

## ДОДАТОК А Апробація результатів роботи

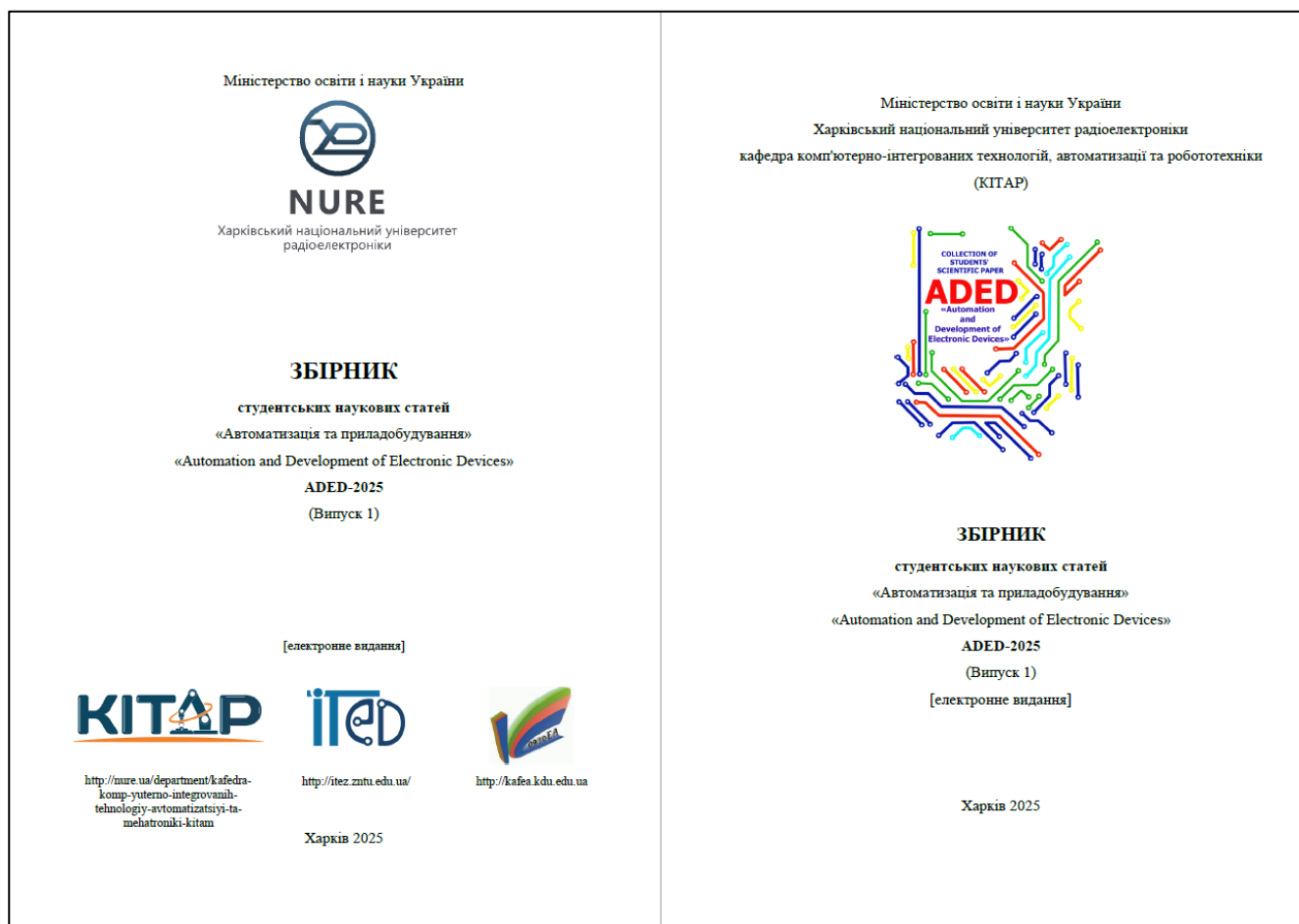


Рисунок А.1 – Титульний аркуш електронного видання

<p><b>Головний редактор</b> Невлюдов Ігор Шакірович, доктор технічних наук, професор, завідувач кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківського національного університету радіоелектроніки.</p> <p><b>Редакційна колегія:</b> Філіпенко Олександр Іванович, доктор технічних наук, професор, декан факультету Автоматики та комп'ютеризованих технологій, Харківського національного університету радіоелектроніки. Цимбал Олександр Михайлович, доктор технічних наук, професор кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківського національного університету радіоелектроніки. Андрусевич Анатолій Олександрович, доктор технічних наук, професор, начальник Криворізького коледжу національного авіаційного університету Косенко Віктор Васильович, доктор технічних наук, професор, зам. директора Державного підприємства «Південний державний проектно- конструкторський та науково-дослідний інститут авіаційної промисловості». Замірець Микола Васильович, доктор технічних наук, професор, директор Державного підприємства Науково-дослідного технологічного інституту приладобудування Свист Володимир Митрофанович, доктор технічних наук, професор, радник директора Державне науково-виробниче підприємство «Об'єднання Комунар». Фомовська Олена Владиславівна, кандидат технічних наук, доцент завідувач кафедри «Електронних апаратів» Кременчуцького національного університету імені Михайла Остроградського. Кухаренко Дмитро Володимирович, кандидат технічних наук, доцент кафедри «Електронних апаратів» Кременчуцького національного університету імені Михайла Остроградського Демська Наталія Павлівна, кандидат технічних наук, доцент кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківського національного університету радіоелектроніки. Фурманова Наталія Іванівна, кандидат технічних наук, доцент, декан факультета Радіоелектроніки і телекомунікацій, Національного університету «Запорізька політехніка».</p> <p><b>Відповідальний редактор:</b> Євсєєв Владислав Вячеславович, доктор технічних наук, професор кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківського національного університету радіоелектроніки.</p>	<p>Автоматизація та Приладобудування («Automation and Development of Electronic Devices» ADED-2025) [Електронний ресурс] : збірник студентських наукових статей / Харківський національний університет радіоелектроніки ; [редкол.: І.Ш. Невлюдов та ін.] . – Харків : ХНУРЕ, 2025. – Вип. 1. – 262с.</p> <p>Collection of Students' Scientific Paper «Automation and Development Of Electronic Devices» ADED-2025 Part 1 (Key infrastructure 2025) - Kharkiv/ The Editorial.: Nevludov I.Sh. (head), that all. Kharkiv: Kind of Kharkiv National University of Radio Electronics [electronic edition], 2025. – 262p with.</p> <p>Рекомендовано рішенням Науково-технічної ради Харківського національного університету радіоелектроніки протокол №6 від 29.11.2018</p> <p>Рекомендовано рішенням Вченої ради факультету Автоматики і комп'ютеризованих технологій Харківського національного університету радіоелектроніки протокол № 5 від 22 травня 2025</p> <p>Збірник містить наукові статті здобувачів першого (бакалаврського), другого (магістерського) рівнів вищої освіти кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки (КІТАР) Харківського національного університету радіоелектроніки, кафедри Інформаційних технологій електронних засобів (ІТЕЗ) Запорізького національного технічного університету та кафедри Електронних апаратів (ЕА) Кременчуцького національного університету ім. М. Остроградського які навчаються за спеціальностями: 151 Автоматизація та комп'ютерно-інтегровані технології, 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка; 172 Телекомунікації та радіотехніка, 171 Електроніка та 163 Біомедична інженерія. Статті надані в авторській редакції.</p> <p style="text-align: right;">©ХНУРЕ, 2025 рік</p> <p style="text-align: center;">«AUTOMATION AND DEVELOPMENT OF ELECTRONIC DEVICES»</p> <p style="text-align: right;">4</p>
---	--

Рисунок А.2 – Інформація про електронне видання

ЗМІСТ			
<i>Андреев А.С.</i>	Розроблення програмного забезпечення для аналізу вхідної інформації робітника приладобудівного виробництва для видачі завдань на виконання .....	<i>Хусмет Саркар Осли Сабуллаев</i>	Інноваційне оснащення складських приміщень .....
<i>Тарасов А.А.</i>	Розроблення 3D моделі пневматичного регулятора тиску .....	<i>Горбачов К.Ю.</i>	Інтеграція штучного інтелекту в медіаіндустрію .....
<i>Обришко Є.В.</i>	Аналіз методів оптимізації роботи системи дистанційного навчання при навантаженні Кузьменко О.С.	<i>Драннік А.С.</i>	Застосування генеративних моделей зі для обробки медіа в реальному часі .....
	Аналіз методів і технологій захвату руків .....	<i>Ткаченко І.А.</i>	Автоматизації логістичних процесів виробничого підприємства .....
<i>Ачкан М.С.</i>	Роль Big Data у розумних містах: автоматизовані рішення .....	<i>Фесенко А.О.</i>	GoLang як сучасна мова програмування для Backend частини сайтів .....
	Інтеграція хмарних технологій в сучасні SCADA системи: перспективи та виклики ....	<i>Хатлін І.О.</i>	Розвиток безпілотних технологій через симуляційне навчання: тенденції та перспективи .....
<i>Барисов А.М.</i>	Функціонування автоматизованої системи пожежної сигналізації спостереження .....	<i>Хатлін І.О.</i>	Інтеграція віртуальної та доповненої у навчальні симуляції для операторів дронів .....
<i>Дарасан В.В.</i>	Веб-інтерфейси для моніторингу та управління роботизованими системами в реальному часі .....	<i>Б.О. Цанга</i>	Дослідження методів автоматичної екстракції виробів 3D-принтерів .....
<i>Softia Driha</i>	Automated Waste Classification for Efficient Recycling Using Machine Learning .....	<i>Шамалюк Р.Р.</i>	Системи прогнозування відмов обладнання на основі аналізу експлуатаційних даних ..
<i>Іванов М.О.</i>	Актуальність віртуалізації та контейнеризації в сучасному ІТ .....	<i>Наговицін К.О.</i>	Modern Vehicle Access Control Technologies at Industrial Facilities .....
<i>А.Катренко</i>	Design of Mine-Detecting Robot Using Yolov8 Object Detection Model .....	<i>Межанов А.А.</i>	Шляхи досягнення цілей сталого розвитку у сфері гуманітарного розмінування із застосуванням робототехнічних комплексів .....
<i>Котіченко О.І.</i>	Analysis of Computer Vision Systems for Object Recognition .....	<i>Дерево Янко Д.І.</i>	Розроблення інтелектуальної системи автоматизації дозування хлорагенту для підготовки питної води .....
<i>Іванов М.О.</i>	Розроблення автоматичної системи розумного будинку на Node-Red .....	<i>Єрофеев С.О.</i>	Автоматизовані диспенсери ліків: сучасний стан та напрями розвитку .....
<i>Литочкін Н.О.</i>	Хмарні середовища для колаборативного проектування в роботехніці: можливості та обмеження .....	<i>Редькін К.С.</i>	Розроблення методу оцінки якості теплопостачання в центральному тепловому пункті .....
<i>Львенок Г.О.</i>	Аналіз алгоритмів планування шляху мобільного робота .....	<i>Берест Б.Р.</i>	Дослідження використання гнучких виробничих систем та їх класифікація .....
<i>Заяць Д.Є.</i>	Штучний інтелект та інтелектуальні помічники .....	<i>Диктенко А.І.</i>	Аналіз сучасних систем моніторингу та аналізу даних на виробництві .....
<i>Котенко V.A.</i>	Advantages and Disadvantages of Surface Robots in Various Fields of Application .....	<i>Демченко А.В.</i>	Аналіз систем керування мобільними роботами класу Mini Sumo для Battle of Robots .....
<i>Маслов А.Д.</i>	Інтелектуальна система керування вуличним освітленням з використанням IoT-технологій та алгоритмів машинного навчання .....	<i>Раєнко Т.В.</i>	Аналіз методів підключення пультів керування FPV-дронами до ПК для симуляції польоту .....
<i>Надьожкіна І.М.</i>	Дослідження систем автоматизації аналізу ґрунту на базі технологій інтернету речей ...	<i>Шляхов П.В.</i>	Методи децентралізованого керування групою колаборативних роботів-маніпуляторів у єдиній робочій зоні з людиною .....
<i>D. Nienova</i>	Inverse Kinematics In Robotics: Case Of Pick-And-Place Manipulators .....		
	5		6
«AUTOMATION AND DEVELOPMENT OF ELECTRONIC DEVICES» AED-2025 Part 1.		«AUTOMATION AND DEVELOPMENT OF ELECTRONIC DEVICES» AED-2025 Part 1.	

Рисунок А.3 – Зміст конференції

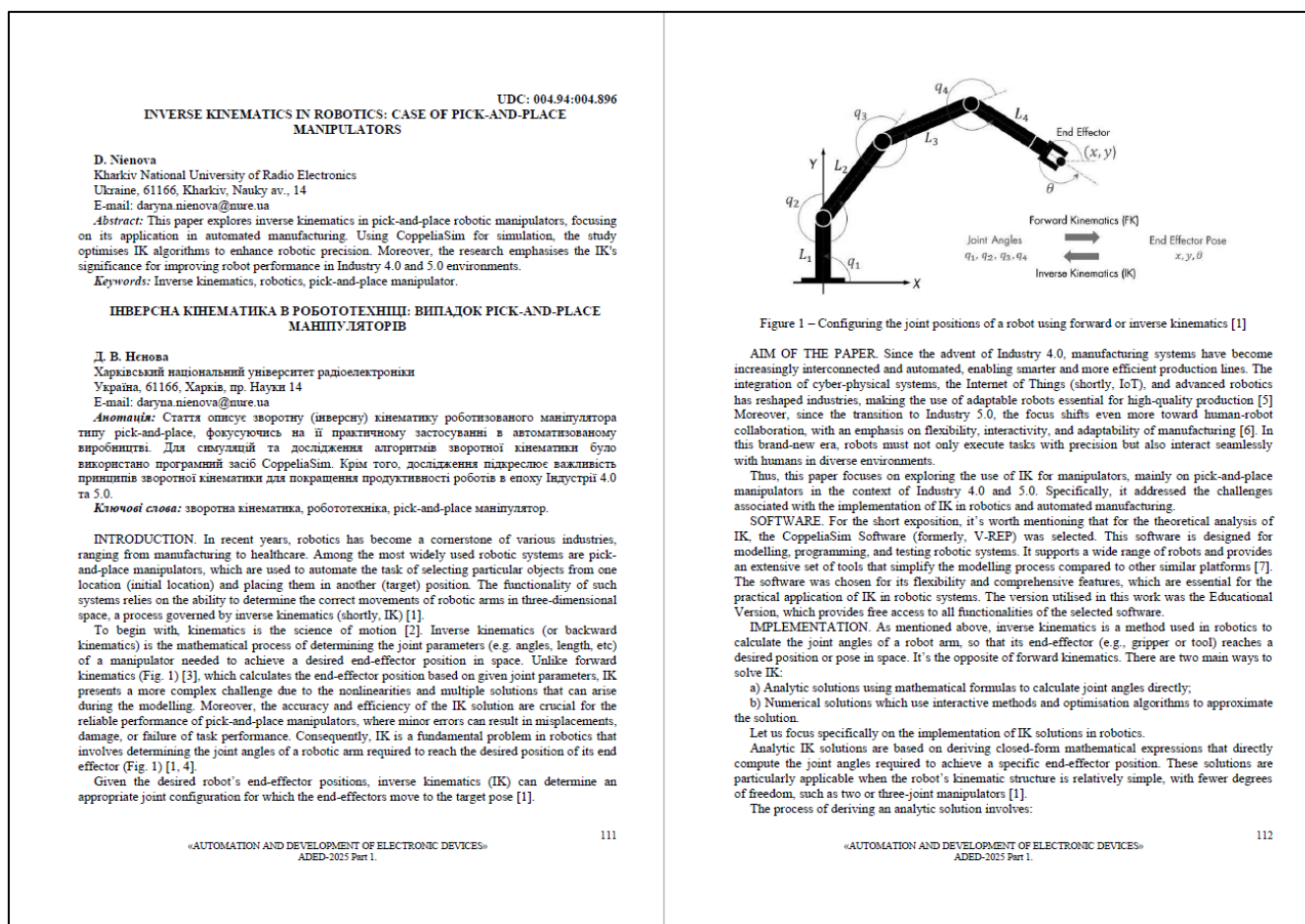


Рисунок А.4 – Опублікована стаття

- Modelling the robot's kinematics (defining the robot's link lengths, joint types, and other geometric parameters);
- Formulating the IK equations; using the robot's Denavit-Hartenberg parameters of transformation matrices, the end-effector position and orientation are expressed as a function of the joint variables;
- Solving the equations. By solving these equations symbolically, the joint angles that will place the end-effector at the target position are obtained.

For instance, for a two-link planar arm, the positions of the joints are described by trigonometric equations, and the joint angles can be derived using inverse trigonometric functions. Once the solution is found, the robot can execute the required motion directly without iterative calculations. While analytic solutions are efficient for simpler robots, they become impractical for more complex robots, such as those with multiple joints or higher degrees of freedom. In such cases, numerical methods are employed to approximate the solution [1].

**ROBOTICS SYSTEM.** To illustrate the application of inverse kinematics in robotics, let's consider the construction of a two-link manipulator system, specifically designed around the principles of IK. This system serves as an effective demonstration of how IK can be utilised in robotic movements. The two-link manipulator has proven to be a useful building block for more advanced robotic systems, particularly in the development of pick-and-place manipulators. These more complex systems often incorporate three-link manipulators, which further enhance their capabilities through the implementation of the IK.

In Figure 2, one can see an example of a two-link manipulator built in CoppeliaSim.

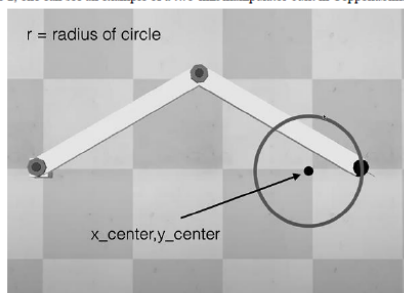


Figure 2 – Two-link manipulator based on the IK mechanism in CoppeliaSim

In accordance with Figure 2, the two-link manipulator consists of two rigid links connected by joints. The manipulator is operated in a 2D plane (typically), where the goal is to control the end-effector (the tool at the end of the manipulator, the so-called manipulator's "hand") to reach a desired position and orientation in space.

Link 1 is the first segment of the manipulator and is attached to the base, link 2 is the second segment connected to the end of Link 1, and the end-effector is attached to this link. Joints provide rotational movement at the connection points.

In terms of IK, the problem is to calculate the required joint angles that allow the manipulator's end-effector to reach a desired position in space.

Additionally, Figure 2 shows the mechanics of the two-link manipulator and the process of solving for the joint angles needed to place the end-effector in a specific position. In a more advanced pick-and-place manipulator, the same principles apply, but the system will often involve more degrees of freedom and additional complexities such as joint limits, workspace constraints, and task-specific requirements.

Through the use of CoppeliaSim and similar platforms, one can test various IK algorithms to find the most efficient solutions for specific tasks, such as pick-and-place operations. By using simulation tools, they can optimise robotic movements, ensuring the manipulator reaches the target position with minimal error while avoiding obstacles and adhering to physical constraints. After thoroughly completing the calculations related to the placement and positioning of the end-effector, IK was applied to develop a robotic system. This system was designed to precisely pick and place objects, moving them to designated target positions. Throughout this process, the robot was also programmed to navigate around intentionally placed obstacles, demonstrating its ability to adapt and function effectively in a dynamic environment (Figures 3 and 4).

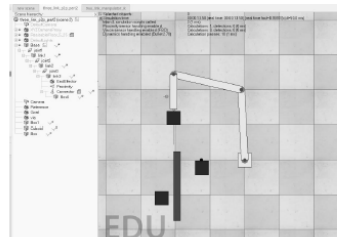


Figure 3 – Pick-and-Place manipulator from the top position

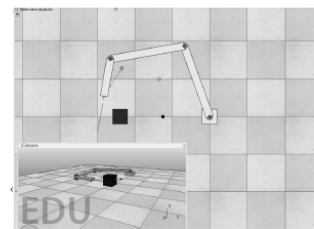


Figure 4 – View from the camera

Consequently, by refining the inverse kinematics algorithms in such simulations, it becomes possible to enhance the overall efficiency, reliability, and versatility of robotic systems used in automated manufacturing and logistics. Additionally, the use of simulation tools like CoppeliaSim enables the testing of various configurations and the optimisation of robotic movements in a safe, controlled virtual environment before physical deployment.

**CONCLUSION:** In conclusion, the study of inverse kinematics represents a cornerstone in development of advanced robotic systems, particularly for tasks such as pick-and-place manipulation. With the rapid evolution of robotics and the ongoing shift towards Industry 4.0 and 5.0, the application of the IK in robotic systems is becoming ever more important in enhancing efficiency, precision, and adaptability in manufacturing and automation environments.

The study has demonstrated that both analytic and numerical solutions to IK can be effectively employed to optimise robotic arm movement for both simple and complex robotic systems. Moreover, the research highlights that mastering the principles of inverse kinematics and refining the algorithms used to solve IK can significantly improve robotic performance in pick-and-place applications. Furthermore, as robotics continues to advance, the integration of more sophisticated IK algorithms in combination with machine learning, artificial intelligence, and real-time simulation will enable robots to adapt to even more complex tasks and environments.

Thus, the further research and development of IK solutions will be key to driving the success of robotics in Industry 5.0. These advancements will further propel the integration of robotics into a variety of industries, ranging from manufacturing to healthcare, thereby making a difference in global technological advancements.

#### References:

1. MathWorks. (n.d.). What Is Inverse Kinematics? [Www.mathworks.com. https://www.mathworks.com/discovery/inverse-kinematics.html](https://www.mathworks.com/discovery/inverse-kinematics.html) (date of access: 07.04.2025)
2. Modeling Inverse Kinematics in a Robotic Arm. MATLAB & Simulink. <https://www.mathworks.com/help/fuzzy/modeling-inverse-kinematics-in-a-robotic-arm.html> (date of access: 07.04.2025)
3. Forward Kinematics, ROS Robotics. (n.d.). Home. <https://www.rosroboticslearning.com/forward-kinematics> (date of access: 07.04.2025)
4. Owen-Hill, A. Inverse Kinematics in Robotics: What You Need to Know. RoboDK Blog. <https://robodk.com/blog/inverse-kinematics-in-robotics-what-you-need-to-know/> (date of access: 07.04.2025)
5. Невлюдов І. Ш., Андрусевич А. О., Євсєєв В. В., Максимова С. С. Моделі та методи кіберфізичних виробничих систем в концепції Industry 4.0 : Монографія. – Харків : 2023. – 321 с.
6. Gurin D. Key features and differences between Industry 5.0 and Industry 4.0.: Тези доповідей. Запоріжжя : XII Міжнародна науково-практичної конференція, 2024. URL: <https://openarchive.nure.ua/handle/document/29571> (date of access: 07.04.2025)
7. CoppeliaSim. Robotics simulation platform. URL: [https://www.mathworks.com/products/connections/product\\_detail/coppeliasim.html](https://www.mathworks.com/products/connections/product_detail/coppeliasim.html) (date of access: 07.04.2025)

*Scientific adviser: Artem Bronnikov, Associate Professor of the Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv National University of Radio Electronics.*

Рисунок А.6 – Висновки та перелік джерел посилання

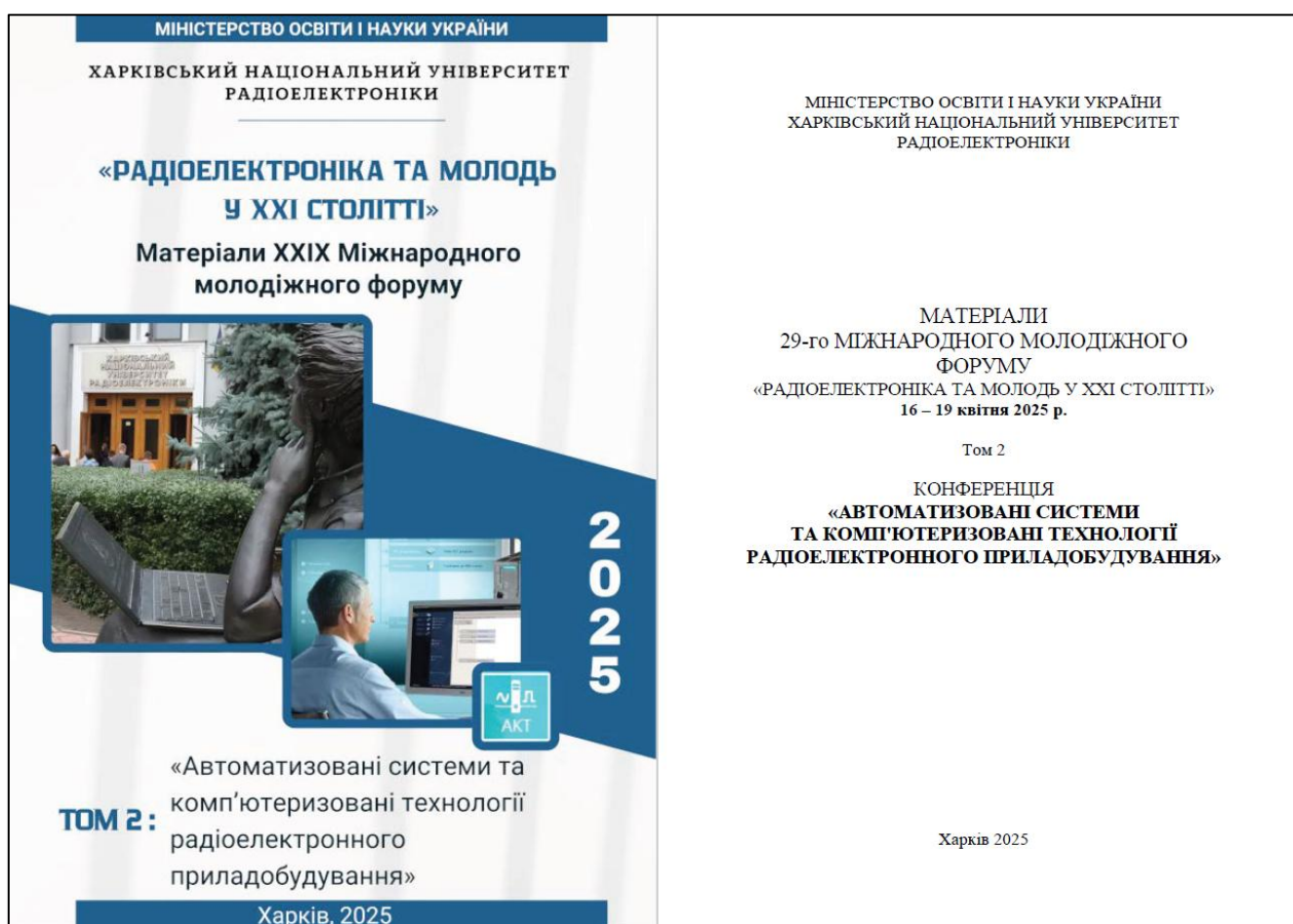


Рисунок А.7 – Титульний аркуш електронного видання

<p>УДК [681.5:004]:[621.37/39.681.2](06) 29-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». 36. матеріалів форуму. Т. 2. – Харків: ХНУРЕ, 2025. – 150 с.</p> <p>У збірнику представлено матеріали доповідей учасників 29-го Міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті».</p> <p>Для науковців, викладачів, практичних працівників, студентів, а також широкого кола читачів, які цікавляться цією проблематикою.</p> <p>Відповідальність за зміст поданого матеріалу несе його автор.</p> <p>Видання підготовлено факультетом автоматичної та комп'ютеризованих технологій Харківського національного університету радіоелектроніки</p> <p>61166 Україна, Харків, просп. Науки, 14 тел./факс: (057) 7021397 E-mail: <a href="mailto:mref21@nure.ua">mref21@nure.ua</a></p> <p>ISBN 978-966-659-406-1</p> <p>© Харківський національний університет радіоелектроніки (ХНУРЕ), 2025</p>	<p style="text-align: center;"><b>ПРОГРАМНИЙ КОМПЕТ КОНФЕРЕНЦІЇ</b></p> <p>Филипенко О. І. доктор технічних наук, професор, декан факультету Автоматичної та комп'ютеризованих технологій (АКТ)</p> <p>Невлюдов І.Ш. доктор технічних наук, професор, завідувач кафедри комп'ютерно-інтегрованих технологій, автоматизації та мехатронки (KITAP)</p> <p>Giergiel M. Ph.D., D.Sc.Eng., AGHUniversity of Science and Technology, Krakow, Poland.</p> <p>Павлиш В.А. кандидат технічних наук, професор кафедри електронних засобів інформаційно-комп'ютерних технологій.</p> <p>Мосьпан В.О. кандидат технічних наук, професор, декан факультету електроніки і комп'ютерної інженерії (ФЕКІ)</p> <p>Єфіменко А.А. доктор технічних наук, професор, завідувач кафедри електронні засоби і інформаційно-комп'ютерні технології (ЕЗІКТ)</p> <p>Хорошайло Ю.Є. кандидат технічних наук, професор, завідувач кафедри проектування та експлуатації електронних апаратів (ПЕЕА).</p> <p>Євсєєв В.В. доктор технічних наук, професор кафедри комп'ютерно-інтегрованих технологій, автоматизації та мехатронки (KITAP)</p> <p>Ключник І.І. кандидат технічних наук, професор, професор кафедри проектування та експлуатації електронних апаратів (ПЕЕА)</p>
---	---

Рисунок А.8 – Інформація про електронне видання

УДК 004.94:004.896

**МОДЕЛЮВАННЯ РОБОТА-МАНІПҮЛЯТОРА У ПРОГРАМНОМУ  
ЗАБЕЗПЕЧЕННІ COPPELIASIM**

Ненова Д. В.

e-mail: daryna.nenova@mure.ua

Харківський національний університет радіоелектроніки, каф. КІТАР  
м. Харків, Україна

This paper focuses on the importance of robotic manipulators in industrial automation processes. Moreover, it highlights how modelling and simulation techniques contribute to improving reliability in manufacturing operations. The implementation of software tools, such as CoppeliaSim, enables the creation of detailed models for pick-and-place manipulators, which this paper proposed. By using scripts for pick-and-place robotic tasks, the software enhances the accuracy and efficiency of the manipulator's operations. The paper also emphasises the significant benefits which make a difference in industrial automation processes, such as improved productivity, reduced errors, and enhanced safety in manufacturing.

Основне питання автоматизації виробничого процесу полягає в створенні високонадійних, точних, швидких та, за необхідності, здатних працювати в небезпечних умовах, автоматизованих систем або компонентів [2,3]. Саме тому проектування, моделювання та безпосереднє використання роботів-маніпуляторів є важливим етапом у розвитку автоматизації виробничого процесу, оскільки ці системи дозволяють значно підвищити ефективність різноманітних процесів на підприємстві. Маніпуляційні роботи призначені для операцій по зміні просторового положення будь-яких предметів, таких як інструменти, деталі чи вузли інших машин. Вони стаціонарні, тобто встановлені в одному місці. Маніпуляції із предметами виконують щодо місця свого закріплення [3]. Проектування роботів-маніпуляторів містить розробку механізмів, що забезпечують високу точність рухів, стабільність роботи та здатність виконувати складні маніпуляції в умовах різноманітних виробничих задач.

Однією із частих потреб автоматизованого виробництва є використання функцій захоплення та розміщення елементів, деталей тощо. Для виконання цієї дії часто використовується маніпулятор захоплення та розміщення (Pickand Place Manipulator) [5], приклад такого маніпулятора, побудованого з використанням програмного забезпечення CoppeliaSim, показано на рисунку 1.1. Створення таких маніпуляторів у запропонованому програмному забезпеченні дозволяє не лише візуалізувати рухи робота, але й налаштувати точність виконання рухів через використання скриптів для управління позиціями об'єктів, що переміщує маніпулятор. Наприклад, створення спеціального скрипту для генерації елементів, що будуть переміщатися по конвеєрній стрічці, як запропоновано на рисунку 1.1.

8

Маніпулятор такого типу здатний ефективно вирішувати завдання, що включають транспортування та укладання компонентів, виконання складних маніпуляцій з деталями, що відрізняються за формою, розмірами та/або вагою. Цей процес є невіддільною частиною багатьох виробничих ліній, включаючи складання, пакування, сортування та інші типи автоматизованих операцій. У цьому контексті використання програмного забезпечення CoppeliaSim надає суттєві переваги інженерам у галузі автоматизації виробничих процесів, оскільки цей інструмент пропонує широкий спектр засобів для розробки, моделювання, програмування та інтеграції роботів і роботизованих систем у виробничу діяльність. Крім того, програмне забезпечення CoppeliaSim дозволяє ефективно візуалізувати та тестувати різноманітні сценарії роботи роботизованих систем, що значно прискорює процес розробки.

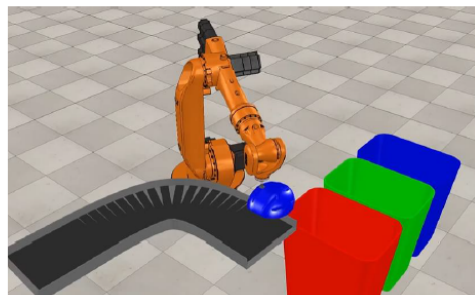


Рисунок 1.1 – Робот-маніпулятор типу pick-and-place у програмному забезпеченні CoppeliaSim

В умовах розвитку Індустрії 5.0, роботизовані системи, такі як маніпулятори типу pick-and-place, інтегруються в нову модель автоматизованого виробництва, де акцент ставиться на співпраці людини та машини. Цей підхід дозволяє досягти більш гнучких і персоналізованих виробничих процесів, де роботи виконують рутинні завдання, а люди можуть зосередитися на стратегічних аспектах роботи. Також важливою технологією є цифрові двійники. Цифровий двійник (digital twin) – це програмний аналог фізичного пристрою, що моделює внутрішні процеси, технічні характеристики й поведінку реального об'єкта в умовах впливу

9

## Рисунок А.9 – Оpubліковані тези доповіді

перешкод та навколишнього середовища [4]. Якщо для традиційної промисловості набуття необхідних характеристик виробу ведеться через численні натурні випробування, то завдання Індустрії 4.0 – проводити багаторазові випробування за допомогою цифрового двійника, а натурні випробування проходити з першого разу[4]. Отже, цифрові двійники дають змогу точніше моделювати роботу роботизованих систем, зменшуючи час на розробку та тестування нових рішень, що дозволяє прогнозувати ефективність роботи маніпуляторів, підвищуючи їх продуктивність і точність. В результаті, такі технології сприяють підвищенню ефективності виробничих процесів.

У результаті, моделювання роботів-маніпуляторів не тільки сприяє досягненню максимальної продуктивності й мінімальних витрат енергії при виконанні операцій, але й допомагає мінімізувати помилки, що часто виникають на підприємствах, заснованих на не роботизованому виконанні завдань [1]. Отже, завдяки застосуванню сучасних програмних засобів (наприклад, програмного забезпечення CoppeliaSim), штучного інтелекту та машинного навчання, роботи-маніпулятори можуть адаптуватися до змінних умов виробництва, працюючи з високою точністю в різноманітних сценаріях [1]. Зокрема, не тільки покращити продуктивність і точність, але й створити нові перспективи для виконання складних завдань автоматизації, підвищуючи ефективність виробництва та мінімізуючи ризики для людей.

#### Список використаних джерел:

1. Іванов В.О., Дегтярьов І.М. Технологічні основи гнучких автоматизованих виробництв : навч. посіб. Суми : Сумський державний університет, 2022. – 204 с.
2. Літвін, О., Паньков, С. Роботизовані маніпулятори особливого призначення. Технічні науки та технології. *Technical Sciences and Technologies*. 1(19), 2020 С.81-88. URL: [https://doi.org/10.25140/2411-5363-2020-1\(19\)-81-88](https://doi.org/10.25140/2411-5363-2020-1(19)-81-88). (дата звернення: 01.03.2025)
3. Невлюдов І. Ш., Андрусевич А. О., Євсєєв В. В., Новоселов С. П., Демська Н. П. Проектування мобільних маніпуляційних роботів : Монографія. – Харків : ХНУРЕ, 2022. – 427 с.
4. Невлюдов І. Ш., Новоселов С. П., Сичова О. В. Застосування цифрових двійників в технічних засобах автоматизації для розроблення програмно-технічних комплексів АСУТП : навч. посіб. Харків : Видавництво Іванченка І.С., 2023. – 267 с.
5. Surati, S., Hedao, S., Rotti, T., Ahuja, V., Patel, N. PickandPlaceRoboticArm: Areviewpaper. *International Research Journal of Engineering and Technology (IRJET)*. Vol. 08. 2021. P. 2121-2129

Рисунок А.10 – Висновки та перелік джерел посилання

## ДОДАТОК Б Код програми

```

function sysCall_init()
  RefPosition=sim.getObjectHandle("Reference")
  EndEffector=sim.getObjectHandle("Endeffector")

  start = sim.getObjectHandle("start")

  Connector=sim.getObjectHandle("UR5_connection")
  Proximity=sim.getObjectHandle("Proximity")
  Vision=sim.getObjectHandle("Vision")

  BlueBin = sim.getObjectHandle("BlueBin")
  RedBin = sim.getObjectHandle("RedBin")

  index=0
  objects_to_move = {};
  while true do
    shape=sim.getObjects(index,sim.object_shape_type)
    if (shape==-1) then
      break
    end
    result1,parameter1 = sim.getObjectInt32Parameter(shape,sim.shapeintparam_static);
    if (parameter1==0) then
      table.insert(objects_to_move, shape)
    end
    index=index+1
  end

  position_start = sim.getObjectPosition(start,-1)
  position_blue_bin = sim.getObjectPosition(BlueBin,-1)
  position_red_bin = sim.getObjectPosition(RedBin,-1)

  -- Adjust bin positions to place boxes above them (increased clearance)
  position_blue_bin[3] = position_blue_bin[3] + 0.15
  position_red_bin[3] = position_red_bin[3] + 0.15

  --- finite state machine (FSM) parameters ----
  state_start = 1;
  state_start2pick = 2;
  state_pick = 3;
  state_lift = 4
  state_lift2start = 5;
  state_start2bin = 6;
  state_lower = 7;
  state_place = 8;
  state_lift_after_place = 9;
  state_bin2start = 10;
  state_stop = 11;

  --- starting state for the state machine

```

```

state=state_start;
duration = 1.5; -- Increased duration for smoother movement
t_f = 1;
box_no = 1;
current_box_color = "";
attachedBox = nil; -- Initialize attached box variable

-- Define safe height for movements
safe_height = 0.2; -- Reduced height above table for safer movement

end

-----
function sysCall_actuation()
    t=sim.getSimulationTime()

    --- pick and place all boxes sequentially ---
    n = table.getn(objects_to_move)
    if box_no <= n then
        position_pick=sim.getObjectPosition(objects_to_move[box_no],-1)
        position_pick[3] = position_pick[3]+0.01; -- Very small offset to touch the box
        state = moveSimpleManipulator(position_pick)
        if (state==state_stop and box_no<n) then
            state = state_start;
            t_f = sim.getSimulationTime()
            box_no = box_no + 1;
            attachedBox = nil; -- Reset for next box
        end
    end
end

-----
function sysCall_sensing()
    -- put your sensing code here
end

-----
function sysCall_cleanup()
end

-----
function getBoxDestination(colour)
function getBoxDestination(box_handle)
    local image, resX, resY = sim.getVisionSensorCharImage(Vision)
    local r_sum, g_sum, b_sum = 0, 0, 0
    local pixel_count = resX * resY
    for i = 0, pixel_count - 1 do
        r = string.byte(image, i * 3 + 1)
        g = string.byte(image, i * 3 + 2)
        b = string.byte(image, i * 3 + 3)
        r_sum = r_sum + r
        g_sum = g_sum + g
        b_sum = b_sum + b
    end
    local r_avg = r_sum / pixel_count

```

```

local g_avg = g_sum / pixel_count
local b_avg = b_sum / pixel_count
local max_val = math.max(r_avg, g_avg, b_avg)
if max_val == 0 then max_val = 1 end
local r_n = r_avg / max_val
local g_n = g_avg / max_val
local b_n = b_avg / max_val
if b_n > 0.6 and r_n < 0.4 and g_n < 0.4 then
    return "blue"
elseif r_n > 0.6 and g_n < 0.4 and b_n < 0.4 then
    return "red"
else
    return "unknown"
end
end

```

```

-----
function moveSimpleManipulator(position_pick)
----- transitions for FSM -----
if (state == state_start and t >= t_f) then
    state = state_start2pick;
    pos_i = position_start;
    pos_f = position_pick;
    t_i = t_f;
    t_f = t_i + duration;

elseif (state == state_start2pick and t >= t_f) then
    state = state_pick;
    t_i = t_f;
    t_f = t_i + 0.5; -- Short delay for picking

elseif (state == state_pick and t >= t_f) then
    state = state_lift;
    pos_i = position_pick;
    pos_f = {position_pick[1], position_pick[2], position_pick[3] + safe_height}; -- Lift box up
    t_i = t_f;
    t_f = t_i + duration;

elseif (state == state_lift and t >= t_f) then
    state = state_lift2start;
    pos_i = {position_pick[1], position_pick[2], position_pick[3] + safe_height};
    pos_f = {position_start[1], position_start[2], position_start[3] + safe_height}; -- Move to start at safe height
    t_i = t_f;
    t_f = t_i + duration;

elseif (state == state_lift2start and t >= t_f) then
    -- Get destination based on box colour (hardcoded)
    current_box_color = getBoxDestination(box_no)
    state = state_start2bin;
    pos_i = {position_start[1], position_start[2], position_start[3] + safe_height};
    if current_box_color == "blue" then
        pos_f = {position_blue_bin[1], position_blue_bin[2], position_blue_bin[3] + safe_height};
    else
        pos_f = {position_red_bin[1], position_red_bin[2], position_red_bin[3] + safe_height};
    end
end

```

```

t_i = t_f;
t_f = t_i + duration;

elseif (state == state_start2bin and t >= t_f) then
    state = state_lower;
    if current_box_color == "blue" then
        pos_i = {position_blue_bin[1], position_blue_bin[2], position_blue_bin[3] + safe_height};
        pos_f = position_blue_bin;
    else
        pos_i = {position_red_bin[1], position_red_bin[2], position_red_bin[3] + safe_height};
        pos_f = position_red_bin;
    end
    t_i = t_f;
    t_f = t_i + duration;

elseif (state == state_lower and t >= t_f) then
    state = state_place;
    t_i = t_f;
    t_f = t_i + 0.5; -- Short delay for placing

elseif (state == state_place and t >= t_f) then
    state = state_lift_after_place;
    if current_box_color == "blue" then
        pos_i = position_blue_bin;
        pos_f = {position_blue_bin[1], position_blue_bin[2], position_blue_bin[3] + safe_height};
    else
        pos_i = position_red_bin;
        pos_f = {position_red_bin[1], position_red_bin[2], position_red_bin[3] + safe_height};
    end
    t_i = t_f;
    t_f = t_i + duration;

elseif (state == state_lift_after_place and t >= t_f) then
    state = state_bin2start;
    if current_box_color == "blue" then
        pos_i = {position_blue_bin[1], position_blue_bin[2], position_blue_bin[3] + safe_height};
    else
        pos_i = {position_red_bin[1], position_red_bin[2], position_red_bin[3] + safe_height};
    end
    pos_f = position_start;
    t_i = t_f;
    t_f = t_i + duration;

elseif (state == state_bin2start and t >= t_f) then
    state = state_stop
end

----- actions in the FSM -----
if (state == state_start2pick
    or state == state_lift
    or state == state_lift2start
    or state == state_start2bin
    or state == state_lower
    or state == state_lift_after_place
    or state == state_bin2start) then

```

```

    moveIK(pos_i,pos_f,t_i,t_f)
end

if (state==state_pick) then
    attachedBox=graspObject()
    print("Attempting to grasp box. AttachedBox:", attachedBox) -- Debug print
end

if (state==state_place) then
    if attachedBox ~= nil then
        releaseObject(attachedBox)
        print("Released box:", attachedBox) -- Debug print
    else
        print("Warning: No box attached to release!")
    end
end

return state
end

-----
function moveIK(position_i,position_f,t_i,t_f)
    x_f = position_f[1];
    y_f = position_f[2];
    z_f = position_f[3];

    x_i = position_i[1];
    y_i = position_i[2];
    z_i = position_i[3];
    dt = t_f - t_i;
    dx = x_f - x_i;
    dy = y_f - y_i;
    dz = z_f - z_i;
    if (t>=t_i and t<=t_i+dt) then
        del_t = (t-t_i);
        x = x_i + (del_t/dt)*dx;
        y = y_i + (del_t/dt)*dy;
        z = z_i + (del_t/dt)*dz;
        position_ref = {x,y,z}
        sim.setObjectPosition(RefPosition,-1,position_ref)
    end
end

-----
function releaseObject(attachedShape)
    sim.setObjectParent(attachedShape,-1,true)
end

-----
function graspObject()
    local attachedShape = nil
    local current_box = objects_to_move[box_no]

    print("Trying to grasp box number:", box_no)
    print("Current box handle:", current_box)

```

```

-- Check if the current box exists
if current_box == nil then
  print("Error: Current box is nil!")
  return nil
end

-- Check proximity sensor with the specific box
local result3, distance = sim.checkProximitySensor(Proximity, current_box)
print("Proximity sensor result:", result3, "distance:", distance)

-- Get box parameters
local result1, parameter1 = sim.getObjectInt32Parameter(current_box, sim.shapeintparam_static)
local result2, parameter2 = sim.getObjectInt32Parameter(current_box, sim.shapeintparam_respondable)

print("Box parameters - Static:", parameter1, "Respondable:", parameter2)

-- Simplified grasping logic - try to grasp the current box directly
if parameter1 == 0 then -- Box is not static (moveable)
  -- Try to attach the box regardless of proximity sensor
  sim.setObjectParent(current_box, Connector, true)
  attachedShape = current_box
  print("Successfully grasped object:", attachedShape)
else
  print("Box is static, cannot grasp")
end

-- Alternative approach: if proximity sensor is working, use it
if attachedShape == nil and result3 == 1 then
  sim.setObjectParent(current_box, Connector, true)
  attachedShape = current_box
  print("Successfully grasped object using proximity:", attachedShape)
end

-- Last resort: force grasp if we're close enough
if attachedShape == nil then
  print("Force grasping the current box...")
  sim.setObjectParent(current_box, Connector, true)
  attachedShape = current_box
  print("Force grasped object:", attachedShape)
end

return attachedShape
end

```

## ДОДАТОК В Демонстраційний матеріал

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ  
КАФЕДРА КІТАР

**МОДЕЛЮВАННЯ РОБОТА-  
МАНІПУЛЯТОРА ТА ЙОГО  
ФУНКЦІЙ ІЗ ВИКОРИСТАННЯМ  
ПРОГРАМНОГО  
ЗАБЕЗПЕЧЕННЯ CORRELIASIM**

Підготувала студ.групи АКТАКІТ-21-1  
Нєнова Д. В.

Керівник: доц. Бронников А. І.



ХАРКІВ, 2025

### ЗМІСТ

- Актуальність роботи
- Мета роботи
- Об'єкт, предмет, методи
- Завдання
- Аналіз літератури
- Вибір та обґрунтування технічних засобів
- Практична частина роботи
- Розробка та моделювання
- Приклад роботи системи
- Висновки

•••

2

Рисунок В.1 – Титульний аркуш та зміст презентації

## Актуальність

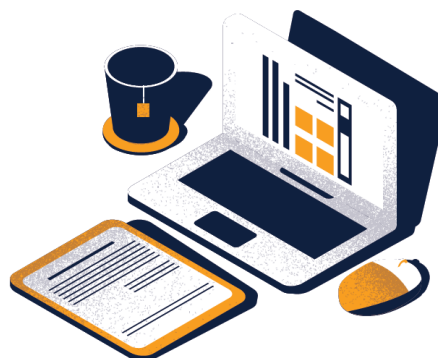
Актуальність розробки обумовлена необхідністю вирішення низки питань, пов'язаних зі зниженням витрат, підвищенням ефективності та гнучкості роботизованих систем у виробничих процесах.



3

## Мета

Моделювання робота-маніпулятора та його функцій у віртуальному середовищі CorreliaSim для демонстрації автоматизованого процесу захоплення та переміщення об'єктів.



4

Рисунок В.2 – Актуальність та мета роботи

## Об'єкт, предмет, методи розробки



### Об'єкт розробки:

Процес моделювання робототехнічних систем типу Pick-and-Place



### Предмет розробки:

Методи та засоби моделювання робота-маніпулятора для автоматизації процесів захоплення та переміщення об'єктів у віртуальному середовищі CoppeliaSim.



### Методи розробки:

аналіз проблеми, розробка моделі та програмного забезпечення для керування рухами робота-маніпулятора типу Pick-and-Place.



5

## Завдання

- проведено аналіз сучасного стану автоматизованих систем;
- проведено підбір елементної бази та обґрунтувати технічні рішення для моделювання;
- розроблено схему роботи модельованої системи;
- проведено математичні розрахунки для втілення поставленої задачі;
- розроблено модель робота у програмному середовищі CoppeliaSim;
- створено програмне забезпечення для керування роботом із використанням мов програмування Python та Lua;
- проведено симуляцію створеної системи.



6

Рисунок В.3 – Об'єкт, предмет, мета та завдання роботи

## Аналіз літератури



### ✓ Автоматизація виробництва

Автоматизація виробництва є важливим критерієм стрімкого розвитку промисловості.

### ✓ Industry 4.0

В умовах швидкого розвитку Industry 4.0 автоматизація виробничих ліній дає широкий спектр можливостей для подальшого розвитку.

### ✓ Новітні технології

Використання IoT, роботизованих систем, систем комп'ютерного зору, моделювання та ШІ значно полегшують виконання складних задач на виробництві.

7

## Технічні засоби



### 🎓 Імітаційне моделювання

### 🎓 CoppeliaSim

- гнучкість у налаштування симуляцій;
- підтримка різних типів робот, маніпуляторів, додаткового устаткування тощо;
- мультимовна підтримка;
- власний API ики.

8

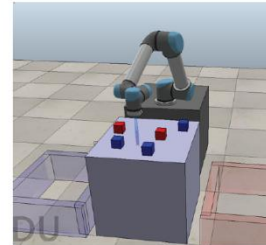
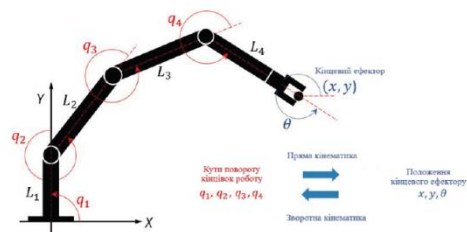
Рисунок В.4 – Аналіз літератури та ТЗ

## Практична частина роботи

Математична частина

Моделювання

Симуляція



9

## Математична частина

Задача зворотної кінематики

ТАКИМ ЧИНОМ, ЗГІДНО ІЗ РОЗРАХУНКАМИ НА КОЖНОМУ КРОЦІ СИМУЛЯЦІЇ ОБЧИСЛЮЄТЬСЯ, НАСКІЛЬКИ КІНЦЕВИЙ ЕФЕКТОР ВІДХИЛЯЄТЬСЯ ВІД БАЖАНОГО ПОЛОЖЕННЯ. ЦЕ ВІДХИЛЕННЯ ПЕРЕВОДИТЬСЯ У ВІДПОВІДНІ ЗМІНИ КУТІВ СУГЛОБІВ ЗА ДОПОМОГОЮ ЗВОРотної МАТРИЦІ ЯКОБІ. ОТРИМАНІ ЗМІНИ КУТІВ ДОДАЮТЬСЯ ДО ПОТОЧНИХ ЗНАЧЕНЬ КУТІВ, ТИМ САМИМ ОНОВЛЮЮЧИ ПОЛОЖЕННЯ МАНІПУЛЯТОРА.

Таблиця 2.2 – Результати ітераційного наближення

Ітерація	$\theta_1$ , рад	$\theta_2$ , рад	x, м	y, м	Помилка $\Delta x$
1	2	3	4	5	6
1	-0,28	0,72	1,31	0,41	0,13
2	-0,32	0,65	1,25	0,47	0,06
3	-0,30	0,63	1,21	0,49	0,02
4	-0,295	0,62	1,20	0,50	0,003

9

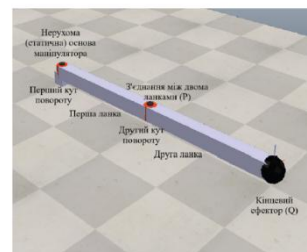
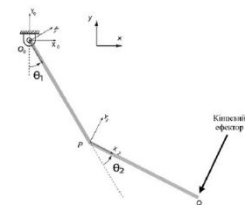
Рисунок В.5 – Практична та математична частини

## Розробка

Перша частина роботи передбачає розробку дволанкового манипулятора.

Друга частина передбачає створення імітаційної моделі робота типу Pick-and-Place у програмному середовищі CoppeliaSim, де реалізуються алгоритми та методи управління.

Третім етапом моделювання є створення тривимірної моделі робота-манипулятора із використанням вбудованих функцій програмного середовища CoppeliaSim.

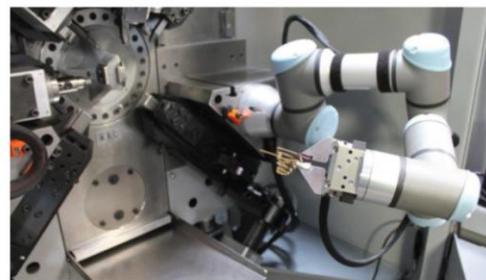


10

## Моделювання

Для моделювання було обрано промисловий робот типу Pick-and-Place типу UR5 (The Universal Robots UR5).

Цей промисловий робот, призначений для автоматизації повторюваних завдань. Він має вантажопідйомність до 5 кг, радіус дії до 850 мм, вагу 18,5 кг і 6 ступенів свободи



11

Рисунок В.6 – Розробка та моделювання системи

## Приклад роботи системи

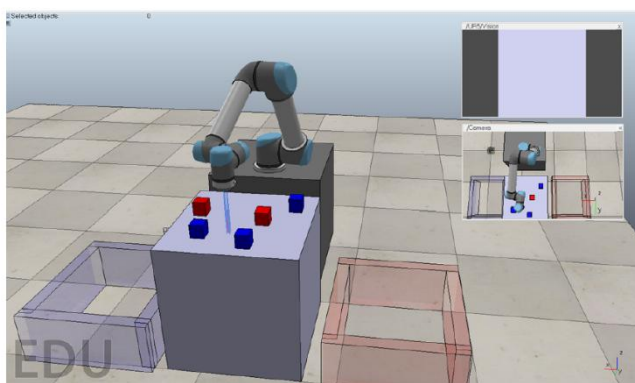



Рисунок В.7 – Приклад роботи системи



## Висновки

**☑ Таким чином,**

У роботі було обґрунтовано вибір програмного забезпечення для моделювання системи, враховуючи його широкі функціональні можливості, підтримку скриптових мов, а також здатність інтеграції з іншими платформами. Проведено порівняльний аналіз переваг та недоліків середовища, що дозволило обрати найефективніший шлях реалізації поставленого завдання

**☑ Крім того,**

Результати симуляцій підтвердили коректність роботи системи, візуалізація рухів у середовищі CoppeliaSim дозволила оцінити точність та ефективність запрограмованих сценаріїв. Крім цього, було визначено основні вимоги до умов роботи програміста та інженера при роботі з автоматизованими системами, сформульовано заходи безпеки при використанні електроустаткування та тестуванні робототехнічних систем.

13

---

...

# ДЯКУЮ ЗА УВАГУ

Нєнова Д.В., каф. КІТАР

...

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ, КАФЕДРА КІТАР

14

Рисунок В.8 – Висновки та фінальний слайд презентації

