Помилка адаптації до еталонної моделі, %
0	0	0.0036
0.1	0.0047	0.0061
0.2	0.0053	0.0087
0.3	0.0092	0.0199

Продовження таблиці 4.1

Різниця у коефіцієнтах підсилення	Помилка адаптації до зміни, %	Помилка адаптації до еталонної моделі, %
0.4	0.014	0.0238
0.5	0.016	0.0393
1	0.021	0.0729
2	0.029	0.0953
3	0.042	0.0945
4	0.047	0.0987
5	0.044	0.1
10	0.06	0.17
20	0.083	0.15
30	0.11	0.21
40	0.132	0.297
45	0.174	0.381
46	0.179	0.383
47	0.174	0.383
48	0.187	0.367
49	0.199	0.398
50	0.2	0.4

Як видно з таблиці, при зміні коефіцієнтів підсилення та постійної часу, показники похибок змінились на деяку величину, максимальне значення якої дорівнює 0.4%, що свідчить про високий рівень адаптації.

#### 4.3 Моделювання адаптивного режиму роботи двох двигунів

Аналогічно до моделі, представленої на рис. 4.6, було вирішено зробити моделювання для двох двигунів (рис. 4.8).



У реальному роботі, що використовувався в експерименті, присутні 2 однакові двигуни, тобто їх параметри ідентичні, тому при дослідженні отримуються аналогічні результати.

На практиці параметри двигунів можуть відрізнятись на 1-5%, тобто передавальні функції будуть різними. Якщо для першого випадка передавальна функція була  $W_0 = \frac{4.466}{779.4s^2 + s}$ , то для другої змінимо параметри на 5% і проведемо дослідження. Передавальна функція другого двигуна буде дорівнювати:

$$W_{0_2} = \frac{4.2427}{740.4s^2 + s}.$$

Результати моделювання двох двигунів та графіки їх перехідних функцій представлено на рис. 4.9 та у таблиці 4.2.

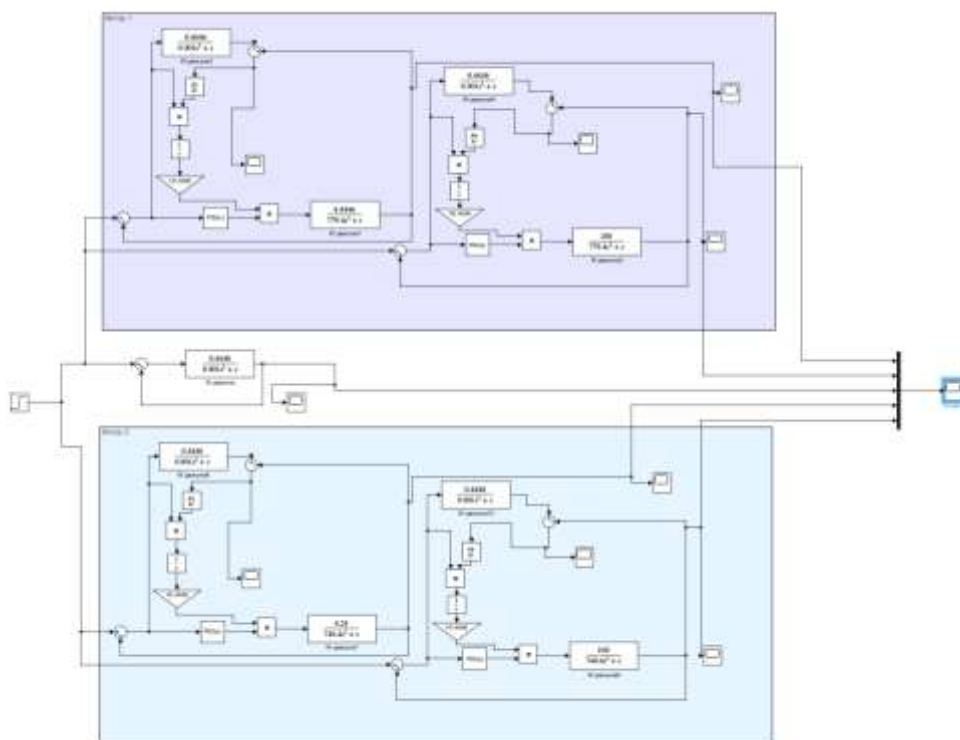


Рисунок 4.8 – Модель адаптивної системи для двох двигунів

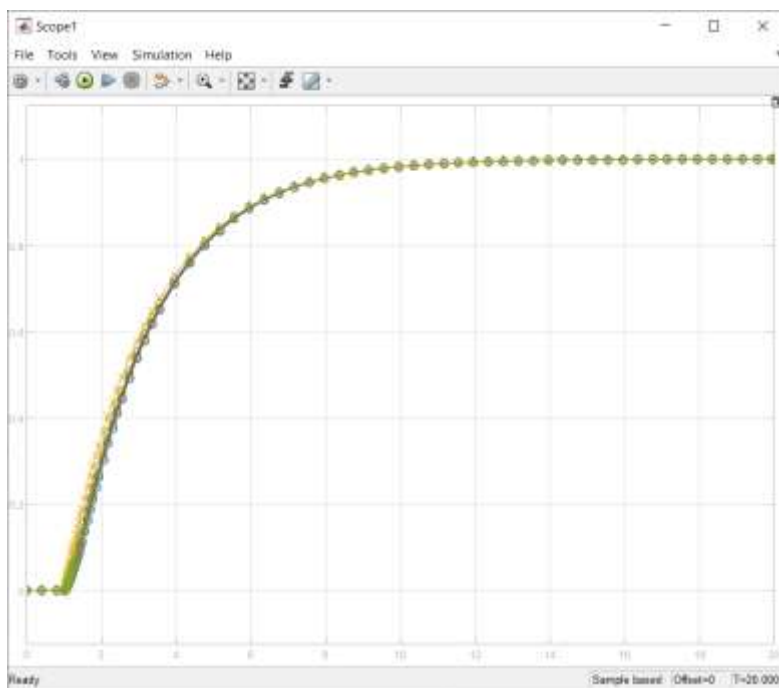


Рисунок 4.9 – Моделювання роботи 2 двигунів

Таблиця 4.2 – Оцінка якості адаптації на 10 секундах для різних значень коефіцієнту підсилення для двох двигунів

Різниця у коефіцієнтах підсилення	Помилка адаптації до зміни двигуна, % до 1	Помилка адаптації до еталонної моделі двигуна, % до 1	Помилка адаптації до зміни 2 двигуна, % до 2	Помилка адаптації до еталонної моделі двигуна, % до 2
0	0	0.0036	0	0.0037
0.1	0.0047	0.0061	0.0048	0.0061
0.2	0.0053	0.0087	0.0054	0.0089
0.3	0.0092	0.0199	0.0091	0.0203
0.4	0.014	0.0238	0.0141	0.0249
0.5	0.016	0.0393	0.0183	0.0398
1	0.021	0.0729	0.027	0.0784
2	0.029	0.0953	0.043	0.0961
3	0.042	0.0945	0.044	0.0977
4	0.047	0.0987	0.068	0.1246

Продовження таблиці 4.2

Різниця у коефіцієнтах підсилення	Помилка адаптації до зміни двигуна, % 1	Помилка адаптації до еталонної моделі двигуна, % 1	Помилка адаптації до зміни 2 двигуна, %	Помилка адаптації до еталонної моделі двигуна, % 2
5	0.044	0.1	0.063	0.192
10	0.06	0.17	0.076	0.184
20	0.083	0.15	0.088	0.167
30	0.11	0.21	0.14	0.263
40	0.132	0.297	0.1347	0.321
45	0.174	0.381	0.1753	0.385
46	0.179	0.383	0.1771	0.388
47	0.174	0.383	0.1778	0.386
48	0.187	0.367	0.1943	0.391
49	0.199	0.398	0.2001	0.404
50	0.2	0.4	0.22	0.42

Як видно з таблиці, при зміні коефіцієнтів підсилення та постійної часу на 5% показники похибок теж змінились на деяку величину, максимальне значення якої дорівнює 0.02%, що теж свідчить про високий рівень адаптації.

#### 4.4 Моделювання розрахунку швидкостей та крутного моменту переміщення робототехнічної платформи

Для того, щоб правильно розрахувати траєкторії руху робота було прийнято рішення у експериментальному знаходженні швидкостей кожного з двигунів робота та їх крутного моменту для кожного з приводних коліс.

Загальна схема розробленої системи представлена на рис. 4.10.

Початкова точка переміщення має 3 складових – координати роботи ( $x, y$ ) та початковий кут повороту відносно осі координат.

Проміжними точками переміщення є точки дискретного простору, які в сумі складають шлях переміщення робота у робочому просторі.

На основі цих даних, отримуються швидкість та кутова швидкість робота. Далі ці швидкості перетворюються в цільові швидкості для лівого та правого колеса.

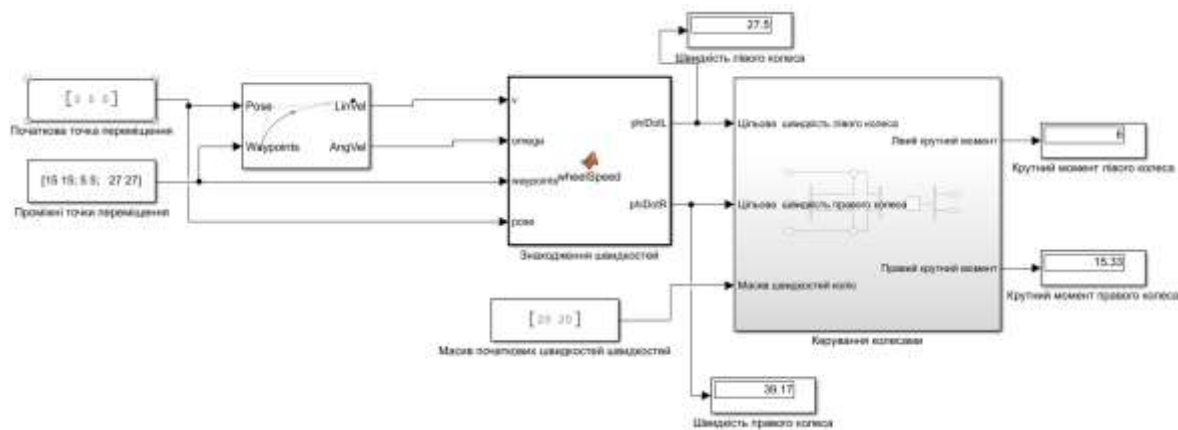


Рисунок 4.10 – Схема моделювання крутного моменту коліс

Потім на основі цих швидкостей і початково заданих швидкостей коліс за допомогою ПІД-регулятора, а саме П-складової вже визначаються крутні моменти для лівого та правого колеса, які можна використовувати для переміщення робота з початкової точки переміщення у першу точку дискретного робочого простору. Далі початкова точка змінюється на першу точку з масиву точок переміщення і вираховується крутний момент для переміщення до наступної точки.

Провівши розрахунки, які було запропоновано у 2 розділі для обрахування шляху, були отримані наступні координати, куди переміститься робот (таблиця 4.3).

Таблиця 4.3 – Координати переміщення робота

Початкова точка переміщення	Кінцева точка переміщення	З використанням даних моделі	Похибка, %
(0;0)	(10;10)	(10.04;9.97)	0.7
(10;17)	(18;20)	(18.1;19.83)	1.3

Продовження таблиці 4.3

Початкова точка переміщення	Кінцева точка переміщення	З використанням даних моделі	Похибка, %
(30;20)	(27;5)	(27;5.09)	1.12
(7;13)	(19;47)	(19.03;46.99)	0.4
(5;44)	(26;13)	(26.01;13.033)	0.6
(40;8)	(20;20)	(19,91;19,94)	0.3
(16;22)	(44;41)	(44.099;41.1)	0.7
(12;2)	(20;18)	(20.17;18.21)	1.1
(25;15)	(11;1)	(11.12;1.12)	0.9
(13;13)	(28;4)	(28.07;3.996)	0.5

Результати, наведені у таблиці 4.3, показують, що похибка переміщення робота відповідно до розробленої моделі, складає не більше 1.3%, що є досить високим показником.

#### 4.5 Моделювання нечіткого адаптивного керування роботом

За допомогою GNU Octave було створено схему системи нечіткого адаптивного керування роботом, що представлено на рис. 4.11.

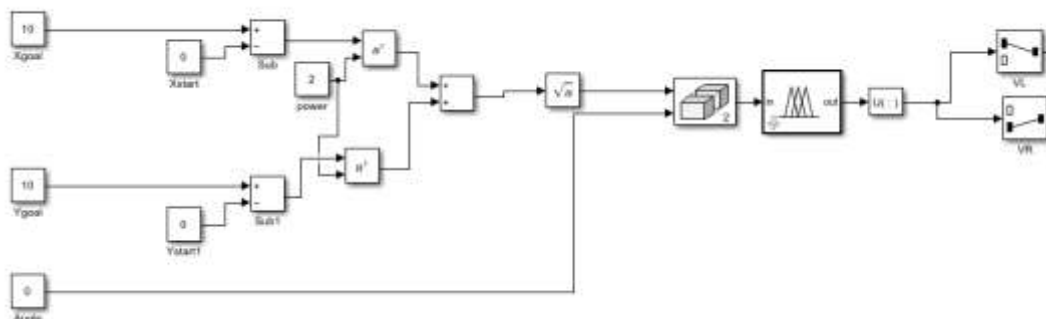


Рисунок 4.11 – Модель нечіткого адаптивного керування роботом

На вході моделі – координати кінцевої точки переміщення робота ( $X_{goal}, Y_{goal}$ ), початкові координати робота ( $X_{start}, Y_{start}$ ) та кут повороту робота ( $Angle$ ), що надходять до системи за допомогою файлу з пакету PyCharm, а саме з програм ідентифікації робота, отримання його координат (центр мас прямокутника, що обмежує робота) і з програми отримання куту повороту робота.

Обраховується маршрут, що потрібно пройти роботу (рис. 4.12).

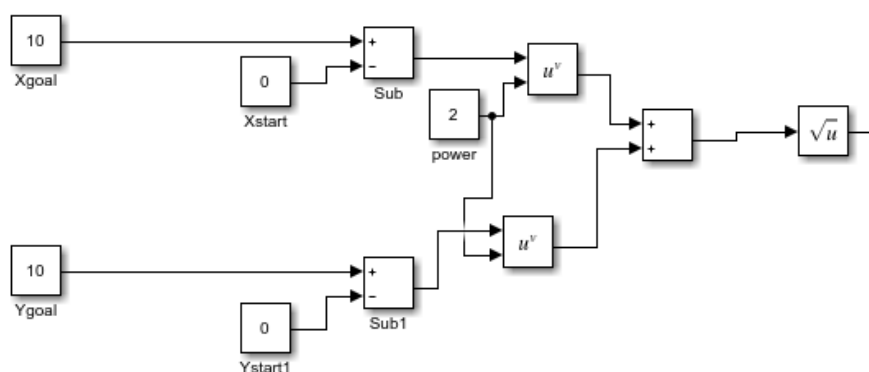


Рисунок 4.12 – Розрахунок шляху робота

Далі параметри шлях та кут повороту робота передаються на блок нечіткого регулятора, який має наступний вид (рис. 4.13).

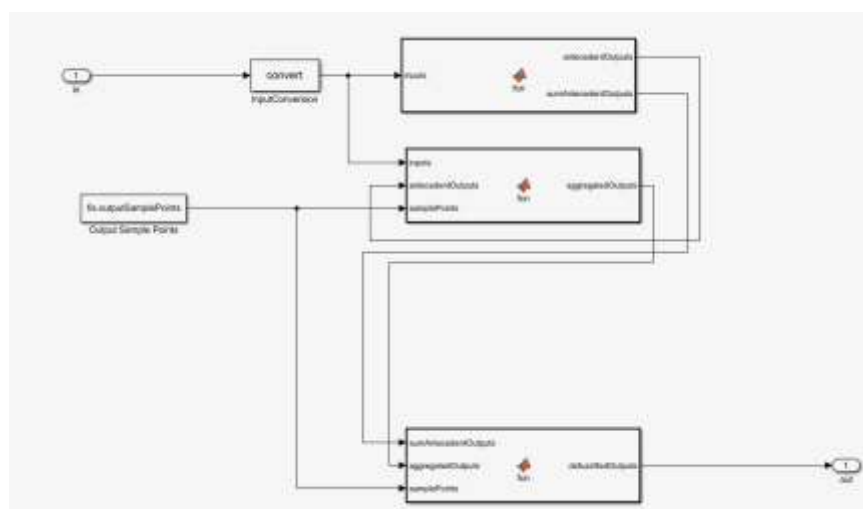


Рисунок 4.13 – Блок нечіткого регулятора

Цей блок включає в себе всі пункти з розділу 3, де створювалась модель нечіткого адаптивного керування.

У якості бази з правилами у цей блок завантажується файл MotorVelo\_2.

Розглянемо докладніше створення цього файлу. Він створювався на основі пакету FuzzyLogicDesigner. Зовнішній вигляд нечіткого регулятора, а саме моделі з правилами представлено на рис. 4.14.

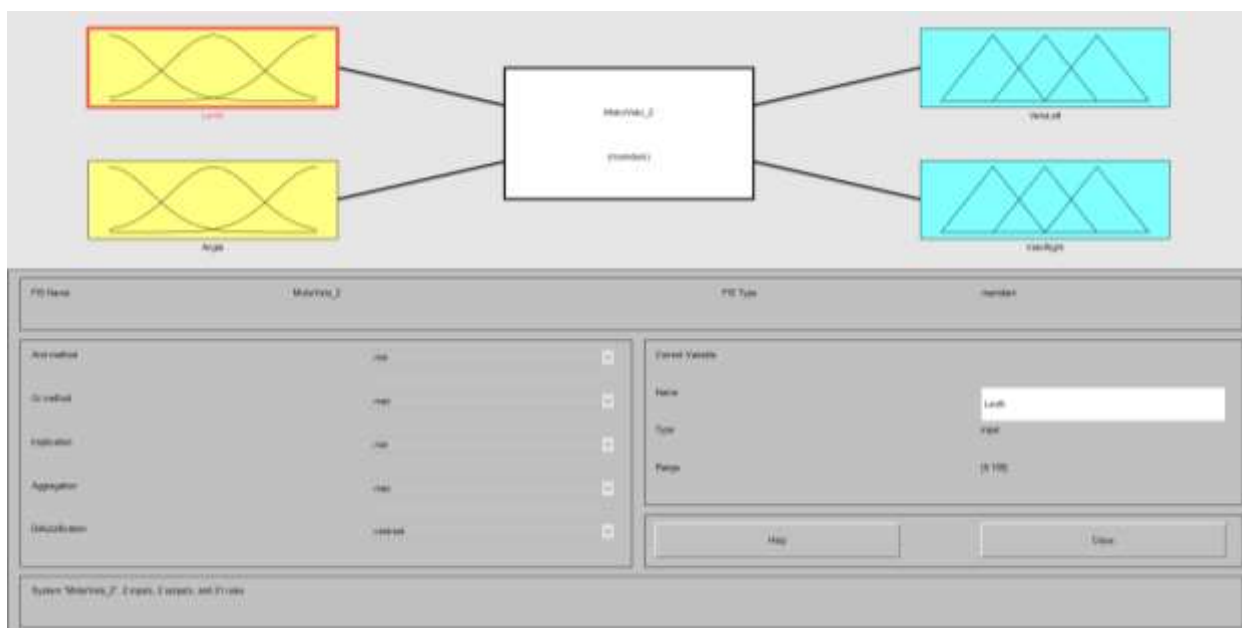


Рисунок 4.14 – Модель нечіткого регулятора

Як було описано у попередній главі – на вході є два параметри. Першим є *Length* – довжина шляху, другим – *Angle* – кут повороту робота.

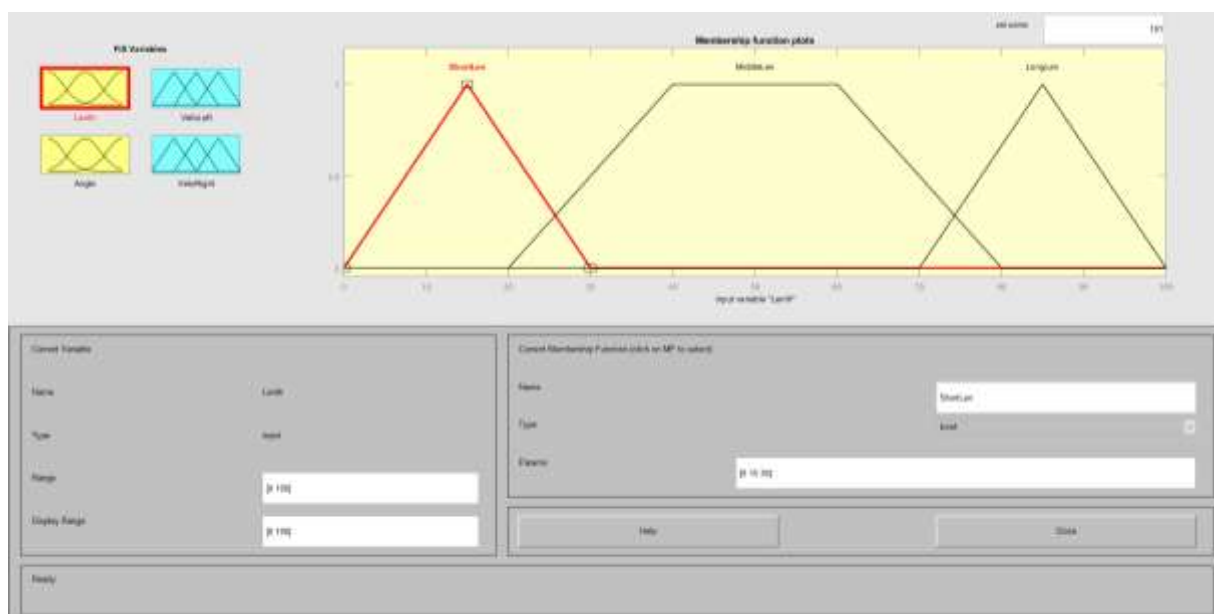
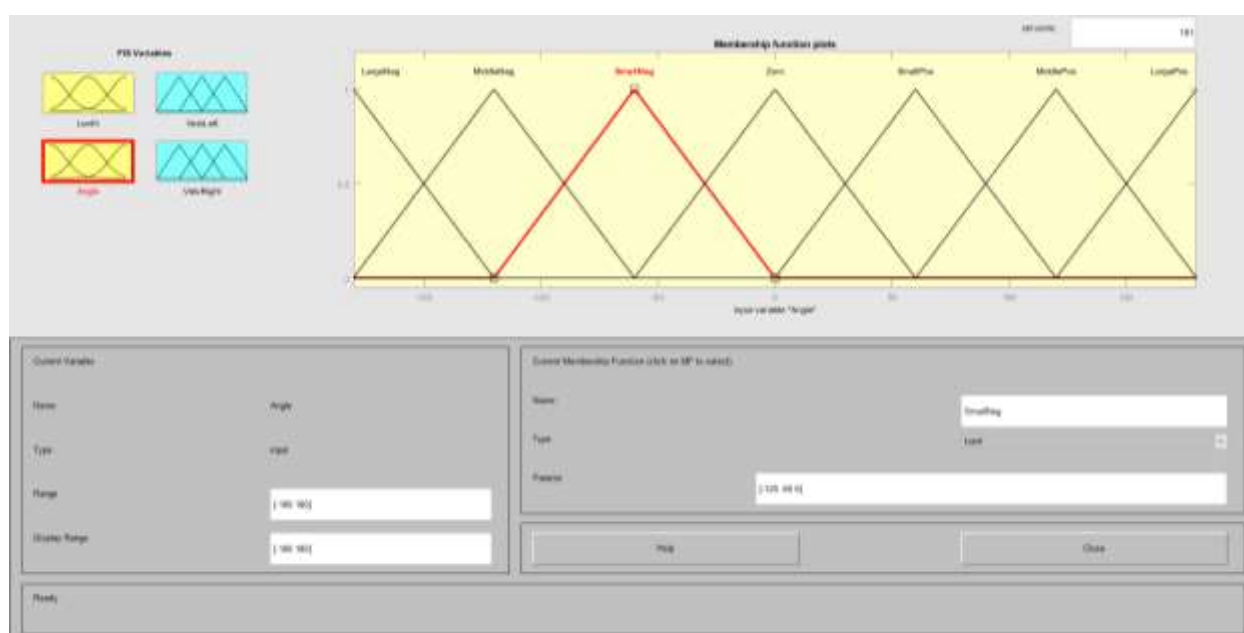
Вихідними є ліва *VeloLeft* і права *VeloRight* швидкості робота.

Функції приналежності для параметру *Length* представлено на рис. 4.15.

Функції приналежності для параметру *Angle* представлено на рис. 4.16.

Функції приналежності для параметрів *VeloLeft* і *VeloRight* представлено на рис. 4.17.

Після цього, правила, описані у попередній главі були занесені до блоку нечіткого регулятора (рис. 4.18).

Рисунок 4.15 – Налаштування вхідного параметру *Length*Рисунок 4.16 – Налаштування вхідного параметру *Angle*



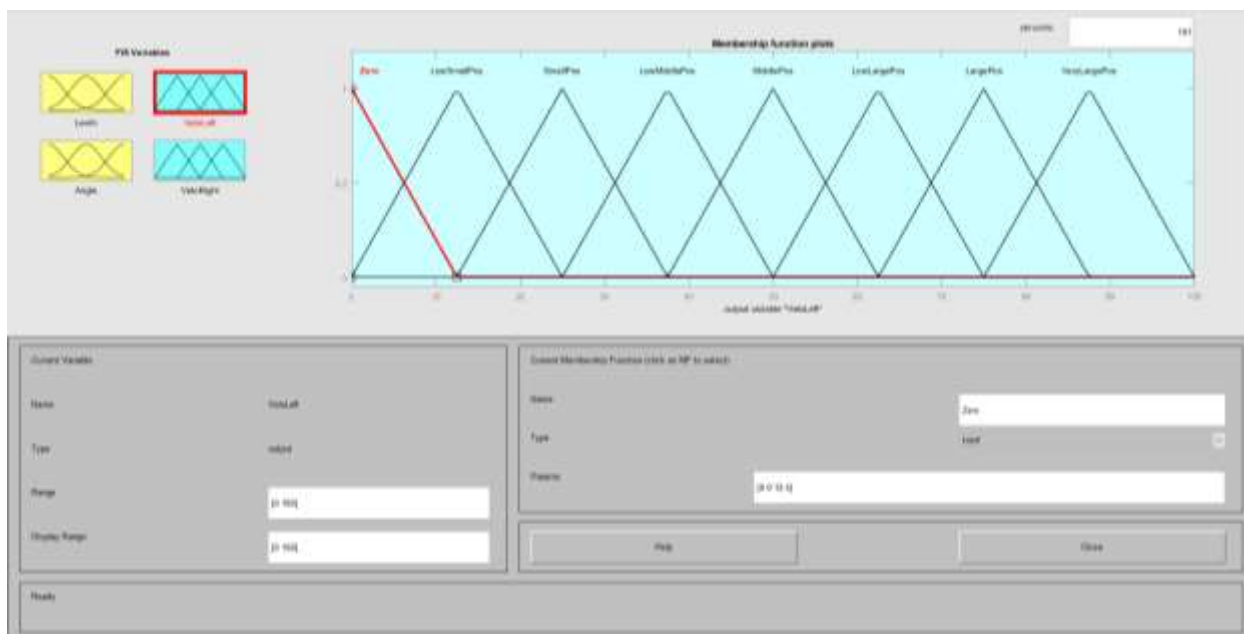


Рисунок 4.17 – Налаштування вихідних параметрів *VeloLeft* і *VeloRight*

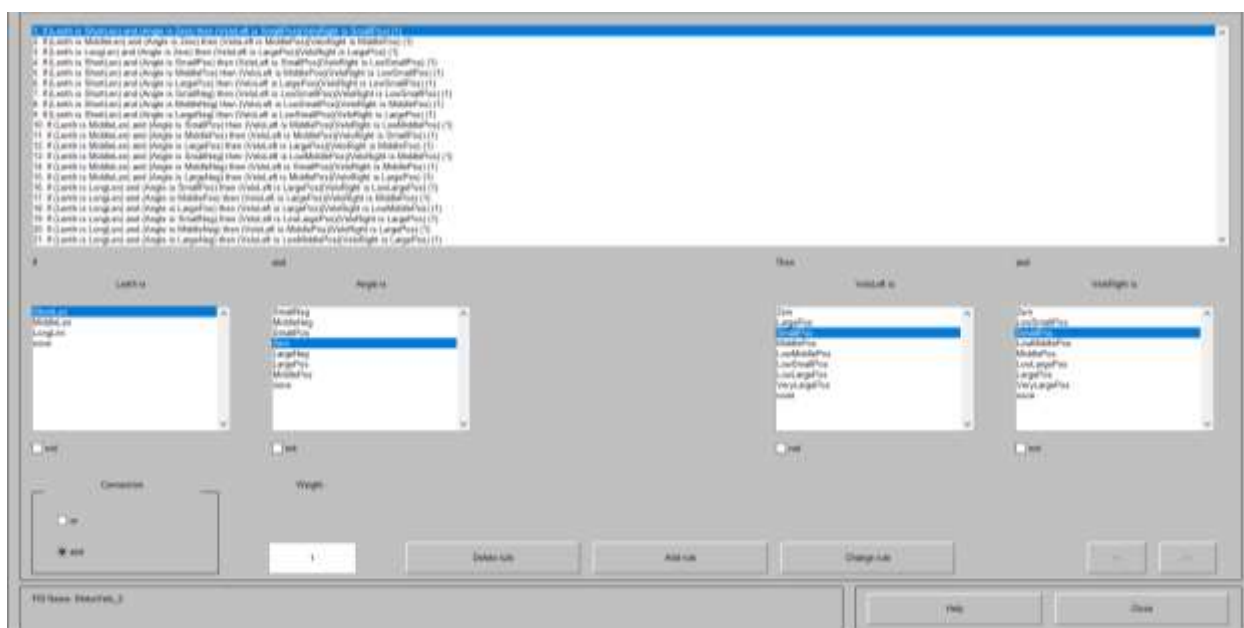


Рисунок 4.18 – Внесення правил нечіткого регулятора

Проведемо перевірку моделювання, згідно правил і різних показників входів нечіткого регулятора.

Результати перевірки представлено на рис. 4.19 з вхідними показниками [50;40] та 4.20 з показниками [90;-120].

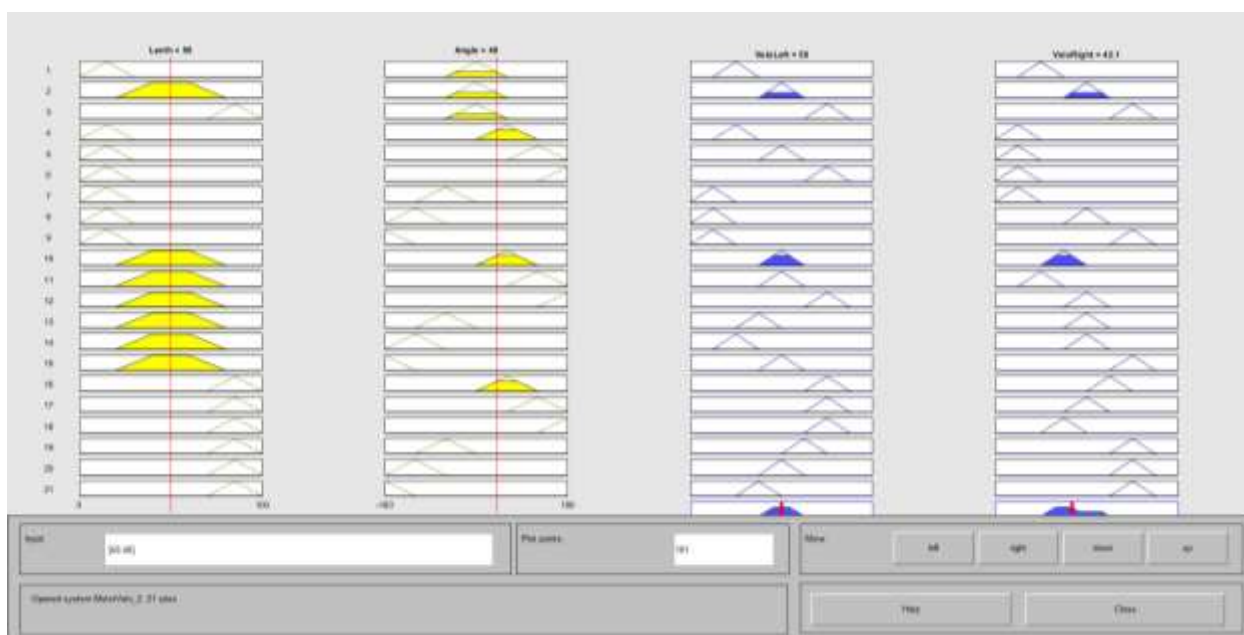


Рисунок 4.19 – Результати виконання правил з параметрами [50;40]

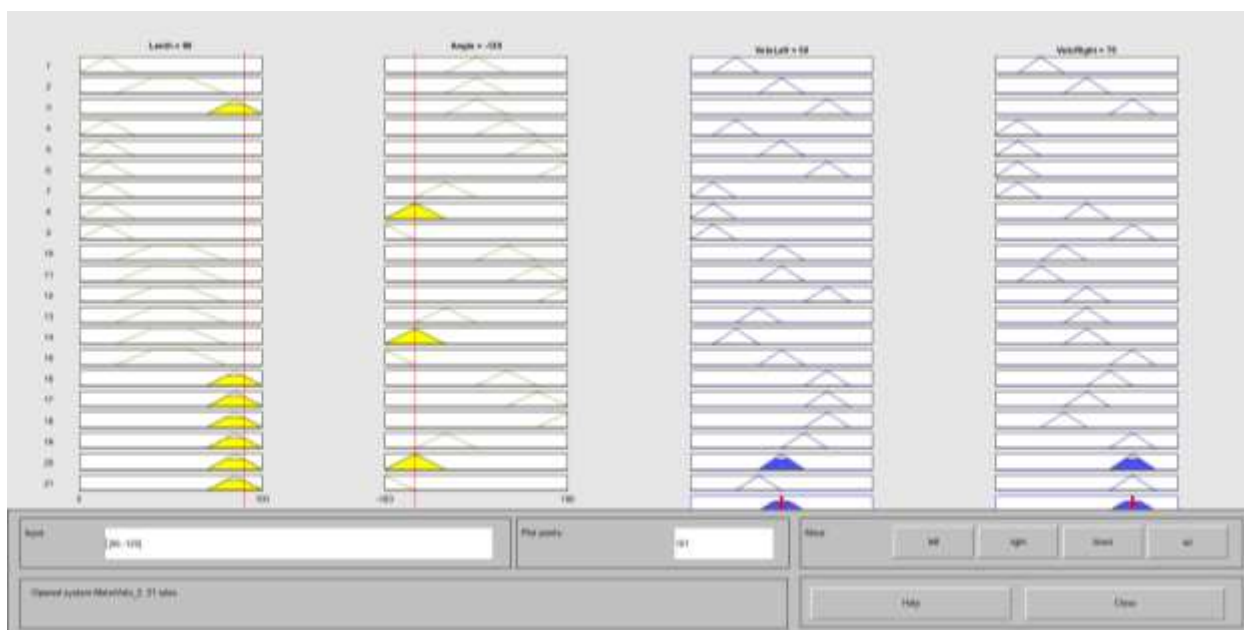


Рисунок 4.20 – Результати виконання правил з параметрами [90;-120]

Залежність між довжиною шляху *Length*, кутом повороту робота *Angle* і швидкістю лівого колеса *VeloLeft* представлено на рис. 4.21-4.23.

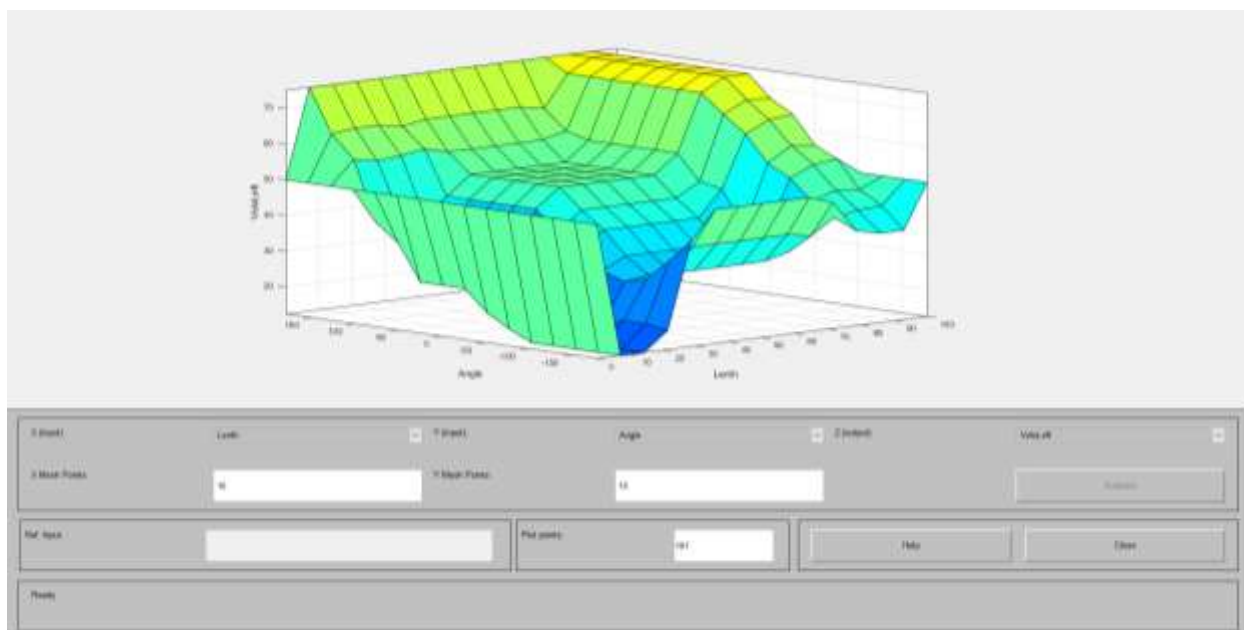


Рисунок 4.21 – Залежність між *Length*, *Angle* і *VeloLeft*

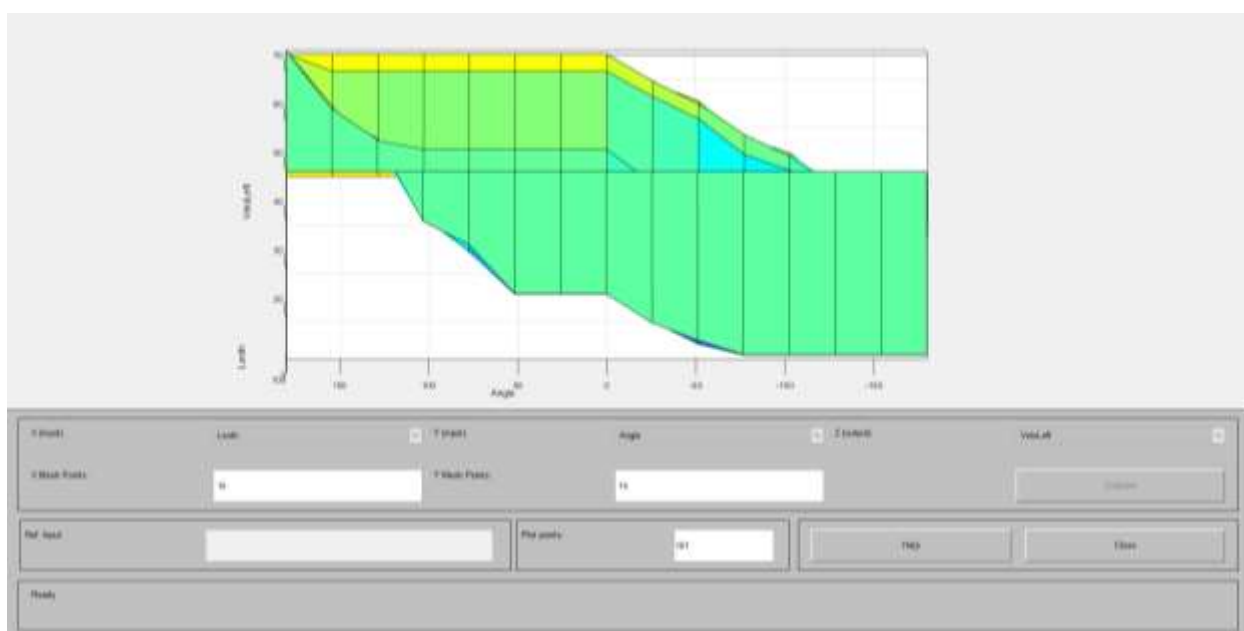


Рисунок 4.22 – Залежність між *Angle* і *VeloLeft*

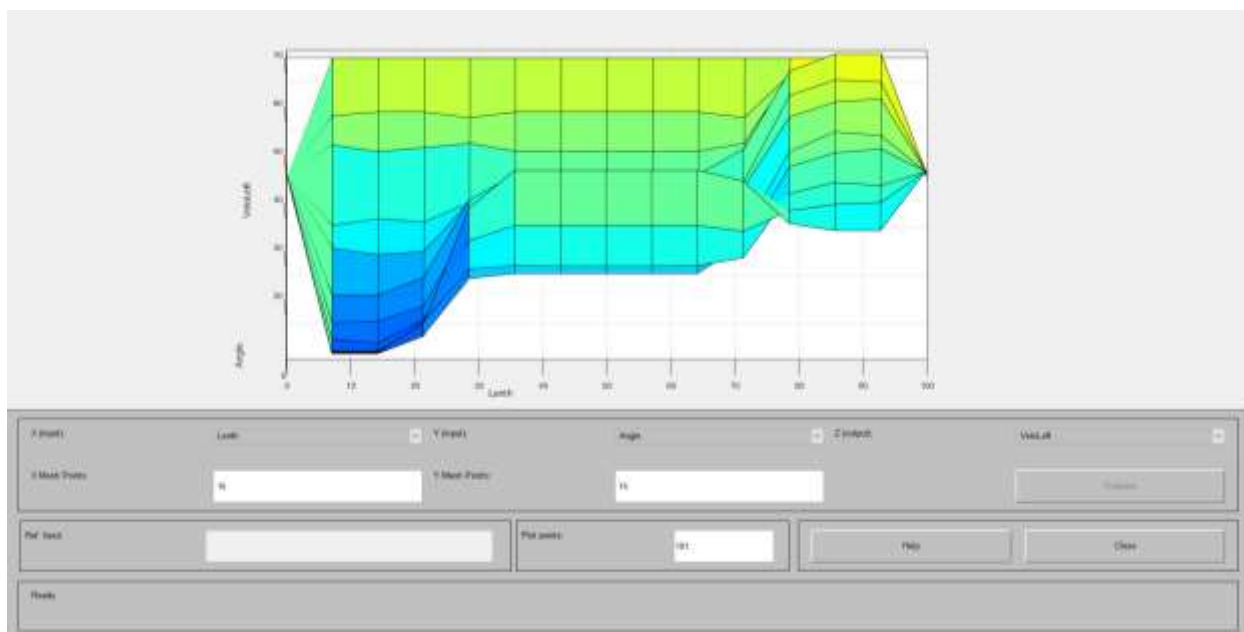


Рисунок 4.23 – Залежність між *Length* і *VeloLeft*

Отже, запропонована нечітка система керування, на вхід якої подається необхідна відстань переміщення та кут повороту робота, на основі введення лінгвістичних змінних та нечітких правил дозволяє отримувати швидкості обертання коліс робота.

#### 4.6 Висновки за розділом 4

В результаті виконання 4 розділу було виконано моделювання запропонованих моделей і методу в середовищі GNU Octave.

Розглянуто схеми адаптивного керування мобільним роботом.

Знайдено передавальні функції двигуна мобільного робота. Проведено моделювання адаптивної роботи одного та двох двигунів. Якщо відбувається зміна параметрів об'єкта керування при взаємодії з еталонною моделлю відбувається налаштування регулятора за допомогою контуру самоналаштування. Головною метою адаптації системи є мінімізація похибки на виході.

Також проведено моделювання розрахунку швидкостей та крутного моменту переміщення робототехнічної платформи.

Окремо розглянуто моделювання нечіткого адаптивного керування мобільним роботом, результати якого демонструють наявність залежності між вхідними (шлях переміщення та кут повороту) і вихідними параметрами (швидкості коліс робота), що може бути використане при нечіткому адаптивному керуванні мобільними транспортувальними роботами.

Основні результати цього розділу опубліковано в роботах [69], [74], [80].

## РОЗДІЛ 5

### РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АДАПТИВНОГО ВІЗУАЛЬНОГО КЕРУВАННЯ РОБОТАМИ

5.1 Розробка програмного забезпечення для розрахунку швидкостей робота з використанням GNU Octave

У розділі 4 у підрозділі 4.3 було проведено моделювання адаптивної системи керування роботом за допомогою GNU Octave для розрахунку швидкостей робототехнічної платформи, а саме функція знаходження швидкостей, що представлено на рис. 4.10. Оскільки система GNU Octave не має в своєму складі готових рішень, призначених для розрахунку параметрів швидкостей мобільних роботів, розроблено функцію у вигляді m-файлу.

На вході дана функція отримує швидкість, кутову швидкість, масив з точками переміщення у дискретному просторі та поточне розташування робота:

```
function [phiDotL, phiDotR] = wheelSpeed(v, omega, waypoints, pose)
```

Далі задаються константи для розрахунків, а саме порогові значення для зупинки та сповільнення робота, а також відстань між колесами і радіус колеса мобільної платформи:

```
stopThreshold = 0.1;
```

```
slowThreshold = 0.3;
```

```
trackWidth = 0.35;
```

```
wheelRadius = 0.03;
```

Також задаються команди для сповільнення і зупинки робота, якщо він досяг порогового значення:

```

distanceToEndpoint = norm(waypoints(end, :) - pose(1:2));
if (distanceToEndpoint < slowThreshold)
    v = distanceToEndpoint/slowThreshold*v;
    if distanceToEndpoint < stopThreshold
        v = 0;
    end
end
end

```

На основі цих даних проводяться розрахунки швидкостей для лівого та правого коліс:

$$\text{phiDotL} = (v - \text{trackWidth}/2 * \omega) / \text{wheelRadius};$$

$$\text{phiDotR} = (v + \text{trackWidth}/2 * \omega) / \text{wheelRadius}.$$

Розроблена функція може бути використана для розрахунків швидкостей переміщення мобільного робота і вбудована в GNU Octave.

## 5.2 Розробка програмного забезпечення для розпізнавання об'єктів робочого простору

Під час створення програмного забезпечення для розпізнавання об'єктів робочого простору використовувались мова програмування Python, бібліотеки NumPy для роботи з багатомірними масивами, а також бібліотека OpenCV для роботи з зображеннями [64], [67].

Розроблене програмне забезпечення ґрунтується на використанні методів відповідності шаблону Template Matching.

Файли з шаблонами об'єктів, що будуть розпізнаватись (роботи, верстати, люди, конвеєри, накопичувачі тощо), існують заздалегідь та

завантажуються у відповідну папку (наприклад, папка symbols). Приклад такого шаблону представлено на рис. 5.1.



Рисунок 5.1 – Приклад шаблону об'єкту, що розпізнається

На початку роботи програми завантажується зображення всього плану робочого простору:

```
image_filename = './plan.png'  
img_bgr = cv2.imread(image_filename)
```

Приклад такого плану зображено на рис. 5.2.



Рисунок 5.2 – Приклад плану

Далі отримується список імен усіх файлів, що містять шаблони об'єктів для розпізнавання:



```
onlyfiles = next(os.walk('./symbols'))[2]
```

Для кожного з об'єктів (файлів) у папці проводяться певні операції – зчитування файлів, перевірка відповідності та нормалізації:

```
template_filename = './symbols/' + f
tm_bgr = cv2.imread(template_filename)
matching = cv2.matchTemplate(img_bgr, tm_bgr, cv2.TM_CCOEFF_NORMED)
matching = cv2.normalize(matching, None, 0, 255, cv2.NORM_MINMAX,
dtype=cv2.CV_8U)
```

На наступному кроці виконується порогове перетворення, що буде використано для знаходження контурів об'єктів розпізнавання:

```
thresh = cv2.threshold(matching, np.max(matching) * 0.9, 255,
cv2.THRESH_BINARY)[1]
```

Контури об'єктів розпізнавання знаходяться за допомогою функції `findContours()`:

```
conts, hier = cv2.findContours(thresh, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
```

За допомогою функції `randint` задаються випадкові кольори для виокремлення певних класів об'єктів розпізнавання:

```
clr = np.random.randint(0, 200, (3,), dtype=np.uint8)
```

Знайдені об'єкти виділяються за допомогою функції `boundingRect()`, яка задає обмежувальні рамки для контурів об'єктів (рис. 5.3):

```

for cnt in conts:
    x, y, w, h = cv2.boundingRect(cnt)
    cv2.rectangle(img_bgr, (x + w // 2, y + h // 2), (x + w // 2 +
tm_bgr.shape[1], y + h // 2 + tm_bgr.shape[0]), clr.tolist(), 6).

```

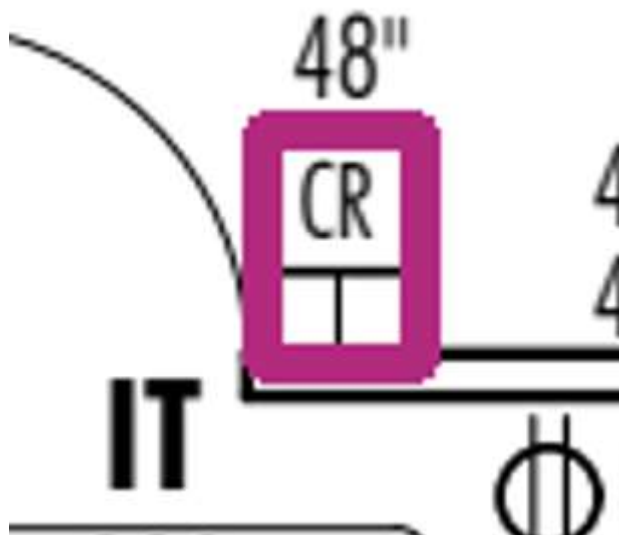


Рисунок 5.3 – Виділення об'єкту

Оскільки розмір вхідного зображення надто великий, його можна зменшити за допомогою функції `resize()`:

```
img_bgr = cv2.resize(img_bgr, (img_bgr.shape[1] // 2, img_bgr.shape[0] // 2)).
```

В кінці роботи програми результати записуються у файл:

```
cv2.imwrite("output.png", img_bgr ).
```

Результат виконання всіх операцій представлено на рис. 5.4.



Рисунок 5.4 – Результат розпізнавання об'єктів у заданому робочому просторі

Розроблене програмне забезпечення працює лише з статичними об'єктами з нульовим кутом нахилу та без зміни масштабу, але існують випадки, коли потрібний об'єкт повернутий відносно осей координат, тому доцільно використовувати метод SIFT, що виявляє і описує локальні ознаки зображення.

### 5.3 Розробка програмного забезпечення для розпізнавання об'єктів робочого простору методом SIFT

Аналогічно до 5.2 для розпізнавання об'єктів у робочому просторі за допомогою метода SIFT формується директорія з зображеннями images.

Для зберігання зображень з об'єктами розпізнавання [67-68], [73], а також для зберігання назв класів цих об'єктів створюються два масиви (зображення та класи об'єктів розпізнавання):

```
imagesList = []  
classNames = [] .
```

Розпізнавання об'єктів використовує метод SIFT [67], зокрема ORB-детектор ключових точок з набором із 1000 властивостей:

```
orb = cv2.ORB_create(nfeatures = 1000).
```

Потім проводиться зчитування усіх зображень з цільовими об'єктами та формується список їх імен:

```
for cls in listDir:  
    currentImage = cv2.imread(f'{path}/{cls}',0)  
    imagesList.append(currentImage)  
    classNames.append(os.path.splitext(cls)[0]) .
```

На наступному кроці знаходяться дескриптори ключових точок , що також записуються до масиву:

```
def findDescriptors(images):  
    descriptorList = []
```

```

for img in images:
    keypoints,descriptors = orb.detectAndCompute(img,None)
    descriptorList.append(descriptors)
return descriptorList.

```

Увесь розглянутий вище програмний код формує функцію `findImgID`, яка знаходить об'єкт розпізнавання на зображенні:

```
findImgID(img, descriptorList,thresh = 15)
```

Порівняння ключових точок об'єктів розпізнавання відбувається методом Brute Force, тобто перебором всіх можливих значень:

```
bf = cv2.BFMatcher()
```

При цьому, опрацьовуються не всі ознаки, а лише ті, для яких відсоток розпізнавання більший 75. Далі вони додаються до списку:

```

for des in descriptorList:
    matches = bf.knnMatch(des, descriptors2, k = 2)
    goodMatches = []
    for m,n in matches:
        if m.distance < 0.75 * n.distance:
            goodMatches.append([m])
    mList.append(len(goodMatches))

```

Функція `knnMatch` повертає співпадиння, в нашому випадку – 2 кращих.

На наступному кроці зображення з відеокамери порівнюється з зображеннями шаблонів бажаних об'єктів (зберігаються в створеній раніше директорії). У випадку співпадиння знайденого на зображенні об'єкта з шаблоном, виводиться ім'я класу шаблону:

```

id = findImgID(img, descriptorList)
if id != -1:
    cv2.putText(imgOrig, classNames[id], (50,50), cv2.FONT_HERSHEY_
_COMPLEX, 1,(0,0,255),2)

```

В результаті отримуємо розпізнаний об'єкт (рис. 5.5).



Рисунок 5.5 – Розпізнаний об'єкт з відео

Таким чином, розроблена програма використовує алгоритм SIFT для пошуку об'єктів, що відповідають бібліотеці шаблонів [73].

5.4 Розробка програмного забезпечення для візуального контролю кута повороту об'єкту на зображенні

Для того, щоб правильно розробити систему керування роботом, потрібно дізнатись не тільки його координати, але і кут його повороту.

Для більш коректного розпізнавання об'єкту, можна використовувати кольорові складові і кольорові простори.

В програмі використано синій колір для пошуку. Йому відповідають такі значення у просторі HSV:

```
hsv_blue_min = np.array((94, 80, 2), np.uint8)
hsv_blue_max = np.array((126, 255, 255), np.uint8)
```

Створюємо об'єкт відеозахвату:

```
cap = cv.VideoCapture(0)
```

Щоб знайти контур об'єкту, потрібно виконати деякі морфологічні операції:

```
img = cv.GaussianBlur(img, (1, 1), 0)
hsv = cv.cvtColor(img, cv.COLOR_BGR2HSV)
thresh = cv.inRange(hsv, hsv_blue_min, hsv_blue_max)
thresh = cv.dilate(thresh, np.ones((25, 25)))
thresh = cv.erode(thresh, np.ones((25, 25)))
```

По-перше – це розмиття по Гаусу. По-друге, перетворюємо кольори у простір HSV. По-третє, знаходимо потрібний колір на зображенні.

В четверте – робимо операції ділатації та ерозії, щоб відсіяти зайве та замкнути контури.

Отримаємо контури об'єкту:

```
contours, hierarchy = cv.findContours(thresh.copy(), cv.RETR_TREE,
cv.CHAIN_APPROX_SIMPLE)
```

Перебираємо у циклі всі контури і знаходимо мінімальні зони прямокутників, що будуть створювати границі контуру об'єкту. Потім їх малюємо:

```

for contour in contours:
    rect = cv.minAreaRect(contour)
    box = cv.boxPoints(rect)
    box = np.int0(box)
    cv.drawContours(img, [box], 0, (255, 0, 0), 2)

```

Виводимо текст з інформацією про кут нахилу об'єкту на зображенні:

```

cv.putText(img, "angle of object rotation = %d degrees" % float(rect[2]),
(0,20), cv.FONT_HERSHEY_PLAIN, 2, (0,0,0)

```

де `rect[2]` – інформація про кут нахилу.

Потім виводимо зображення з текстом про кут повороту. З результатом роботи програми можна ознайомитись на рис. 5.6.

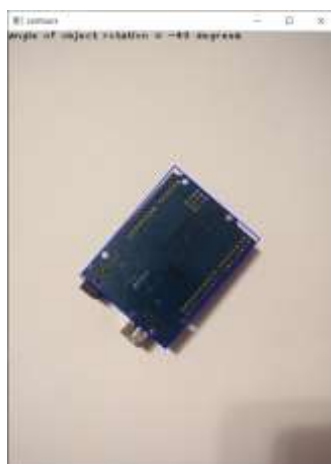


Рисунок 5.6 – Отримання кута повороту об'єкта

### 5.5 Розробка програмного забезпечення для адаптивного візуального керування мобільним роботом

Візуальне цілевказання підвищує швидкодію процесу керування об'єктами, його ефективність вища за ручне введення приблизно на 25% [106-107].



Розглянемо оцінку швидкодії введення інформації в системі керування мобільним роботом.

Для того, щоб ввести координати цільової точки переміщення робота, необхідно витратити час на наступні дії:

- переміщення до поля введення в інтерфейсі керування роботом – 1.1 сек;

- введення 8 чисел координат –  $8 \times 0.2 = 1.6$  сек ;

- натискання на клавішу вводу – 0.2 сек.

Таким чином, час введення координат з клавіатури складає 2.9 сек.

На аналогічне введення координат за допомогою миші потрібні наступні дії:

- фіксація розташування робота у початковій позиції натиском лівої клавіші миші – 0.2 сек;

- переміщення курсору миші в кінцеву точку переміщення – 1.1 сек;

- натискання правої кнопки миші для фіксації кінцевої точки маршруту робота – 0.2 сек.

Таким чином, час введення координат за допомогою візуального цілевказання складає 1.5 сек.

Отже, швидкість введення координат за допомогою візуального цілевказання майже в 2 рази вища, ніж за допомогою клавіатури.

В якості базової моделі робота було обрано Lego NXT Mindstorms, що представлено на рис. 5.7, система комп'ютерного зору – Web-камера Logitech, розташована над робочим простором.

Після запуску програми необхідно встановити з'єднання з роботом та проініціалізувати всі датчики та мотори робота:

```
Connection *connection = new Bluetooth();
```

```
Sensor *sensor1 = new Touch(IN_4, connection);
```

```
Sensor *light1 = new Light(IN_3, connection);
```

```
Sensor *dist1 = new Sonar(IN_4, connection);  
Motor *motorA = new Motor(OUT_A, connection);  
Motor *motorB = new Motor(OUT_B, connection);  
Motor *motorC = new Motor(OUT_C, connection);
```

Робочий простір мобільного робота представляється набором масивів даних, до якого входять поточна позиція  $sp$ , початкова та кінцева позиції  $sr$  та  $er$ , поточна та попередні положення  $curPoint$  та  $prevPoint$ , також ознаки зайнятості та відвідування комірок робочого простору  $bp$  та  $vp$ :

```
CPoint cp[33][25],sp,er,curPoint,prevPoint;  
bool bp[33][25], vp[33][25];
```



Рисунок 5.7 – Загальний вигляд робота NXT Mindstorms

За допомогою Web-камери система керування роботом отримує інформацію про поточний стан робочого простору – положення самого робота та всіх об'єктів, що знаходяться в ньому. Одночасно проводиться заповнення масивів даних, розмір яких пропорційний розподільній здатності камери (використано розмір 32x24, пропорційний до 640x480, або 800x600 пікселів).

Цільова точка переміщення мобільного робота вказується за допомогою миші на екрані, в результаті чого робот отримує цільові координати в локальному робочому просторі.

Отримання координат забезпечується обробником повідомлення миші:

```
switch(ev)
{
  case CV_EVENT_LBUTTONDOWN:
    if(bStart)
      {sp=FindCellCv(CPoint(x,y));
      freeInitialPos(sp);
      InitPoint.x = 1.47 - sp.x / 32.0 * 1.47;
      InitPoint.y = 1.1 - sp.y / 24.0 * 1.10;
      cvCurPoint.x=(float)x;cvCurPoint.y=(float)y;
      RobotCommand(1);
      bStart=false;
      if(sp==CPoint(0,0))AfxMessageBox("StartPoint isn't set");
      else { curPoint=sp;
            vp[sp.x][sp.y]=true;
            setpoint=true;          }          } break;
  case CV_EVENT_RBUTTONDOWN:
    sp.x=(cmd.currentPose[0])*32.0/1.47;
    sp.y=(cmd.currentPose[1])*24.0*1.1;
    wsprintf(msg,("(%d %d)",x,y);
    if(setpoint)
      { ep=FindCellCv(CPoint(x,y));
      if(ep==CPoint(0,0))AfxMessageBox("EndPoint isn't set");
      else { cell *ac2=new cell(curPoint,NULL);
            pl=pathCv(ac2);
            r_pl=reversCv(pl);
            r_pl=roadripper(r_pl);
```

```

while(pl){FillCellCv(pl->curpos.x,pl->curpos.y,2);pl=pl-
>next;}

if(r_init)AfxBeginThread(RobotRouter,NULL);
for(int i=1;i<33;i++)
for(int j=1;j<25;j++)    vp[i][j]=false;
sp=ep;                    }          }
else AfxMessageBox("StartPoint isn't set");
break; }

```

За допомогою натискання лівою кнопкою миші формується початкова точка маршруту, за допомогою натискання правої кнопки – кінцева та наступні точки маршруту мобільного робота.

Зокрема, в режимі натискання лівої клавіші миші, проводиться пошук комірки РП, в якій знаходиться робот та перетворення екранних координат в координати системи керування МР. В режимі фіксації цільової точки (однієї або декількох) за допомогою правої клавіші миші, проводиться пошук комірки РП, куди слід переміститись та відповідні перетворення систем координат. Пошук маршруту у просторі комірок здійснюється функцією pathCv(), обробка та оптимізація маршруту – за допомогою функції roadRipper(). Для виконання переміщень, відповідних маршруту, виконується потік RobotRouter().

Створюються змінні відстеження об'єктів. Нам потрібно відокремити роботу від інших об'єктів навколишнього середовища.

```

vector<PixelCluster> objects;
PixelCluster* _robot;
CvPoint* robotPt_UI;
float** neighbor_dist = NULL;
double RGBmean[3];
bool show_color_image = false;

```

Для завдання маршруту роботу використовується функція `UINT RobotLoop(LPVOID WinObjPtr)`, що є циклом виконання команд керування роботом.

В ній задається цільовий вектор переміщення, а також окремі команди для переміщення робота та просторових перетворень (ре-локалізація, зупинка, відкриття та закриття захвату, вказання поточної точки маршруту `waypoint`, вектору переміщення до цілі):

```

        case RELOCALIZE:
            robot.SetPose(cmd.currentPose[0],cmd.currentPose[1],cmd.currentPose[2]);
                memcpy(lastUpdatePose,robot.pose,3*sizeof(double));
                txn_type = STOP;
                break;
        case STOP:
            robot.halt();
            break;
        case ROTATE:
            targetDir.setRT(0,RADIANS_TO_DEGREES(cmd.dir.theta+cmd.currentPose[2])); break;
        case GRIPPER_OPEN:
            targetDir.setRT(0,45+RADIANS_TO_DEGREES(cmd.currentPose[3]));
                break;
        case GRIPPER_CLOSE:
            targetDir.setRT(0,-45+RADIANS_TO_DEGREES(cmd.currentPose[3]));
                break;
        case ROTATE_MOVE:
            targetDir.setXY(cmd.dir.x,cmd.dir.y);
            targetDir.rotate_z(RADIANS_TO_DEGREES(cmd.currentPose[2]));
                break;

```

```

case TARGET_VECTOR:
    targetDir.setXY(robot.pose[0],robot.pose[1]);
cmd.dir.rotate_z(RADIANS_TO_DEGREES(robot.pose[2]));
    targetDir.add(&(cmd.dir));
    break;
case WAYPOINT:
    targetDir.setXY(cmd.dir.x,cmd.dir.y);
    txn_type = TARGET_VECTOR;
    break;
case DISCONNECT:
    robot.halt();
    Sleep(100);
    exit(-1);
    break;          }

```

Також знаходяться вільні пікселі, де немає ніяких об'єктів:

```

bool isGoodPixel(IplImage* image, int x, int y, CvScalar& outP1){
    outP1 = cvGet2D(image,y,x);
    if(dist3D(outP1.val[0]-RGBmean[0],outP1.val[1]-
RGBmean[1],outP1.val[2]-RGBmean[2])>color_thresh)
        return true;
    return false;}

```

Вільні пікселі формуються у кластер пікселів і знаходяться визначаються у робочому просторі за допомогою функцій:

```

PixelCluster* FloodFillCluster(IplImage* image, int startX, int startY, float
max_pixel_dist)
    void clusterDetection(IplImage* frame)

```

Далі формується маршрут за вільними пікселями і робот переміщується у цільову точку. З роботою програмного забезпечення можна ознайомитись на рис. 5.8-5.9.

Спостереження робочого простору МР здійснюється в онлайн-режимі за допомогою функцій бібліотеки OpenCV, які забезпечують диференційні перетворення зображень (функція Лапласа), пошук меж зображень (функція Кенні), пошук контурів об'єктів (функція cvFindContours()), пошук моментів та на їх основі центрів мас об'єктів РП [87-88]. У разі знаходження об'єктів, що можуть бути перешкодами для переміщення МР, відбувається позначення комірок як зайнятих (функція setBusyCell()).

```

for( ; contour != 0; contour = contour->h_next )
    { cvMoments(contour, &moments, 0 );
      centre_of_mass.x=(int) (moments.m10/moments.m00);
      centre_of_mass.y=(int) (moments.m01/moments.m00);
      if(!centre_of_mass.x      &&      !centre_of_mass.y      &&
!cvCheckContourConvexity(contour))continue;
      double l=cvArcLength(contour,CV_WHOLE_SEQ,-1);
      double s=cvContourArea(contour);
      double itRobot=cvPointPolygonTest(contour,cvCurPoint,1);
      if(fabs(s)<400 && fabs(s)>50 && l>150 && l<1000 && itRobot<=
50)// && !it){ CvRect cont=cvBoundingRect(contour,0);
      cvRectangle(dst4,cvPoint(cont.x,cont.y),cvPoint(cont.x+cont.width,cont.y+c
ont.height),CV_RGB(0,0,255),CV_FILLED);
      setBusyCell(cont);
      cvSeqPush(sel_contour,&centre_of_mass);
      orientation      =      atan(2*moments.m11/(moments.m20
moments.m02));}

```



Рисунок 5.8 – Переміщення робота в робочому просторі на основі СКЗ

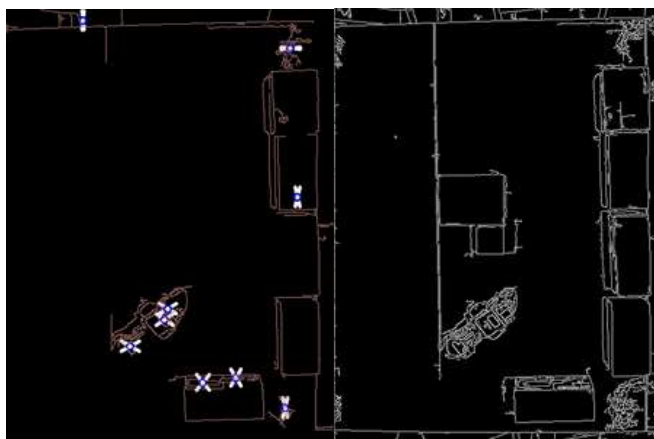


Рисунок 5.9 – Морфологічні операції з зображенням

У аналогічний спосіб розроблене ПЗ для адаптивного візуального керування мобільним роботом Festo Robotino (рис. 5.10). До особливостей його використання належить застосування команд віддаленого керування ОС Linux.



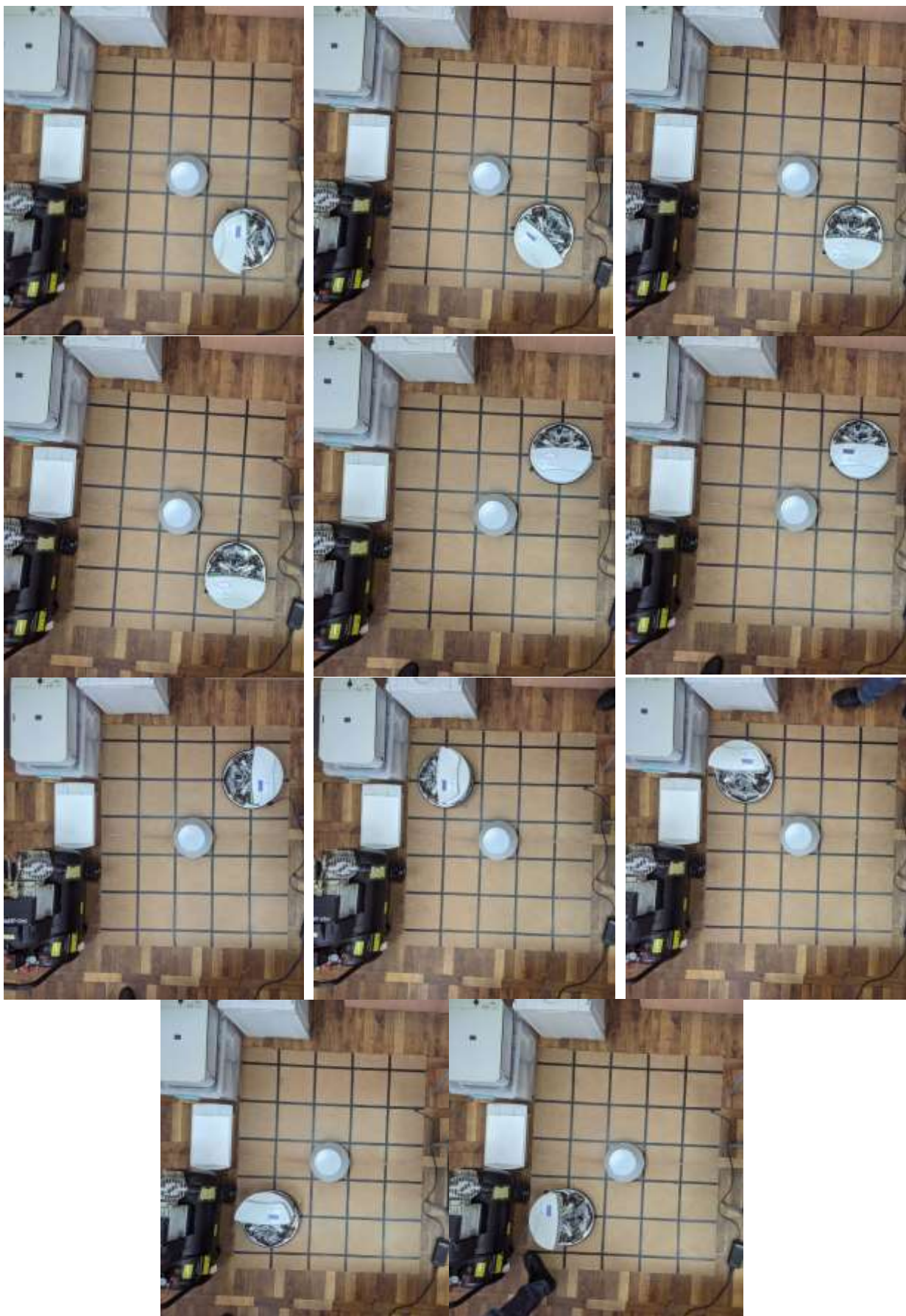


Рисунок 5.10 – Переміщення робота Robotino у робочому просторі ГІВС

Таким чином, розроблене інформаційне та програмне забезпечення практично реалізує запропоновані в розділах 2 та 3 та змодельовані в розділі 4 інформаційно-логічну та функціональну моделі, модель розпізнавання та ідентифікації, модель керування мобільним роботом у просторі ГІВС та метод адаптивного візуального керування [90-92].

#### 5.6 Висновки за розділом 5

В результаті виконання 5 розділу було розроблено програмне забезпечення для розрахунку швидкостей робота для моделі 4 розділу; розроблено програмне забезпечення для розпізнавання об'єктів робочого простору, а саме їх виділення певним кольором; розроблено програмне забезпечення для розпізнавання об'єктів робочого простору методом SIFT – для тих об'єктів, в яких є певні геометричні викривлення; розроблено програмне забезпечення для візуального контролю куту повороту об'єкту на зображенні для отримання його куту повороту при розрахунку переміщення за допомогою адаптивного візуального керування; розроблено програмне забезпечення для адаптивного керування мобільним роботом у робочому просторі ГІВС.

Основні результати цього розділу опубліковано в роботах [64], [67], [68], [73].

## ВИСНОВКИ

У дисертаційній роботі розв'язана нова актуальна науково-практична задача розробки моделей та методу адаптивного візуального керування, що забезпечує підвищення продуктивності та швидкодії робототехнічних засобів гнучких інтегрованих виробничих систем.

Внаслідок проведення теоретичних та практичних досліджень отримано такі результати:

1. Проведено дослідження сучасного стану гнучких інтегрованих роботизованих систем, засобів адаптивного та інтелектуального керування показало наявність протиріччя між сучасним рівнем розробки технології і методів комп'ютерного зору та рівнем їх впровадження у системах інтелектуального керування мобільних та маніпуляційних роботів. Обґрунтовано, що розробка моделей та методів адаптивного візуального керування мобільними роботизованими засобами, є своєчасною та актуальною задачею;

2. Вперше розроблено інформаційно-логічну модель робочого простору, що відображає об'єкти ГІВС, характерні для завдань інтелектуального керування роботами та зв'язки між об'єктами. Ця модель враховує дискретність та заповненість виробничого робочого простору; ґрунтується на інформації, отриманій від об'єктової системи комп'ютерного зору; визначає та враховує властивості об'єктів, що розміщуються в робочому просторі; враховує взаємодію, упорядкованість та сумісність об'єктів;

3. Вперше розроблено функціональну модель ГІВС, яка забезпечує опис взаємодії об'єктів на рівні виконання поставлених перед ГІВС виробничих, транспортувальних, сервісних, моніторингових, управлінських та інших завдань, та, ґрунтується на інформаційно-логічній моделі робочого простору враховує особливості технологічних операцій, містить детальний опис

властивостей об'єктів та їх взаємодії, включає можливість кількісної оцінки вартості та якості робіт;

3. Вдосконалено метод адаптивного візуального керування, що описує сукупність прийомів та операцій, які слід застосувати у інтелектуальних системах керування роботів для планування завдання переміщень на основі отримання зорової інформації від систем комп'ютерного (технічного) зору та орієнтований на використання в мобільних робототехнічних системах;

4. Отримала подальший розвиток модель керування мобільним роботом у просторі ГВС, що здійснює керування шасі двоколісного робота, та, на відміну від існуючих, функціонує на основі інформації від об'єктових/локальних систем комп'ютерного зору та виконує функції розпізнавання та ідентифікації об'єктів ГВС, забезпечує розрахунок координат та швидкостей роботизованої платформи та інших об'єктів ГВС, розрахунок лінійних та криволінійних траєкторій переміщень у дискретному робочому просторі, подання маршруту у вигляді послідовностей переміщень, забезпечує організацію керування транспортувальними роботами за їх візуальним положенням; містить нечітку складову з розгорнутим описом процедур формування лінгвістичних змінних, пов'язаних з динамікою руху мобільної роботизованої платформи та формалізує набір продукційних правил нечіткого регулятора;

5. Розроблено інформаційне та програмне забезпечення адаптивної візуальної системи керування мобільним транспортувальним роботом, яке реалізує запропоновані моделі і метод, практично забезпечує аналіз робочого простору за допомогою системи комп'ютерного/технічного зору, розпізнавання та ідентифікацію об'єктів робочого простору та робототехнічних засобів, отримання їх просторових координат, підтримку візуального цілевказання маршруту мобільного робота та функціонування інтелектуальної системи підтримки прийняття рішень;

6. Результати дисертаційної роботи впроваджені в освітньому процесі Харківського національного університету радіоелектроніки, а також у наукову

діяльність у ННЦ «Інститут судових експертиз ім. Засл. проф. М.С. Бокаріуса».

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Zaremba Marek. Modern Manufacturing: Information Control & Technology [Текст] / Marek Zaremba, Brian Prasad. Springer, 2011. – 415 p.
2. Ємельянов В.В. Теорія і практика еволюційного моделювання [Текст] / В.В Ємельянов і інш. – М.: Физматлит, 2003. – 432 с.
3. Ямпільський Л.С. Автоматизація проектування та керування в гнучкому виробництві [Текст] / Л.С. Ямпільський, З. Банашак - Київ: Техніка 1989, Варшава - Науково технічне видавництво, 1989 – 214с.
4. Технологічні основи гнучких виробничих систем. Підручник для машинобудівних спеціальностей вузів [Текст]. В.А.Медведев, В.П.Вороненко, В.Н.Брюханов і ін. / Під ред.Ю.М. Соломенцева. – Вища школа, 2000 р. – 256 с.
5. Solvang W. D., Developing automated and integrated flexible manufacturing system, [Текст] / W. D. Solvang, H. Yu and G. SziebigЮ – 2014 5th IEEE Conference on Cognitive Infocommunications (CogInfoCom), Vietri sul Mare, 2014, pp. 437-442.
6. Sethi, A. K Flexibility in Manufacturing: A Survey. [Текст] / Sethi, A. K., Sethi S. P. – International Journal of Flexible Manufacturing System, Services and Operations Management 2 (4). – 1990, pp 289–328.
7. Florescu Adriana Modeling and Simulation of a Flexible Manufacturing System –A Basic Component of Industry 4.0 [Текст] / Adriana Florescu and Sorin Adrian Baraba. – 2020, 20 p.
8. Kaushal Arvind. Flexible Manufacturing System A Modern Approach To Manufacturing Technology [Текст] / Arvind Kaushal, Ajay Vardhan, R. S. Rajput. – 2011. – pp 825-829.
9. Kaschel Hector, Importance of Flexibility in Manufacturing Systems. International Journal of Computers Communications & Control [Текст] / Hector Kaschel, Luis Sánchez, Bernal,.– Vol. I. , 2006, pp 53-60.

10. Brettel Malte, The relevance of manufacturing flexibility in the context of Industrie 4.0 [Текст] / Malte Brettel, Manuel Klein, Niklas Friederichsen. – Procedia CIRP, Volume 41, 2016, pp 105-110,

11. Salvador, Fabrizio Mix flexibility and volume flexibility in a build-to-order environment: Synergies and trade-offs. [Текст] / Fabrizio Salvador, Manus Rungtusanatham, Cipriano Forza, Alessio Trentin. – International Journal of Operations & Production Management. 27. pp. 1173-1191.

12. Goyal, Manu Volume Flexibility, Product Flexibility, or Both: The Role of Demand Correlation and Product Substitution [Текст] / Manu Goyal, Serguei Netessine. – Manufacturing & Service Operations Management., vol 13. – 2011, pp 180-193.

13. Chang, An-Yuan. On the measurement of routing flexibility: A multiple attribute approach. [Текст] – International Journal of Production Economics. Vol. 109, – 2007, pp 122-136.

14. Pauwels, Pieter. Strategic flexibility in export expansion: Growing through withdrawal [Текст] / Pieter Pauwels, Paul Matthyssens.– International Marketing Review. Volume 21, – 2004, pp. 496-510.

15. Monostori, Laszlo Cyber-physical systems in manufacturing. [Текст] / Laszlo Monostori, Botond Kádár, Thomas Bauernhansl, Shinsuke Kondoh, Soundar Kumara, Gunther Reinhart, Olaf Sauer, Günther Schuh, Wilfried Sihm, Ueda.- CIRP Annals Manufacturing Technology. vol. 65. – 2016, pp. 621–641.

16. Євсєєв В.В. Розробка інформаційної моделі взаємозв'язків «Сутність» баз даних для автоматизації створення адитивного кібердизайну кіберфізичних виробничих систем [Текст] / В.В. Євсєєв, А.І. Бронніков, Збірник наукових праць №3, 2020 с. 56-61

17. Berberich, J., Combining Prior Knowledge and Data for Robust Controller Design [Текст] / J.Berberich, , C. Scherer, F. Allgöwer. – ArXiv, – 2020, 21 pp.

18. AGVNETWORK [Електронний ресурс]. – Режим доступу: <https://www.agvnetwork.com/types-of-navigation-systems-automated-guided-vehicles> (дата звернення: 10.12.2020).
19. AGVNETWORK [Електронний ресурс]. – Режим доступу: <https://www.agvnetwork.com/what-is-a-laser-guided-vehicle-lgv>
20. YouTube [Електронний ресурс]. – Режим доступу: <https://www.youtube.com/watch?v=bjQP9Oe3M8I>
21. AGVNETWORK [Електронний ресурс]. – Режим доступу: <https://www.agvnetwork.com/natural-navigation-automated-guided-vehicles>
22. Acgov [Електронний ресурс]. – Режим доступу: [https://www.acgov.org/gsa/purchasing/bidContent\\_ftp/rfpDocs/Specification.pdf](https://www.acgov.org/gsa/purchasing/bidContent_ftp/rfpDocs/Specification.pdf)
23. ScottAutomation [Електронний ресурс]. – Режим доступу: <https://www.scottautomation.com/news/articles/types-of-agv-navigation-technology/>
24. AGVNETWORK [Електронний ресурс]. – Режим доступу: [https://www.acgov.org/gsa/purchasing/bidContent\\_ftp/rfpDocs/Specification.pdf](https://www.acgov.org/gsa/purchasing/bidContent_ftp/rfpDocs/Specification.pdf)
25. AGVNETWORK [Електронний ресурс]. – Режим доступу: <https://www.scottautomation.com/news/articles/types-of-agv-navigation-technology/>
26. Paluszek, Michael Optical Navigation System. [Текст] / Michael Paluszek, & Michael Littman, Joseph Mueller AIAA Infotech at Aerospace, - 2010 p., 8 p.
27. Sabikan, Sulaiman Vision-Based Automated Guided Vehicle for Navigation and Obstacle Avoidance [Текст] / Sulaiman Sabikan, , Najib Marizan, Salim Sy, Najib Sy Miskon Bin, Fahmi Muhammad. – IntechOpen, 2010, 12 p.
28. Halbaoui Khaled Introduction to Robust Control Techniques [Текст] / Khaled Halbaoui, Djamel Boukhetala, Fares Boudjema. – IntechOpen, 2011, 24 p.
29. Дорф Р., Бішоп Р. Сучасні системи керування. [Текст] Пер. з англ. Б. І. Копилова. - М .: Лабораторія базових знань, 2002. – 832 с.



30. Юревич, Євген Іванович. Керування роботами і робототехнічними системами [Електронний ресурс]: Учеб. посібник / Є.І. Юревич. - Електрон. текстові дан. (1 файл: 8,69 Мб). - Загл. з титул. екрану. - Електрон. версія печ. публікації 2001 г. - Вільний доступ з мережі Інтернет (читання, друк, копіювання). - Adobe Acrobat Reader 4.0. - <URL: <http://elib.spbstu.ru/dl/326.pdf>>.

31. Пупков К.А., Єгупов Н.Д. Методи сучасної теорії автоматичного керування. Том 5, Підручник в 5-й тт [Текст] .; 2-е изд., Перераб. і доп. МГТУ ім. Н. Е. Баумана, 2004. - 784с.

32. Mester, G. Adaptive Dynamic Hybrid Control of Industrial Robots with Flexible Joints In Croatian [Текст] / G.Mester; S.Pletl, G. Pajor. – Proceedings of the XX. Yugoslavien Congress in Theoretical and Applied Mechanics. Kragujevac, –1993, pp.94-97.

33. Krupitzer, Christian. A survey on engineering approaches for self-adaptive systems [Текст] / Christian Krupitzer, Felix Roth, Sebastian VanSyckel, Gregor Schiele, Christian Becker. – Pervasive and Mobile Computing. vol 17, – 2014, 18 p.

34. Rohr M. A classification scheme for self-adaptation research [Текст] / M. Rohr, S. Giesecke, W. Hasselbring, M. Hiel, W.-J. van den Heuvel, H. Weigand, in: Proc. SOAS, – 2006, p. 5.

35. Weyns D. On patterns for decentralized control in self-adaptive systems [Текст] / D. Weyns, B.R. Schmerl, V. Grassi, S. Malek, R. Mirandola, C. Prehofer, J. Wuttke, J. Andersson, H. Giese, K.M. Göschka. Software Engineering for Self-Adaptive Systems II, in: LNCS, vol. 7475, Springer, –2013, pp. 76–107.

36. McKinley P. Composing adaptive software [Текст] / P. McKinley, S. Sadjadi, E. Kasten, B.H.C. Cheng. – IEEE Comput. vol 37 (7).– 2004, pp 56–64.

37. Salehie M. Self-adaptive software: landscape & research challenges [Текст] / M. Salehie, L. Tahvildari. – ACM Trans. Auton. Adapt. Syst. 4 (2). –2009, 14 p.

38. Мартыненко, Ю.Г. Управление движением мобильных колесных роботов. [Текст] / Ю.Г. Мартыненко // *Фундаментальная и прикладная математика*. – 2005, т. 11, № 8, С. 29-80.
39. Stengel, Robert. Toward intelligent flight control [Текст] / Robert F. Stengel // *IEEE Transactions on systems, man and cibernetics*. – 1993. – Vol. 23, No. 6. – P. 1699-1717.
40. Nau, D. SHOP2: An HTN Planning System [Текст] / D. Nau, T. Au, O. Ilghami, U. Kuter, J.W. Murdock, D. Wu, F. Yaman // *Journal of Artificial Intelligence Research*. – 2003, № 20. – P. 379-404.
41. Myers, K. Towards a Framework for Continuous Planning and Execution [Электронный ресурс] / Karen L. Myers. – Режим доступа: <http://www.ai.sri.com/~cpef>.
42. Искусственные иммунные системы и их применение [Текст] / Под ред. Д. Дасгупты. – М.: Фихматлит, 2006. – 344 с.
43. Ефимов, Е.И. Решатели интеллектуальных задач [Текст] / Е.И. Ефимов. – М.: Наука, 1982. – 320 с.
44. Рассел, С. Искусственный интеллект: современный подход [Текст] / С. Рассел, П. Норвиг. – М.: Издательский дом «Вильямс», 2006. – 1408 с.
45. Application of intelligent control in industrial process control automation [Текст] / Wang Hong and Li YanYing *J. Phys.: Conf. Ser.* – 2020, 5 p.
46. Zou Xiantao Application analysis of intelligent control engineering in mechanical and electronic engineering [Текст] / Zou Xiantao, Lan Tong Ling . – *Information communication*, – 2019, pp. 278-279.
47. Chen Ying. Research on intelligent control technology in mechatronics system [Текст] / Ying Chen. – *Electronic production*, – 2019, pp. 76-78.
48. Ge Jinsheng. Application of intelligent control in mechatronics system [Текст] / Jinsheng Ge. – *Mining equipment*, – 2019, pp. 154-155.

49. Liu Yan Reform and practice of innovative intelligent control teaching [Текст] / Yan Liu, Yanjun Li, Shuwen Pan, Xuejie Wang, Zhuo Chen. – Modern city, 2019, pp. 58-60.
50. Korikov A. Artificial intelligence in robot control systems [Текст] / A. Korikov. – II International Conference "Cognitive Robotics, – 2018, 5 p.
51. Нильсон, Н. Искусственный интеллект [Текст] / Н. Нильсон. – М.:Мир, 1973. – 270 с.
52. Nilsson, N. Artificial Intelligence: A New Synthesis [Электронный ресурс] / N. Nilsson. – San Francisco: Morgan Kaufmann, 1998. Режим доступа: <http://ai.stanford.edu/~nilsson/errata.pdf>.
53. Strandberg, Morten. Robot Path Planning: An Object-Oriented Approach. Submitted to the School of Electrical Engineering [Текст] : Dissertation for the degree of Doctor of Philosophy / Morten Strandberg. – Royal Institute of Technology (KTH). – Stockholm, Sweden, 2004. – 254 p.
54. Aguirre, Marco. Metrics for sampling-based motion planning [Текст] : Dissertation for the degree of Doctor of Philosophy / Marco Antonio Morales Aguirre. – Office of Graduate Studies of Texas A&M University, 2007. – 131 p.
55. Stengel, Robert. Toward intelligent flight control [Текст] / Robert F. Stengel // IEEE Transactions on systems, man and cibernetics. – 1993. – Vol. 23, No. 6. – P. 1699-1717.
56. Aqe Musbah J. Decision Support Systems Classification in Industry [Текст] / Musbah J. Aqe, Omar A. Nakshabandi, Adeniyi Ayodeji. – Periodicals of Engineering and Natural Sciences. Vol. 7– 2019, 12 p.
57. Khakifirooz Marzieh. Decision Support for Smart Manufacturing [Текст] / Marzieh Khakifirooz Mahdi Fathi Panos M. Pardalos Daniel J. Power. – Mississippi State University.– 2019, 12 p.
58. Baumann Andrea. Robot Motion Planning in Time-varying Environments [Текст] / Andrea Baumann – Technischen UniversitaEt M€unchen. – 2001, 90 p.

59. FISpen W.H. Path Planning in Dynamic Environments [Текст] / W.H. FISpen. – Universiteit Utrecht opgezagvanderectormagnificus. – 2007, 200 p.
60. Burns Brendan. Exploiting Structure: A Guided Approach to Sampling-Based Robot Motion Planning [Текст] / Brendan Burns. – Graduate School of the University of Massachusetts. – 2007, 179 p.
61. Техническое зрение роботов [Текст] / Под. ред.. А. Пью. – М.:Машиностроение, 1987. – 320 с.
62. 3d-diy.ru [Электронный ресурс]. – Режим доступа: <https://3d-diy.ru/wiki/arduino-mechanics/motor-postoyannogo-toka-reduktorom-1-48/> (дата звернення: 07.06.2020).
63. Artem Bronnikov, Nevliudov Igor, Oleksandr Tsymbal. Flexible manufacturing tendencies and improvements with visual sensing / Eskisehir Technical University Journal of Science and Technology. Applied Sciences and Engineering, 2019. Vol. 20, ICONAT issue, P. 77-83.
64. Цымбал А.М., Бронников А.И., Литвинова А.Е., Чернышенко О.Е. Технологии программирования и робототехника / Восточно-Европейский журнал передовых технологий, Харьков, 2009, № 3/2 (39), С. 56-60.
65. Цимбал О.М., Бронников А.И. Адаптивність у прийнятті рішень роботів / Восточно-Европейский журнал передовых технологий, Харьков, 2011, № 4/4 (52), С. 40 - 43.
66. Цимбал О.М., Бронников А.И. Адаптивні процеси у завданнях робототехніки // Системи обробки інформації, Харків. – 2012. – Вип. 3 (101), том 1, С. 68 -73.
67. Цимбал О.М., Бронников А.И., Куценко О.И., Шеін Є.С. Концепція інтелектуальних виробничих агентів та особливості її реалізації // Восточно-Европейский журнал передовых технологий, Харьков, 2014, № 1/2 (67), С. 9 - 13.
68. Бронников А.И., Цимбал О.М., Фомовський Ф.М., Інтелектуальні засоби в системі керування виробничим агентом / Технологія приборостроєння. Харьков, 2014, № 2 , С. 26 - 30.

69. Бронніков А.І., Методи пошуку оптимального керування / Технологія приборостроєння. Харків, 2015, № 2 , С. 53 - 55.

70. Бронніков А.І., Адаптивне візуальне управління виробничим агентом / Технологія приборостроєння. Харків, 2016, № 3, С. 3 - 6.

71. Цимбал О.М., Бронніков А.І., Нгуєн Т.М.Л., Бекметова А.О. Інтелектуальні виробничі системи та перспективи їх реалізації / Технологія приборостроєння. Харків, 2016, № 1, С. 29 - 34.

72. Бронніков А.І., Цимбал О.М., Сіногін А.М. Мехатроніка у освітньому та загальнонауковому контексті / Системи керування, навігації та зв'язку. Полтава, 2017, № 2 (42), С. 30 - 32.

73. Цимбал О.М., Бронніков А.І. Прямий метод адаптації для керування роботою технічною платформою. / Технологія приборостроєння. Харків, 2016, № 3, С. 19 - 21.

74. Nevlyudov I., Tsymbal O., Bronnikov A. Intelligent means in the system of managing a manufacturing agent / Сучасний стан наукових досліджень та технологій в промисловості. 2018. № 1 (3). – С. 33-47.

75. Бронніков А.І. Сучасний стан та перспективи розвитку гнучких інтегрованих систем / Технологія приборостроєння. Харків, 2018, № 1, С. 54 - 57.

76. Tsymbal O., Bronnikov A. Decision-making information technology for flexible integrated manufacturing / Сучасний стан наукових досліджень та технологій в промисловості. 2019. № 2 (8). С. 105-112.

77. Nevlyudov I., Tsymbal O., Bronnikov A., Mordyk O. Internet of things for robotic projects / Сучасний стан наукових досліджень та технологій в промисловості. 2020. № 3 (13). – С. 58-64.

78. Цимбал О.М., Замірець М.В., Мордик О.О., Бронніков А.І., Корнілова Ю.Б. Інтелектуальні методи відеоспостереження: аналіз методів реалізації / Технологія приборостроєння. Харків, 2020, № 1, С. 35 - 40.

79. Igor Nevliudov, Vladyslav Yevsieiev, Murad Omarov, Artem Bronnikov and Viacheslav Liashenko. Method of Algorithms for Cyber-Physical Production Systems Functioning Synthesis //International Journal of Emerging Trends in Engineering Research (IJETER), Volume 8 No.10 (October 2020), pp: 7465 – 7473. ISSN 2347 - 3983 (DOI:10.30534/ijeter/2020/1278102020)

80. Цымбал А.М., Бронников А.И. Моделирование адаптивного принятия решений в ИСУ роботом // Вестник БГТУ им. В.Г. Шухова. – 2013. – №4, С. 173-176.

81. V. Yevsieiev, A. Bronnikov. Development of databases interconnection “essences” information model for cyber-physical production systems additive cyber design creation automation //Збірник наукових праць національного університету кораблебудування ім. адмірала Макарова, №3 (481), 2020р. pp.: 56- 62 ISSN 2311-3405 (Print), ISSN 2313-0415 (Online) (DOI: 10.15589/znp2020.3(481).7)

82. Oleksandr Tsymbal, Artem Bronnikov, Andriy Yerokhin. Adaptive Decision-making for Robotic tasks // Proceeding of IEEE 8th International Conference on Advanced Optoelectronics and Lasers, CAOL\*2019, DSMOLE\*2019: Scientific Workshop “Data Science in Modern Optoelectronics and Laser Engineering”, September 6-8, Sozopol, Bulgaria. – P. 594-597.

83. Oleksandr Tsymbal, Artem Bronnikov, Paolo Mercorelli. Decision-making models for Robotic Warehouse // Proceedings of 2020 International Symposium on Power Electronics, Electrical Drives, Automation and Motion, Virtual Meeting, June 24-26, P. 546-551.

84. Цымбал А.М. Программное моделирование системы управления мобильным роботом / А.М. Цымбал, А.И. Бронников // Автоматизация: проблемы, идеи, решения: Материалы международной научно-технической конференции 6 – 10 сентября 2010 г.: тезисы докл. – Севастополь, 2010. – С. 224 – 226.

85. Цимбал О.М., Бронніков А.І. Особливості адаптивного візуального керування мобільними роботами // Перша Всеукраїнська науково-практична конференція «Актуальні проблеми створення електронних засобів

промислових автоматизованих систем»: матеріали конференції. – Сєвєродонецьк: Технол. Ін-т Східноукр. Нац. Ун-ту ім. В. Даля, 2011. – С. 26 – 28.

86. Цимбал О.М. Адаптивні методи та їх реалізація в робототехніці / О.М. Цимбал, А.І. Бронніков // Системний аналіз та інформаційні технології: Матеріали міжнародної науково-технічної конференції 23–28 травня 2011 р.: тези доп.– Київ, 2011.– С. 335.

87. Tsymbal A.M. Decision-making in Robotics and adaptive tasks / A.M. Tsymbal, A. I. Bronnikov // Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2012), Kharkov, Sept. 14-17, 2012. – P. 417-420.

88. Бронніков А.І., Цимбал О.М. Інтелектуальне керування мобільним роботом в обмеженому робочому просторі // Матеріали 5-ї міжнародної науково-технічної конференції «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів керування». 21-22 квітня 2016 р. – Харків: ДП «ХНДІТП», 2016. – С. 33 – 34.

89. Цимбал О.М., Сінотін А.М., Бронніков А.І. Інтервальні засоби у моделі керування роботом // Матеріали першої міжнародної науково-практичної конференції «Computer and informational systems and technologies». 20-22 квітня 2017 р. – Харків: ХНУРЕ, 2017. – С. 43-44.

90. Цимбал О.М., Бронніков А.І. Інтелектуальне керування гнучкою роботизованою системою // Матеріали першої міжнародної науково-практичної конференції «Computer and informational systems and technologies». 20-22 квітня 2017 р. – Харків: ХНУРЕ, 2017. – С. 45-46.

91. О.М. Цимбал, А.І. Бронніков. Формування стратегій прийняття рішень в завданнях робототехніки // Матеріали V Міжнародної науково-технічної Інтернет-конференції Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами», 22 листопада 2018 С. 213-214.

92. Цимбал О.М., Бронніков А.І. Інтернет роботизованих речей: огляд концепції // Матеріали 3-ї міжнародної науково-практичної конференції

«Computer and informational systems and technologies». 23-24 квітня 2019 р. – Харків: ХНУРЕ, 2019. – С. 95-96.

93. Nevliudov Igor, Oleksandr Tsymbal, Artem Bronnikov. The Implementation of Manufacturing Agent's Concept for Flexible Production Systems // 1st International Scientific-Practical Conference - Modern Information, Measurement and Control Systems: Problems and Perspectives 2019, 1-2 July 2019, Baku. – P. 187.

94. Vladyslav Yevsieiev, Artem Bronnikov, Analysis of architectural models for representing the integration of cyber-physical production systems hierarchical levels // Manufacturing & Mechatronic Systems 2020: Proceedings of IVth International Conference, Kharkiv, October 22-23, 2020: Theses of Reports / [Ed. I.Sh. Nevlyudov (chief editor).] .- Kharkiv .: [electronic version], 2020., pp:17-19.

95. Vladyslav Yevsieiev, Artem Bronnikov Analysis of the cyber-physical production systems implementation impact to achieve the goals of lean production //The II th International scientific and practical conference «Development of scientific and practical approaches in the era of globalization» (September 28-30, 2020). Boston, USA 2020. p.241. pp. 221-226. ISBN - 978-1-64945-867-4. (DOI: 10.46299/ISG.2020.II.II) (Google Scholar, Crossref)

96. Yevsieiev V., Bronnikov A. Analysis of the CMMI model application for solving the tasks of CPPS control processes automation development // The IV th International scientific and practical conference «Actual Trends of Modern Scientific Research» (11-13 October 2020) MDPC Publishing, Munich, Germany, pp.128- 132, 386p. ISBN 978-3-954753-02-4

97. Yevsieiev V., Bronnikov A. Information systems development methodologies application analysis for cyber-physical production systems development // III International scientific-practical conference “Theory, science and practice” (October 05-08, 2020), Tokyo, Japan, pp. 399-401, p.450, 2020. ISBN - 978-1-64945-868-1 (DOI: 10.46299/ISG.2020.II.III) (Google Scholar, Crossref)

98. Yevsieiev V., Bronnikov A. Analysis of the multi-agent systems application to solve the problem of cyberphysical production systems development



// The IV<sup>th</sup> International scientific and practical conference «Integration of scientific bases into practice», (October 12-16, 2020), Stockholm, Sweden, pp. 459 – 462, 524. ISBN - 978-1-64945-864-3 (DOI:10.46299/ISG.2020.IV) (Google Scholar, Crossref)

99. Yevsieiev V., Bronnikov A. Structural model of a cyber-physical production system based on multi-agent systems analysis // Матеріали VII Міжнародної науково-технічної Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами», 26 листопада 2020. [Електронний ресурс] – К: НУХТ, 2020 – 314 с. 312-313.

100. Yevsieiev V., Bronnikov A. (2020) Comlexity development analysis of cyber-physical production systems for smart manufacturing // The X<sup>th</sup> International scientific and practical conference «Trends in the development of modern scientific thought» November 23-26, 2020 Vancouver, Canada. 789 p. pp. 699-703 (ISBN - 978-1-63649-921-5) (DOI - 10.46299/ISG.2020.II.X) (Google Scholar, Crossref).

101. Кулешов Сергей Викторович Применение скрытых марковских моделей для распознавая лиц / С. В. Кулешов // КурГУ, Курган, – 2003 г., 40 с.

102. Reddy R. Canny Scale Edge Detection / R.Reddy, C. Nagaraju, I Reddy. // IJETT. – 2016, 4 p.

103. Yousefi J. Image Binarization using Otsu Thresholding Algorithm / J. Yousefi / University of Guelph. - 2011

104. Lowe David Distinctive Image Features from Scale-Invariant Keypoints / David Lowe // International Journal of Computer Vision, - 2004, 28 pp.

105. Hartley R Multiple View Geometry in Computer Vision / R. Hartley and A. Zisserman// Cambridge University Press, - 2001., 12 pp.

106. Vertegaal, Roel. A Fitts Law comparison of eye tracking and manual input in the selection of visual targets / Roel Vertegaal. – 2008, pp. 241-248.

107. Gegenfurtner Karl. A comparison of localization judgments and pointing precision. *Journal of Vision* / Karl R. Gegenfurtner, Volker H. Franz: *Journal of Vision*, – 2007, 11 p.

**ДОДАТОК А**  
**Список публікацій**

## Публікації у фахових виданнях:

1. Artem Bronnikov, Nevliudov Igor, Oleksandr Tsymbal. Flexible manufacturing tendencies and improvements with visual sensing / Eskisehir Technical University Journal of Science and Technology. Applied Sciences and Engineering, 2019. Vol. 20, ICONAT issue, P. 77-83. (фахове періодичне видання за кордоном, Туреччина)
2. Цымбал А.М., Бронников А.И., Литвинова А.Е., Чернышенко О.Е. Технологии программирования и робототехника / Восточно-Европейский журнал передовых технологий, Харьков, 2009, № 3/2 (39), С. 56-60.
3. Цимбал О.М., Бронніков А.І. Адаптивність у прийнятті рішень роботів / Восточно-Европейский журнал передовых технологий, Харьков, 2011, № 4/4 (52), С. 40 - 43.
4. Цимбал О.М., Бронніков А.І. Адаптивні процеси у завданнях робототехніки // Системи обробки інформації, Харків. – 2012. – Вип. 3 (101), том 1, С. 68 -73.
5. Цимбал О.М., Бронніков А.І., Куценко О.І., Шеін Є.С. Концепція інтелектуальних виробничих агентів та особливості її реалізації // Восточно-Европейский журнал передовых технологий, Харьков, 2014, № 1/2 (67), С. 9 - 13.
6. Бронніков А.І., Цимбал О.М., Фомовський Ф.М., Інтелектуальні засоби в системі керування виробничим агентом / Технология приборостроения. Харьков, 2014, № 2, С. 26 - 30.
7. Бронніков А.І., Методи пошуку оптимального управління / Технология приборостроения. Харьков, 2015, № 2, С. 53 - 55.
8. Бронніков А.І., Адаптивное визуальное управление производственным агентом / Технология приборостроения. Харьков, 2016, № 3, С. 3 - 6.

9. Цимбал О.М., Бронніков А.І., Нгуєн Т.М.Л., Бекметова А.О. Інтелектуальні виробничі системи та перспективи їх реалізації / Технологія приборостроєння. Харків, 2016, № 1 , С. 29 - 34.

10. Бронніков А.І., Цимбал О.М., Сіногін А.М. Мехатроніка у освітньому та загальнонауковому контексті / Системи управління, навігації та зв'язку. Полтава, 2017, № 2 (42) , С. 30 - 32.

11. Цимбал О.М., Бронніков А.І. Прямий метод адаптації для керування роботою технічною платформою. / Технологія приборостроєння. Харків, 2016, № 3, С. 19 - 21.

12. Nevlyudov I., Tsymbal O., Bronnikov A. Intelligent means in the system of managing a manufacturing agent / Сучасний стан наукових досліджень та технологій в промисловості. 2018. № 1 (3). – С. 33-47.

13. Бронніков А.І. Сучасний стан та перспективи розвитку гнучких інтегрованих систем / Технологія приборостроєння. Харків, 2018, № 1, С. 54 – 57.

14. Tsymbal O., Bronnikov A. Decision-making information technology for flexible integrated manufacturing / Сучасний стан наукових досліджень та технологій в промисловості. 2019. № 2 (8). С. 105-112.

15. Nevliudov I., Tsymbal O., Bronnikov A., Mordyk O. Internet of things for robotic projects / Сучасний стан наукових досліджень та технологій в промисловості. 2020. № 3 (13). С. 58-64.

16. Цимбал О.М., Замірець М.В., Мордик О.О., Бронніков А.І., Корнілова Ю.Б. Інтелектуальні методи відеоспостереження: аналіз методів реалізації / Технологія приборостроєння. Харків, 2020, № 1, С. 35 - 40.

17. Igor Nevliudov, Vladyslav Yevsieiev, Murad Omarov, Artem Bronnikov and Viacheslav Liashenko. Method of Algorithms for Cyber-Physical Production Systems Functioning Synthesis //International Journal of Emerging Trends in Engineering Research (IJETER), Volume 8 No.10 (October 2020), pp: 7465 – 7473. ISSN 2347 - 3983 (DOI:10.30534/ijeter/2020/1278102020) (фахове видання за кордоном).

18. Цымбал А.М., Бронников А.И. Моделирование адаптивного принятия решений в ИСУ роботом // Вестник БГТУ им. В.Г. Шухова. – 2013. – №4, С. 173-176. (фахове видання за кордоном).

19. V. Yevsieiev, A. Bronnikov. Development of databases interconnection “essences” information model for cyber-physical production systems additive cyber design creation automation //Збірник наукових праць національного університету кораблебудування ім. адмірала Макарова, №3 (481), 2020р. pp.: 56- 62 ISSN 2311-3405 (Print), ISSN 2313-0415 (Online) (DOI: 10.15589/znp2020.3(481).7).

#### Публікації матеріалів конференцій:

1. Oleksandr Tsymbal, Artem Bronnikov, Andriy Yerokhin. Adaptive Decision-making for Robotic tasks // Proceeding of IEEE 8th International Conference on Advanced Optoelectronics and Lasers, CAOL\*2019, DSMOLE\*2019: Scientific Workshop “Data Science in Modern Optoelectronics and Laser Engineering”, September 6-8, Sozopol, Bulgaria. – P. 594-597. (Scopus)

2. Oleksandr Tsymbal, Artem Bronnikov, Paolo Mercorelli. Decision-making models for Robotic Warehouse // Proceedings of 2020 International Symposium on Power Electronics, Electrical Drives, Automation and Motion, Virtual Meeting, June 24-26, P. 546-551. (Scopus)

3. Цымбал А.М. Программное моделирование системы управления мобильным роботом / А.М. Цымбал, А.И. Бронников // Автоматизация: проблемы, идеи, решения: Материалы международной научно-технической конференции 6 – 10 сентября 2010 г.: тезисы докл. – Севастополь, 2010. – С. 224 – 226.

4. Цимбал О.М., Бронников А.І. Особливості адаптивного візуального керування мобільними роботами // Перша Всеукраїнська науково-практична конференція «Актуальні проблеми створення електронних засобів промислових автоматизованих систем»: матеріали конференції. –

Сєвєродонєцьк: Технол. Ін-т Схїдноукр. Нац. Ун-ту ім. В. Даля, 2011, С. 26 – 28.

5. Цимбал О.М. Адаптивні методи та їх реалізація в робототехніці / О.М. Цимбал, А.І. Бронніков // Системний аналіз та інформаційні технології: Матеріали міжнародної науково-технічної конференції 23–28 травня 2011 р.: тези доп.– Київ, 2011.– С. 335.

6. Tsymbal A.M. Decision-making in Robotics and adaptive tasks / A.M. Tsymbal, A. I. Bronnikov // Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2012), Kharkov, Sept. 14-17, 2012. – P. 417-420.

7. Бронніков А.І., Цимбал О.М. «Інтелектуальне керування мобільним роботом в обмеженому робочому просторі» // Матеріали 5-ї міжнародної науково-технічної конференції «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління». 21-22 квітня 2016 р. – Харків: ДП «ХНДІТП», 2016. – С. 33 – 34.

8. Цимбал О.М., Сінотін А.М., Бронніков А.І. Інтервальні засоби у моделі керування роботом // Матеріали першої міжнародної науково-практичної конференції «Computer and informational systems and technologies». 20-22 квітня 2017 р. – Харків: ХНУРЕ, 2017. – С. 43-44.

9. Цимбал О.М., Бронніков А.І. Інтелектуальне керування гнучкою роботизованою системою // Матеріали першої міжнародної науково-практичної конференції «Computer and informational systems and technologies». 20-22 квітня 2017 р. – Харків: ХНУРЕ, 2017. – С. 45-46.

10. О.М. Цимбал, А.І. Бронніков. Формування стратегій прийняття рішень в завданнях робототехніки // Матеріали V Міжнародної науково-технічної Інтернет-конференції Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами», 22 листопада 2018, С. 213-214.

11. Цимбал О.М., Бронніков А.І. Інтернет роботизованих речей: огляд концепції // Матеріали 3-ї міжнародної науково-практичної конференції

«Computer and informational systems and technologies». 23-24 квітня 2019 р. – Харків: ХНУРЕ, 2019. – С. 95-96.

12. Nevliudov Igor, Oleksandr Tsymbal, Artem Bronnikov. The Implementation of Manufacturing Agent's Concept for Flexible Production Systems // 1st International Scientific-Practical Conference - Modern Information, Measurement and Control Systems: Problems and Perspectives 2019, 1-2 July 2019, Baku. – P. 187.

13. Vladyslav Yevsieiev, Artem Bronnikov, Analysis of architectural models for representing the integration of cyber-physical production systems hierarchical levels // Manufacturing & Mechatronic Systems 2020: Proceedings of IVth International Conference, Kharkiv, October 22-23, 2020: Theses of Reports / [Ed. I.Sh. Nevlyudov (chief editor).] .- Kharkiv .: [electronic version], 2020, pp 17-19.

14. Vladyslav Yevsieiev, Artem Bronnikov Analysis of the cyber-physical production systems implementation impact to achieve the goals of lean production //The II th International scientific and practical conference «Development of scientific and practical approaches in the era of globalization» (September 28-30, 2020). Boston, USA 2020. p.241. pp. 221-226. ISBN - 978-1-64945-867-4. (DOI: 10.46299/ISG.2020.II.II) (Google Scholar, Crossref)

15. Yevsieiev V., Bronnikov A. Analysis of the CMMI model application for solving the tasks of CPPS control processes automation development // The IV th International scientific and practical conference «Actual Trends of Modern Scientific Research» (11-13 October 2020) MDPC Publishing, Munich, Germany, pp.128- 132, 386p. ISBN 978-3-954753-02-4

16. Yevsieiev V., Bronnikov A. Information systems development methodologies application analysis for cyber-physical production systems development // III International scientific-practical conference “Theory, science and practice” (October 05-08, 2020), Tokyo, Japan, pp. 399-401, p.450, 2020. ISBN - 978-1-64945-868-1 (DOI: 10.46299/ISG.2020.II.III) (Google Scholar, Crossref)



17. Yevsieiev V., Bronnikov A. Analysis of the multi-agent systems application to solve the problem of cyberphysical production systems development // The IV th International scientific and practical conference «Integration of scientific bases into practice», (October 12-16, 2020), Stockholm, Sweden, pp. 459 – 462, 524. ISBN - 978-1-64945-864-3 (DOI:10.46299/ISG.2020.IV) (Google Scholar, Crossref)

18. Yevsieiev V., Bronnikov A. Structural model of a cyber-physical production system based on multi-agent systems analysis // Матеріали VII Міжнародної науково-технічної Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами», 26 листопада 2020. [Електронний ресурс] – К: НУХТ, 2020 – 314 с. 312-313.

19. Yevsieiev V., Bronnikov A. (2020) Complexity development analysis of cyber-physical production systems systems for smart manufacturing // The X th International scientific and practical conference «Trends in the development of modern scientific thought» November 23-26, 2020 Vancouver, Canada. 789 p. pp. 699-703 (ISBN - 978-1-63649-921-5) (DOI - 10.46299/ISG.2020.II.X) (Google Scholar, Crossref).

## **ДОДАТОК Б**

**Код програми, що реалізує запропоновані моделі та метод адаптивного  
візуального керування мобільними роботами**

## // Файл App.h

```

#pragma once
#pragma once
#pragma comment(lib,"cv210.lib")
#pragma comment(lib,"cvaux210.lib")
#pragma comment(lib,"cxcore210.lib")
#pragma comment(lib,"highgui210.lib")

#include "afxwin.h"
#include <afxcmn.h>
#include <afxmt.h>
#include <math.h>
#include <highgui.h>
#include <cv.h>
#include "LegoNXT.h"

#define USE_ROBOT
#define DEFAULT_COMPORT 6

class CApp :
    public CWinApp
{
public:
    BOOL InitInstance(void);
};

class llist
{public:
    CPoint curpos;
    llist *next;
    llist(){curpos=CPoint(0,0);next=NULL;}
    llist(int cp,llist *p){curpos=cp;next=p;}
};

class cell
{public:
    CPoint curpos;
    cell *next;
    cell(){curpos=CPoint(0,0);next=NULL;}
    cell(CPoint cp,cell *n){curpos=cp;next=n;}
};

class CMainWin :
    public CFrameWnd
{
public:
    CMainWin(void);
    void OnCamera();
    void OnClose();
    DECLARE_MESSAGE_MAP()
};

struct motorCommand{
    VECT dir;
    int type;           // 1 - Continuous Vector, 2 - Target Vector
    bool isNew;
    double objectLoc[2]; // For investigating objects
    double currentPose[4];
    CMutex wait;
};

```

```

void on_mouse( int event, int x, int y, int flags, void* param );
CPoint FindCellCv(CPoint loc);
cell *pathCv(cell *c_path);
CPoint GetNextStepCv(CPoint cp, CPoint tp);
double distCv(CPoint cp, CPoint tp);
cell *reversCv(cell *in);
void FillCellCv(int i, int j, int c);

CPoint cellAdress(int x, int y);
UINT RobotLoop(LPVOID);
int ControlLoop(int i1);
void RobotStart();
void RobotCommand(int);
void rpl2waypoint();
cell *roadripper(cell *);
void setBusyCell(CvRect);
int min_el(double a[8]);

UINT RobotRouter(LPVOID WinObjPtr);
UINT GripperRouter(LPVOID WinObjPtr);
void pathfinder();
bool robotItself(CvPoint);
void freeInitialPos(CPoint);

typedef struct
{float x;
 float y;
 float fi;
 int txn_type;
}fPoint;

typedef struct areaObject
{fPoint centerPoint;
 bool state;
 float area;
 char name[255];
 fPoint currentPosition;
 fPoint finalPosition;
}pAreaObject;

enum{STOP, ROTATE, ROTATE_MOVE, TARGET_VECTOR, WAYPOINT, RELOCALIZE,
DISCONNECT,CURPOS,SET_GRIPPER, BACKUP};

```

```
// Файл App.cpp
```

```

#include <afxwin.h>
#include <math.h>

#include "App.h"
#include "LegoNxt.h"
#include "resource.h"

extern CvCapture *capture;
extern CvFont im_font;
extern IplImage *img;
extern LegoNXT robot;
bool fcv=false;
bool r_init=false;

int maxX,maxY;

```

```

int p1,q1,p2,q2;
CPoint cp[33][25],sp,ep,curPoint,prevPoint;
bool bp[33][25],d=true,not_subrout=true;
bool vp[33][25];
cell *p1=new cell;
cell *r_p1=new cell;
bool setpoint=false;
CvScalar selColor;
int harrist=0,harrist2=0;;
int redt=0,greent=0,bluet=0;

CMainWin::CMainWin(void)
{Create(NULL,"Adaptive decision-maker",WS_OVERLAPPEDWINDOW);
 int rc=MessageBox("Will you control NXT","NXT Control",MB_ICONQUESTION|MB_YESNO);
 if(rc==IDYES){ if(robot.init(DEFAULT_COMPORT))
                 {AfxMessageBox("Connected",MB_ICONINFORMATION);

                 AfxBeginThread(RobotLoop,this);
                 r_init=true;
                 }
                 else AfxMessageBox("No connection to NXT \n Emulation
mode",MB_ICONERROR);
                 }
 else AfxMessageBox("No connection to NXT \n Emulation mode",MB_ICONERROR);
 int resp=MessageBox("Do you want camera to use",NULL,MB_YESNO|MB_ICONQUESTION);
 if(resp==IDYES)
 {
     if(capture=cvCaptureFromCAM(1))
     { cvNamedWindow("mainWin",CV_WINDOW_AUTOSIZE);
       cvNamedWindow("Contour",CV_WINDOW_AUTOSIZE);
       cvCreateTrackbar("HarrisTrack","Contour",&harrist,100,NULL);
       cvSetMouseCallback("mainWin", on_mouse,this);//
       cvInitFont(&im_font, CV_FONT_HERSHEY_PLAIN, 0.7f,0.7f,0,1,CV_AA);
       OnCamera();
     }
 }
 else MessageBox("Camera reading Error");
 }

BOOL CApp::InitInstance(void)
{
    m_pMainWnd = new CMainWin;
    m_pMainWnd->ShowWindow(m_nCmdShow);
    m_pMainWnd->UpdateWindow();
    return TRUE;
}

CApp App;

BEGIN_MESSAGE_MAP(CMainWin,CFrameWnd)
END_MESSAGE_MAP()

int t_counter=0;

bool f=false;

void FillCellCv(int i, int j, int c)
{if(!c)selColor=CV_RGB(192,192,192);
 else {if(c==2)selColor=CV_RGB(0,255,0);
       else if(vp[i][j])selColor=CV_RGB(255,255,0);
       else selColor=CV_RGB(255,0,0);
     }
 bp[i][j]=true;
 fcv=true;p1=cp[i][j].x-12;q1=cp[i][j].y-12;p2=cp[i][j].x+13;q2=cp[i][j].y+13;
}

```

```

cvRectangle(img,cvPoint(p1,q1),cvPoint(p2,q2),CV_RGB(255,255,0),CV_FILLED);
}

```

```

cell *pathCv(cell *c_path)
{int i=curPoint.x; int j=curPoint.y;
  int i0=ep.x; int j0=ep.y;
if(curPoint==ep) return c_path;
else if(CPoint(i,j-1)==ep)
    {   curPoint=CPoint(i,j-1);
        cell *ac2=new cell(curPoint,c_path);
        return ac2;
    }
else if(CPoint(i,j+1)==ep)
    {   curPoint=CPoint(i,j+1);
        cell *ac2=new cell(curPoint,c_path);
        return ac2;
    }
else if(CPoint(i-1,j)==ep)
    {   curPoint=CPoint(i-1,j);
        cell *ac2=new cell(curPoint,c_path);
        return ac2;
    }
else if(CPoint(i+1,j)==ep)
    {   curPoint=CPoint(i+1,j);
        cell *ac2=new cell(curPoint,c_path);
        return ac2;
    }
else if(CPoint(i-1,j-1)==ep && !bp[i-1][j] && !bp[i][j-1])
    {   curPoint=CPoint(i-1,j-1);
        cell *ac2=new cell(curPoint,c_path);
        return ac2;
    }
else if(CPoint(i-1,j+1)==ep && !bp[i-1][j] && !bp[i][j+1])
    {   curPoint=CPoint(i-1,j+1);
        cell *ac2=new cell(curPoint,c_path);
        return ac2;
    }
else if(CPoint(i+1,j-1)==ep && !bp[i][j-1] && !bp[i+1][j])
    {   curPoint=CPoint(i+1,j-1);
        cell *ac2=new cell(curPoint,c_path);
        return ac2;
    }
else if(CPoint(i+1,j+1)==ep && !bp[i][j+1] && !bp[i+1][j])
    {   curPoint=CPoint(i+1,j+1);
        cell *ac2=new cell(curPoint,c_path);
        return ac2;
    }
}
//
else {CPoint cp2=curPoint;
      curPoint=GetNextStepCv(curPoint,ep);
      prevPoint=cp2;
      cell *ac2=new cell(curPoint,c_path);
      return pathCv(ac2);
}
}

cell *reversCv(cell *in)
{cell *out=new cell(in->curpos,NULL);
while(in->next)
{in=in->next;
  out=new cell(in->curpos,out);
}
return out;}

```

```

double distCv(CPoint cp, CPoint tp)
{
    return sqrt((double)((cp.x-tp.x)*(cp.x-tp.x)+(cp.y-tp.y)*(cp.y-tp.y)));
}

CPoint FindCellCv(CPoint loc)
{if(loc.x<5 || loc.y<5 || loc.x>(32*25+25) || loc.y>(24*25+25))return CPoint(0,0);
  else return CPoint(loc.x/25+1,loc.y/25+1);
}

int min_el(double a[8])
{
    int j=0;
    while(!a[j])j++;
    double m=a[j];
    int index=j;
    for(int i=j+1;i<8;i++)
        if(a[i]<m && a[i]){m=a[i];index=i;}
    return index;
}

CPoint GetNextStepCv(CPoint cp, CPoint tp)
{
    int i=cp.x,j=cp.y;
    int w=(int)(bp[i-1][j]+vp[i-1][j]*not_subrout);//+tmp[i-1][j]);
    int e=(int)(bp[i+1][j]+vp[i+1][j]*not_subrout);//+tmp[i+1][j]);
    int n=(int)(bp[i][j-1]+vp[i][j-1]*not_subrout);//+tmp[i][j-1]);
    int s=(int)(bp[i][j+1]+vp[i][j+1]*not_subrout);//+tmp[i][j+1]);
    int nw=(int)(bp[i-1][j-1]+vp[i-1][j-1]*not_subrout);//+tmp[i-1][j]);
    int sw=(int)(bp[i-1][j+1]+vp[i-1][j+1]*not_subrout);//+tmp[i-1][j]);
    int ne=(int)(bp[i+1][j-1]+vp[i+1][j-1]*not_subrout);//+tmp[i+1][j]);
    int se=(int)(bp[i+1][j+1]+vp[i+1][j+1]*not_subrout);//+tmp[i+1][j]);

    if(curPoint!=prevPoint)
    {
        if(prevPoint==CPoint(i-1,j))w++;
        if(prevPoint==CPoint(i+1,j))e++;
        if(prevPoint==CPoint(i,j-1))n++;
        if(prevPoint==CPoint(i,j+1))s++;
    }
    if(w && e && n && s)return cp;
    if(w && e && n)return CPoint(i,j+1); // s
    if(w && e && s)return CPoint(i,j-1); // n
    if(s && n && w)return CPoint(i+1,j); // e
    if(s && n && e)return CPoint(i-1,j); // w

    double ds=distCv(CPoint(i,j+1),ep); double dn=distCv(CPoint(i,j-1),ep);
    double de=distCv(CPoint(i+1,j),ep); double dw=distCv(CPoint(i-1,j),ep);
    // diagonal addon
    double dnw=distCv(CPoint(i-1,j-1),ep); double dsw=distCv(CPoint(i-1,j+1),ep);
    double dne=distCv(CPoint(i+1,j-1),ep); double dse=distCv(CPoint(i+1,j+1),ep);

    if(s && n || w && e)
    {dne=0.0;dse=0.0;dsw=0.0;dnw=0.0;}

    if(n && e)dne=0.0;
    if(s && e)dse=0.0;
    if(w && s)dsw=0.0;
    if(w && n)dnw=0.0;

    if(n)dne=dnw=dn=0.0;
    if(s)dse=dsw=ds=0.0;
    if(w)dnw=dsw=dw=0.0;
    if(e)dne=dse=de=0.0;
    if(nw)dnw=0;
    if(sw)dsw=0;
    if(ne)dne=0;
    if(se)dse=0;
}

```

```

double dlist[8]={de,dne,dn,dnw,dw,dsw,ds,dse};
int ra=min_el(dlist);
switch(ra)
{case 0: return CPoint(i+1,j);
 case 1: return CPoint(i+1,j-1);
 case 2: return CPoint(i,j-1);
 case 3: return CPoint(i-1,j-1);
 case 4: return CPoint(i-1,j);
 case 5: return CPoint(i-1,j+1);
 case 6: return CPoint(i,j+1);
 case 7: return CPoint(i+1,j+1);
}
return cp;
}

void pathfinder()
{int i=0,j=0;
while(r_pl)
{i=r_pl->curpos.x;j=r_pl->curpos.y;
 curPoint=CPoint(i,j);
 FillCellCv(r_pl->curpos.x,r_pl->curpos.y,1);
 vp[i][j]=true;
 not_subrout=true;
 if(r_pl->next)
     if(!bp[r_pl->next->curpos.x][r_pl->next->curpos.y])
         {r_pl=r_pl->next;not_subrout=true;
          prevPoint=curPoint;
         }
     else {cell *ac2=new cell(CPoint(i,j),NULL);
           not_subrout=false;
           pl=pathCv(ac2);
           r_pl=reversCv(pl);
          }
 else AfxMessageBox("Rout is gone");
}
}

```

// Файл OpenCV.cpp

```

#include "App.h"

CvCapture *capture;
IplImage *img,*img2,*dst=0,*dst1=0,*dst2=0,*dst3=0,*dst4=0,*dst5=0,*dst6=0;
int x;
char str[50];CString ss;
double cont_area, cont_prev, orientation, tangens, spatial, spatial_prev;
bool bStart=true;
CvFont im_font;

CvPoint origin;
fPoint InitPoint;
CvRect selection;
CvRect track_window;

CvMoments moments;
CvSeq *contour=0;
CvSeq *sel_contour=0;
CvPoint centre_of_mass, point1, point2, text_out;
extern CPoint sp,ep,cp;
extern bool setpoint;
extern CPoint curPoint;

```



```

extern CvPoint2D32f cvCurPoint;
extern bool bp[33][25],vp[33][25],fcv,r_init;
extern cell *pl,*r_pl;
extern int p1,q1,p2,q2;
extern CvScalar selColor;
extern LegoNXT robot;
extern motorCommand cmd;
extern int harrist,harrist2,redt,greent,bluet;
int select_object = 0;
int track_object = 0;
char msg[100];

void CMainWin::OnCamera()
{harrist=0;
 int lk=0;
 bool firstFrame=true;
 while(1)
 {CvMemStorage *storage=cvCreateMemStorage(0);
  img=cvQueryFrame(capture);
  cvFlip(img,NULL,-1);
  int contour_counter=0;
  {dst=cvCreateImage(cvGetSize(img),IPL_DEPTH_32F,3);
   dst1=cvCreateImage(cvGetSize(img),IPL_DEPTH_8U,3);
   dst2=cvCreateImage(cvGetSize(img),IPL_DEPTH_8U,1);
   dst3=cvCreateImage(cvGetSize(img),IPL_DEPTH_8U,1);
   dst4=cvCreateImage(cvGetSize(img),IPL_DEPTH_8U,3);
   dst5=cvCreateImage(cvGetSize(img),IPL_DEPTH_8U,3);
   dst6=cvCreateImage(cvGetSize(img),IPL_DEPTH_8U,1);
   firstFrame=false;
  }
  CvSeq *sel_contour = cvCreateSeq(
CV_SEQ_ELTYPE_POINT,sizeof(CvSeq),sizeof(CvPoint),storage);
  fcv=false;

  cvCvtColor(img,dst2,CV_RGB2GRAY);
  cvThreshold(dst2,dst3,1,255,CV_THRESH_BINARY_INV|CV_THRESH_OTSU); // OK Lab 3
  cvCanny(dst3,dst3,128,255);

  CvScalar color = CV_RGB( rand()&255, rand()&255, rand()&255 );
  int contrNum=cvFindContours(dst3, storage, &contour, sizeof(CvContour), CV_RETR_LIST,
CV_LINK_RUNS);
  if(!lk || lk==10 || lk==20)
  {
   for( ; contour != 0; contour = contour->h_next )
   {
    cvMoments(contour, &moments, 0 );
    centre_of_mass.x=(int) (moments.m10/moments.m00);
    centre_of_mass.y=(int) (moments.m01/moments.m00);
    if(!centre_of_mass.x && !centre_of_mass.y &&
!cvCheckContourConvexity(contour))continue;
    double l=cvArcLength(contour,CV_WHOLE_SEQ,-1);
    double s=cvContourArea(contour);
    double itRobot=cvPointPolygonTest(contour,cvCurPoint,1);
    if(fabs(s)<400 && fabs(s)>50 && l>150 && l<1000 && itRobot<-50)// &&
!it)
    {CvRect cont=cvBoundingRect(contour,0); // selection by rectangle

cvRectangle(dst4,cvPoint(cont.x,cont.y),cvPoint(cont.x+cont.width,cont.y+cont.height),CV_
RGB(0,0,255),CV_FILLED);
    setBusyCell(cont);
    cvSeqPush(sel_contour,&centre_of_mass);
    orientation = atan(2*moments.m11/(moments.m20 - moments.m02));
    point1.x = centre_of_mass.x+15*cos(orientation);
    point1.y = centre_of_mass.y+15*sin(orientation);

```

```

        point2.x = centre_of_mass.x-15*cos(orientation);
        point2.y = centre_of_mass.y-15*sin(orientation);
        cvLine (dst4, point1, point2,CV_RGB(255,255,255), 4, CV_AA);

        point1.x = centre_of_mass.x+15*cos(orientation);
        point1.y = centre_of_mass.y+15*sin(-orientation);
        point2.x = centre_of_mass.x-15*cos(orientation);
        point2.y = centre_of_mass.y-15*sin(-orientation);
        cvLine (dst4, point1, point2,CV_RGB(255,255,255), 4, CV_AA);

        cvCircle(dst4,centre_of_mass, 5,CV_RGB(0,0,255),2);

        cvDrawContours( dst4, sel_contour, color, color, -1,1, 8 );
        text_out.x=point1.x;
        text_out.y=point1.y;
        contour_counter++;
        ss.Format(_T("%f %f %d %d"),s,l,centre_of_mass.x,centre_of_mass.y);
        if(ss)cvPutText(dst4,ss, text_out, &im_font,CV_RGB(255,255,255));
    }
    cvShowImage("Contour",dst4);
}
}
cvShowImage("mainWin",img);

    cvReleaseMemStorage(&storage);

    cvReleaseImage(&dst);
cvReleaseImage(&dst1);cvReleaseImage(&dst2);cvReleaseImage(&dst3);
    cvReleaseImage(&dst4);cvReleaseImage(&dst5);cvReleaseImage( &dst6);

    x=cvWaitKey(10);
    if(x=='x'){RobotCommand(2);break;}
    lk++;
    if(lk==30){for(int i=0;i<26;i++)for(int j=0;j<20;j++)bp[i][j]=false;lk=0;}
}
cvReleaseCapture(&capture);
cvDestroyWindow("mainWin");
cvDestroyWindow("Laplace");
cvDestroyWindow("Canny");
cvDestroyWindow("Contour");
}

void CMainWin::OnClose()
{
    cvReleaseCapture(&capture);
    cvDestroyWindow("mainWin");
    cvDestroyWindow("Laplace");
    cvDestroyWindow("Canny");
    DestroyWindow();
}

void on_mouse( int ev, int x, int y, int flags, void* param )
{
    CMainWin *ptr=(CMainWin*)param;
    if( !img ) return;

    if( img->origin ) y = img->height - y;

    switch(ev)
    {
    case CV_EVENT_LBUTTONDOWN:
        if(bStart)
            {sp=FindCellCv(CPoint(x,y));
            freeInitialPos(sp);

```

```

to RobCtrl
    InitPoint.x = 1.47 - sp.x / 32.0 * 1.47; // transformation of WinProgram
    InitPoint.y = 1.1 - sp.y / 24.0 * 1.10;
    cvCurPoint.x=(float)x;cvCurPoint.y=(float)y;
    RobotCommand(1);
    bStart=false;
    if(sp==CPoint(0,0))AfxMessageBox("StartPoint isn't set");
    else {curPoint=sp;
        vp[sp.x][sp.y]=true;
        setpoint=true;
    }
}
break;
case CV_EVENT_RBUTTONDOWN:
    sp.x=(cmd.currentPose[0])*32.0/1.47;
    sp.y=(cmd.currentPose[1])*24.0*1.1;//,cmd.currentPose[2]
    wsprintf(msg,"%d %d",x,y);
    if(setpoint)
    {
        ep=FindCellCv(CPoint(x,y));
        if(ep==CPoint(0,0))AfxMessageBox("EndPoint isn't set");
        else {cell *ac2=new cell(curPoint,NULL);
            pl=pathCv(ac2);
            r_pl=reversCv(pl);
            r_pl=roadripper(r_pl);
            while(pl){FillCellCv(pl->curpos.x,pl->curpos.y,2);pl=pl->next;}
            if(r_init)AfxBeginThread(RobotRouter,NULL);
            for(int i=1;i<33;i++)
            for(int j=1;i<25;i++)    vp[i][j]=false;
            sp=ep;
        }
    }
    else AfxMessageBox("StartPoint isn't set");
break;
}
}

void setBusyCell(CvRect BusyArea)
{
    for(int i=BusyArea.x/25;i<=(BusyArea.x+BusyArea.width)/25+1;i++)
        for(int j=BusyArea.y/25;j<=(BusyArea.y+BusyArea.height)/25+1;j++)
            FillCellCv(i,j,3);
}

void freeInitialPos(CPoint cp)
{for(int i=cp.x-4;i<cp.x+4;i++)
    for(int j=cp.y-4;j<cp.y+4;j++)
        bp[i][j]=false;
}

bool robotItself(CvPoint centre_of_mass)
{bool res=false;
    for(int i=curPoint.x-1;i<curPoint.x+1;i++)
        for(int j=curPoint.y-1;j<curPoint.y+1;j++)
            if(CPoint(i,j)==FindCellCv(CPoint(centre_of_mass.x,centre_of_mass.y)))res=true;
    return res;
}

```

// Файл nxt\_ctrl.cpp

```

#include "App.h"
#include "vect.h"
#include "constants.h"

```

```

#include "n7.h"
#include "resource.h"

#include <cstdlib>
#include <iostream>
#include <string>
#include <conio.h>
#include "nxt.h"
#include <vector>

using namespace std;
double pi = 3.1415926535;
//enum {NONE, FORWARD, BACK, LEFT, RIGHT};
//int dir=NONE, dist=0, tmm=0, i1;
bool bStartCtrl=false;

CRobotPlan *evil=new CRobotPlan;
CRobotPlan *sub_evil;
CRobotPlan *plan=new CRobotPlan;
bool decision=false;
CListBox *decList;
CvPoint2D32f cvCurPoint;
extern cell *r_pl;
extern fPoint InitPoint;
fPoint path[200];
bool gripperState[200];
//int pathLength=8;
int pathLength=0;

LegoNXT robot;
char input[255];

// Global Variables... easier than passing args
motorCommand cmd;
#define ROBOT_MAX_SPEED      0.4           // meters/sec
#define MAX_WHEEL_SPEED     PI/3         // Radians/sec
#define WHEEL_RADIUS 0.095 // Radius of each wheel on the robot
#define MAX_ROT_POWER       0.5         // no idea what the units are
#define MIN_ROT_POWER       0.1         // no idea what the units are
// *****

#define MAX_VECTOR           1.00

#define MAX_TURN_SPEED      0.4 // roughly 60 deg/s in rad/s
#define MAX_DRIVE_SPEED     0.1 // m/s

#define MIN_DRIVE_SPEED     0.1 // the slowest value to send the robot
#define MIN_TURN_SPEED     0.1

#define CLOSE_ENOUGH_RAD   0.1 // what is an acceptable threshold?
#define CLOSE_ENOUGH_DIST 0.08 // how close to a waypoint does the robot need
to get?

#define NUM_PARTICLES      500
#define NUM_RANDOM         0

UINT RobotLoop(LPVOID WinObjPtr)
{
    //CRCDialog *ptr=(CRCDialog*)WinObjPtr;
    VECT targetDir,tmpDir;
    double lastUpdatePose[4];
    int txn_type = STOP; // i.e. continuous, or target vector...
    double transV = 0;
    double radV = 0;

```

```

targetDir.setXY(0,0);
VECT start; //,out;
int countLoops;
int count = 0;
double objectCoord[2];
CSingleLock sLock(&(cmd.wait));
double deltaP[3];
int FSA_STEP = 0;
memcpy(lastUpdatePose,robot.pose,4*sizeof(double));
while(1){ // loop forever
    robot.update();
    robot.GetDeltaPose(deltaP[0],deltaP[1],deltaP[2]);
    sLock.Lock();
    if(cmd.isNew){
        memcpy(lastUpdatePose,robot.pose,4*sizeof(double));
        cmd.isNew=false;
        txn_type = cmd.type;
        count = 0;
        FSA_STEP = 0;
        switch(txn_type){
            case RELOCALIZE:

robot.SetPose(cmd.currentPose[0],cmd.currentPose[1],cmd.currentPose[2]);
                memcpy(lastUpdatePose,robot.pose,3*sizeof(double));
                txn_type = STOP;
                break;
            case STOP:
                robot.halt();
                break;
            case ROTATE:

targetDir.setRT(0,RADIANS_TO_DEGREES(cmd.dir.theta+cmd.currentPose[2]));
                break;
            case SET_GRIPPER:
                targetDir.setRT(0,RADIANS_TO_DEGREES(cmd.dir.theta));
                break;
            case ROTATE_MOVE: // Rotate and Move in specified direction
                targetDir.setXY(cmd.dir.x,cmd.dir.y);
                targetDir.rotate_z(RADIANS_TO_DEGREES(cmd.currentPose[2]));
                break;
            case TARGET_VECTOR:

targetDir.setXY(robot.pose[0],robot.pose[1]);
                cmd.dir.rotate_z(RADIANS_TO_DEGREES(robot.pose[2]));
                targetDir.add(&(cmd.dir));
                break;
            case WAYPOINT: // Waypoint Mission (use a planner with map)
                // target coordinates are provided in global reference frame
                targetDir.setXY(cmd.dir.x,cmd.dir.y);
                txn_type = TARGET_VECTOR;
                break;
            case BACKUP:
                // target coordinates are provided in global reference frame
                targetDir.setXY(cmd.dir.x,cmd.dir.y);
                break;
            case DISCONNECT:
                robot.halt();
                Sleep(100);
                exit(-1);
                break;
        }
    }
    memcpy(cmd.currentPose,robot.pose,4*sizeof(double));

```

```

        tmpDir.setXY(cmd.currentPose[0]-lastUpdatePose[0],cmd.currentPose[1]-
lastUpdatePose[1]);
        if(tmpDir.r>0.5){ // need to update current pose on robot

robot.SetPose(cmd.currentPose[0],cmd.currentPose[1],cmd.currentPose[2]);
        memcpy(lastUpdatePose,robot.pose,4*sizeof(double));
        }
        sLock.Unlock();

        switch(txn_type){
        case ROTATE:
            tmpDir.setRT(0,RADIANS_TO_DEGREES(targetDir.theta-robot.pose[2]));
            if(fabs(tmpDir.theta)<CLOSE_ENOUGH_RAD)
                tmpDir.setRT(0,0);
            if(fabs(tmpDir.theta)>3)
                tmpDir.setRT(0,RADIANS_TO_DEGREES(3));
            break;
        case ROTATE_MOVE:
            tmpDir.setXY(targetDir.x,targetDir.y);
            tmpDir.rotate_z(RADIANS_TO_DEGREES(-robot.pose[2]));
            break;
        case TARGET_VECTOR: // rotate until close enough
            tmpDir.setXY(targetDir.x-robot.pose[0],targetDir.y-robot.pose[1]);
            tmpDir.rotate_z(RADIANS_TO_DEGREES(-robot.pose[2]));
            if(tmpDir.r<CLOSE_ENOUGH_DIST)
                tmpDir.setRT(0,0);
            break;
        case BACKUP:
            tmpDir.setXY(targetDir.x-robot.pose[0],targetDir.y-robot.pose[1]);
            tmpDir.rotate_z(RADIANS_TO_DEGREES(-robot.pose[2]));
            if(tmpDir.r<CLOSE_ENOUGH_DIST)
                tmpDir.setRT(0,0);
            break;
        case WAYPOINT:
            tmpDir.setXY(targetDir.x-robot.pose[0],targetDir.y-robot.pose[1]);
            tmpDir.rotate_z(RADIANS_TO_DEGREES(-robot.pose[2]));

            if(fabs(tmpDir.r)<CLOSE_ENOUGH_DIST){
                robot.halt();
            }
            break;
        case SET_GRIPPER:
            tmpDir.setRT(0,RADIANS_TO_DEGREES(targetDir.theta-robot.pose[3]));
            break;
        default:
            break; // don't do anything
        }
        if(txn_type!=STOP){

if((fabs(tmpDir.r)<CLOSE_ENOUGH_DIST)&&(fabs(tmpDir.theta)<CLOSE_ENOUGH_RAD)){
            robot.halt();
            txn_type = STOP;
        }else if(txn_type==SET_GRIPPER)
            robot.openGripper(tmpDir);
        else if(txn_type==BACKUP)
            robot.backUpRobot(tmpDir, MAX_VECTOR);
        else
            robot.driveRobot(tmpDir, MAX_VECTOR);
        }
        Sleep(10);
    }
    delete &robot;
}

```

```

void RobotStart()
{
    AfxBeginThread(RobotRouter, NULL);
}

void RobotCommand(int i1)
{
    double x,y,t;
    char a[5],b[5],c[5];
    CString d;
    cmd.dir.x = 0;
    cmd.dir.y = 0;
    cmd.isNew = false;
    cmd.objectLoc[0] = 0;
    cmd.objectLoc[1] = 1;
    cmd.type = 1;

    CSingleLock sLock(&(cmd.wait));
    switch(i1){
        case 4:
            sLock.Lock();
            cmd.type = STOP;
            cmd.isNew = true;
            sLock.Unlock();
            AfxMessageBox("Stopping");
            break;
        case 6:
            sLock.Lock();
            cmd.type = STOP;
            cmd.isNew = true;
            sLock.Unlock();
            x = (atof(a));
            y = (atof(b));
            d.Format(_T("Follow Vector [%f,%f]"),x,y);
            AfxMessageBox(d);
            sLock.Lock();
            cmd.type = ROTATE_MOVE;
            cmd.isNew = true;
            cmd.dir.setRT(x,y);
            sLock.Unlock();
            break;
        case 5:
            sLock.Lock();
            cmd.type = STOP;
            cmd.isNew = true;
            sLock.Unlock();

            x = (atof(a));
            y = (atof(b));
            d.Format(_T("Sending Target Vector Command [%f,%f]"),x,y);
            AfxMessageBox(d);
            sLock.Lock();
            cmd.type = TARGET_VECTOR;
            cmd.isNew = true;
            cmd.dir.setXY(x,y);
            sLock.Unlock();
            break;
        case 0:
            sLock.Lock();

            d.Format(_T("%.3f,%.3f,%.3f"),cmd.currentPose[0],cmd.currentPose[1],cmd.current
Pose[2]);
            AfxMessageBox(d);
            sLock.Unlock();

```

```

        break;
    case 1:
        sLock.Lock();
        cmd.type = STOP;
        cmd.isNew = true;
        cmd.dir.setXY(0,0);
        sLock.Unlock();
        x=InitPoint.x; y=InitPoint.y; t=0;
        d.Format(_T("Relocalize to [%f,%f,%f]"),x,y,t);
        AfxMessageBox(d);
        t = DEGREES_TO_RADIANs(atof(input));
        sLock.Lock();
        cmd.currentPose[0] = x;
        cmd.currentPose[1] = y;
        cmd.currentPose[2] = t;
        cmd.type = RELOCALIZE;
        cmd.isNew = true;
        sLock.Unlock();
        break;
    case 7:
        sLock.Lock();
        cmd.type = STOP;
        cmd.isNew = true;
        sLock.Unlock();
        x = (atof(a));
        y = (atof(b));
        sLock.Lock();
        cmd.type = WAYPOINT;
        cmd.isNew = true;
        cmd.dir.setXY(x,y);
        sLock.Unlock();
        break;
    case 3:
        sLock.Lock();
        cmd.type = STOP;
        cmd.isNew = true;
        sLock.Unlock();
        x = (atof(a));
        d.Format(_T("Sending Rotation Command %f"),x);
        AfxMessageBox(d);
        sLock.Lock();
        cmd.type = ROTATE;
        cmd.isNew = true;
        cmd.dir.setRT(0,x);
        sLock.Unlock();
        break;
    case 2:
        AfxMessageBox("Shutting Down");
        sLock.Lock();
        cmd.isNew = true;
        cmd.type = DISCONNECT;
        sLock.Unlock();
        break;
    case 8:
        sLock.Lock();
        cmd.type = STOP;
        cmd.isNew = true;
        cmd.type = SET_GRIPPER;
        sLock.Unlock();
        break;
    default:
        AfxMessageBox("Command not recognized");
}
}

```



```

UINT RobotRouter(LPVOID WinObjPtr)
{
    int state = 0;
    int waypoint = 0;
    CSingleLock sLock(&cmd.wait);
    double pose[4];
    rpl2waypoint();
    while(waypoint<pathLength)
    {
        sLock.Lock();
        memcpy(pose,cmd.currentPose,4*sizeof(double));
        sLock.Unlock();
        switch(state)
        {
        case 0: // transition
            if(fabs(path[waypoint].fi-RADIANS_TO_DEGREES(pose[3]))>5)
            {
                sLock.Lock();
                cmd.isNew = true;
                cmd.type = SET_GRIPPER;
                cmd.dir.setRT(0,(double)path[waypoint].fi);
                sLock.Unlock();
            }else{
                sLock.Lock();
                cmd.isNew = true;
                cmd.type = WAYPOINT;
                cvCurPoint.x=(1.0-cmd.currentPose[1])*1.57;
                cvCurPoint.y=cmd.currentPose[0]*1.27;
                cmd.dir.setXY(path[waypoint].x,path[waypoint].y);
                sLock.Unlock();
                state=1; // moving robot
            }
            break;
        case 1:
            if(dist2D(path[waypoint].x-pose[0],path[waypoint].y-
pose[1])<CLOSE_ENOUGH_DIST)//&&(fabs(path[waypoint].fi-RADIANS_TO_DEGREES(pose[3]))<6))
            {
                waypoint++;
                if(waypoint>=pathLength){
                    sLock.Lock();
                    cmd.isNew = true;
                    cmd.type = STOP;
                    sLock.Unlock();
                    RobotCommand(0);
                }
                state = 0;
            }
            break;
        }
    }
    return 1;
}

void rpl2waypoint()
{
    path[0].x=cmd.currentPose[0];
    path[0].y=cmd.currentPose[1];
    path[0].fi=cmd.currentPose[2];
    path[0].txn_type = WAYPOINT;
    int i=1;r_pl=r_pl->next;
    while(r_pl)

```

```

{path[i].y=1.47-(r_pl->curpos.x)/32.0*1.47; // transformation of WinProgram to RobCtrl
  path[i].x=1.1-(r_pl->curpos.y)/24.0*1.1;
  path[i].fi=0.0;
  path[i].txn_type = WAYPOINT;
  r_pl=r_pl->next;
  i++;
}
pathLength=i;
CString s; s.Format(_T("start point %f %f : end point %f
%f"),path[0].x,path[0].y,path[i-1].x,path[i-1].y);
  AfxMessageBox(s);
}

cell *roadripper(cell *r_pl)
{if(!r_pl->next || !r_pl->next->next)return r_pl;
  if((r_pl->curpos.y==r_pl->next->curpos.y && r_pl->curpos.y==r_pl->next->next->curpos.y)
  ||
    (r_pl->curpos.x==r_pl->next->curpos.x && r_pl->curpos.x==r_pl->next->next-
>curpos.x))
    {cell *r= new cell(r_pl->curpos,r_pl->next->next);
      delete r_pl->next; r=roadripper(r); return r;
    }
  if(abs(r_pl->curpos.x - r_pl->next->next->curpos.x)==2 && abs(r_pl->curpos.y - r_pl-
>next->next->curpos.y)==2)
    {cell *r= new cell(r_pl->curpos,r_pl->next->next);
      delete r_pl->next; r=roadripper(r); return r;
    }
  if((r_pl->curpos.x==r_pl->next->curpos.x && r_pl->next->curpos.y==r_pl->next->next-
>curpos.y)||
    (r_pl->curpos.y==r_pl->next->curpos.y && r_pl->next->curpos.x==r_pl->next->next-
>curpos.x))
    {cell *r= new cell(r_pl->curpos,r_pl->next->next);
      delete r_pl->next; r=roadripper(r); return r;
    }
  return new cell(r_pl->curpos,roadripper(r_pl->next));
}

```

**ДОДАТОК В****Акти про впровадження результатів дисертаційної роботи**



## А К Т

про впровадження результатів дисертаційної роботи  
Броннікова Артема Ігоровича  
у науково-дослідну діяльність

Комісія у складі голови – Сімакової-Єфремян Елли Борисівни, заступника директора інституту з наукової роботи, доктора юридичних наук, та членів комісії: Можяєва Михайла Олександровича, завідувача сектору комп'ютерно-технічних та телекомунікаційних досліджень лабораторії інженерно-технічних та військових досліджень, кандидата технічних наук; Мороховського Юрія Дмитровича, провідного судового експерта цієї ж лабораторії, склала цей акт про впровадження у науково-дослідну діяльність ННЦ «Інститут судових експертиз ім. Засл. проф. М.С. Бокаріуса» результатів дисертаційної роботи Броннікова А.І. за темою «Моделі та методи адаптивного візуального керування роботами».

Комісія підтверджує, що результати дисертаційної роботи Броннікова А.І., зокрема, розроблені інформаційно-логічна модель робочого простору, функціональна модель взаємодії об'єктів, метод адаптивного візуального керування, розроблене інформаційне та програмне забезпечення адаптивної системи візуального керування, можуть використовуватись у науковій діяльності відповідних підрозділів ННЦ «Інститут судових експертиз ім. Засл. проф. М.С. Бокаріуса» для розв'язання завдань комплексної автотехнічної експертизи, експертизи відеозапису дорожньо-транспортних

пригод (ДТП) , зокрема, завдань визначення режиму роботи світлофорного об'єкту на місці пригоди, встановлення фактичних сигналів світлофора на час виникнення ДТП, визначення марки і моделі транспортного засобу за відеозаписами та фотозображенням, визначення розташування і траєкторії транспортного засобу на проїзній частині під час його руху.

Акт складений для пред'явлення до спеціалізованої вченої ради із захисту дисертацій і не є підставою для фінансових розрахунків.

Голова комісії



Е.Б. Сімакова-Єфремян

Члени комісії:



М.О. Можаяев



Ю.Д. Мороховський

«Затверджую»

Ректор Харківського національного  
університету радіоелектроніки

проф. Семенець В.В.

« 6 »

2011 р.

**АКТ**

з впровадження в освітній процес  
результатів дисертаційної роботи  
Броннікова Артема Ігоровича

Комісія у складі голови: декана факультету автоматики і комп'ютеризованих технологій – д.т.н., проф. Филипенка Олександра Івановича, членів комісії: завідувача кафедри комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки (КІТАМ) д.т.н., проф. Невлюдова Ігоря Шакировича, заступника зав. кафедри КІТАМ з навчальної роботи к.т.н., ст. викладача Демської Наталії Павлівни, заступника зав. кафедри КІТАМ з наукової роботи к.т.н., професора Євсєєва Владислава В'ячеславовича, склала акт про впровадження результатів дисертаційної роботи Броннікова А.І. на тему «Моделі і методи адаптивного візуального керування роботами», подану на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.07 – автоматизація процесів керування, в освітній процес Харківського національного університету радіоелектроніки.

Склад впровадження:

1. Декларативна та функціональна моделі робочого простору гнучкої інтегрованої роботизованої системи, що забезпечують подання інформації про об'єкти робочого простору, зв'язки між ними, функціональні та логічні залежності;

2. Прикладне програмне забезпечення візуального моніторингу гнучкої інтегрованої системи, що реалізує метод адаптивного візуального керування.

Комісія встановила, що наукові результати, запропоновані в дисертаційній роботі Броннікову А.І. впроваджені в освітній процес на кафедрі КІТАМ при викладанні дисциплін: дисциплін: «Комп'ютерні системи управління рухомими об'єктами», «Автоматизовані системи керування рухомими об'єктами», «Системи адаптації роботів», для студентів спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології першого (бакалаврського) та другого (магістерського) рівня вищої освіти.

Голова комісії  
Члени комісії

  
О. І. Филипенко  
  
І. Ш. Невлюдов  
  
Н. П. Демська  
  
В. В. Євсєєв