

ДОДАТОК А

Апробація результатів роботи

Міністерство освіти і науки України



NURE

Харківський національний університет
радіоелектроніки

ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2024

(Випуск 2)

[електронне видання]



<http://nure.ua/department/kafedra-komp-yuterno-integrovanih-tehnologiy-avtomatizatsiyi-ta-mehatroniki-kitam>



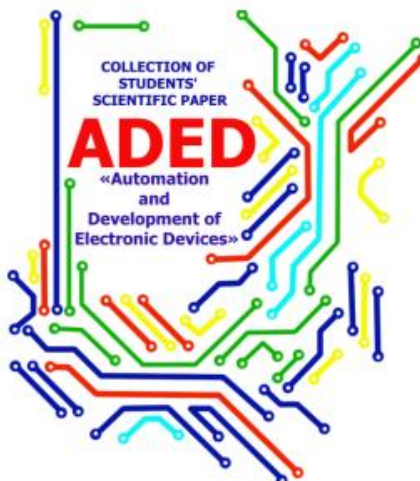
<http://itez.zntu.edu.ua/>



<http://kafea.kdu.edu.ua>

Харків 2024

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(КІТАР)



ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2024

(Випуск 2)

[електронне видання]

Харків 2024

- Головий редактор** **Невлюдов Ігор Шакирович**, доктор технічних наук, професор, завідувач кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківського національного університету радіоелектроніки.
- Редакційна колегія:** **Филипенко 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.
Свищ 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, декана факультета Радіоелектроніки і телекомунікацій, Національного університету «Запорізька політехніка».
- Відповідальний редактор:** **Євсєєв Владислав В'ячеславович**, доктор технічних наук, професор кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківського національного університету радіоелектроніки.

УДК 004.738.5

АВТОМАТИЗАЦІЯ ЛОГІСТИЧНИХ СИСТЕМ З ВИКОРИСТАННЯМ КІБЕРФІЗИЧНИХ ПІДХОДІВ

Краснопоров М.Р., Казановська К.А.

Харківський національний університет радіоелектроніки

Україна, 61166, Харків, пр. Науки, 14

E-mail: mykhailo.krasnoporov@nure.ua, kateryna.kazanovska@nure.ua

Анотація: У статті розглянуто логістичні системи. Доведено, що автоматизація логістичних систем є перспективним напрямком розвитку галузі комп'ютеризованих та робототехнічних системи, що може бути розширений та удосконалений шляхом використання кіберфізичних підходів.

Ключові слова: автоматизація, кіберфізичні системи, складська логістика, автоматизовані складські системи, роботи, RFID.

AUTOMATION OF LOGISTICS SYSTEMS USING CYBER- PHYSICAL APPROACHES

M.Krasnoporov, K. Kazanovska

Kharkiv National University of Radio Electronics

Ukraine, 61166, Kharkiv, pr. Nauki, 14

E-mail: mykhailo.krasnoporov@nure.ua, kateryna.kazanovska@nure.ua

Abstract: The article examines logistic systems. It has been proven that the automation of logistic systems is a promising direction for the development of computer-based and robotic systems, which can be expanded and improved through the use of cyber-physical approaches.

Keywords: automation, cyber-physical systems, warehouse logistics, automated warehouse systems, robots, RFID.

ВСТУП. В наш час автоматизація, роботизація різних процесів є дуже важливою, і такою, яка направлена на підвищення продуктивності, без втрати якості на будь-якому виробництві. Також, автоматизовані процеси стрімко розвиваються в усіх сфер людської діяльності. Не є виключенням промислової автоматизація в логістичних процесах: виробництва, складування, відвантаження, транспортування та зберігання. Для реалізації цих напрямків, часто, стали використовувати концепції кіберфізичних систем, так званих cyber-physical system.

Кіберфізична система (КФС) – це механізм, що контролюється або відстежується комп'ютерними алгоритмами і тісно пов'язаний з Інтернетом та його користувачами. В кіберфізичних системах програмне забезпечення тісно пов'язано з фізичними об'єктами. Компоненти КФС взаємодіють на різних часових та просторових рівнях та можуть мати різні, відмінні одна від одної моделі поведінки та взаємодіяти одна з одною різними шляхами, які можуть змінюватися в залежності від контексту.

Прикладами кіберфізичних систем можна вважати «розумні» логістичні, безпілотні автомобільні системи, автоматизовані системи керування,

робототехнічні системи, самокеровані літальні апарати [1]. У КФС використовується міждисциплінарний підхід, який поєднує теорію кібернетики, мехатроніку, промисловий дизайн та науковий метод. Контроль процесів часто пов'язують з вбудованими системами, в яких більше уваги приділяють обчислювальним складовим, і менше — інтенсивному прив'язуванню обчислень до фізичних об'єктів. КФС дещо схожі за архітектурою на Інтернет речей, проте вони мають більш високий рівень взаємозв'язку між фізичними та комп'ютерними компонентами [2-3].

Враховуючи значну потребу нашої держави в постійних, швидких логістичних послугах, можна вважати, обрану тему актуальною та своєчасною.

Автоматизація складу дозволяє оптимізувати робочу силу і здійснити процеси на виробництві без втручання людини, але під повним її контролем. А також вирішить проблеми з витратами на утримання складу. Координують процес комп'ютери або промислові контролери, шляхом керування мобільними платформами, шатловими стелажми, конвеєрними системами і іншими об'єктами для автоматизації роботи складу.

Типи автоматизованих логістичних систем. Автоматизовані стелажі – це універсальне рішення, якщо потрібно оптимізувати простір складських приміщень. Мобільний стелаж – високоефективне рішення для організації для зберігання, що забезпечує найвищу щільність зберігання за рахунок максимального використання вільного простору приміщення. Ще їх називають пересувні стелажі, оскільки вони переміщуються в по напрямних та створюють прохід у необхідному місці.

Шатлові стелажі – відрізняються наявністю рейок на бічній частині. По рейках здійснюється транспортування піддонів за допомогою самохідної платформи, яка і називається шатлом. Завдяки взаємодії платформи зі стелажми площа складу оптимізується і використовується більш опціонально, в порівнянні з іншими способами організації зберігання вантажів. Шатл має автономне джерело живлення. При транспортуванні вантажів він переміщується по рейках уздовж одного каналу.

Переміщення між каналами виконується стандартним навантажувачем. Така конструкція має компактні розміри (часто невелику висоту) і береже робочий простір. Збільшує ефективність роботи в два рази, за рахунок гнучкості в роботі, можливо використовувати різні види палет в одній системі, працюють в режимі non-stop за рахунок легкої заміни батарей.

Мобільні системи збільшують ефективність використання площі від 50-100%. Конструкція влаштована таким чином: мобільні бази рухаються по напрямних, які встановлені в бетонній підлозі. У порівнянні зі звичайними фіксованими стійками, для яких потрібно прохід для роботи вантажної техніки через кожні 2 ряди, для мобільних систем необхідний тільки один проїзд і оператор за допомогою пульта відкриває той, який йому необхідний для роботи.

Гравітаційні стелажі є 2-х видів: поличні і палетні, які забезпечують максимальну щільність зберігання. У той же час, гравітаційні поличні системи часто використовують для комплектації замовлень, оскільки вони дозволяють виключити проходи оператора зібравши максимальну кількість товару на мінімальній площі. Конструкція гравітаційних стелажів уможливує поділ потоків товару: з одного боку - розвантаження, з іншого-підбір.

Конвеєрні системи – це, транспортна система, яка забезпечує переміщення товару в необхідну точку на складі. Застосовуються в процесі завантаження-вивантаження, накопичення, переміщення палет і коробів.

Існує великий асортимент конвеєрних систем, серед яких, найбільшою популярністю в сфері складської логістики користуються стрічковий (прямий і підйомний), роликівий, гнучкий і спіральний конвеєра. Конвеєра сприяють скороченню часу і трудовитрат при переміщенні вантажів.

Особливо корисні на складах з високим товарообігом, адже вони дозволяють більш ефективно використовувати площу складу і збільшити його продуктивність в два рази.

Використовуючи зазначене обладнання та кіберфізичні підходи логічним є створення автоматизованих складських систем.

Автоматизована складська система або автоматизована транспортно-складська система –це така система, що, призначена для автоматизації керування транспортними і складськими пристроями для складання, зберігання, тимчасового нагромадження, розвантаження та доставляння предметів та засобів праці, технологічного оснащення й видалення відходів.

Автоматизовані складські системи не тільки виключають ручну працю, а й дозволяють економити складські площі, прискорювати операції і покращувати контроль за матеріально-технічними запасами, оскільки комп'ютерно-інтегрований комплекс стежить за місцезнаходженням виробу на складі. Ці системи називають також автоматизованими складами.

Робота автоматизованих складів залежить від технологічних інновацій. До таких нововведень відносяться програмне забезпечення для управління складом і складські роботи. Деякі склади використовують один метод автоматизації, тоді як інші інтегрують різні види.

Фізична автоматизація складу означає використання складських роботів для управління запасами на складі. Таке управління передбачає: розвантаження пакетів, запис прийому, систематичне розташування пакетів, отримання конкретних пакетів для виконання замовлень. Усе це трудомісткі завдання, на виконання яких у ручному режимі потрібні години. Саме з цієї дилеми виникає потреба у фізичній автоматизації складів. Існують різні типи складських роботів, рекомендованих для різних складських застосувань.

Шарнірні руки робота – вони з'єднані роботизовані руки які є гнучкими та можуть бути запрограмовані на вивантаження, сортування та впорядкування пакетів. Вони часто сумісні зі складським програмним забезпеченням. Добрий виробник складських роботів також можна налаштувати їх за допомогою датчиків та інших засобів відповідно до потреб вашого складу.

Технології, сприятливі для людини – вони використовуються для збору пакетів і доставки їх персоналу складу, на відміну від працівників, які перетинають склад і перевозять вантажі. Приклади технологій GTP включають автоматизовані керовані транспортні засоби (AGV), Автоматизовані системи зберігання та пошуку (AS/RS) і автономні мобільні роботи (AMR).

Безпілотні керовані апарати – також широко відомі як дрони. Вони використовуються для перевезення невеликих вантажів серед інших завдань з обробки запасів.

Автоматизовані складські роботи мають вбудовані комп'ютеризовані системи, пов'язані з програмним забезпеченням колективного складу. Саме за допомогою цього програмного забезпечення персонал може спілкуватися з роботами, щоб спонукати їх виконувати різні завдання. Їхні рухи контролюються за допомогою датчиків. Це виключає AGV, для яких потрібен маршрут, намічений за допомогою магнітної смуги.

З іншого боку, шарнірні руки робота часто програмуються індивідуально для конкретних завдань. Вони також можуть бути оснащені датчиками, які керуватимуть ними під час сортування та впорядкування пакетів. Однак, на відміну від автоматизованих роботів, вони обробляють запаси зі стаціонарного положення.

Хоча складські роботи в основному розроблені для транспортування пакетів усередині складу, вони також мають інші допоміжні функції. Наприклад, дрони та AMR можуть ідентифікувати пакунки за допомогою датчиків RFID. Таким чином, їх можна використовувати не тільки для перевезення пакунків, але й для виконання інших завдань, як-от інвентаризація та сортування інвентарю.

Автоматизація цифрового складу також називають програмним забезпеченням для автоматизації цифрових процесів або автоматизації складу. Це використання унікального програмного забезпечення для обробки запасів на складі. В ідеалі, з моменту отримання інвентаризації до її відправлення, має бути запис про це в програмному забезпеченні автоматизації складу.

Програми для складської логістики містять, зазвичай, записи про всі предмети, місце їх зберігання та час їх надходження на склад, інші відомості, такі як дати закінчення терміну придатності, які є важливими для керування запасами.

Для збору даних використовують різні методи, такі як сканери штрих кодів, безпілотні літальні апарати, камери, давачі.

Співробітники використовують мобільний сканер штрих-кодів, щоб сканувати деталі штрих-коду інвентарю, а потім вони автоматично завантажуються в програмне забезпечення складу. Автоматизований збір даних проводять з залученням безпілотників або AMP с давачів: RFID використовуються для сканування інвентаризації та оновлення даних у режимі реального часу.

Мітки радіочастотної ідентифікації (RFID) закріплюються або вбудовуються у предмети. Вони передають та отримують інформацію за допомогою читача та антени.

Дані RFID дозволяють відстежувати товари, контролювати обладнання, аналізувати умови транспортування та виявляти неефективність операцій без ручної втручання. В результаті покращується видимість, зменшуються витрати на роботу, збільшується точність та швидкість.

Успіх автоматизації операцій в складському господарстві вимагає постійного зв'язку між кількома елементами в центрах дистрибуції. IoT зв'язує всі різні пристрої та системи в екосистемі складу та відстежує все, щоб збільшити ефективність роботи. Це технологія, яка забезпечує синхронну роботу роботизованих систем, AGV, AMR, ERP та WMS. Реалізуючи пристрої IoT, менеджери складів можуть контролювати та координувати все з одного пункту взаємодії – від відвантаження, рівнів виробництва, запасів, доставок тощо.

ВИСНОВКИ. Отже, можна вважати, що логістика є перспективним напрямком розвитку для впровадження автоматизованих та автоматичних систем. При правильному та раціональному використанні інструментів автоматизації перевезень, зберігання, складування та зберігання, автоматизовані логістичні системи можуть працювати безперебійно, а їх обслуговування набагато дешевше.

Прийнято рішення продовжувати дослідження у напрямку промислової автоматизації для удосконалення процесів логістики, направляючи вектор на підвищення точності та ступеню автоматизації, шляхом використання та використання кіберфізичних підходів.

ЛІТЕРАТУРА

1. Євсєєв В. В. Методи та моделі кібер-фізичного керування процесами в організаційно-технічних виробничих об'єктах: автореф. дис. ... д-ра техн. наук : 05.13.07 "Автоматизація процесів керування" / В. В. Євсєєв ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2021. – 45 с.
2. Невлюдов І. Ш., і ін. (2019) Трансфер технологій у сучасній науці, освіті та виробництві в умовах четвертої промислової революції «ІНДУСТРІЯ 4.0» / Невлюдов І. Ш., Чала О. О., Олександров Ю. М. // Сучасний рух науки: тези доп. VIII міжнародної науково-практичної інтернет-конференції, 3-4 жовтня 2019 р. – Дніпро, 2, 604-608.
3. Чала, О., Сливка, А. (2023) Рівні засобів ПоТ в інформаційних технологіях. Виробництво & Мехатронні Системи: матеріали VII Міжнародної конференції, Харків, С. 51-60.
4. Шостенко С. С. Архітектура програмного забезпечення для супроводження автоматизованих систем оповіщення на виробництві / С. С. Шостенко, О. О. Чала // Виробництво & Мехатронні Системи 2022 : зб. тез. доп. VI-ої Міжнар. конф., 21-22 жовтня 2022 р. – Харків, 2022. – С. 115-117.
5. Vladyslav Basiuk, Svitlana Maksymova, Olena Chala, & Olha Miliutina. (2023). Mobile Robot Position Determining Using Odometry Method. Multidisciplinary Journal of Science and Technology, 3(3), 227–234. Retrieved from <https://mjstjournal.com/index.php/mjst/article/view/224>
6. Nevliudov Igor, Maksymova Svitlana, Chala Olena, Bronnikov Artem, & Vzhesnievskyi Maksym. (2023). Automated Logistics Processes Improvement in Logistics Facilities. Multidisciplinary Journal of Science and Technology, 3(3), 157–170. Retrieved from <https://mjstjournal.com/index.php/mjst/article/view/202>
7. Nevliudov, I., Vzhesnievskyi, M., Romashov, Y. і Chala, O. (2023) «Математичне моделювання мехатронних шатлів як об'єктів автоматизації для багаторівневих систем внутрішньоскладської логістики», Сучасний стан наукових досліджень та технологій в промисловості, (4 (26), с. 135–144. doi: 10.30837/ITSSI.2023.26.135.
8. Невлюдов І.Ш., Демська Н.П., Чала О.О., Демська А.І. (2018). Групове управління гнучкими виробничими системами у виготовленні МЕМС виробів. Міжнародна науково-практична конференція «Математичне моделювання процесів в економіці та управлінні проектами і програмами (ММП2018)», Коблево, 10-14 вересня 2018 р. Харків: ХНУРЕ, 101 -103

9. Lighting Control Module Development / Y. Vizir, O. Chala, S. Maksymova, Ahmad Alkhalaileh // Journal of Universal Science Research. – 2023. – № 1(12). – P. 645–657. URI <https://openarchive.nure.ua/handle/document/27446>
10. Automated System Development for the Printed Circuit Boards Optical Inspection Using Machine Learning Methods / I. Nevliudov, I. Botsman, O. Chala, K. Khrustalev // INFORMATION SYSTEMS AND TECHNOLOGIES (IST'2021) : proceedings of the 10-th International Scientific and Technical Conference, September 13-19, Odessa, 2021. – P. 234-238. URI <https://openarchive.nure.ua/handle/document/18350>
11. Nevliudov, I., Vzhesnievskyi, M., Romashov, Y., & Chala, O. (2023). Mathematical modeling of mechatronic shuttles as automation objects for multilevel systems of intra-warehouse logistics. INNOVATIVE TECHNOLOGIES AND SCIENTIFIC SOLUTIONS FOR INDUSTRIES, (4(26), 135–144. <https://doi.org/10.30837/ITSSI.2023.26.135>
12. O. Chala, A. Bronnikov, N. Igor and D. Mospan, "The Use of Neural Networks for the Technological Objects Recognition Tasks in Computer-Integrated Manufacturing," 2022 IEEE 4th International Conference on Modern Electrical and Energy System (MEES), Kremenchuk, Ukraine, 2022, pp. 1-5, doi: 10.1109/MEES58014.2022.10005750.
13. Гіль А. Промислові інтерфейси та протоколи передачі даних інтегрованих систем для автоматизованого управління в умовах Industry 4.0 / А. Гіль, О. Чала, О. Филипенко // Виробництво & Мехатронні Системи 2021 : матеріали V-ої Міжнар. конф., 21-22 жовтня 2021 р. - Харків, 2021. - С. 127-30. <https://openarchive.nure.ua/handle/document/18375>

ДОДАТОК Б

Код програми

```

//CommandRecieverRunnable.h
#pragma once

#include "CoreMinimal.h"
#include "HAL/Runnable.h"
#include "Networking.h" // для FIPv4Address, FInternetAddr
#include "Sockets.h" // для FSocket
#include "SocketSubsystem.h" // для ISocketSubsystem
#include "Templates/Function.h" // для TFunction

/**
 * FRunnable, який у своєму потоці:
 * 1) читає 8-байтний заголовок ("RESL"+uint32 довжини),
 * 2) читає payload заданого розміру,
 * 3) парсить ASCII-рядок "BoxID,VolW,FlapIndex",
 * 4) викликає колбек OnReceive(BoxID, VolumetricWeight, FlapIndex).
 */
class FCommandReceiverRunnable : public FRunnable
{
public:
    // Сигнатура колбека: ID коробки, об'ємна вага, індекс заслонки
    using FOnReceive = TFunction<void(int32 BoxID, float VolumetricWeight,
int32 FlapIndex)>;

    FCommandReceiverRunnable(
        const FString& InIP,
        int32 InPort,
        FOnReceive InCallback
    );
    virtual ~FCommandReceiverRunnable() override;

    /** Точка входу потоку */
    virtual uint32 Run() override;
    /** Зупиняє цикл прийому */
    virtual void Stop() override;

private:
    /** Читає рівно BytesToRead байт або повертає false */
    bool RecvAll(FSocket* Sock, void* Data, int32 BytesToRead);

    FString ServerIP;
    int32 ServerPort;
    FOnReceive OnReceive;
    FThreadSafeBool bStop;
};

//CommandRecieverRunnable.cpp
#include "CommandReceiverRunnable.h"
#include "Logging/LogMacros.h"
#include "HAL/PlatformProcess.h"

```

```

FCommandReceiverRunnable::FCommandReceiverRunnable(
    const FString& InIP,
    int32          InPort,
    FOnReceive     InCallback
)
    : ServerIP(InIP)
    , ServerPort(InPort)
    , OnReceive(MoveTemp(InCallback))
    , bStop(false)
{
}

FCommandReceiverRunnable::~FCommandReceiverRunnable() = default;

uint32 FCommandReceiverRunnable::Run()
{
    // 1) створюємо та підключаємо сокет
    ISocketSubsystem* Subsys =
ISocketSubsystem::Get(PLATFORM_SOCKETSUBSYSTEM);
    TSharedPtr<FSocket> Socket(Subsys->CreateSocket(NAME_Stream,
TEXT("CmdRecv"), false));

    FIPv4Address Addr;
    if (!FIPv4Address::Parse(ServerIP, Addr))
    {
        UE_LOG(LogTemp, Error, TEXT("Runnable: некоректна IP '%s'"),
*ServerIP);
        return 0;
    }

    TSharedRef<FInternetAddr> Endpoint = Subsys->CreateInternetAddr();
    Endpoint->SetIp(Addr.Value);
    Endpoint->SetPort(ServerPort);

    if (!Socket.IsValid() || !Socket->Connect(*Endpoint))
    {
        UE_LOG(LogTemp, Error, TEXT("Runnable: не вдалось підключитися до
%s:%d"), *ServerIP, ServerPort);
        return 0;
    }

    // 2) головний цикл прийому
    while (!bStop)
    {
        UE_LOG(LogTemp, Verbose, TEXT("[Runnable] очікування заголовка..."));

        // 2.1) читаємо 8 байт заголовка
        uint8 Header[8];
        if (!RecvAll(Socket.Get(), Header, 8))
        {
            UE_LOG(LogTemp, Warning, TEXT("[Runnable] не прочитано
заголовок"));
            break;
        }

        // 2.2) формуємо тег із перших 4 байт
        TCHAR Tag[5];
        for (int i = 0; i < 4; ++i) Tag[i] = TCHAR(Header[i]);
        Tag[4] = 0;
    }
}

```

```

// 2.3) дістаємо довжину payload
uint32 PayloadSize = *reinterpret_cast<uint32*>(Header + 4);

UE_LOG(LogTemp, Verbose,
    TEXT("[Runnable] заголовок '%s', розмір %u"), Tag, PayloadSize);

// 2.4) якщо не наша мітка – пропускаємо
if (FCString::Strcmp(Tag, TEXT("RESL")) != 0)
{
    UE_LOG(LogTemp, Verbose, TEXT("[Runnable] тег %s ігнорується"),
Tag);
    continue;
}

// 2.5) читаємо payload
TArray<uint8> Payload;
Payload.SetNumUninitialized(PayloadSize + 1);
if (!RecvAll(Socket.Get(), Payload.GetData(), PayloadSize))
{
    UE_LOG(LogTemp, Warning, TEXT("[Runnable] не прочитано
payload"));
    break;
}
Payload[PayloadSize] = 0;

// 2.6) конвертуємо в FString і парсимо "BoxID,VolW,FlapIndex"
FString Cmd = FString(UTF8_TO_TCHAR(reinterpret_cast<const
ANSICHAR*>(Payload.GetData())));
UE_LOG(LogTemp, Verbose, TEXT("[Runnable] отримано cmd = \"%s\"",
*Cmd);

TArray<FString> Parts;
Cmd.ParseIntoArray(Parts, TEXT(","), true);
if (Parts.Num() >= 3)
{
    int32 BoxID = FCString::Atoi(*Parts[0]);
    float VolumetricWeight = FCString::Atof(*Parts[1]);
    int32 FlapIndex = FCString::Atoi(*Parts[2]);
    UE_LOG(LogTemp, Log,
        TEXT("[Runnable] парсинг: ID=%d, VolW=%.2f, Flap=%d"),
        BoxID, VolumetricWeight, FlapIndex);

    // викликаємо коллбек → AsyncTask у GameThread → Blueprint
    OnReceive(BoxID, VolumetricWeight, FlapIndex);
}
else
{
    UE_LOG(LogTemp, Warning, TEXT("[Runnable] невірний формат cmd"));
}
}

Socket->Close();
return 0;
}

void FCommandReceiverRunnable::Stop()
{
    bStop = true;
}

```

```

bool FCommandReceiverRunnable::RecvAll(FSocket* Sock, void* Data, int32
BytesToRead)
{
    int32 TotalRead = 0;
    while (TotalRead < BytesToRead)
    {
        int32 ThisRead = 0;
        if (!Sock->Recv(reinterpret_cast<uint8*>(Data) + TotalRead,
            BytesToRead - TotalRead,
            ThisRead) ||
            ThisRead <= 0)
        {
            return false;
        }
        TotalRead += ThisRead;
    }
    return true;
}

// CommandReceiverComponent.h

#pragma once

#include "CoreMinimal.h"
#include "Components/ActorComponent.h"
#include "CommandReceiverRunnable.h"
#include "CommandReceiverComponent.generated.h"

// Делегат, що сповіщає Blueprints про нову команду сортування:
//   BoxID – ідентифікатор коробки
//   VolumetricWeight – об'ємна вага в кілограмах
//   FlapIndex – індекс заслонки (0..3)
DECLARE_DYNAMIC_MULTICAST_DELEGATE_ThreeParams(
    FOnCommandReceivedSignature,
    int32, BoxID,
    float, VolumetricWeight,
    int32, FlapIndex
);

/**
 * Компонент, який запускає FCommandReceiverRunnable
 * та через Blueprint-делегат передає вгору кожну отриману команду.
 */
UCLASS(ClassGroup = (Custom), meta = (BlueprintSpawnableComponent))
class OPENCVTEST_API UCommandReceiverComponent : public UActorComponent
{
    GENERATED_BODY()

public:
    UCommandReceiverComponent();

    /** Blueprint-делегат: підпишіться на нього, щоб отримувати BoxID, VolW
i FlapIndex */
    UPROPERTY(BlueprintAssignable, Category = "Sorting")
    FOnCommandReceivedSignature OnCommandReceived;

protected:
    virtual void BeginPlay() override;
    virtual void EndPlay(const EEndPlayReason::Type EndPlayReason) override;

```

```

private:
    /** Власне FRunnable, що слухає TCP і парсить команди */
    FCommandReceiverRunnable* ReceiverRunnable;

    /** Потік, у якому виконується ReceiverRunnable */
    FRunnableThread* ReceiverThread;
};

// CommandReceiverComponent.cpp

#include "CommandReceiverComponent.h"
#include "Async/Async.h"
#include "Engine/Engine.h"

UCommandReceiverComponent::UCommandReceiverComponent()
    : ReceiverRunnable(nullptr)
    , ReceiverThread(nullptr)
{
    // Відключаємо тик, бо всі операції відбуваються в окремому потоці
    PrimaryComponentTick.bCanEverTick = false;
}

void UCommandReceiverComponent::BeginPlay()
{
    Super::BeginPlay();

    UE_LOG(LogTemp, Warning, TEXT(">>> CommandReceiverComponent::BeginPlay()
запущено <<<"));

    // Налаштування TCP-клієнта
    const FString ServerIP = TEXT("127.0.0.1");
    const int32 ServerPort = 5006;

    // Лямбда, що викликається усередині FRunnable у момент отримання команди
    auto OnRecvLambda = [this](int32 BoxID, float VolumetricWeight, int32
FlapIndex)
    {
        // Переносимо виклик у головний потік, щоб безпечно працювати з
делегатом
        AsyncTask(ENamedThreads::GameThread, [this, BoxID,
VolumetricWeight, FlapIndex]()
        {
            UE_LOG(LogTemp, Log,
                TEXT("CmdReceiver → Broadcast: ID=%d, VolW=%.2f кг,
Flap=%d"),
                BoxID, VolumetricWeight, FlapIndex);

            // Розсилаємо подію всім підписникам (Blueprint-скриптам)
            OnCommandReceived.Broadcast(BoxID, VolumetricWeight,
FlapIndex);
        });
    });

    // Створюємо FRunnable та запускаємо його у новому потоці
    ReceiverRunnable = new FCommandReceiverRunnable(ServerIP, ServerPort,
MoveTemp(OnRecvLambda));
    ReceiverThread = FRunnableThread::Create(
        ReceiverRunnable,

```

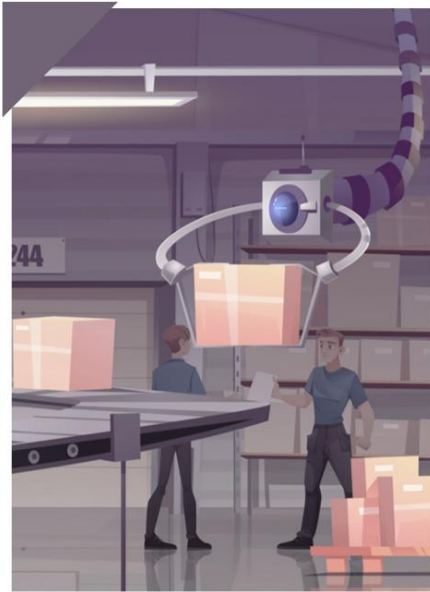
```
        TEXT("CommandReceiverThread"),
        0, // розмір стека за замовчуванням
        TPri_BelowNormal // пріоритет трохи нижчий за нормальний
    );
}

void UCommandReceiverComponent::EndPlay(const EEndPlayReason::Type
EndPlayReason)
{
    // Зупиняємо потік і звільняємо ресурси
    if (ReceiverRunnable)
    {
        ReceiverRunnable->Stop();
        ReceiverThread->WaitForCompletion();
        delete ReceiverThread;
        delete ReceiverRunnable;
        ReceiverThread = nullptr;
        ReceiverRunnable = nullptr;
    }

    Super::EndPlay(EndPlayReason);
}
```

ДОДАТОК В

Демонстраційний матеріал



Розроблення віртуального макету для моделювання автоматизованої лінії розподілу товарів в внутрішньоскладських логістичних системах

*Підготував ст. гр. АКТАКІТ-21-3
Краснопьоров М.Р.*

Керівник доц. каф. КІТАР Чала О.О.

1

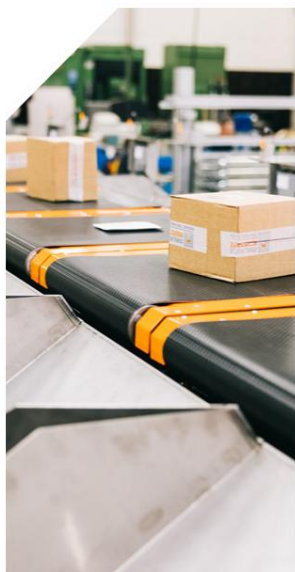
Вступ

Мета роботи – підвищення якості автоматизованого сортування в внутрішньоскладських логістичних системах шляхом проведення симуляції виробничих параметрів розподілу товарів при сортуванні.

Предмет розробки – віртуальний макет для моделювання автоматизованої лінії розподілу товарів.



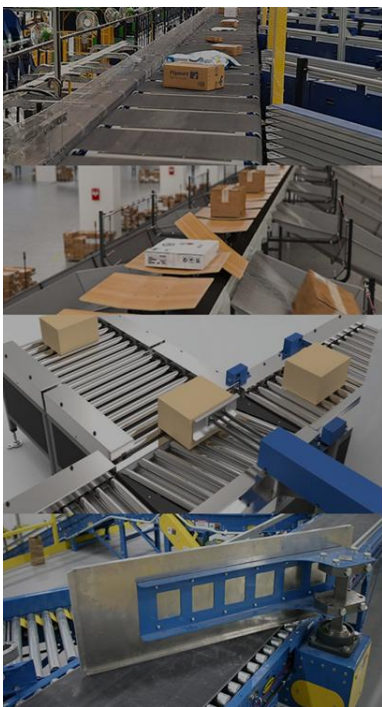
2



Актуальність

Тема є актуальною в умовах стрімкого зростання обсягів інтернет-торгівлі та потреби у швидкому й точному обробленні вантажів.

Використання віртуального макету дозволяє відпрацьовувати й оптимізувати алгоритми руху коробок, розташування конвеєрів і сенсорів без витрат на створення фізичних прототипів, що значно скорочує час і кошти на впровадження нових схем сортування.



Аналіз модулів розподілу товарів в автоматизованих лініях

Автоматизація розподілу(сортування) товарів – це комплекс заходів, технічних і програмних рішень спрямованих на оптимізацію процесу сортування.

Основними модулями на автоматизованій сортувальній лінії є: модуль подачі товарів у належній орієнтації і відстані між один одним (сингулятор), конвеєри для безперервного транспорту товарів по складу та сортувальний модуль.

Серед сортувальних модулів існують наступні види сортувальників, а саме: поперечний стрічковий сортувальник, сортувальники з перекидними лотками, сортувальники штовхачів або тягачів, лопаткові сортувальники, сортувальники з шарнірними колесами.

Аналіз системи



Зазвичай розподіл відбувається за місцем доставки або за габаритними розмірами коробки в якій знаходиться товар.

У випадку створення віртуального макету буде доцільним сортувати товари за об'ємною вагою їх коробок.

Для визначення об'ємної ваги необхідно знати габаритні розміри коробок товарів, тобто довжину ширину і висоту, а отже треба визначитись з методом визначення їх розмірів.

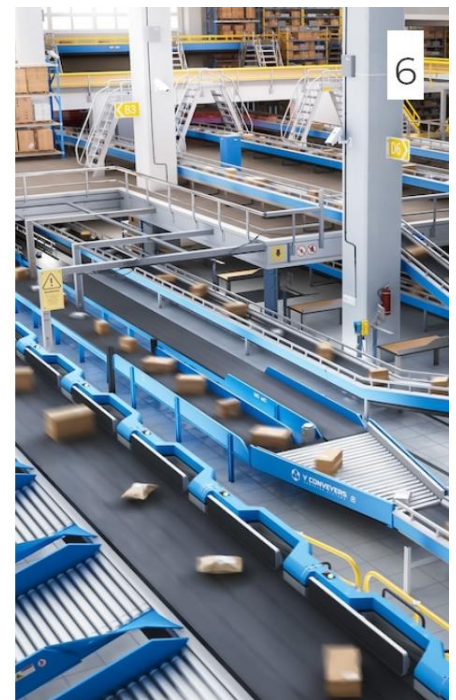
Для цього будемо використовувати метод з використанням Time-of-flight (ToF) камери. Time-of-flight камера – відеокамера, що формує так звану карту глибини.

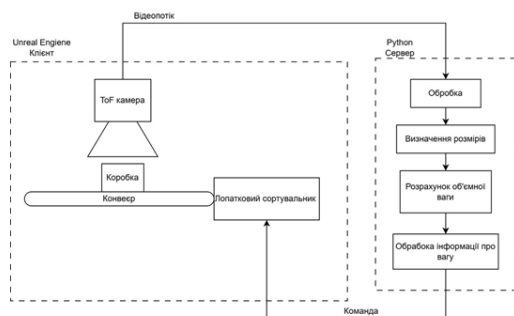
Аналіз системи

Для створення симуляції роботи конвеєра, ToF камери та сортувального модуля буде використовуватись рушій Unreal Engine 5.

Для обробки даних, які надходять з симульованої ToF камери, необхідно використовувати бібліотеки комп'ютерного зору, а оскільки маємо ще глибинне зображення то виникає потреба його опрацьовувати і визначати по ньому необхідні розміри.

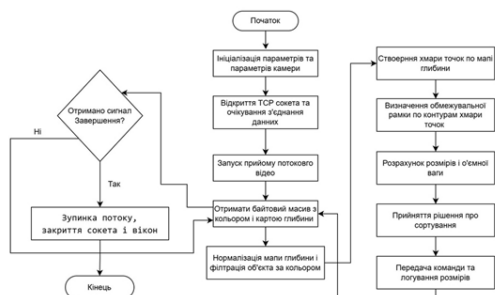
Для цього будемо використовувати сервер з мовою програмування Python, оскільки на ній зручно реалізована робота зі сторонніми бібліотеками, наприклад OpenCV і Open3D. Данні на сервер будуть передаватись через TCP/IP підключення





Структурна схема системи і алгоритм обробки зображення

Для більш чіткого розуміння про те як буде працювати система було розроблено структурну схему системи і алгоритм роботи Python програми для обробки зображення та визначення розмірів



Результат роботи

В результаті маємо віртуальний макет сортувальної лінії, що складеться з сцени у рушії Unreal Engine, та зовнішню логіку визначення об'ємної ваги та рішення щодо сортування на мові Python.

Експериментальним методом було підібрано наступні параметри системи:

- швидкість конвеєра: 3,5 м/с;
- відстань між товарами: 2,8 м.

За цими параметрами пропускною здатністю лінії є 4500 товарів на годину, а похибка сортування не перевищала 1%.

Демонстрація роботи

Висновки

В результаті виконання кваліфікаційної роботи було розроблено віртуальний макет для симуляції та налаштування лінії розподілу товарів в внутрішньоскладських логістичних системах, що дало можливість тестування та виключення помилок в роботі лінії та підвищити якість сортування. Проведена симуляція цифрового двійника лінії сортування показала, що при отриманих оптимальних показниках середня похибка сортування не перевищує 1%.

Також було досліджено систему на стійкість, і за критерієм Гурвіца система є стійкою.

Матеріали за тематикою роботи було опубліковано у статті.

Проведені дослідження відповідають цілям сталого розвитку (ЦСР), а саме: ЦСР 4, ЦСР 9.

