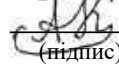


Міністерство освіти і науки України
Національний аерокосмічний університет
«Харківський авіаційний інститут»

Кафедра Інформаційних технологій проєктування (№ 105)

ЗАТВЕРДЖУЮ

Гарант освітньої програми


(підпис)

Олександр КАРАТАНОВ
(ім'я та ПРІЗВИЩЕ)

« 28 » 08 2025 р.

**СИЛАБУС ОBOB'ЯЗKОВОЇ
HABЧАЛЬНОЇ ДИСЦИПЛІНИ**

Програмне забезпечення роботизованих систем
(назва навчальної дисципліни)

Галузь знань: 12 Інформаційні технології

(шифр і найменування галузі знань)

Спеціальність: 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма: Інформаційні технології проєктування

(найменування освітньої програми)

Рівень вищої освіти: перший (бакалаврський)

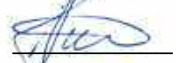
Силабус введено в дію з 01.09.2025

Харків – 2025 р.

Розробник (и): Каратанов О.В., доцент каф.105, к.т.н., доцент
(прізвище та ініціали, посада, науковий ступінь і вчене звання)


(підпис)

Артьомов І.В., асистент
(прізвище та ініціали, посада, науковий ступінь і вчене звання)


(підпис)

Силабус навчальної дисципліни розглянуто на засіданні кафедри _____

Інформаційних технологій проєктування

(назва кафедри)

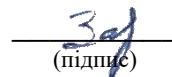
Протокол № 1 від « 28 » серпня 2025 р.

В.о. зав. кафедри к.т.н., доцент
(науковий ступінь і вчене звання)


(підпис)

Аліна АРТЬОМОВА
(ім'я та ПРІЗВИЩЕ)

Погоджено з представником здобувачів освіти:


(підпис)

Андрій ЗОЛОТАВІН
(ім'я та ПРІЗВИЩЕ)

1. Загальна інформація про викладача



ПІБ: Артёмов Ігор Володимирович

Посада: асистент

Науковий ступінь: -

Вчене звання: -

Перелік дисциплін, які викладає:

Цифрова схемотехніка

Системне програмне забезпечення

Міжкомп'ютерні комунікації

Програмне забезпечення роботизованих систем

Напрями наукових досліджень:

Моделювання маршрутів БпЛА

Контактна інформація:

i.artomov@khai.edu

2. Опис навчальної дисципліни

Форма здобуття освіти	Денна
Семестр	6-й, 7-й
Мова викладання	Українська
Тип дисципліни	Обов'язкова
Обсяг дисципліни: кредити ЄКТС/ кількість годин	<u>денна</u> : 6,5 кредитів ЄКТС / 195 годин (96 аудиторних, з яких: лекції – 32, лабораторні – 32, практичні - 32; СРЗ – 99);
Види навчальної діяльності	Лекції, лабораторні, самостійна робота
Види контролю	Поточний контроль, модульний контроль, семестровий контроль – іспит
Пререквізити	Міжкомп'ютерні комунікації
Кореквізити	Хмарні технології Управління проєктами і програмами Технологія захисту інформації Технологія розподілених систем та паралельних обчислень
Постреквізити	Технологія комп'ютерного проєктування

3. Мета та завдання навчальної дисципліни, переліки компетентностей та очікуваних результатів навчання

Мета – формування у студентів системного уявлення про архітектуру, принципи побудови та функціонування програмного забезпечення роботизованих систем, а також розвиток компетентностей у проектуванні, моделюванні, програмуванні та тестуванні програмних компонентів для керування роботами та робототехнічними комплексами з урахуванням вимог ефективності, надійності та безпеки.

Завдання – ознайомлення студентів із сучасними підходами до проектування, розробки та інтеграції програмного забезпечення для роботизованих систем на базі платформи ROS2. Формування розуміння архітектури ROS2, її основних компонентів, механізмів обміну повідомленнями, а також принципів побудови модульних і масштабованих робототехнічних додатків. Навчання основам розробки ROS2-вузлів, організації їх взаємодії через топіки, служби та дії, а також обробці сенсорних даних і керуванню апаратними компонентами робота. Пояснення методів симуляції та тестування роботизованих систем із застосуванням інструментів Gazebo і RViz2. Розвиток практичних навичок створення, налаштування та запуску ROS2-пакетів, написання програм на Python та C++, використання launch-файлів для автоматизації запуску систем. Ознайомлення з методами планування завдань, синхронізації та міжвузлової взаємодії в ROS2. Формування умінь документування, налагодження та тестування програмного забезпечення роботів, а також розуміння ролі симуляції для перевірки алгоритмів. Підвищення компетенцій у розробці комплексних робототехнічних систем із використанням відкритих технологій та стандартів.

Компетентності, які набуваються:

Інтегральна компетентність:

Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

Загальні компетентності (ЗК)

Після закінчення цієї програми здобувач освіти буде здатен:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК8. Здатність генерувати нові ідеї (креативність).

Спеціальні компетентності (СК)

Після закінчення цієї програми здобувач освіти буде здатен:

СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

Програмні результати навчання (ПРН):

ПР5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислювальних функцій.

4. Зміст навчальної дисципліни

МОДУЛЬ 1

Змістовний модуль 1. Основи програмного забезпечення роботизованих систем у ROS2.

Тема 1. Вступ до робототехніки. Архітектура та типи роботизованих систем.

У темі розглядаються базові поняття та компоненти сучасних роботизованих систем. Вивчаються класифікація роботів (мобільні, промислові, сервісні, автономні), архітектура роботизованої системи (сенсори, виконавчі механізми, контролери, програмне забезпечення), а також загальні підходи до програмного забезпечення роботів. Особливу увагу приділено взаємозв'язку між апаратним і програмним забезпеченням, поняттям реального часу, розподіленої обробки та необхідності стандартизації інтерфейсів. Формується системне бачення ПЗ як «мозку» роботизованої платформи.

Лекція 1 – Сучасні роботизовані системи: структура, типи, сфери застосування.

Лекція 2 – Програмне забезпечення роботів: завдання, підходи, вимоги.

Лабораторна робота 1 – Встановлення ROS2 та створення робочого середовища.

Самостійна робота: Опрацювання матеріалів лекцій і оформлення звіту до лабораторної роботи. Дослідити історію розвитку ROS (від ROS1 до ROS2): ключові етапи, виклики, рішення. Проаналізувати проблеми, які вирішує ПЗ в автономних роботах (навігація, планування, сприйняття, інтерфейси).

Тема 2. Основи ROS2: архітектура, середовище, базові компоненти.

Тема присвячена структурі та архітектурі **Robot Operating System 2 (ROS2)** – сучасного середовища розробки ПЗ для роботів. Розглядаються основні компоненти: вузли (nodes), топіки (topics), служби (services), дії (actions), параметри (parameters), трансформації (TF), а також нововведення ROS2 у порівнянні з ROS1: підтримка DDS (Data Distribution Service), покращена підтримка реального часу, безпеки та багатоплатформності. Вивчається модель обміну повідомленнями між вузлами та принципи публікації/підписки. Тема формує фундамент для практичної розробки ROS-додатків.

Лекція 3 – Порівняння ROS1 і ROS2. DDS, компоненти ROS 2, модель обміну даними.

Лекція 4 – Пакети, вузли, топіки, служби, параметри — основні концепції ROS2.

Лабораторна робота 2 – Робота з топіками (Publisher/Subscriber).

Самостійна робота: Опрацювання матеріалів лекцій і оформлення звіту до лабораторної роботи. Опрацювати архітектуру DDS (Data Distribution Service), описати її рівні та ролі в ROS2. Вивчити поняття «middleware abstraction» та як ROS2 реалізує незалежність від платформи.

Тема 3. Програмування в ROS2.

У цій темі студенти набувають практичних навичок створення вузлів у ROS 2 за допомогою Python (через бібліотеку `rclpy`). Вивчаються базові шаблони створення вузлів, механізм ініціалізації та завершення, робота з повідомленнями, створення видавців (`publisher`) та підписників (`subscriber`). Окрему увагу приділено параметрам вузлів, логуванню, таймерам, використанню стандартних типів повідомлень (наприклад, `std_msgs`) та роботі з командним рядком (`ros2 topic`, `ros2 node`). Ця тема дозволяє студентам впевнено реалізовувати власні вузли та інтегрувати їх у ROS2-проекти.

Лекція 5 – Розробка вузлів у ROS2 на Python (`rclpy`) та C++ (`rclcpp`).

Лекція 6 – Робота з топіками: публікація та підписка, обробка повідомлень.

Лабораторна робота 3 – Параметри вузлів та таймери.

Самостійна робота: Опрацювання матеріалів лекцій і виконання лабораторної роботи. Пояснити життєвий цикл ROS2-вузла: створення, ініціалізація, виконання, завершення. Розібрати типові типи повідомлень ROS2 (`std_msgs`, `geometry_msgs`, `sensor_msgs`) – короткі описи і структура.

Тема 4. Запуск та конфігурація ROS 2-проектів.

Тема присвячена ефективній організації запуску системи в ROS2. Студенти вивчають створення **launch-файлів**, які дозволяють автоматизовано запускати кілька вузлів одночасно. Розглядаються способи передачі параметрів вузлам, використання конфігураційних файлів YAML, організація умовного запуску компонентів, створення вкладених та багаторазових запусків. Також розглядаються утиліти *`colcon build`*, *`ros2 launch`*, *`ros2 param`*, *`ros2 lifecycle`*. Тема дає змогу переходити від простих тестів до повноцінної модульної ROS2-системи.

Лекція 7 – Launch-файли, параметри, середовище виконання.

Лабораторна робота 4 – Створення та використання launch-файлів

Самостійна робота: Опрацювання матеріалів лекцій і виконання лабораторної роботи. Вивчити концепцію launch-системи ROS2: як вона працює, що дозволяє, які має особливості. Проаналізувати роль параметризації ROS2-системи через YAML-файли. Вивчити принципи ієрархічного запуску: включення інших launch-файлів, умовний запуск вузлів. Дослідити поняття lifecycle-менеджменту вузлів (`managed nodes`, `states`, `transitions`).

Модульний контроль 1

Змістовний модуль 2. Прикладне ROS 2-програмування, сенсори та симуляція.

Тема 5. Служби та дії в ROS2.

Ця тема охоплює реалізацію **сервісів (services)** та **дій (actions)** – механізмів синхронної та асинхронної взаємодії між ROS2-вузлами. Вивчаються формати *.srv* та *.action* файлів, реалізація серверної та клієнтської частини, особливості обробки запитів та зворотного зв'язку. Пояснюється, коли доцільно використовувати служби (швидкий запит/відповідь), а коли – дії (тривалі операції, наприклад, рух до точки). Демонструються приклади застосування у навігації, керуванні маніпуляторами, автономному рішенні задач.

Лекція 8 – Створення служб (services): клієнт-серверна модель.

Лекція 9 – Створення дій (actions): асинхронне управління (наприклад, рух до цілі).

Лабораторна робота 5 – Реалізація сервісу та клієнта у ROS2.

Самостійна робота: Опрацювання матеріалів лекції і оформлення звіту до лабораторної роботи. Розмежувати призначення топіків, сервісів і дій у ROS2 (коли і що доцільно використовувати). Дослідити структуру *.srv* та *.action* файлів: які поля включає, як передається інформація. Вивчити принципи роботи клієнт-серверної взаємодії в ROS2, типи помилок та механізми обробки. Ознайомитися з патернами використання Action-серверів у задачах автономного пересування.

Тема 6. Інтеграція обладнання та обробка сенсорної інформації.

У темі розглядаються принципи підключення фізичних сенсорів до ROS2-систем, робота з драйверами та публікація даних у топіки. Охоплюються типові сенсори: IMU, LiDAR, ультразвукові датчики, GPS, енкодери. Розглядаються способи обробки вхідних даних у вузлах: фільтрація, перетворення, перевірка достовірності. Також аналізуються принципи побудови зворотного зв'язку для керування рухом. Особливу увагу приділено безперервному циклу «сприйняття – обробка – дія».

Лекція 10 – Підключення сенсорів (IMU, LiDAR, камери), драйвери, обробка повідомлень.

Лекція 11 – Актуатори та керування: PID-регулятори, реакція на сенсори.

Лабораторна робота 6 – Підключення сенсора (наприклад, IMU або LiDAR).

Самостійна робота: Опрацювання матеріалів лекції і оформлення звіту до лабораторної роботи. Розібрати, як відбувається публікація сенсорних даних у ROS2: типові формати, швидкість оновлення, обробка шуму. Теоретично пояснити фільтрацію сенсорних даних (moving average, Kalman filter, outlier removal). Вивчити повідомлення *sensor_msgs* – описати найважливіші типи повідомлень.

Тема 7. Симуляція роботів і візуалізація даних.

Тема присвячена роботі з камерами та зображеннями у ROS2. Розглядається інтеграція USB або симульованої камери, передача зображень через топіки, використання `sensor_msgs/Image`, підключення OpenCV через `cv_bridge`. Студенти вчаться обробляти зображення: виявляти кольорові об'єкти, контури, відслідковувати рух, створювати маски. Також розглядається основи машинного зору (computer vision) у контексті ROS2. Застосовується для задач орієнтації, навігації та взаємодії з довкіллям.

Лекція 12 – Симуляція в Gazebo: інтеграція з ROS2, запуск моделей.

Лекція 13 – Візуалізація в RViz2: налаштування, відображення топіків, навігація.

Лабораторна робота 7 – Обробка зображень з камери в ROS2.

Самостійна робота: Опрацювання матеріалів лекцій і оформлення звіту до лабораторної роботи. Вивчити концепцію потокового передавання зображень у ROS2 (інтерфейс `Image`, використання `cv_bridge`). Описати типові етапи обробки зображення у робототехніці: фільтрація, сегментація, виявлення контурів. Ознайомитися з прикладами використання OpenCV у ROS-проектах. Дослідити підходи до виявлення об'єктів у реальному часі на базі відеопотоку з камери.

Тема 8. Проєктування та тестування ROS 2-систем.

У завершальній темі студенти опановують симуляцію роботизованих систем у середовищі **Gazebo**, а також візуалізацію роботи вузлів та даних у **RViz2**. Розглядається запуск готових моделей (наприклад, `TurtleBot3`), підключення сенсорів у симуляції, виведення даних лазера, камери, TF-структури, векторів руху. Аналізується повноцінна ROS2-система як набір взаємодіючих вузлів: обробка вхідних даних, керування, реакція. Також проводиться тестування створених рішень – з використанням емуляції поведінки, параметризації, журналювання.

Лекція 14 – Від прототипу до готового рішення: тестування та налагодження ROS 2-систем.

Лабораторна робота 8 – Робота з симулятором Gazebo та RViz2.

Самостійна робота: Опрацювання матеріалів лекції і виконання лабораторної роботи. Вивчити структуру URDF-моделі робота (Unified Robot Description Format) – як описуються ланки, сенсори, приводи. Ознайомитися з RViz2 як інструментом візуалізації: можливості, типи даних, плагіни. Теоретично обґрунтувати важливість тестування та симуляції при розробці ROS2-систем.

Модульний контроль 2

Практичні заняття використовуються для групових консультацій по курсовому проєкту в семестрі 7.

5. Індивідуальні завдання

Курсовий проект: Розробка базової ROS2-системи взаємодії вузлів у симульованому середовищі

6. Методи навчання

Лекції проводяться з використанням демонстрації окремих прийомів роботи в середовищі обговорюваних програмних середовищ.

Лабораторні роботи виконуються з використанням ліцензійних примірників програмного забезпечення.

Самостійна робота включає підготовку до практичних занять, модульного контролю та іспиту, виконання позааудиторної частини індивідуального завдання з використанням навчально-методичної літератури та документації до програмного забезпечення.

7. Методи контролю

Контроль здійснюється згідно з «Положенням про модульно-рейтингову систему оцінювання знань студентів» Передбачено проведення поточного контролю відповідно до повноти, якості та своєчасності виконання практичних завдань; письмового модульного контролю; підсумкового контролю у вигляді письмового іспиту.

8. Критерії оцінювання та розподіл балів, які отримують здобувачі освіти

Таблиця 8.1 – Розподіл балів, які отримують здобувачі освіти

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Активність під час аудиторної роботи	0...1	7	0...7
Виконання і захист лабораторних робіт	0...5	4	0...20
Модульний контроль	0...25	1	0...25
Змістовний модуль 2			
Активність під час аудиторної роботи	0...1	7	0...7
Виконання і захист лабораторних/практичних робіт	0...5	4	0...20
Модульний контроль	0...25	1	0...25
Виконання і захист РГР (РР, РК)	0...10	1	0...10
Усього за семестр			0...100

Семестровий контроль (іспит) проводиться у разі відмови здобувача освіти від балів підсумкового контролю й за наявності допуску до іспиту. Під час складання семестрового іспиту здобувач освіти має можливість отримати максимум 100 балів.

Білет для іспиту складається з двох теоретичних питань. Максимальна кількість балів за кожне питання - 50 балів (сума – 100 балів).

Таблиця 8.2 – Розподіл балів, які отримують здобувачі освіти за виконання курсового проекту.

Пояснювальна записка	Ілюстративна частина	Захист роботи	Сума
0...25	0...25	0...50	100

Таблиця 8.3 – Шкали оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційний залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

Критерії оцінювання роботи здобувача освіти протягом семестру

Задовільно (60-74) – мати знання і вміння для забезпечення програмних результатів навчання. Виконати та здати розрахунково-графічну роботу. Написати дві модульні роботи. Знати основи архітектури та компонентної структури ROS2. Уміти створювати прості ROS2-пакети та запускати вузли. Уміти реалізовувати взаємодію між вузлами через топіки та сервіси. Уміти запускати симуляцію за допомогою базових launch-файлів. Уміти орієнтуватися в інструментах RViz2 та Gazebo для візуалізації та відлагодження. Уміти описати загальну структуру роботизованої системи та її програмного забезпечення. Уміти працювати з базовими командами ROS2 CLI та командним рядком Linux.

Добре (75-89) – мати знання, уміння й навички для забезпечення програмних результатів навчання. Виконати та здати розрахунково-графічну роботу. Написати дві модульні роботи. Додатково до вимог, які визначено для отримання задовільної оцінки: Уміти створювати багатовузлові ROS2-системи із взаємодією через топіки, сервіси й дії. Уміти налаштовувати параметри вузлів, структурувати проєкт і працювати з launch.py. Знати принципи QoS-політик у ROS2 та їх вплив на передачу даних. Уміти реалізовувати базові функції навігації, обробки сенсорних даних або візуалізації. Уміти адаптувати ROS2-систему до типових задач (наприклад, мобільна платформа або логістичний робот). Знати принципи тестування та

налагодження ROS2-додатків. Уміти застосовувати отримані знання до реалізації проєктного рішення в межах лабораторної чи розрахунково-графічної роботи.

Відмінно (90-100) – мати знання, уміння й навички, що дадуть змогу самостійно, вільно та обґрунтовано відповідати на будь-які питання щодо проєктування програмного забезпечення роботизованих систем з урахуванням інженерних, технічних, безпекових, економічних та етичних аспектів. Виконати та здати розрахунково-графічну роботу. Написати дві модульні роботи. Уміти самостійно проєктувати повноцінну ROS 2-систему, здатну працювати з кількома сенсорами, виконавчими механізмами та взаємодіяти з іншими цифровими системами. Уміти реалізовувати розширені сценарії з використанням дій, параметризації, запуску декількох вузлів у симуляції. Уміти оптимізувати структуру ROS2-додатків, використовувати публічні пакети, адаптувати їх під конкретні задачі. Уміти проводити комплексний аналіз програмної архітектури роботизованої системи та пропонувати шляхи її вдосконалення. Уміти готувати технічну документацію, формувати вимоги до системи та комунікувати з потенційними замовниками або користувачами. Уміти інтегрувати ROS2 із зовнішніми системами або обладнанням, розширювати можливості базових компонентів.

9. Політика навчального курсу

Відвідування занять. Регуляція пропусків. Інтерактивний характер курсу передбачає обов'язкове відвідування практичних занять. Здобувачі освіти, які за певних обставин не можуть відвідувати практичні заняття регулярно, повинні протягом тижня узгодити із викладачем графік індивідуального відпрацювання пропущених занять. Окремі пропущені заняття мають бути відпрацьовані на найближчій консультації протягом тижня після їх пропуску. Відпрацювання занять здійснюється усно у формі співбесіди за питаннями, визначеними планом заняття. В окремих випадках дозволяється письмове відпрацювання пропущених занять шляхом виконання індивідуального письмового завдання.

Дотримання вимог академічної доброчесності здобувачами освіти під час вивчення навчальної дисципліни. Під час вивчення навчальної дисципліни здобувачі освіти мають дотримуватися загальноприйнятих морально-етичних норм і правил поведінки, вимог академічної доброчесності, передбачених Положенням про академічну доброчесність Національного аерокосмічного університету «Харківський авіаційний інститут» (<https://khai.edu/assets/files/polozhennya/polozhennya-pro-akademichnu-dobrochesnist.pdf>). Очікується, що роботи здобувачів освіти будуть їх оригінальними дослідженнями або міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших здобувачів освіти становлять, але не обмежують, приклади можливої

академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі здобувача освіти є підставою для її незарахування викладачем незалежно від масштабів плагіату чи обману.

Вирішення конфліктів. Порядок і процедури врегулювання конфліктів, пов'язаних із корупційними діями, зіткненням інтересів, різними формами дискримінації, сексуальними домаганнями, міжособистісними стосунками та іншими ситуаціями, що можуть виникнути під час навчання, а також правила етичної поведінки регламентуються Кодексом етичної поведінки в Національному аерокосмічному університеті «Харківський авіаційний інститут» (<https://khai.edu.ua/university/normativna-baza/ustanovchi-dokumenti/kodeks-etichnoi-povedinki/>).

10. Методичне забезпечення

<https://mentor.khai.edu/course/view.php?id=9290>

11. Рекомендована література

Базова

1. ROS2 Documentation. Open Robotics. *(Офіційна документація ROS2: архітектура, розробка, туторіали)*. <https://docs.ros.org/en/humble/index.html>
2. ROS2 Tutorials. Open Robotics. *(Покрокові інструкції: створення вузлів, пакетів, робота з сервісами, топіками тощо)* <https://docs.ros.org/en/humble/Tutorials.html>
3. Rico, F.M. (2022). A Concise Introduction to Robot Programming with ROS2. *(Посібник для початківців: приклади, код, структура ROS 2-проектів)* GitHub репозиторій книги: https://github.com/fmrico/book_ros2
4. AlejandroBC. ROS2 Tutorial Book *(Збірка лекцій, конспектів та вправ з ROS 2, ліцензія MIT)*. <https://github.com/AlejandroBC/ros2-tutorial-book>

Допоміжна

1. ROS-Industrial Training Materials (Fraunhofer IPA). *(Слайди, приклади, вправи для промислових застосувань ROS2)* <https://discourse.ros.org/t/open-sourcing-our-ros2-industrial-training-material/21179>
2. Murilo's ROS2 Tutorial. *(Відкритий навчальний ресурс, ліцензія CC BY-NC-ND)*. <https://ros2-tutorial.readthedocs.io>
3. Maruyama, Y. (2022). ROS2 Design, Architecture, and Uses in the Wild. *(Аналіз архітектури ROS2 та її застосувань)*. <https://arxiv.org/abs/2211.07752>