



**НАВЧАЛЬНІ
ВИДАННЯ**

Опришко О.О., Ромашук О.М., Цицюрський Ю.Л.

Робототехнічні комплекси і системи

Навчальний посібник



**НАВЧАЛЬНІ
ВИДАННЯ**

Опришко О.О., Ромащук О.М., Цицюрський Ю.Л.

Робототехнічні комплекси і системи

Навчальний посібник

Київ

2025

УДК: 651.51:004.896(072)

*Рекомендовано до друку рішенням Вченої ради
Національного університету біоресурсів та природокористування України
(протокол № від)*

Рецензенти:

Сташук В.А., академік НААН, доктор технічних наук, професор (Київський аграрний університет Національної академії аграрних наук України, м.Київ);

Адамчук В.В., академік НААН, доктор технічних наук, професор (Інститут механіки та автоматики агропромислового виробництва НААН, смт. Глеваха);

Іващук В.В., доктор технічних наук, доцент (Національний університет біоресурсів та природокористування України, м.Київ)

Робототехнічні комплекси і системи: навчальний посібник / О. О. Опришко, О. М. Ромащук, Ю. Л. Цицюрський. – Київ : НУБіП України, 2025. - с.

ISBN

Посібник буде корисним студентам технічних спеціальностей, аспірантам та викладачам закладів вищої освіти.



*Пам'яті видатного вченого,
сумлінного педагога
та незабутнього наставника
академіка НАН та НААН України*

Івану Івановичу Мартиненку

присвячується ...

Цей посібник є результатом колективної праці в галузі робототехніки та автоматизації, і ми хочемо присвятити його нашому безцінному наставнику, вченому, педагогу та засновнику кафедри автоматизації сільськогосподарського виробництва, академіку Івану Івановичу Мартиненку.

Іван Іванович став знаковою фігурою в галузі автоматизації та робототехніки в Україні, і його внесок у розвиток цих наук має величезне значення для сучасної технічної освіти та наукової спільноти. У 2024 році ми відзначили 100-річчя з дня його народження, і це була чудова нагода ще раз згадати про його безцінний вклад у розвиток науки, освіти та практики. Саме Іван Іванович Мартиненко першим виділив автоматизацію аграрного виробництва як окремий науковий напрям, обґрунтувавши його специфіку необхідністю роботи з біотехнічними об'єктами, які поєднують технічну складову з біологічною мінливістю, що потребує особливих підходів до керування.

Іван Іванович умів дивитися в майбутнє. Ще кілька десятиріч тому він бачив перспективу, пов'язану з новим технологічним укладом, який включає широке впровадження роботів саме в аграрному виробництві. І сьогодні ми можемо з упевненістю сказати: без тих наукових основ, що були ним закладені, цей прорив був би неможливим.

Ми віримо, що Іван Іванович був би гордий бачити цей посібник як ще одну сходинку до розвитку новітніх технологій в Україні, і сподіваємось, що його наукова спадщина продовжить жити в роботах нових поколінь інженерів та науковців.

Це наш щирий внесок у вшанування пам'яті великого вченого і педагога, чия праця неоціненна для нашої галузі.

Олексій Опришко та Юрій Цицюрський

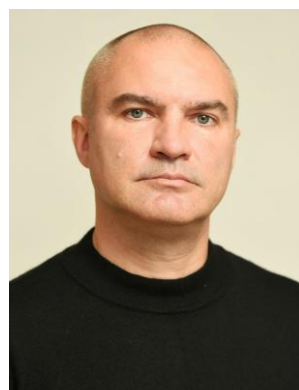


Опришко Олексій Олександрович,

кандидат технічних наук, в.о. завідувача кафедри автоматичної та робототехнічних систем ім. академіка І.І. Мартиненка ННІ енергетики, автоматичної і енергозбереження НУБіП України

(<https://nubip.edu.ua/node/2097>)

Фахівець з автоматичної та робототехніки, наукові інтереси: дистанційний моніторинг, робототехнічні системи і комплекси



Цицюрський Юрій Леонтійович,

асистент кафедри автоматичної та робототехнічних систем ім. академіка І.І. Мартиненка ННІ енергетики, автоматичної і енергозбереження НУБіП України

(<https://nubip.edu.ua/node/79404>)

Фахівець з автоматичної та біотехнологій, наукові інтереси: робототехнічні системи і комплекси, автоматизація біотехнічних об'єктів



Ромашук Олександр Миколайович

доктор філософії, асистент кафедри автоматичної та робототехнічних систем ім. акад. І.І. Мартиненка НУБіП України

(<https://nubip.edu.ua/node/1376>).

Фахівець з систем диспетчеризації та автоматизації (BMS), автоматичної та робототехніки, наукові інтереси: дослідження та створення роботизованих комплексів, автоматизація в енергетиці, мікроклімат, відновлювальні джерела енергії.

ЗМІСТ

Передмова

РОЗДІЛ 1 ОСНОВИ РОБОТОТЕХНІКИ

- 1.1 Загальні питання робототехніки
 - 1.1.1 Вступ
 - 1.1.2 Основні класи робототехнічних пристроїв
 - 1.1.3 Основи конструювання роботів
 - 1.1.4 Системний підхід щодо вирішення робототехнічних задач
 - 1.1.5 Аналогові роботи
 - 1.1.6 Висновки
- 1.2 Огляд стану світового ринку робототехніки для сільського господарства
 - 1.2.1 Автономні трактори
 - 1.2.2 Роботизовані пристрої та комплекси для рослинницької галузі
 - 1.2.3 Роботи для збору врожаю
 - 1.2.4 Роботи для виноградників і садів
 - 1.2.5 Самохідні роботи для моніторингу полів
 - 1.2.6 Роботи для тваринництва
 - 1.2.7 Робоферми та міські вертикальні ферми
 - 1.2.8 Поточний стан і перспективи ринку робототехніки
- 1.3 БПЛА
 - 1.3.1 Огляд світового ринку БПЛА та сфер їх застосування
 - 1.3.2 Світовий досвід використання БПЛА в тваринництві
 - 1.3.3 Світовий досвід застосування БПЛА в рослинництві
 - 1.3.4 Світовий досвід застосування БПЛА для біологічної боротьби зі шкідниками
 - 1.3.5 Переваги використання БПЛА в сільському господарстві
- 1.4 Питання для самоперевірки
- 1.5 Література до розділу 1

РОЗДІЛ 2 ЕЛЕМЕНТИ КОНСТРУКЦІЙ РОБОТОТЕХНІЧНИХ СИСТЕМ

- 2.1 Приводи роботів**
 - 2.1.1 Класифікація приводів
 - 2.1.2 Пневматичні приводи
 - 2.1.3 Гідравлічні приводи
 - 2.1.4 Електропривід промислових роботів
 - 2.1.5 Штучні м'язи
- 2.2 Спеціалізовані виконавчі механізми**
 - 2.2.1 Маніпулятори
 - 2.2.2 Захоплювачі

- 2.2.3 Аналіз ринку роботів маніпуляторів
- 2.3 Навігаційні системи робототехнічних систем**
- 2.3.1 Інерціальна навігаційна система
- 2.3.2 Системи супутникової навігації
 - 2.3.2.1 Спрощений алгоритм роботи GPS-приймача
 - 2.3.2.2 DGPS – Диференціальна GPS
 - 2.3.2.3 RTK – кінематика в реальному часі.
- 2.3.3 Системи автономної навігації
 - 2.3.3.1 Візуальні датчики (системи технічного зору)
 - 2.3.3.2 Лазерні та інфрачервоні датчики
 - 2.3.3.3 Радари
 - 2.3.3.4 Ультразвукові сенсори
- 2.4 Платформи мобільних робототехнічних систем**
- 2.4.1 Наземні платформи
- 2.4.2 Повітряні робототехнічні системи
 - 2.4.2.1 Конструкції планерів
 - 2.4.2.2 Силова установка БПЛА
 - 2.4.2.3 Готові комплекти програмних рішень для польотних контролером
- 2.5 Акумулятори**
- 2.5.1 Перспективні системи накопичення енергії
- 2.5.2 Саморазряд акумуляторних систем
- 2.5.3 Водневі технології та зворотні паливні елементи
- 2.6 Питання для самоперевірки
- 2.7 Література до розділу 2
- РОЗДІЛ 3 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ**
- 3.1 Операційні системи та програмні середовища робототехнічних систем
 - 3.1.1 ROS (Robot Operating System) — Операційна система для роботів
 - 3.1.2 Рівень файлової системи ROS
 - 3.1.3 Обчислювальний граф ROS
 - 3.1.4 Рівень спільноти ROS
- 3.2 Microsoft Robotics Developer Studio
- 3.3 Середовище NXT-G
- 3.4 Середовище LabVIEW
- 3.5 Програмування колаборативних роботів на прикладі мови програмування Melfa Basic. Робота із інтерфейсом програмування на комп'ютері
 - 3.5.1 Синтаксис мови програмування Melfa Basic
 - 3.5.1.1 Типи змінних
 - 3.5.1.2 Побудова алгоритмічних конструкцій
 - 3.5.2 Команди керування рухом
 - 3.5.3 Взаємодія із зовнішніми пристроями

- 3.6 Навчальний комплект LEGO Mindstorms NXT 2.0.
- 3.6.1 Складові частини комплекту Lego Mindstorms NXT 2.0. Технологія NXT
- 3.6.2 Середовище програмування NXT-G
 - 3.6.2.1 Профілі
 - 3.6.2.2 Створення і відкриття програм
 - 3.6.2.3 Доступ до програмних блоків
- 3.6.3 Перша програма
- 3.6.4 Програмування серводвигунів: команда Move
- 3.6.5 Стани і події. Давачі
- 3.6.6. Цикли
- 3.6.7 Розгалуження в NXT-G
- 3.6.8 Створення блоків користувача
- 3.6.9 Змінні та константи
- 3.6.10 Потоки
- 3.6.11 Керування переміщенням робота за допомогою системи із від'ємним зворотним зв'язком
- 3.7 Питання для самоперевірки
- 3.8 Література до розділу 3

1.1 Загальні питання робототехніки

1.1.1 Вступ.

За даними Вікіпедії Робот (чеськ. robot, від robota — «примусова праця») це автоматичний пристрій, призначений для виконання різноманітних механічних операцій, який діє за заздалегідь встановленою програмою. З таким твердженням можуть посперечатись програмісти оскільки існує поняття «програмний робот» чи RPA (Robotic Process Automation). RPA це технологія, що дозволяє швидко розробляти та запускати програмні "роботи", які можуть імітувати дії людини під час роботи з різними системами, додатками, електронною поштою, базами даних та іншим програмним забезпеченням.

Ідея створення роботів, повністю ідентичних за функціями людині, захоплює уми людства вже майже сто років. У прагматичному розумінні роботами називають будь-які відносно складні механізми, що керуються за допомогою обчислювальної техніки й здатні виконувати операції, складніші за ті, що виконуються звичайними верстатами, механізмами або типовими транспортними засобами.

Зокрема, механічні маніпулятори, які виконують збір складної продукції на конвеєрі, можуть належати до галузі робототехніки. Сюди ж відносять безпілотні транспортні засоби, здатні самостійно орієнтуватися у просторі й на місцевості, визначати та коригувати траєкторію руху залежно від ситуації й переміщуватися до потрібної мети, виконуючи поставлені завдання.

Роботів також часто пов'язують із поняттям штучного інтелекту. У цьому питанні серед фахівців виникли певні розбіжності. Деякі з них вважають, що термін «штучний інтелект» неправильно застосовується до комп'ютерів, які вирішують відносно складні завдання, але все ж не володіють інтелектом у повному розумінні цього слова. Інші ж вважають, що цей термін слід використовувати максимально широко, включаючи до нього майже всі відносно складні обчислювальні пристрої, а для позначення людиноподібного інтелекту пропонується залишити термін «штучний розум».

Щоб уникнути більшості непродуктивних дискусій, доцільно заздалегідь домовитися про термінологію та надалі чітко її дотримуватися.

На різноманітних молодіжних конкурсах часто в змаганнях роботів беруть участь механічні пристрої, які не є роботами, тоді як ті пристрої, що цілком відповідають визначенню робота, іноді так не називають.

Не кожен механічний пристрій, навіть якщо він зовні нагадує людину й здатен виконувати дії кінцівками, схожі на людські, можна вважати роботом. Щоб пристрій можна було назвати роботом, воно має вміти аналізувати принаймні деякі зовнішні сигнали, що несуть інформацію про навколишнє середовище, орієнтацію або місцезнаходження пристрою. Крім того, воно повинно діяти відповідно до отриманої інформації. Якщо ж усі функції управління сигналами датчиків виконує людина-оператор, такий пристрій зазвичай не називають роботом, а маніпулятором.

Водночас відносно новий напрямок, пов'язаний із розробкою екзоскелетів, безумовно, належить до робототехніки. Це не суперечить визначенню, оскільки

екзоскелети використовують датчики зусиль або інші сенсори, які зчитують сигнали з організму пілота. Ці сигнали формують команди для електромеханічних приводів, які приводять конструкцію в рух. Для підвищення точності та ефективності майже завжди у таких пристроях встановлюють датчики фактичного положення елементів конструкції, сигнали з яких використовуються для зворотного зв'язку, що забезпечує необхідну динамічну та статичну точність управління рухом.

Не є роботами різноманітні іграшки чи автомати, які виконують послідовність дій за заздалегідь встановленою програмою незалежно від положення, орієнтації чи зовнішніх обставин. Це лише механічні іграшки, навіть якщо вони створені із застосуванням обчислювальної техніки. Мікроконтролер у таких пристроях може генерувати керуючі сигнали виключно на основі внутрішніх тактових сигналів.

Особливо важливо зазначити: вимірювання часу не є використанням сенсорів зовнішньої реальності, а управління, засноване лише на даних часу, не можна вважати управлінням на основі аналізу навколишньої дійсності.

Розробка робота вимагає таких дій:

1. створення загальної концепції функціонування пристрою;
2. розробка окремих елементів та способу їх з'єднання;
3. проєктування системи, яка забезпечить приведення всіх елементів і рухомих частин у дію;
4. розробка набору сенсорів (датчиків);
5. визначення правил управління рухомими елементами на основі сигналів із датчиків;
6. реалізація цих правил шляхом створення алгоритмів управління;
7. створення керуючого пристрою із використанням цифрових технологій (зазвичай мікропроцесорної техніки або мікроконтролерів);
8. виготовлення всіх необхідних елементів конструкції та електроніки відповідно до розробленої концепції та їх інтеграція в єдину систему;
9. розробка програмного забезпечення та тестування пристрою в комплексі.

Цей перелік дій умовний і його можна доповнити за необхідністю.

1.1.2 Основні класи робототехнічних пристроїв

Роботи-андроїди

Ці роботи створені для максимально точного відтворення механічних рухових функцій людини (рис.1.1). У ідеалі передбачається, що такі роботи також зможуть відтворювати всі засоби сприйняття реальності, якими володіє людина, тобто всі п'ять органів чуття. У майбутньому планується, що такі роботи зможуть мати схожий інтелект, а отже, бути майже повністю ідентичними людині.

На вершині цього ідеалістичного уявлення знаходиться створіння, зовні невідмінне від людини, яке має окремі функції сприйняття і (або) управління, які перевершують людські можливості.



Рисунок 1.1 Роботи андроїди

Прикладом таких андроїдів є широко відомий термінатор із серії фантастичних фільмів режисера Кемерона. До цього класу також належать роботи, при створенні яких конструктори не прагнули зробити їх зовні схожими на людину, а лише намагалися відтворити функціональну ідентичність рухів.

Роботи, подібні до інших живих істот

До цієї групи належать різноманітні механічні пристрої, які рухаються подібно до інших живих істот у природі. Наприклад, існують роботи, що пересуваються як чотириногі тварини (коні, собаки тощо) або як комахи — шестиногі жуки, павуки, краби, багатоніжки тощо (рис.1.2).



Рисунок 1.2 Роботи кішка та мул відповідно

Роботи, які використовують техногенні способи пересування

До цієї категорії належать безпілотні пристрої, що рухаються, використовуючи всі можливі досягнення людської інженерної думки (рис.1.3). Сюди входять безпілотні літаки, вертольоти, дрони. Дронами називають зокрема, спеціальні вертольоти з більш ніж двома гвинтами, зазвичай чотирма або більше, що забезпечує їм підвищену маневровість. Вони стали дуже популярними в останні роки завдяки принципово меншій вартості, оскільки вони не потребують дорогої механічної частини – автомату перекосу, що використовується для руху в горизонтальній площині традиційними коптерами.



Рисунок 1.3 Робот на гусеничному ході, колесах зверху та коптерні дрони знизу відповідно

Сюди також належать різноманітні підводні та повітряні роботи, які використовують гребні гвинти, крила (що не здійснюють махових рухів), реактивну тягу, наземні пристрої з колісним і гусеничним ходом тощо. До цього типу роботів належать також балансуєчі роботи, які пересуваються на двох колесах, одному колесі або навіть на сферичній опорі, так звані «шароботи».

Екзоскелети

До цього класу роботів належать зовнішні конструкції, які надягаються на людину для посилення дій її кінцівок (рис. 1.4). У деяких випадках екзоскелети дозволяють значно розширити можливості людини для виконання специфічних завдань, таких як гасіння пожеж, перенесення важких вантажів, точне маніпулювання масивними предметами тощо. В інших випадках екзоскелети допомагають людям з інвалідністю здійснювати рухи, які здорова людина виконує без допомоги таких пристроїв.

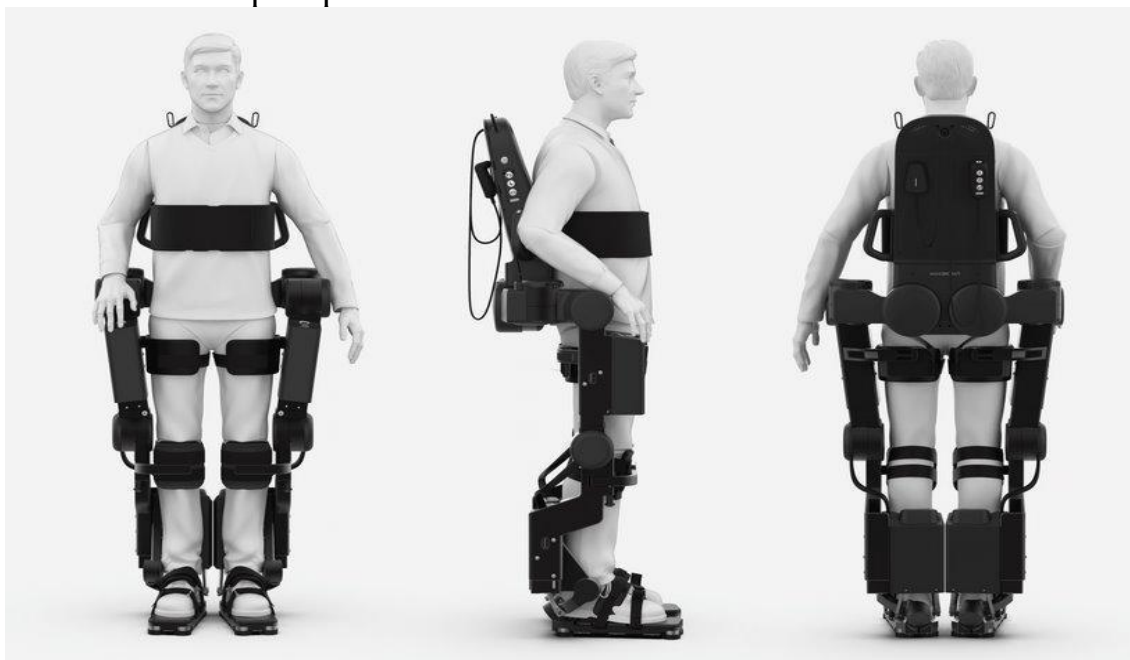


Рисунок 1.4 Медичний екзоскелет Atalante французької компанії Wandercraft для пацієнтів із хворими ногами. Використовується для реабілітаційних процедур

Протези кінцівок та інших органів.

Цей клас робототехнічних пристроїв має на меті відновлення втрачених функцій для осіб з інвалідністю. Це можуть бути штучні кисті рук або навіть ціла рука, штучні ноги або навіть дві ноги, а також роботи, які повністю замінюють обидві руки чи обидві ноги (рис. 1.5).



Рисунок 1.5 Штучна рука то нога.

Технічно поки що неможливо створити роботів, що замінюють людські очі, але значно поліпшити можливості зору можуть прості окуляри, які не відносяться до виробів робототехніки. Також до цієї категорії не належать слухові апарати. Однак, якщо буде створена «штучна шкіра», це, ймовірно, стане черговим досягненням робототехніки, так само як і «штучне око» (рис.1.6). Повідомляється, що це пристрій вже може повністю замінити людське око: «Argus II — новітня розробка американської компанії Second Sight Medical Products. Це перше у світі штучне біонічне око. Воно допоможе "відновити" зір навіть абсолютно сліпим людям. Найголовніше, що протягом тестування, яке тривало рік, всі випробувані отримали тільки позитивні результати. Робота над Argus II, як назвали його розробники, зайняла у вчених біля року досліджень. Імплантат був виготовлений в лабораторії американської компанії Second Sight Medical Products. Але винахід уже здобуло схвалення й у європейців. Пристрій працює так: клітини фоторецепторів перетворюють світло, що потрапляє в око, на імпульс, який передається в мозок через зоровий нерв. Для цього в сітківку біо-ока були вбудовані 60 електродів. Проведені клінічні випробування за участю 30 осіб віком від 28 до 77 років, всі з яких були абсолютно сліпими. Успіх кожного був строго індивідуальним: комусь це допомогло більше, комусь менше. Але головне, що вдалося з'ясувати розробникам, — око справді працює. Американці розраховують зробити своє винахід загальнодоступним. Наступним кроком розробників стане створення більш удосконаленої версії біонічного ока з більшою кількістю мікроімплантів та, як наслідок, з вищою роздільною здатністю».

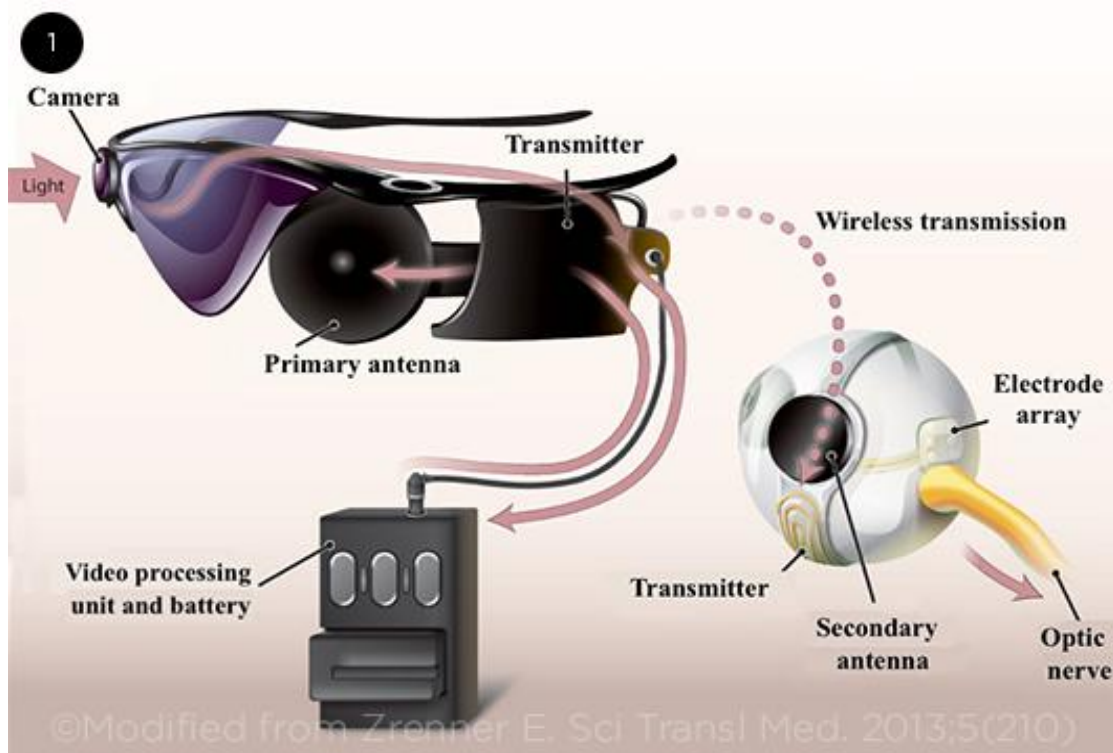


Рисунок 1.6 Компоненти системи ARGUS II. «Штучне око»

Роботи, які перевищують функціональні можливості людини

Повне дублювання можливостей людини не завжди є найкращим рішенням. Відомо, що людина не є досконалою в багатьох фізичних аспектах, навіть у порівнянні з іншими живими істотами. Наприклад, найкращим кольоровим зором володіє рак-богомол, найбільшою чутливістю — сова, ширшим кутом зору — павук із його вісьмома очима, а більш повним круговим оглядом, ймовірно, володіє стрекоза або муха, найкращим стереоскопічним зором — риба-молот тощо. Людина ж бачить лише те, що знаходиться в порівняно невеликому куті огляду, стереоскопічний зір досягається ще в більш обмеженому куті й залежить від орієнтації. Якщо поставити невелике вертикальне перешкоду поблизу переносиці, людина зможе побачити, що знаходиться за ним, завдяки наявності двох очей, але якщо перешкода буде розташована горизонтально, ця можливість зникає. Дві пари техногенних очей, що розташовані одна над одною, дозволили б отримати цей ефект навіть у разі горизонтального перешкоди. Кілька таких пар навколо голови дозволили б бачити ситуацію в усіх напрямках, а наявність таких пристроїв у вертикальному напрямку (наприклад, на спеціальному шоломі) дозволила б спостерігати за ситуацією зверху і, у разі небезпеки, уникнути її.

Подібні удосконалення можуть стосуватися й інших органів чуття. У цьому контексті можна виокремити два напрямки розвитку. По-перше, людина, яка виконує особливо складні або відповідальні роботи, могла б бути оснащена додатковою інформацією, яка передавалася б у тій формі, яку їй легше сприймати. Наприклад, відсутність очей на потилиці ускладнює успішне керування транспортом, але це компенсується наявністю дзеркал заднього виду, а в новітніх версіях ці функції ще ефективніше виконують відеокамери заднього виду. Такі додаткові камери допомагають успішніше паркуватися, рухатися заднім ходом, коли простір для маневрів обмежений. Подібні пристрої можуть значно покращити функціональні можливості людини поза транспортним засобом. Прикладом можуть бути окуляри нічного бачення, дзеркала або камери заднього виду на шоломі тощо.

Інший напрямок застосування такого підходу — оснащення безпілотних транспортних засобів додатковими датчиками для більш повного сприйняття простору, більш детального, ніж це робить пілот (рис. 1.7). Таке надлишкове сприйняття просторової картини дозволяє розробляти безпілотні транспортні засоби, які не лише змагатимуться з пілотованими щодо безпеки, а й значно перевищуватимуть їх можливості, оскільки могли б набагато ефективніше передбачати небезпеки, ніж людина за кермом.



Рис. 1.7. Прототип безпілотного таксі Tesla Cybercab, яка штатно не має руля керування.

1.1.3 Основи конструювання роботів

Перелік основних компонент робота

Будь-який робот зазвичай містить такі компоненти:

Механічна частина:

- Корпус;
- Засоби переміщення (колеса, гусенична тяга, пропелери чи інші "рухові механізми");
- Засоби виконання робіт (захвати, ковші, щітки, ріжучі інструменти, зварювальні апарати та інші маніпулятори);

Датчики:

- Відеокамери, лідари, оптронні пари та інші оптичні датчики;
- Ультразвукові датчики;
- Датчики рівноваги;
- Датчики швидкості і (або) кута повороту власних приводів;
- Датчики температури, тиску, електромагнітного поля та інші;

Електронна керуюча частина:

- Посилювачі сигналів датчиків;
- Засоби перетворення сигналів (аналогові);
- Засоби перетворення аналогових сигналів в цифрові;
- Засоби обробки цифрових сигналів;
- Засоби перетворення цифрових сигналів в аналогові;
- Посилювачі сигналів (драйвери) для керування механічними пристроями;

Усі зазначені компоненти розробляються різними виробниками, незалежно від розробників роботів, а також за їхнім замовленням. Тому при розробці нових роботів доцільно ознайомитись з найсучаснішими та ефективними компонентами для кожного з вузлів, і лише у разі відсутності готових рішень доцільно розробляти і виготовляти власні елементи.

На ранніх етапах найбільш простими деталями були колеса, постійного та змінного струму двигуни, крокові мотори, зубчаті передачі (редуктори, черв'ячні передачі) та подібні механізми для формування та перетворення механічних рухів. З цих елементів будувалися відносно неуклюжі механізми, які вже називали роботами, оскільки ентузіазм творців часто випереджав їх реальний технічний потенціал.

Якщо згадати фільм «Термінатор», де показано внутрішній пристрій штучної руки робота-андроїда з фантастичного майбутнього, всі суглоби в ній зроблені на основі підшипників, оскільки творці фільму саме так уявляли собі майбутні технічні можливості. Це так само примітивно, як уявлення ранніх фантастів, які вважали, що майбутні роботи будуть зроблені на радіоламах. Радіолампи відійшли в далеке минуле (за винятком кількох екзотичних сфер їхнього застосування), а розумні роботи досі ще не створені. Чим ближче технології підходять до фактичного створення найефективніших роботів, тим ясніше стає перевага вже існуючих рішень в живій природі над примітивними механізмами, створеними в двадцятому столітті за допомогою зубчатих передач і простих двигунів.

На сьогоднішній день одними з найактуальніших завдань є створення штучних м'язів, щупалець, присосок та інших технічних засобів, які використовують принципи формування руху в кращих представників живої природи. Період захоплення здібностями різних живих істот вже минув, і настав час створення технічних засобів, які використовують ці можливості. Тісне вивчення насіння реп'яха призвело до винаходу застібки «липучки», що кардинально змінила багато видів одягу та оснащення для туристів, військових, пожежних, альпіністів тощо. Аналіз лап комах та земноводних, здатних переміщатися по вертикальних поверхнях і навіть по стелі, дозволив створити аналогічні перчатки, які можуть прикріплюватися до вертикальних поверхонь, хоча їх застосування поки що не є широко поширеним.

Необхідно більш уважно розглядати найбільш ефективні способи переміщення, навіть якщо вони здаються нетрадиційними або на перший погляд їхнє копіювання може виглядати не доцільним. Глибше вивчення таких механізмів може відкрити потенціал для нових, більш ефективних технічних рішень. Особливої уваги заслуговують щупальця восьминога, тіло дощового черв'яка і змії, підшва равлика, лапи павука, мурашки, багатоніжки, мухи, гекона, кузнечика, кенгуру, краба, крила стрекози, метелика, орла, сови, летючої миші та інші.

На основі цього можна зробити висновок, що штучна рука, що має з'єднання на основі традиційних шарнірів, є пережитком минулого. Суглоби людини демонструють структуру, яка значно ефективніше відповідає завданням тих видів

рухів, для яких вони призначені. Раніше спрощення конструкцій призводило до зниження вартості, оскільки виготовлення надмірно складних деталей призводило до значного подорожчання технічного обладнання. Однак із розвитком різноманітних технологій, зокрема 3D-друку, коли необхідні деталі можна спроектувати за допомогою комп'ютера та просто роздрукувати на 3D-принтері, навіть найскладніші форми з'єднань вже не є складними або неоправданими. Навпаки, надмірне спрощення вже не є доцільним. Суглоби людини забезпечують свободу руху в необхідних напрямках і обмеження в тих, де це потрібно, що набагато ефективніше, ніж традиційні підшипники. Тому, якби автори фільму «Термінатор» використовували сучасні уявлення про напрямки розвитку робототехніки, штучна рука виглядала б не тільки схоже на реальну (в своїй штучній оболонці), а й по внутрішньому будовою, тобто її скелет був би фактично повністю ідентичний скелету людської руки (хоча це, ймовірно, було б менш видовишно).

Розвиток механічних засобів робототехніки відбувається паралельно в двох напрямках: це і ускладнення, і спрощення.

З одного боку, важливим завданням залишається розробка маніпуляторів, які все більше нагадують людську руку, а іноді навіть перевершують її за можливостями. З іншого боку, якщо, наприклад, маніпулятор має лише викручувати згорілі лампочки та викручувати нові, йому зовсім не обов'язково бути схожим на людську руку. Досить мати три надувних щупальця, які при підвищенні тиску обхоплюють предмет, а при зниженні тиску — відпускають його. Це одночасно вирішує проблему обмеження тиску на захоплюваний предмет і сприяє його збереженню. Такий маніпулятор не здавить скляну лампочку, на відміну від металевої руки з недостатнім контролем тиску.

Поєднання принципів телескопічних кінцівок, що дозволяють легко збільшувати або зменшувати довжину, з принципами гідравліки дає великі перспективи для розвитку роботизованих рук порівняно з людською. Така рука може значно подовжуватися чи вкорочуватися, що, зрозуміло, неможливо для людини. Цей приклад, а також приклад збільшення кута огляду та спектра сприйняття оптичних зображень, демонструють, що при створенні робота-андроїда зовсім не обов'язково копіювати можливості людини там, де їх можна легко перевершити. Однак досі є параметри, за якими технології не тільки не можуть перевершити людські можливості (або загальні можливості біологічних сенсорів), але й навіть не наблизилися до них. Наприклад, на сьогоднішній день технології не можуть створити такі ефективні аналізатори запахів, як сенсори молей або нюх у деяких ссавців. У цій сфері можливий розвиток шляхом поєднання біологічних можливостей із механічними пристроями, хоча на цей момент таких наукових розробок дуже мало. Першою такою роботою, ймовірно, є експерименти Гальвані, який використовував лягушачу лапку для детектування слабких електричних струмів, оскільки в його час ще не існувало механічних пристроїв з такою високою чутливістю. Це не означає, що, наприклад, можна використовувати вусики молі для

розпізнавання різних запахів — виявлення навіть найменших домішок у повітрі. Але такі можливості вже не слід повністю виключати, і, можливо, цей підхід приховує ще не досліджені потенціали.

Зворотний підхід, що полягає в оснащенні живих істот (переважно людини) технічними засобами для розширення сенсорних можливостей (сприйняття реальності) і кінематики (взаємодія з реальністю), вже активно розвивається. Люди отримують протези кінцівок, слухові апарати, пристрої нічного бачення, і вже йдеться про штучний зір тощо. Є ціла низка біостимуляторів, здатних забезпечити повноцінне життя людям, які без цих стимуляторів або втратили б багато функцій, або взагалі не могли б існувати.

Деякі тварини, які перебувають на службі у людини, також можуть бути оснащені додатковими технічними засобами. Повідомлялося про оснащення морських ссавців засобами боротьби з підводними човнами противника, існують нашийники для собак, які дозволяють мисливцеві визначати місцезнаходження тварини та навіть давати їй команди, а також засоби для відстеження міграцій рідкісних тварин — ссавців, птахів і риб. Для домашніх тварин і диких тварин встановлюють відеокамери для спостереження за їхньою діяльністю тощо. Однією з важливих особливостей цього підходу є можливість тривалого забезпечення харчування цих пристроїв, що обумовлено, по-перше, відносно низьким споживанням електроенергії цими пристроями, а по-друге, прогресом у розвитку джерел батарейного живлення.

1.1.4 Системний підхід щодо вирішення робототехнічних задач

Усі елементи робота повинні працювати з високою точністю, інакше робот не зможе вирішувати поставлені завдання. Забезпечення точності управління вирішується методами автоматичного управління, які вивчає теорія автоматичного управління (ТАУ). Ця теорія вирішує завдання тільки управління в контурі з негативним зворотним зв'язком, оскільки тільки використання такого контуру дозволяє забезпечити необхідну точність рухів навіть у випадку, коли на рухомий об'єкт впливають різні зовнішні збурюючі чинники, що вносять свій вклад у зусилля, які відповідають за подальше переміщення частин робота.

Робот можна уявити як велику систему окремих маніпуляторів, суглобів та інших рухомих пристроїв, кожне з яких повинно рухатися суворо відповідно до закладеної в ньому програми, причому ця програма може змінюватися в реальному часі досить швидко. У кожній з таких підсистем є інструкція, яка формується центральним процесором (аналог електронного мозку робота). Якщо ця інструкція безпосередньо впливає на привод, що здійснює рух цього елемента (суглоба), то рух майже завжди буде відрізнятися від необхідного. Навіть при використанні крокових двигунів фактичний поворот осі може відрізнятися від заданого через вплив зовнішніх факторів, а при використанні інших засобів впливу фактичний рух може суттєво відрізнятися від заданого. Тому навіть у крокових двигунах при вимогах до високої точності використовують вимірювання фактичного повороту осі,

наприклад, за допомогою інтерферометричної лазерної вимірювальної системи. Це фактичне положення осі використовується для коригування реального положення таким чином, щоб фактичне та істинне положення осі збіглися. В такій корекції і полягає суть негативного зворотного зв'язку.

Розглянемо гіпотетичний випадок, де окремий вузол робота є системою автоматичного управління, в якій об'єктом управління є вся механічна частина, а завданням є команда на зміну положення рухомої частини цього вузла. Регулятором виступає вся електронна частина, яка формує силові сигнали, подаються на виконавчу частину електромеханічного вузла (наприклад, на обмотку електродвигуна). Об'єднання об'єкта та регулятора такого вузла в систему вимагає аналізу їхніх математичних моделей та зв'язків між ними. На основі цього здійснюється розрахунок (синтез) регулятора та його реалізація.

Необхідно розрізняти замкнуті системи, про які йшлося раніше, від з'єднання елементів у ланцюзі. Замкнута система відрізняється введенням зворотного зв'язку, що створює петлі в ланцюзі проходження сигналу. Сигнали в ланцюгах без петель можна обчислювати послідовно від входу до виходу. Підключення наступних елементів не змінює попередніх сигналів. В ланцюгах з петлями такий підхід застосовувати неможливо. Потрібно аналізувати весь контур системи в цілому.

Однією з найважливіших властивостей замкнутої системи (яку необхідно забезпечити) є стійкість, тобто здатність залишатися в рівноважному стані або в його близькості за відсутності зовнішніх впливів. Якщо зовнішній вплив виводить таку систему з рівноваги, вона під дією внутрішніх сил, сформованих замкнутим контуром, повертається назад у вихідний рівноважний стан. Розімкнений ланцюг стабільний, якщо всі його елементи стабільні, і наявність хоча б одного нестабільного елемента робить ланцюг нестабільним. В замкнутих петлях такої закономірності немає. Система, що складається з нестабільних елементів, може бути стабільною, а система, що складається зі стабільних елементів, може бути нестабільною, але все може бути й інакше — тут немає прямої залежності стабільності системи в цілому від стабільності її складових елементів. Прикладом стабільної системи з нестабільних елементів є система з інтегральним регулятором. Прикладом нестабільної системи з стабільних елементів може бути система високого порядку (наприклад, фільтр шостого порядку) з великим коефіцієнтом посилення.

У системах автоматичного управління (САУ) реалізується принцип негативного зворотного зв'язку, тобто будь-яке відхилення керованої величини від бажаного значення зводиться до нуля за рахунок того, що воно посилюється з негативним знаком і подається назад у точку виникнення. Коефіцієнт передачі петлі, умовно розірваної в довільній точці, має бути великою від'ємною безрозмірною величиною. Цей принцип іноді називають принципом великого коефіцієнта.

Спеціально впроваджені в систему регулятори забезпечують її стабільність, а також потрібну швидкість реакції та точність роботи. Методи аналізу таких систем

і розробки регуляторів вивчаються в теорії автоматичного управління (ТАУ). Ці завдання завжди виникають при об'єднанні елементів в єдину систему.

Для кожної лінійної системи можна визначити одне рівноважне стан, тобто таке, яке система приймає за відсутності зовнішніх впливів (сигналів). Мірилом стійкості є здатність системи повертатися до цього стану після її відхилення через зовнішні перешкоди. Якщо відхилення, яке виникає в системі, спричиняє її рух у напрямку збільшення цього відхилення (тобто є позитивний зворотний зв'язок), можна очікувати, що така система не здатна зберігати рівновагу, оскільки навіть незначні зовнішні впливи призводять до значних змін у системі. Якщо відхилення від рівноваги викликає впливи, які повертають систему до початкового стану (тобто є негативний зворотний зв'язок), система стабільно утримується в цьому стані навіть за наявності значних зовнішніх перешкод.

Методи розробки регуляторів різноманітні, але найбільш ефективними на сьогодні є методи, що базуються на числовій оптимізації коефіцієнтів.

1.1.5 Аналогові роботи

Традиційно як промислові, так і інші типи роботів зазвичай створюються на основі схеми управління з центрального процесора, що функціонує за певним алгоритмом, збереженим у вигляді програми в пам'яті мікропроцесора чи контролера. Такий підхід до створення роботів доцільно використовувати в умовах, де робочі процеси є повторюваними й передбачуваними. Водночас впровадження робототехнічних систем стримується їх високою вартістю. Однак у практичному використанні є й приклади аналогових роботів.

У цьому розділі розглядається конструювання роботів, які не мають у своїй основі центрального процесорного пристрою. Подібні роботи отримали назву ВЕАМ-роботи, яку запропонував Марк Тілден. Особливістю цих роботів є їх функціонування на основі нейронних мереж, тобто вони працюють за принципом стимул-реакція. Стимульно-реактивні схеми, що застосовуються в ВЕАМ-роботах, також називають поведінково орієнтованими схемами, схемами нервової організації, нейронними мережами або схемами предикативної архітектури.

Сучасні підходи до вдосконалення аналогових робототехнічних систем, зокрема часткова заміна аналогових пристроїв цифровими, а також використання сучасних засобів аналогової електроніки, включаючи аналогові нейронні мережі, є актуальною темою досліджень сучасної науки.

На початку 1990-х років Марк Тілден запропонував інноваційний підхід до проектування роботів, розробивши концепцію ВЕАМ-роботів. ВЕАМ є аббревіатурою від англійських слів **Biology** (біологія), **Electronics** (електроніка), **Aesthetics** (естетика), **Mechanics** (механіка). Що ж об'єднує ці, на перший погляд, несумісні поняття?

- **Біологія.** Багато аспектів дизайну та функцій BEAM-роботів запозичено з живої природи; зазвичай вони не використовують такі механічні елементи, як підшипники чи колеса.
- **Електроніка.** Для визначення поведінки роботів використовуються прості аналогові схеми.
- **Естетика.** Робот має виглядати привабливо.
- **Механіка.** Механічні компоненти забезпечують пересування робота.

Концепція BEAM-роботів

Головною відмінною особливістю концепції, запропонованої Марком Тілденом, було створення роботів, здатних «жити самостійним життям» незалежно від людини.

У BEAM-роботах не використовуються мікропроцесори, натомість застосовуються прості аналогові схеми. Їх не потрібно програмувати, що дозволяє зменшити енергоспоживання та мінімізувати ризик відмови. Такі роботи мають простий дизайн, а їхня вартість є низькою.

BEAM-роботи часто імітують поведінку живих істот: комах, рептилій або простих ссавців. Ідея Тілдена полягала в розробці механічних «тварин», здатних існувати без допомоги людини, самостійно знаходити джерела енергії для підтримки своєї життєдіяльності, подібно до того, як живі істоти шукають їжу.

Три закони BEAM-роботів за Марком Тілденом:

1. **Робот має захищати своє існування за будь-яку ціну.** Більшість BEAM-роботів оснащені різними сенсорами, які дозволяють їм виявляти перешкоди. Наприклад, сенсори можуть бути виконані у вигляді механічних «вусиків» (як у комах або дрібних ссавців), які запобігають зіткненню робота з об'єктами й допомагають їх оминати.
2. **Робот має отримувати доступ до джерел енергії та підтримувати свою працездатність.** Використання традиційних акумуляторів у багатьох роботах є неефективним, оскільки це суттєво збільшує їхню масу. Крім того, у деяких конструкціях застосовуються потужні двигуни, що ускладнює та здорожчує проєкт. Це суперечить концепції BEAM, яка вимагає максимальної простоти й дешевизни. Тому BEAM-роботи зазвичай оснащуються фотоелектричними модулями (сонячними батареями), що забезпечують надійне та автономне живлення, дозволяючи їм тривалий час працювати без зовнішнього втручання.
3. **Робот має знаходити ефективне джерело енергії.** Функція пошуку джерел енергії в BEAM-роботах моделюється за аналогією з поведінкою простих живих істот, які постійно прагнуть покращити умови свого існування. У роботах використовуються фотоелементи, що виявляють рівень освітленості та керують рухом робота у напрямку до найяскравішого джерела світла, де він може отримати більше енергії для живлення своєї системи.

Принципи розробки ВЕАМ-роботів:

- Використовувати мінімальну кількість компонентів у конструкції, максимально спрощуючи модель.
- Застосовувати відновлювані джерела енергії (наприклад, сонячну енергію) для живлення системи.

Приклад аналогового роботу

Завдання роботу – пересуватися вздовж «чорної лінії» (рис.1.8).

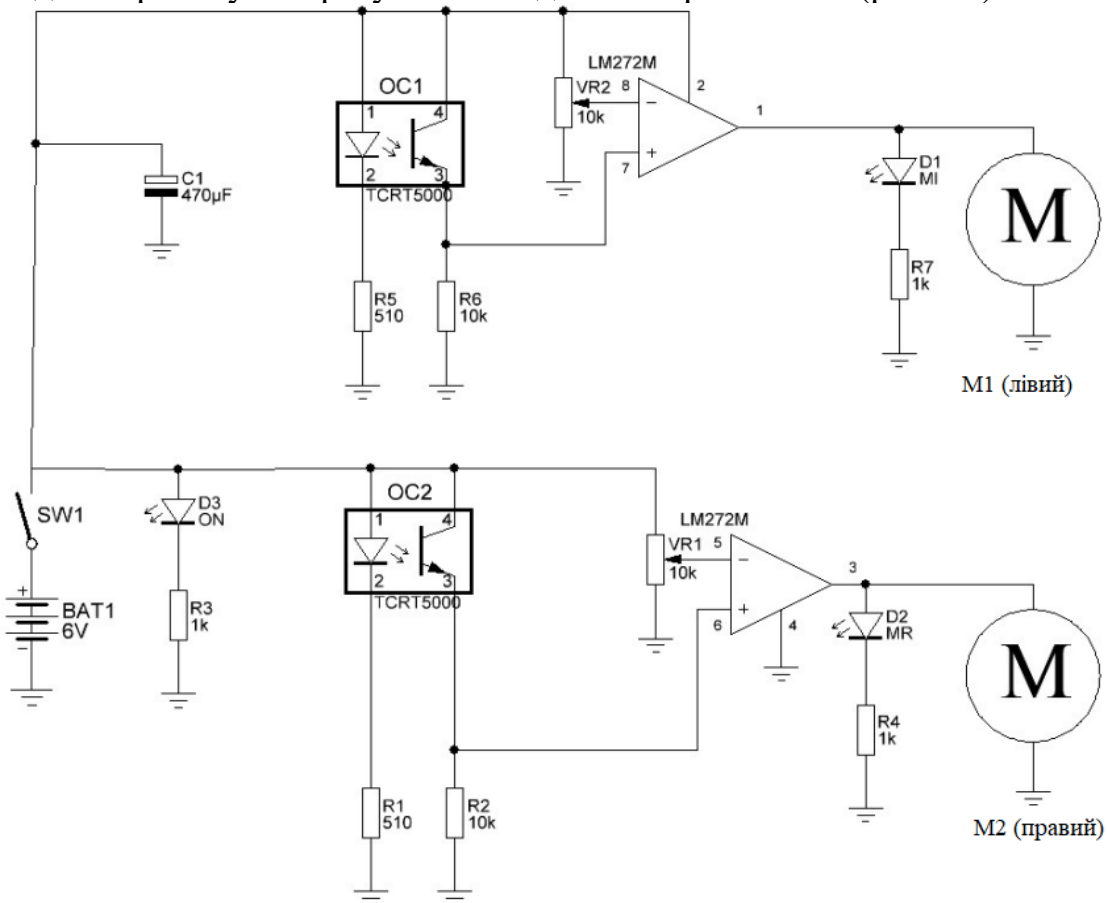


Рисунок 1.8 Схема електрична-принципова аналогового роботу, що слідує вздовж «чорної лінії».

Принципи роботи ВЕАМ-роботів ґрунтуються на здатності конструкції реагувати на зовнішні стимули. Механізм симуляції поведінки робота за допомогою штучних нейронів був розроблений Марком Тілденом. Нейронна схема Тілдена являє собою систему управління рухом роботів.

Для керування ВЕАМ-роботами замість мікропроцесора використовується штучна «нервова мережа». Така мережа складається з великої кількості штучних нейронів. У найпростішій формі ВЕАМ-нейрон являє собою схему затримки імпульсу, яка складається з резистора, конденсатора та інвертора (пристрою, що перетворює високий рівень вхідної напруги на низький, і навпаки).

Нейромережа зібрана на мікросхемі **LM272M**. Прямий вхід (+) кожної частини мікросхеми підключений до інфрачервоного відбивача ОС, який використовується як датчик «чорної лінії». Інверсний вхід (-) пов'язаний із змінними резисторами VR1, VR2, які призначені для налаштування чутливості датчиків ОС1 і ОС2.

Інфрачервоний датчик **TCRT5000** працює за принципом відбиття світла. Цей датчик об'єднує в одному корпусі інфрачервоний передавач (ІЧ-світлодіод із довжиною хвилі 950 нм) і приймач (фототранзистор), розділені перегородкою. Пристрій випромінює світло в інфрачервоному діапазоні та приймає його відбиття від поверхні. Інтенсивність відбитого сигналу змінюється залежно від типу й кольору поверхні.

Коли напруга на прямому вході перевищує напругу на інверсному вході, вихідна напруга мікросхеми стає приблизно рівною напрузі живлення. У цьому випадку світлодіод вмикається, а мотор працює. Якщо напруга, що надходить із датчика ОС на прямий вхід, є нижчою за опорну напругу на інверсному вході (яка встановлюється за допомогою змінних резисторів VR1 або VR2), двигун постійного струму на виході зупиняється, а світлодіод не горить.

Таким чином, ця схема працює як компаратор напруги для керування роботою двигунів, використовуючи показники датчиків відстеження лінії, щоб визначити умови руху.

Компаратор – це операційний підсилювач без зворотного зв'язку з великим коефіцієнтом підсилення. Якщо подати на один його вхід (наприклад, інверсний) постійний рівень опорної напруги, а на інший вхід (прямий) змінний сигнал, вихідна напруга змінюватиметься стрибком від мінімального до максимального значення в момент, коли рівень вхідного сигналу перевищує рівень опорної напруги, встановленої на іншому вході, і навпаки.

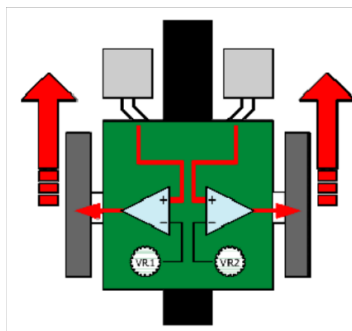
Компаратори мають два входи: прямий і інверсний. Залежно від бажаного результату, опорна й порівнювана напруги можуть підключатися до будь-якого з входів. Якщо напруга на прямому вході перевищує напругу на інверсному, вихідний транзистор компаратора відкривається, а якщо стає нижчою – закривається. Тобто компаратор виконує функцію порівняння напруг.

Основне завдання компаратора – порівнювати два сигнали й видавати вихідну напругу лише тоді, коли рівень сигналу на одному вході перевищує або стає нижчим за рівень опорної напруги іншого входу.

Ця схема керує двома двигунами постійного струму.

Розглянемо, як це працює.

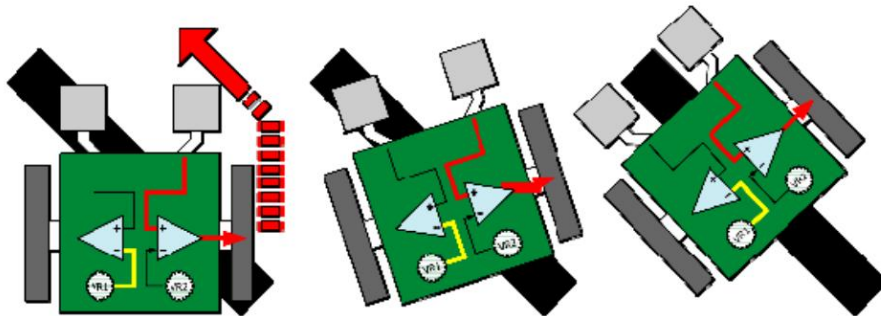
Випадок 1: лівий і правий датчики, які відстежують чорну лінію, виявляють білу поверхню .



Рух робота вперед

Це означає, що робот знаходиться над чорною лінією. Обидва датчики видають високий рівень напруги (наближений до напруги живлення). У цьому випадку напруга на прямих входах обох частин компаратора є високою, і обидва двигуни працюють, тобто робот рухається вперед.

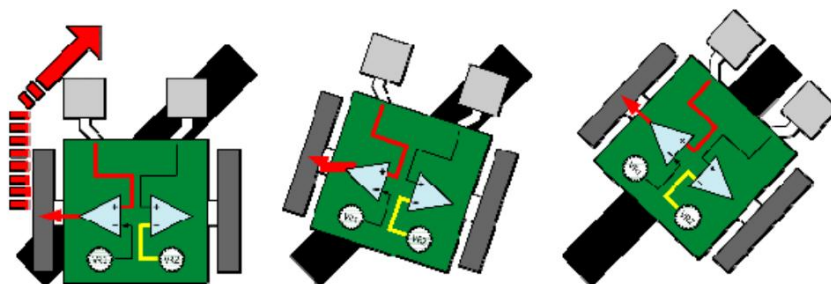
Випадок 2: лівий датчик виявив чорну лінію.



Поворот робота, що рухається вздовж лінії, вліво

Кожного разу, коли лівий датчик виявляє чорну лінію, а правий датчик фіксує білу поверхню, лівий датчик видає низьку напругу (нижче опорної). Це призводить до того, що одна частина компаратора видає нульову напругу на виході, лівий електродвигун зупиняється, а правий двигун продовжує працювати, оскільки правий датчик все ще фіксує білу поверхню. Робот повертає ліворуч.

Випадок 3: лівий датчик виявляє білу поверхню, а правий датчик фіксує чорну лінію.



Поворот робота, що рухається вздовж лінії, вправо

Правий датчик видає низьку напругу на виході. Правий двигун зупиняється, поки лівий двигун продовжує працювати, оскільки лівий датчик все ще фіксує білу поверхню. Робот повертає праворуч.

Випадок 4: обидва датчики відстеження лінії виявляють чорну лінію.

У цьому випадку робот зупиняється.

Переваги робототехнічного проєкту з використанням концепції BEAM:

- Робота може бути реалізована персоналом з низьким рівнем кваліфікації.
- Проєкт простий у конструюванні, має низьку вартість компонентів, вищий рівень надійності та менше енергоспоживання порівняно з програмованими роботами.

ВИСНОВКИ

Усі розглянуті завдання, що виникають при створенні роботів, необхідно вирішувати поетапними ітераціями. На кожному етапі потрібні спеціальні знання та навички в кількох суміжних галузях. Водночас завдяки зусиллям компаній, що виробляють іграшки, формується хибне уявлення, що створення робота — це проста задача. Хоча подібні набори і корисні для навчального процесу, слід пам'ятати, що справжній прорив у робототехніці може зробити лише той спеціаліст або колектив, який створює щось нове, власне, спираючись на власне розуміння того, як слід вирішувати кожен окрему задачу і підзадачу. Зібрати стандартного або майже стандартного робота з готових деталей, спеціально призначених для цієї мети, — це діяльність, яка може привити інтерес до розробки та конструкторської творчості, але це не те, що визначає прогрес науки, техніки та технологій. Попри високу цінність для освіти та популяризації, цей вид творчості залишатиметься недостатньо продуктивним, оскільки більше нагадує гру в конструктор, ніж процес інженерного створення. Тут рухова установка (колісний чи гусеничний привід або інший) вибирається із наявного набору, як і датчики, двигуни та інтелектуальні пристрої. Піком відмови від творчості є пропозиція закачати в контролер стандартне програмне забезпечення, яке дозволяє створеному роботу рухатися за бажанням його "автора". В цьому, на думку авторів, вже немає творчості. Це просто процес зборки готового виробу, розробленого кимось до вас, як складання меблів за інструкцією або приготування їжі з попередньо нарізаних порцій, слідуючи рецепту.

Для того, щоб дійсно навчитися розробляти робототехнічні пристрої, розробники повинні брати паяльник до рук, а не просто приєднувати деталі одна до одної. Вони повинні самостійно створювати програми на основі розроблених алгоритмів, а не завантажувати їх за заданим посиланням.

1.2 ОГЛЯД СТАНУ СВІТОВОГО РИНКУ РОБОТОТЕХНІКИ ДЛЯ СІЛЬСЬКОГО ГОСПОДАРСТВА

1.2.1 Автономні трактори

Ідея створення безпілотного трактора виникла ще на початку 1940-х років. Її автором став Френк Андрю. У 1950-х роках компанія «Форд» також розробила свій власний безпілотний трактор, який так і не був запущений у виробництво, оскільки його конструкція передбачала укладання кабелю під землею по всьому полю. І лише з появою сучасних комп'ютерних технологій і GPS стала можливою реалізація ідеї безпілотних тракторів.

Функціональні можливості безпілотних тракторів:

- виконують необхідні маневри з мінімальними помилками і визначають межі поля;
- працюють цілодобово, керуються за допомогою планшета;
- відрізняють уявні перешкоди від реальних;
- можуть рухатися за системою «слідуй за мною»: один автономний трактор, керований людиною, координує рух кількох безпілотних тракторів на полі, задаючи їм необхідну швидкість і напрямок руху.

За наявною інформацією, у 2016 році було продано понад 300 тис. тракторів з автопілотами. Триває робота над концепцією «слідування за лідером». Ведучі виробники сільськогосподарської техніки працюють над створенням повністю безпілотного трактора.

Одним із яскравих прикладів концепту роботизованого трактора є автономний трактор, представлений компанією «Case IH» у 2016 році (рис. 1.9).



Рисунок 1.9 Case IH безпілотний трактор

Машина створена для інтерактивного інтерфейсу, що дозволяє віддалено контролювати попередньо запрограмовані операції. Бортова система автоматично обирає найбільш ефективні шляхи, враховуючи тип ґрунту, перешкоди та наявність

іншої техніки на полі. Віддалений оператор може здійснювати спостереження і коригувати маршрути через комп'ютер або планшет.

Трактор оснащений радаром, лідаром і бортовими камерами, які дозволяють реєструвати стаціонарні або рухомі перешкоди на шляху і зупинити машину, поки оператор, попереджений про проблему, не задасть новий маршрут. Машина також зупиняється при втраті GPS-сигналу або інших даних позиціонування, або при натисканні оператором відповідної кнопки. Завдання трактора можуть змінюватися в реальному часі через віддалений інтерфейс або автоматично, з урахуванням прогнозу погоди.

Ще один яскравий концепт на ринку безпілотних тракторів розробляють учасники бізнес-акселератора «Y Combinator» (США), які працюють над безпілотними тракторами «Bear Flag Robotics». Як і «Case IH», компанія «Bear Flag» застосовує різні датчики, включаючи лідар, камери, GPS та інші, а бортові комп'ютери обробляють дані для забезпечення ситуаційної обізнаності, виконання завдань, а також для відстеження перешкод і моніторингу стану системи.

Особливість цієї розробки полягає в тому, що трактор може орієнтуватися без використання GPS. Замість цього система навчається розпізнавати навколишнє середовище за допомогою датчиків і камер, оскільки робота трактора передбачає використання різного навісного обладнання, яке може перешкоджати GPS-сигналу.

Ще одним важливим гравцем на ринку роботизованих тракторів є японська корпорація «Yanmar», яка представила серію сільськогосподарських тракторів з автономним режимом роботи під брендом «Robot Tractor» (рис. 1.10).



Рисунок 1.10 Робот трактор фірми **Yanmar** для посадки рису

Ці машини є наполовину роботами і можуть працювати під керівництвом людини, виконувати віддалені команди або слідувати запрограмованому маршруту. Їх

особливістю є широкий спектр підтримуваних завдань, що дозволяє замінити старіючих фермерів на японських полях. Компанія також анонсувала автономну машину для посадки рисових саженців, створену на базі системи «Smartpilot».

За результатами проведеного дослідження основні технології та принципи роботи, на яких базується функціонування роботизованого трактора, можна описати наступним чином.

- У безпілотних тракторах використовуються лазери, сенсори, радарні системи, GPS та інші бездротові технології. Управління такими тракторами може здійснюватися або через планшет, або з однієї з машин, що працюють на полі (тобто на поле можна вивести кілька тракторів, але їх роботу координує один тракторист з одного трактора — це технологія "слідуй за мною").
- У безпілотних тракторах застосовані складні алгоритми поведінки. При виявленні перешкод — людей, тварин або інших великих об'єктів — трактори миттєво зупиняються. Водночас вони здатні відрізнити реальні перешкоди від мнимих. Наприклад, сенсорні системи можуть "розпізнавати" високі стебла кукурудзи чи соняшника та не вважати їх перешкодою для руху.
- Трактори можуть бути як на гусеничному, так і на колісному ході, мати кабіну або бути без неї, оснащуватися дизельними або електричними двигунами.

Ось деякі перспективні технології на ринку розробок роботизованих тракторів:

- дистанційне управління або «повна автономність», що не вимагає присутності спеціаліста на місці;
- автономна колективна робота групи тракторів;
- орієнтація без використання GPS;
- електрична трансмісія.

На основі проведеного дослідження характеристик безпілотних тракторів можна зробити висновок про ряд переваг їхнього використання в аграрному секторі:

- цілодобова робота, що дозволяє скоротити терміни виконання робіт;
- безпечне робоче місце;
- вирішення кадрової проблеми;
- вирішення питання відтоку робочої сили;
- підвищення точності обробки землі, а отже, ефективності сільськогосподарських операцій;
- системи точного позиціонування, які дозволяють зменшити зону перекриття, знизити перевитрату добрив і хімікатів, а отже, знизити операційні витрати;
- підвищення продуктивності працівників завдяки автоматизації повторюваних функцій і зниженню вимог до досвіду роботи.

1.2.2 Роботизовані пристрої та комплекси для рослинницької галузі

Роботизовані пристрої та комплекси широко використовуються для вирішення завдань як у рослинництві, так і в тваринництві. Для якісного дослідження ринку роботизовані пристрої були класифіковані за типом виконуваних робіт.

До таких пристроїв належать роботизовані комплекси для знищення бур'янів та роботи-проріджувачі.

За деякими оцінками аналітиків, резистентність розвинули 255 видів бур'янів на 92 культурах у 70 країнах. Останній принципово новий гербіцид вийшов на ринок більше 20 років тому. Це створює попит на роботів-прополювальників — як автономних, так і керованих людиною. Обсяг цього ринку прогнозується на рівні 400 млн дол. США вже в 2025 році.

В роботах-прополювальниках розробники заклали функції розпізнавання та знищення бур'янів. Методи знищення бур'янів можуть бути різними: за допомогою пестицидів, лазера або механічно (вбивання в ґрунт, видалення з ґрунту). З урахуванням посилення вимог до засобів захисту рослин перспективними є технології видалення бур'янів без застосування гербіцидів (лазерна, механічна).

Однією з перспективних є технологія, що застосовується в прототипі робота-прополювальника «Hyperweeder» (рис. 1.11).



Рисунок 1.11 Роботизована система контролю бур'янів **Hyperweeder**

Робот оцінює форму бур'яна, щоб визначити, де знаходиться його меристема — частина рослини, яка відповідає за постійне нарощування маси. Потім він видаляє бур'ян за допомогою мікролазера, нагріваючи рослину до 95 °С. Цю технологічну платформу можна буде застосовувати до всіх бур'янів та ідентифікувати всі типи культур.

Роботи-проріджувачі мають схожу технологію розпізнавання з роботами-прополювальниками. Вони визначають кількість рослин і необхідну відстань між

ними, після чого застосовується хімікат для видалення небажаних рослин. Прополувальники затребувані на фермах по вирощуванню салату.

Салат часто висаджують набагато гуще, ніж потрібно. Це гарантує урожай у випадку проблем з проростанням або хворобами. Після того, як рослини проростають, необхідно прибирати зайві паростки для досягнення правильної відстані. Використання роботів різко скорочує ручну працю.

Переваги застосування роботів-прополувальників:

- виключення надмірного використання гербіцидів за допомогою точкового обприскування, а отже, зниження витрат на хімікати та зменшення негативного впливу на навколишнє середовище;
- при застосуванні лазерних та механічних технологій — виключення впливу гербіцидів на навколишнє середовище;
- підвищення продуктивності праці завдяки точності та цілодобовій роботі;
- підвищення ефективності роботи шляхом усунення можливості людської помилки при виконанні ряду однотипних та трудомістких завдань.

На основі проведеного дослідження, перспективними напрямками розвитку роботів-прополувальників можуть бути розробки технологій, що дозволяють точно визначати вид бур'яна та на основі аналізу застосовувати найбільш ефективний метод його видалення, а також розробка технологій, що скорочують використання гербіцидів.

1.2.3 Роботи для збору врожаю

Можливість збору плодів за допомогою роботів реалізується завдяки технології комп'ютерного зору. Колір ягід є основним показником зрілості, тому ймовірність помилок комп'ютерних систем зору мінімальна. Робот імітує рухи людської руки.

Основою роботи роботів для збору плодів є інтеграція машинного зору, сенсорів тиску, м'яких маніпуляторів, датчиків, що визначають прикладену силу, а також здатності самостійно переміщатися в просторі. Одна з складнощів при зборі врожаю — це те, що кожен фрукт, овоч чи ягода мають унікальний розмір, колір і форму. Для вдосконалення алгоритму процесу збору плодів вчені розробляють і вдосконалюють інтелектуальні системи зондуючого обстеження.

Роботи здатні зібрати від 70 до 100 % врожаю, що робить їх конкурентоспроможними з традиційним способом збору (рис. 1.12). Фотографуючи плоди на грядці, робот порівнює їх з закладеними в його пам'ять «образами об'єктів» і завдяки цьому здійснює більш точний збір, у той час як люди можуть помилково збирати незрілі овочі.

Що стосується швидкості роботи, то наразі машини можуть відставати від людини. Однак, на відміну від звичайних працівників, машина не втомлюється і може працювати до 10 годин без перерв, збираючи овочі навіть у повній темряві, підсвічуючи грядки ліхтариком.



Рисунок 1.12 Робот для вирощування та збирання врожаю суниць садових **Agrobot SW6010** (Іспанія). Роботизована рука перебирає по одній ягоді. За три дні один робот «Agrobot» може зібрати полуницю з 800 соток. Він забезпечує на 50% зниження ціни свіжих ягід і здешевлення до 90% сировини для промислового виробництва ягідного пюре і наповнювачів для йогуртів.

Один із головних гравців на ринку роботів для збору ягід — це США, оскільки ця країна є найбільшим виробником полуниці в світі (понад 1360 тис. т щорічно).

Головна перевага роботів для збору плодів — це заміна монотонної ручної праці, здатність пристрою працювати безперервно в будь-який час доби з однаковою продуктивністю, що значно підвищує ефективність процесу. Важливим чинником для автоматизації процесів у сільському господарстві є можливість складання певних алгоритмів, які лежать в основі роботи робототехніки.

Роботи для збору врожаю вирішують проблему залучення сезонних працівників. Під час збору врожаю час має вирішальне значення, оскільки деякі культури потрібно збирати швидко. Це вимагає великої кількості робочої сили протягом відносно короткого періоду, що створює проблеми з підтриманням зайнятості на постійній основі.

На основі проведеного аналізу можна виділити такі перспективні напрямки технологічного оснащення роботів для збору ягід та фруктів:

- повна автоматизація;
- збільшення швидкості збору;
- реалізація функції розпізнавання пошкоджених плодів;
- впровадження аналітичних можливостей для розбиття інформації про зібрані плоди за деревом, акром та загальним ділянкою.

1.2.4 Роботи для виноградників і садів

До окремої категорії роботизованих комплексів виділено роботи для виноградників та роботи для садів і теплиць, оскільки вони спеціально розроблені для конкретних галузей сільського господарства (робота на виноградниках, вирощування кімнатних рослин) і, крім універсальних функцій, можуть виконувати специфічні завдання (обрізка лози, прищеплення, полив).

Використання роботизованих комплексів на виноградниках виправдане їхньою ефективністю. Як правило, виноградники мають неоднорідний ландшафт з численними схилами різного ухилу. Деякі ділянки більш врожайні, інші — менш. Інформація, яку фермер отримує від роботи, дозволяє виявити ці ділянки та обробляти їх залежно від особливостей, найбільш ефективним способом.

Для роботи на виноградниках цікаві не лише універсальні роботи, здатні виконувати всі основні функції робітників по догляду, але й роботизовані комплекси для виконання специфічних однотипних завдань (рис. 1.13).



Рисунок 1.13 Сільгоспробот для роботи на виноградниках **Wall-Ye V.I.N.** (розробка Крістофа Майлота). Робот призначений для обрізки виноградної лози і видаленню непродуктивних пагонів а також збору інформації про локацію, кількість та якість плодів

Попит на іригаційні роботизовані системи, такі як концепт RAPID, обумовлений необхідністю диференційованого внесення води на виноградниках, що не можуть забезпечити існуючі системи поливу.

Робот-косарка «VITIROOVER» дозволяє знизити витрати на робочу силу, зменшити екологічне навантаження (працює на сонячних батареях), підвищити ефективність виконання робіт завдяки точності (рис. 1.14).



Рисунок 1.14 Робот косарка для виноградників **VITIROOVER**.

Актуальність розробки робота для прищеплення рослин обумовлена тим, що прищеплені саджанці помідорів, перцю, огірків та баклажанів використовуються все ширше, оскільки це дозволяє підвищити стійкість рослин до ґрунтових патогенів та шкідників, а також збільшує силу росту рослин. Завдяки появі нових оптичних технологій стало можливим створення роботів для прищеплення рослин.

Прищеплення — трудомістка операція, яка потребує високої кваліфікації персоналу, оскільки проводиться на ранніх стадіях розвитку рослин, у фазі розгорнутих сім'ядолей, коли сіянці дуже тендітні і крихкі. Все частіше виробники овочевих саджанців стикаються з нестачею робочих рук для виконання цієї операції, тому зрозумілий інтерес до автоматизації процесу прищеплення.

Роботи для виконання монооперацій поступово замінюють низькооплачувану механічну працю. Універсальний робот для садових центрів «HV-100», який з 2013 року використовується у багатьох садових центрах США, створений для роботи у квітковій індустрії (рис. 1.15). Він призначений для переміщення горщиків рослин у господарствах, що вирощують саджанці в контейнерах. Завдяки своїй всепогодності «HV-100» забезпечує необхідну точність незалежно від тривалості робочого дня і працює без перерв.



Рисунок 1.15 Робот **HV-100** від компанії «Harvest Automation» (США). Робот створений для переміщення контейнерів з саджанцями, здатний виконувати операції по проріджуванню і збору рослин в розсадниках або теплицях (<https://public.harvestai.com/>).

Серед перспективних розробок для роботів-виноградарів та роботів для садових центрів можна виділити:

- повну автономність роботи та здатність до самостійного прийняття рішень;
- багатозадачність — виконання додаткових завдань паралельно з основним (наприклад, одночасно з обрізкою лози здійснювати перевірку стану рослин за показниками росту, вологості чи симптомами захворювань).

1.2.5 Самохідні роботи для моніторингу полів

Необхідність проведення моніторингових досліджень, таких як виявлення процесів деградації ґрунтів, контроль фаз вегетації сільськогосподарських культур та прогнозування врожайності з використанням сучасних технологій, стає дедалі більш актуальною та знаходить підтримку серед зацікавлених сторін і організацій.

Для своєчасного виявлення проблемних ділянок на полях із посівами необхідно регулярно спостерігати за станом рослин. Проте ручний огляд культур на великих площах є надзвичайно неефективним. Одним із можливих рішень є застосування роботів різних конструкцій. Хоча для цього часто використовують безпілотники, вони не завжди ефективні на невеликих ділянках і не підходять для безперервного моніторингу полів, оскільки потребують регулярної заміни батарей.

Цікавим прикладом роботів для моніторингу є проєкт інженерів із Технологічного інституту Джорджії, які розробили робота-лінивця. Цей робот пересувається над полем за допомогою тросів. Як зазначають розробники, спосіб руху, що імітує пересування лінивців, є надзвичайно енергоефективним. У перспективі на такого робота планують встановити сонячні батареї. Завдяки цьому він зможе залишатися над полем протягом кількох місяців, здійснюючи спостереження за рослинами та передаючи знімки з камери фермерам. Для пересування робот використовує земну гравітацію та інерцію так само, як і його природний прототип (рис. 1.16).

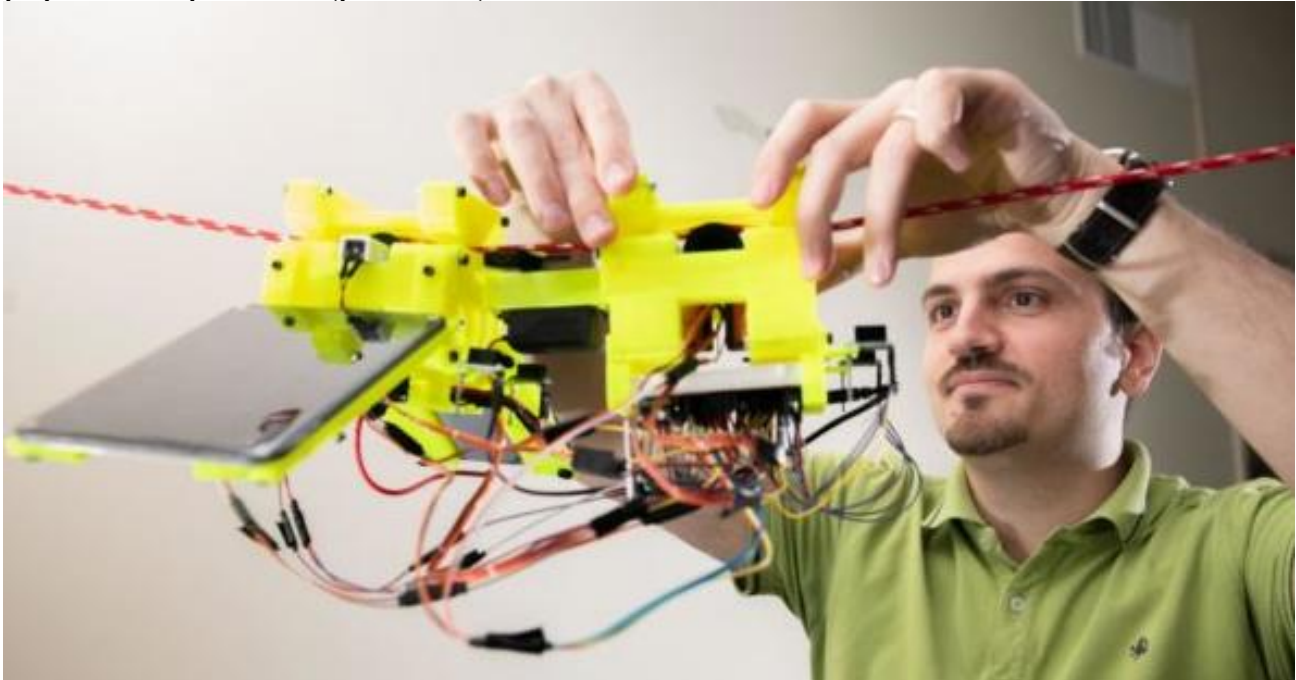


Рисунок 1.16 Прототип Робота для моніторингу. Розробники створили енергоефективного робота, який вмiє пересуватися тросами так, як ліновець рухається по гілках. Із сонячними панелями він зможе здійснювати моніторинг впродовж місяців.

Інші роботи оснащені колісною базою, камерами та сенсорами. Проте їх основний недолік у порівнянні з роботом-лінивцем полягає в тому, що під час руху на колесах вони можуть застрягнути, перекинутися, пошкодити рослини або зіткнутися між собою чи з іншою технікою.

Однією з найуспішніших розробок роботів для моніторингу є австралійський робот «Ladybird» (2014 рік). Окрім виконання моніторингових завдань, він здатний виконувати широкий спектр універсальних функцій у полі. Цей робот обладнаний системою лазерного наведення та інтегрованим автоматизованим маніпулятором для збору врожаю. Він може працювати навіть уночі, звільняючи фермера від ручного виконання таких завдань, як прополка чи боротьба зі шкідниками (рис. 1.17).



Рисунок 1.17 Сільськогосподарський робот-помічник **Ladybird** («Сонечко»). Заявленим завданням роботу є збір різних видів даних, що стосуються землеробства.

Моніторинг на полі необхідно здійснювати не лише за станом культур, але й за станом ґрунту. Надмірне використання азотних добрив, що широко застосовуються у світі, спричиняє виділення закису азоту (N_2O), який має негативний вплив на екологію та рослини. Це може призвести до пожовтіння листя, руйнування клітинних мембран або уповільнення росту. Щоб запобігти таким негативним наслідкам, перш за все, необхідно визначити рівень N_2O на полі. Для цих цілей у 2017 році був створений «Field Flux Robot» (рис. 1.18). У перспективі розробники планують зменшити його розміри та підвищити продуктивність.



Рисунок 1.18 **Adigo Field Flux** — агроробот, який здатний проводити високоточний аналіз ґрунту і визначати кількість азоту (N_2O) на полі. Перевага всіх роботів для моніторингу полягає в їх здатності працювати цілодобово, аналізуючи стан полів. Ця інновація спрощує життя не лише фермерам,

а й селекціонерам, які проводять масштабні експерименти з тисячами сортів рослин і змушені вручну вимірювати ключові показники.

Основним недоліком самохідних роботів для моніторингу порівняно з БПЛА є їхня обмежена прохідність за несприятливих погодних умов (наприклад, весняної багнюки) або на специфічному рельєфі місцевості (наприклад, виноградники). Тому навіть використання робота-лінивця не можна вважати універсальним рішенням.

1.2.6 Роботи для тваринництва

У галузі сільського господарства, такій як тваринництво, роботизовані системи на сьогодні знайшли найбільш широке застосування. Аналіз сучасних розробок і рішень, представлених на ринку, показав, що майже всі роботизовані системи в тваринництві орієнтовані на виконання завдань із годування та доїння великої рогатої худоби (ВРХ), а також очищення проходів у тваринницьких приміщеннях від гною.

Роботів у тваринництві можна поділити на такі групи:

- Системи годування тварин: автоматизовані пасовищні системи, дозатори-змішувачі, кормозмішувачі-роздавачі, вирівнювачі кормів і інтегровані роботизовані системи годування.
- Доїльні роботи: роботи-дояри, інтегровані системи доїння та управління стадом.
- Роботи для очищення приміщень: автоматизовані прибирачі гною скребкового типу та автономні системи для очищення стійл.

Найширше в тваринництві використовуються інтегровані роботизовані системи. Це технологічні комплекси, у яких роботи взаємодіють один з одним, виконуючи всі необхідні операції, наприклад, приготування багатокомпонентних кормових сумішей та їх роздачу відповідно до потреб тварин.

Серед провідних виробників роботизованих систем для тваринництва можна виокремити такі компанії: «Lely», «Delaval», «GEA Farm Technologies» та «VouMatic».

Доїльний робот

Це автоматизована система, призначена для доїння корів у фермерських господарствах із безприв'язним утриманням та добровільним доїнням. Такі системи дозволяють практично повністю усунути вплив людського фактора, створюючи оптимальні умови для тварин, що сприяє зниженню витрат на оплату праці та збільшенню надоїв (рис. 2.11).

Перші доїльні роботи з'явилися на ринку на початку 1990-х років. До 2002 року було представлено 11 одно- та багатобоксових систем для роботизованого доїння. Однак навіть найпродуктивніший робот-дояр із п'ятьма постами одночасного доїння (розробка компанії «GEA Farm Technologies») здатен обслуговувати не більше 220–250 голів на добу (рис. 1.19). В Україні станом на 2025 рік найбільш популярними є однокосові системи, які здатні обслуговувати 50–70 корів.



Рисунок 1.19 Робот дояр від компанії **Delaval**. Система забезпечує контроль якості молока та моніторинг стану здоров'я корови за допомогою датчика, прикріпленого до вуха або шиї тварини.

Ця ситуація створила передумови для появи нового типу продукту, що поєднує точність і автоматизацію доїння з високою продуктивністю для великих стад. Так з'явилася роботизована «карусель». «Карусель» — це роторна установка, де одночасно доються великі групи тварин (рис. 1.20). Водночас усі операції з обробки вимені виконуються роботами-маніпуляторами, інтегрованими у конструкцію «каруселі».



Рисунок 1.20 Карусель для доїння корів **Gea Farm Technologies DairyProQ**

Робот-кормороздавач

У 2010 році компанія «Lely» розробила концепцію системи годування «Vector», яка спочатку застосовувалася виключно на фермах у Нідерландах.

Протягом двох років система проходила тестування, що дозволило створити високоякісний продукт для споживачів. Один робот-кормороздавач «Lely Vector» здатен обслуговувати від 25 до 300 голів великої рогатої худоби. Якщо використовувати два таких роботи разом, вони можуть забезпечити годування стада до 500 голів.

Технологія годування базується на багаторазовому русі кормороздавача вздовж стійл. Під час цього робот розподіляє концентровані, грубі та інші види кормів відповідно до рівня надоїв і фізіологічного стану кожної корови, що лактує.

Рухаючись за заданим маршрутом, робот не лише роздає корм, а й за допомогою лазерних датчиків оцінює кількість залишків корму, його розташування і, за потреби, підсуває його ближче до тварин. Якщо під час проходу робот виявляє, що на кормовому столі залишилося недостатньо корму, він самостійно повертається до «кухні», готує нову порцію та роздає її.

Система годування «Lely Vector» (рис. 1.21) може бути застосована на фермах із будь-якою організацією праці. Однак найбільш ефективно та рентабельно вона працює на фермах, де використовуються доїльні роботи «Lely Astronaut» та інше обладнання від «Lely», інтегроване в єдину систему управління стадом Т4С («Time for Cows»).



Рисунок 1.21 Робот-кормороздавач **Lely Vector** (Нідерланди). Бункер цього робота вміщує до 600 кг корму. Інтелектуальне програмне забезпечення дозволяє змішувати корми, розподіляти їх та підсувати ближче до тварин.

Програма управління стадом Т4С надає фермерам корисну статистику як по всьому стаду, так і по кожній корові окремо. Вона дозволяє оцінювати динаміку ваги й активності тварин, аналізує кількість доїнь, паузи між ними, час пережовування корму, швидкість віддачі молока, кількість соматичних клітин та споживання сухої речовини. Ці дані дають змогу фермеру в реальному часі відстежувати стан здоров'я тварин. Статистика надається у вигляді зрозумілих

звітів і графіків, що дозволяють швидко визначити, якій корові потрібна персональна увага. У результаті до 80% трудових витрат на фермі спрямовуються лише на обслуговування тих тварин, які цього потребують, наприклад, лікування, осіменіння чи інші заходи.

Робот для прибирання гною

Для забезпечення комфортних умов утримання тварин фермерам необхідно подбати про якісний корм, достатню кількість води та створення оптимального середовища. Швидке прибирання гною є важливим елементом цього процесу. Вологий гній на щілинній підлозі часто стає джерелом інфекцій копит. Корови з такими захворюваннями дають менше молока й мають труднощі з пересуванням. Крім того, накопичення вологого гною, особливо поблизу стійл, знижує якість молока через забруднення.

Мобільний робот-очищувач корівника «Lely Discovery» (рис. 1.22) ідеально підходить для очищення рівних щілинних підлог. Окрім цього, його можна застосовувати для коротких ділянок підлоги без щілин (максимальною довжиною до 5 метрів). Завдяки компактним розмірам і невеликій висоті робот без зусиль проходить під розділовими ґратами.



Рисунок 1.22 **Lely Discovery 120 Collector** — робот-прибиральник гною, має пристрої технічного зору завдяки яким легко орієнтується в корівнику.

Робот працює на акумуляторній батареї. Після кожного циклу очищення він автоматично повертається до зарядної станції, розташованої в зручному місці корівника. Ця станція також є початковою точкою маршруту. У нижній частині робота встановлений скребок, який спрямовує гній у отвори щілинної підлоги. Рух робота забезпечується двома колесами, які також відповідають за його керування.

Фермер задає маршрут роботи «Lely Discovery», що дозволяє більш ретельно очищати найбільш інтенсивно використовувані ділянки корівника. Велике горизонтальне колесо забезпечує рух робота вздовж стін і обходить перешкоди, такі як перегородки чи ноги тварин.

1.2.7 Робоферми та міські вертикальні ферми

Вертикальне рослинництво — це інноваційний підхід до міського фермерства, який дозволяє вирощувати сільськогосподарські культури на урбанізованих територіях в контрольованих умовах закритих приміщень, де можна регулювати освітлення, температуру та постачання поживних речовин, а також на відкритих майданчиках. Ця технологія значно знижує споживання прісної води, водночас зберігаючи землю та ґрунт. Вертикальне фермерство є гнучким і постійно вдосконалюється, даючи змогу обирати метод вирощування залежно від рівня складності та вартості. Такі ферми вже активно функціонують у Китаї, Південній Кореї, Японії, Сінгапурі, ОАЕ, Нідерландах, Італії, Великобританії, США та Канаді.

Архітектори давно розробляли ідеї інтеграції будівель із природою, ще до появи вертикальних ферм. Вирощування рослин у гідропонних системах застосовували ще стародавні цивілізації: висячі сади Вавилону, плавучі сади ацтеків у Мексиці та гідропонні конструкції стародавнього Китаю. Давньоєгипетські рукописи також описують методи вирощування рослин у воді.

У XIX столітті вчені та садівники проводили експерименти з гідропонікою, прагнучи вирощувати свіжі продукти в місцях, непридатних для традиційного землеробства. У 1915 році Гілберт Елліс Бейлі вперше використав термін «вертикальне сільське господарство» у своїй книзі. Він вважав, що вирощування рослин у контрольованому вертикальному середовищі принесе як економічні, так і екологічні переваги.

Реалізація таких проєктів міського фермерства гальмується здебільшого через організаційно-технічні складнощі. Водночас існує успішний досвід переобладнання старих складів та ангарів на промислових територіях. Яскравим прикладом є проєкт Sky Greens у Сінгапурі — першої у світі комерційної вертикальної ферми. Рослини вирощуються у 38-ярусних баштах висотою близько 9 метрів, форма яких нагадує літеру "А". За технологією контейнери з рослинами обертаються навколо алюмінієвих конструкцій, що забезпечує рівномірний розподіл води, сонячного світла та повітря. Така конструкція також спрощує догляд за рослинами, зокрема збирання врожаю зокрема з використанням роботів (рис. 1.23).

Існують і інші технології міського фермерства до яких добре адаптуються робототехнічні комплекси, зокрема аквапоніка. Аквапоніка — це штучна екосистема, в якій взаємодіють три ключові групи організмів: водні тварини (зазвичай риби), рослини та бактерії. Ця технологія є екологічно безпечною і базується на принципах взаємозв'язку риб і рослин: риби забезпечують рослини поживними речовинами, а рослини очищують воду. Основна ідея методу полягає у

використанні продуктів життєдіяльності водних тварин (риб, креветок) як живильного середовища для рослин (рис. 1.24).



Рисунок 1.23. Вертикальне фермерство Sky Greens, м. Сінгапур



Рисунок 1.24. Схема аквапоніки (зліва) та фото діючої аквапонної ферми, де вирощують овочі та рибу (справа)

Водні організми виділяють сполуки азоту, калію, фосфору та вуглекислий газ, які для них токсичні. Накопичення цих речовин створює проблему як у закритих системах промислової аквакультури, так і в звичайних акваріумах. Однак ці самі речовини є необхідними для рослин у гідропоніці, де їх зазвичай додають у воду для створення поживних розчинів. У системах аквапоніки це питання вирішується природним шляхом: продукти життєдіяльності риб переробляються бактеріями та засвоюються рослинами.

Перша у світі робоферма — «**HAND Free Hectare**».

Цей проєкт був реалізований університетом Харпера Адамса у Шропширі (Велика Британія) спільно з компанією «Precision Decisions». Робоферма дала можливість аграріям лише спостерігати за ростом культур. Автономні трактори та дрони самостійно виростили на площі 2,5 га 4,5 т ячменю. Усі процеси — від посіву насіння до збирання врожаю — були автоматизовані, а управління здійснювалося технічним персоналом із диспетчерської. Дрони з мультиспектральними датчиками здійснювали зйомку угідь, а спеціальні машини брали зразки ґрунту, аналізували його склад і визначали необхідні добрива. Камери в реальному часі повідомляли про шкідників і бур'яни.

Робоферма «Iron Ox».

У Каліфорнії робоферма «Iron Ox» була розташована на території колишнього складу площею понад 740 м². Її продуктивність — до 26 тисяч качанів зелені на рік, що у п'ять разів більше, ніж у традиційних господарствах аналогічної площі. Грядки з салатом розміщені у піддонах і підсвічуються світлодіодами. У майбутньому планується використовувати природне освітлення, тому модулі розташовані горизонтально, а не вертикально. На фермі працюють роботизовані маніпулятори й вантажні роботи: маніпулятори пересаджують рослини з одного контейнера до іншого, а вантажні роботи переміщують контейнери з водою масою до 362 кг (рис. 1.25).



Рисунок 1.25 Маніпулятори на роботизованій фермі **Iron Ox**

При побудові ферми потрібно було вирішити питання забезпечення злагодженої взаємодії між різними роботами. У «Iron Ox» було розроблено програмне забезпечення «The Brain», яке координує роботу всіх систем ферми:

контролює вміст азоту, температуру, місцезнаходження роботів, управляє ними та надає рекомендації людям щодо необхідного втручання.

Проекти робоферм не лише підвищують ефективність сільського господарства, але й вирішують проблему дефіциту робочої сили в аграрному секторі. Інноваційні технології особливо підходять для вирощування молодих рослин, зернових культур і створення здоровішої продукції без використання пестицидів. Попри те, що технології ще не дозволяють повністю замінити людину, вертикальне сільське господарство має всі шанси стати майбутнім агробізнесу.

Логічним та перспективним напрямом роботизації ферм є завдання повної автоматизації всіх процесів вирощування сільськогосподарських культур. Такі інноваційні проекти вже розробляються та впроваджуються у світі. Робоферми покликані автоматизувати всі етапи — від посіву насіння до збору врожаю в рослинництві та повний цикл обслуговування дійного стада в тваринництві. Технічний персонал керує машинами з диспетчерської, не втручаючись фізично в самі процеси.

Агропромисловий комплекс залишається енергоємною галуззю, у якій широко застосовується ручна праця, що супроводжується важкими монотонними операціями, а також шкідливими й небезпечними умовами. Використання роботизованих систем сприятиме підвищенню обсягів виробництва продовольства, зниженню енергозатрат на одиницю продукції та одночасному збільшенню врожайності сільськогосподарських культур і продуктивності тварин.

Роботизація стає не лише заміною дорогої людської праці, а й потужним рушієм для розвитку галузі, який можна порівняти з революцією після впровадження тракторів і добрив. Роботи здатні працювати в ширшому діапазоні умов, ніж люди, виконуючи завдання цілодобово та з увагою до деталей, що було б занадто дорогим при ручній праці.

1.2.8 Поточний стан і перспективи ринку робототехніки

Аналіз наявних даних свідчить, що світовий ринок сільськогосподарських роботів активно розвивається. Конкуренція на цьому ринку поки що невелика й представлена головним чином боротьбою великих розробників за нові ринки, а також змаганням стартапів, які прагнуть вирішити схожі проблеми.

Окрім роботів, що вже активно використовуються в молочному тваринництві, існує велика кількість концептів промислових зразків, які перебувають на стадії доопрацювання за підсумками польових випробувань. Аналітики прогнозують запуск серійного виробництва багатьох із них протягом найближчих двох-трьох років. Попит на ринку залишається високим, і є потреба в інноваційних ідеях та технологіях.

Основними гравцями в цьому секторі є США, Японія та Китай. Роботи мають безліч переваг: вони працюють із високою точністю, без участі фермера, накопичують і аналізують дані для прийняття ефективних рішень. Завдяки доступу

до онлайн-баз даних, прогнозів погоди та інформації з внутрішніх і зовнішніх датчиків, роботи можуть знати про стан ферми більше, ніж людина.

Особливо перспективними вважаються роботи з електродвигунами та сонячними батареями, які дозволяють знизити енергоспоживання та викиди вуглецю. Роботизація сприяє тому, щоб сільське господарство ставало більш екологічним.

Основні виклики для впровадження робототехніки

Попри перспективи, існує низка труднощів, які гальмують розвиток:

- високі початкові інвестиції;
- неоднорідність робочого середовища для роботів;
- складність ідентифікації цілей та перешкод;
- недостатній розвиток технологій навігації;
- ризики для безпеки працівників (роботи можуть не помітити людину);
- потреба перепланування старих ферм для нових систем;
- високе споживання електроенергії.

Переваги роботизації

Попри складнощі, застосування роботів у сільському господарстві має значні переваги:

- здатність працювати на вологих ґрунтах;
- робота 24/7 у будь-яких погодних умовах;
- зниження екологічного навантаження;
- підвищення рентабельності;
- можливість оперативного аналізу даних для прийняття рішень;
- компенсація дефіциту робочої сили та заміна сезонних працівників;
- зменшення ризику ущільнення ґрунту;
- зростання продуктивності праці.

Майбутнє роботизації в агропромисловості

Очікується, що в найближчі роки роботизація проникне в усі сфери сільського господарства, трансформуючи способи ведення фермерської діяльності та змінюючи ланцюжки створення вартості. Інтегровані мережі, у яких усе обладнання обмінюється інформацією, забезпечать автономність і ефективність ферм, роблячи їх менш шкідливими для навколишнього середовища й більш прибутковими.

1.3 БЕЗПЛОТНІ ЛІТАЛЬНІ АПАРАТИ

Важливим напрямом використання роботів в аграрній сфері є впровадження безпілотних літальних апаратів (БПЛА), завдяки яким щорічно з'являються нові інструменти та можливості для аграріїв. Великі науково-дослідницькі компанії прогнозують значний ріст цього сектору і домінування аграрної галузі як основного напрямку використання безпілотників. За оцінками компанії «Pricewaterhouse Coopers», через кілька десятиліть ринок «сільськогосподарських» дронів (без урахування безпілотних літаків) може досягти близько 32,4 млрд доларів США. Ринок БПЛА перебуває на стадії формування та зростання.

Основні гравці цього ринку зосереджені в США та Китаї. До найбільших світових компаній, орієнтованих на сільське господарство, належать такі представники, як «AeroVironment Inc.», «AgEagle», DJI, «Yamaha» та інші. Дистанційне зондування на основі БПЛА відкриває широкі можливості для точного землеробства; БПЛА можуть бути не лише потужним інструментом моніторингу, а й ефективно виконувати конкретні сільськогосподарські операції.

Безпілотники активно використовуються як у рослинництві, так і в тваринництві. Основною перевагою застосування БПЛА є спрощення доступу до інших технологій, що дозволяють вирощувати більше і витратити менше, при цьому це стосується не лише агрохолдингів, а й фермерів.

1.3.1. Огляд світового ринку БПЛА та сфер їх застосування

Сільське господарство, зокрема землеробство, є одним з найбільш перспективних напрямків застосування безпілотних літальних апаратів (БПЛА), головним чином для вирішення завдань точного землеробства. Система точного землеробства (precision farming) — це новий етап у розвитку ІТ-агрономії. Це не набір конкретних методик і технологій, а загальна концепція, що ґрунтується на використанні супутникового позиціонування (GPS), геоінформаційних систем (GIS), точного картографування полів та інших інструментів. Виконати завдання точного землеробства неможливо без БПЛА.

За оцінками компанії «Juniper Networks», у 2016 році до 48 % комерційних дронів використовувалося в сільському господарстві. До 2026 року цей показник може досягти 80 %.

Загальносвітовий підхід до використання безпілотників передбачає, що це пристрої, що працюють на висоті до 120 м у межах прямої видимості, хоча в ряді країн питання залишається дискусійним. БПЛА активно використовуються в сільському господарстві в США, країнах Європи, Бразилії та Аргентині. Проте найбільше інновація застосовується в Китаї, який займає лідируючі позиції з виробництва БПЛА. В Україні також з'явилося багато компаній, що займаються розробкою безпілотників, але запчастини для них, зазвичай, імпортні. БПЛА оснащуються мультиспектральними камерами, які дозволяють точно визначити проблемні ділянки поля завдяки високій чіткості зображень. Також на них

встановлюються різноманітні датчики, системи супутникової навігації, компактні бортові комп'ютери та обладнання для внесення хімікатів.

Дрони поділяються на два основні класи: літакового типу (з нерухомим крилом) та мультироторні (коптерні). Однією з основних різниць між ними є те, що перші здатні за один політ охоплювати територію, на порядок більшу, ніж коптери. Однак літакові платформи вимагають принципово вищої кваліфікації від операторів. Окремо є питання щодо зльоту то особливо посадки таких недешевих апаратів. (рис. 1.26).



Рисунок 1.26 БПЛА літакового типу **Spectator-M** виробництва компанії Мерідіан (<http://merydian.kiev.ua/>) у польоті (ліворуч) та у переносному кейсі (праворуч).

Модель є конверсійним рішенням і може використовуватись як у складних погодних умовах так і при значних радіоперешкодах.

Певні моделі легких літакових апаратів масою до 10 кг можуть запускатись з руки чи катапульти та приземлятися як власне «на корпус» так і з допомогою парашуту. Парашутна система дозволяє більш ощадливо експлуатувати сенсорне обладнання, вартість якого може бути кратно вищою ніж у БПЛА, проте займає об'єм та частину корисного навантаження (рис. 1.27).



Рисунок 1.27 Фотознімок моменту виходу парашуту з БПЛА (<https://smartdrones.ua/>)

Літакові БПЛА мають сенс при обстеженні площ в десятки чи сотні гектарів впродовж одного польоту. Такі великі обсяги притаманні країнам пострадянського простору а також для Північної та південної Америки. В країнах Азії та Європи площі полів фермерських господарств складають кілька гектарів і для їх обстеження цілком достатньо дешевих коптерних платформ (рис. 1.28).



Рисунок 1.28 Сенсор **Survey3**, встановлена на квадрокоптер **DJI Phantom 4** (зліва) та на модулі підвісу (справа)

Існують і конструкції з гнучким крилом по типу пароплана чи дельтаплана, що мають переваги щодо логістики та зберігання. Перспективними вважаються і безпілотні дирижаблі здатні перебувати у польоті від кількох днів до кількох місяців. Такі апарати цілком здатні знайти свою нішу в аграрному виробництві де потрібні тривале спостереження зокрема в рослинницькій галузі (рис. 1.29).



Рисунок 1.29 БПЛА **Sky Dragon** (зліва) та його комплекс сенсорного обладнання **EWS NADATS** (справа)

Безпілотні літальні апарати в сільському господарстві можуть вирішувати такі завдання:

- створення електронних карт полів (будівництво 3D-моделей полів);
- інвентаризація сільськогосподарських угідь;
- оцінка обсягу робіт і контроль їх виконання для оптимального побудови систем зрошення та меліорації;
- оцінка якості пропашності;
- оперативний моніторинг стану посівів (БПЛА дозволяють швидко і ефективно створювати карти сходів);
- розрахунок нормалізованого індексу вегетації (NDVI) для ефективного внесення добрив;
- оцінка всхожості сільськогосподарських культур;
- виконання посівних робіт;
- прогноз урожайності сільськогосподарських культур;
- екологічний моніторинг сільськогосподарських земель;
- охорона сільськогосподарських угідь;
- обприскування посівів хімічними препаратами для боротьби з шкідниками і хворобами;
- оцінка хімічного складу ґрунту;
- моніторинг поголів'я великої і дрібної рогатої худоби.

Застосування БПЛА в сільському господарстві є економічно вигідним доповненням малим літакам і супутникам. Придбання та обслуговування безпілотнику обходиться дешевше, ніж будь-якого пілотованого літака. Крім того, багато завдань, які виконуються в сільському господарстві, БПЛА здатні вирішити швидше, ніж супутники. Аерофотознімальні БПЛА дозволяють отримувати знімки високої роздільної здатності в заданих точках за GPS-координатами. Для кожного знімка формується повний набір цифрової інформації, що включає координати центральної точки даних, які можна перенести та використовувати в загальноприйнятих системах (наприклад, «ArcView» або «MapInfo»). Таким чином, усі фотографії є геоприв'язаними, і їх можна об'єднати в один великий ортофотоплан поля. Детально питання моніторингу розглянуто в посібнику [1]

Одним із лідерів ринку виробників безпілотників для сільського господарства є китайська компанія DJI, яка у 2015 році розробила БПЛА «DJI Agras MG-1». Цей дрон має властивості захисту від вологи та пилу, оскільки виготовлений із матеріалів, стійких до корозії. Після завершення роботи його можна легко промити та скласти для транспортування. Восьмимоторний «Agras MG-1» здатний нести до 10 кг рідини для обприскування й обробляти площу від 3,2 до 4 км² на годину, що в 40 разів ефективніше, ніж ручне обприскування. Дрон може рухатися зі швидкістю до 8 м/с і автоматично регулювати інтенсивність розпилення залежно

від швидкості польоту. На базі досвіду використання було розроблено більш доскональні моделі такі як DJI Agras T20 (рис.1.30), здатний завдяки вбудованому радару обходити перешкоди та більш ефективно вносити бакові суміші.



Рисунок 1.30 Мультикоптер **DJI Agras T20**

1.3.2. Світовий досвід використання БПЛА в тваринництві

Дрони успішно використовуються не лише в рослинництві, а й у комерційному тваринництві. Вони застосовуються як для випасу великої рогатої худоби, так і для моніторингу стану тварин. Завдяки здатності створювати повітряні завихрення від гвинтів, дрони допомагають власникам господарств точно утримувати худобу в межах визначених пасовищ. Крім того, оснащення дронів тепловізійними камерами, такими як «DJI Zenmuse XT», дозволяє знаходити корів у лісових заростях, оскільки тепловізор допомагає відрізнити тварин від інших джерел тепла. Такі камери також є ефективним інструментом для виявлення небезпечних хижаків поблизу фермерських угідь.

Світовий інтерес до точного тваринництва постійно зростає. Наприклад, у Канаді Дослідницька рада з природничих наук та інженерії виділила трирічний грант у розмірі майже 664 тисяч доларів США на розробку інноваційних рішень у тваринництві. Один із учасників проекту, доктор Дж. Черч із Технологічного інституту Південної Альберти (Канада), використовував дрон «DJI Phantom 3 Professional» для демонстрації ефективності таких пристроїв у точному тваринництві. Експерименти показали, що багато типових завдань на фермі можна виконувати значно швидше за допомогою дронів. Наприклад, це стосується щоденного підрахунку поголів'я худоби або випасу корів. Практика показала, що навіть найкращі пастуші собаки не завжди можуть впоратися з такими завданнями.

Серед нових ідей доктора Черча — використання RFID-міток для кожної тварини. За допомогою дронів та спеціального програмного забезпечення можна буде точно визначати місцезнаходження кожної корови та моніторити її стан.

1.3.3 Світовий досвід застосування БПЛА в рослинництві

У рослинництві безпілотні літальні апарати (БПЛА) використовують не лише для моніторингу стану ґрунту, а й для диференційованого внесення добрив та засобів захисту рослин, що є особливо важливим для полів із високою варіабельністю родючості ґрунтів [6].

В сільському господарстві Японії агродрони застосовують для зрошення полів, внесення добрив та сівби. Дрони, обладнані спеціальними датчиками, здатні визначати, які ділянки полів пересихають або потребують обробки. Крім того, за допомогою дронів під час росту культур можна оцінити густоту та якість потенційного врожаю. У 2012–2013 роках на японських полях вже працювало близько 3000 агродронів, більшість із яких працювали на бензині чи дизельному пальному. Сучасні дрони здебільшого оснащені акумуляторними батареями, проте для БПЛА по внесенню реагентів двигуни внутрішнього згорання займають суттєву нішу ринку, оскільки на відміну від акумуляторів, їх заправка триває лічені хвилини. Найбільш перспективними вважаються гібридні моделі БПЛА, які поєднують переваги як двигунів внутрішнього згорання так і суто електричних апаратів (рис. 1.31).



Рисунок 1.31 Гібридний дрон **Yeair** австрійської компанії **Airstier** (<https://yeair.de>). Здатний нести до 5 кілограм корисного навантаження, із швидкістю до 100 км/год. Заявлена вартість станом на 2018 рік становила 1400 Євро.

У Японії діє обов’язкова реєстрація безпілотних вертольотів, які використовуються для розпилення пестицидів, внесення добрив і сівби (рис. 1.32).



Рисунок 1.32. Бензиновий дрон **Yamaha RMAX**, розроблений в Японії 90-х роках. Вантажопідйомність 28-31 кг. Вартість в 90-роки складала біля 100 тисяч доларів.

Така реєстрація дронів була запроваджена у 2015 році. Станом на січень 2018 року кількість зареєстрованих дронів сягнула 673, що майже утричі більше, ніж у березні 2017 року. Купівля агродронів у Японії значно дешевша, ніж оренда звичайних вертольотів для обробки полів.

Найпопулярніші японські дрони для розпилення пестицидів мають резервуари об'ємом від 8 до 10 літрів. Для обробки одного гектара рисових полів потрібно близько 10 літрів пестицидів. Фермеру зазвичай потрібно кілька годин, щоб обробити гектар вручну, тоді як дрон виконує цю роботу приблизно за 10 хвилин. На відміну від звичайних сільськогосподарських безпілотників, БПЛА здатні робити різкі повороти та не створюють сильного повітряного потоку, який може пошкодити листові овочі. Попри широку рекламу інноваційних засобів внесення засобів захисту рослин їх використання залишається дискусійним питанням. Так в Канаді, країні із передовою рослинницькою галуззю, використання БПЛА для внесення хімічних засобів захисту рослин заборонено на законодавчому рівні. Це пов'язано з вкрай поганим методичним супроводженням та недосконалими протоколами по внесенню препаратів, завдяки яким відбувається дрейф отрутохімікатів на сусідні поля із їх враженням. Існує кілька шляхів вирішення як суто технічного плану (вдосконалення конструкції БПЛА, розпилюючих елементів, розташування гвинтів тощо) так і методичного плану (склад бакових сумішей, висота польоту, розмір краплі тощо). Більш детально

питання використання дронів для внесення бакових сумішей розглянуто в посібнику [2].

Дослідження ринку БПЛА дозволило виокремити ключові завдання, які вони виконують у рослинництві.

Зокрема, БПЛА застосовують для:

- оцінки ділянок ґрунту, схильних до засухи, та наявності водних ресурсів;
- дистанційного зондування стану рослин (ця технологія базується на аналізі кольору листя та інших частин рослин у видимому спектрі (RGB) та інфрачервоному тепловому випромінюванні. Серед вторинних показників використовують індекс NDVI, індекс поглинання хлорофілу (TCARI), індекс фотохімічного відбиття (PRI) та оптимізований вегетаційний індекс ґрунту (OSAVI));
- оцінки водозабезпечення рослин (визначається різниця температур листя та повітря, яка пов'язана зі стомато-провідністю листя, за допомогою теплових нормованих індексів);
- моніторингу рівномірності росту зернових культур (комп'ютерний аналіз отриманих зображень виявляє ділянки із повільнішим ростом, що дозволяє коригувати кількість внесених добрив і покращувати якість посівів);
- диференційованого внесення добрив і пестицидів;
- оцінки шкоди, завданої дикими тваринами (наприклад, запустивши дрон уночі, коли активні олені та дикі кабани, можна простежити їхню поведінку, маршрути та місця вторгнення на ферми).

Японський стартап Skyrobot розробляє систему на базі БПЛА для захисту сільськогосподарських угідь від нападів диких тварин.

Компанія пропонує програмно-апаратний комплекс, до складу якого входять квадрокоптер, камера з інфрачервоним датчиком та система зі штучним інтелектом [11]. За задумом розробників, дрон патрулюватиме визначену територію, виявляючи тварин, які наближаються до полів, та відлякуючи їх за допомогою високочастотних сигналів або звуків, що імітують вибухи петард. Завдяки камері, здатній працювати як у звичайному режимі, так і в інфрачервоному, потенційних порушників можна виявити навіть уночі чи в лісовій місцевості. Система штучного інтелекту аналізуватиме дані, зібрані дроном, щоб за слідами та поведінкою тварин прогнозувати їхні подальші дії.

1.3.4 Світовий досвід застосування БПЛА для біологічної боротьби зі шкідниками

Застосування БПЛА для боротьби зі шкідниками розпочалося кілька років тому, тому методи використання постійно вдосконалюються. Сьогодні дрони використовуються на великих площах завдяки збільшенню місткості резервуарів для біологічних засобів захисту рослин, які встановлюються на безпілотики. Сам

біологічний метод захисту рослин ґрунтується на використанні живих організмів проти шкідників, збудників хвороб і бур'янів. Такий захист має великі перспективи, оскільки приведені вище проблеми пов'язані з дрейфом препарату не несуть шкоди сусіднім посівам.

На ринку присутні серійні зразки дронів саме під внесення біологічного захисту, який відрізняється від хімічних засобів захисту принципово меншою вагою і відповідно дозою внесення (рис. 1.33).



Рисунок 1.33 Дрон для внесення трихограми на базі БПЛА DJI Phantom 4 виробник компанія Дрон уа (<https://drone.ua/pages/trihograma>)

Політ дрона можна налаштувати для внесення біологічного захисту через задані інтервали часу, а GPS-навігація допоможе створити оптимальний маршрут для обробки полів. Ідея полягає в тому, щоб поєднати кілька дронів у так званий «рій», який синхронно рухатиметься полем, розпорошуючи біопрепарати з точним дотриманням відстаней, що суттєво підвищить ефективність їх застосування. Такий метод, наприклад, для внесення трихограми (паразитичних комах), ідеально підходить для великих плантацій площею у сотні або тисячі гектарів. Водночас розробники працюють над вдосконаленням систем контролю роботи цих пристроїв, щоб забезпечити максимально точну обробку.

Попри те, що дрони наразі поступаються ефективністю малої авіації, вони дозволяють боротися зі шкідниками на плантаціях, які є надто великими для ручної обробки, але недостатньо великими для застосування малої авіації. Це особливо важливо для органічних господарств, де інше сільськогосподарське обладнання потребує регулярної дезінфекції та очищення, що ускладнює його використання.

За даними організації AUVSI, у доповіді «The Economic Impact of Unmanned Systems Integration in the United States» зазначено, що використання БПЛА в сільському господарстві домінуватиме над іншими методами. Очікується, що до 2025 року близько 80 % безпілотників у США працюватимуть у сільському господарстві.

1.3.5 Переваги використання БПЛА в сільському господарстві

Основні переваги застосування безпілотників у сільськогосподарській діяльності:

- **Точність у застосуванні хімікатів.** Безпілотники дозволяють збирати детальну інформацію про стан рослин, що забезпечує можливість точного внесення пестицидів та гербіцидів лише в тих місцях, де це необхідно. Це сприяє економії хімічних засобів, а також зменшенню негативного впливу на довкілля.
- **Створення картографічної бази.** З використанням БПЛА можна створювати карти з високою точністю, що включають координати всіх об'єктів. Це дозволяє виконувати детальний візуальний аналіз із роздільною здатністю до кількох сантиметрів на піксель. На такі карти можна наносити векторні шари, зокрема межі полів, дороги та інші об'єкти інфраструктури. Це дає змогу обчислювати точні площі, відстані, потреби в ресурсах тощо. Зручно визначати площі ріллі, пасовищ, сінокосів, залежних земель, парів, посівів, недосівів та перерослих ділянок.
- **Кадастровий облік.** Дані, отримані в результаті аерофотозйомки, можна використовувати для постановки земельних ділянок на кадастровий облік.
- **Висока деталізація зображень.** Аерофотозйомка з БПЛА забезпечує детальність завдяки висоті польоту від 100 до 600 метрів. Безпілотники можуть проводити зйомку навіть за умов хмарності, що недоступно супутникам і часто ускладнює роботу авіації.
- **Оперативність коригування.** Безпілотники дозволяють отримувати знімки безпосередньо під час польоту, а у разі потреби маршрут можна коригувати в реальному часі. Продуктивність зйомки досягає 30 км²/год для площинної зйомки та 35 км/год для лінійних об'єктів.
- **Оптимізація руху техніки.** Завдяки навігаційним системам БПЛА можна прокладати оптимальні маршрути для сільськогосподарської техніки, уникаючи надмірного ущільнення ґрунту, що негативно впливає на врожайність.
- **Економія трудових ресурсів.** Безпілотники можуть працювати цілодобово з високою продуктивністю. Наприклад, дрон може за три години засіяти 10 км² лісу насінням, тоді як людині на це знадобиться ціла доба.
- **Перспективи автоматизації.** У майбутньому можливий перехід до повністю роботизованих систем. Такі дрони зможуть автоматично заряджати свої акумулятори, виконувати обльоти за заданим графіком, проводити зйомку або відеоспостереження, передавати зібрані дані в автоматизовані системи обробки та повертатися на базу для обслуговування.

Отже перспективи впровадження БПЛА у сільське господарство є багатообіцяючими завдяки їхній здатності інтегруватися з іншими сучасними технологіями. Це спрощує та здешевлює перехід до точного землеробства, що оптимізує використання ресурсів і знижує витрати.

Світовий ринок послуг на базі БПЛА активно формується. Сервісні компанії впроваджують дрони у різні процеси та пропонують їх своїм клієнтам. Інновації у

сфері штучного інтелекту й спеціалізованих сенсорів виведуть використання дронів на новий рівень.

1.4 Питання для самоперевірки:

1. Яка головна мета створення роботів-андроїдів?
2. Чому ідея повної ідентичності андроїдів із людиною є ідеалістичною?
3. Чим роботи-андроїди відрізняються від інших класів роботів?
4. Чому деякі роботи намагаються лише відтворити функціональну ідентичність рухів, а не зовнішню схожість із людиною?
5. Які технологічні виклики стоять перед розробниками андроїдів?
6. Які види живих істот найчастіше наслідують роботи?
7. Чому вивчення рухів тварин важливе для робототехніки?
8. Які переваги мають біонічні роботи у порівнянні з традиційними механізмами?
9. Які відмінності між рухом роботів, що імітують тварин, та техногенними способами пересування?
10. Які практичні сфери застосування таких біонічних роботів?
11. Які основні техногенні способи пересування використовуються у робототехніці?
12. Чим відрізняються дрони від традиційних безпілотних літальних апаратів?
13. Які переваги багатогвинтових дронів перед традиційними вертольотами?
14. Коли виникла ідея створення безпілотного трактора, і які компанії працюють у цьому напрямку?
15. Які основні функціональні можливості сучасних автономних тракторів?
16. Чим технологія “слідуй за мною” покращує ефективність роботи безпілотних тракторів?
17. Які сенсори та системи навігації використовуються в роботизованих тракторах?
18. Які переваги автономних тракторів у порівнянні з традиційними?
19. Які галузі можуть отримати найбільшу користь від екзоскелетів?
20. Які методи використовують аграрні роботи для знищення бур'янів?
21. Чому лазерні та механічні методи боротьби з бур'янами стають популярнішими за хімічні?
22. Чому моніторинг полів є важливим для ефективного ведення сільського господарства?
23. Які переваги мають самохідні роботи для моніторингу у порівнянні з БПЛА?
24. Як роботизовані системи можуть аналізувати стан ґрунту та рослин?
25. Які сенсори та камери використовують роботи для моніторингу агрополів?
26. Як автоматизований моніторинг може допомогти у підвищенні врожайності?
27. Які переваги можуть мати роботи з покращеними сенсорними системами порівняно з людьми?
28. Які основні типи роботів використовуються у тваринництві?

29. Як роботи-дояри впливають на продуктивність молочного господарства?
30. Чому інтегровані системи автоматизації годування є ефективнішими за традиційні методи?
31. Як роботизовані системи можуть покращити утримання тварин?
32. Які перспективи розвитку роботизованих комплексів у тваринництві?
33. Що таке безпілотний літальний апарат (БПЛА) і які його основні характеристики?
34. Які існують основні класифікації БПЛА за розмірами та функціональністю?
35. Які ключові технології використовуються для керування безпілотниками?
36. У чому відмінність між автономними БПЛА та тими, що керуються оператором?
37. Які сфери застосування БПЛА є найбільш поширеними сьогодні?

1.5 Література до розділу 1

- 1.1 Somayya Madakam, Rohit Sharma, Shridhar Samant (2025) "**Chapter 21 - Is hyperautomation playing a significant role in smart agriculture?**", Editor(s): Sartajvir Singh, Vishakha Sood, Arun Lal Srivastav, Yiannis Ampatzidis, Hyperautomation in Precision Agriculture, Academic Press, pp. 251-257, <https://doi.org/10.1016/B978-0-443-24139-0.00021-7>
- 1.2 Veerendra Dakulagi, Kim Ho Yeap, Humaira Nisar, Rohini Dakulagi, G N Basavaraj, Miguel Villagómez Galindo (2025) "**Chapter 4 - An overview of techniques and best practices to create intuitive and user-friendly human-machine interfaces**", Editor(s): Abdulhamit Subasi, Saeed Mian Qaisar, Humaira Nisar, In Artificial Intelligence Applications in Healthcare and Medicine, Artificial Intelligence and Multimodal Signal Processing in Human-Machine Interaction, Academic Press, pp. 63-77, <https://doi.org/10.1016/B978-0-443-29150-0.00002-0>.
- 1.3 David J. Bruemmer, Mark S. Swinson (2003) "**Humanoid Robots**", Editor(s): Robert A. Meyers, Encyclopedia of Physical Science and Technology (Third Edition), Academic Press, pp. 401-425, <https://doi.org/10.1016/B0-12-227410-5/00317-3>.
- 1.4 James P. Ostrowski (2003) "**Robotic Locomotion Systems, Unconventional**", Editor(s): Robert A. Meyers, Encyclopedia of Physical Science and Technology (Third Edition), Academic Press, pp. 263-273, <https://doi.org/10.1016/B0-12-227410-5/00664-5>.
- 1.5 Nicolas A. Lyons, Juan I. Gargiulo, Cameron E.F. Clark, Sergio C. Garcia (2022) "**Technology and Robotic Milking in Dairy Production**", Editor(s): Paul L.H. McSweeney, John P. McNamara, Encyclopedia of Dairy Sciences (Third

Edition), Academic Press, pp. 823-829, <https://doi.org/10.1016/B978-0-12-818766-1.00289-0>.

- 1.6 Norah O'Shea, Mark A. Fenelon (2022) "**Process Instrumentation, Robotics, Wearables and Industry Internet of Things**", Editor(s): Paul L.H. McSweeney, John P. McNamara, Encyclopedia of Dairy Sciences (Third Edition), Academic Press, pp. 371-380, <https://doi.org/10.1016/B978-0-12-818766-1.00198-7>.
- 1.7 Ekaterina Glebova, Jeffrey Philpott (2024) "**Robotics**", Reference Module in Social Sciences, Elsevier, <https://doi.org/10.1016/B978-0-443-13701-3.00377-7>.
- 1.8 **Штучний інтелект у плануванні та управлінні виробництвом** : підручник для студ. вузів, які навч. за спец. "Комп'ютеризовані інтегровані системи та робототехніка", "Інтелектуальні інтегровані системи" / Л. С. Ямпольський, О. А. Лавров. - К. : Вища шк., 1995.. - 255 с.:іл.. - ISBN 5-11-004406-6
- 1.9 **Робототехніка та мехатроніка : навч. посіб.** / Л. І. Цвіркун, Г. Грулер ; під заг. ред. проф. Л. І. Цвіркуна ; ДВНЗ "Нац. гірн. ун-т". - Вид. 3-тє, перероб. і допов.. - Дніпро : НГУ, 2017.. - 223 с. : рис., табл.. - Бібліогр.: с. 212-214.
- 1.1 **Основні методи та методики підготовки кваліфікаційної роботи магістрів** : навч. посіб. для студентів зі спец. 174 - Автоматизація, комп'ютерно-інтегровані технології та робототехніка / [В. П. Лисенко та ін.] ; Нац. ун-т біоресурсів і природокористування України, Каф. автоматики та робототехн. систем ім. І. І. Мартиненка. - Київ : Ямчинський О. В. [вид.], 2023.. - 462 с. : рис., табл.. - Бібліогр.: с. 334-336 .
- 1.1 **ВЕАМ робототехніка : навч. посіб.** / Невлюдов І. Ш., Євсєєв В. В., Максимова С. С. ; архів. нац. ун-т радіоелектроніки, Каф. комп'ютер.-інтегр. технологій, автоматизації та робототехніки (КІТАР). - Кривий Ріг : Чернявський Д. О. [вид.] ; Харків, 2024.. - 219 с. : рис., табл.. - Бібліогр.: с. 207-218.

2. ЕЛЕМЕНТИ КОНСТРУКЦІЙ РОБОТОТЕХНІЧНИХ СИСТЕМ

2.1 ПРИВОДИ РОБОТІВ

2.1.1. Класифікація приводів

Привід, включає передусім двигун та пристрій для його керування. Крім того, до складу привода можуть входити різноманітні механізми для передачі та перетворення руху (редуктори, перетворювачі обертального руху на поступальний і навпаки), гальмо і муфта.

До приводів, які застосовуються в роботах, висувають дуже жорсткі та специфічні вимоги. Оскільки приводи інтегруються в виконавчі системи роботів — маніпулятори та системи пересування, їх розміри та маса повинні бути мінімальними. Приводи в роботах зазвичай працюють у несталих режимах і під змінним навантаженням, причому перехідні процеси повинні бути практично без коливань.

Важливими характеристиками приводів для роботів є також їх надійність, вартість і зручність експлуатації. Вимоги до методів керування, швидкодії та точності приводів безпосередньо залежать від технічних вимог до роботів загалом, які розглянуті у параграфі 1.3.1 під час класифікації роботів. Зокрема, зазвичай потрібно, щоб середня швидкість поступального руху на виході привода становила від часток до кількох метрів на секунду, а похибка переміщення була в межах часток міліметра.

У роботах використовуються практично всі відомі типи приводів: електричні, гідравлічні та пневматичні; з поступальним і обертальним рухом; регульовані (за положенням і швидкістю) та нерегульовані; із замкненими (зворотним зв'язком) і розімкненими системами; безперервної та дискретної дії (зокрема крокові).

Застосування пневматичних приводів у робототехніці пояснюється їхньою простотою, дешевизною та надійністю. Проте вони мають низьку керованість, тому зазвичай використовуються як нерегульовані системи з циклічним керуванням. Пневматичні приводи застосовуються переважно в роботах із малою вантажопідйомністю — до 10, рідше до 20 кг.

Гідравлічні приводи є складнішими та дорожчими у порівнянні з пневматичними й електричними. Однак при потужності 500–1000 Вт і вище вони мають найкращі масо-габаритні характеристики, через що є основним типом привода для важких і надважких роботів. Завдяки гарній керованості гідравлічні приводи також застосовуються в роботах середньої вантажопідйомності, коли потрібні високі динамічні характеристики.

Електричний привід, попри його гарну керованість, простоту енергозабезпечення, вищий коефіцієнт корисної дії та зручність в експлуатації, поступається гідравлічним і пневматичним приводам за масо-габаритними характеристиками. Однак стрімкий розвиток технологій створення нових типів електричних двигунів, спеціально розроблених для робототехніки, дозволяє

створювати компактні приводи всіх необхідних типів. Завдяки цьому частка електромеханічних роботів у загальному парку роботів у світі швидко зростає. Основна сфера застосування електричних приводів у робототехніці на сьогодні — це роботи середньої вантажопідйомності (десятки кілограмів), легкі роботи з високою точністю керування, а також мобільні роботи.

2.1.2 Пневматичні приводи

Пневматичний привід однієї ступені рухомості складається з двигуна, розподільного пристрою та регулятора швидкості (рис. 2.1). Двигун може бути поступального типу (пневмоциліндр) або поворотного.

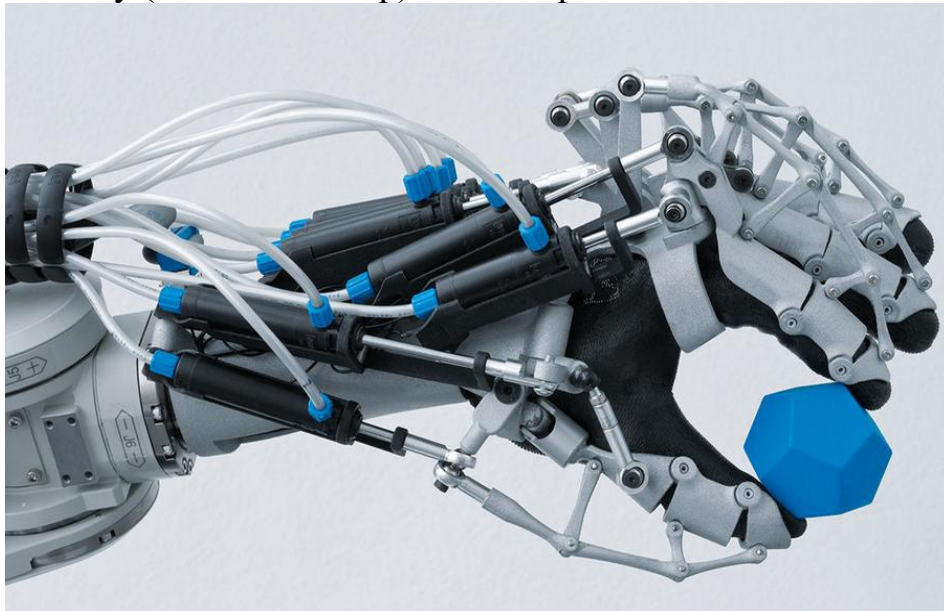


Рисунок 2.1 Пневматичний захоплювач

Пневмоциліндр включає гільзу, виготовлену з прецизійної труби, зазвичай з фторопластовим покриттям всередині, поршень із ущільненням, яке має низьке тертя по фторопласту, та шток. Часто пневмоциліндри оснащують гальмом, що спрацьовує в кінці ходу поршня. Поршень разом зі штоком, прикріпленим до навантаження, рухається під дією стисненого повітря, яке подається в одну з порожнин циліндра. Порожнина з іншого боку поршня з'єднується з атмосферою для виходу витісненого повітря. Напрямок руху поршня залежить від того, з якого боку подається стиснене повітря.

Поворотні пневматичні двигуни, що використовуються в робототехніці, мають обмежений кут повороту (неповоротні). Їх рухома частина — це лопать, закріплена на вихідному валу, яка обертається в кільцеподібному корпусі. Усередині корпусу є перегородка, яка створює дві повітряні порожнини, розділені рухомою лопаттю. Також існують поворотні двигуни, що складаються з

пневмоциліндрів і механічної передачі типу рейка-шестерня, яка перетворює поступальний рух рейки на обертання шестерні.

Розподільні пристрої контролюють подачу повітря в двигун. Вони зазвичай виготовляються на основі золотників або клапанів, оснащених електромагнітним приводом, сигнали для якого надходять від системи керування робота. Регулятори швидкості підтримують задану швидкість руху шляхом стабілізації витрати повітря, що подається в двигун (наприклад, за допомогою дроселя із зворотним клапаном).

Пневматичні двигуни працюють на повітрі, стисненому до тиску 0,3–0,6 МПа. Стиснене повітря подається до приводів від блоку живлення, який включає апаратуру для підготовки повітря і редуктор. Підготовка повітря полягає в його очищенні від вологи та механічних домішок і додаванні розпиленого мастила для змащування рухомих частин. Редуктор підтримує заданий рівень тиску повітря.

У стаціонарних системах стиснене повітря подається з пневмомережі, яка отримує його від компресора. У мобільних роботах повітря подається з балонів під високим тиском.

Більшість пневматичних приводів використовує циклічне управління. Привід рухається з початкового положення в кінцеве, яке задається механічним упором, встановленим на штоку пневмоциліндра або вихідному валу поворотного двигуна. Для плавного зупинення упор взаємодіє з демпфером, який гасить енергію руху. У малопотужних приводах застосовуються пружинні демпфери, тоді як у потужніших — гідравлічні. Інший спосіб гальмування — це протитиск, коли подача повітря перемикається таким чином, що його потік протидіє руху поршня чи лопаті, зупиняючи двигун без демпфера.

Точність циклічного управління завдяки механічним упорам може досягати менш ніж 0,1 мм. Для проміжного позиціонування використовують конструкції з декількома виступаючими упорами, що розташовані вздовж шляху штока. Наприклад, барабан з упорами, закріпленими під кутом, обертається і виставляє упори в заданих точках, забезпечуючи зупинку приводу в потрібних позиціях.

Мінімальна дискретність програмування точок позиціонування обмежується довжиною шляху гальмування, яка при швидкості 1 м/с може становити до 100 мм. Для підвищення точності в таких системах часто використовують електромагнітні гальма. Хоча позиційне управління із зворотним зв'язком за положенням теоретично можливе, через стисливість повітря точність такого управління значно поступається точності циклічного.

Одним із варіантів дискретного позиційного управління є використання послідовно з'єднаних циклічних пневмоприводів з різними довжинами ходу, які відрізняються вдвічі. Змінюючи комбінацію включення цих приводів, можна отримати сумарне переміщення, що складається з кінцевого числа дискретних значень. Наприклад, три таких приводи дозволяють реалізувати 8 точок позиціонування, а чотири — 16 точок тощо.

Пневматичні приводи з циклічним управлінням мають значні переваги:

- **Доступність енергоносія.** Там, де встановлений компресор, забезпечується постійний доступ до енергоносія — стисненого повітря. Для роботи пневмосистем достатньо лише організувати приплив повітря до компресора.
- **Простота конструкції, надійність і ремонтпридатність.** Пневматичні приводи відрізняються простотою конструкції та довговічністю. Основний компонент пневмоциліндра — це порожня гільза з кришками, поршнем і, за необхідності, штоком. На цій базовій конструкції побудовані також поворотні приводи, захвати та інші пристрої. Система управління пневматикою — це здебільшого канали з механічними перемикачами напрямку потоку повітря.
- **Можливість накопичення та тривалого зберігання енергії.** Стиснене повітря може накопичуватися і зберігатися у балонах, подібно до газових балонів для зварювання або дайвінгу, забезпечуючи тривале збереження енергії для подальшого використання.
- **Стійкість до несприятливих умов.** Пневматичні системи можуть працювати в агресивних умовах, таких як екстремальні температури, висока вологість, сильне забруднення чи хімічний вплив.
- **Стійкість до електромагнітних полів.** Пневматичні приводи зберігають свою функціональність навіть у сильних електромагнітних полях, що робить їх незамінними в умовах, де електромагнітні завади можуть впливати на роботу інших систем.
- **Стійкість до механічних впливів.** Пневмоприводи витримують значні механічні навантаження та деформації без втрати працездатності. Наприклад, при використанні пневмоприводів для затиску заготовок, немає необхідності переналаштовувати систему для обробки деталей різних розмірів.
- **Екологічна безпека.** Продукти викиду пневматики умовно ідентичні повітрю, що було використане. Проте в умовах роботи у чистих приміщеннях або при контакті з харчовими продуктами необхідно забезпечити додаткову очистку повітря.
- **Пожжево- та вибухобезпека.** На відміну від електричних або гідравлічних систем, пневматика не є джерелом займання або вибуху, оскільки стиснене повітря саме по собі не підтримує горіння.

Пневматичні приводи мають ряд суттєвих недоліків, які варто враховувати під час вибору приводу для конкретної задачі. Нижче описані ключові аспекти, що обмежують застосування пневматики.

- **Вартість стисненого повітря.** Попри те, що повітря зазвичай є безкоштовним, його стиснення потребує значних витрат на обладнання та витратні матеріали. Так придбання, експлуатація та обслуговування компресора — це значні статті витрат. У деяких випадках компресор може бути відсутній взагалі, і прокладання повітропроводів лише для одного компонента (наприклад, пневматичного захоплювача) може бути економічно невигідним. До того ж

повітряна магістраль, що включає фільтри, відводи, регулятори, та інші компоненти, потребує регулярної заміни та обслуговування.

- **Технічні особливості.** Пневматика має ряд конструктивних та експлуатаційних обмежень. Через властивість стисненого повітря точність позиціонування пневмоприводів у проміжних точках може бути низькою. Під час початкового руху приводи здійснюють ривок, а на низьких швидкостях рух взагалі може бути нерівномірним. Пневматичні приводи при однакових габаритах створюють менше зусилля, ніж гідравлічні приводи. У деяких випадках для досягнення потрібного зусилля необхідно застосовувати приводи більшого розміру, які можуть не відповідати допустимим габаритним обмеженням. Чим вищий клас очистки повітря, тим вища його вартість. Наприклад, для фарбувального обладнання потрібен сухий і очищений від вологи та пилу повітряний потік, тоді як для пневматичних приводів таких жорстких вимог немає. У таких випадках доцільно використовувати дві окремі пневмомагістралі. До того ж для виконання однакової корисної роботи пневматика споживає у 5 разів більше енергії, ніж гідравліка, і у 10 разів більше, ніж електроприводи.
- **Фізичні особливості.** Постійний різкий шиплячий звук повітря, що виходить з системи, є невід'ємною частиною роботи пневмоприводів. Такий шум може спричинити травми слуху для персоналу. Також потрібно враховувати, що усунення витоків в пневматиці є ускладненим оскільки на відміну від гідравліки, де виток можна виявити візуально, у пневматиці виток через з'єднання часто є невидимими і потребують застосування спеціальних приладів для їх виявлення.

2.1.3. Гідравлічні приводи

Гідроприводи знайшли основне застосування у важких і надважких роботах, а також у роботах середньої вантажопідйомності, коли потрібне особливо якісне управління (рис. 2.2). Гідропривод складається з тих самих основних частин, що й пневмопривод. Основою його є двигун поступального руху (гідроциліндр) або кутового руху (поворотний гідродвигун), які за своєю будовою подібні до аналогічних пневматичних двигунів, проте замість стисненого повітря в них використовується рідина (зазвичай масло) під тиском до 20 МПа. Це дозволяє створювати зусилля на два порядки більше. На відміну від пневмоприводів, гідроприводи мають власний блок живлення, що входить до складу робота. Цей блок включає гідронасос, фільтр, регулятор тиску, пристрій охолодження (зазвичай водяний) та масляний акумулятор із запасом масла.

У робототехніці широко застосовуються гідравлічні приводи з електричним керуванням, які поєднують переваги як електричних, так і гідравлічних елементів. У таких системах керуючий електричний сигнал за допомогою електромагнітного перетворювача або електродвигуна (постійного струму чи крокового) трансформується в переміщення золотника, який регулює потік рідини в гідростанції.

Гідравлічні приводи застосовуються в роботах із такими типами керування: цикловим, позиційним та контурним.



Рисунок 2.2 Промисловий робот з гідравлічним приводом

Гідроприводи з цикловим керуванням. У цьому типі приводів електромагнітний перетворювач і золотниковий розподільник працюють у двох положеннях («увімкнено-вимкнено» або «вперед-назад»). Разом вони утворюють електрогідравлічний клапан, який спрямовує потік робочої рідини без регулювання швидкості руху гідродвигуна.

Гідроприводи з позиційним і контурним керуванням. У таких системах використовуються дроселюючі золотникові розподільники. Вони не лише розподіляють потік рідини за напрямками, але й регулюють швидкість руху гідродвигуна.

У позиційних гідроприводах для керування золотником використовується кроковий електропривод, що складається з крокового електродвигуна та схеми управління.

У слідкувальних (позиційних і контурних) гідроприводах золотник переміщується за допомогою двигуна постійного струму, яким управляє сигнал регулятора через електронний підсилювач. У такому випадку дроселюючий золотниковий розподільник і гідродвигун формують гідравлічний підсилювач моменту електродвигуна, утворюючи комплектний електрогідравлічний привод робота.

Гідроприводи всіх ступенів рухливості разом із додатковим обладнанням, таким як гідравлічні насосні станції, трубопроводи, запобіжні та перепускні клапани, гідравлічні акумулятори, розподільники, дроселі тощо, утворюють автономну гідравлічну систему робота.

Основні переваги гідроприводів:

- Висока питома потужність гідропривода – це передавана потужність, що припадає на одиницю сумарної ваги елементів. Цей параметр у гідравлічних приводів у 3–5 разів вищий, ніж у електричних, причому ця перевага зростає зі збільшенням передаваної потужності.
- Відносна простота забезпечення безступеневого регулювання швидкості вихідного ланцюга гідропривода у широкому діапазоні.
- Висока швидкодія гідропривода. Операції запуску, реверсу та зупинки виконуються гідроприводом значно швидше, ніж іншими приводами. Це обумовлено малим моментом інерції виконавчого органу гідродвигуна (момент інерції обертових частин гідромотора у 5–10 разів менший, ніж у електродвигуна).
- Високий коефіцієнт підсилення гідропідсилювачів за потужністю, який досягає значення 10^5 .
- Порівняльна простота виконання технологічних операцій у заданому режимі, а також можливість простого та надійного захисту приводного двигуна та елементів гідропривода від перевантажень.
- Простота перетворення обертального руху в поступальний.
- Гнучкість компонування агрегатів гідропривода.
- Можливість підключення до гідропривода будь-якого гідравлічного обладнання, такого як відбійні молотки, дискові пилки, різноманітні ковші та захвати.
- Слабкий вплив вібрацій на руки оператора.

Основні недоліки гідроприводів

Незважаючи на вищезгадані переваги гідроприводів, під час їх проектування чи вирішення питання про доцільність їх використання варто враховувати й недоліки, властиві цьому типу приводів. Ці недоліки здебільшого обумовлені властивостями робочого середовища (рідини).

- Порівняно невисокий ККД гідропривода та великі втрати енергії під час її передачі на великі відстані.
- Залежність характеристик гідропривода від умов експлуатації (температура, тиск). В'язкість робочої рідини залежить від температури, а низький тиск може спричинити кавітацію в гідросистемі або виділення з рідини розчинених газів.
- Чутливість до забруднення робочої рідини та необхідність високої культури обслуговування. Забруднення рідини абразивними частинками призводить до швидкого зношування елементів прецизійних пар у гідравлічних агрегатах і їх виходу з ладу.
- Зниження ККД і погіршення характеристик гідропривода у міру вироблення ним або його елементами експлуатаційного ресурсу. Насамперед це стосується зношування прецизійних пар, що призводить до збільшення зазорів у них, зростання витоків рідини та зменшення об'ємного ККД.

2.1.4 Електропривід промислових роботів

Електроприводи промислових роботів повинні відповідати специфічним вимогам, зумовленим технологічними процесами їх роботи та конструктивними особливостями.

Основним робочим циклом робота є позиціонування. Швидкість переміщення може досягати 30 м/хв, а точність позиціонування — 0,01 мм.

Режим роботи електроприводів передбачає короткочасне повторне включення з частими пусками та зупинками. Оскільки в роботі відсутні процеси різання, коливання навантаження відсутні.

Хоча потужність приводів зазвичай невелика (не більше 1 кВт), вона повинна бути достатньою для забезпечення великого моменту при низьких швидкостях.

З технологічної точки зору, електропривід робота повинен мати широкий діапазон регулювання, здатний витримувати значні перевантаження в момент пуску (короткий час для досягнення максимальної швидкості і швидке гальмування). Для отримання великих моментів у механізми робота часто вбудовують редуктори або розробляють спеціальні двигуни. Конструктивно промисловий робот є складним пристроєм з різноманітними кінематичними ланцюгами, що взаємно впливають один на одного. Механічні передачі мають низьку жорсткість, моменти інерції змінюються, іноді перевищуючи власний момент інерції двигуна. Переміщення робочого органу робота, разом з приводом, вимагає використання двигунів з малими розмірами та масою. Отже, через конструктивні особливості, електропривід робота повинен мати високі динамічні характеристики для виконання складних алгоритмів управління. Виконавчий двигун повинен швидко реагувати на команди управління та мати компактні габарити і малу масу.

Завдяки високим експлуатаційним характеристикам, електричні машини з постійними магнітами (вентильні двигуни) є найбільш перспективними в діапазоні малих та середніх потужностей, зокрема для моментних систем електроприводу. Вони мають просту конструкцію, високу надійність, жорсткі механічні характеристики та не потребують енергії для збудження, що робить їх ідеальними для використання у промислових роботах та маніпуляторах.

Електроприводи використовуються як для безперервної, так і для дискретної роботи (крокові двигуни). Порівняно з пневматичними та гідравлічними приводами, електроприводи мають переваги, зокрема відсутність трубопроводів, легкість монтажу та налаштування, а також простоту в експлуатації. Нещодавно з'явилися універсальні електромеханічні модулі для окремих типів рухів (підйом, обертання тощо), з яких можна скласти різні виконувані пристрої роботів.

У промислових роботах застосовуються автоматизовані електроприводи замкнутого типу з двигунами постійного струму.

У розімкнутому електроприводі можна виділити три основні частини:

1. Механічна частина приводу, яка включає робочий механізм або виконавчий орган та передавальний механізм (редуктор). У деяких випадках механічну частину доповнюють гідроусилювачем моментів, коли електродвигун має малу потужність, що відносить схему до електрогідравлічної.
2. Електродвигун, який може бути шаговим дискретним електродвигуном, що використовується в розімкнутих системах.
3. Система управління, яка складається з електронного комутатора, що перетворює вхідний сигнал в відповідну напругу, та підсилювача потужності, що забезпечує відповідний струм.

Кроковий дискретний розімкнутий привод з різними типами крокових двигунів широко застосовується у промислових роботах. Цей привод не потребує датчиків зворотного зв'язку, що спрощує його структуру та функціональну схему. Крок двигуна можна вибирати за умовами необхідної точності, що дозволяє кроковому приводу відтворювати всі типи механічних рухів, доступні безперервним системам.

Завдяки зазначеним характеристикам, дискретний привод з кроковим двигуном можна визначити як синхронно-імпульсний слідкуючий привод, що поєднує можливості глибокого частотного регулювання кута швидкості з можливістю числового задання шляху.

Інші типи розімкнутих електроприводів поки що не знайшли широкого застосування в промислових роботах.

Замкнуті електроприводи набувають все більшого застосування в промислових роботах завдяки своїм очевидним перевагам. Для промислових роботів потрібні малогабаритні електродвигуни потужністю від 100 Вт до 2 кВт. Наразі розробляються двигуни з різними типами роторів (наприклад, з печатним або гладким ротором). Бажано, щоб такі двигуни мали вбудовані датчики швидкості, такі як тахогенератори, що дозволяє створити більш компактну конструкцію приводу.

2.1.5 Штучні м'язи

Важливим біонічним напрямом у робототехніці є створення приводів типу штучні (технічні) м'язи. Сучасні приводи, що використовуються в робототехніці, за своїми масогабаритними параметрами значно поступаються поперечно-смугастим м'язам тварин та людини. Оскільки параметри приводів у значній мірі визначають масогабаритні та енергетичні характеристики роботів в цілому, це обумовлює інтерес до проблеми створення технічних аналогів таких м'язів.

Однією з очевидних причин незадовільних масових параметрів сучасних приводів порівняно з м'язами є використання в них металів, зокрема ферромагнітних, у електричних двигунах. Тому перші розробки приводів, які отримали назву штучний м'яз, були пневматичними приводами, де циліндр з поршнем замінювався еластичною трубкою (гума, полімери) в оплетці,

переплетеній хрест-навхрест (нейлон). При подачі повітря під тиском в трубку оплетка змушує трубку роздуватися, що призводить до її скорочення на величину до однієї третьої довжини, імітуючи роботу м'яза. Такі приводи мають в 3-4 рази меншу масу, ніж пневматичні циліндри при тій же потужності, тому вони знайшли застосування, зокрема, в протезуванні кінцівок. Джерелом стисненого газу зазвичай служать одноразові балончики.

На рис.2.3 показано пристрій реверсивного приводу такого типу, застосованого в промисловому роботі "Софторм" компанії "Бріджстоун" (Японія). Пристрій складається з двох пневматичних приводів односпрямованої дії. При підвищенні тиску в одному з них та одночасному зниженні на ту ж величину в іншому перший привід скорочується, а другий подовжується. В результаті через трос відбувається обертання блоку, який з'єднаний з ланкою маніпулятора. Кут повороту блоку практично прямо пропорційний різниці тисків у приводах. Вантажопідйомність шарнірного маніпулятора з п'ятьма ступенями свободи робота "Софторм" — 3 кг при масі 5,5 кг. Похибка позиціонування — 1,5 мм.

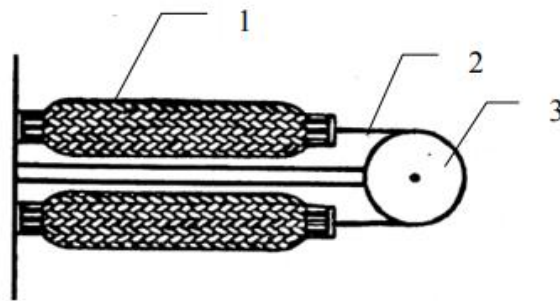


Рисунок 2.3. Реверсивний гнучкий пневматичний привід компанії «Бріджстоун», де: 1 – гнучкий пневматичний привід; 2 – трос; 3 – блок.

На рис. 2.4 показаний варіант пневматичної штучної м'язи, в якій газ під тиском генерується безпосередньо в самій трубці в результаті нагріву електричним струмом. При проходженні струму через термоелемент наповнювач нагрівається та виділяє газ. В результаті всередині трубки підвищується тиск, і вона, роздуваючись, скорочується по довжині. Після вимкнення струму наповнювач охолоджується, знову поглинаючи виділений газ, і м'яз повертається в початковий стан.

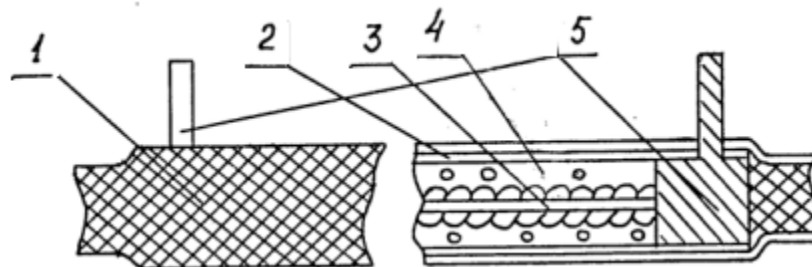


Рисунок 2.4. Сорбційна штучна м'язи, де: 1 – хордовий оболонка; 2 – еластична трубка; 3 – термоелемент; 4 – наповнювач; 5 – електричні виводи.

Довжина м'яза – 150-300 мм, діаметр 3-5 мм. Розмір скорочення – 10-15 %. Розвивається зусилля – 100-300 гр., при масі м'яза – одиниці грамів. Головний недолік цього м'яза – дуже низька швидкодія: час стиснення більше 30 с, а зворотне розслаблення ще в 2-3 рази більше.

В середньому описані пневматичні та гідравлічні штучні м'язи порівняно з аналогічними традиційними приводами (циліндри) в 3 рази легші, в два рази менші за габаритами та розвивають в десятки разів більше зусилля на одиницю ваги.

Окрім реалізованих типів пневматичних та гідравлічних штучних м'язів, в світі активно проводяться дослідження та розробки штучних м'язів, заснованих на інших фізичних принципах. Найбільш близькі до розглянутих пристроїв – це розробки електромагнітних та електростатичних штучних м'язів.

Електромагнітна штучна м'яза складається з багатовиткової обмотки, стиснутої в джгут, в якому провідники з електричним струмом протилежного напрямку притискаються один до одного. При пропусканні струму через них вони, завдяки електромагнітному силовому взаємодії, розходяться в поперечному напрямі. У результаті довжина джгута скорочується подібно до скорочення пневматичних і гідравлічних м'язів, описаних раніше.

Електростатична штучна м'яза має форму циліндра, по осі якого розташований центральний електрод, а по утворюючим циліндра з'єднані між собою гнучкі периферійні електроди. При подачі електричного потенціалу на центральний і периферійний електроди виникає електрична сила, що відштовхує периферійні електроди від центрального. В результаті периферійні електроди згинаються, викликаючи скорочення м'яза. Інший варіант електростатичної м'язи складається з стовпчика, що складається з тонких металевих дисків (плівок), розділених еластичними ізолюючими прокладками. Всі непарні та всі парні диски з'єднані між собою. При подачі на них електричної напруги різної полярності диски зближаються, стискаючи ізолюючі прокладки, і м'яз скорочується. Замість ізолюючих прокладок може використовуватися газ.

Наступний тип штучних м'язів – це м'язи на іонізованих полімерних гелях та іонних полімерно-металевих композитних матеріалах. В електричному полі молекули цих матеріалів іонізуються і піддаються механічним силам, які викликають деформацію (вигин) матеріалу в напрямку, перпендикулярному до градієнта електричного поля. Розробляються також штучні м'язи на основі різних штучних волокон, що деформуються під дією хімічних реагентів. Для всіх цих матеріалів характерне низьке швидкодія (секунди) і величина ходу (деформації) порядку 10-15 %.

Замість описаних пневматичних м'язів з тепловим впливом, ще один варіант таких пневматичних м'язів – це заповнення трубки легкокиплячою рідиною з зовнішнім тепловим впливом. В якості прикладу можна привести рішення від компанії Clone Robotics (рис. 2.5).



Рисунок 2.5. Приклад штучних м'язів компанії Clone Robotics

По суті, вони запропонували сітчасті трубки з кульками всередині. У міру розширення кульки, зазвичай приводимої в рух пневматичним або гідравлічним насосом, що знаходиться зовні м'яза, її радіальне розширення змушує сітку скорочуватися в подовжньому напрямку. Компанія не хотіла використовувати громіздкі зовнішні насоси; команді потрібна була м'яза, до якої можна було б просто прикласти електричний струм і змусити її скорочуватися достатньо контрольованим способом. Тому вони придумали тримати кульки, заповнені рідиною, і пропускати через них потужний нагрівальний елемент. При подачі струму елемент швидко кип'ятить рідину — у випадку з ацетальдегідом, підвищуючи його тиск від атмосферного при 20 °С до 6,6 разів більшого при 70 °С.

Нажаль принциповим недоліком такої конструкції є складність охолодження для повернення системи у вихідний стан – тобто забезпечення швидкодії ускладнено.

2.2 СПЕЦІАЛІЗОВАНІ ВИКОНАВЧІ МЕХАНІЗМИ

2.2.1 МАНІПУЛЯТОРИ

Промисловий пристрій, яким людина може керувати в ручному режимі, називається роботом-маніпулятором. Такі пристрої можуть функціонувати як самостійні одиниці або бути частиною складних роботизованих комплексів. Маніпулятор складається з вузлів, кожен з яких виконує певні функції: одні забезпечують обертальні рухи, інші – поступальні.

Сучасні технології досягли високого рівня розвитку, і на виробництвах усе частіше встановлюють роботів, тим самим зменшуючи потребу в ручній праці. У найскладніших виробничих процесах роботи можуть виконувати як керуючі, так і рухові функції. Роботи-маніпулятори здатні замінити кілька десятків фахівців, що робить їх особливо актуальними. Використання таких роботів дозволяє автоматизувати практично будь-який промисловий процес. Завдяки роботам-маніпуляторам можна знизити кількість помилок у виробничому процесі, зменшити кількість бракованої продукції, скоротити втрати сировини та підвищити рівень безпеки праці. Робототехніка дозволяє оптимізувати підприємства під різні технології, забезпечити необхідний рівень продуктивності та безпеки.

Типи роботів-маніпуляторів та їх характеристики

Оскільки загальної класифікації роботів-маніпуляторів не існує, їх умовно поділяють на кілька типів залежно від технологічного призначення, конструктивних особливостей та інших параметрів.

За типом монтажу роботизоване обладнання може бути мобільним або стаціонарним. Мобільні роботи-маніпулятори вирізняються компактними розмірами, що дозволяє легко транспортувати їх і долати перешкоди під час роботи. Однак такі пристрої мають високу вартість і обмежену вантажопідйомність, тому підходять лише для роботи з легкими об'єктами. Натомість стаціонарні роботи-маніпулятори є найпоширенішими завдяки великому радіусу дії та високій вантажопідйомності. Вони можуть бути закріплені на підлозі, стелі або стіні, що забезпечує гнучкість їхнього використання в різних виробничих умовах.

За орієнтацією в просторі роботи-маніпулятори поділяються на горизонтальні та вертикальні. Горизонтальні пристрої можуть мати довжину до кількох десятків метрів і застосовуються у виробничих зонах, де потрібно обслуговувати кілька об'єктів одночасно. Вертикальні роботи-маніпулятори використовуються в умовах обмеженого простору, де важливо ефективно задіяти вертикальну площину.

Вибір типу робота-маніпулятора визначається конкретними завданнями, доступним робочим простором і вимогами до вантажопідйомності та мобільності.

За типом застосування:

Автономні (програмовані) роботи-маніпулятори можуть розташовуватися поруч із оброблюваними об'єктами або працювати спільно з іншими програмованими пристроями. Якщо оснастити такого робота колесами з всеспрямованим рухом, він зможе пересуватися в будь-які важкодоступні для людини місця без її участі.

Роботи-маніпулятори з ручним управлінням забезпечують високу точність у виготовленні деталей, що робить їх незамінними для виконання складних завдань, які вимагають ювелірної точності.

Колаборативні маніпулятори (коботи) створені для спільної роботи з людьми. Вони мають жорсткі вимоги до конструкційної безпеки та програмного забезпечення, що дозволяє мінімізувати ризики травмування працівників.

По корисному навантаженню та робочому простору.

Умовно колаборативні роботи поділяються по корисному навантаженню на 5 груп 0 – 20 кг, 20 – 80 кг, 80 – 300 кг, 300-1000 кг та 1000 – 3000 кг.

Маніпулятори можуть мати різну кількість осей, тобто різну кількість площин в яких можуть здійснювати переміщення. Ця можливість забезпечується наявністю електромоторів або сервоприводів усередині кожного робота. На рис. 2.6 можна побачити, як саме може рухатися шестивісний маніпулятор.

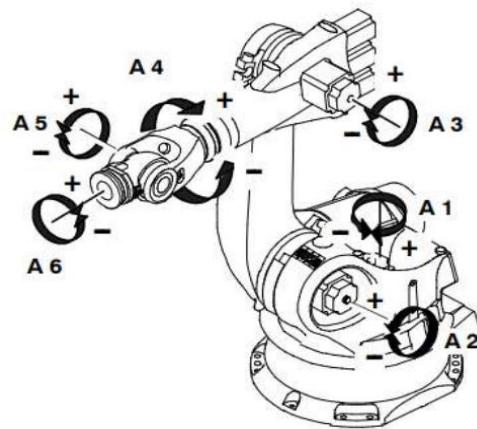


Рисунок 2.6 Можливі переміщення маніпулятора.

Доступний робочий простір може суттєво змінюватись в залежності від моделі маніпулятора. На рис. 2.7 приведено опис робочих просторів маніпуляторів KUKA KR QUANTEC. Номінальне корисне навантаження / кількість осей для KR 150 R3100-2 F становить 150 кг. / 6, а для KR 240 R3200-2 PA - 240 кг. / 6 відповідно. Робочі простори спроектовано під типові операції притаманні маніпуляторам і проектувальники можуть підібрати оптимальні моделі під власні потреби.

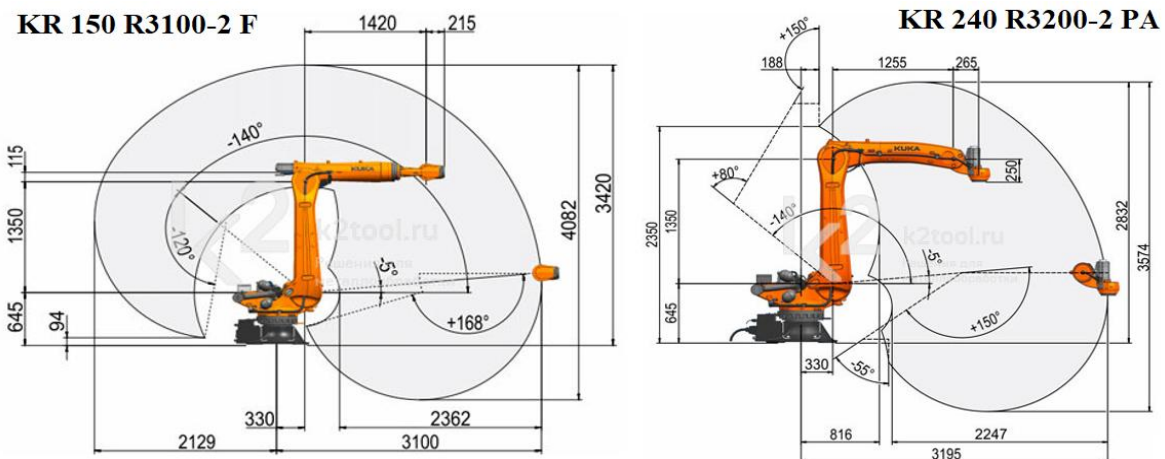


Рисунок 2.7 Схеми робочих просторів маніпуляторів **KUKA KR QUANTEC**

По типу виконуваних функцій.

У сфері збірки та розбирання роботи-маніпулятори дозволяють виконувати збирання виробів будь-яких розмірів без участі людини, що зменшує потребу у використанні обладнання з великою вантажопідйомністю. Для пайки та зварювання роботи забезпечують високу якість зварних швів і стабільність дуг, значно підвищують швидкість роботи та дозволяють використовувати низький струм, що неможливо при ручному зварюванні.

Під час обробки матеріалів роботи-маніпулятори забезпечують ідеально гладкі поверхні та підвищують міцність металу завдяки застосуванню холодної ковки, яка зміцнює його стінки більш ніж на 30%. Вони також ефективно використовуються для очищення поверхонь за допомогою водяного струменя під високим тиском, піскоструминної обробки або фарбування готової продукції.

Процедури різання та обробки, які є небезпечними для людини, роботи виконують абсолютно безпечно. Завдяки високій точності різання скорочується час обробки та мінімізуються втрати матеріалу. У будівельній галузі роботи-маніпулятори дозволяють значно прискорити процеси зведення об'єктів, забезпечуючи при цьому максимальну точність. Вони працюють безперервно, не залежать від погодних умов та технічних особливостей процесу.

Таким чином, роботи-маніпулятори знаходять застосування в різних сферах, забезпечуючи ефективність, безпеку та якість виконання завдань.

По типу приводу

Гідравлічні роботи-маніпулятори використовують рідину для приведення в рух виконавчого механізму. Вони застосовуються для переміщення вантажів масою понад 100 кг, що робить їх незамінними для важких робіт.

Електричні роботи-маніпулятори працюють завдяки електричному струму. Вони вирізняються високою продуктивністю та точністю виконання завдань, що робить їх універсальними в багатьох сферах.

Пневматичні роботи-маніпулятори використовують енергію стисненого повітря, яке подається в пневматичну систему за допомогою компресора. Цей тип маніпуляторів зазвичай застосовується для легких і швидких операцій.

Залежно від технічного завдання робот може бути оснащений різними виконавчими механізмами.

2.2.2 ЗАХОПЛЮВАЧІ

Типи захоплень у роботів-маніпуляторів

Захоплюючі пристрої (ЗП) призначені для захоплення та утримання об'єктів маніпулювання. Сучасні промислові роботи комплектуються стандартними захоплюючими пристроями, однак вони можуть бути спроектовані індивідуально для кожного випадку. Захоплюючі пристрої повинні забезпечувати надійне захоплення та утримання деталей, що мають різну масу, розміри та конструкцію (в

межах встановлених параметрів робота); стабільність базування; швидкість переналаштування; і при цьому не повинні пошкоджувати деталі у місці захоплення.

Конструкція захоплюючих пристроїв залежить від системи ЧПУ, типу обслуговуваного обладнання, типу робота та серійності виробництва. Наприклад, робот, який обслуговує групу верстатів у серійному виробництві, повинен мати захоплюючі пристрої з широким діапазоном або автоматичною зміною захоплення. Захоплюючі пристрої для обслуговування патронних токарних верстатів відрізняються конструкцією від тих, що обслуговують центровані верстати.

Класифікація захоплюючих пристроїв промислових роботів проводиться за різними ознаками. Розглянемо деякі з них. За принципом дії розрізняють механічні, магнітні, вакуумні захоплюючі пристрої, а також пристрої з еластичними камерами, які деформуються під впливом повітря або рідини, що нагнітаються всередину (рис. 2.8).

Захоплюючі пристрої всіх типів можуть бути одно-, дво- або багатозахоплюючими.

Залежно від характеру базування, захоплюючі пристрої поділяються на центруючі (визначають положення осі або площини симетрії захопленої заготовки), базуючі (визначають положення базової поверхні), перебазовувальні (переорієнтують об'єкт) та фіксуючі (утримують об'єкт у тому положенні, яке він мав на момент захоплення).

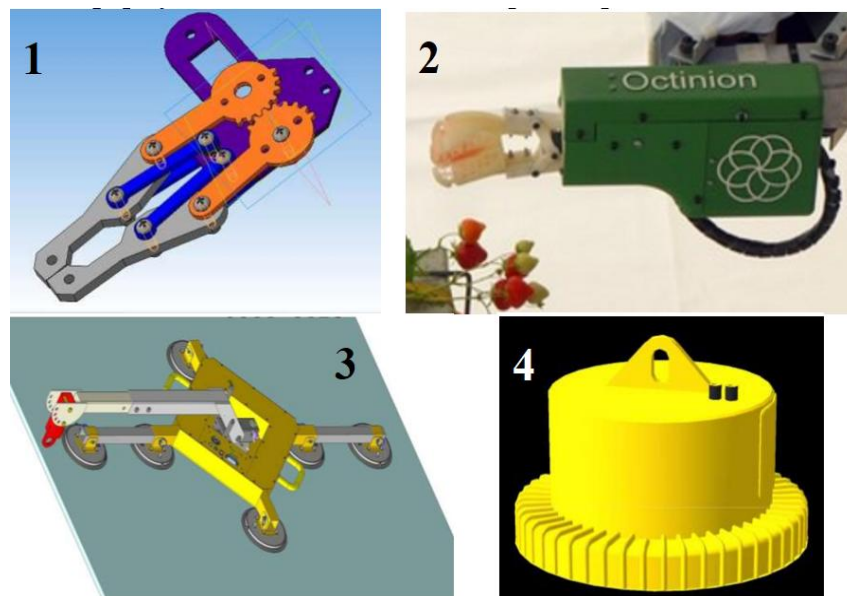


Рисунок 2.8 Захоплюючі пристрої, де: 1 – механічне жорстке, 2 – механічне еластичне (виробляється компанією Octinion для збору полуниці), 3 – вакуумний, 4 – магнітний.

Що стосується характеру кріплення, захоплюючі пристрої можуть бути незмінними, змінними, швидкозмінними або з автоматичною зміною.

Залежно від виду управління, захоплюючі пристрої поділяються на некеровані, командні, жорсткопрограмовані та адаптивні. До некерованих відносяться, наприклад, захоплюючі пристрої з постійними магнітами, де для звільнення заготовки необхідно застосувати силу, більшу, ніж сила утримання. Командні захоплюючі пристрої отримують лише команди на захоплення або відпускання об'єкта. Жорсткопрограмовані захоплюючі пристрої управляються програмованим пристроєм управління (ПУ), який визначає в залежності від програми величину переміщення губок, силу зажиму, взаємне розташування робочих елементів тощо. Адаптивні захоплюючі пристрої також управляються ПУ, але оснащені датчиками, які дають інформацію про об'єкт та зовнішнє середовище, наприклад, про форму поверхні об'єкта маніпулювання або про зусилля, що виникає в місці захоплення.

Існують також "антропоморфні" захоплюючі пристрої, які імітують кисть руки людини. Наприклад, трьохпальцевий пристрій такого типу завдяки згинанню "фаланг пальців" може забезпечити 11 ступенів рухливості. Кожне з'єднання управляється окремо і приводиться в рух електродвигунами постійного струму. Такі пристрої можуть захоплювати та базувати деталі різних розмірів та форм.

Розглянемо конструкції захоплюючих пристроїв різних типів. Механічні захоплюючі пристрої промислових роботів є найбільш поширеними. Їх класифікують за типом приводу (пружинні, пневматичні, гідравлічні, електромеханічні), за типом губок (жорсткі, регульовані, гнучкі або пружні), за видом передавального механізму (важільні, рейкові, клинові). Механічні захоплюючі пристрої з пневмоприводом прості, зручні, не мають витоків, але при тих самих габаритах з гідроприводом останній забезпечує значно більшу силу захоплення. Пневмопривод, на відміну від гідроприводу, не дозволяє точно регулювати силу затискування.

Захоплюючі пристрої з електромеханічним приводом зустрічаються рідше. Для заготовок невеликої маси та габаритів можуть використовуватися захоплюючі пристрої одностороннього дії, у яких губки відкриваються або закриваються за допомогою пружини.

Щоб уникнути пошкодження деталі під час затискання, часто використовують гнучкі, еластичні або пристрої, які рівномірно розподіляють зусилля затиску. Для захоплення крихких об'єктів застосовуються захвати у вигляді надувних подушок або надувних пальців (рис. 2.9 а). Пальці виготовляються з цілісної гуми та мають поступовий перехід від тонкостінної частини 4 до гофрованої 3 і товстостінної 2 (рис. 2.9 б). При подачі повітря через канал 1 тонкостінна частина пальця подовжується більше, ніж гофрована чи товстостінна, що спричиняє деформацію пальців уздовж лінії 5 та затискання деталі.

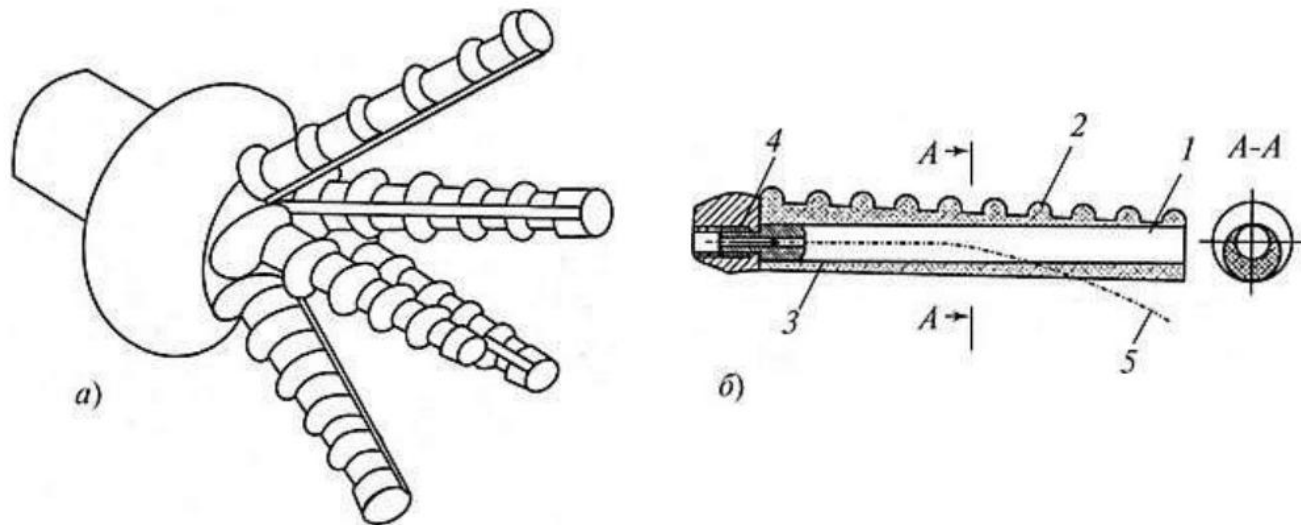


Рисунок 2.9 Захватний пристрій із надувними пальцями

Аналіз існуючих конструкцій захватних пристроїв промислових роботів, які використовуються в приладобудуванні та машинобудуванні, свідчить про їх значне різноманіття, адже вони розробляються для конкретних деталей. За типами такі пристрої розподіляються наступним чином: механічні – 67%, вакуумні – 12%, магнітні – 4%, спеціалізовані – 17%.

2.2.3 АНАЛІЗ РИНКУ РОБОТІВ МАНІПУЛЯТОРІВ

За даними Research and Markets, обсяг світового ринку промислових роботів у 2018 році досяг 16,9 мільярда доларів (без урахування вартості компонентів, програмного забезпечення та системної інтеграції). Передбачається, що середній темп зростання ринку за останні п'ять років становив 12%, а в грошовому вимірі – 31,7 мільярда доларів. У 2018 році було поставлено приблизно 420 тисяч роботів, і цей показник, як очікується, щороку зростатиме на 13%, досягнувши 774 тисяч одиниць через два роки.

Загальний обсяг світового ринку роботизованих пристроїв у 2018 році оцінювався приблизно в 46 мільярдів доларів, і прогнозується, що через два роки ця цифра досягне майже 70 мільярдів, зі щорічним приростом у 7,9%. Однак у 2019 році Research and Markets переглянули свої підходи до оцінювання, внаслідок чого ринок індустріальних роботів у 2018 році оцінено в 48,7 мільярда доларів. Очікується, що за шість років цей показник зросте до 75,7 мільярда доларів із середньорічним темпом зростання у 10%.

З отриманих даних простежується тенденція до зменшення кількості висококваліфікованих працівників на виробництвах, що веде до подальшої автоматизації із залученням колаборативних роботів.

Наприкінці 2020 року Міжнародна федерація робототехніки (International Federation of Robotics, **IFR**) оцінила, що кількість промислових роботів, встановлених на виробництвах за чотири роки до цього, становила 422 271

одиницю, а загальний ринок досяг майже 17 мільярдів доларів (без урахування компонентів та системних розробок). Попри спад продажів промислових роботів на 12% після 2018 року, загальна кількість роботів, які використовуються на виробництвах, наблизилася до трьох мільйонів одиниць.

Згідно з даними IFR, найбільша кількість промислових роботів встановлюється в країнах Азії – у 2019 році цей показник склав 285 тисяч одиниць. Для порівняння, експорт роботів до Європи становив 27% від азійських поставок (приблизно 74 тисячі одиниць), а до країн Америки – 19% (близько 54 тисяч одиниць).

Найбільші та найвідоміші компанії виробники роботів на світовому ринку:

- **FANUC Ltd.**, (Японія) є однією з найбільш впливових робототехнічних компаній, що спеціалізується на виробництві та монтажі промислових систем а також промисловій автоматизації та лазерних системах. FANUC займається роботизацією заводських підприємств уже понад 60 років. На виробничих лініях компанії працюють близько 2 тисяч роботів, які збирають та випробовують промислові системи. Особливість компанії полягає в тому, що вона самостійно розробляє та виготовляє всі ключові елементи своїх систем — від апаратного забезпечення до програмного. Спочатку FANUC був частиною японської компанії Fujitsu (назва розшифровується як Fuji Automatic Numerical Control), проте згодом розширився до американського підрозділу FANUC America Corporation і європейського FANUC Europe Corporation. FANUC входить до складу концерну Furukawa, Японія.
- **ABB Robotics**. Публічна компанія, що спеціалізується на промислових роботах та маніпуляторах. Штаб-квартира розташована в Цюриху, Швейцарія. Провідний виробник промислових роботів, модульних виробничих систем та постачальник послуг.
- **KUKA Robotics Corporation**. Виробник промислових роботів і робототехнічних систем, що також надає послуги з системної інтеграції. Штаб-квартира компанії знаходиться в Німеччині. Близько 95% акцій належать китайській промисловій групі Midea.
- **Yaskawa Motoman Robotics**, Японія. Розробляє промислові роботи та роботи для освітніх цілей у сфері промислової робототехніки. Спеціалізується на збиральних, фарбувальних, зварювальних, пакувальних та інших роботах. Надає послуги з системної інтеграції.

Китай став найбільшим ринком для промислових роботів з 2017 року, що спричинило появу значної кількості виробників у цій галузі. За даними IFR абсолютна більшість промислових роботів припадає саме на Китай (рис.2.10).

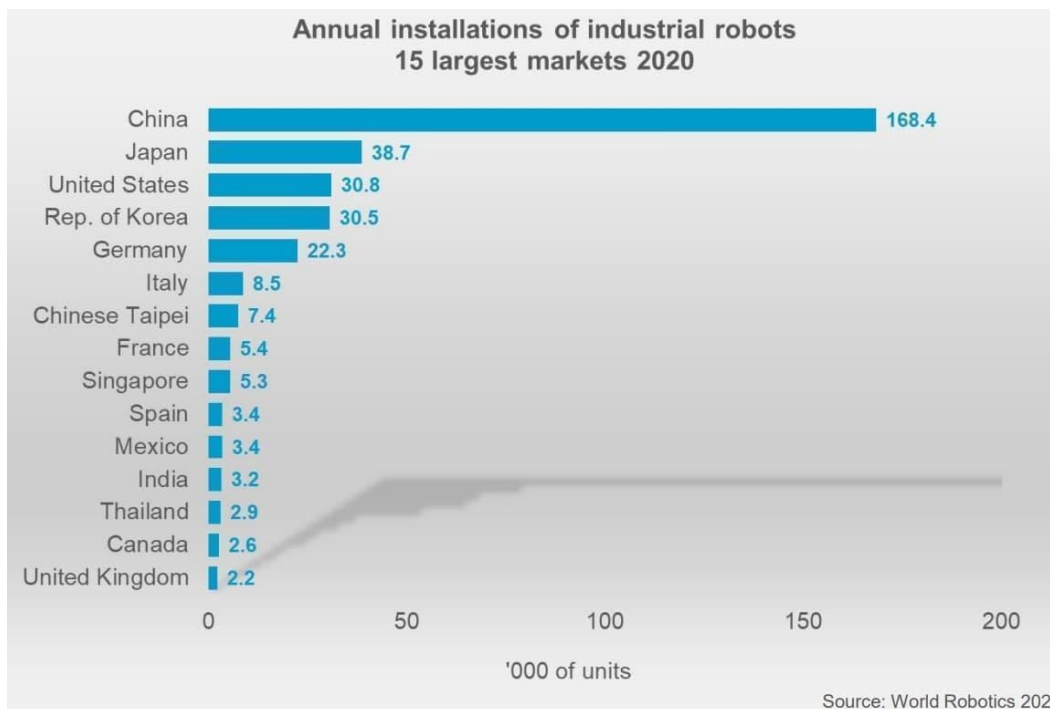


Рисунок 2.10 Розподіл промислових роботів по країнам світу станом на 2020 рік.

Ринок дуже великий і відповідно місцеві виробники приклади зусилля для заповнення цієї ніши. Ось список п'яти провідних виробників промислових роботів у Китаї:

QJAR була заснована у 2013 році та знаходиться під управлінням компанії Zhejiang ASD Electric Co., Ltd. Це високотехнологічне підприємство в галузі робототехніки, яке володіє незалежними правами на інтелектуальну власність та основними технологіями. Компанія спеціалізується на виробництві корпусів і ключових компонентів промислових роботів, включаючи контролери, сервоприводи, системне програмне забезпечення, редуктори, дизайн корпусів, системи машинного зору та інші основні технології.

SIASUN була заснована у 2000 році та входить до складу Китайської академії наук. Це високотехнологічна компанія, основним напрямом діяльності якої є робототехніка. На сьогодні SIASUN створила повноцінний виробничий ланцюг, що об'єднує незалежні основні технології, ключові компоненти, базові продукти та галузеві системні рішення. Як провідний гравець у робототехнічній індустрії Китаю, SIASUN впровадила понад сто інновацій у цій галузі. Компанія успішно розробила більше ста продуктів у п'яти основних серіях: промислові роботи, колаборативні роботи, мобільні роботи, спеціалізовані роботи та сервісні роботи, при цьому всі продукти мають незалежні права на інтелектуальну власність.

Компанія **ESTUN** була заснована у 1993 році та в 2015 році зареєстрована на Шеньчженьській фондовій біржі (код акції 002747). Сьогодні вона має професійну команду високого рівня для досліджень і розробок, а також технічні переваги завдяки синхронній розробці з передовими світовими технологіями промислових

роботів. Компанія пропонує повний асортимент промислових роботів, включаючи серії роботів Delta і Scara, із стандартними технічними характеристиками, які варіюються від 6 до 450 кг. Сфери застосування включають: точкове зварювання, дугове зварювання, завантажувально-розвантажувальні роботи, обслуговування обладнання тощо.

Компанія **EFORT** зареєстрована в Раді з інновацій у сфері науки і технологій. Вона є віце-головою Китайського альянсу робототехніки, входить до десятки найкращих робототехнічних підприємств Китаю та сотні провідних інтелектуальних виробничих підприємств країни. EFORT займає провідну позицію в індустрії промислових роботів і може надавати клієнтам та партнерам як промислових роботів, так і міжгалузеві рішення для інтелектуального виробництва.

ЖАКА Robot — глобальний новатор у сфері колаборативних роботів. Заснована у 2014 році, назва «ЖАКА» (Just Always Keep Amazing) перекладається як «завжди залишатися досконалыми». Це високотехнологічне підприємство, яке займається інноваційними дослідженнями та розробкою нового покоління колаборативних роботів і розумних фабрик. Робот ЖАКА успадкував напрацювання в галузі основних технологій і ген досліджень та розробок Інституту робототехніки Шанхайського університету Цзяотун, що ведуться з 1979 року. Компанія сформувала потужну дослідницьку команду, яка складається з понад 10 авторитетних експертів у цій сфері та понад 100 провідних інженерів. На сьогодні ЖАКА інтегрує управління та досягла інноваційних проривів у багатьох сферах, таких як інтеграція, об'єднані з'єднання, програмування методом перетягування та бездротове з'єднання.

2.3 НАВІГАЦІЙНІ СИСТЕМИ РОБОТОТЕХНІЧНИХ СИСТЕМ

Експлуатація мобільних роботів потребує використання систем навігації. Існують різні за вартістю, точністю та експлуатаційними витратами системи навігації такі як інерційні, супутникові тощо.

Починаючи з 1960-х років, сучасна навігація використовує гібридні (інтегровані) навігаційні системи, які поєднують різні електронні датчики для збору інформації, необхідної для визначення "безперервного" місцеположення транспортного засобу і зменшення похибок інерціальних сенсорів. Були розроблені інтегровані системи, що об'єднують кілька незалежних навігаційних датчиків, таких як інерціальний вимірювальний блок, доплерівський радар та інші пристрої. Розглянемо та порівняємо продуктивність і ефективність ІНС, GPS та інтегрованих навігаційних систем GPS/ІНС.

2.3.1 Інерціальна навігаційна система

Протягом останніх чотирьох десятиліть спостерігається експоненційне зростання використання інерціальних технологій для задоволення військових та

цивільних потреб в авіаційній і наземній навігації. Останніми роками значно зріс інтерес до аналітичних інерціальних навігаційних систем (ІНС), що зумовлено появою швидких і відносно недорогих мікропроцесорів.

Принципи інерціальної навігації базуються на законах механіки, сформульованих ще Ньютоном, які описують рух тіл відносно інерціальної системи відліку (для рухів у межах Сонячної системи — відносно зірок).

Інерціальна навігація – це методи навігації, що базуються на використанні явища інерції. Вона є автономною та не потребує зовнішніх орієнтирів або сигналів. Неавтономні методи навігації, навпаки, спираються на зовнішні орієнтири чи сигнали та забезпечують вищу точність, оскільки не піддаються накопиченню похибок через зовнішні впливи. Проте в окремих випадках реалізація неавтономної навігації неможлива через відсутність видимості або наявності перешкод для радіосигналів.

Відповідно інерціальна навігаційна система (ІНС) – базується на вимірюванні прискорень та кутових швидкостей за допомогою сенсорів. До переваг ІНС відносяться висока точність на коротких інтервалах часу завдяки швидкому оновленню даних із сенсорів. ІНС можуть ефективно експлуатуватись при наявності радіоперешкод чи спотворення даних.

До недоліків ІНС відноситься накопичення похибок (дрейф), оскільки через найменші неточності у вимірюваннях акселерометрів і гіроскопів помилки інтегрування накопичуються, що призводить до відхилення від реального положення. Також ІНС потребує періодичного калібрування та корекції з допомогою зовнішніх систем, таких як GPS або радіомаяки.

Історичний приклад: Перша у світі балістична ракета Фау-2 мала саме інерційну систему навігації та аналоговий комп'ютер і мала точність, достатню для влучання в місто Лондон. Вона не могла вразити окремих будинок чи район, але її похибка складала близько ± 20 км. Згодом німцям вдалося зменшити її до ± 10 км, хоча дальність запуску становила лише 250 км.

Сучасні системи використовують надзвичайно точне обладнання — лазерні гіроскопи та оптичні акселерометри. Однак навіть із точністю 0,01% на великих відстанях похибка залишається значною.

Основні компоненти ІНС:

- Інерціальні навігаційні системи складаються з трьох основних елементів:
- Гіроскопи – вимірюють кутові швидкості.
- Акселерометри – визначають лінійне прискорення та виконують функцію корекції гіроскопів.
- Високопродуктивний обчислювальний блок із математичними алгоритмами – виконує розрахунки навігаційних задач, обробку та фільтрацію даних.

Інерціальні навігаційні системи поділяються на платформні (ПІНС) з гіростабілізованою платформою та безплатформні (БІНС).

У платформних ІНС тип системи визначається взаємозв'язком між акселерометрами та гіроскопами, які забезпечують орієнтацію акселерометрів у просторі. Виділяють три основні типи платформних інерціальних систем:

Геометричні системи. Вони мають дві платформи: одна стабілізується у просторі за допомогою гіроскопів, а друга, з акселерометрами, орієнтована відносно площини горизонту. Координати об'єкта обчислюються на основі даних про взаємне розташування платформ. Такі системи забезпечують високу точність позиціонування відносно поверхні планети (наприклад, Землі), але малоефективні на високоманеврових апаратах і в космічному просторі. Основне застосування: далекомагістральні літаки (цивільні та військово-транспортні), підводні човни, великі надводні кораблі.

Аналітичні системи. У цих системах акселерометри та гіроскопи залишаються нерухомими у просторі (відносно зірок або галактик). Координати об'єкта обчислюються на основі сигналів з акселерометрів і пристроїв, що визначають обертання об'єкта відносно датчиків. Такі системи мають порівняно невисоку точність поблизу поверхні Землі, але добре працюють на маневрових об'єктах (винищувачі, вертольоти, ракети, швидкохідні судна) та у космосі.

Напіваналітичні системи. Платформа стабілізується відносно місцевого горизонту, на ній розміщені гіроскопи та акселерометри. Координати літака чи іншого апарата визначаються за допомогою зовнішнього обчислювача.

Безплатформні ІНС. У безплатформних ІНС акселерометри та гіроскопи жорстко закріплені на корпусі пристрою. Найпередовішою технологією у виробництві БІНС є волоконно-оптичні гіроскопи, які працюють на основі ефекту Саньяка.

Особливості безплатформних ІНС на базі волоконно-оптичних гіроскопів:

- Відсутність рухомих частин.
- Абсолютна безшумність.
- Висока механічна міцність.
- Відсутність потреби у спеціальному обслуговуванні.
- Тривалий термін безвідмовної роботи (до 80 тис. годин у деяких моделях).
- Низьке енергоспоживання (десятки ват).

Компактні та недорогі інерціальні навігаційні системи знаходять застосування в багатьох сферах, таких як персональна навігація, автомобільна навігація, безпілотні літальні апарати (БПЛА) та загальна авіація (рис. 2.11). Прогрес у галузі волоконно-оптичних гіроскопів і мікроелектромеханічних систем (MEMS) відкриває перспективи подальшого розвитку таких систем.



Рисунок 2.11. Інерціальний блок **IMU-NAV-100** компанії Inertial Labs (США). Вихідні сигнали системи: прискорення, кутові швидкості, вертикальний кут, кут крену, відносний курс, температура. Частота оновлення даних — до 2000 Гц. Час запуску системи <1 секунди. Вага – 100 грам.

У порівнянні з системами вищого класу, недорогі ІНС можуть мати значні похибки у визначенні місцеположення та орієнтації навіть за короткі проміжки часу. Це зумовлено вищими похибками у вихідних даних датчиків. Через це ІНС, побудовані на основі таких датчиків, стають чутливими до нелінійних помилок, особливо коли похибки орієнтації набувають значних значень.

Покращення точності недорогих ІНС дозволить не лише зменшити витрати на існуючі застосування, але й сприятиме створенню нових. Наприклад, така ІНС може використовуватися як автономна система визначення місцеположення у випадках, коли GPS недоступний протягом більше ніж 10 хвилин. Такі пристрої можуть бути актуальними зокрема для БПЛА обприскувачів, специфіка використання яких передбачає польоти на малих висотах в тому числі під лініями електромереж високої напруги. В умовах інтенсивних електромагнітних перешкод, обмовлених високими напругами, були непоодинокі випадки втрати ручного керування чи позиціонування по GPS що призводило аварій БПЛА.

Сучасні інерціальні навігаційні системи часто інтегрують з іншими системами, такими як GPS, для компенсації накопичення похибок і підвищення точності. Інтеграція GPS із ІНС покращує як якість, так і надійність обох систем: GPS використовується для калібрування похибок інерціальних датчиків, тоді як ІНС сприяє підвищенню ефективності відстеження та повторного отримання сигналів GPS-приймачем.

Існує два підходи до калібрування похибок, які можна реалізувати в інтегрованій системі GPS/ІНС:

- Метод прямого зв'язку (або відкритого контуру).
- Метод зворотного зв'язку (або замкнутого контуру).

Ці методи дозволяють оптимізувати роботу інтегрованих систем і забезпечують високу точність навігації, як показано на рис. 2.12

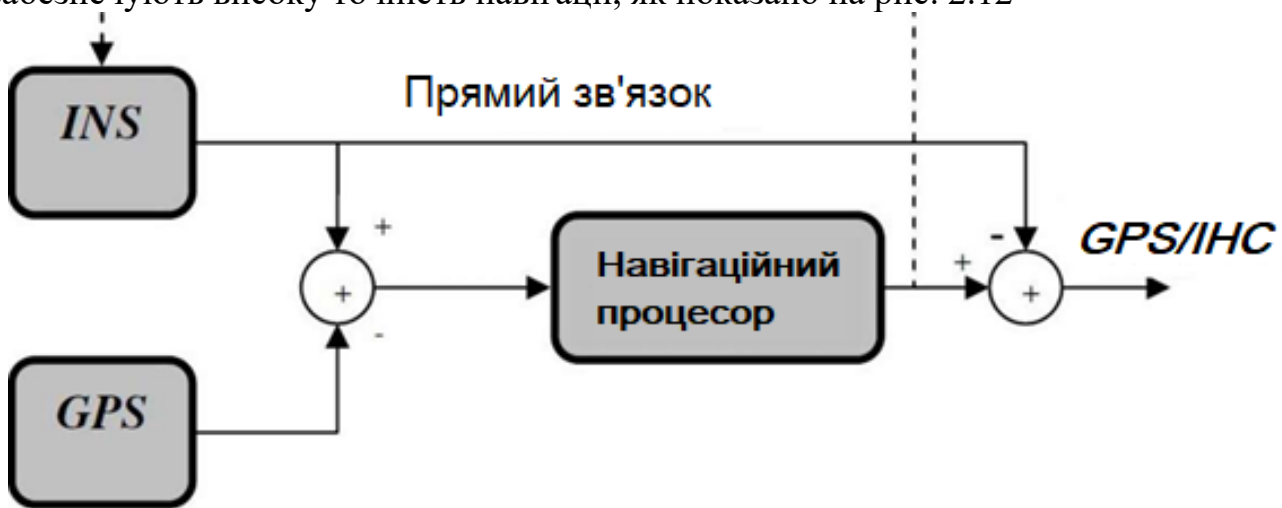


Рисунок 2.12 Концепція інерціального допоміжного засобу

При інтеграції даних GPS та ІНС у систему навігаційний процесор всередині GPS-приймача обчислює місцезнаходження та швидкість, використовуючи лише дані, отримані від GPS. Зовнішній навігаційний фільтр обчислює місцезнаходження, швидкість і орієнтацію на основі необроблених вимірювань інерціального датчика, використовуючи GPS-дані про місцезнаходження та швидкість для калібрування похибок ІНС.

2.3.2. Системи супутникової навігації.

Глобальні навігаційні супутникові системи (GNSS).

Термін «GPS» настільки закріпився у свідомості громадськості, що часто використовується для позначення будь-якої супутникової навігаційної системи. Проте це не зовсім правильно. GPS (Global Positioning System) — це назва системи, розробленої військовими США, яка сьогодні також має цивільне застосування.

Термін GNSS (глобальна навігаційна супутникова система) часто використовується для позначення будь-якої глобальної мережі супутників, які передають навігаційні сигнали на Землю. Це узагальнена назва, яка охоплює такі системи, як GPS, ГЛОНАСС, Galileo тощо. Розрізнення цих термінів важливе, особливо для точних вимірювань, оскільки кожна система має свої особливості. Наприклад, точність GPS є відносно рівномірною по всій поверхні Землі, тоді як ГЛОНАСС забезпечує краще покриття над Європою.

Ідея супутникової навігації виникла ще в 1950-х роках. Під час запуску першого штучного супутника Землі в СРСР американські вчені під керівництвом Річарда Кершнера виявили, що частота сигналу супутника змінюється через ефект Доплера: вона зростає при наближенні і знижується при віддаленні. Це

спостереження дало змогу зрозуміти, що знаючи координати на Землі, можна обчислити положення супутника, і навпаки — знаючи місце розташування супутника, можна визначити власні координати.

У 1973 році була ініційована програма DNSS, яку пізніше перейменували на Navstar-GPS, а згодом — просто GPS. Перший тестовий супутник цієї системи був запущений 14 липня 1974 року. У відповідь на запуск радянських супутників системи позиціонування «Циклон» (цивільний варіант системи відомий як «Цикада») у 1982 році Конгрес США виділив додаткові кошти на прискорення розробки GPS через загострення холодної війни. У 1983 році активізувалися роботи над системою, і до 1993 року на орбіту було виведено всі 24 супутники, необхідні для глобального покриття. Система стала доступною для військового використання, забезпечуючи точне наведення ракет на стаціонарні та рухомі цілі. Спочатку GPS створювалася як виключно військовий проєкт. Однак після трагічного інциденту в 1983 році, коли літак «Боїнг-747» Кореїських авіаліній був збитий радянським винищувачем через помилкову дезорієнтацію екіпажу, систему дозволили використовувати в цивільних цілях із певними обмеженнями.

Більшість GNSS-систем складаються з двох основних компонентів: антени та приймача (блоку обробки). Антена відповідає за прийом сигналів від супутників, тоді як приймач аналізує отриману інформацію та перетворює її у зрозумілі дані, наприклад, широту, довготу та висоту над рівнем моря. Варто зазначити, що вимірювання, які виконує система, стосуються розташування саме антени. Це важливий нюанс, оскільки довжина кабелів антен може призводити до того, що приймач розташовується на певній відстані від реального місця вимірювання. У звичайних GPS-пристроях або продуктах для супутникової навігації це зазвичай не є проблемою, оскільки їхня точність рідко перевищує кілька метрів.

Для кращого розуміння принципу роботи GNSS можна розділити її на основні складові та розглянути їх функції. Розберемо на прикладі GPS, оскільки ця система є найбільш знайомою для широкого загалу:

- Космічний сегмент
- Сегмент управління
- Сегмент користування

Космічний сегмент охоплює супутники на орбіті. У складі GPS-угруповання на середній навколоземній орбіті перебувають 32 супутники, хоча не всі вони активні. Через військове походження системи було закладено резервні супутники, які можуть замінити основні в разі їх виходу з ладу. У системі NAVSTAR функціонує 24 активні супутники. Кожен із них здійснює повний оберт навколо Землі за 11 годин 58 хвилин і 2 секунди на середній висоті 20 200 км (радіус орбіти — 26 571 км). Супутники розташовані у шести площинах, кожна з яких нахилена до екватора під кутом 55°, і кожна площина містить по чотири супутники.

Ця висота вибрана з метою забезпечення стабільності орбіт супутників та мінімізації впливу атмосферного опору. Орбітальна структура основних систем GNSS показана на рис. 2.13

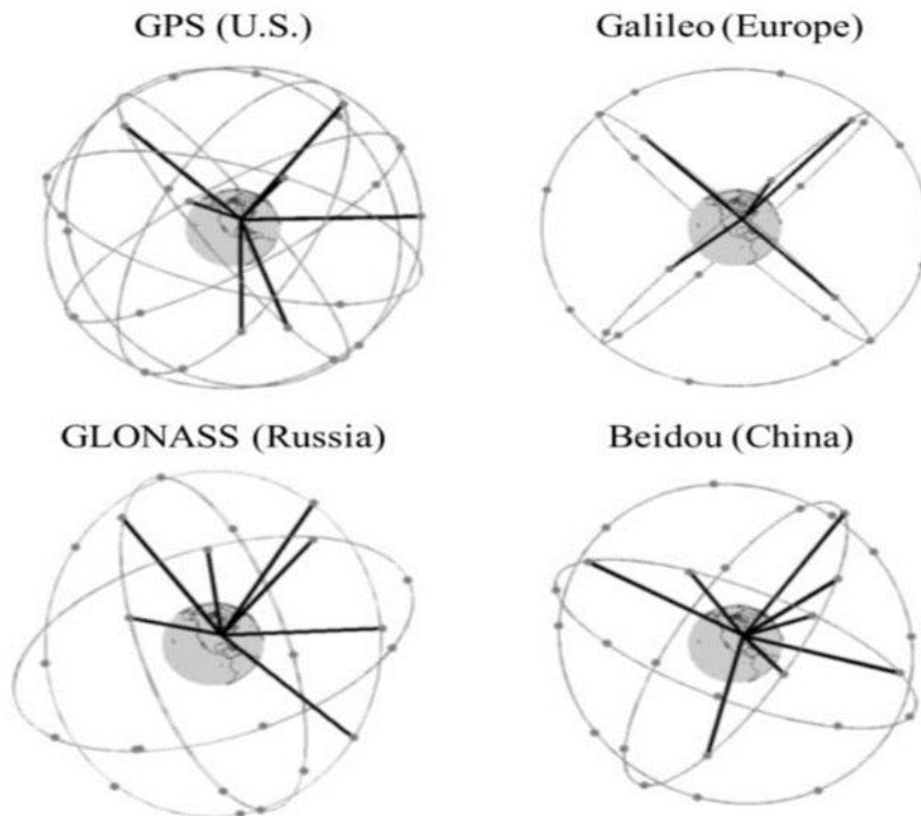


Рисунок 2.13. Орбітальні побудови основних GNSS

Сегмент управління складається зі станцій, розташованих по всьому світу, переважно поблизу екватора. Вони виконують функції моніторингу, управління та передачі даних до кожного супутника GPS. Орбітальні дані, які передаються супутникам, є критично важливими, оскільки дозволяють визначати їхнє точне місцезнаходження під час передачі сигналів. Ця інформація надходить до супутників і далі передається на GPS-приймач у формі навігаційного повідомлення.

Сегмент користувачів. Цей сегмент охоплює всі пристрої, оснащені GPS-приймачем: супутникові навігаційні системи, мобільні телефони, БПЛА, системи безпеки тощо. Сузір'я супутників, розташованих над Землею, постійно передає інформацію користувачам із швидкістю світла. Основою визначення місцезнаходження є метод трилатерації.

Важливо зазначити, що приймачі GNSS, які встановлені в наших телефонах чи навігаційних пристроях, є повністю пасивними — вони лише приймають сигнали, але не передають їх на супутники. Єдиним винятком є європейська система Galileo, приймачі якої мають аварійну функцію передачі даних у надзвичайних ситуаціях, але це не стосується стандартного режиму роботи.

2.3.2.1 Спрощений алгоритм роботи GPS-приймача:

Приймач визначає, скільки часу зайняв сигнал для досягнення його від кожного видимого супутника (враховуючи різницю в часі між хронометрами

супутника і приймача). Цей час множиться на швидкість світла, щоб розрахувати відстань до кожного супутника. На основі трилатерації приймач обчислює своє місцезнаходження щодо щонайменше трьох супутників. Знаючи точне положення супутників під час передачі сигналу, він може визначити своє розташування в системі координат Землі (рис. 2.14).

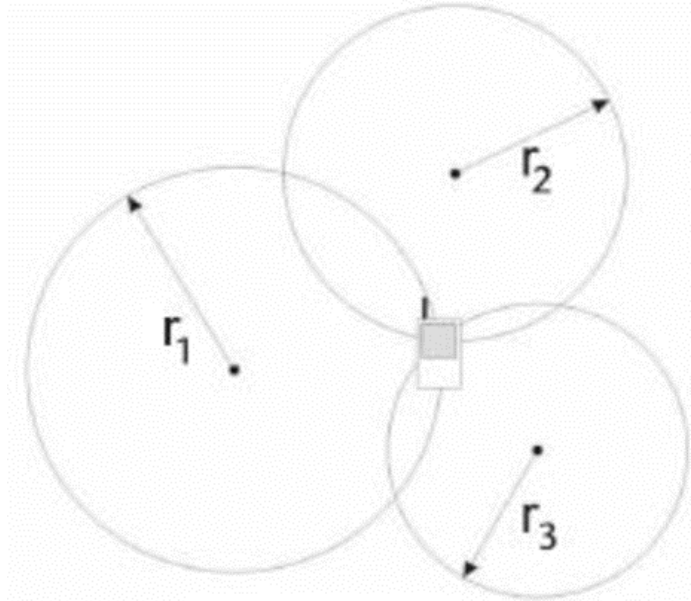


Рисунок 2.14. Схема розрахунку позиціонування GNSS

Особливістю GNSS є використання трьох координат: широти, довготи та висоти. Якщо на площині перетин трьох кіл визначає одну точку (або трикутник похибок у реальних умовах), то в об'ємному просторі при взаємодії зі сферами система видає два можливі рішення: одне розташоване в космосі, а інше — на поверхні Землі. Користувач, відповідно, працює з другим варіантом.

Найскладнішим у розумінні роботи GPS є сам сигнал, який передається супутниками. Що саме він містить? Кожен супутник передає унікальний сигнал, до складу якого входять:

- Код C/A (грубий код);
- P-код (точний код, що у зашифрованому вигляді називається Y-кодом);
- Навігаційне повідомлення.

P-код призначений для військових і зашифрований для запобігання спуфінгу та контролю доступу до системи, тому в цивільному використанні його можна не враховувати. Код C/A, доступний для цивільних користувачів, передається на частоті $L1 = 1575,42$ МГц. Він складається з 1023 біт, кожен повний цикл передачі коду займає лише одну мілісекунду й повторюється нескінченно. Окрім коду C/A, на тій же частоті передається навігаційне повідомлення. Це повідомлення містить значний обсяг важливої інформації, однак через низьку швидкість передачі повне повідомлення надходить протягом 12,5 хвилин.

Варіанти системи супутникового позиціонування.

Стандартна служба позиціонування (SPS - Standard Positioning Service) є базовим рівнем визначення місця розташування, який забезпечують системи GPS, використовуючи код C/A. SPS характеризується найнижчою точністю серед методів позиціонування GPS, яка зазвичай становить від 3 до 10 метрів. Для визначення координат приймач GPS з'єднується з чотирма або більше супутниками та розраховує відстань до них, використовуючи так звані виміри псевдодальності.

Точність SPS залежить від здатності приймача точно співставити свій внутрішньо згенерований код C/A із сигналом, отриманим на антені. Однак, навіть за ідеального поєднання, точність все одно обмежується архітектурою системи. Деякі приймачі здатні досягати рівня юстування до 0,6%, що дозволяє отримувати точність близько 1,8 метра. Хоча це прийнятно для супутникової навігації, багато інших застосувань вимагають більш високої точності.

2.3.2.2. DGPS – Диференціальна GPS

Для підвищення точності розроблені альтернативні методи, такі як DGPS (диференціальні поправки) та RTK (кінематичне позиціонування в реальному часі).

При наявності диференціальних поправок точність SPS може бути значно покращена за рахунок корекції часових затримок супутникових сигналів. Це стає можливим завдяки точному вимірюванню впливу атмосфери на швидкість сигналу замість простих оцінок (рис. 2.15).

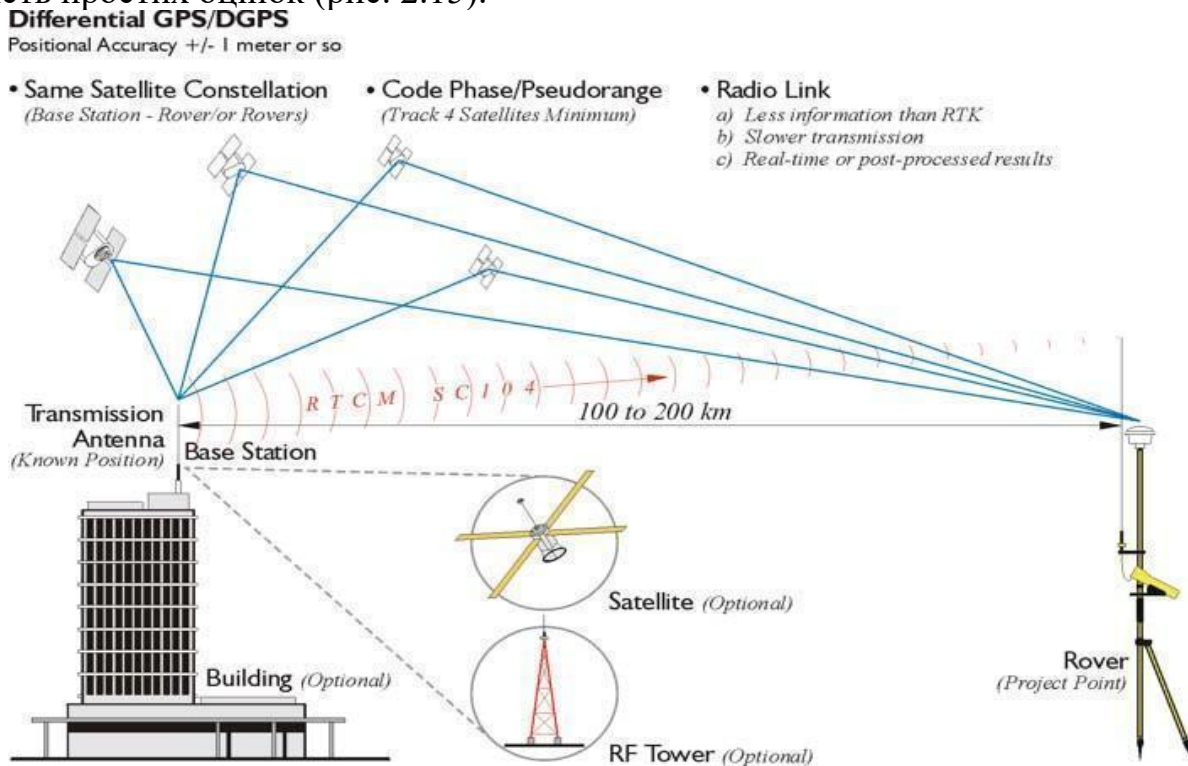


Рисунок 2.15. Схема DGPS

DGPS (диференціальні поправки GPS) може бути реалізовано кількома способами, проте їхній принцип роботи загалом однаковий. Основна причина похибок у вимірюваннях SPS полягає в тому, що приймач не може точно визначити, як іоносфера вплинула на швидкість супутникового сигналу під час його проходження.

Хоча навігаційне повідомлення містить певні оцінки для припущень щодо можливих впливів, вони є лише приблизними. Це означає, що швидкість сигналу, використана для обчислення псевдодальності, може бути некоректною, а отже, і розраховане місцезнаходження має похибки.

Для виправлення цих похибок використовується базова станція GPS — приймач, розміщений у точці з наперед відомими координатами. Базова станція, отримуючи дані від супутників, порівнює своє розраховане положення із точними відомими координатами. Будь-яка різниця між ними свідчить про затримку сигналу одного чи кількох супутників. Система визначає необхідні поправки для кожного супутника, які потім передаються для корекції даних GPS.

Реалізація системи: Поправки обчислюються у режимі реального часу та передаються з базової станції на ровер або кілька роверів для кожного супутника в зоні видимості. Ровери, оснащені відповідними модулями вводу/виводу (I/O), отримують ці дані та коригують свої координати. Передача коригуючих сигналів може здійснюватися через різні канали, наприклад, радіозв'язок, геостаціонарні чи низькоорбітальні супутники, стільникові мережі тощо.

При використанні зовнішніх базових станцій приймач передає своє поточне місцезнаходження на сервер, який визначає найближчу опорну станцію та повертає відповідні поправки.

Інший підхід до отримання поправок реалізовано через супутникові системи доповнень (SBAS). У цьому випадку наземні станції у відомих точках обчислюють диференціальні поправки для певних регіонів. Ці дані завантажуються на супутники, які потім передають їх приймачам, що підтримують відповідне обладнання. Проте доступ до таких послуг часто є платним, оскільки передача даних зазвичай здійснюється у зашифрованому вигляді.

Описаний метод є позиціонуванням на основі коду. У цьому випадку приймач корелює псевдовипадкові коди, передані чотирма або більше супутниками, для визначення відстані до них і розрахунку свого положення. Знаючи ці значення та розташування супутників, приймач здатний визначити своє місцезнаходження з точністю до кількох метрів. Використовуючи технологію DGPS, можна досягти точності позиціонування близько 40 см. RTK є наступним етапом розвитку в порівнянні з DGPS.

2.3.2.3 RTK – кінематика в реальному часі.

Цей метод розробили виробники GPS для підвищення точності порівняно з використанням лише коду C/A. Він базується на аналізі самих сигналів, а не на коді чи навігаційному повідомленні. Ідею RTK вперше запропонував доктор Бенджамін

Ремонді у середині 1980-х років, заклавши основи кінематики GPS/GNSS (Global Positioning System/Global Navigation Satellite System).

Суть роботи RTK полягає у визначенні кількості хвиль між антеною приймача і супутником. Кожен супутник передає унікальний код C/A із 1023 біт, який транслюється зі швидкістю 1023 Мбіт/с. Це означає, що кожен біт надсилається кожному мікросекунду. Враховуючи швидкість світла, сигнал долає за цей час близько 300 метрів. Однак хвиля, що модулює код, має частоту 1575,42 МГц, що відповідає довжині хвилі близько 19 см. Якщо обчислити кількість повних хвиль між супутником і антеною, а також частину хвилі (фазовий кут), можна значно підвищити точність визначення відстані.

Алгоритми RTK Integer та RTK Float працюють одночасно, виконуючи різні задачі. RTK Float використовується для статистичного оцінювання ймовірного розташування, покращуючи поточну точність DGPS. Для цього потрібно щонайменше чотири спільні супутники з базовою станцією. Алгоритм визначає найбільш ймовірну позицію у межах кола, окресленого поточним вимірюванням. Точність RTK Float зазвичай становить 40 см, але в оптимальних умовах може досягати 20 см. RTK Integer, натомість, орієнтований на вирішення проблеми неоднозначності, тобто визначення точної кількості повних хвиль. Для цього необхідно п'ять спільних супутників. Після знаходження правильного рішення система може виміряти відстань із точністю до 0,6% довжини хвилі (19 см), що дозволяє досягти позиціонування з точністю до 1 см.

Для роботи RTK-системи потрібна базова станція, а також щонайменше один стаціонарний еталонний приймач і один рухомий приймач – ровер (рис. 2.16).

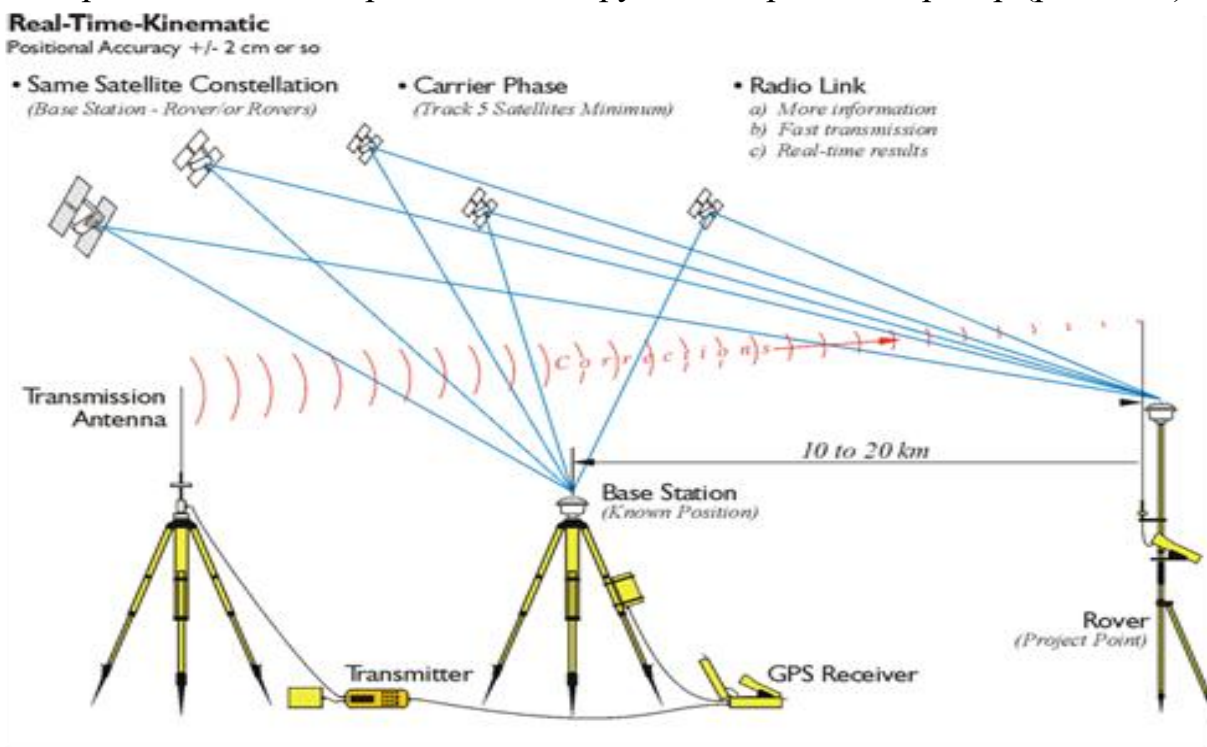


Рисунок 2.16 Принцип дії RTK

Усі приймачі в системі одночасно спостерігають за одними й тими ж супутниками. Базові приймачі залишаються стаціонарними на контрольних точках, тоді як ровери переміщуються між проектними точками, роблячи короткі зупинки в кожній новій точці. Дані, зібрані під час таких вимірювань, дозволяють обчислювати вектори між роверами та базовими приймачами в реальному часі.

Оптимальна робота RTK досягається, коли відстань між базовою станцією та ровером не перевищує 20 км. Однак ефективна довжина базової лінії може бути додатково обмежена дальністю радіозв'язку, яка залежить від умов передачі сигналу. На якість сигналу можуть впливати високовольтні лінії електропередачі та інші перешкоди.

Робота з одиночною базовою станцією

Наприклад, одиночна базова GNSS-станція зручно використовується для виконання аерофотозйомки за допомогою дрона, оснащеного RTK-приймачем. У цьому випадку кожен знімок, отриманий під час польоту, буде мати високу точність геоприв'язки, що дозволяє фіксувати координати центру кадру з максимальною точністю.

Робота з RTK-мережею

RTK-мережа – це сучасна наземна GNSS-інфраструктура, яка включає комплекс апаратно-програмних засобів, що складається з базових GNSS-станцій та обчислювального центру.

Базова станція (або RTK-станція) – це комплекс геодезичного обладнання, що включає GNSS-приймач і GNSS-антену. Антени розміщують на стаціонарних пунктах, де вони проходять високоточну геодезичну прив'язку. Координати фазового центру антен визначають з точністю до 1–2 мм. Приймачі зазвичай є мультичастотними і багатосистемними (приймають сигнали GPS, GALILEO, ГЛОНАСС, BEIDOU). Для створення RTK-мережі потрібно щонайменше п'ять базових GNSS-станцій, розташованих на відстані до 70–80 км одна від одної. Максимальна кількість станцій не обмежується. Базові станції зазвичай є стаціонарними та формують єдину RTK-мережу.

Обчислювальний центр обробляє дані від усіх базових станцій, виконує розрахунки для створення RTK-поправок і забезпечує їх доступ для користувачів мережі. Його функціонал реалізований через спеціальне програмне забезпечення для управління RTK-мережею. Найпоширенішими прикладами такого ПЗ є Trimble Pivot, Leica Spider, TopNET+, BNC тощо.

Постачальники послуг RTK

В Україні послуги RTK, а також обладнання та його обслуговування, надають кілька приватних компаній. На державному рівні цими питаннями опікується відділ застосування та експлуатації навігаційних систем, який входить до складу

Національного центру управління та випробувань космічних засобів (<https://net.spacecenter.gov.ua/>) (рис. 2.17).

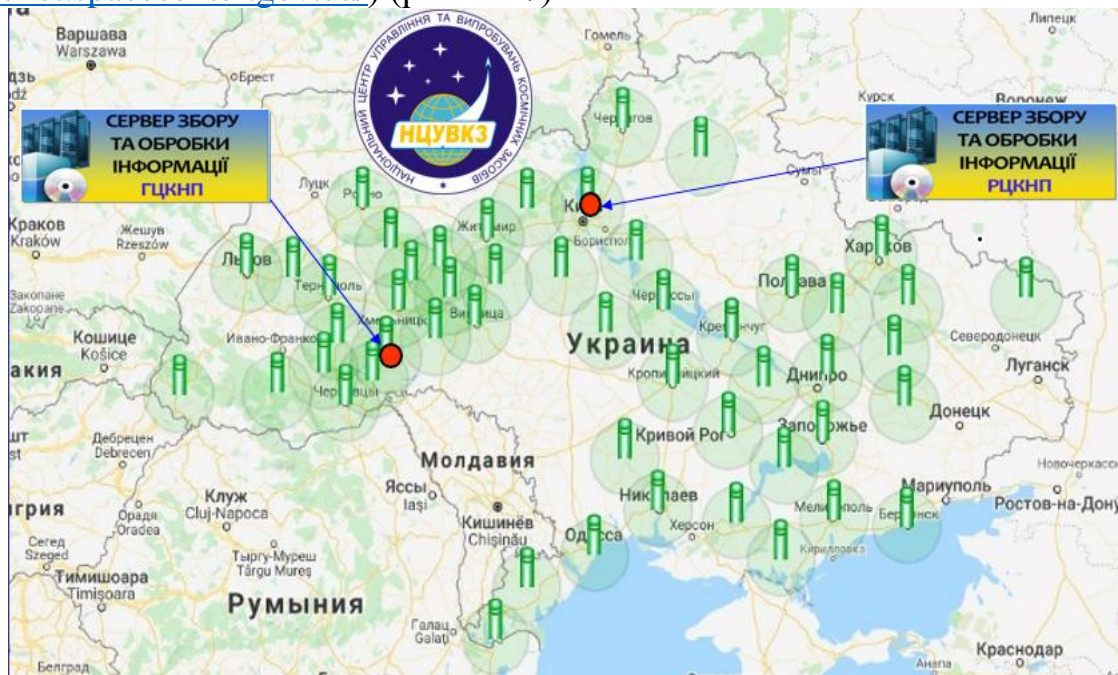


Рисунок 2.17. Покриття території України RTK мережею РЦКНП станом на 2022 рік

Для зменшення впливу багатопробності використовуються антени типу «choke ring». Дросельна кільцева антена (англ. choke ring) – це спеціальна конструкція всепрямованої антени, яка застосовується на високих частотах. Вона складається з кількох концентричних провідних циліндрів, розташованих навколо центральної антени. Через складність конструкції антени її захищають спеціальним кожухом або обтічником, особливо при використанні в умовах відкритого середовища.

У наземних GPS-приймачах така антена забезпечує міліметрову точність вимірювань, що робить її ідеальною для геодезичних і геологічних завдань. Кожна базова станція RTK-мережі приймає сигнали від усіх доступних навігаційних супутників усіх GNSS-систем і передає ці дані через канали зв'язку до серверів для обробки. На основі цих даних формуються диференційно-коригувальні поправки RTK (рис. 2.18).

Споживачі та організація передачі даних у RTK-мережі

Користувач, оснащений навігаційним RTK-приймачем, отримує сигнали від усіх видимих супутників GNSS і RTK-поправки. Врахування цих поправок дозволяє здійснювати високоточні вимірювання.

Для забезпечення передачі даних у RTK-мережі була створена технологія NTRIP (Networked Transport of RTCM via Internet Protocol), яка забезпечує передачу даних диференційної корекції (наприклад, у форматі RTCM-104) та інших GNSS-потоків через Інтернет. Ця технологія дозволяє користувачам отримувати поправки

через мобільні IP-мережі, такі як GSM, GPRS, EDGE або UMTS. Мережа NTRIP складається з трьох компонентів: джерел даних, трансляторів і клієнтів.

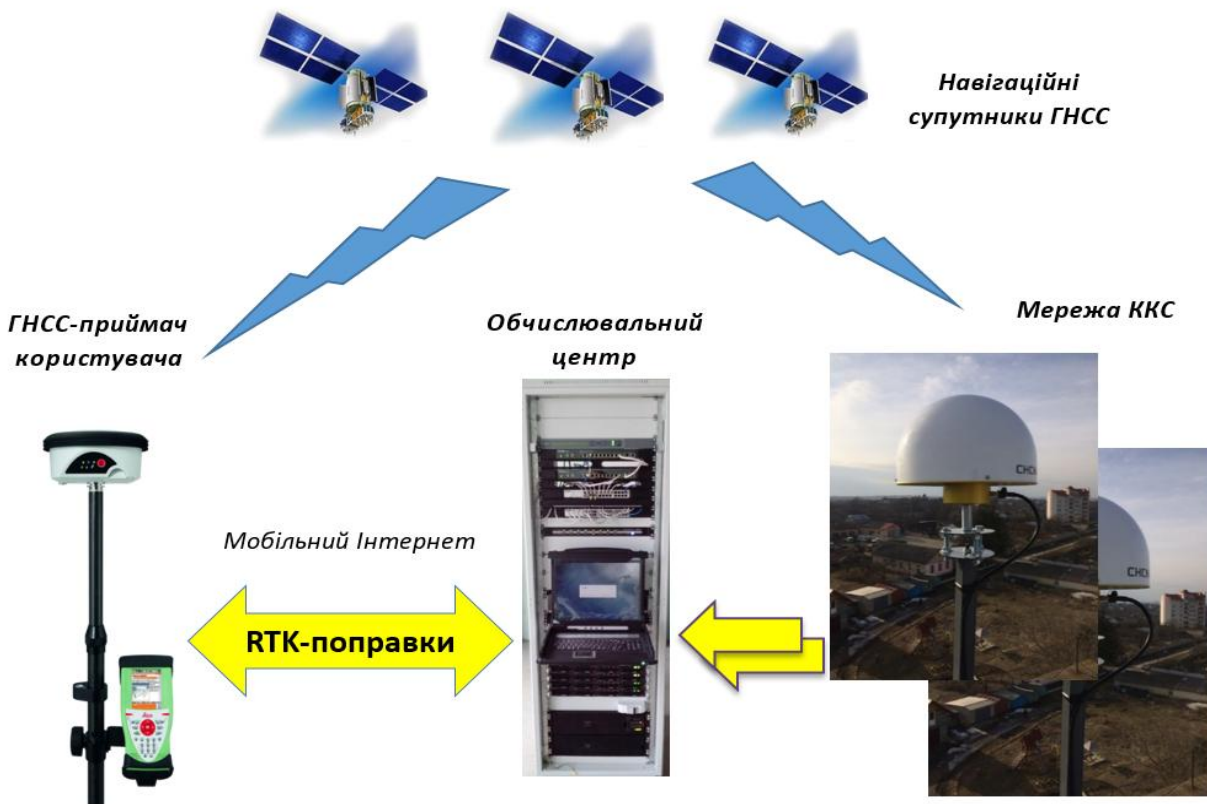


Рисунок 2.18. Загальний алгоритм функціонування RTK-мережі

Основне завдання трансляторів NTRIP – розподіл даних, отриманих від базових GNSS-станцій (джерел), між численними клієнтами. Передача інформації між джерелами, трансляторами та клієнтами здійснюється за допомогою HTTP-протоколу. Технологія NTRIP підтримує передачу будь-яких потоків GNSS-даних (GPS, GLONASS, Galileo, EGNOS, WAAS тощо) зі швидкістю від 0,2 до 10 Кбіт/с, зокрема:

- RTK-поправки, DGPS, трансляції ефемерид або корекції орбіти/годинника;
- необроблені дані приймачів;
- інші формати даних GNSS.

Сьогодні транслятори NTRIP використовуються в усьому світі для розповсюдження GNSS-даних у реальному часі. Перелік відомих установок доступний на сайті www.rtcn-ntrip.org.

Переваги технології NTRIP

- Використання HTTP-протоколу: Передача даних здійснюється через стандартний порт 80, що дозволяє уникнути проблем із брандмауерами та проксі-серверами.

- Підтримка масового доступу: Кастер, розташований між сервером і клієнтами, дублює вхідні й вихідні дані, забезпечуючи одночасний доступ для великої кількості користувачів.
- Гнучкість: Технологія дозволяє передавати будь-які GNSS-дані, незалежно від їх формату чи кодування.
- Безпека: Постачальники та користувачі не контактують безпосередньо, а потоки даних захищені від блокування мережевими елементами.
- Мобільність: Потоки передаються через будь-які мобільні IP-мережі завдяки використанню TCP/IP.

Основні характеристики технології NTRIP

- Простота реалізації завдяки популярному стандарту HTTP.
- Можливість одночасного підключення сотень користувачів до тисяч потоків.
- Незалежність від типу даних, що передаються.
- Висока ефективність роботи через мобільні мережі.

Отже NTRIP – це ефективний протокол для передачі диференційних поправок GPS (DGPS) через Інтернет для забезпечення RTK-режиму. Він базується на TCP-протоколі, що дозволяє забезпечити надійну багатокористувацьку передачу даних у режимі реального часу.

Слід враховувати, що для повноцінної роботи референційної (базової) станції в режимі реального часу необхідно, щоб приймач включав дві компоненти NTRIP: NtripServer і NtripCaster. Рухомий користувач повинен мати можливість використовувати третю компоненту – NtripClient.

Використання GNSS у сільськогосподарській техніці

Інсталяція GNSS-систем можлива на більшість тракторів і комбайнів, які вже використовуються в господарствах. Одним із прикладів такого обладнання є системи паралельного водіння. У цьому випадку роль оператора зводиться до того, щоб привести машину до стартової точки, а далі – здійснювати лише контроль за виконанням операцій у разі виникнення надзвичайних ситуацій.

Ця інновація суттєво знижує вимоги до кваліфікації операторів, зменшує їхню втомлюваність, підвищує точність виконання робіт і поліпшує умови технічного обслуговування техніки.

Сучасні технології також дозволяють використовувати повністю автономні трактори. Завдяки високоточному GNSS-позиціонуванню вони стають важливим елементом програм точного землеробства. Такі трактори допомагають зменшити витрати на робочу силу, паливе та технічне обслуговування, а також підвищують ефективність виконання завдань.

Окрім виконання основних аграрних операцій, автономні трактори функціонують як мобільні точки збору даних із датчиків, розташованих на полі. Вони розробляються для підтримки широкого спектра завдань – від точного висіву до обприскування та збирання врожаю (рис. 2.19).



Рисунок 2.19. Трактори пілотований (зліва) і безпілотний (справа) із RTK спорядженням

Попри переваги RTK технологій можливі ситуації втрати сигналу, в цьому випадку, як і з базовим GPS, існують комбіновані рішення - інерційні системи та RTK. На рис. 2.20 показано модуль INS-B-OEM — це версія Original Equipment Manufacturer (OEM) комбінованої системи ІНС + GNSS, призначена для точного та надійного визначення координат, компонентів вектора швидкості та параметрів кутової орієнтації (Heading, Pitch, Roll).



Рисунок 2.20. Модуль **INS-B-OEM** компанії Inertial Labs (США). Вага 115 грам.

Блок INS-B-OEM оснащений OEM-платою GNSS-приймача OEM617 від компанії NovAtel Inc., барометром, тривісними магнітометрами, відкаліброваними в повному діапазоні робочих температур, акселерометрами та гіроскопами. Це забезпечує точне визначення положення, компонентів вектора швидкості, напряму, тангажу та крену вимірюваного пристрою. Блок має вбудовані датчики, оптимальні навігаційні алгоритми та спеціальне програмне забезпечення для калібрування.

2.3.3 Системи автономної навігації

Система управління мобільного робота має здатність автономно (без участі оператора) приймати рішення щодо своєї поведінки в умовах невизначеності та самостійно орієнтуватися на місцевості. Важливу роль у цьому процесі відіграє навігаційна система, яка дозволяє роботу аналізувати навколишнє середовище та планувати маршрут, враховуючи наявність рухомих об'єктів на його шляху.

Для вирішення цього завдання часто використовуються навігаційні системи, що базуються на застосуванні систем технічного зору, лазерних сканувальних далекомірів (лідарів), радірів та ультразвукових сенсорів. На відміну від системи супутникової навігації, яка використовує інформацію від зовнішніх джерел (пасивний режим) ці сенсори можуть працювати автономно, навіть в умовах потужних електромагнітних перешкод.

2.3.3.1 Візуальні датчики (системи технічного зору)

Система технічного зору робота (СТЗ) – це сенсорна система, що забезпечує сприйняття, обробку та аналіз візуальної інформації про навколишнє середовище. Основна її функція – отримання зображень робочої сцени та їх подальша інтерпретація.

Зображення в цьому контексті розглядається як функція, що визначає значення яскравості для кожної точки або ділянки сцени. У рамках системи технічного зору виділяють такі етапи: формування, первинна обробка, сегментація, опис, аналіз та інтерпретація зображень.

Системи технічного зору розробляються для виконання вузькоспеціалізованих завдань, таких як підрахунок об'єктів на конвеєрі, зчитування серійних номерів або виявлення поверхневих дефектів. Їх перевага полягає у високій швидкості роботи, здатності функціонувати 24/7 та високій точності повторюваних вимірювань.

Типова система технічного зору складається з таких елементів:

- одна або кілька цифрових чи аналогових камер (чорно-білих або кольорових) з відповідною оптикою для знімання зображень;
- програмне забезпечення для обробки отриманих зображень (для аналогових камер – оцифровувач зображень);
- процесор (сучасний ПК із багатоядерним процесором або вбудований процесор, наприклад, цифровий сигнальний процесор – ЦСП);
- спеціалізоване програмне забезпечення технічного зору для розробки окремих застосунків;

- пристрої введення-виведення або канали зв'язку для передавання отриманих результатів;
- "розумна" камера – інтегроване рішення, що поєднує всі вище перераховані компоненти в одному пристрої;
- спеціалізовані джерела освітлення (світлодіоди, люмінесцентні та галогенні лампи тощо);
- програмні додатки для обробки зображень і виявлення необхідних характеристик;
- датчик синхронізації (оптичний або магнітний) для точного визначення моменту знімання й обробки зображень;
- приводи певної конструкції, які використовуються для сортування або відбраковування дефектних деталей.

Принцип роботи датчика синхронізації. Цей датчик визначає момент, коли деталь, що рухається по конвеєру, досягає зони перевірки. Він активує камеру для знімання зображення деталі у відповідний момент, часто синхронізуючи цей процес із імпульсом освітлення для забезпечення чіткого та якісного зображення.

Освітлення, яке використовується для підсвічування деталей, призначене для акцентування важливих характеристик об'єкта та зменшення впливу небажаних елементів, таких як тіні чи відблиски. Для цього часто застосовують світлодіодні панелі відповідного розміру та розташування.

Зображення, отримане камерою, передається до пристрою захоплення кадрів або безпосередньо в пам'ять комп'ютера.

Пристрій захоплення кадрів – це оцифровувач (вбудований у "розумну" камеру або реалізований у вигляді окремої плати в комп'ютері), який перетворює аналоговий відеосигнал у цифровий формат і зберігає зображення в пам'яті комп'ютера для подальшої обробки за допомогою програмного забезпечення технічного зору.

Хоча більшість систем технічного зору використовують чорно-білі камери, кольорові моделі набувають все більшого поширення. Крім того, зростає популярність цифрових камер з прямим підключенням, що дозволяє відмовитися від окремих пристроїв захоплення кадрів, спрощуючи систему та знижуючи витрати.

"Розумні" камери, обладнані вбудованими процесорами, займають дедалі більшу частку ринку технічного зору, забезпечуючи високу швидкість обробки зображень і автономність.

На рис. 2.21 зображена камера технічного зору FQ-M, інтегрована в робота SCARA, призначеного для виконання швидкісних маніпуляцій за принципом «взяти-покласти».



Рисунок 2.21 – Камера технічного зору FQ-M у складі робота SCARA

Нова серія датчиків технічного зору **FQ-M** розроблена спеціально для автоматизованих систем захоплення, переміщення та встановлення деталей. Всі моделі оснащені вбудованим портом **EtherCAT**, що дозволяє легко інтегрувати їх у різні виробничі процеси.

FQ-M – це компактний, високошвидкісний пристрій, який підтримує підключення інкрементного енкодера для відстеження та калібрування. Для налаштування датчика ідеально підходить програмне забезпечення **Sysmac Studio** від компанії **Omron**. У поєднанні з консоллю **TouchFinder** воно забезпечує можливість моніторингу системи безпосередньо на об'єкті.

FQ-M здатний одночасно відстежувати до **32 об'єктів** та обробляти до **5000 об'єктів на хвилину**. Новий алгоритм контурного пошуку забезпечує високу точність і надійність роботи датчика.

Методи обробки інформації від сенсорів технічного зору

Програмні пакети для технічного зору, зокрема з відкритим вихідним кодом, зазвичай містять набір методів обробки зображень, серед яких:

- Підрахунок пікселів – визначає кількість світлих або темних пікселів у зображенні.
- Бінаризація – перетворює градаційне зображення у двоколірне (чорно-біле).
- Сегментація – використовується для виявлення та підрахунку об'єктів.
- Зчитування штрих-кодів – розпізнає та декодує 1D і 2D коди, призначені для машинного зчитування.
- Оптичне розпізнавання символів (OCR) – автоматично зчитує текст, наприклад, серійні номери.
- Вимірювання – визначає розміри об'єктів у міліметрах або дюймах.
- Виявлення контурів – знаходить краї об'єктів на зображенні.
- Співставлення шаблонів – здійснює пошук, ідентифікацію та підрахунок об'єктів за заданими зразками.

У більшості випадків системи технічного зору використовують комбінацію цих методів для повноцінного контролю якості. Наприклад, система, яка зчитує штрих-коди, може одночасно перевіряти поверхню на наявність дефектів (подряпин чи пошкоджень) та вимірювати довжину і ширину деталей.

2.3.3.2 Лазерні та інфрачервоні датчики

Лазерні та інфрачервоні датчики відрізняються типом використовуваного випромінювання, проте працюють за спільним принципом — триангуляції.

Метод триангуляції базується на вимірюванні кута між лазерним променем, що відбивається від об'єкта, і об'єктивом далекоміра. Визначивши відстань між лазером і об'єктивом та вимірявши відповідний кут, можна обчислити відстань до об'єкта: що менший кут, то більша відстань.

Розглянемо два принципи характерні виключно для лазерних сенсорів:

Принцип Time-of-Flight (ToF) – вимірювання часу проходження сигналу. У цьому методі лазерний діод випромінює імпульси, які відбиваються від об'єкта та повертаються назад, де їх фіксує фотоприймач датчика. Вимірюючи час між випуском імпульсу та його поверненням, система розраховує відстань до цілі. Цей метод найчастіше використовується в датчиках далекої дії, оскільки при великих відстанях похибка вимірювання таймера стає менш значущою.

Принцип фазового зсуву. На відміну від ToF, цей метод передбачає безперервну роботу лазера, проте його випромінювання амплітудно модулюється сигналом певної частоти (зазвичай до 500 МГц). Довжина хвилі лазера при цьому залишається постійною (у межах 500–1100 нм). Відбитий від об'єкта сигнал сприймається фотоприймачем, після чого його фаза порівнюється з фазою опорного сигналу лазера. Затримка у розповсюдженні хвилі спричиняє фазовий зсув, який і використовується для визначення відстані.

Слід враховувати, що і лазерне, і інфрачервоне випромінювання є світловими хвилями, тому датчики, що працюють на їх основі, мають певні обмеження. Вони погано взаємодіють з матеріалами, які пропускають або поглинають світло. Наприклад, скло може залишатися "невидимим" для таких датчиків, а темні об'єкти будуть виявлятися з меншої відстані (чим темніший колір, тим більше світла він поглинає).

Однак між лазерними та інфрачервоними датчиками існують і суттєві відмінності. Інфрачервоне випромінювання поширене в природі, тому такі датчики можуть давати значні перешкоди, особливо під впливом прямого сонячного світла. Лазерне випромінювання, навпаки, є вузьконаправленим, монохроматичним і когерентним, що мінімізує вплив перешкод, оскільки в природі лазери не зустрічаються, і єдиним джерелом завад може бути інший лазер. Проте така

перевага має свою ціну: лазерні джерела випромінювання значно дорожчі за інфрачервоні. Компактні гаджети розміром з смартфон здатні вимірювати відстані в середньому до 100 метрів.

Подальшим розвитком лазерного сканування стали обертові та скануючі випромінювачі. Такі пристрої можуть не тільки вимірювати відстань до однієї точки, а й створювати повноцінні карти рельєфу або 3D-моделі об'єктів. Саме такі пристрої найчастіше називають лідарами. Більшість лідарів працюють на одній із трьох довжин хвиль — 850, 905 або 1550 нм. Джерела випромінювання з довжинами хвиль 850 і 905 нм за високої потужності можуть пошкоджувати сітківку ока, проте їх виробництво є максимально дешевим і доступним. Випромінювання з довжиною хвилі 1550 нм є безпечним для людського ока навіть при значній потужності, однак для створення таких випромінювачів потрібні досить рідкісні ресурси, наприклад, арсенід галію-індію.

Попри свої значні переваги, лідари мають і певні недоліки: вони менш ефективні у замкнених просторах та при яскравому сонячному світлі, мають обмежений кут огляду в азимутальній площині та відзначаються високим енергоспоживанням. У автомобілях Tesla лідарами не використовують — для навігації компанія покладається виключно на камери, радар і ультразвукові датчики. Ілон Маск відкрито заявляє, що лідарами — це надто дорогий і неефективний спосіб забезпечення навігації, який у майбутньому, з розвитком технологій комп'ютерного зору, стане повністю зайвим. На його думку, камери в поєднанні з потужними алгоритмами обробки даних забезпечують достатній рівень точності й безпеки для автономного водіння.

2.3.3.3 Радари (Radio Detection and Ranging) — це пристрої, які використовують радіохвилі для визначення відстані, напрямку та швидкості об'єктів. Можливо використання різних частотних діапазонів для моніторингу, що обумовлено їх унікальними характеристиками. Розглянемо основні діапазони частот та їх застосування:

- Низькі частоти (30-300 МГц). Такі радарні системи зазвичай застосовуються для дослідження різних типів ґрунтів. Перевагою цих систем є гарна проникаюча здатність хвиль, а недоліками — порівняно великі розміри антен. Ці системи не підходять для використання на мобільних роботах.
- Середні частоти (300 МГц - 3 ГГц). Ці системи часто використовуються в авіації та метеорології для моніторингу руху хмар. Дальність таких радарів може досягати 100 км за умови прямої видимості та відсутності перешкод. Однак точність цих систем досі не висока, а розміри антен не дозволяють встановлювати їх на мобільні роботи.
- Високі частоти (3-30 ГГц). Ці радарні системи знаходять застосування в автомобільній промисловості, системах запобігання зіткнень та навігаційних системах роботів. Перевагою є висока роздільна здатність і компактні розміри

антен. Однак їх недоліком є обмежена дальність роботи, хоча для невеликих автономних роботів це є цілком прийнятним.

- Дуже високі частоти (30-300 ГГц). Такі системи застосовуються для автономних транспортних засобів, роботів і біомедичних застосувань. Вони мають дуже високу роздільну здатність і здатність виявляти дрібні об'єкти. Недоліками є обмежена дальність роботи та висока чутливість до погодних умов.

В таблиці 2.1 приведено порівняльну характеристику наявних на ринку радарних систем

Таблиця 2.1

Характеристика	Texas Instruments IWR6843 (рис.2.22)	Aeva 4D LiDAR	Navtech Radar CIR
Діапазон частот	60-64 ГГц	Н/Д (LiDAR)	76-77 ГГц
Роздільна здатність	до 4 мм	до 1 см	до 10 см
Кут огляду	до 180°	120°/30°	360°
Дальність виявлення	до 200 м	до 300 м	до 500 м



Рисунок 2.22 Радар Texas Instruments IWR6843

Отже радары здатні відігравати значну роль у робототехніці, забезпечуючи навігацію, уникнення перешкод і виконання спеціалізованих завдань. Зокрема їх використання у сільськогосподарських дронах, таких як DJI, значно підвищує ефективність і безпеку виконання аграрних задач.

2.3.3.4 Ультразвукові сенсори

Ультразвуковий дальномір працює за принципом випромінювання звукових хвиль із частотою понад 20 кГц. Коли ці хвилі стикаються з перешкодою, вони відбиваються та повертаються до приймача, який вимірює час між випромінюванням і прийомом сигналу. На основі цього часу та швидкості поширення звуку в повітрі обчислюється відстань до об'єкта.

Однак цей метод має певні обмеження. Для точного вимірювання об'єкт повинен мати плоску поверхню, розташовану перпендикулярно до напрямку ультразвукового сигналу, і бути здатним ефективно відбивати звукові хвилі. Якщо відстань до об'єкта занадто велика, відбитий сигнал може значно ослабнути, як показано на рис. 2.22а. Якщо ж об'єкт розташований занадто близько, відбитий імпульс може накладатися на випромінений сигнал, що ускладнює його реєстрацію (рис. 2.22б). Маленькі об'єкти відбивають недостатньо енергії ультразвукового імпульсу, через що їх важко виявити (рис. 2.22в). Крім того, якщо поверхня розташована під кутом до променя, сигнал може частково відбитися вбік і не досягти приймача (рис. 2.22г).

На точність вимірювань впливають також зовнішні фактори, такі як температура та вологість, які змінюють швидкість поширення звуку. Додатковий вплив мають повітряні потоки, що створюють перехідні області з різною щільністю, які можуть відбивати або розсіювати ультразвукові хвилі, викликаючи похибки у вимірюваннях.

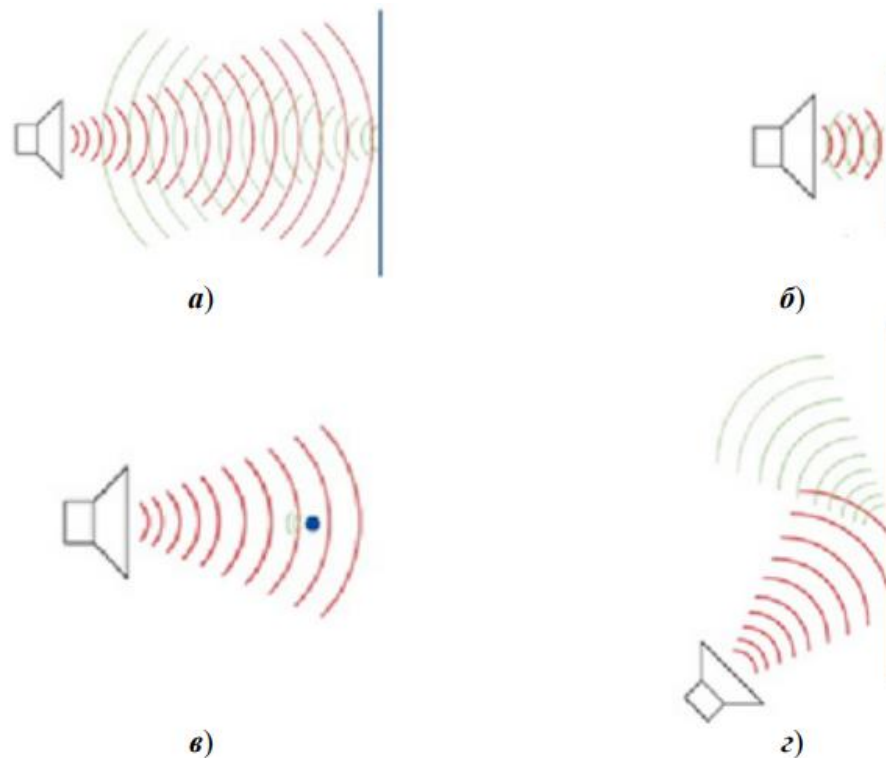


Рисунок 2.23 Вплив середовища на похибку вимірювання відстані

Варто зазначити, що звук розсіюється в просторі значно сильніше, ніж світлові хвилі, що обмежує максимальну дальність вимірювань. У представлених

на ринку рішень цей показник досягає максимуму 10 метрів, але в середньому становить від 0,4 до 5 метрів

2.4 ПЛАТФОРМИ МОБІЛЬНИХ РОБОТОТЕХНІЧНИХ СИСТЕМ

2.4.1 НАЗЕМНІ ПЛАТФОРМИ

Різноманіття способів пересування наземних робототехнічних систем значно зросло за останні роки. Основним елементом будь-якого мобільного наземного робота є шасі, конструкція якого визначає його функціональні можливості. Шасі або самохідне шасі – це сукупність компонентів, які забезпечують передачу механічної енергії від двигунів до активних елементів рушія, таких як ведучі колеса, зірочки, шківів, виконавчі механізми зміни геометрії шасі або механізми крокування. Ці компоненти разом із приводами утворюють функціональну конструктивну підсистему.

У цьому розділі проведено структурний аналіз та систематизацію сучасних конструкцій шасі.

Принципи пересування наземних робототехнічних систем

Шасі та обраний тип рушія значною мірою визначають конструктивний вигляд мобільного робота і його функціональні можливості, зокрема здатність пересуватися в непередбачуваних умовах та адаптуватися до складних перешкод. У конструкціях наземних роботів часто використовуються колісні та гусеничні рушії, які зазвичай мають найпростіші шасі фіксованої конфігурації з різною кількістю активних елементів.

Водночас розвиток робототехнічних систем у напрямі зменшення розмірів і ваги, а також підвищення їх здатності долати складні перешкоди, призводить до ускладнення конструкцій шасі шляхом додавання пасивних або активних механізмів адаптації. Це дозволяє значно покращити прохідність роботів та розширити їхні можливості.

Також активно розробляються нові теоретичні підходи та створюються конструкції мобільних роботів, зокрема малогабаритних, із крокуючими (наприклад, компанія Boston Dynamics у США) та повзучими (наприклад, Hirose Fukushima Robotics у Японії) типами рушіїв. Повзучі схеми представляють собою багатомодульні зчленовані структури, які пересуваються завдяки рухам у шарнірах між модулями.

Класифікація шасі

Як і у випадку з транспортними засобами, шасі наземних робототехнічних систем поділяються на класи за принципом пересування:

- Колісні
- Гусеничні
- Крокуючі
- Гібридні або комбіновані

- Зчленовані

Гібридні шасі об'єднують елементи кількох класів, наприклад, колісно-гусеничні, колісно-крокуючі, гусенично-важільні конструкції тощо. У таких шасі головну роль відіграють механізми адаптації.

Прикладом може бути робот під назвою SandFlea (розроблений компанією Boston Dynamics), який оснащений спеціальним механізмом для подолання великих перешкод. Цей механізм включає важелі для нахилу корпусу та поршневий привід, що здатний виштовхувати робота на висоту до восьми метрів. Основою для цієї конструкції слугує просте чотириколісне шасі.

В іншому випадку адаптаційним механізмом є маніпулятор із пасивними колесами, який дозволяє підвищити прохідність робота. Така конструкція відноситься до гібридних шасі з шестиколісним рушієм і додатковими можливостями завдяки виконавчому обладнанню.

Критерії класифікації

Класифікація побудована на основі покрокового аналізу схем шасі за ступенем їх конструктивного ускладнення з метою підвищення адаптаційних можливостей. Аналіз здійснюється шляхом визначення кількості ступенів рухливості (тобто рухомих елементів) у певній функціональній підсистемі шасі. Усього виділяють три основні функції шасі: пересування, маневрування та подолання перешкод.

На першому етапі визначається спосіб пересування і кількість рухомих елементів рушія (двоколісні, триколісні, чотириколісні, шестиколісні, восьмиколісні, двогусеничні, чотиригусеничні, шестигусеничні тощо). Далі, насамперед для колісного рушія, уточнюється кількість ведучих і ведених елементів, а також обраний спосіб маневрування (бортовий або за допомогою керованих елементів).

Для гусеничних шасі, з огляду на їх конструктивні особливості, зазвичай застосовується бортовий спосіб повороту, який вважається стандартним. Однак у дуже рідкісних випадках використовуються схеми повороту за рахунок вигину гусеничної стрічки (наприклад, у шасі танка Mk VII Tetrarch, згаданого в дослідженні). У галузі мобільної робототехніки можна навести приклад робота Augora компанії Automatika Inc. (США), створеного за одногусеничною схемою, де поворот здійснюється шляхом вигину гусениці.

У колісних роботах у переважній більшості випадків також використовується бортовий (або "танковий") спосіб повороту. Тому він вважається стандартним, а наявність керованих елементів рушія зазначається окремо.

На останньому етапі аналізу оцінюються адаптаційні можливості шасі. Розглядається тип адаптаційного механізму (шарніри зламів рами або корпусу, поворотні модулі – колісні, гусеничні, важільні тощо), а також визначається кількість ступенів рухливості (активних і пасивних).

У підсумку, на основі проведеного аналізу, для досліджуваної схеми шасі створюється формула – компактний запис структурного складу шасі мобільного робота, що складається за певними правилами.

Формульний запис схеми шасі мобільного робота

Для складання формули використовуються літери українського алфавіту (виняток – латинська «S» (від англ. Segway)), арабські цифри та спеціальні символи (знак множення «×» і роздільник «/»).

Позначення елементів формули:

- К – колісний рушійний елемент;
- Г – гусеничний рушійний елемент;
- О – рушійний елемент у вигляді «омні-колеса» (колесо типу Mecanum-wheel у схемі Omni Directional Drive представлено на рис. 2.24);

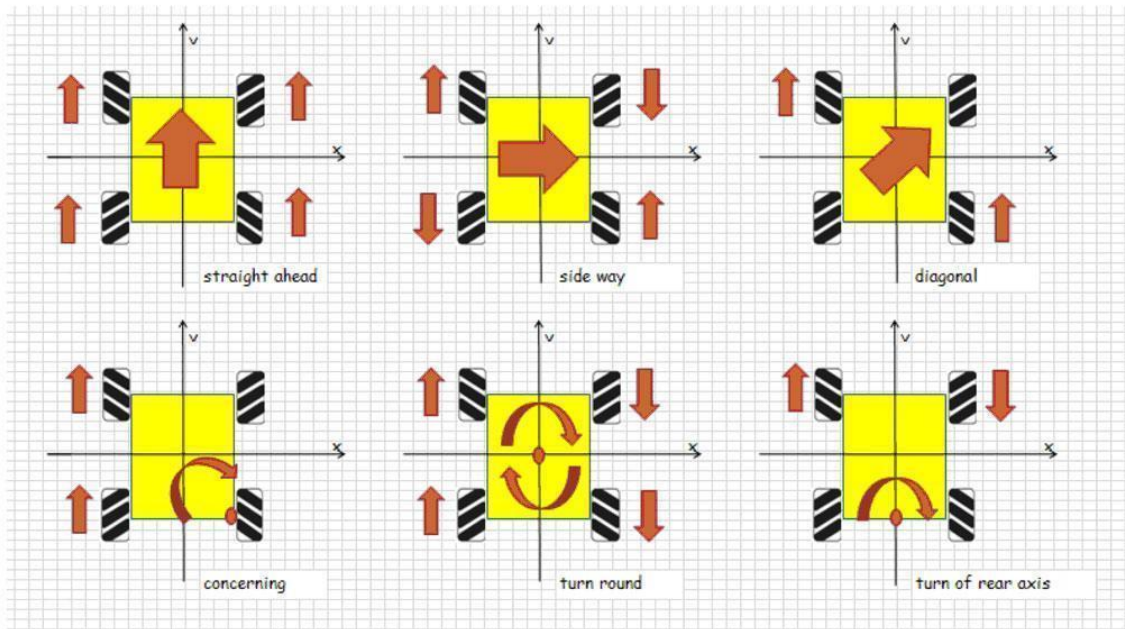


Рисунок 2.24 Схема керування всепрямованими роботами Mecanum-wheels

- Шр – сферичний рушійний елемент;
- Ру – рульовий механізм;
- У – керований (поворотний) рушійний елемент;
- Л – опорний елемент типу «лижний опорник»;
- Н – опорний елемент типу «нога»;
- Р – опорний елемент типу «важіль»;
- Б – опорний елемент типу балансиране (рояльне) колесо;
- Кп – опорний елемент типу пасивне колесо з нерухомою віссю;
- П – пасивний шарнір (ступінь рухливості);
- А – активний шарнір (ступінь рухливості);

- Ш – шарнір механізму крокування;
- М – маніпулятор (виконавчий пристрій мобільного робота);
- Пр – механізм стрибання;
- С – секція шарнірної машини;
- мС – позначення багатомодульної шарнірної схеми;
- мР – позначення багатомодульної реконфігурованої схеми;
- S – система динамічної стабілізації, подібна до тієї, що використовується в мобільних платформах фірми Segway.

Допущення у позначеннях схем:

За замовчуванням вважається, що у колісній схемі кожне колесо має індивідуальний привід, якщо це не зазначено додатково. Таким чином, загальноприйнята для колісної техніки формула типу 4×4 (чотири колеса, усі чотири ведучі) у межах цих позначень скорочується до 4К. Аналогічно формула 6×6 записується як 6К тощо.

Правила складання формул:

Кожен елемент формули позначається великою літерою або комбінацією великої та малої літер (наприклад: К – колесо, А – активна (приводна) ступінь рухливості, Ру – рульовий механізм, мС – багатомодульна шарнірна схема).

Цифра перед позначенням елемента вказує кількість таких елементів (наприклад: 4К – чотириколісний рушій, 2А – два активні шарніри механізмів зміни геометрії шасі).

Елементи формули, які мають однакову кількість і йдуть поспіль, для скорочення запису об'єднуються під однією цифрою, яка стосується всіх цих елементів. Наприклад: запис 6К6У (шестиколісна схема з шістьма керованими колесами) скорочується до 6КУ, а запис 4Г4У4А (чотиригусенична схема з чотирма приводами повороту гусениць навколо вертикальної осі та чотирма приводами обертання модулів) скорочується до 4ГУА.

Знак «×» використовується для позначення нетипової для мобільних роботів колісної формули. Наприклад: 4×2К (автомобільна схема: чотири колеса, два ведучі) або 12×4К (дванадцятиколісна схема з чотирма приводними модулями). Також цей знак може вказувати, що елементи, зазначені після нього, входять у кожен із елементів перед ним. Наприклад: 2СП×2Г1П (двосекційна шарнірна схема з двоступеневим з'єднувальним шарніром, кожна секція якої є двогусеничним шасі з одним пасивним шарніром).

Знак «/» використовується як роздільник, наприклад, для відокремлення двох послідовних літер, якщо цифра перед ними стосується лише першої літери. Наприклад: 6К/М3А – мобільний робот із шестиколісним рушієм без механізмів зміни геометрії, який використовує триступеневий маніпулятор для підвищення прохідності.

Формула шасі складається за покроковою послідовністю аналізу, враховуючи конструкцію шасі (кількість рушійних елементів – кількість керованих елементів – тип і склад адаптаційних механізмів).

Класифікація схем шасі (СШ)

Найбільше поширення серед колісних СШ завдяки їхній простоті отримали чотири- та шестиколісні схеми (4К і 6К) з індивідуальними приводами коліс (мотор-колеса) та бортовим способом повороту. Двоколісні СШ через свою статичну нестабільність розрізняються за типом застосованої пасивної опори (у вигляді важеля або «хвоста» – схема 2К, у вигляді балансиру колеса – схема 2К/Б) або використовують систему динамічної стабілізації на основі гіроскопічних датчиків (схема 2К/S).

Існують чотириколісні СШ «автомобільного» типу із задніми ведучими та передніми керованими колесами (схема 4×2К1Ру). Як правило, подібні розробки базуються на досвіді створення радіокерованих моделей машинок (РС-моделі). Також знаходять застосування схеми з усіма керованими колесами..

Розвитком простих колісних схем стало створення СШ із пасивними шарнірами зламу корпусу (наприклад, схема 6К1П), шарнірами підвіски (марсоходи NASA (США) – 6КУ4П, 6К3П та ін.) і приводними (активними) ступенями рухливості (схема 6К1А, 8К1А та ін.), що сприяють підвищенню прохідності МР.

Подальше підвищення ефективності та функціональності колісного рушія для наземних РТС пов'язане, по-перше, з удосконаленням конструкції колеса (розробки Macroswiss тощо), по-друге, зі створенням гібридних СШ (колісно-крокуючі, колісно-гусеничні, колісно-важільні).

Найбільш поширені гусеничні СШ типу 2Г у порівнянні зі схемами 2К і 6К забезпечують нижчий питомий тиск і вищі тягові можливості на слабонесучих ґрунтах, але відрізняються дещо більшою конструктивною складністю і більшими динамічними навантаженнями при подоланні перешкод. Для підвищення прохідності застосовуються схеми гусеничних рушіїв із змінною геометрією (3Г) різних конструкцій: від двогусеничних СШ із змінним положенням важелів із направляючими колесами (2Г1А) до шестигусеничних. У останніх, зокрема, для одного борту використовується одна основна і дві додаткові гусениці, що обертаються від одного спільного привода. Додаткові гусениці встановлюються на поворотних важелях із кінцевими роликками та направляючими гусеницями. Важелі можуть обертатися як від загальних приводів (схема 6Г2А), так і кожний від свого індивідуального привода (6Г4А).

Прагнення до підвищення ефективності гусеничних СШ призвело до створення різних гібридних схем.

Гібридні (комбіновані) схеми СШ є спробою поєднати переваги різних механізмів пересування. В першу чергу це стосується розробок колісно-крокуючих механізмів. До цього класу належать «крокуючі» (точніше, псевдокрокуючі) СШ,

схожі на робот RHEX. Принцип пересування цього робота (БРШ) заснований на одноосевому обертанні кожного з шести важелів спеціальної конфігурації, тому такий спосіб умовно відносять до «колісного» крокування.

Інші гібридні схеми включають використання маніпуляторів (6К/М3А, 2Г/М2А) та створення стрибучих роботів. Особливе місце серед адаптивних РТС займають розробки реконфігурованих шасі, наприклад, схеми з трьома варіантами переналаштування (6Г2А, 4К, 4КР2А).

Перспективним напрямом є створення зчленованих СШ, які розвиваються у бік багатомодульних, реконфігурованих, самоорганізуючихся РТС.

Крокуючі роботи. Цей тип мобільних платформ став актуальним відносно нещодавно, що зумовлено значною складністю їх конструкції та управління. Незважаючи на це, в певних умовах крокуючі роботи здатні замінити колісні та гусеничні аналоги. Головною перевагою крокуючих роботів є їхня здатність рухатися по діагоналях, описувати коло без зміни положення корпусу, долати перешкоди, підніматися сходами, змінювати висоту і нахил корпусу, а також пересуватися по сходах. Це лише частина їхніх можливостей.

Розглянувши переваги крокуючих платформ, слід проаналізувати їхні різновиди, сильні та слабкі сторони представлених конструктивних рішень.

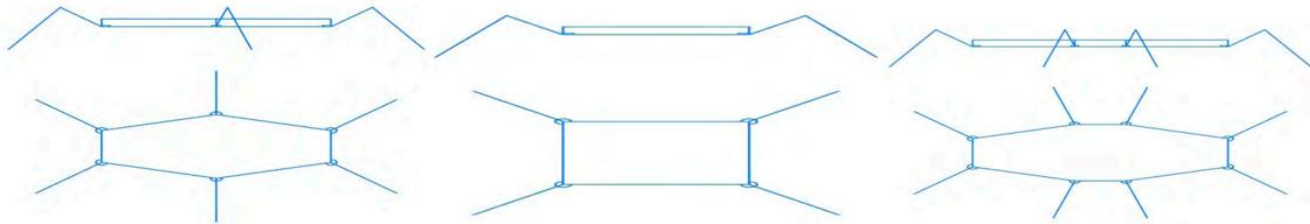
На даний момент найбільш поширеними є два типи крокуючих платформ. Перший тип імітує рухи тварин, зокрема ссавців (марсохід за схемою 2ЛШ), а другий – членистоногих (робот-павук Asterisk, схема 6Н24Ш). Основні відмінності між цими системами полягають у конструкції кінцівок, їхній кількості та особливостях руху. У випадку платформ, що імітують ссавців, часто використовують конструкції типу «робот-собака» (наприклад, Little Dog, схема 4Н8Ш). Вони імітують собак як функціонально, так і конструктивно. У таких платформах лапи можуть бути однаковими або парними: наприклад, передні й задні лапи мають різну конструкцію (рис.2.25). Цей тип роботів відзначається високою швидкістю пересування та здатністю бігати.



Рисунок 2.25 Основні види конструкцій роботів собак

У конструкціях, що імітують членистоногих, різноманіття набагато більше. Це пояснюється простішою конструкцією та ширшим спектром можливостей. Зазвичай у таких платформах використовують три основні варіанти: квадропод,

гексапод і октопод, які мають відповідно чотири, шість і вісім лап (рис. 2.26). Найпоширенішою є шестинога платформа, оскільки вона забезпечує оптимальний баланс між простотою та стійкістю.



Гексапод

Квадропод

Октопод

Рисунок 2.26 Основні види конструкцій крокуючих роботів, що імітують членистоногих

Цей тип платформ відрізняється значною варіативністю у конструкції лап, що обумовлено різноманіттям завдань, які вони виконують. Виділяють три основні типи конструкцій лап залежно від кількості приводів. Перший тип має два приводи, що максимально спрощує конструкцію та знижує її вартість. Другий тип обладнаний трьома приводами, що ускладнює конструкцію, але значно розширює можливості руху. Лапи в цьому випадку можуть виконувати функції простих маніпуляторів. У третьому типі використовується більше трьох приводів, що забезпечує найвищу маневровість, але водночас підвищує складність конструкції та її вартість.

Загалом крокуючі роботи можна умовно поділити на два основні типи: дослідницькі (розвідувальні) та робочі (технічні). Ці типи найбільш точно відображають їхні можливі сфери застосування та допомагають визначити придатність до виконання різних завдань. Потенціал застосування крокуючих роботів є вищим на складних, непідготовлених поверхнях. Крокуючі роботи можуть бути більш ефективними у використанні, ніж колісні та гусеничні, у сільській місцевості, під час ліквідації природних і техногенних катастроф, а також у військовій сфері.

2.4.2 ПОВІТРЯНІ РОБОТОТЕХНІЧНІ СИСТЕМИ

2.4.2.1 Конструкції планерів

БПЛА літакового типу (з нерухомим крилом)

БПЛА літакового типу оснащений нерухомими крилами, які створюють підйомну силу та забезпечують політ. Управління здійснюється дистанційно або автоматично за допомогою комп'ютера. Найбільш функціональними вважаються моделі з монокрилом.

Такі дрони здатні долати великі відстані, залишаючись у повітрі тривалий час, та відзначаються високою швидкістю у порівнянні з іншими типами безпілотників. Вони є відносно недорогими, простими в обслуговуванні та ремонті. Зазвичай

польоти виконуються за заздалегідь визначеною програмою, яка включає маршрут, ціль і налаштування датчиків. Після завершення місії літаковий БПЛА повертається на стартову точку або інше задане місце. Однак для таких дронів потрібні спеціально облаштовані майданчики для зльоту та посадки.

БПЛА вертолітного типу (однороторні коптери)

Однороторні дрони конструктивно нагадують вертольоти. Вони оснащені одним основним несучим гвинтом і, за необхідності, додатковим – на хвості для керування траєкторією польоту. Рух вперед таких апаратів забезпечується за рахунок автомату перекоосу - прецизійної силової механіки яка має високу вартість як виробництва так і експлуатації (рис. 2.27). Завдяки двигунам внутрішнього згоряння однороторні коптери мають триваліший час польоту порівняно з квадрокоптерами. Такі дрони можуть використовувати різні типи двигунів. Разом із тим, їх конструкція складніша, енерговитрати вищі і обслуговування коштує дорожче, ніж у мультироторних БПЛА. Тому в БПЛА такі схеми використовувались раніше до появи потужних та компактних акумуляторів таких як літій-полімерні.



Рисунок 2.27. БПЛА Yamaha RMAX призначений для розпилення бакових сумішей. Може нести до 31 кг корисного навантаження. Вартість на початку 90-х років біля 100 тисяч \$.

Мультироторні БПЛА

Мультироторні дрони – це платформи з трьома або більше безколекторними двигунами, кожен із яких оснащений пропелером. Під час польоту вони перебувають у горизонтальному положенні, здатні зависати в повітрі, рухатися у будь-якому напрямку та обертатися навколо своєї осі. Маневрування забезпечується зміною тяги на кожному моторі.

Ці дрони прості у керуванні, мають високу точність позиціонування та відносно недорогі. Недоліки включають низьку швидкість, обмежений радіус дії, високу енерговитратність, малу вантажопідйомність і короткий час польоту.

Розглянемо типові схеми:

Трикоптер – це БПЛА з трьома променями, кожен із яких оснащений мотором (рис. 2.28). Передньою частиною вважається місце з'єднання двох променів (конфігурація Y3). Кут між променями зазвичай становить 120° . Для компенсації гіроскопічного ефекту від непарної кількості роторів та для зміни кута повороту задній мотор повинен обертатися, що досягається встановленням звичайного RC сервоприводу. У конфігурації Y4 сервопривід замінюють парою соосних моторів на задньому промені. Перевагами трикоптера є незвичайний дизайн, найкращі льотні характеристики під час прямолінійного польоту та низька вартість завдяки меншій кількості моторів і регуляторів ESC. *ESC - Electronic speed controller / електронний регулятор швидкості - це пристрій, що приєднується до акумулятору, мотору та контролеру польоту і контролює швидкість обертання мотору.* Трикоптер має асиметричну конструкцію, вимагає використання сервоприводу та складний у виготовленні заднього променя. До того ж не всі контролери польоту підтримують цю конфігурацію.

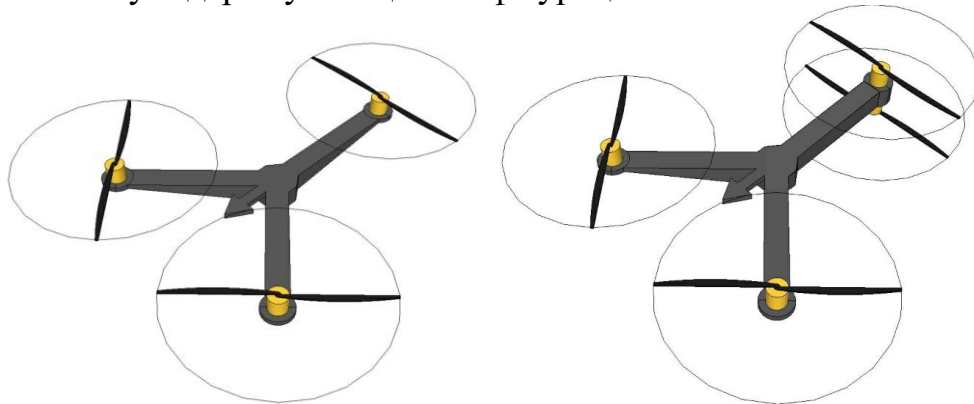


Рисунок 2.28. Трикоптер. Конфігурація Y3 (ліворуч) та Y4 (праворуч).

Квадрокоптер – це БПЛА з чотирма променями, кожен із яких оснащений мотором. У конфігурації «X» передньою частиною вважається сторона між двома променями, а у конфігурації «+» – поздовжній промінь (рис. 2.29). Це найпоширеніший мультироторний дизайн завдяки своїй простоті, універсальності та симетрії променів і моторів відносно двох осей. Усі сучасні контролери польоту підтримують цю конфігурацію. Основним недоліком квадрокоптера є відсутність надлишковості, що означає падіння у разі збою системи, особливо елементів силової установки.

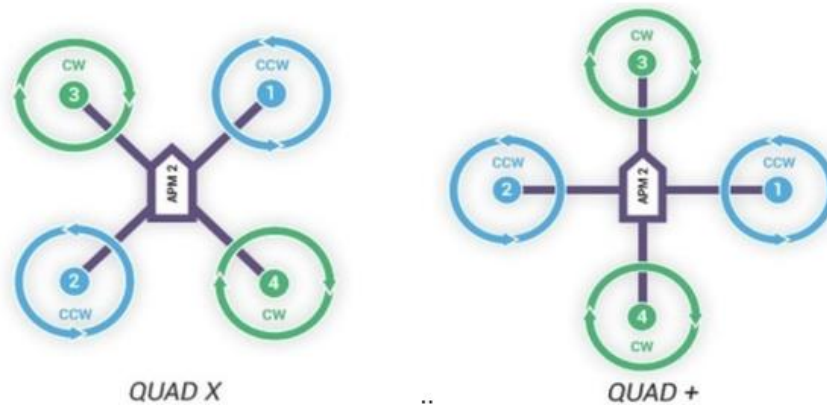


Рисунок 2.29. Схема дії гвинтів квадрокоптера.

Гексакоптер має шість променів, кожен із яких оснащений мотором. Передньою частиною вважається сторона між двома променями або поздовжній промінь (рис. 7). Його конструкція дозволяє легко додати два додаткові промені з моторами для збільшення тяги, що дає змогу піднімати більший вантаж. У разі виходу з ладу одного мотора дрон може здійснити м'яку посадку. Гексакоптери мають модульну конструкцію рами й підтримуються більшістю контролерів польоту. Недоліками є громіздкість, висока вартість і необхідність використовувати більш ємні акумулятори через додаткову вагу.

Y6 – це гексакоптер із трьома променями, кожен із яких оснащений парою соосних моторів - усього шість моторів (рис. 2.30).

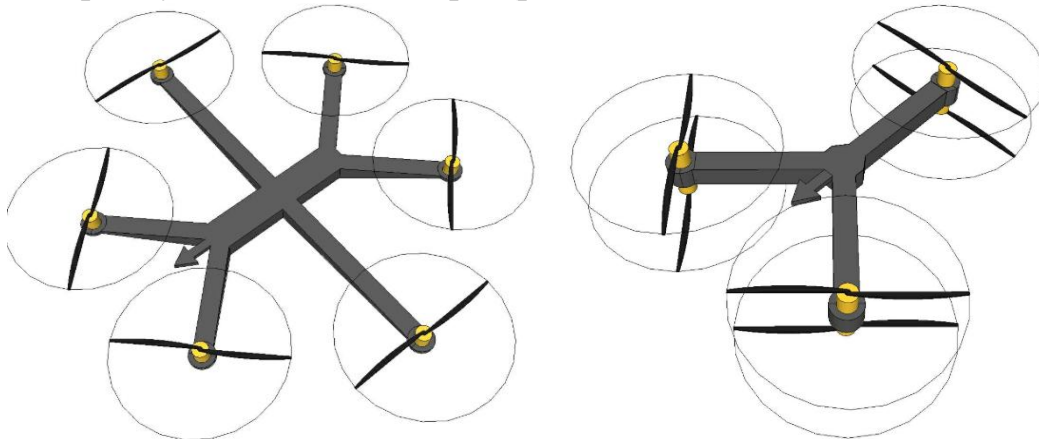


Рисунок 2.30 Гексакоптер (ліворуч) та Y6 (праворуч)

Нижні пропелери спрямовують тягу вниз. У порівнянні з гексакоптером він має менше компонентів, більшу вантажопідйомність і виключає гіроскопічний ефект завдяки зустрічному обертанню пропелерів. У разі виходу з ладу одного мотора дрон, має певні шанси здійснити м'яку посадку. Проте Y6 дорожчий за квадрокоптер, має більшу вагу через додаткові мотори та трохи нижчу тягу через взаємний вплив пропелерів. Крім того, не всі контролери польоту підтримують цю конфігурацію. Октокоптер має вісім променів, кожен із яких оснащений мотором. Передньою частиною вважається сторона між двома променями. Більша кількість

моторів забезпечує високу тягу та надлишковість, що робить його ідеальним для перенесення важкого й дорогого обладнання, наприклад DSLR-камер. Водночас октокоптер коштує дорого, потребує потужного акумулятора та використовується переважно у професійній сфері.

X8 – це різновид октокоптера з чотирма променями, кожен із яких оснащений парою соосних моторів (усього вісім моторів). Він має всі переваги октокоптера, включаючи велику тягу та надлишковість, що дає більше шансів м'яко посадити дрон у разі відмови мотора. Однак X8 дорожчий, важчий і потребує потужного акумулятора, через що його використовують лише в професійних цілях.

БПЛА змішаного типу (гібридні)

Гібридні дрони поєднують характеристики літакових і мультироторних БПЛА. Поворотні або фіксовані гвинти працюють як підйомні під час зльоту й посадки, а під час звичайного польоту – як тягові.

До переваг належать вертикальний зліт, висока швидкість, маневровість, оптимальне співвідношення маси та корисного навантаження, а також економія енергоресурсів. Серед недоліків – складність обслуговування, залежність від погодних умов і висока вартість.

БПЛА аеростатичного типу (аеростати)

Аеростатичні БПЛА піднімаються за рахунок балонів, наповнених газом або нагрітим повітрям. Зазвичай це непілотовані дирижаблі, які поєднують аеростат і електродвигуни з гвинтами (рис. 2.31). Політ і переміщення здійснюються за допомогою системи управління орієнтацією.

Перевагами є тривалість польоту до кількох днів або навіть тижнів та висока вантажопідйомність. Водночас аеростати мають низьку маневровість, обмежену швидкість, значну залежність від погодних умов, а також великі розміри та масу.



Рисунок 2.31 БПЛА аеростатного типу фірми Minizepp (Швейцарія)

2.4.2.2 Силова установка БПЛА

Загальні поняття.

Вибір двигунів для силової установки БПЛА визначає максимальне навантаження, час та висоту польоту. В мультикоптерних апаратах силова установка має складатися з двигунів однієї марки та моделі оскільки це забезпечує збалансовану роботу. На практиці навіть при таких заходах відбувається незначна різниця в роботі двигунів яку потрібно компенсувати польотним контролером.

Стосовно двигунів для БПЛА потрібно враховувати специфічний параметр **KV** який означає максимальну кількість обертів двигуна на один вольт поданого напруги без втрати потужності. Для більшості багатороторних БПЛА важливе низьке значення KV (наприклад, 500–1000), що забезпечує стабільність. Для акробатичних польотів актуальними є значення KV у межах 1000–1500 із пропелерами меншого діаметра.

Наприклад, якщо KV двигуна становить 650 об./В, то при напрузі 11,1 В двигун обертається зі швидкістю: $11,1 \times 650 = 7215$ об./хв.

При використанні нижчої напруги (наприклад, 7,4 В), частота обертів зменшиться: $7,4 \times 650 = 4810$ об./хв.

Варто пам'ятати, що зниження напруги зазвичай призводить до збільшення споживаного струму (оскільки потужність = струм \times напруга).

Деякі виробники безщіткових двигунів зазначають у специфікації максимальну тягу (**Thrust**), яку може створювати двигун із рекомендованим пропелером. Тяга зазвичай вимірюється в кілограмах (кг), фунтах (lbs) або ньютонів (N).

Наприклад, якщо ви будете квадрокоптер із тягою одного двигуна 0,5 кг (з 11-дюймовим пропелером), чотири такі двигуни забезпечать максимальну тягу: 2 кг. Відповідно, якщо загальна вага квадрокоптера близька до 2 кг, він зможе злетіти лише на максимальних обертах. У цьому випадку варто або зменшити вагу дрона, або вибрати потужнішу комбінацію «двигун + пропелер».

Щіткові vs Безщіткові двигуни

Для БПЛА критичними є висока потужність при мінімальній масі, відповідно були розроблені спеціальні двигуни, охолодження яких здійснюється за рахунок повітряного потоку від пропелерів БПЛА. Найчастіше використовуються щіткові та безщіткові двигуни. У щіткових (Brushed) двигунах ротор із обмоткою обертається всередині статора, на якому магніти закріплені нерухомо. У безщіткових (Brushless) двигунах усе навпаки: обмотка закріплена на внутрішній частині статора, а магніти розташовані на валу і обертаються разом із ним.

У більшості випадків варто звертати увагу на безщіткові двигуни постійного струму (рис. 2.32). Вони широко використовуються для створення найрізноманітніших пристроїв.



Рисунок 2.32 Безколекторний двигун для БПЛА T-Motor MN6007 II. Вага 180 грам, максимальна потужність 936 Вт.

Безщіткові двигуни типу «Panсake» відрізняються більшим діаметром, більш плоскою конструкцією, високим крутним моментом і низьким значенням KV. У невеликих БПЛА (зазвичай розміром із долоню) часто використовують маленькі щіткові двигуни через їхню нижчу вартість і прості контролери з двома проводами. Хоча безщіткові двигуни бувають різних розмірів, вибір меншого двигуна не завжди означає зменшення ціни.

Inrunner vs Outrunner

Існує кілька типів безщіткових двигунів постійного струму:

- **Inrunner** – внутрішній ротор. Обмотка нерухомо закріплена на статорі, а магніти знаходяться на валу ротора, що обертається. Зазвичай їх використовують у радіокерованих човнах, вертольотах і автомобілях через високе значення KV.
- **Outrunner** – зовнішній ротор. Магніти розташовані на зовнішньому статорі, який обертається навколо нерухомої обмотки. Такі мотори зазвичай мають більший крутний момент.
- **Hybrid Outrunner** – технічно є «Outrunner», але виконаний у корпусі «Inrunner». Це рішення поєднує високий крутний момент зовнішнього ротора і захищену конструкцію без зовнішніх рухомих частин, як у Inrunner.

Пропеллери

Більшість пропелерів багатороторних БПЛА мають дві або три лопаті. Найбільш поширеними є дволопатеві пропелери. Варто зазначити, що збільшення кількості лопатей не завжди призводить до зростання тяги: кожна лопать працює в потоці, збуреному попередньою, що знижує ефективність пропелера. Пропелер малого діаметра має меншу інерцію, тому його легше розганяти й зупиняти, що є важливим для польотів в складних умовах.

Тяга, яку створює пропелер, залежить від щільності повітря, швидкості обертання, діаметра, форми, площі лопатей і їхнього кроку. Ефективність пропелера визначається кутом атаки (рис. 2.33), що є різницею між кроком лопаті та кутом спіралі (кутом між напрямком обертання лопаті й результативною

відносною швидкістю). Ефективність пропелера – це відношення вихідної потужності до вхідної.

Добре спроектовані пропелери зазвичай мають ККД понад 80%. На ефективність впливають відносна швидкість і передній край лопаті, який повинен бути максимально гладким. Хоча пропелери з регульованим кроком є найефективнішими, їхня складність у конструкції майже виключає їхнє використання в багатороторних системах.

Пропелери можуть обертатися за годинниковою стрілкою (CW) або проти неї (CCW). Напрямок обертання визначається нахилом лопатей (дивлячись на пропелер збоку). Якщо права кромка лопаті вища – це CCW, якщо ліва – CW. У разі використання перевернутого розташування двигунів (наприклад, конфігурації Y6, X8) напрямок обертання пропелерів потрібно змінити так, щоб тяга була спрямована вниз.

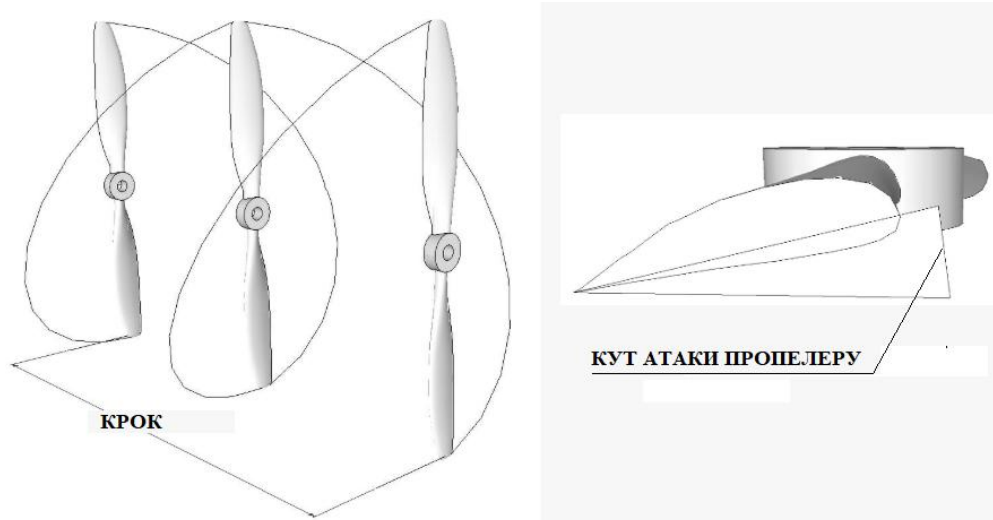


Рисунок 2.33. Кут атаки та крок пропелеру

Лицьова сторона пропелера завжди має бути спрямована вгору. У документації до контролера польоту зазвичай зазначається напрямок обертання кожного пропелера для підтримуваних конфігурацій багатороторних систем.

Пропелери є високотехнологічними пристроями, складними як за геометрією так і за матеріалом. Використовують як цільні так і складні пропелери. Складні пропелери мають центральну частину, з'єднану з двома рухомими лопатями (рис. 2.34). Під час обертання центрифугальні сили виводять лопаті в робоче положення, і пропелер стає жорстким. Через низький попит і складність конструкції складні пропелери зустрічаються рідше. Їхня основна перевага – компактність: у поєднанні зі складною рамою можна значно зменшити розмір дрона для транспортування. У разі аварії замінюється лише пошкоджена лопать, що також є перевагою.



Рисунок 2.34. Складаний пропелер

Найчастіше для виготовлення використовують пластики та композитні матеріали. Пластик (ABS/нейлон тощо) – найпопулярніший вибір для багатороторних БПЛА через низьку ціну, хороші характеристики й високу довговічність. Під час аварії зазвичай ламається щонайменше один пропелер, тому на етапі навчання завжди потрібно мати запас. Жорсткість і ударостійкість пластикових пропелерів можна покращити, додавши вуглецеві волокна. Композити (армовані полімери) – сучасний і міцний матеріал. Пропелери з вуглецевого волокна важче зламати, вони забезпечують мінімальні втрати ефективності й зазвичай не потребують балансування. Проте вони дорожчі та можуть завдати більшої шкоди в разі зіткнення.

Контролерне обладнання.

8051 vs AVR vs PIC vs ARM: Сімейства мікроконтролерів, що є основою більшості сучасних польотних контролерів Arduino базується на AVR (рис. 2.35), і спільнота, схоже, зосереджена на MultiWii як на кращому коді. Microchip є основним виробником чипів PIC. Твердження про те, що одне краще за інше, некоректне — усе залежить від можливостей програмного забезпечення. ARM (наприклад, STM32) використовує 16/32-бітну архітектуру, тоді як AVR і PIC здебільшого використовують 8/16-бітні процесори.

З появою одноплатних комп'ютерів, які стають дедалі дешевшими, очікується розвиток польотних контролерів нового покоління, здатних працювати на повноцінних операційних системах, таких як Linux або Android. Впровадження комп'ютерів до складу бортових комплексів БПЛА є необхідним для реалізації автономного польоту. Це стає невід'ємною умовою з огляду на подальший розвиток безпілотних технологій у напрямку застосування рою — коли БПЛА взаємодіють між собою в режимі реального часу. За наявності технічних засобів машинного зору можна інтегрувати перспективні технології візуальної орієнтації на місцевості. Завдяки використанню машинного зору, дрони можуть орієнтуватися на місцевості

без залежності від GPS, що особливо важливо в умовах поганого сигналу супутникової навігації або його повної відсутності. Це забезпечує високу точність позиціонування, оскільки дрони можуть аналізувати навколишнє середовище, розпізнавати наземні об'єкти та коригувати маршрут.

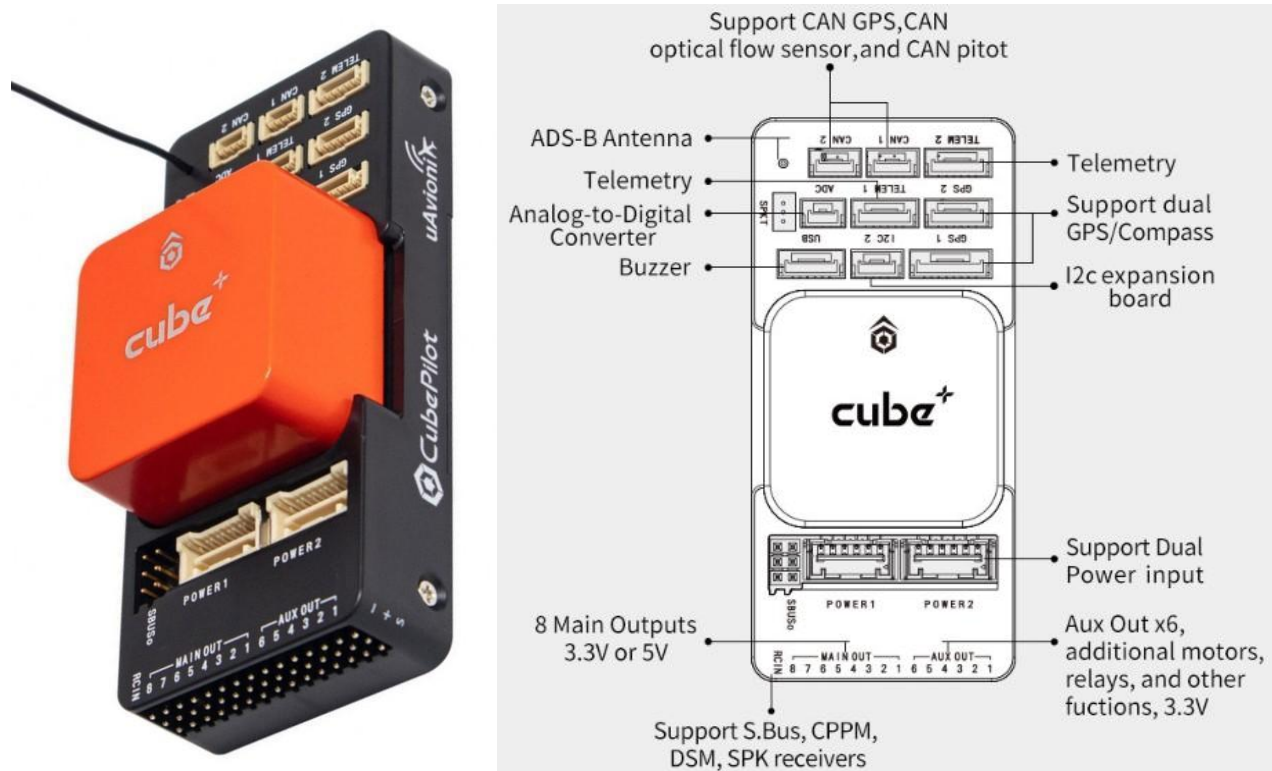


Рисунок 2.35. Польотний контролер **CUBE ORANGE+ Std Set** (HS 9014.20.00) з вбудованим приймачем ADS-B для пристроїв на базі Ardupilot. Зовнішній вигляд (ліворуч) призначення портів (праворуч).

Візуальна орієнтація є важливим елементом для впровадження технологій рою, де кілька дронів взаємодіють між собою, координуючи дії в реальному часі. Інтеграція систем машинного зору з алгоритмами штучного інтелекту дозволяє дронам не лише орієнтуватися за візуальними орієнтирами, але й розпізнавати об'єкти, оцінювати ситуацію та приймати автономні рішення.

Такі рішення вже реалізуються на практиці зокрема в конструкторі програмованого квадрокоптеру «Клевер» (рис. 2.36). Набір включає політний контролер COEX Pix із політним стеком PX4, Raspberry Pi 4 як керуючий бортовий комп'ютер, модуль камери для реалізації польотів із використанням комп'ютерного зору, а також набір різноманітних датчиків і іншої периферії. Платформа Клевер також містить попередньо налаштований образ для Raspberry Pi з повним набором необхідного програмного забезпечення для роботи з усією периферією та програмування автономних польотів. Вихідний код платформи Клевер і цієї документації є відкритим і доступним на GitHub (<https://github.com/CopterExpress/clover>).



Рисунок 2.36. Квадрокоптер Клевер із інтегрованим бортовим моноплатним комп'ютером Raspberry Pi

Отже інтеграція бортового комп'ютера надає додаткові можливості в той час як політний контролер являє собою центральний вузол управління який інтегрує сенсори, виконавчі механізми та системи зв'язку для реалізації заданих завдань у реальному часі.

Більшість мікроконтролерів мають цифрові та аналогові порти для підключення датчиків, сервоприводів, підвісів, зумерів тощо. Частина сенсорів може бути відразу інтегрована в польотний контролер (в тому числі із кратним дублюванням).

Контролер польоту зазвичай включає такі вбудовані сенсори:

- **Акселерометр:** вимірює лінійне прискорення за трьома осями (X, Y, Z) і може визначати напрямок «вниз»;
- **Гіроскоп:** вимірює кутові швидкості за трьома осями (альфа, бета, гамма);
- **Компас.**
- **Барометр.**

В якості прикладу реалізації таких пристроїв у вигляді мікросхеми можна привести модуль MPU-6050, який є одним із найпоширеніших мікромеханічних інтегрованих 6-осьових пристроїв для відстеження руху (рис. 2.37). Він поєднує 3-осьовий гіроскоп, 3-осьовий акселерометр і цифровий процесор руху (DMP). Завдяки підтримці I2C-шини модуль може приймати сигнали від зовнішнього 3-осьового компаса, що збільшує кількість вимірюваних параметрів до 9. MPU-6050 також оснащений вбудованим датчиком температури, а за необхідності може бути доповнений датчиком тиску.

Оскільки сигнали від чутливих елементів в модулі MPU-6050 подаються в аналоговому форматі, модуль містить шість 16-розрядних аналого-цифрових перетворювачів для обробки даних гіроскопа та акселерометра. Діапазон вимірювань можна налаштувати: для гіроскопів — від ± 250 до ± 2000 $^{\circ}/\text{с}$, для акселерометрів — від $\pm 2g$ до $\pm 16g$. Для підключення до зовнішніх датчиків використовується високошвидкісна I2C-шина з частотою до 400 кГц.

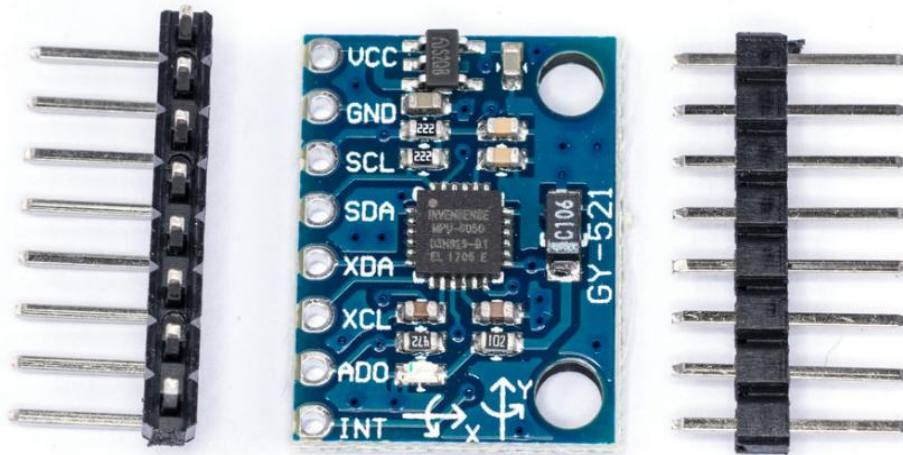


Рисунок 2.37 Модуль MPU-6050 встановлений на платі для інтеграції у зовнішні пристрої. Габарити модуль 4×4×0,9 мм.

Модуль підтримує два режими роботи: основний і режим прямого проходження. В основному режимі MPU-6050 виступає ведучим пристроєм, виконуючи опитування вбудованих або зовнішніх датчиків, таких як датчик температури. У цьому режимі можна зчитувати до 24 байт даних із 4 додаткових датчиків. У режимі прямого проходження роль ведучого виконує зовнішній комп'ютер або мікроконтролер, при цьому логіка управління I2C-шиною на модулі відключається, а управління передається зовнішньому ядру.

При використанні MPU-6050 слід враховувати фактори, які можуть впливати на точність вихідних даних. Через конструктивні особливості пристрою та вплив зовнішніх умов вихідні сигнали можуть бути спотворені, що робить необхідним застосування методів обробки даних.

Під час тривалої роботи з модулем через дрейф нуля виникають значні похибки в даних гіроскопа. Для мінімізації цих похибок виробник інтегрував цифровий процесор руху (DMP) з алгоритмами, розробленими компанією InvenSense. DMP обробляє дані акселерометра, гіроскопа та додаткових датчиків, мінімізуючи вплив похибок кожного сенсора. Для зручності роботи з DMP користувачам надаються спеціальні бібліотеки, зокрема популярна "MPU6050_6Axis_MotionApps20.h", яка забезпечує доступ до функцій розрахунку та перетворення даних.

В залежності від потужності польотного контролеру та програмного забезпечення керуванням БПЛА можливі різні режими польоту:

- **ACRO**: використовується лише гіроскоп. Підходить для акробатичних польотів.
- **ANGLE**: стабільний режим; використовується гіроскоп і акселерометр.
- **HORIZON**: комбінує стабільність ANGLE і маневреність ACRO.

- **BARO (Altitude Hold):** утримання висоти за допомогою гіроскопа, акселерометра та барометра.
- **MAG (Heading Hold):** утримання курсу за допомогою компаса (yaw).
- **HEADFREE:** ігнорується орієнтація yaw, дозволяючи рух відповідно до положення стиків.
- **GPS/Return to Home:** автоматичне повернення до місця зльоту за допомогою GPS.
- **GPS/Waypoint:** автономне слідування за попередньо встановленими точками.
- **GPS/Position Hold:** утримання позиції з використанням GPS і барометра.
- **Failsafe:** режим відмовостійкості, що активується при збоях у програмному забезпеченні.

2.4.2.3 Готові комплекти програмних рішень для польотних контролером Програмне забезпечення UAV Toolbox

UAV Toolbox надає інструменти та приклади готових вузлів для розробки, симуляції, тестування та розгортання застосунків для безпілотних літальних апаратів. За його допомогою можна розробляти автономні алгоритми польоту, планувати місії UAV та створювати контролери управління.

Одним із ключових компонентів UAV Toolbox є застосунок *Flight Log Analyzer*, який дозволяє інтерактивно аналізувати дані телеметрії, кути тангажу та показники датчиків, використовуючи поширені формати журналів польотів. Це забезпечує детальний розбір польотних характеристик і допомагає вдосконалювати алгоритми керування БПЛА.

Для настільної симуляції та тестування в контурі обладнання (HIL) UAV Toolbox дозволяє генерувати та тестувати різні сценарії польотів. У процесі симуляції можна моделювати вихідні параметри датчиків. Ці дані можуть бути змодельовані у фотореалістичному 3D-середовищі або спрощеному 2.5D-середовищі симуляції. Вони використовуються для тестування алгоритмів стабілізації, навігації та ухилення від перешкод.

UAV Toolbox пропонує два середовища симуляції для тестування алгоритмів Cuboid Simulation Environment та Unreal Engine Simulation.

Cuboid Simulation Environment. Використовується для моделювання простих сценаріїв та генерації даних датчиків. Підходить для тестування:

- Контролерів;
- Алгоритмів навігації;
- Поєднання даних із різних датчиків.

Для роботи з цим середовищем застосовується об'єкт `uavScenario`. Cuboid Simulation Environment дозволяє швидко оцінювати алгоритми керування, перш ніж переходити до складніших тестувань у реалістичних середовищах.

Unreal Engine Simulation. Це середовище використовує графічний рушій Unreal Engine® від Epic Games®, що дозволяє візуалізувати:

- Реалістичні сценарії польотів;
- Дані лідача, камери та радара з високою частотою;
- Системи сприйняття у режимі «людина в контурі».

Це середовище доступне в Simulink і працює лише на платформі Windows®. Воно особливо корисне для тренування моделей штучного інтелекту, тестування систем ухилення від перешкод та перевірки автономних рішень на складних маршрутах.

UAV Toolbox містить приклади готових вузлів, зокрема автономну доставку вантажів мультикоптерами UAV. Підтримується генерація коду на C/C++, що спрощує:

- Прототипування нових автономних функцій;
- Програмно-апаратне тестування;
- Розгортання на реальних пристроях, таких як автопілот Pixhawk®.

Крім того, інструментарій включає в себе функції для адаптації алгоритмів до змінних умов навколишнього середовища, що робить його корисним для динамічних сценаріїв польотів.

Однією з опцій цього програмного забезпечення є сервіс ведення журналів польотів. Вони містять значні обсяги даних у різних форматах. UAV Toolbox підтримує імпорт і аналіз файлів телеметрії, зокрема:

- TLOG (журнали ArduPilot)
- ULOG (журнали PX4)

Підтримка MAVLink. MAVLink — це протокол зв'язку для обміну повідомленнями між БПЛА та наземними станціями. Він використовує шаблон публікації та підписки для передавання даних. У UAV Toolbox підтримуються функції MAVLink, що дозволяють:

- Визначати стандартні та користувацькі діалекти;
- Налаштовувати клієнтів;
- Надсилати та отримувати повідомлення.

Це дозволяє безпілотним системам легко інтегруватися в складніші мережеві рішення, забезпечуючи ефективне управління групами дронів або зв'язок із хмарними сервісами.

Отже UAV Toolbox надає потужні інструменти для розробки, симуляції та тестування алгоритмів безпілотних апаратів. Використовуючи цей тулбокс, можна ефективно проєктувати автономні системи управління БПЛА, аналізувати їхню поведінку та розгортати на реальних пристроях. Завдяки широким можливостям інтеграції, симуляції та аналізу польотних даних, UAV Toolbox є незамінним інструментом для розробників автономних літальних апаратів.

2.5 АКУМУЛЯТОРИ

Електрохімічні системи накопичення енергії зберігають її в електрохімічній (хімічній) формі та є найбільш поширеним типом пристроїв для зберігання енергії серед усіх наявних. Фактично, для компактних систем накопичення електроенергії сьогодні немає альтернатив.

Існують три основні типи хімічних (електрохімічних) джерел струму:

- Первинні гальванічні елементи, які необоротно перетворюють хімічну енергію на електричну.
- Вторинні (перезаряджувані) елементи або акумулятори, у яких електрохімічна реакція є зворотною, тобто після розрядки їх можна зарядити від зовнішнього джерела.
- Паливні елементи, які виділяють в окрему категорію.

Окремі електрохімічні елементи зазвичай з'єднують у батареї, щоб збільшити загальну напругу та ємність.

Останнім часом інтерес до акумуляторних батарей із великою ємністю значно зріс у зв'язку з розробкою електромобілів. Наприклад, американська компанія Tesla серійно випускає електромобілі з літій-іонними батареями ємністю до 100 кВт·год, що дозволяє новому акумулятору забезпечувати пробіг до 650 км на одному заряді.

На сьогодні існує щонайменше п'ять основних типів перезаряджуваних електрохімічних акумуляторів:

- Свинцево-кислотний, номінальна напруга одного елемента – 2 В.
- Нікель-кадмієвий (NiCd), номінальна напруга одного елемента – 1,2–1,3 В.
- Нікель-металогідридний (NiMH), номінальна напруга одного елемента – 1,2–1,3 В.
- Літій-іонний (Li-ion), номінальна напруга одного елемента – 3,6 В (із використанням електрода на основі літій-залізо-фосфату – 3,3 В).
- Літій-полімерний (Li-pol), номінальна напруга одного елемента – 3,7 В.

Незважаючи на низькі питомі характеристики (енергія до 40 Вт·год/кг), свинцево-кислотні акумулятори мають найнижчу вартість накопиченої енергії на кіловат-годину. Тому вони залишаються безальтернативним варіантом для використання як акумуляторні батареї стартових генераторів у транспортних засобах.

На рис. 2.38 показані питомі енергія (на одиницю маси) та енергетична щільність (на одиницю об'єму) для цих шести типів електрохімічних систем. Як видно, найбільш популярні літій-іонні та літій-полімерні акумулятори значно перевершують інші за цими параметрами. Наприклад, літій-полімерний акумулятор має енергетичну щільність до 300 Вт·год/л та питому енергію до 200 Вт·год/кг.

До недоліків цих типів акумуляторів належать великий рівень саморозряду під час зберігання та значне зменшення ємності при зниженні температури.

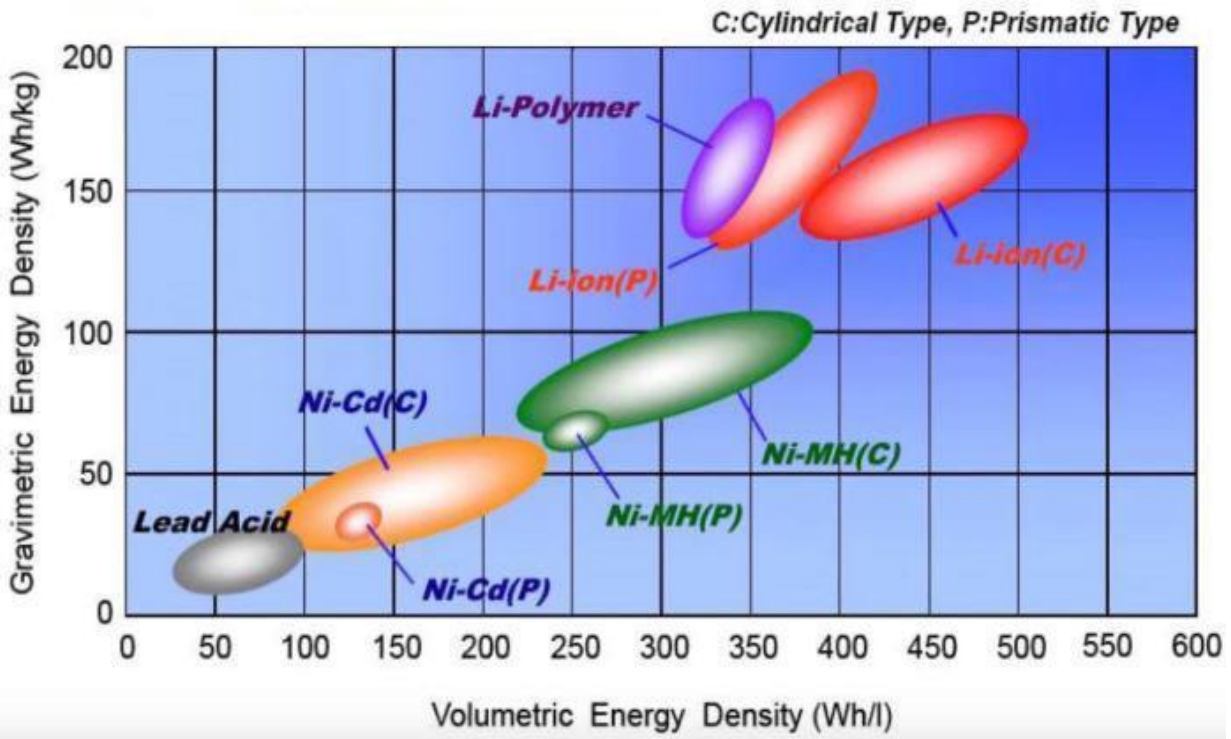


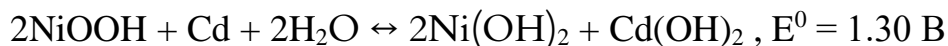
Рисунок 2.38. Графік ефективності різних видів акумуляторів.

Сьогодні найпоширенішими акумуляторами залишаються свинцево-кислотні (-) Pb|H₂SO₄|PbO₂(+).

Нікель-кадмієві акумулятори мають вищу питому енергоємність у порівнянні зі свинцево-кислотними – до 65 Вт·год/кг. Вони здатні ефективно працювати навіть при низьких температурах, включаючи температури нижче нуля.

Тривалість експлуатації цих акумуляторів становить від 100 до 500 циклів зарядки-розрядки, а термін служби – від 8 до 10 років.

Токоутворююча зворотна реакція для нікель-кадмієвих акумуляторів виглядає так:



Катодом у нікель-кадмієвих акумуляторів виступає гідрат оксиду нікелю Ni(OH)₂ з додаванням графітового порошку, електролітом є гідроксид калію (KOH), іноді з додаванням LiOH. Додавання LiOH дозволяє утворювати літій-нікелати, що збільшує ємність акумулятора на 21–25 %. Анодом служить гідрат оксиду кадмію Cd(OH)₂ або металевий кадмій Cd у вигляді порошку.

Науковці активно працюють над збільшенням кількості циклів зарядки-розрядки та зменшенням гістерезису, поте досягти рівня питомої потужності, характерного для літій-іонних акумуляторів, навряд чи вдасться.

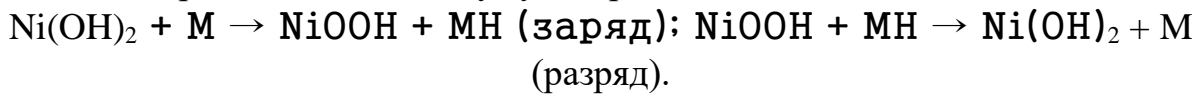
Нікель-металогідридний акумулятор (NiMH)

Це вторинне хімічне джерело струму, в якому: анод — металогідридний водневий електрод (зазвичай це гідрид нікель-лантану або нікель-літію), електроліт — гідроксид калію, катод — оксид нікелю.

NiMH-акумулятори почали розробляти ще в 1970-х роках як заміну нікель-кадмієвим акумуляторам. Проте спершу виникали труднощі з отриманням достатньо стабільних металогідридних сполук. Лише до 1980 року вдалося розробити такі матеріали для використання в акумуляторах.

Починаючи з кінця 1980-х років, нікель-металогідридні акумулятори активно вдосконалювали, зосереджуючись на збільшенні щільності накопиченої енергії. Вони мають приблизно на 20 % більшу ємність за однакових розмірів, порівняно з нікель-кадмієвими акумуляторами. Однак термін їхньої служби менший — від 200 до 300 циклів зарядки-розрядки. Рівень саморозряду у NiMH-акумуляторів приблизно у 1,5–2 рази вищий, ніж у нікель-кадмієвих.

Загальна реакція в NiMH-акумуляторі виглядає так:



Під час зарядки на негативному електроді метал перетворюється на гідрид. Під час розрядки відбувається зворотний процес. Основним матеріалом, який визначає характеристики NiMH-акумулятора, є водень-поглинаючий сплав, здатний поглинати об'єм водню, що перевищує його власний об'єм у 1000 разів. Найбільш поширеними є сплави типу LaNi_5 де частина нікелю замінена марганцем, кобальтом і алюмінієм, щоб підвищити стабільність і активність сплаву.

Для зниження вартості деякі виробники замість лантану використовують мішметал — суміш рідкісноземельних елементів, склад якої близький до пропорцій у природних рудах. Окрім лантану, такі руди містять церій, празеодим і неодим.

Літій-іонні акумулятори

Технології постійно вдосконалюються, і традиційні нікель-кадмієві та нікель-металогідридні акумулятори поступово замінюються літій-іонними. При майже однаковій масі одного елемента літій-іонні акумулятори мають значно більшу ємність: вони перевершують NiCd у 4–5 разів, а NiMH у 3–4 рази. Крім того, вони забезпечують вищу напругу на одному елементі — 3,6 В у порівнянні з 1,2 В у NiCd і NiMH. Це дозволяє замінити батарею з двох-трьох елементів одним літій-іонним.

Щодо кількості робочих циклів, літієві акумулятори займають проміжне положення між NiCd і NiMH. Даних про реальну кількість циклів поки що небагато, тому заявлені виробниками характеристики слід сприймати критично. Однак технології виготовлення Li-Ion-акумуляторів швидко розвиваються, що сприяє збільшенню їхнього терміну служби.

У літієвих акумуляторах анод виготовлений з металевого літію — одного з найактивніших хімічних елементів, найлегшого металу з найбільшим електрохімічним потенціалом. Це забезпечує високу щільність енергії та

максимальну теоретичну ємність. Однак висока хімічна активність літію ускладнює виробничий процес і вимагає особливої герметичності, що впливає на собівартість.

Довгий час літій-іонні акумулятори не могли бути використані комерційно через проблеми з безпекою. Якщо температура всередині акумулятора перевищує точку плавлення літію, відбувається хімічна реакція з електролітом, що може призвести до вибуху. Наприклад, у 1990-х роках багато літієвих акумуляторів, проданих у Японії, було повернуто через вибухи в мобільних телефонах, які спричинили опіки у людей. Подібний інцидент стався у 2003 році з телефонами Nokia, коли постраждало понад 20 осіб.

Для вирішення проблеми безпеки металевий літій замінили на його сполуки з іншими металами в оксидах. Спочатку використовували літій-кобальт оксид (LiCoO_2) як активний матеріал, але цей матеріал дорогий і нестабільний при відхиленнях напруги від норми (наприклад, 3,6 В — межі від 2,7 до 4,2 В). Тому акумулятори на його основі потребують спеціального контролера, що регулює напругу, струм і температуру.

Цікаво, що виробники свідомо відмовилися від виробництва літій-іонних акумуляторів у розмірах, схожих на побутові типи, щоб уникнути їх використання в непристосованих пристроях. Сьогодні під назвою «літієві батареї» об'єднують джерела енергії з різними хімічними складниками.

- літій/тіонілхлоридні (Li/SOCl_2);
- літій/оксид сірки (Li/SO_2);
- літій/оксид нікелю (Li/NiO_2);
- літій/оксид марганцю (Li/MnO_2);
- літій-залізофосфат (LiFePO_4).

Найбільш досліджений і технологічно відпрацьований тип літієвих батарей — це елементи на основі літію та оксидів марганцю (Li/MnO_2 і $\text{Li/Mn}_2\text{O}_4$). Вони є найбільш доступними за ціною серед усієї групи літієвих джерел живлення. Їхня ємність нижча, ніж у батарей на основі кобальту, але вони дешевші й не потребують складного контролера для керування процесами заряджання та розряджання.

Li/NiO_2 має вищу ємність порівняно з оксидами марганцю, але складніший у виробництві й менш безпечний. Щоб підвищити рівень безпеки в акумуляторах великої ємності, почали використовувати змішані оксиди кобальту й нікелю (20–30% нікелю).

Батареї Li/SOCl_2 характеризуються найвищою вихідною напругою (3,6 В), широким температурним діапазоном (від -55 до $+85$ °С), низьким рівнем саморозряду та невеликим типовим струмом розряду. Проте вони чутливі до високих температур. При значних струмах розряду через внутрішній опір батареї може виділятися тепло, що перевищує допустимий рівень. Тому до конструкції додають струмообмежувальні запобіжники (терморезистори), які запобігають перевантаженням. Окремі серії Li/SOCl_2 -елементів можуть витримувати підвищені струми розряду й нормально працювати при високих температурах завдяки

спеціальній конструкції циліндричного корпусу, який захищає від проникнення вологи, але забезпечує вихід газів.

Подібні обмеження стосуються й батарей на основі Li/SO₂. Вони також чутливі до високих температур, не підтримують високострумове розрядження й мають меншу робочу напругу (3,0 В). Варто зазначити, що ця серія батарей з'явилася на ринку раніше за інші літєві джерела живлення.

Зараз активно проводяться дослідження з метою заміни кобальту на доступніший, дешевший, менш токсичний і вибухобезпечний матеріал. Перспективним варіантом є використання літій-залізо-фосфату (LiFePO₄). У 2003 році в Массачусетському технологічному інституті розпочали експерименти зі зменшення розмірів частинок LiFePO₄ до 100 нанометрів (глибина інтеркаляції літію в кристалічну структуру LiFePO₄ становить 50 нм). Завдяки значному збільшенню активної поверхні та поліпшенню електропровідності, досягнутому за рахунок покриття наночастинок LiFePO₄ вуглецем, батареї з катодом із наноструктурованого LiFePO₄ перевершили традиційні батареї з катодом із кобальтату літію за струмами розрядки. До того ж кристалічна структура LiFePO₄ практично не змінювалася під час циклічної роботи, що дозволило збільшити кількість робочих циклів батареї до 5000. Ці батареї ефективно функціонували навіть за низьких температур.

Хоча нові матеріали електродів мають у кілька разів нижчу питому енергію, ніж чистий літій, батареї на їх основі є безпечнішими за умови дотримання заходів безпеки під час заряджання та розряджання. При цьому літій-іонні акумулятори на основі оксидів перевершують NiCd- і NiMH-акумулятори щонайменше вдвічі за зарядно-розрядними характеристиками, забезпечують високу ефективність при великих струмах (що важливо для мобільних телефонів і портативних комп'ютерів) і мають низький рівень саморозряду (2–5% на місяць у сучасних батареях).

Літєві акумулятори, як і всі інші, схильні до старіння, але в меншій мірі, ніж конкуруючі технології: навіть через 2 роки експлуатації вони зберігають понад 80% своєї ємності.

Однак технології Li-Ion все ще потребують дотримання заходів безпеки. Кожен акумуляторний блок повинен бути обладнаний електронною схемою управління, яка обмежує пікову напругу кожного елемента під час заряджання, а також запобігає зниженню напруги нижче допустимого рівня під час розряджання, що є критично важливим для тривалої роботи батарей. Крім цього, необхідно обмежувати максимальний струм заряджання та розряджання, а також контролювати температуру елемента. Такі заходи підвищують вартість літєвих акумуляторів, що є основною перепоною для їх масового використання. Крім того, сам літій і технологія виробництва цих батарей є дорогими через необхідність використання інертного середовища, очищення неводних розчинників тощо.

Таким чином, головний недолік літій-іонних акумуляторів — це їхня висока вартість. Проте ринок літєвих елементів і батарей малої ємності, де ціна не є

визначальним фактором, постійно розширюється. З'являються нові сфери їхнього застосування, що робить літій-іонні акумулятори одними з найперспективніших.

У 1991 році компанія **Sony Energetic** вперше розпочала комерційне виробництво літій-іонних акумуляторів і на сьогодні залишається одним із найбільших постачальників. За матеріалом негативного електрода літій-іонні акумулятори можна розділити на два основні типи: з електродом на основі коксу (технологія Sony) і на основі графіту.

Акумулятори з графітовим негативним електродом мають більш стабільну розрядну характеристику з різким падінням напруги наприкінці циклу розрядження порівняно з більш плавною характеристикою акумуляторів із коксовим електродом. Тому для досягнення максимальної ємності кінцева напруга розрядження акумуляторів із коксовим електродом зазвичай встановлюється нижчою, ніж для акумуляторів із графітовим електродом. Наприклад, літій-іонні акумулятори однієї компанії з номінальною напругою 3,6 В, як правило, мають коксовий електрод, а ті, що мають 3,7 В — графітовий. Таким чином, виробники спеціально вводять різницю в номінальній напрузі, щоб збалансувати характеристики.

Сьогодні більшість виробників віддають перевагу Li-Ion-акумуляторам із графітовим негативним електродом, оскільки вони забезпечують більший струм навантаження й менший нагрів під час заряджання та розряджання порівняно з акумуляторами на основі коксу.

Серед основних переваг сучасних літій-іонних акумуляторів у порівнянні з іншими технологіями можна виділити такі:

- Найвища питома ємність і щільність розрядного струму.
- Мінімальний саморозряд. Для деяких типів літій-іонних батарей при температурі 20 °С він не перевищує 3% на рік.
- Тривалий термін служби. До 10 років.
- Велика кількість циклів зарядки-розрядки. Забезпечується понад 1000 циклів.
- Висока збереженість енергії та постійна готовність до роботи.

Стандартні елементи Li-Ion забезпечують підвищену ємність при середніх струмах і високу напругу розряду у широкому діапазоні температур. У свою чергу, елементи зі збільшеною ємністю гарантують довший термін служби при низьких струмах. Останніми роками технологія виробництва літій-іонних джерел струму значно вдосконалилася. Виробники активно впроваджують сучасні матеріали для електродів та нові склади електролітів. Одночасно з цим здійснюються заходи для підвищення безпеки експлуатації як окремих акумуляторних елементів, так і керуючих електронних схем.

Головний недолік літій-іонних акумуляторів — їхня висока вартість. Сьогодні триває робота над заміною оксиду кобальту на дешевші матеріали, що дозволить знизити їхню вартість приблизно вдвічі протягом найближчих років. Крім того, вдосконалення безпеки технології також сприяє зниженню вартості.

Окрім високої ціни, літій-іонні акумулятори мають інші недоліки. Найкраще вони працюють при кімнатній температурі, а підвищена температура прискорює старіння елементів через зростання внутрішнього опору, що зменшує термін служби. Також вони чутливі до низьких температур.

Науковці з Американської національної лабораторії INEEL у штаті Айдахо повідомили про створення нової конструкції літійової батареї. Ця розробка суттєво розширює функціональні можливості традиційних пристроїв і долає їхні основні недоліки. Основна зміна в конструкції полягає у використанні суміші гелеподібного полімеру та керамічного порошку, які утворюють прозору мембрану, що виконує роль електроліту при контакті з двома електродами. Така конструкція має низку переваг порівняно з традиційними, де електролітом виступають рідини або гелі. У новій конструкції повністю виключено можливість витоків електроліту, оскільки він твердий, а також відсутнє осадження ізолюючого шару на поверхні електродів, що в традиційних батареях призводить до зниження тривалості їх роботи, а зрештою – до втрати працездатності.

Відсутність рідкого електроліту, який є потенційно пожежонебезпечним і може спричинити вибух під час зарядки батарей, значно підвищує безпеку використання. Головною перевагою твердого електроліту є можливість застосування батарей у ширшому діапазоні температур. Він не розплавиться при високих температурах і не замерзає при низьких, залишаючись працездатним навіть за $-73\text{ }^{\circ}\text{C}$.

Літій-іонні акумулятори також мають низку недоліків. Вони не переносять глибокого розряду, чутливі до температурного діапазону, бояться перезаряду, а при порушенні герметичності можуть бути вибухонебезпечними. З часом вони поступово втрачають ємність, тобто старіють навіть без навантаження. Проте всі ці недоліки компенсуються їхньою високою питомою енергоемністю.

Літій-полімерний акумулятор є вдосконаленою конструкцією літій-іонного акумулятора. Як електроліт у ньому використовується твердий полімерний матеріал із включеннями гелеподібного літій-провідного наповнювача. Завдяки відсутності рідкого електроліту такі акумулятори здатні працювати при низьких температурах до $-20\text{ }^{\circ}\text{C}$.

Удосконалення полімерних електролітів для цих батарей активно проводиться в напрямі введення спеціальних добавок, які перешкоджають утворенню дендритів, а також розробки нових полімерних нанокомпозитів із покращеними властивостями.

2.5.1 Перспективні системи накопичення енергії

Останнім часом у сфері технологій на основі літію спостерігається перехід до літій-полімерних акумуляторів (Lithium-Polymer battery). Принципових відмінностей між цими технологіями немає, однак за майже такої ж щільності енергії, як у літій-іонних акумуляторів, літій-полімерні батареї можуть

виготовлятися в різних гнучких геометричних формах, що особливо актуально для мініатюрних пристроїв. Нетипові для звичайних акумуляторів форми літій-полімерних батарей дозволяють заповнювати весь вільний простір усередині портативного пристрою і не потребують спеціального відсіку, як раніше. Таким чином, використання літій-полімерної батареї з такою ж питомою ємністю, як у традиційного циліндричного акумулятора, завдяки оптимальному вибору форми й заповненню всіх невикористаних об'ємів дозволяє зберігати на 20–30% більше енергії без змін у формі самого портативного пристрою.

Головна відмінність літій-полімерних (Li-Pol, Li-Polymer) акумуляторів від літій-іонних закладена в їхній назві та полягає у типі застосовуваного електроліту. Сухий твердий полімерний електроліт (або електроліт у вигляді полімерного гелю) схожий на пластикову плівку і не проводить електричний струм, але допускає обмін іонами. Це дозволяє спростити конструкцію елемента, оскільки полімерний електроліт не схильний до витоку, а отже, немає потреби забезпечувати герметичність. Полімерний електроліт фактично замінює традиційний пористий сепаратор, просочений електролітом. Така конструкція елементів є більш безпечною, спрощує процес виготовлення і дозволяє створювати тонкі акумулятори довільної форми. Однак, на жаль, сухі полімерні електроліти поки що мають недостатню електропровідність навіть за кімнатної температури. Їхній внутрішній опір занадто високий і не забезпечує струму, необхідного сучасним портативним пристроям.

Крім того, через недостатню розробленість технології виробництва Li-Pol-акумулятори досі є надто дорогими й недовговічними — гарантована кількість повних циклів зарядки-розрядки у них щонайменше вдвічі менша, ніж у Li-Ion. Утім, проміжні рішення із застосуванням рідкого гелевого електроліту вже є досить надійними і широко використовуються.

Редокс-акумулятори

Редокс-акумулятори (від англ. Redox — Reduction (відновлення) і Oxidation (окислення)) або проточні акумулятори складаються з резервуарів із електролітами (розчини солей або інших хімічних сполук з різними ступенями окислення одного чи кількох елементів), насосів для перекачування електроліту через контури анодів і катодів, а також електрохімічних ячеек із йонообмінними мембранами.

Принцип роботи таких акумуляторів полягає у тому, що на одному з електродів відбувається оборотна реакція окислення елемента з передачею електрона через зовнішній ланцюг і протона через мембрану на інший електрод, де протікає реакція відновлення. Основною перевагою цієї системи є можливість незалежного масштабування енергоємності (визначається об'ємом електролітів) і потужності (залежить від кількості й площі електрохімічних ячеек). Перезарядка може здійснюватися як у зворотному напрямку при подачі струму, так і заміною витраченого електроліту на свіжий (рис. 2.39).

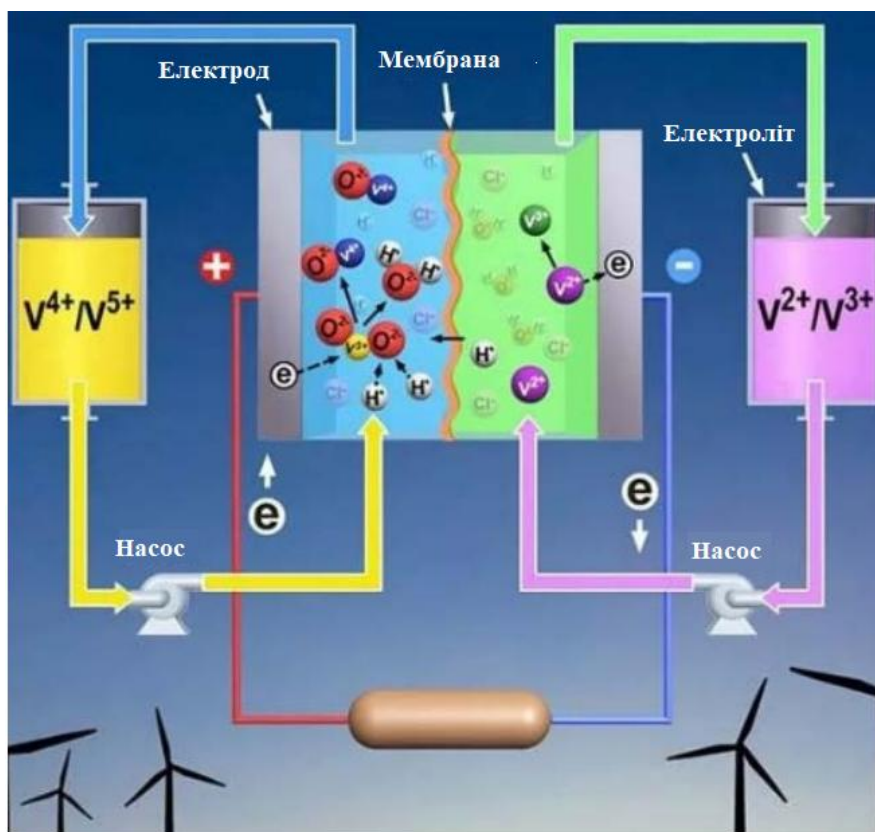


Рисунок 2.39 Принцип дії редокс-акумулятору

Такі системи мають потенціал для використання як потужні накопичувачі енергії для автономних робототехнічних систем та відновлюваних джерел енергії, перевершуючи за масштабованістю лише гідроакумуляційні станції. У Японії перші великі проточні акумулятори потужністю понад 100 кВт були введені в експлуатацію ще у 1996 році, а сучасні системи досягають десятків мегават. Наприклад, у Японії існує установка потужністю 60 МВт для регулювання частоти електричного струму.

Ванадієві редокс-акумулятори отримали найбільше поширення. У таких системах застосовуються вуглецеві неткані матеріали з великою площею поверхні для електродів, композитні біполярні пластини з вуглецево-полімерних матеріалів, а також протонобмінні мембрани типу Nafion (у випадку використання сірчанокислового електроліту). Хоча каталізатори на електродах не використовуються, деякі дослідження передбачають застосування наноструктурованих систем на основі рутенію та паладію для підвищення густини струму.

ККД системи становить 75–80 %. Основними недоліками є обмеження енергоємності через граничну концентрацію ванадієвих солей у сірчаній кислоті (до 32 Вт·год/кг, що порівняно зі свинцево-кислотними акумуляторами), а також

висока вартість систем (3000–4000 \$/кВт), що обмежує їхнє використання унікальними проектами.

Суперконденсатори

Суперконденсатори (або іоністори) – це перспективний тип систем накопичення енергії. Їх також називають двошаровими електрохімічними конденсаторами. У таких пристроях електродами слугує подвійний електричний шар, що утворюється на межі електрода й електроліту.

За своїми характеристиками суперконденсатори займають проміжне положення між звичайними конденсаторами та хімічними джерелами струму. Для досягнення максимальної ємності важливо забезпечити велику площу контакту електроліту з електропровідним матеріалом. Як матеріали для електродів широко використовують вуглецеві нанотрубки й наноструктурований вуглець із розвиненою поверхнею пор. Вони забезпечують:

- велику площу поверхні, що змочується електролітом;
- високу електронну провідність;
- хімічну й електрохімічну стабільність;
- відносно низьку вартість.

Ємність таких пристроїв на порядок вища, ніж у сучасних сухих і електролітичних конденсаторів.

Двошаровий суперконденсатор (ДСК) складається з двох пористих електродів із електропровідних матеріалів, розділених сепаратором, заповненим електролітом. Енергія накопичується завдяки розділенню зарядів на електродах і різниці потенціалів між ними.

Подвійний електричний шар на кожному електроді формує окремий конденсатор, а їхня ємність визначає загальний електричний заряд. Електроди з'єднані послідовно через електроліт, який забезпечує іонну провідність.

Окрім симетричних суперконденсаторів, існують асиметричні або гібридні суперконденсатори, які відрізняються використанням електродів різних типів. Наприклад, у таких пристроях позитивний електрод виготовляється з оксидів металів, а негативний – із активованого вуглецю.

Гібридні суперконденсатори поділяються на два типи залежно від типу електроліту: з водним і безводним. У них один із електродів (негативний) виконаний з активованого вугілля, а інший – «фарадеївський електрод», який може бути виготовлений з оксидів металів (наприклад, $\text{NiOOH}/\text{KOH}/\text{C}$, $\text{PbO}_2/\text{H}_2\text{SO}_4/\text{C}$) або матеріалів, здатних до оборотних реакцій у безводних електролітах. Робочий діапазон потенціалів таких конденсаторів становить 0,6–2,1 В.

Гібридні суперконденсатори мають вищу питому енергоємність, ніж двошарові суперконденсатори (ДСК), однак вона все ще поступається акумуляторам.

Суперконденсатори відрізняються від акумуляторів:

- Меншою енергоємністю: у комерційних зразках вона становить 6–10 Вт·год/кг (або 5–8 Вт·год/л), а в новітніх розробках на основі графену – до 83 Вт·год/кг (або 60 Вт·год/л) у лабораторних умовах.
- Вищою питомою потужністю: 2–10 кВт/кг.
- Тривалішим ресурсом: до 100 000 циклів і більше.
- Відсутністю обмежень на глибину розряду.

Завдяки високій питомій потужності суперконденсатори застосовуються там, де потрібне короткочасне різке збільшення потужності, наприклад:

- у транспортних засобах із водневими паливними елементами;
- у акумуляторних системах, щоб уникнути необхідності надмірного збільшення потужності генератора чи акумулятора. Наприклад, суперконденсатор масою 30 кг може забезпечити потужність 100 кВт для електродвигуна протягом 10 секунд.

2.5.2 Саморазряд акумуляторних систем

Будь-яка акумуляторна система характеризується здатністю зберігати накопичену енергію протягом певного часу. Це властивість визначається так званим саморазрядом, що є наслідком фізико-хімічних процесів у матеріалах і конструкції акумулятора. Важливо не плутати саморазряд із терміном служби акумулятора, який визначається кількістю циклів заряд-розряд або тривалістю роботи до моменту, коли ємність падає до неприйняттого рівня (зазвичай 80% від початкової ємності).

Системи з найнижчим рівнем саморазряду зазвичай мають енергоносії, ізольований у герметичній ємності або резервуарі. До таких належать гідроакуюлюючі станції, традиційні системи на основі вуглеводневого палива, а також регенеративні водневі системи, де водень, отриманий електролізом, зберігається у балонах високого тиску або у вигляді металевих/інтерметалічних гідридів для подальшої генерації електроенергії за допомогою паливних елементів.

Перші електрохімічні системи, **свинцево-кислотні акумулятори**, демонструють саморазряд, зумовлений побічними реакціями, зокрема розчиненням свинцю у сірчаній кислоті з виділенням водню. Ця реакція відбувається під час зберігання зарядженого акумулятора.

Швидкість саморазряду залежить від температури, об'єму та концентрації електроліту, а також від чистоти компонентів. Побічні реакції, як-от спонтанне відновлення діоксиду свинцю на позитивному електроді, також сприяють саморазряду. З часом саморазряд збільшується через утворення дендритних містків зі свинцю або великих кристалів сульфату свинцю на негативному електроді.

Для нових акумуляторів втрати ємності через саморазряд зазвичай не перевищують 2–3% на місяць. Однак під час експлуатації ці втрати швидко

зростають, і для герметичних необслуговуваних акумуляторів при температурі 20°C втрата ємності становить близько 5% на місяць.

Саморазряд **герметичних нікель-кадмієвих (Ni-Cd)** акумуляторів значною мірою визначається термодинамічною нестабільністю окисно-нікелевого позитивного електрода. На початкових етапах експлуатації вплив мікровитоків між електродами мінімальний, але з часом він зростає. Після відключення акумулятора від зарядного пристрою потенціал на поверхні зарядженого електрода поступово знижується, а заряди поверхневих і глибинних шарів вирівнюються, що знижує швидкість саморазряду. Через варіації у виробництві рівень саморазряду та стабілізації залишкової ємності можуть суттєво різнитися навіть між акумуляторами однієї серії. Процес саморазряду також призводить до зниження напруги на 30–50 мВ.

Зберігання акумуляторів при низьких температурах знижує втрати: при 0°C саморазряд удвічі менший, ніж при 20°C. В середньому саморазряд на рівні 15–20% на місяць при температурі 20°C вважається досить високим показником.

Варто зазначити, що початковий гідроксид нікелю у Ni-Cd акумуляторах може існувати в різних формах, що певною мірою впливає на рівень саморазряду.

На відміну від Ni-Cd акумуляторів, **Ni-MH акумулятори** не мають «ефекту пам'яті», проте їм також притаманний саморазряд. При зберіганні протягом місяця за кімнатної температури втрата ємності може досягати 20–30%, що є значним показником. Однак із часом ці втрати зменшуються до 3–7% на місяць. У результаті через рік зберігання традиційний Ni-MH акумулятор зазвичай зберігає менше ніж 10% початкової ємності.

Акумулятори з низьким саморазрядом (англ. low self-discharge nickel-metal hydride battery **LSD Ni-MH**). У листопаді 2005 року компанія Sanyo вперше представила нікель-металогідридні акумулятори з низьким саморазрядом під брендом Eneloop. Пізніше багато виробників почали випускати подібні акумулятори.

LSD Ni-MH акумулятори можуть зберігати заряд протягом більш ніж трьох тижнів після заряджання. Звичайні Ni-MH акумулятори втрачають до 10% ємності заряду протягом перших 24 годин після зарядки, а потім швидкість саморазряду стабілізується на рівні до 0,5% ємності на добу. Для LSD Ni-MH акумуляторів цей показник зазвичай становить 0,04–0,1% ємності на добу.

Виробники стверджують, що вдосконалення електроліту та електродів дозволило значно знизити швидкість саморазряду LSD Ni-MH акумуляторів у порівнянні зі звичайними моделями.

Літій-іонні (Li-ion) акумулятори, які мають рекордну питому щільність зберігання енергії, також характеризуються відносно низьким рівнем саморазряду. У перший місяць саморазряд становить 4–6% при повній зарядці, а при меншому рівні зарядки ці втрати ще нижчі. У наступні місяці швидкість саморазряду суттєво

зменшується, і за рік акумулятори втрачають лише 10–20% накопиченої ємності. Ці втрати значно менші порівняно з нікель-кадмієвими акумуляторами як при 20°C, так і при 40°C. Ресурс роботи Li-іон акумуляторів становить 500–1000 циклів заряд-розряд.

Максимальна швидкість саморазряду спостерігається у перший місяць, і в деяких менш якісних акумуляторах вона може сягати 10%. Однак надалі ця швидкість суттєво знижується, і сучасні літій-іонні акумулятори можуть зберігати 75–80% своєї початкової ємності через рік зберігання, якщо вони були заряджені на 100%. Саморазряд Li-іон акумуляторів залежить від температури: зі зниженням температури саморазряд зменшується, але одночасно сповільнюється токогенеруюча реакція, що негативно впливає на потужність і працездатність акумулятора при температурах нижче 4°C. Для подолання цих обмежень нині активно проводяться дослідження щодо додавання спеціальних речовин до електроліту. Зокрема, вивчається вплив триметилборату в карбонатному електроліті, який може значно покращити характеристики літій-іонних акумуляторів.

Первинні джерела струму демонструють найкращу здатність до збереження накопиченої енергії. Зокрема, усі літійові елементи характеризуються низьким рівнем саморазряду, тобто мають високу збережуваність. Найкращі показники серед первинних джерел струму мають елементи системи літій-йод, у яких втрати ємності через саморазряд не перевищують 10% за 10 років зберігання.

Для елементів інших систем втрати ємності становлять 0,5–3% на рік. Наприклад, літій-марганцеві первинні джерела струму можуть зберігати енергію протягом кількох років. У звичайних побутових «лужних» батарейках (алкалінових) рівень саморазряду становить близько 3,5% на рік.

Наразі ведуться дослідження зі створення гнучких плівкових перезаряджуваних батарей цієї системи, однак поки що практично значущих результатів не досягнуто.

Саморазряд усіх джерел струму зростає зі збільшенням температури. Як правило, при підвищенні температури на 10°C рівень саморазряду подвоюється. Велика кількість циклів зарядки-розрядки також сприяє збільшенню саморазряду. Наприклад, у металогідридних систем саморазряд починає помітно зростати після 300–400 циклів, тоді як у нікель-кадмієвих акумуляторів це явище спостерігається лише після 1000 циклів.

Для літій-іонних акумуляторів рівень саморазряду не залежить від кількості циклів, але більш суттєвий вплив мають температура та стан повного заряду. Зі зростанням температури та підвищенням ступеня зарядженості саморазряд літій-іонних акумуляторів також збільшується.

На рівень саморазряду всіх хімічних джерел струму значно впливає якість їх виробництва. Наявність домішок в електродах та електроліті підвищує корозію струмовідводів, що призводить до збільшення саморазряду.

Суперконденсатори мають переваги над електрохімічними системами з точки зору потужності, що віддається за одиницю часу, а також у разі комбінованого використання з іншими системами. Якщо тривалі коливання генерації можна компенсувати за допомогою електрохімічних акумуляторів, то короткочасні, але потужні флуктуації доцільніше нівелювати саме суперконденсаторами. Саме тому для завдань, у яких зазвичай застосовують суперконденсатори, швидкість їхнього саморозряду не є критично важливим параметром. Для малопотужних суперконденсаторів на органічному електроліті (з максимальною робочою напругою до 2,5 В) струм саморозряду зазвичай складає 1 мкА. У середньому вважається, що якісний суперконденсатор на органічному електроліті втрачає половину накопиченої енергії за місяць. Для суперконденсаторів на водних розчинах кислот або лугів (з максимальною напругою до 1,2 В) струм саморозряду становить приблизно 10 мкА.

Явища саморозряду найсильніше проявляються у суперконденсаторах із кислотними електролітами, які є найдешевшими й найпростішими у виготовленні. У загальному випадку причинами саморозряду суперконденсаторів можуть бути струми витоку, електрохімічне розкладання активних матеріалів електродів і компонентів електроліту, корозія струмознімальних матеріалів, а також небажані окисно-відновлювальні процеси на поверхні електродів.

Дослідження механізмів саморозряду суперконденсаторів та методів його зменшення проводяться в багатьох країнах світу. Перспективні результати як у підвищенні ємності, так і в зменшенні струмів саморозряду демонструють наукові роботи, присвячені суперконденсаторам із полімерними електролітами. Зокрема, вивчаються електроліти з електронною провідністю (наприклад, поліанілін) і протонною провідністю (перфторсульфополімер, відомий під торговою маркою Nafion).

2.5.3 Водневі технології та зворотні паливні елементи

Особливе місце серед перспективних акумуляторних систем займають системи, пов'язані з водневою енергетикою. Це пояснюється, насамперед, високою питомою енергоємністю водню. Використання так званого водневого циклу, який включає електролізер води, систему зберігання водню та паливний елемент, дозволяє накопичувати енергію як на короткий термін (години, дні), так і на тривалий (місяці й більше).

Основними перевагами накопичувачів енергії на основі водневих технологій є: відсутність саморозряду, самобалансування, велика кількість циклів заряд-розряд, здатність зберігати дуже великі обсяги енергії у вигляді водню.

Зменшення вартості, маси та габаритів таких систем на 30-40 % може бути досягнуто шляхом об'єднання електролізера води та паливного елемента в біфункціональний електрохімічний пристрій — зворотний (регенеративний) паливний елемент (ЗПЕ). Такий пристрій працює поперемінно: у режимі

електролізера (виробляє водень) або в режимі паливного елемента (генерує електричну енергію та, за потреби, тепло).

Компактними акумуляторами енергії можуть бути системи, побудовані на основі протонно-обмінної твердополімерної мембрани (ТПМ) (рис. 2.40).

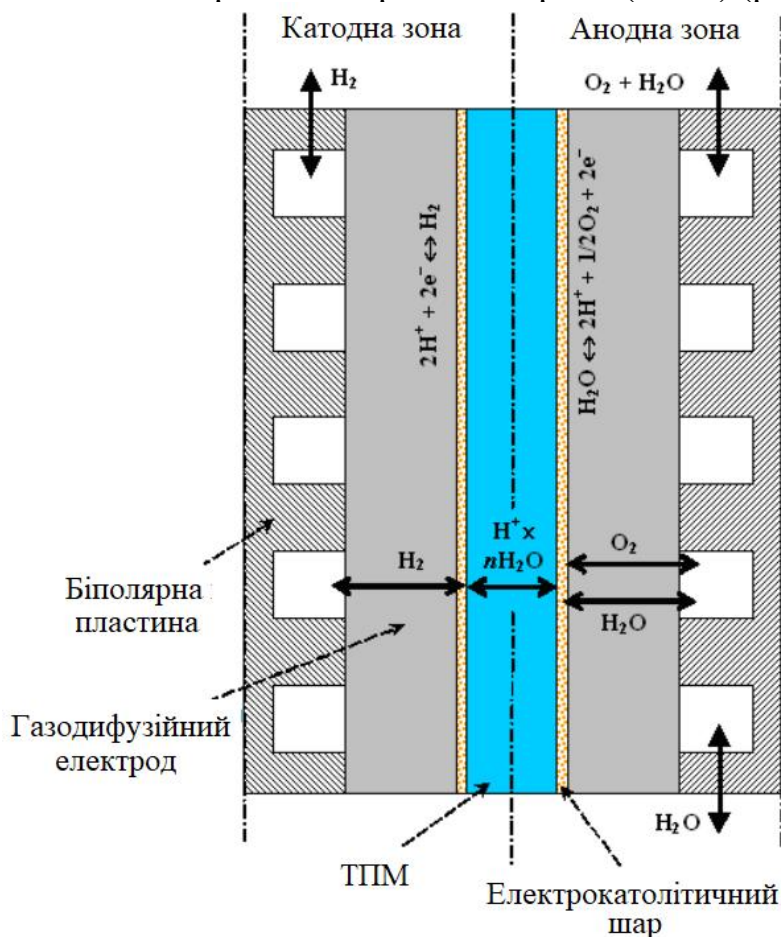


Рисунок 2.40 Схема зворотного паливного елемента з твердополімерним електролітом.

Електрохімічні системи з ТПМ характеризуються: коротким часом відгуку, високою питомою потужністю, високим електричним ККД (близько 50 % для паливних елементів і понад 85 % для електролізерів), мінімальним часом запуску, екологічною чистотою та безпекою. Ці характеристики дозволяють вважати технологію на основі ТПМ оптимальною для створення низькотемпературних ЗПЕ.

Очевидно, що в короткостроковій перспективі акумуляторні батареї демонструють кращу ефективність, оскільки їхній загальний ККД становить близько 80 % і більше. Однак у довгостроковій перспективі (особливо при сезонних коливаннях) водневі системи виявляються значно продуктивнішими. Це пояснюється тим, що ефективність акумуляторів може знижуватися до 0 % через саморозряд, наприклад, якщо батарея залишається без заряджання кілька тижнів.

Зворотні паливні елементи (ЗПЕ) мають широкий спектр можливих застосувань, зокрема на автомобільних заправних станціях, у рухових установках для супутників та БПЛА аеростатного типу, а також для зберігання електроенергії у складі різноманітних енергетичних установок, що працюють разом із відновлюваними джерелами енергії.

Для забезпечення роботи ЗПЕ можуть використовуватися як хімічно зворотні кисневі й водневі електроди, так і електроди, що не змінюють своєї окислювальної чи відновної функції при перемиканні режимів роботи ЗПЕ (рис. 2.41).

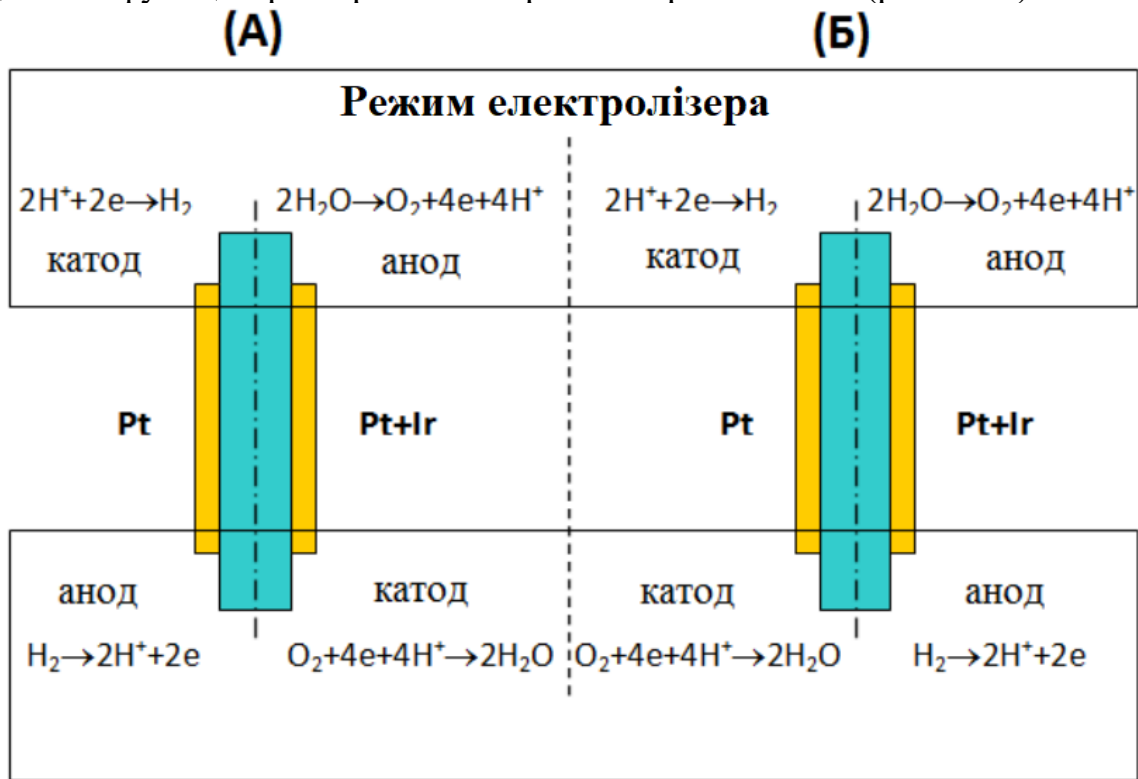


Рисунок 2.41. Схема мембранно-електродного блоку ЗПЕ для:
А – електрохімічно оборотних електродів, **Б** – електродів, які не змінюють своєї окисно-відновлювальної функції при перемиканні режиму роботи комірки.

У першому випадку (схема А) при зміні режиму роботи ЗПЕ гази, які виробляються чи споживаються в результаті електрохімічної реакції на електроді, залишаються незмінними. Наприклад, електрод, що виділяє водень у режимі електролізера, також використовується для його окислення в режимі паливного елемента. Основною перевагою цієї схеми є відсутність необхідності продувки електродних камер при зміні режиму роботи системи. Однак така схема має недолік, пов'язаний із корозією та можливим затопленням кисневого електрода.

Друга схема (схема Б) передбачає, що процеси окислення завжди відбуваються на одному електроді, а відновлення — на іншому, незалежно від режиму роботи елемента. Ця схема потребує певних витрат на продувку електродних камер при зміні режиму, проте дозволяє використовувати традиційні

матеріали газодифузійних електродів (ГДЕ), характерні для систем з твердополімерним електролітом (ТПЕ): пористий титан для анода та вуглецеву тканину або папір для катода. Електрокаталізатори та електрокаталітичні шари відіграють вирішальну роль у визначенні ефективності роботи електрохімічних систем із ТПЕ.

Електрокаталізатори для зворотних паливних елементів (ЗПЕ) повинні мати такі властивості, як висока каталізаторна активність, тривалий термін служби, достатня електронна провідність і, що особливо важливо, корозійна стійкість до кислих середовищ та високих анодних потенціалів. Однак найкращими електрокаталізаторами для цих цілей є дорогоцінні метали та їх оксиди, наприклад, Pt, Ir, Ru, IrO₂ і RuO₂, що призводить до підвищення вартості ЗПЕ.

Робочі параметри ЗПЕ з твердополімерним електролітом (ТПЕ) загалом відповідають параметрам окремих пристроїв, таких як електролізер і паливний елемент. Наприклад, ЗПЕ, розроблені LLNL, мають активну площу 46 см², працюють за температури 40–82°C і атмосферного тиску в обох режимах. У режимі паливного елемента газу зволожуються за температури 75°C. Максимальний робочий тиск біфункціонального елемента LLNL становить 0,55 МПа для водню та 1,1 МПа для кисню.

Як окислювач у ЗПЕ може використовуватися як чистий кисень, отриманий у режимі електролізу, так і атмосферний кисень. У першому випадку вольтамперні характеристики ЗПЕ, що працює в режимі паливного елемента, значно вищі, однак для цього необхідна система зберігання кисню.

Робота ЗПЕ в режимі електролізу під тиском, щонайменше 3,0 МПа, є доцільною, оскільки це дозволяє безпосередньо заправляти різні системи зберігання водню. Порівняння ЗПЕ з різними типами акумуляторів демонструє певні переваги ЗПЕ.

Як приклад **гібридної конструкції**, яка поєднує класичний акумулятор та зворотний паливний елемент, можна навести **нікель-водневий акумулятор**. Позитивний електрод такого акумулятора виготовляється зі спеченого пористого нікелевого диска, який містить гідроксид нікелю. У негативному водневому електроді використовується платиновий каталізатор, пов'язаний з фторопластом, а також сепаратор із цирконієвих волокон. Електролітом є розчин лугу. Під час заряджання на водневому електроді генерується водень, який накопичується під тиском у герметичному корпусі. Тиск у ячійці лінійно залежить від ступеня заряду. Твердофазний електрод функціонує аналогічно до електродів нікель-кадмієвого акумулятора. Під час розряджання водень споживається на водневому електроді, а тиск у ячійці повертається до початкового значення, відповідного стану повного розряду.

Нікель-водневий акумулятор (NiH₂), як зворотне хімічне джерело струму, що складається з нікелевого та водневого електродів, відрізняється від нікель-металогідридного акумулятора використанням газоподібного водню, який

зберігається у стиснутому стані в ячійці під тиском понад 8,0 МПа. Ячейки NiH_2 , що використовують 26%-ний розчин гідроксиду калію (KOH) як електроліт, забезпечують термін експлуатації понад 15 років. Щільність енергії таких акумуляторів становить 75 Вт·год/кг, а напруга на контактах зарядженої ячійки досягає 1,55 В.

Однак нікель-водневі акумулятори мають недолік — відносно високий саморазряд, пропорційний тиску водню в ячійці. У порівнянні з іншими акумуляторами, NiH_2 мають чудові електричні характеристики, які роблять їх привабливими для зберігання енергії на космічних апаратах. Наприклад, Міжнародна космічна станція (МКС), апарат Messenger, Mars Odyssey і Mars Global Surveyor були обладнані нікель-водневими акумуляторами. Телескоп «Хаббл» використовував NiH_2 батареї протягом 19 років, доки їх не замінили.

2.6 Питання для самоперевірки:

1. Що таке привід у робототехнічній системі і яка його основна функція?
2. Які існують основні типи приводів у роботах?
3. За якими критеріями класифікують приводи роботів?
4. У чому полягають переваги і недоліки кожного типу приводу?
5. Як працює пневматичний привід і де його доцільно використовувати?
6. Які основні елементи входять до складу пневматичної системи?
7. Яка принципова відмінність гідравлічного приводу від пневматичного?
8. У яких умовах гідравлічні приводи мають перевагу над іншими типами?
9. Які типи електродвигунів використовуються в промислових роботах?
10. Які переваги має електропривід у порівнянні з іншими приводами?
11. Що таке штучні м'язи та які матеріали використовуються для їх створення?
12. У яких галузях найбільше застосовуються штучні м'язи?
13. Наведіть приклади спеціалізованих виконавчих механізмів у роботах.
14. Які вимоги ставляться до виконавчих механізмів у промислових умовах?
15. Які основні типи маніпуляторів використовуються в промисловості?
16. Як визначається ступінь свободи маніпулятора?
17. Які типи захоплювачів існують і які принципи їх дії?
18. Як вибрати захоплювач для конкретного об'єкта?
19. Які основні гравці на ринку роботів-маніпуляторів?
20. Які тенденції розвитку простежуються на ринку маніпуляторів?
21. Які типи навігаційних систем використовуються в роботах?
22. У чому полягає роль навігаційної системи у роботизованій платформі?
23. З яких компонентів складається інерціальна навігаційна система (INS)?
24. Які переваги та недоліки має INS?
25. Як працюють системи GNSS (GPS, Galileo тощо)?
26. У чому відмінність GPS, GLONASS, Galileo та BeiDou?
27. Які етапи включає робота GPS-приймача для визначення місцеположення?

28. Яка точність GPS при стандартних умовах?
29. У чому полягає принцип роботи DGPS?
30. Як DGPS покращує точність позиціонування?
31. Як працює технологія RTK?
32. У яких галузях RTK навігація найбільш ефективна?
33. Які алгоритми використовуються для автономної навігації мобільних роботів?
34. Які датчики зазвичай застосовуються в автономній навігації?
35. Як працюють системи технічного зору?
36. У чому переваги 2D та 3D візуальних систем?
37. Які функції виконують лазерні та інфрачервоні датчики в робототехніці?
38. У чому різниця між LIDAR та інфрачервоним сенсором?
39. Як працює радар у робототехнічній системі?
40. Де застосування радарів дає найбільшу перевагу?
41. Який принцип роботи ультразвукових сенсорів?
42. Які обмеження мають ультразвукові датчики?
43. Які типи мобільних платформ існують?
44. Як вибір платформи впливає на функціональність робота?
45. У чому особливості конструкції наземних роботів?
46. Які типи колісних і гусеничних платформ використовуються?
47. Які типи повітряних роботів існують?
48. У чому переваги мультикоптерів над іншими літальними платформами?
49. Які типи конструкцій планерів використовуються в БПЛА?
50. Як форма крила впливає на аеродинаміку?
51. Які двигуни використовуються у БПЛА?
52. Як обирають силову установку для конкретного типу дрону?
53. Які існують популярні польотні контролери і з яким ПЗ вони працюють?
54. Як працює зв'язок між контролером та сенсорами БПЛА?
55. Які типи акумуляторів використовуються у БПЛА?
56. У чому різниця між Li-Po та Li-ion акумуляторами?
57. Які технології накопичення енергії вважаються перспективними?
58. Які характеристики мають суперконденсатори?
59. Що таке саморазряд і як він впливає на ефективність акумулятора?
60. Які фактори впливають на швидкість саморазряду?
61. Як працюють водневі паливні елементи?
62. Які переваги та виклики мають водневі системи в мобільній робототехніці?

2.7 Література до розділу 2

- 2.1 **Датчики та сенсори робототехнічних систем** : навч. посіб. для підгот. бакалаврів за спец. 174 "Автоматизація, комп'ютерно-інтегровані технології та робототехніка" / Трунов О. М. ; Чорномор. нац. ун-т ім. Петра Могили. - Миколаїв : ЧНУ ім. Петра Могили, 2023.. - 55 с.

- 2.2 **Системи телебачення та технічного зору** : навч. посіб. / Г. Д. Дорощенко, В. П. Кожем'яко, С. В. Павлов ; Вінниц. нац. техн. ун-т. - Вінниця : ВНТУ, 2015.. - 208 с.
- 2.3 **Основи побудови сучасних мобільних систем технічного зору** : навч. посіб. : [у 3 ч.] / Нац. аерокосм. ун-т ім. М. Є. Жуковського "Харків. авіац. ін-т". - Харків : ХАІ, 2018 - 2019.
- 2.4 **Основи теорії пневмоавтоматики та електропневмоавтоматики** : метод. посіб. до викон. лаб. занять для студ., що навчаються за фахом "Гідравлічні машини, гідропневмоавтоматика та гідропривід" денної форми навч. / С. В. Носко [и др.] ; ред. В. М. Матієга ; Національний технічний ун-т України "Київський політехнічний ін-т", Національний технічний ун-т "Харківський політехнічний ін-т". - Чернівці : Зелена Буковина, 2004.. - 96 с.
- 2.5 **Гідропривід сільськогосподарської техніки** : підручник для підгот. бакалаврів з напрямку "Механізація та електрифікація сільського господарства" (спец. "Механізація сільського господарства") та як навч. посібник для підгот. бакалаврів з напрямку "Інженерна механіка" (спец. "Машини та обладнання с.-г. виробництва") в аграрних вищих навч. закл. II-IV рівнів акредитації / О. М. Погорілець [и др.] ; ред. О. М. Погорілець. - К. : Вища освіта, 2004.. - 368 с.
- 2.6 Гетерогенні комп'ютерні мережі критичного застосування на основі роїв та зграй БПЛА : монографія / Ірина Журавська ; Чорномор. нац. ун-т ім. Петра Могили. - Миколаїв : ЧНУ ім. Петра Могили, 2019. – 190 с.
- 2.7 **Свинцеві акумулятори: улаштування, виробництво, розрахунки** : навч. посіб. / Нефедов В. Г., Поліщук Ю. В.; – Київ : ІСДО, 2013. – 220 с.
- 2.8 **Системи накопичення електричної енергії** : навч. посіб. / Сінчук І. О., Бойко С. М., за ред. О. М. Сінчука. – Київ : Ліра-К, 2020. – 272 с.
- 2.9 Siciliano, B., & Khatib, O. (Eds.). (2016). **Springer Handbook of Robotics** (2nd ed.). Springer. – 2226 p. <https://doi.org/10.1007/978-3-319-32552-1>
- 2.10 Bräunl, T. (2022). **Embedded Robotics: From Mobile Robots to Autonomous Vehicles with Raspberry Pi and Arduino** (4th ed.). Springer. <https://doi.org/10.1007/978-981-16-0804-9>
- 2.11 Kaplan, E. D., & Hegarty, C. J. (Eds.). (2017). **Understanding GPS/GNSS: Principles and Applications** (3rd ed.). Artech House. <https://doi.org/10.1049/SBRA022E>
- 2.12 Zekavat, S. A., & Buehrer, R. M. (Eds.). (2019). **Handbook of Position Location: Theory, Practice and Advances** (2nd ed.). Wiley-IEEE Press. <https://doi.org/10.1002/9781119434610>

РОЗДІЛ 3 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

3.1. Операційні системи та програмні середовища робототехнічних систем

3.1.1. ROS (Robot Operating System) — Операційна система для роботів

ROS — це фреймворк для програмування роботів, який забезпечує функціонал для розподіленої роботи. Фреймворк (від англ. framework, буквально — "каркас" або "структура") — це набір інструментів, бібліотек, правил і концепцій, який допомагає розробникам створювати програмне забезпечення певного типу швидше й ефективніше. Фреймворки можна порівняти з каркасом будинку: будівельники/дизайнери додають свої матеріали, конструктиви, тощо (свій код), але основа структури вже готова.

Спочатку ROS був розроблений у 2007 році під назвою **switchyard** у Лабораторії штучного інтелекту Стенфордського університету для проєкту **STAIR**. У 2008 році розвиток ROS продовжився в **Willow Garage**, каліфорнійському науково-дослідному стартапі/інституті/інкубаторі робототехніки, у співпраці з більш ніж двадцятьма іншими дослідницькими інститутами.

Що таке ROS? ROS — це мета-операційна система (з відкритим кодом) для роботів. Слово мета- в цьому контексті є визначається як узагальнення, абстракцію або проміжність.

ROS забезпечує стандартні служби операційної системи, зокрема:

- апаратну абстракцію;
- низькорівневий контроль пристроїв;
- реалізацію часто використовуваних функцій;
- передачу повідомлень між процесами;
- управління пакетами.

Крім того, ROS надає інструменти та бібліотеки для створення, побудови, написання й виконання коду на кількох комп'ютерах одночасно.

Дві основні складові ROS:

- **сторона операційної системи ROS** — базові функції й сервіси для роботи з роботами.
- **ros-pkg** — набір підтримуваних користувачами пакетів, організованих у стеки, які реалізують різні функції робототехніки, такі як: SLAM (локалізація та побудова карти), планування, сприйняття, моделювання тощо.

Система ROS охоплює широкий функціонал: драйвери, алгоритми, графічні інтерфейси моделювання та парадигми управління. Вона стала корисним

інструментом для багатьох дослідників і науковців, які поділилися своїм кодом із спільнотою ROS. ROS є системою з відкритим вихідним кодом, яка спочатку використовувала додатки з проекту Player Project та інших open source проектів. Поняття відкритого коду передбачає можливість використовувати та модифікувати програмне забезпечення під свої потреби. Головне завдання ROS — забезпечити можливість повторного використання коду в робототехнічних дослідженнях і розробках.

Основні компоненти ROS:

- **Інфраструктура:** ROS забезпечує інфраструктуру комунікації за схемою публікації-підписки (publish–subscribe messaging), що дозволяє швидко та легко створювати розподілені обчислювальні системи.
- **Інструменти:** ROS пропонує широкий набір інструментів для налаштування, запуску, аналізу, налагодження, візуалізації, ведення журналів, тестування та завершення роботи розподіленої обчислювальної системи.
- **Можливості:** ROS містить великий набір бібліотек для реалізації функцій роботів із акцентом на мобільність, маніпуляцію предметами та сприйняття довкілля.

Особливістю ROS є її функціонування більше як операційної системи, ніж окремого застосунку. В архітектурі моделювання та управління застосовується клієнт-серверна архітектура.

В ROS використовуються вузли (nodes), які з'єднуються з основним процесом (roscore) і виступають як клієнти та сервер. ROS є надбудовою над операційною системою, що спрощує створення систем управління для роботів.

ROS об'єднує низку популярних бібліотек, таких як:

- **OpenCV** – бібліотека для комп'ютерного зору та обробки зображень з відкритим вихідним кодом, яка підтримує мови C/C++, Python, Java та інші.
- **PCL (Point Cloud Library)** – бібліотека для обробки 2D/3D зображень та хмар точок з відкритим вихідним кодом.
- **OGRE (Object-Oriented Graphics Rendering Engine)** – об'єктно-орієнтований графічний рушій на основі C з відкритим кодом, призначений для спрощення візуалізації та моделювання поведінки роботів.
- **OROCOS (Open Robot Control Software)** – бібліотека для управління роботами, зокрема для розрахунків кінематики.
- **CARMEN (Carnegie Mellon Robot Navigation Toolkit)** – це бібліотека для управління мобільними роботами, яка дозволяє виконувати базові операції, такі як сенсорне управління, обхід перешкод, прокладання маршрутів і створення карт.

Окрім бібліотек, ROS також містить драйвери для різних маніпуляторів, сенсорів (зокрема MS Kinect) і датчиків. Однією з головних переваг системи ROS є клієнт-серверна архітектура, яка забезпечує механізм передачі повідомлень між об'єктами та можливість створення розподілених систем.

Структуру ROS можна реалізувати на будь-якій сучасній мові програмування, наприклад Python, C++, Lisp, а також експериментально – на Java і Lua. ROS не є системою реального часу, хоча її можна інтегрувати з кодом реального часу.

ROS має три концептуальні рівні а саме: файловий, обчислювальний граф і рівень спільноти.

3.1.2. Рівень файлової системи ROS

На рівні файлової системи (рис. 3.1.1) основним блоком організації програмного забезпечення в ROS є пакет.

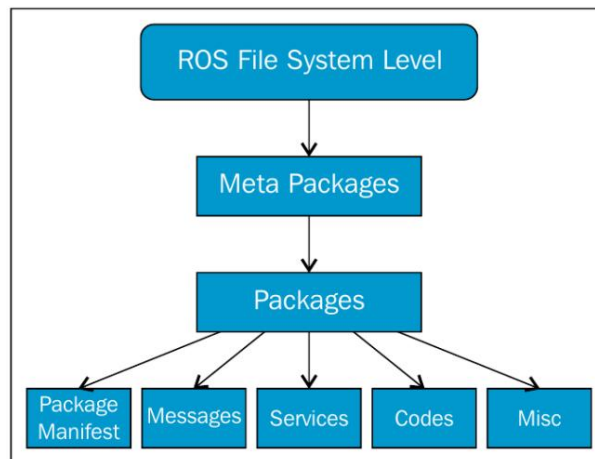


Рис. 3.1.1. Рівні файлової системи ROS

Packages. Пакет може містити вихідний код і виконуваний файли вузлів, бібліотеки, описи повідомлень і сервісів, бази даних, файли конфігурації та інші ресурси, які логічно згруповані разом. Кожен пакет обов'язково включає файл маніфесту, що надає метадані про пакет, зокрема інформацію про ліцензію, залежності та параметри компіляції. Така структура спрямована на забезпечення можливості повторного використання ресурсів. Типова структура пакету ROS представлена на рис. 3.1.2.

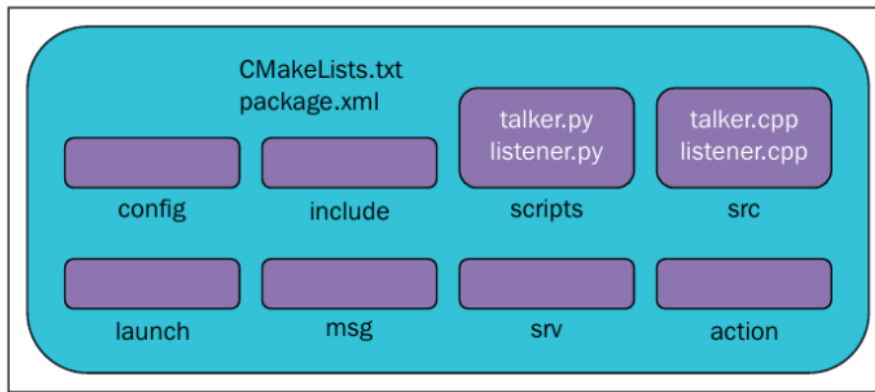


Рис. 3.1.2. Типова структура пакету ROS

Структура пакета має такий вигляд:

- **config:** У цій папці зберігаються всі файли конфігурації, які використовуються в цьому ROS-пакеті. Папку створює користувач (зазвичай її називають *config*) для зберігання конфігураційних файлів.
- **include/package_name:** Ця папка містить заголовки та бібліотеки, які використовуються всередині пакета.
- **package.xml:** Файл маніфесту (*package.xml*) містить метадані про пакет, зокрема його назву, версію, опис, інформацію про ліцензії, залежності та експортовані пакети, які можуть бути використані іншими пакетами.
- **CMakeLists.txt:** Це файл збірки CMake для цього пакета.

Приклади команд для створення, зміни та роботи з ROS-пакетами:

- **catkin_create_pkg:** команда для створення нового пакета.
- **rospack:** команда для отримання інформації про пакет у файловій системі.
- **catkin_make:** команда для збірки пакетів у робочій області.
- **roscdep:** команда для встановлення системи відповідно до залежностей цього пакета.

Сервіси (Services): визначають запити та структури даних.

Мета-пакет (Meta Packages): це група пов'язаних пакетів. Мета-пакет не містить папок і файлів, які характерні для звичайного пакета. Вони просто об'єднують кілька пакетів у єдиний логічний блок.

У файлі *package.xml* мета-пакета присутній тег експорту:

```
<export>
  <metapackage/>
</export>
```

Крім того, у мета-пакетах немає залежностей *<buildtool_depend>* для catkin; є лише залежності *<run_depend>*, які вказують на пакети, згруповані в мета-пакет. На рис. 3.1.3 наведено приклад навігаційного стека ROS, який є гарним прикладом мета-пакета.

```

<package>
  <name>navigation</name>
  <version>1.12.2</version>
  .....
  <buildtool_depend>catkin</buildtool_depend>
  .....
  <run_depend>amcl</run_depend>
  <run_depend>carrot_planner</run_depend>
  .....
  <export>
    <metapackage/>
  </export>
</package>

```

Рис. 3.1.3. Структура мета-пакету ROS

Додаткові папки пакета:

- **scripts:** у цій папці зберігаються виконувані сценарії Python.
- **src:** тут зберігаються вихідні коди на C++.
- **launch:** ця папка містить файли запуску для запуску одного або кількох вузлів ROS.
- **msg:** тут зберігаються визначення користувацьких повідомлень.
- **srv:** папка містить визначення сервісів.
- **action:** у цій папці розташовані визначення дій.

3.1.3. Обчислювальний граф ROS

Рівень обчислювального графа в ROS — це концепція представлення даних у різних формах обробки. Під час роботи ROS створюється «граф» — однорангова мережа (peer-to-peer network), яка складається з процесів, що взаємодіють один з одним через інфраструктуру ROS (рис. 3.1.4).

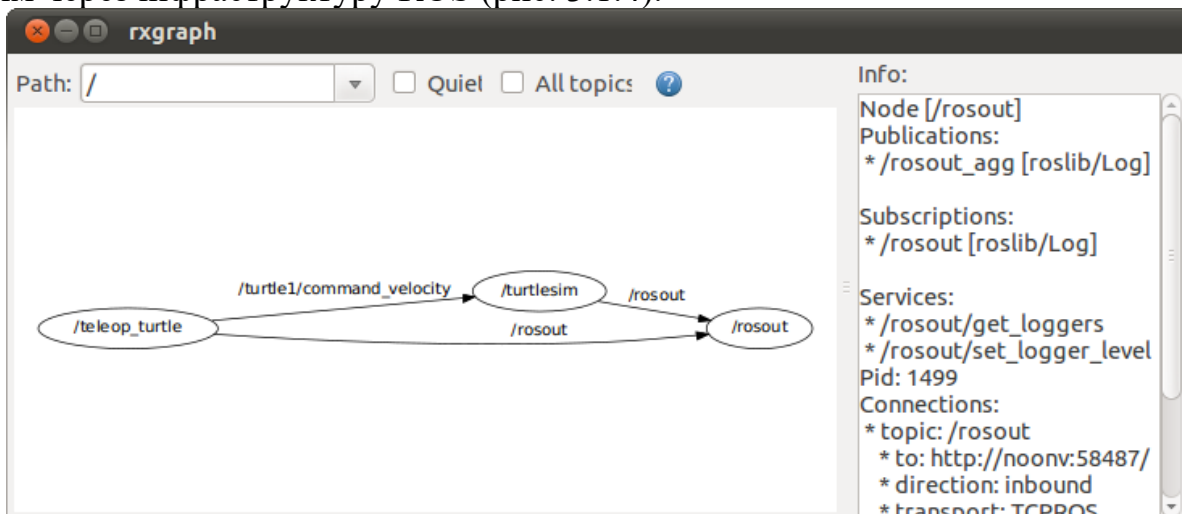


Рис. 3.1.4. Приклад графа в ROS

ROS реалізує кілька різних стилів взаємодії:

- синхронну взаємодію (у стилі RPC) через сервіси. Модель публікація/підписка є дуже гнучкою парадигмою комунікації, але її модель транспортування "багато до багатьох" одностороння передача даних не підходить для взаємодій типу запит/відповідь (RPC), які часто необхідні в розподіленій системі. Запит/відповідь здійснюється через Сервіс, який визначається парою повідомлень: одне для запиту, інше для відповіді. Вузол, що надає сервіс, пропонує його під певним рядковим ім'ям, а клієнт викликає сервіс, надсилаючи повідомлення запиту та чекаючи на відповідь. Клієнтські бібліотеки зазвичай представляють цю взаємодію програмісту так, ніби це віддалений виклик процедури (RPC). Сервіси визначаються за допомогою файлів `srv`, які компілюються у вихідний код за допомогою бібліотеки клієнта ROS. Клієнт може встановити постійну з'єднання з сервісом, що дозволяє забезпечити вищу продуктивність за рахунок зниження стійкості до змін постачальника сервісу.
- асинхронні потоки даних через топіки (Topic). Topics — це іменовані канали, через які вузли обмінюються повідомленнями. Topics мають анонімну семантику публікації/підписки, що розділяє виробництво інформації від її споживання. Зазвичай вузли не знають, з ким вони взаємодіють. Замість цього вузли, зацікавлені в даних, підписуються на відповідну Topic; вузли, що генерують дані, публікують їх на відповідній Topic. На одну Topic може бути кілька видавців і підписників. Topics призначені для односторонньої, потокової комунікації. Вузли, які потребують виконання віддалених викликів процедур (тобто отримання відповіді на запит), повинні використовувати сервіси.
- зберігання даних на сервері параметрів (Parameter Server). Сервер параметрів — це спільний, багатовимірний словник, який доступний через мережеві API. Вузли використовують цей сервер для збереження та отримання параметрів під час виконання. Оскільки він не призначений для високопродуктивної роботи, найкраще його використовувати для статичних, не бінарних даних, таких як параметри конфігурації. Він призначений для глобального доступу, щоб інструменти могли легко перевіряти стан конфігурації системи і при необхідності змінювати його. Сервер параметрів реалізований за допомогою XMLRPC і працює всередині ROS Master, що означає, що його API доступне через стандартні бібліотеки XMLRPC.

Будь-який вузол у системі може отримати доступ до цієї мережі, взаємодіяти з іншими вузлами, переглядати інформацію, яку вони надсилають, і передавати свої дані в мережу.

Розглянемо основні компоненти обчислювального графа ROS (рис. 3.1.5):

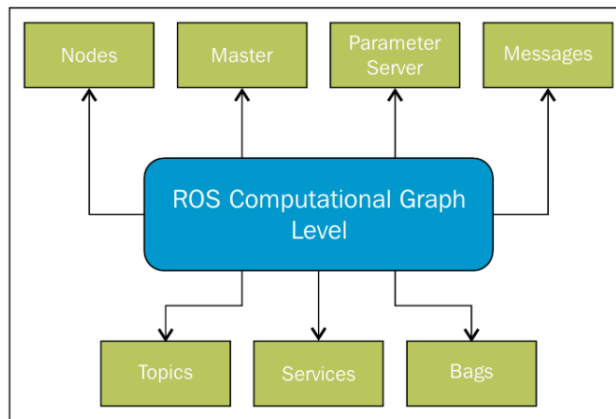


Рис. 3.1.5. Рівень обчислювального графа ROS

- Вузли (Nodes):** Це процеси, які виконують обчислення. ROS є модульною системою, де система керування роботом зазвичай включає безліч вузлів, кожен із яких виконує певну функцію. Наприклад, один вузол керує лазерним далекоміром, інший — колесами, ще один виконує локалізацію, планування шляху чи візуалізацію системи. Вузли пишуться за допомогою клієнтських бібліотек ROS, таких як `roscpp` або `rospy`.

З іншого боку Nodes - це різні виконувані файли в ROS, які взаємодіють один з одним через топіки та сервіси. Вони використовуються для підвищення ефективності та ремонтпридатності системи. Ноди є унікальними в системі, і імена нод не можуть повторюватися. Це дозволяє зробити зв'язки між нодами унікальними.
- ROS Master:** Він забезпечує зв'язок між вузлами. Без Master вузли не зможуть знаходити один одного, обмінюватися повідомленнями чи викликати сервіси.
- Сервер параметрів (Parameter Server):** Це спільний багатовимірний словник, доступний через мережу. Вузли використовують сервер для збереження та отримання параметрів під час виконання.
- Повідомлення (Messages):** Вузли взаємодіють між собою, передаючи повідомлення. Повідомлення — це структура даних, яка містить поля з типами (цілі числа, числа з плаваючою комою, логічні значення тощо) або масивами. Повідомлення можуть містити вкладені структури і масиви, подібно до синтаксису мови C.
- Теми (Topics):** Повідомлення передаються через транспортну систему за принципом «видавець-підписник». Вузол публікує повідомлення в певній темі, а інші вузли підписуються на цю тему для отримання даних. Тема — це ім'я, що ідентифікує тип вмісту повідомлення. Один вузол може одночасно бути видавцем і підписником кількох тем. Видавці та підписники не знають про існування один одного, що забезпечує розділення виробництва інформації від її споживання. Тема схожа на строго типізовану шину повідомлень.
- Сервіси (Services):** Хоча модель тем дуже гнучка, вона не підходить для взаємодії типу «запит-відповідь», що часто потрібна в розподілених системах.

Сервіси використовуються для таких випадків: вузол пропонує послугу під загальним іменем, клієнт відправляє запит і очікує відповіді.

- **Беги (Bags):** Це формат для збереження та відтворення даних повідомлень ROS. Беги є важливим інструментом для зберігання даних, наприклад, із сенсорів, які складно збирати, але вони потрібні для розробки та тестування алгоритмів. Їх часто використовують під час роботи зі складними роботами.

Обчислювальний граф у поєднанні з пакетами утворює систему ROS.

ROS Master і його роль в обчислювальному графі.

ROS Master виконує функцію сервісу імен у обчислювальному графі ROS. Він зберігає інформацію про реєстрацію тем (topics) і сервісів (services) для вузлів ROS. Вузли спілкуються з Master для передачі даних про свою реєстрацію. Отримуючи ці дані, Master надає вузлам інформацію про інші зареєстровані вузли, дозволяючи їм встановлювати необхідні з'єднання. Master також викликає зворотні функції (callbacks) у вузлах, якщо реєстраційна інформація змінюється. Це забезпечує динамічне створення з'єднань під час запуску нових вузлів.

Вузли з'єднуються між собою напряму — Master лише надає інформацію про місцезнаходження, подібно до DNS-сервера. Вузли, що підписуються на певну тему, надсилають запити на з'єднання до вузлів, які публікують повідомлення цієї теми, і встановлюють це з'єднання через узгоджений протокол. Найпоширенішим протоколом у ROS є **TCPROS**, який використовує стандартні TCP/IP-сокети.

Ця архітектура забезпечує незалежність вузлів, де імена слугують основним інструментом для побудови великих і складних систем. Імена в ROS мають важливе значення: кожен вузол, тема, сервіс чи параметр має своє ім'я. Усі клієнтські бібліотеки ROS підтримують можливість переназначення імен через командний рядок, що дозволяє змінювати топологію обчислювального графа під час виконання програми.

Приклад використання: Для керування лазерним далекомірром Hokuyo ми можемо запустити драйвер `hokuyo_node`, який взаємодіє з лазером і публікує повідомлення типу `sensor_msgs/LaserScan` у темі `scan`. Щоб обробити ці дані, ми можемо створити вузол із використанням `laser_filters`, який підписується на повідомлення теми `scan`. Після підписки наш фільтр автоматично почне отримувати повідомлення з лазера.

Зверніть увагу на відокремленість двох вузлів. Вузол `hokuyo_node` лише публікує дані сканування, не знаючи, чи хтось підписаний на цю тему. Водночас фільтр лише підписується на дані, не маючи інформації, чи хтось їх публікує. Ці два вузли можна запускати, зупиняти й перезапускати в будь-якому порядку без ризику виникнення помилок.

Динамічне перепризначення імен: Якщо пізніше ми додамо ще один лазер у нашу систему, перепризначення імен дозволить швидко внести зміни. Наприклад, для першого лазера ми можемо змінити ім'я теми з `scan` на `base_scan` як у драйвері `hokuyo_node`, так і у вузлі-фільтрі. Тепер обидва вузли будуть взаємодіяти через

тему `base_scan` і не отримуватимуть повідомлення в темі `scan`. Потім ми просто запускаємо ще один вузол `hokuyo_node` для нового лазера, залишаючи систему працювати коректно.

3.1.4. Рівень спільноти ROS

Концепції рівня спільноти ROS стосуються ресурсів, які дозволяють різним спільнотам обмінюватися програмним забезпеченням і знаннями. До цих ресурсів належать:

- **Дистрибутиви:** Дистрибутиви ROS — це набори програмного забезпечення певних версій, які можна встановити. Вони виконують роль, подібну до дистрибутивів Linux: спрощують установку програмного забезпечення та забезпечують узгоджені версії компонентів у межах системи.
- **Репозиторії:** ROS використовує розподілену мережу кодових репозиторіїв, у межах якої різні організації можуть розробляти й випускати власні програмні компоненти для роботів.
- **Вікі ROS:** Вікі-спільнота ROS є основною платформою для документації та обміну інформацією. Кожен може створити обліковий запис, додавати власну документацію, виправляти помилки, оновлювати інформацію або писати навчальні матеріали.
- **Система звітів про помилки:** Для отримання інформації про подачу звітів про помилки використовуйте відповідний розділ «Tickets».
- **Поштова розсилка:** Список розсилки **ros-users** є головним каналом для повідомлень про оновлення в ROS, а також форумом для обговорення й вирішення питань, пов'язаних із програмним забезпеченням ROS.
- **ROS Answers:** Сайт питань і відповідей, створений для допомоги у вирішенні питань, пов'язаних із ROS.
- **Блог:** Блог на сайті `ros.org` регулярно публікує оновлення, включаючи фото та відеоматеріали.

Names (Імена). Окрім трьох рівнів концепцій, ROS також визначає два типи імен — імена ресурсів пакета та імена ресурсів графа.

Graph Resource Names (Імена ресурсів графу). Імена ресурсів графу забезпечують ієрархічну структуру, яка використовується для всіх ресурсів у графі обчислень ROS, таких як вузли (**Nodes**), параметри (**Parameters**), теми (**Topics**) і сервіси (**Services**). Ці імена є ключовим елементом у ROS, який забезпечує складання великих і складних систем. Тому важливо розуміти, як працюють ці імена та як ними можна керувати.

Ось кілька прикладів імен:

- / (глобальний простір імен)

- /foo
- /stanford/robot/name
- /wg/node1

Імена ресурсів графу є важливим механізмом в ROS для забезпечення інкапсуляції. Кожен ресурс визначається в межах простору імен, який він може розділяти з багатьма іншими ресурсами. Загалом, ресурси можуть створювати інші ресурси у своєму просторі імен і отримувати доступ до ресурсів у своєму просторі чи у вищих просторах імен. Зв'язки між ресурсами в різних просторах імен можливі, але зазвичай вони здійснюються за допомогою інтеграційного коду, що знаходиться над обома просторами. Ця інкапсуляція ізолює різні частини системи від випадкового доступу до неправильного ресурсу чи зміни глобальних імен.

Імена розв'язуються відносно, тому ресурси не потребують знати, в якому просторі імен вони знаходяться. Це спрощує програмування, оскільки вузли, які працюють разом, можуть бути написані так, ніби вони знаходяться у верхньому рівні простору імен. Коли ці вузли інтегруються в більшу систему, їх можна розмістити в просторі імен, який визначає їхню групу коду.

Наприклад, можна взяти демонстрацію Stanford і демонстрацію Willow Garage і об'єднати їх у нову демонстрацію з підграфами **stanford** і **wg**. Якщо в обох демонстраціях є вузол із назвою camera, конфліктів не буде. Інструменти (наприклад, візуалізація графу) і параметри (наприклад, demo_name), які мають бути видимі для всього графу, можуть бути створені на рівні вузлів верхнього рівня.

Valid Names (Дійсні імена). Дійсні імена мають такі характеристики:

- Перший символ – літера ([a-zA-Z]), тильда (~) або коса риска (/).
- Наступні символи можуть бути буквено-цифровими ([0-9a-zA-Z]), підкресленням (_) або косою рисою (/).
- Виняток: базові імена (описані нижче) не можуть містити косих рисок (/) або тильд (~).

Ось регулярні вирази для перевірки імен:

- (?=.*[A-z0-9_])^[/~A-z][A-z0-9_/]*\$
- ^[A-z][A-z0-9_]*\$ (для базових імен).

Resolving (Розв'язування імен). У ROS існує чотири типи імен ресурсів графу: базові, відносні, глобальні та приватні, які мають таку синтаксис:

- **Базове:** base
- **Відносне:** relative/name
- **Глобальне:** /global/name
- **Приватне:** ~private/name

Розв'язування імен за замовчуванням виконується відносно простору імен вузла. Наприклад, вузол /wg/node1 знаходиться в просторі імен /wg, тому ім'я node2 буде розв'язано як /wg/node2.

- Імена без жодних кваліфікаторів простору є **базовими**. Вони є підтипом відносних і мають такі ж правила розв'язування. Базові імена найчастіше використовуються для ініціалізації імен вузлів.
- Імена, які починаються з /, є **глобальними** і вважаються повністю розв'язаними. Глобальні імена слід уникати, оскільки вони обмежують портативність коду.
- Імена, що починаються з ~, є **приватними**. Вони перетворюють ім'я вузла на простір імен. Наприклад, вузол node1 у просторі /wg/ матиме приватний простір імен /wg/node1. Приватні імена корисні для передачі параметрів конкретному вузлу через сервер параметрів.

Приклади розв'язування імен:

Вузол	Відносне (за замовчуванням)	Глобальне	Приватне
/node1	bar -> /bar	/bar -> /bar	~bar -> /node1/bar
/wg/node2	bar -> /wg/bar	/bar -> /bar	~bar -> /wg/node2/bar
/wg/node3	foo/bar -> /wg/foo/bar	/foo/bar -> /foo/bar	~foo/bar -> /wg/node3/foo/bar

Remapping (Переназначення імен). Будь-яке ім'я у вузлі ROS може бути переназначене під час запуску вузла з командного рядка. Для отримання додаткової інформації див. [Remapping Arguments].

Імена ресурсів пакетів. Імена ресурсів пакетів використовуються в ROS із концепціями файлової системи, щоб спростити процес посилення на файли та типи даних на диску. Вони складаються з назви пакета та назви ресурсу. Наприклад, ім'я std_msgs/String посилається на тип повідомлення String у пакеті std_msgs.

Файли, на які можна посилатися через імена ресурсів пакетів:

- Типи повідомлень (msg).
- Типи сервісів (srv).
- Типи вузлів.

Імена ресурсів пакетів схожі на шляхи до файлів, але значно коротші завдяки здатності ROS знаходити пакети на диску і робити припущення про їхній вміст. Наприклад, опис повідомлення std_msgs/String є скороченням для path/to/std_msgs/msg/String.msg. Аналогічно, тип вузла foo/bar відповідає пошуку файлу bar у пакеті foo з правами на виконання.

При виборі фреймворку ROS не є системою реального часу, проте може використовувати такі системи зокрема OROCOS.

3.2. Microsoft Robotics Developer Studio

Програмне забезпечення **Microsoft Robotics Developer Studio** (надалі MRDS), призначене для розробки додатків у сфері робототехніки, симуляції та керування роботами.

Як випливає з назви, середовище MRDS є продуктом компанії Microsoft. Штаб-квартира цієї транснаціональної корпорації з виробництва програмного забезпечення розташована в місті Редмонд (США, штат Вашингтон). Про бажання зайнятися сферою робототехніки Microsoft оголосила влітку 2006 року, а вже в грудні цього ж року вийшла перша версія Microsoft Robotics Studio. До моменту анонсу близько 30 компаній — відомих постачальників у галузі робототехніки — мали свої продукти, реалізовані на основі інструментарію Microsoft. З того часу регулярно виходять нові версії цієї платформи. Microsoft тісно співпрацює з провідними науково-дослідними інститутами в цьому напрямку. Основні зміни спрямовані на спрощення процесу розробки і покращення продуктивності, удосконалення інструментів програмування та засобів забезпечення безпеки.

Останні версії програмного забезпечення MRDS безкоштовні. Перед початком процесу інсталяції самого пакету відбувається установка всіх необхідних для роботи компонентів. Разом із програмою поставляється безліч прикладів і інструкцій, які допомагають освоїти MRDS.

Середовище MRDS — розробка компанії Microsoft, призначена для програмування складних багатопоточних додатків з реактивною моделлю поведінки, які використовуються для управління робототехнічними системами. Під реактивною поведінкою розуміють активність, джерелом якої переважно є зовнішні подразники, умови, в яких перебуває робототехнічна система, обставини тощо, тобто певні зовнішні сили визначають характер і напрямок поведінки.

Необхідність створення таких додатків є не лише в робототехніці, тому Robotics Developer Studio використовується і для створення додатків, які не стосуються робототехніки (наприклад, соціальна мережа MySpace використовує MRDS як складову частину серверного ПЗ). Програми в Robotics Developer Studio створюються у вигляді діаграм на візуальній мові VPL (Visual Programming Language), яка по суті є візуалізатором зв'язків між окремими паралельно виконуваними компонентами (або веб-сервісами), з яких складається програма (рис. 3.1.6).

Система складається з таких основних частин:

- **Concurrency and Coordination Runtime (CCR)** – бібліотека для роботи з паралельними та асинхронними потоками даних. Бібліотека дозволяє прозоро організувати розподілені та паралельні обчислення, виконуючи завдання на різних обчислювальних пристроях. Це дуже корисно при програмуванні роботів, оскільки програми для роботів за своєю природою реактивні та потребують обробки потоків даних, що надходять одночасно з кількох сенсорів, причому частина обчислень може виконуватися безпосередньо на роботіві, а частина – на комп'ютері поза ним.

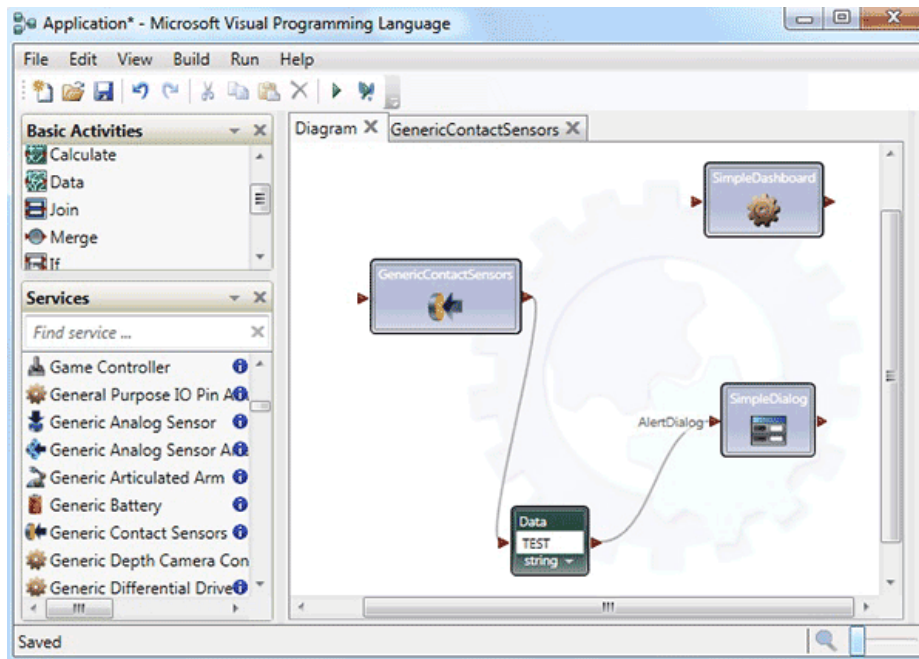


Рис. 3.1.6. Інтерфейс програмного забезпечення Visual Programming Language

- **Decentralized Software Services (DSS)** – середовище виконання, яке забезпечує подання компонентів програми у вигляді веб-сервісів і спрощує організацію взаємодії між ними. Взаємодія між веб-сервісами здійснюється за спеціальним протоколом
- **Decentralized Software Services Protocol (DSSP)**. DSS дозволяє створювати розподілені додатки, яким не важливо, на якому обчислювальному пристрої виконується той чи інший компонент — на одному з комп'ютерів робота чи на комп'ютері поза ним, якщо вони з'єднані єдиною мережею.
- **Visual Programming Language (VPL)** – візуальна мова і редактор для неї, використовуваний для конфігурації сервісів. Сервіси можна перетягнути на діаграму, зв'язати їх входи та виходи, налаштувати їх атрибути. Отримана на такій мові діаграма сильно нагадує діаграми LabView, відображаючи залежність між компонентами за даними.
- **Visual Simulation Environment (VSE)** – тривимірне середовище симуляції поведінки робота у віртуальному світі. Для створення реалістичності використовується технологія NVIDIA PhysX. Має досить багаті можливості для симуляції фізики та потужні засоби відображення тривимірної графіки, що дозволяє створювати складні та красиво виглядаючі моделі світу, з яким робот може взаємодіяти. У комплект середовища входить кілька моделей оточення, зокрема модель квартири, в якій працює так звана "стандартна модель" робота — трьохколісна платформа з обчислювальним модулем, встановленим на ній, сенсором Microsoft Kinect, інфрачервоними датчиками відстані та сонаром (рис. 3.1.7).

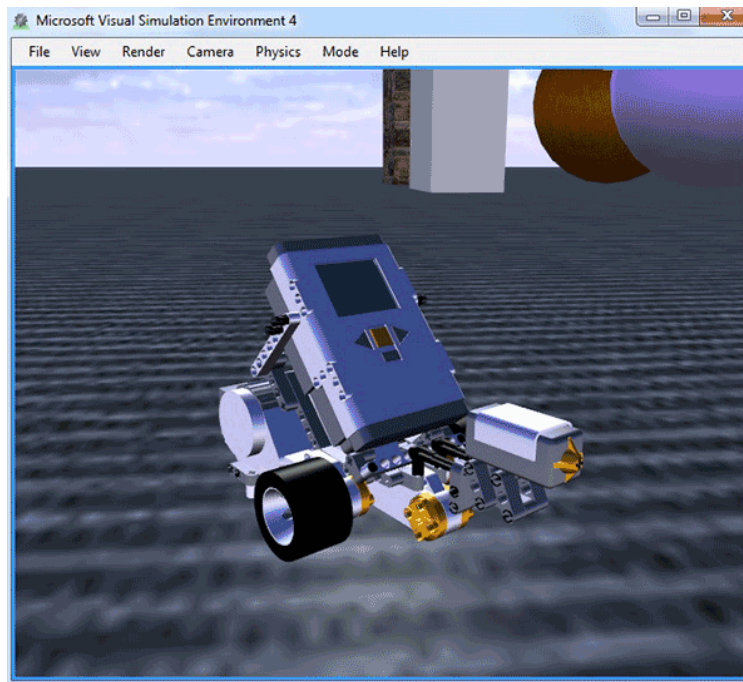


Рис. 3.1.7. Інтерфейс Visual Simulation Environment

Серед інших особливостей платформи MRDS слід відзначити підтримку мов програмування C#, JScript, Visual Basic .NET і IronPython; спрощений механізм доступу до мікроконтролера, датчиків та виконавчих пристроїв робота; модульність при підключенні нових сервісів.

3.3. Середовище NXT-G

NXT-G — єдиний інструмент програмування, який постачається в комплекті з конструктором *Lego Mindstorms NXT*. Середовище базується на системі LabView, середовищі візуального програмування від компанії National Instruments. В LabView як мова програмування використовується візуальна мова G (через що NXT-G і отримала свою назву). Мова G моделює процес обчислень, орієнтований на дані, де явно визначається не послідовність виконання операторів, а зв'язки між блоками за даними. Блок програми може надавати деякі вихідні дані, які можуть слугувати вхідними даними для іншого блоку. Блоки починають виконуватись, коли мають дані на всіх входах. Якщо кілька блоків одночасно мають дані на всіх входах, вони виконуються паралельно. Такий підхід значно відрізняється від підходу, прийнятого в імперативному програмуванні, але тим не менше він широко розповсюджений серед інженерів і вчених. Наприклад, на тих же принципах побудовано інше відоме візуальне середовище програмування для наукових обчислень і моделювання Matlab/Simulink.

Основна проблема цього середовища полягає у досить слабкій підтримці математичних виразів. Математичні формули тут, як і вся програма, будуються з

блоків. Є блоки арифметичних операцій, блоки для читання та запису значень у змінні, блок, що зчитує значення константи, блоки для зчитування показників з сенсорів. Таким чином, навіть для програмування нескладної формули потрібно зображати блоками дерево розбору виразу, яке цю формулу задає. Для ілюстрації серйозності цієї проблеми достатньо сказати, що програма, яка реалізує пропорційно-диференціальний регулятор для руху робота вздовж лінії або навколо перешкоди, на мові C займає близько десятка рядків, тоді як на NXT-G не поміщається на одному екрані і є досить складною для розуміння.

Що стосується простоти і зручності в роботі, середовище NXT-G спеціально створювалося для початківців і тому досить ергономічне. За думкою деяких користувачів, воно навіть занадто ергономічне, оскільки не дозволяє вільно розміщувати блоки на діаграмі, автоматично (і не завжди вдало) прокладає з'єднувальні лінії між блоками тощо. Для застосування NXT-G важлива ще одна його особливість: більшість властивостей елементів не відображаються на діаграмі, а доступні лише через редактор властивостей. Також у NXT-G немає засобів налагодження, текстова форма програми не генерується.

До переваг продукту можна віднести те, що він постачається разом з конструктором і доступний для завантаження з сайту виробника безкоштовно. Крім того, продукт досі розвивається і оновлюється. Засобами LabView можливо додавати сторонні блоки, крім того, сам NXT-G дозволяє виділити набір блоків в підпрограму і використовувати її як новий блок.

3.4. Середовище LabVIEW

Вищезазначене середовище програмування NXT-G було розроблено Массачусетським технологічним інститутом на основі більш серйозного інструменту – графічного середовища LabVIEW, створеного для інженерів електронних пристроїв компанією National Instruments. Цей інструмент дозволяє створювати програми, орієнтовані на конкретні пристрої або системи пристроїв, він містить величезний набір засобів для створення програм для практично всіх сучасних пристроїв, а також для зручного побудови схем взаємодії пристроїв. На сьогодні цей інструмент є дуже популярним. По суті, він є значно удосконаленим варіантом простих середовищ Mindstorms, більш стабільним, що дозволяє виконувати складні математичні операції, а також легко організовувати взаємодію між блоком NXT і комп'ютером або між кількома блоками. Останнє є суттєвим плюсом LabVIEW, оскільки у інших інструментів передача даних організована складніше. Перехід від найпростіших середовищ програмування до LabVIEW набагато легший для початківців, ніж перехід від графічного середовища до текстового. Основним недоліком LabVIEW є його обмежена доступність. Інший мінус цього інструменту полягає в тому, що з ростом складності програми, намальованої у вигляді піктограм, втрачається наочність, і якщо проста програма

виглядає зрозуміло, то більш складна – хаотично, велика кількість піктограм на екрані одночасно сильно ускладнює її читабельність.

3.5. Програмування колаборативних роботів на прикладі мови програмування Melfa Basic. Робота із інтерфейсом програмування на комп'ютері

Робоче вікно (рис. 3.2.1) студії розробки для промислового робота серії Melfa Basic складається із п'яти областей:

1. візуалізація симуляції роботи робота (маніпулятора);
2. вікно списку файлів;
3. візуалізація програмного коду;
4. таблиця із координатами збережених точок;
5. вікно системних сповіщень.

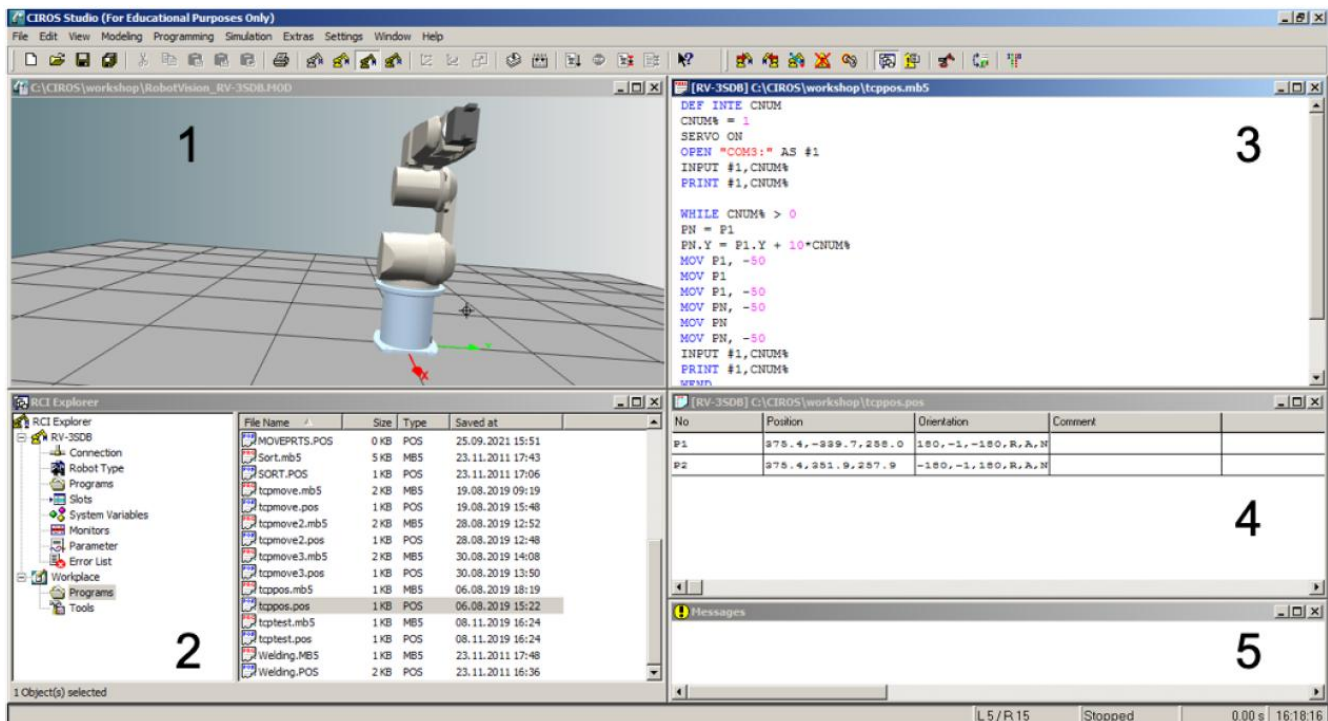


Рис. 3.2.1. Робоче вікно студії розробки для промислового робота серії Melfa Basic

Розглянемо порядок дій при складанні програми, для чого знадобиться панель інструментів (рис. 3.2.2). У першу чергу, за допомогою натискання кнопки «Init Connection» встановлюємо зв'язок із роботом, і після появи вікна із назвою робота погоджуємося натисканням кнопки «ОК». Далі визначаємо і зберігаємо важливі точки у таблиці, для чого

підводимо робот до бажаної точки у ручному режимі. Натискаємо на панелі інструментів кнопку «Robot position → PC», в результаті чого симуляція робота повинна переміститися у таку ж саму конфігурацію.

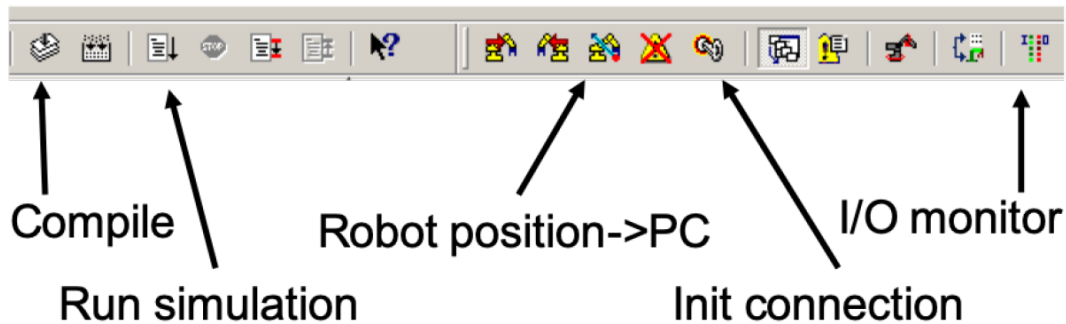


Рис. 3.2.2. Розташування кнопок на панелі інструментів робочого вікна Melfa Basic

Необхідні важливі для програми точки зберігаються наступним чином. Натискаємо правою кнопкою миші на вільному просторі таблиці точок і обираємо «Insert position». Для збереження нових значень необхідно обрати пункт програми (він повинен підсвітитися), натиснути на ньому правою кнопкою миші і обрати «Accept position».

Тепер необхідно записати код програми у відповідному файлі. Перевірка коду на наявність помилок здійснюється при натисканні кнопки «Compile» згаданої панелі інструментів – у вікні системних повідомлень з’являться помилки компіляції «error». Симуляція робота запускається кнопкою «Start», і якщо симулятор показав успішність виконання заданих дій, файли можна завантажувати на робот.

Вікно файлів «RCI Explorer» містить два основних розділи: «RV-3SDB» – тут зберігаються файли робота, і «Workplace» – місце для локальних файлів на комп’ютері. Для завантаження файлів на робот потрібно перейти у папку «Programs» розділу «Workplace», виділити одразу два файли програми, наприклад, «MOVEPRTS.MB5, MOVEPRTS.POS», натисканням правої кнопки миші викликати контекстне меню і вибрати пункт «Download», після чого погодитися із системними сповіщеннями. Для того, щоб робот почав виконувати програму, необхідно виконати наступне: перевести робот у автоматичний режим; у вікні файлів перейти у папку «Programs» розділу «RV-3SDB», обрати файл із програмою «MOVEPRTS», натиснути правою кнопкою миші і у контекстному меню обрати пункт «Start (CYC)». Зупинити виконання програми можна або натиснувши кнопку «STOP» на пульті керування, або натиснувши правою кнопкою миші обрати пункт «STOP» у контекстному меню. Зовнішній вигляд вікна при запуску програми зображений на рис. 3.2.3.

Якщо при виконанні програми на роботі виникають помилки, інформацію про них можна отримати або в пункті «Error list» розділу «RV-3SDB» (рис. 3.2.4), де

список помилок супроводжується часом і датою виникнення, описом, типом повідомлення (попередження або системна помилка), або прочитати на головному екрані пульта дистанційного керування робота.

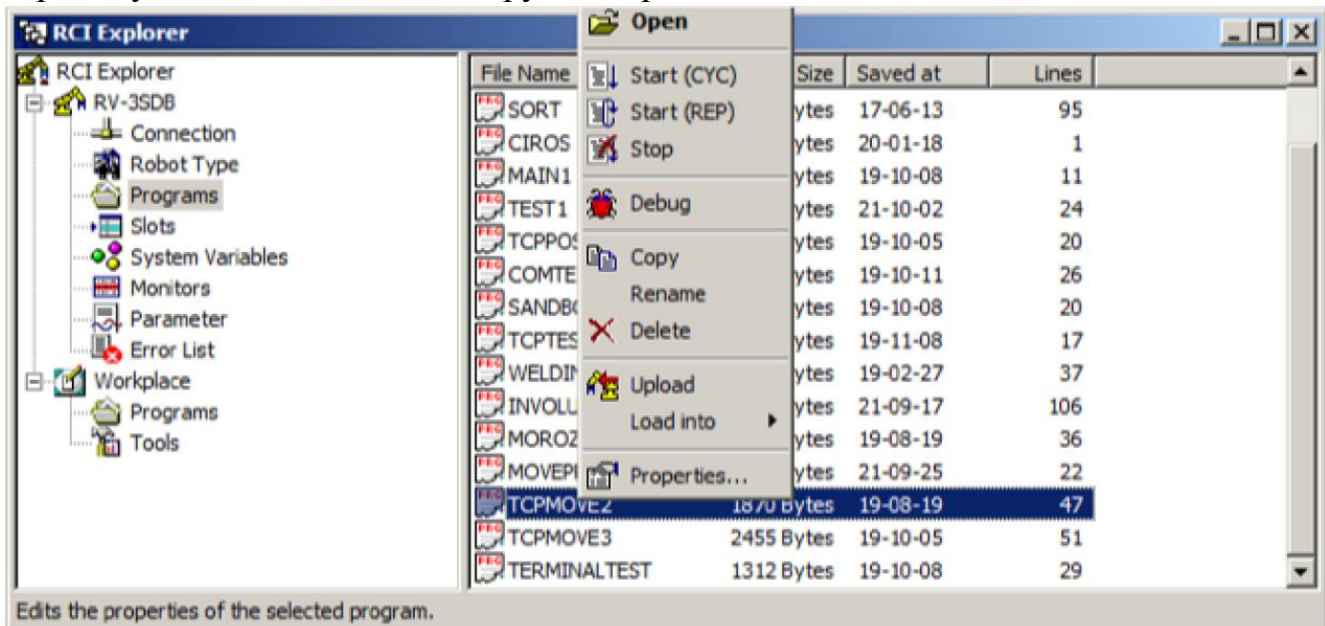


Рис. 3.2.3. Запуск програми у вікні списку файлів

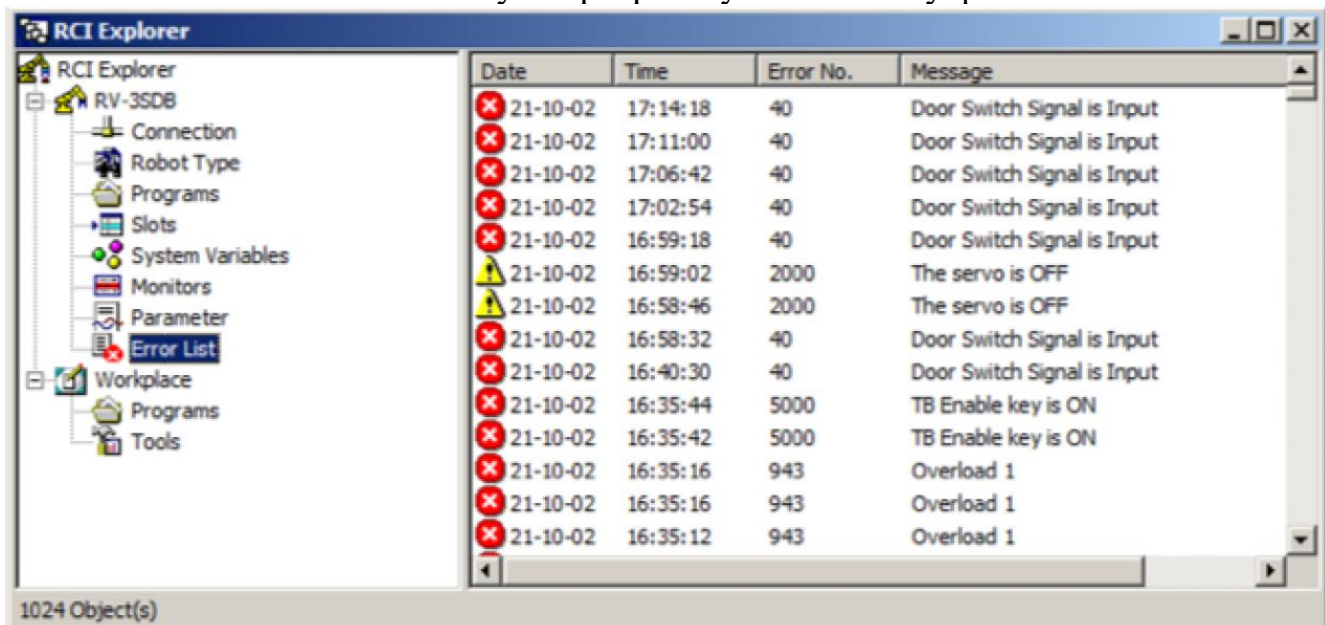
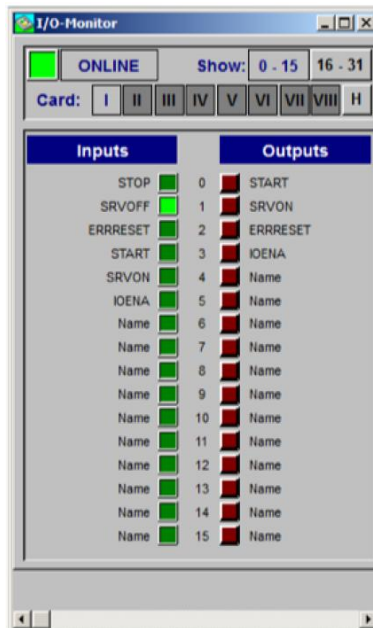


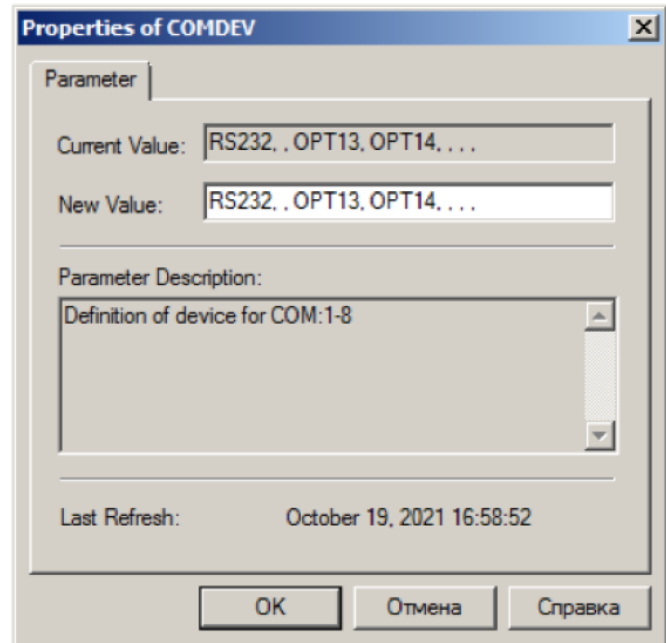
Рис. 3.2.4. Вікно списку помилок при виконанні програми на контролері робота

Вікно «I/O Monitor» (рис. 3.2.5, а), яке викликається натисканням відповідної кнопки на панелі інструментів, допомагає відслідковувати стани сигналів вводу-виведення. Номер карти розширення, що встановлена в контролер, обирається в рядку «Card:», де світло-сірим кольором показані карти, які доступні для вибору, а темно-сірим – кольором виділені вільні слоти. Знизу, в центрі вікна, показані номери портів. Ліва колонка «Inputs» виводить логічне значення на порті вводу

(світлий колір означає логічну 1, темний – 0). За допомогою натискання кнопок у правій колонці «Outputs» можна задати значення на виході відповідного порту.



а)



б)

Рис. 3.2.5. Вікно «I/O Monitor» сигналів портів вводу-виводу (а); вікно налаштування системного параметра COMDEV (б)

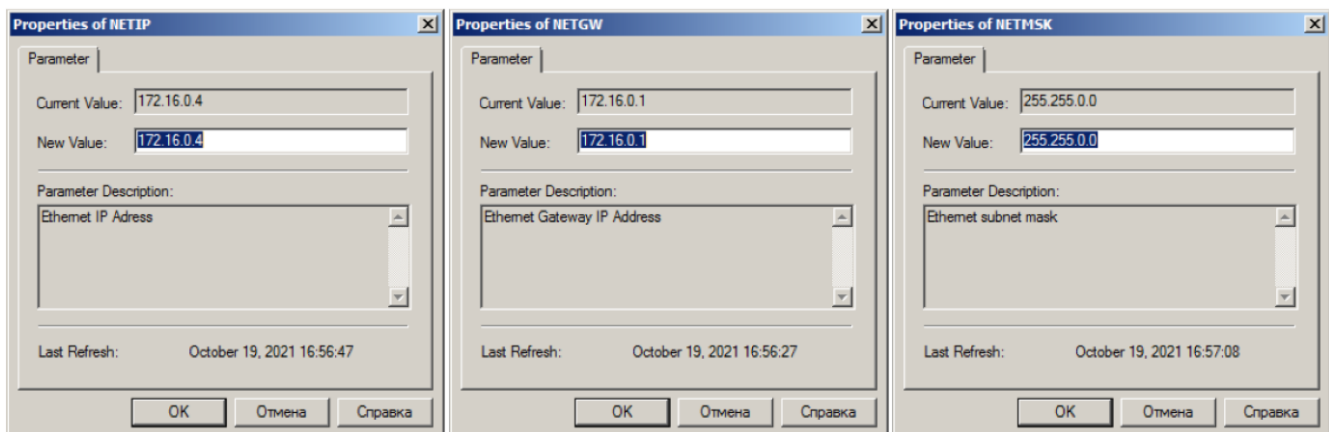


Рис. 3.2.6. Вікно налаштування мережевих параметрів контролера

Для реалізації мережевих можливостей в алгоритмі необхідно вказати список доступних інтерфейсів у вікні налаштування системного параметра COMDEV. Можливість застосування різних інтерфейсів визначають встановлені карти розширення. Усі інтерфейси, які будуть прив'язані в алгоритмі до системних портів COM1-8, записуються через кому у властивостях системного параметра. Так, на рис. 3.2.5, б, зображений приклад налаштування, коли змінній COM1 відповідає

інтерфейс RS232, COM3 відповідає OPT13, COM4 — OPT14. Решта змінних COM не прив'язані до жодного інтерфейсу, їм відповідають пусті коми у властивостях налаштування.

Інтерфейс TCP/IP налаштовується у вікнах системних параметрів NETIP, NETGW і NETMSK (рис. 3.2.6) для визначення, відповідно, IP-адреси контролера, адреси маршрутизації і маски підмережі. Вказані значення аналогічним чином повинні бути внесені у налаштування студії розробки у вікні Communication Port (рис. 3.2.7).

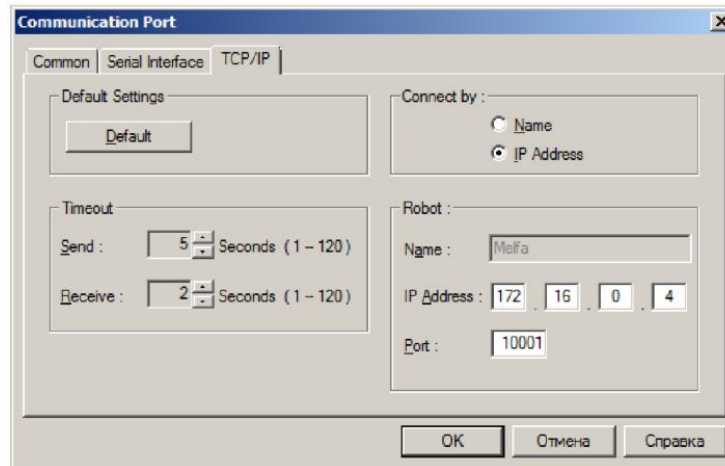


Рис. 3.2.7. Вікно налаштування параметрів підключення студії розробки

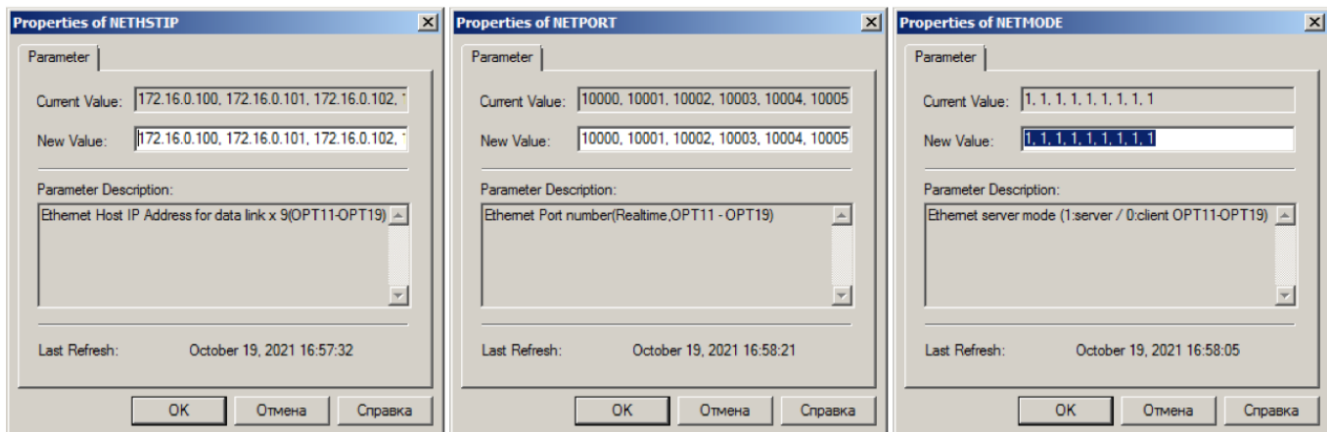


Рис. 3.2.8. Вікна налаштування вхідних мережевих параметрів

Для взаємодії по TCP/IP повинні бути також налаштовані вхідні мережеві параметри (рис. 3.2.8). Так, у вікні NETHSTIP вказуються IP-адреси комп'ютерів, що входять до складу локальної мережі і від яких дозволяється отримувати вхідні повідомлення. У вікні NETPORT зазначаються номери портів, які будуть відкриті для підключень. У вікні NETMODE визначається роль, яку відіграє контролер для кожного підключення (за замовчуванням встановлена роль «сервер», опціонально обирається «клієнт»). У кожному із трьох вікон параметри вказується через кому, вони будуть відповідно використані для віртуальних інтерфейсів OPT11-OPT19.

3.5.1 Синтаксис мови програмування Melfa Basic

Розглянемо синтаксис мови Melfa Basic, необхідний для складання програм автоматизації різних технологічних процесів.

3.5.1.1 Типи змінних

Змінна із визначенням її типу у мові Melfa Basic створюється за допомогою команди **DEF**, після якої вказується обов'язковий параметр із типом змінної, яка створюється, і параметр ім'я.

Основними типами змінних у Melfa Basic є:

- DEF INTE WORK** – чисельні змінні цілого типу (+/- 32767)
- DEF FLOAT SMALLSIM** – чисельні змінні із плаваючою точкою (+/-1.7E+38)
- DEF DOUBLE BIGSIM** – чисельні змінні із плаваючою точкою (+/-1.7E+308)
- DEF POS PHELPX** – змінні позиціонування у прямокутній системі координат

Змінній певне значення присвоюється за допомогою знаку рівності:

WORK = 100

Для того, щоб задати змінну позиціонування у прямокутній системі координат, необхідно як параметр вказати матрицю, що складається з шести елементів, які відповідають значенням координат X, Y, Z, A, B, C (навіть за умови, що вони нульові):

PHELPX = (+75.00,+0.00,+0.00,+0.00,+0.00,+0.00)

Синтаксис Melfa Basic передбачає можливість задати змінну без використання команди **DEF** у скороченому форматі. Так, наприклад, якщо у імені змінної на першому місці прописана літера P, буде створена змінна позиціонування у прямокутній системі координат

PY = (0,-100,0,0,0,0)

якщо ж літера I – чисельна змінна цілого типу

IX = 2

3.5.1.2. Побудова алгоритмічних конструкцій

Команди початку і завершення роботи

На початку програми необхідно вказати команду, яка забезпечує вмикання електроприводів, за рахунок яких здійснюється рух робота:

```
SERVO ON
```

Регулювання значення максимальної кутової швидкості, що розвивається роботом, здійснюється у відсотках від максимальної, може бути вказане як на початку програми, так і всередині програмного коду при необхідності регулювання швидкості:

```
JOVRD 100
```

Аналогічно, регулювання значення максимальної лінійної швидкості, яка розвивається роботом, здійснюється у відсотках від максимальної:

```
SPD 100
```

Команда на відключення електроприводів вказується у кінці програми:

```
SERVO OFF
```

У випадках тривалого простою робота (наприклад, під час тривалого очікування команди від інших пристроїв), при побудові алгоритмів також можна використовувати команди на відключення або вмикання електродвигунів.

Завершення виконання алгоритму і переведення контролера у режим очікування забезпечується відповідною командою:

```
END
```

Створення умов

Умова створюється наступним чином. Всередині між операторами **IF ... THEN** вказується умова істинного значення. Після оператора **THEN** перераховуються ті команди, які будуть виконані у випадку підтвердження умови істинності. Далі прописується оператор **ELSE**, після якого зазначаються команди, що необхідно виконати при невиконанні умови. Уся конструкція створення умови завершується командою **ENDIF**:

```
IF M1=1 THEN  
    MOV P1  
ELSE  
    MOV P2  
ENDIF
```

Можлива конструкція, коли створюється неповна умова із перерахуванням команд тільки істинної умови – тоді команда **ELSE** не вказується, конструкція завершується командою **ENDIF**. Приклад використання конструкції неповної умови (команда **MOV P2** буде виконана в будь-якому випадку):

```
IF M1=1 THEN
    MOV P1
ENDIF
MOV P2
```

Цикли

Створення *циклу із використанням ітеративної змінної* починається з команди **FOR**, після чого зазначається ім'я змінної та обидва значення границі ітерації. Далі йде перелік команд, що будуть виконуватися у цьому циклі. Змінну ітерації можна використати для здійснення розрахунків всередині циклу. Закінчується цикл командою зміни значення змінної **NEXT**, після якої вказується ім'я змінної ітерації.

Наприклад, цикл триразового повторення руху (три ітерації з індексами 0, 1, 2) виглядає наступним чином:

```
FOR I1=0 TO 2
    MOV P1
    MOV P2
NEXT I1
```

Цикл із попередньою умовою починається командою **WHILE**, після якої прописується умова виходу із циклу. Подальші рядки містять перелік команд для повторення у циклі. Завершується цикл командою **WEND** із подальшою повторною перевіркою умовою виходу з циклу. Для коректної роботи циклу із попередньою умовою необхідна внутрішня реалізація зміни значень, які використовуються у перевірці умови. Цикл із попередньою умовою виглядає так:

```
WHILE M1>0
    ...
WEND
```

За необхідності можна створити *нескінчений цикл*, переривання якого здійснюється внутрішньою умовою із командами виходу з циклу або завершення роботи. При розробці алгоритму вихід із нескінченного циклу можна здійснити за допомогою переривання або за умови здійснення зовнішньої події.

Приклад побудови нескінченного циклу:

```
WHILE 1  
    ...  
WEND
```

Примусовий вихід із циклу можна здійснити за допомогою команди **BREAK**, яка часто використовується із умовою перевірки змінних, наприклад:

```
WHILE M1 <> 0  
    IF M2 = 0 THEN BREAK  
WEND
```

Процедури

При необхідності переходу до певної послідовності дій використовується команда **GOTO**, після якої через пробіл прописується рядок коду, який необхідно виконати. Необхідний рядок попередньо потрібно помітити відповідним номером.

У наведеному нижче прикладі команда **MOV P1** виконана не буде, а команда **MOV P1** – буде:

```
70 GOTO 90  
80 MOV P1  
90 MOV P2
```

Часто зручно не нумерувати кожний рядок програмного коду, як це показано у наведеному прикладі, а використовувати текстові мітки. Рядок із міткою починаються символом «*», далі вказується текстова назва мітки. Тоді у наведеному прикладі після команди **GOTO** необхідно вказати назву текстової мітки:

```
GOTO *L1  
MOV P1  
*L1  
MOV P2
```

У випадках, коли в програмі наявні певні послідовності дій, які повинні виконуватися в різні моменти часу, застосування команди **GOTO** може бути ускладненим або незручним, оскільки необхідно повертатися на різні ділянки програмного коду. Для опису структур процедур потрібно застосовувати команду **GOSUB** для переходу до необхідної послідовності дій із утворенням точки виклику і можливістю повернення до неї. Після команди через пробіл вказується назва текстової мітки процедури. Повернення до точки виклику здійснюється за

допомогою команди RETURN. Нижче наведений код, за яким послідовно будуть виконані точки P1, P11, P12, P2.

```
SERVO ON
MOV P1
GOSUB *L1
MOV P2
SERVO OFF
END

*L1
    MOV P11
    MOV P12
RETURN
```

Для команд GOTO та GOSUB можливо створювати конструкції за умовою із вибором з декількох варіантів із використанням команди ON, як, наприклад, це показано нижче:

```
ON M1 GOSUB *L1, *L6
```

У цьому випадку M1 є змінною, яка буде перевірятись на відповідність певній умові: якщо M1=1, відбудеться перехід до процедури *L1; якщо M1=2 – перехід до процедури *L6. У випадку M1=0 здійсниться перехід до наступного рядку. Виклик команди GOSUB викликає створення точки виклику для команди RETURN. Замість команди GOSUB у такій конструкції можна також використати команду GOTO без створення точки виклику.

Створення функцій

Існує дуже зручний спосіб скорочення опису часто використовуваних математичних операцій – створення функцій та звернення до них за їх ім'ям.

Функції створюються за допомогою команди DEF із зазначенням ім'я функції, у дужках вказуються параметри функції, а після знаку рівності записується вираз для розрахунку. Наведений нижче приклад ілюструє створення функції розрахунку середнього значення двох чисел:

```
DEF FNMAVE(MA,MB)=(MA+MB)/2
```

де FNMAVE – ім'я функції;
(MA,MB) – параметри функції;
(MA+MB)/2 – операція функції.

Програмний код, де створюється функція, присвоюються значення двом змінним із подальшим обчисленням їх середнього значення виглядатиме наступним чином:

```
DEF FNMAVE(MA,MB)=(MA+MB)/2
MDATA1=20
MDATA2=30
MAVE=FNMAVE(MDATA1,MDATA2)
```

Використання переривання і таймерів

Перериванням називають таку *подію*, яка зупиняє виконання основного коду алгоритму програми і призводить до виконання спеціального коду, що описаний саме для цієї події. Може викликатися за *умовою*, описаною для виклику переривання. Умова, в свою чергу, описується для таймера / таймерів, для змінної алгоритму, для портів вводу-виводу. Переривання використовується для обробки певних подій за часом, для контролювання даних від зовнішніх пристроїв (давачів, кнопок керування та ін.), для контролю стану системи.

Переривання оголошується за допомогою команди **DEF ACT**, після якої вказуються параметри переривання: порядковий номер переривання (1), умова виконання (2), далі, через пробіл, команда, яку необхідно виконати у разі істинності умови (3). Як команду виконання нерідко зазначають команду переходу до процедури обробки переривання **GOSUB**. Приклад оголошення переривання:

```
DEF ACT 1, M_IN(10)=1 GOSUB *SUB1
```

де **DEF ACT 1** – оголошення переривання із порядковим номером 1;

M_IN(10)=1 – умова спрацювання оголошеного переривання;

GOSUB *SUB1 – команда переходу до процедури ***SUB1** у випадку спрацювання умови.

У мові Melfa Basic переривання створюються неактивованими. Це означає, що у пам'яті контролера створюється умова виклику переривання, однак воно не буде перевірятися при виконанні основної частини програми у фоновому режимі. Для перевірки умови переривання необхідно активувати спеціальною командою у потрібному місці початкового коду:

```
ACT 1 = 1
```

Якщо робот повинен виконати послідовність дій, яка не може бути зупинена з міркувань безпеки, за необхідності створене переривання може бути деактивованим. Так, наприклад, якщо робот здійснює наповнення форм певною рідкою речовиною, виконання переривання може призвести до порушення технологічного процесу і до збитків для виробника.

Переривання деактивується виконанням такої команди:

```
ACT 1 = 0
```

З метою обробки системних подій у часі викликання переривань часто супроводжується використанням *таймерів*. Таймери запускаються та перевіряються за допомогою системних змінних `M_TIMER` із зазначенням у дужках номера таймера. Розглянемо приклад визначення часу, що необхідний для виконання двох команд: спочатку створюється системний таймер №1 із нульовим початком відліку часу, потім виконуються дві команди, і на завершення у змінну `M1` записується значення часу таймера, яке було витрачено на виконання двох команд:

```
M_TIMER(1)=0  
MOV P1  
MOV P2  
M1=M_TIMER(1)
```

Наступний приклад ілюструє випадок, коли під час виконання програми необхідно викликати переривання кожні 10 секунд. Для виконання поставленої задачі у коді описується переривання, створюється таймер і активується переривання, а при спрацюванні переривання таймер скидається на нуль:

```
DEF ACT 1, M_TIMER(1)>10 GOSUB *SUB1  
M_TIMER(1) = 0  
ACT 1 = 1  
...  
*SUB1  
    MOV P_RES  
    M_TIMER(1) = 0  
RETURN 0
```

Створення пауз

У випадках, коли необхідно призупинити виконання програми на певний визначений час, використовується команда `DLY`. Так, наприклад, паузи корисно застосовувати для очікування на завершення виконання дій зовнішніми пристроями, в яких не реалізований зворотній зв'язок. Після об'яви команди `DLY` через пробіл зазначається тривалість паузи в секундах. Приклад паузи довжиною 0,1 сек.:

```
DLY 0.1
```

За умов наявності зовнішніх сигналів, які означають, що певна подія відбулася, можна використати команду **WAIT**. Об'ява цієї команди призупиняє виконання програми до того часу, доки умова, описана у якості параметра команди, не стане істинною. Використання цієї команди повинно супроводжуватись зовнішньою умовою, яка призведе до спрацювання умови, в протилежному випадку виконання програми буде призупинене до вимикання контролера. Такий вид пауз часто використовують при очікуванні або сигналу від кнопок керування (наприклад, «Пуск», «Стоп»), або сигналу про завершення виконання певної дії зовнішнього пристрою. Як приклад використання команди на очікування:

```
WAIT M_IN(1) = 1
```

3.5.2 Команди керування рухом

Наближений рух до певної точки

Переміщення робота із поточного положення у певну точку за траєкторією, що наближена до прямої лінії, здійснюється за допомогою команди

```
MOV P1
```

де **P1** – змінна, що складається з координат прямокутного простору у форматі (X, Y, Z, A, B, C), які прив'язані до площини переміщення робота.

Переміщення буде наближеним до прямолінійного, однак за наявності певних кінематичних обмежень, рух може бути зображений у вигляді дуги для виключення недопустимих положень. Ілюстрація такого руху наведена на рис. 3.3.1 (**P_CURR** – поточне положення робота).



Рис. 3.3.1. Ілюстрація використання команди **MOV**

У промисловій автоматизації дуже часто використовуються команди наближення до деталі або віддалення від неї, тому синтаксис команди **MOV** передбачає наявність спеціального параметра:

```
MOV P1, -50
```

де -50 – зміщення від початкової точки **P1** у системі координат інструмента по осі Z на -50 мм.

Напрямок зміщення визначається орієнтацією інструменту робота у точці P1. Наприклад, якщо поточне положення інструмента зорієнтоване вниз, значення -50 мм по осі Z означає, що інструмент повинен зміститися вгору на 50 мм.

Наведемо приклад для інструменту, який є вертикально орієнтований і направлений вниз. Необхідно на початку програми із початкового положення перейти у точку на 100 мм вище деталі, а потім опустити до положення самої деталі у точку P1. Програма матиме вигляд:

```
MOV P1, -100
MOV P1
```

При написанні коду програми для опису зміщення у прямокутній системі координат, яка прив'язана до робота, можна створювати *допоміжні змінні координат*, де у дужках зазначаються Декартові координати зміщення (відповідно, X, Y, Z, A, B, C):

```
RHELPX=(75,0,0,0,0,0)
```

Окрім явного задання точок (як, наприклад, наведено вище, P1), у команді MOV можна здійснювати *математичні операції* над точками, наприклад:

- сума змінних для зміщення

```
MOV P1+RHELPX, -50
```

- добуток на змінні для виконання операцій у циклі

```
MOV P1+RHELPX*I1, -50
```

Логіка виконання зазначених операцій описується матричними перетвореннями: при добутку на скаляр кожний елемент матриці помножується на нього, а при додаванні змінних положень додаються відповідні елементи матриці координат.

Оскільки виконання команди руху робота в певну точку, особливо на малих швидкостях, займає тривалий час, синтаксис команди MOV передбачає додатковий параметр WITHIF, за допомогою якого здійснюється перевірка додаткової умови під час переміщення. Ця конструкція особливо цікава у тому випадку, коли при отриманні зовнішнього сигналу необхідно зупинити рух у точку, як це показано у прикладі:

```
MOV P1 WITHIF M_IN(11) = 0, SKIP
```

де M_IN(11) = 0 – умова, яка перевіряється;

SKIP – команда переходу до наступного рядку у разі істинності виразу.

Точний рух до певної точки

Переміщення робота із поточного положення у певну точку по прямолінійній траєкторії оголошується командою

MVS P1

Призначення цієї команди – точне позиціонування інструменту та прямолінійний рух.

Слід зауважити: на початку програми цю програму використовувати не рекомендується. Якщо переміщення із поточного положення у бажане по прямій перетинає області обмеженої кінематики, команда не виконається і робот зупиниться із помилкою. Виконання команди **MVS** ілюструє рис. 3.3.2.



Рис. 3.3.2. Ілюстрація використання команди **MVS**

Аналогічно синтаксису команди **MOV**, разом із командою **MVS** можна використовувати допоміжний параметр і математичні операції:

MVS P1+PHELPX, -50
MVS P1+PHELPX*I1, -50

Переміщення по дузі

Мова Melfa Basic, що розглядається в даному розділі, для вирішення конкретних задач передбачає використання декількох команд для руху робота або маніпулятора по дузі. Вони подібні за синтаксисом, але мають принципові відмінності за виконанням.

У першій команді **MVR** оголошується три параметри **P1, P2, P3**. При її виконанні робот із поточного положення по прямій (або майже по прямій) переміщується в точку **P1**, після чого рухається по дугоподібній траєкторії (рис. 3.3.3) послідовно через точки **P1, P2** та **P3**:

MVR P1,P2,P3

В іншій команді **MVR2** також оголошуються три параметри **P1, P2, P3**:

MVR2 P1,P2,P3

При її виконанні, аналогічно із командою **MVR**, робот наближено прямолінійно рухається до точки **P1**, далі по дугоподібній траєкторії, що охоплює усі три точки **P1, P2, P3**, але, пройшовши точку **P2**, до **P3** не доходить (рис. 3.3.4).

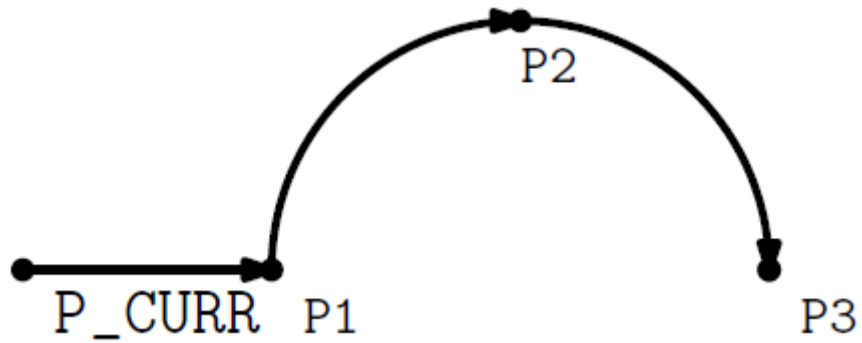


Рис. 3.3.3. Ілюстрація використання команди MVR

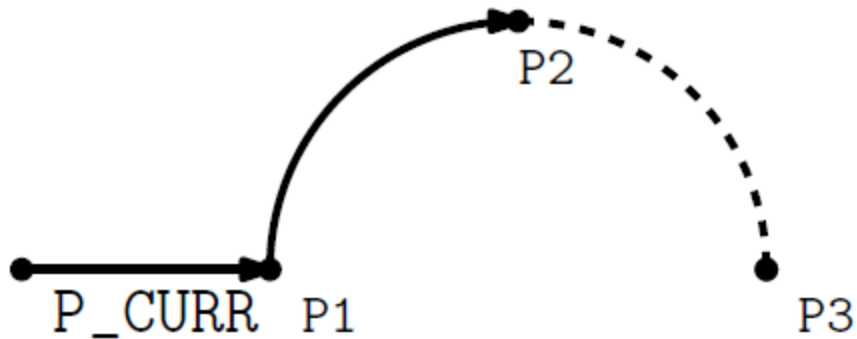


Рис. 3.3.4. Ілюстрація використання команди MVR2

У третій команді MVR3 також прописані три параметри P1, P2 та P3. При виконанні цієї команди робот із поточного положення прямолінійно (або наближено прямолінійно) переміщується в точку P1, далі рухається траєкторією напівкола, що охоплює точки P1 та P2, а точка P3 є центром радіусу цього напівкола (рис. 3.3.5). Синтаксис команди MVR3 аналогічний попереднім MVR та MVR2:

MVR3 P1,P2,P3

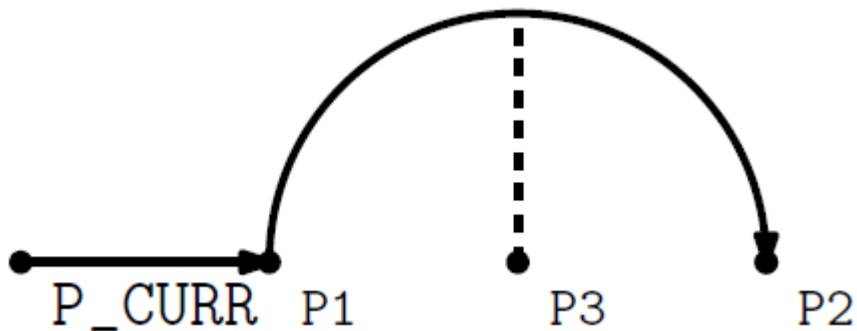


Рис. 3.3.5. Ілюстрація використання команди MVR3

Четверта команда MVC, яка, як і попередні, містить три параметри P1, P2, P3, призначена для руху робота по повному колу:

MVC P1,P2,P3

При виконанні команди MVC робот із поточного положення прямолінійно переміщується в точку P1, після чого рухається по повному колу, яке охоплює усі три точки P1, P2 та P3 (рис. 3.3.6).

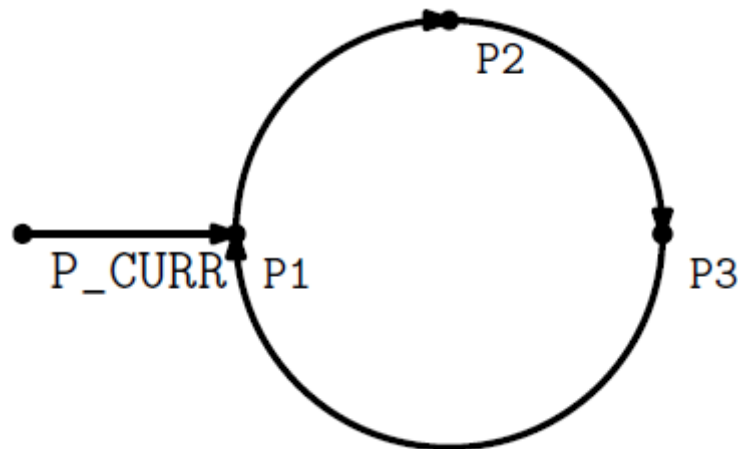


Рис. 3.3.6. Траєкторія руху робота при виконанні команди MVC

Вочевидь, синтаксис наведених вище команд для улаштування різних видів руху робота або маніпулятора необхідно мати *доступну для вимірювання систему координат*. Необхідні координати положення можуть бути виміряні:

- в ручному режимі шляхом підведення інструмента робота до необхідної точки;

- у системах *автоматизованого проектування і розрахунку (САПР)* при наявності просторової моделі робочої області.

Формування складних траєкторій

Складні траєкторії переміщення робота формуються за допомогою поєднання кількох простих ділянок із використанням наведених вище команд. При цьому кінець поточної ділянки є початком наступної, і можуть використовуватись однакові змінні положення для позначення границь траєкторій.

На рис. 3.3.7 зображений приклад траєкторії, що складається з послідовно з'єднаних (1) дуги, (2) прямої, (3) дуги, (4) прямої, (5) дуги, (6) прямої лінії. Отже, траєкторія починається з дуги (MVR P2, P3, P4), кінець якої є початком для прямої лінії (MVS P5), яка переходить в дугу (MVR P5, P6, P7), тобто точка P5 є спільною для обох команд MVS і MVR, і т.д.

Програмний код, який описуватиме рух робота по вказаній траєкторії (рис. 3.3.7), виглядає наступним чином:

```
MVR P2, P3, P4  
MVS P5  
MVR P5, P6, P7  
MVS P8  
MVR P8, P9, P10  
MVS P11
```

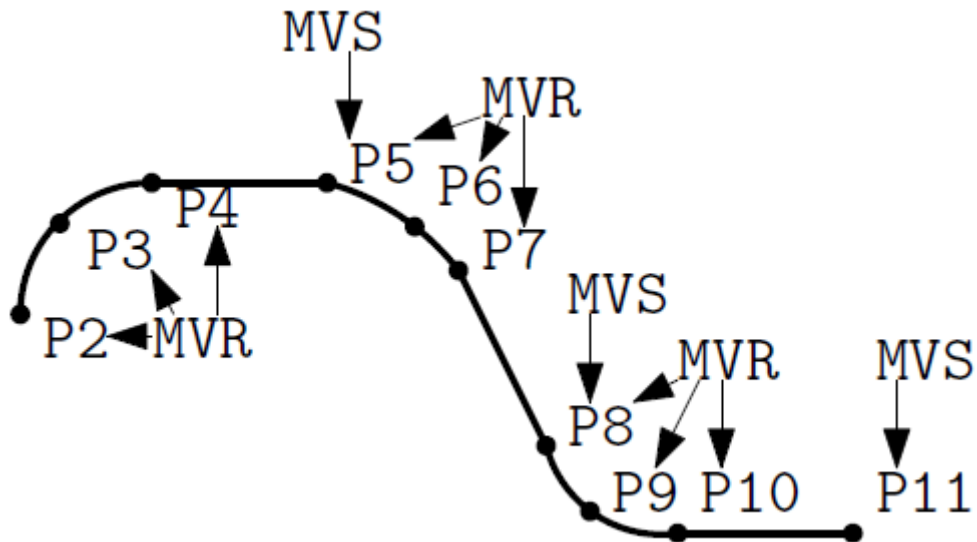


Рис. 3.3.7. Приклад складної траєкторії

3.5.3. Взаємодія із зовнішніми пристроями

Взаємодія із зовнішніми пристроями здійснюється за допомогою під'єднання до плат розширення. Плати розширення виготовляються для широкого спектру пристроїв робототехніки, зокрема для керування *портами введення-виведення, мережевих плат, пристроями захоплення (маніпуляторами)*.

Взаємодія із портами введення-виведення

Встановлені плати розширення дають змогу керувати дискретним станом портів введення та виведення. Керування станом портів дозволяє відправляти дискретні команди на зовнішні пристрої або отримувати сигнали від них. Логічна одиниця на дискретному порті введення або виведення відповідає напрузі близько $U_1 = 24 \text{ В}$, логічний нуль відповідає нульовій напрузі $U_0 = 0 \text{ В}$.

Зчитування значень стану дискретних портів введення здійснюється за рахунок звернення до *системної змінної* M_IN із подальшим вказанням у дужках номеру порту. Значення системної змінної може бути використаним у логічних конструкціях (затримках, умовах, циклах із попередньою умовою). Перед оголошенням команди зчитування стану порту може знадобитися пауза для зупинки руху робота і коректної обробки сигналу, як це показано у прикладі:

```
DLY 0.1  
IF M_IN(2) = 0 THEN  
    . . .
```

Відправка дискретного сигналу на порт виведення здійснюється після створення змінної, яка містить назву порту і системний номер:

```
DEF IO MAGN = Bit,8  
DLY 0.1  
MAGN = 1  
DLY 0.1
```

де MAGN – назва змінної; 8 – номер порту.

Вказівка значення на порті виведення здійснюється за допомогою присвоєння змінної логічного значення MAGN = 1. Наведений приклад ілюструє, що інколи корисно супроводжувати паузами команду вказівки сигналу на дискретному порті виведення.

Керування пристроями захоплення

Важливим елементом промислових роботизованих систем є *пристрої захоплення*, які у загальному випадку закріплюються на встановлювальний фланець промислового робота для утримання та переміщення різноманітних деталей або інструментів, із якими робот здійснює технологічні операції.

За принципом дії пристрої захоплення поділяються на:

- електромеханічні;
- гідравлічні та пневматичні;
- вакуумні та магнітні;
- пристрої еластичної деформації.

Дія *електромеханічних пристроїв захоплення* (рис. 3.3.8) відбувається за рахунок керування встановленою системою електроприводу (набором електродвигунів із редукторами або сервоприводів, крокових двигунів і т.п.). Ці пристрої можуть мати декілька ступенів свободи. Для надійного утримання предметів складної форми їх конструкція передбачає можливість точного керування положенням «пальців» (фіксуєчих елементів)

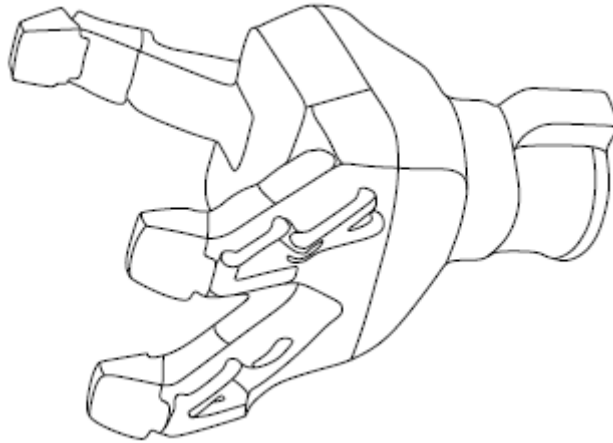


Рис. 3.3.8. Електромеханічний пристрій захоплення

Гідравлічні та пневматичні пристрої захоплення керуються за допомогою відповідно гідравлічних або пневматичних систем. Для таких пристроїв характерні наявність двох крайніх положень (закрито, відкрито), підвищена вантажопідйомність (гідравлічні) та швидкодія (пневматичні), але вимагають встановлення додаткових виконавчих механізмів і регулювальних органів. Додаткове обладнання може бути використане як для регулювання тиску у загальній системі одночасно працюючих кількох роботів, так і для керування пристроєм захоплення окремого робота. На рис. 3.3.9 наведений приклад пневматичного пристрою захоплення.

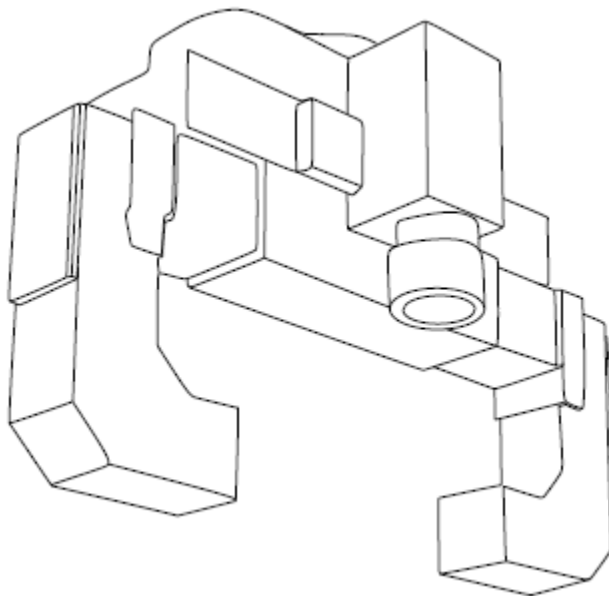


Рис. 3.3.9. Пневматичний пристрій захоплення

Найпоширенішою сферою застосування *вакуумних і магнітних пристроїв захоплення* є перенесення листових матеріалів. Для них характерні два активні стани – утримуючий і неутримуючий. У випадку, коли листовий матеріал має

велику площу поверхні і схильний до згинання, використовуються пристрої захоплення із декількома активними елементами для рівномірного розподілу навантаження по усій площі. На рис. 3.3.10 зображений приклад вакуумного пристрою захоплення із чотирма активними елементами.

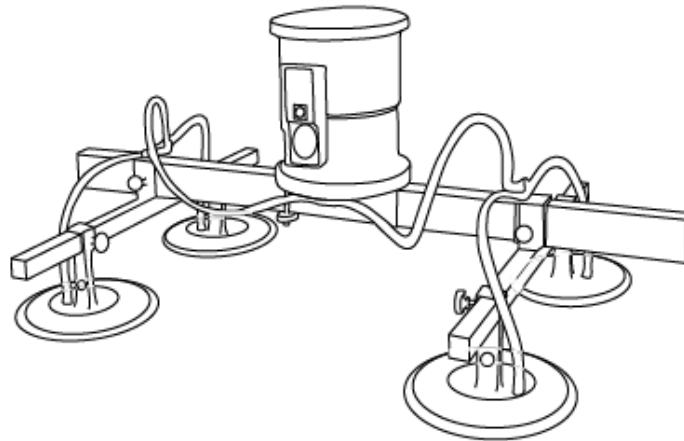


Рис. 3.3.10. Вакуумний пристрій захоплення

Пристрої захоплення еластичної деформації (рис. 3.3.11) застосовуються в особливих випадках для утримання крихких об'єктів без стискання. Як «пальці» у таких пристроях використовуються еластичні елементи, що під дією тиску пневматичної системи можуть деформуватися, забезпечуючи тим самим утримання охопленням крихких об'єктів. Характерною особливістю таких пристроїв є низька вантажопідйомність.

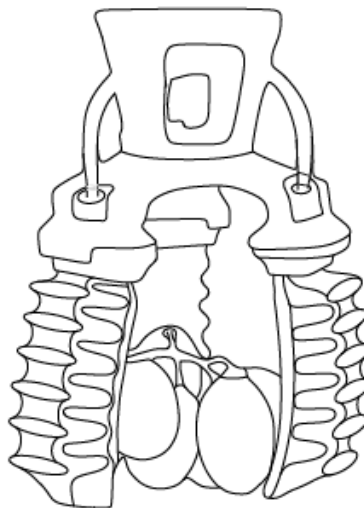


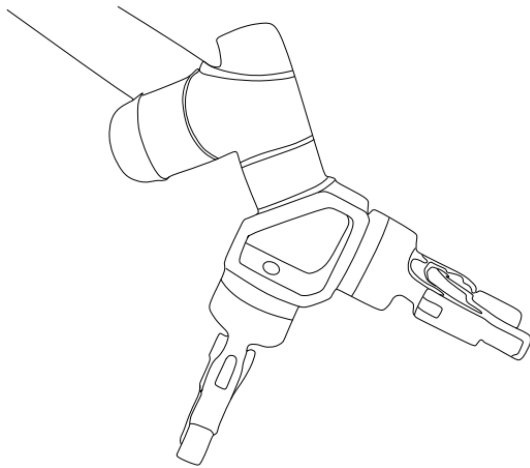
Рис. 3.3.11. Пристрій захоплення еластичної деформації

Плата керування пристроями захоплення (маніпуляторами) реалізує дискретний контроль положення. Часто виробники реалізують можливість

підключення до однієї плати розширення одразу декількох пристроїв. Базова конфігурація передбачає задання двох крайніх положень: "відкрито" і "закрито". *Відкриття* пристрою захоплення оголошується командою

НOPEN 1

де через пробіл вказується *обов'язковий* параметр із номером пристрою захоплення, навіть у випадку, якщо на робот у даний час встановлений лише один маніпулятор. Приклади простого і складного варіантів декількох пристроїв захоплення на один робот представлені на рис. 3.3.12.



а)



б)

Рис. 3.3.12. Варіанти встановлення двох пристроїв захоплення:

а – два пристрої на одному шарнірі; б – два маніпулятори, кожний з яких встановлений на окремому агрегаті, що складається із кількох шарнірів, на одному роботі

Закриття пристрою захоплення оголошується командою

HCLOSE 1

Зважаючи на технічні характеристики роботів, зокрема на можливість проектованого робота переміщувати робочі органи із високою швидкістю, до команд взаємодії із пристроєм захоплення необхідно додавати команди пауз. У наведеному нижче прикладі пауза до команди закриття потрібна для того, щоб робот встиг зупинитися у необхідній точці після виконання попередньої команди переміщення. Пауза після закриття необхідна для того, щоб пристрій захоплення встиг виконати необхідні механічні рухи до початку виконання команди переміщення до нової точки. Часто 0,1 сек. достатньо для виконання таких операцій, в іншому випадку тривалість пауз визначають експериментально.

MOV P1

DLY 0.1

```
HCLOSE 1
DLY 0.1
MOV P2
```

Нагадаємо, що команда **DLY** оголошує паузу в секундах.

Керування мережевими інтерфейсами

Для взаємодії робота із мережевими інтерфейсами спочатку необхідно відкрити їх за допомогою такої команди:

```
OPEN "COM1:" AS #1
```

де **COM1** – номер інтерфейсу в конфігурації контролера робота;

#1 – локальний короткий номер інтерфейсу всередині алгоритму.

Тип інтерфейсу, що відкривається, залежить від системних налаштувань у пам'яті контролера. Подальше звернення до інтерфейсу всередині алгоритму буде відбуватись за коротким номером. Закрити окремий інтерфейс можна за його номером із командою:

```
CLOSE #1
```

або закрити одразу усі інтерфейси

```
CLOSE
```

Відправлення повідомлень у відкритий інтерфейс здійснюється за допомогою команди **PRINT**, після якої через пробіл прописується короткий номер інтерфейсу та значення, що відправляються у пакеті. Зазначати для відправки можна як імена змінних, так і прямі значення. Наведемо приклад застосування команди відправки повідомлення:

```
PRINT #1,"OUTPUT TEST",M1,P1
```

де **#1** – короткий номер інтерфейсу;

"OUTPUT TEST" – значення, які безпосередньо відправляються;

M1,P1 – змінні, які відправляються.

Вхідні повідомлення від інтерфейсу зчитуються командою **INPUT**, після якої через пробіл зазначається короткий номер інтерфейсу, і через кому змінні, у які зберігаються зчитані значення:

```
INPUT #1,IX, IY
```

Під час використання команди **PRINT** виконання програми призупиняється до того часу, поки у вхідному буфері не з'явиться повідомлення. При цьому, якщо

вхідне повідомлення раніше використання команди PRINT, воно залишиться у буфері, доки виконання алгоритму не дійде до команд обробки.

У мові програмування Melfa Basic передбачене зчитування вхідних повідомлень одразу за перериванням. *Переривання за вхідним повідомленням* створюється за допомогою такої команди:

```
ON COM(1) GOSUB *RECV
```

де COM(1) – номер підключення для переривання;

GOSUB *RECV – перехід до процедури обробки переривання *RECV за вхідною подією.

Як і для загальних переривань, які розглядалися у попередніх пунктах, існує можливість активувати або деактивувати переривання за вхідним повідомленням:

COM(1) ON – активується перериванням за вхідним повідомленням;

COM(1) OFF – деактивується переривання;

COM(1) STOP – призупиняється переривання до наступної команди.

Наведемо приклад, що ілюструє використання команд обробки повідомлень мережевих інтерфейсів:

```
OPEN "COM1:" AS #1
ON COM(1) GOSUB *RECV
COM(1) ON
MOV P1
COM(1) STOP
MOV P2
...
COM(1) OFF
CLOSE #1
END

*RECV
    INPUT #1,M0001
    OUTPUT #1,P0001
RETURN 1
```

3.6. Навчальний комплект LEGO Mindstorms NXT 2.0.

3.6.1. Складові частини комплекту Lego Mindstorms NXT 2.0. Технологія NXT

NXT є інтелектуальним роботом на базі елементів LEGO і системи MINDSTORMS, керування яким здійснюється за допомогою комп'ютера.

Система MINDSTORMS отримала свою назву завдяки книзі Сеймура Пейперта "Переворот у свідомості: Діти, комп'ютери та продуктивні ідеї", у якій автором аналізуються способи формування дитячого мислення і роль комп'ютерів у цьому процесі.

Основою комплекту Lego Mindstorms NXT 2.0 є **програмований блок NXT**, інтерактивні **серводвигуни** та декілька **давачів (сенсорів)**. Комплект 8547 складається з:

- двох давачів натискання – кнопок;
- давач кольору – RGB-давач, за допомогою якого робот розрізняє кольори і вимірює яскравість світла оточуючого середовища;
- ультразвуковий давач – "очі" робота, що дозволяють йому вимірювати відстань до об'єкта.

До складу навчального комплекту входить також давач звуку, який вимірює рівень гучності звуку, а замість RGB-давача наявний давач освітленості. Існують і інші сенсори, які не входять до складу комплекту і реалізуються окремо, наприклад, для вимірювання температури, рН середовища, гіроскопічний давач, компас та ін.

Усі давачі та серводвигуни підключаються до NXT-блоку через порти введення та виведення за допомогою шестипроводних кабелів чорного кольору. Давачі приєднуються до портів введення під номерами 1 - 4, а серводвигуни – до портів виведення, які на блоці NXT мають назву А, В, С.

Виробник рекомендує для підключення сенсорів і серводвигунів використовувати стандартні порти (рис. 3.4.1):

- Порт 1: Давач дотику
- Порт 2: Давач дотику
- Порт 3: Давач кольору
- Порт 4: Ультразвуковий давач
- Порт А: Серводвигун для додаткових функцій
- Порт В: Серводвигун для переміщення
- Порт С: Серводвигун для переміщення

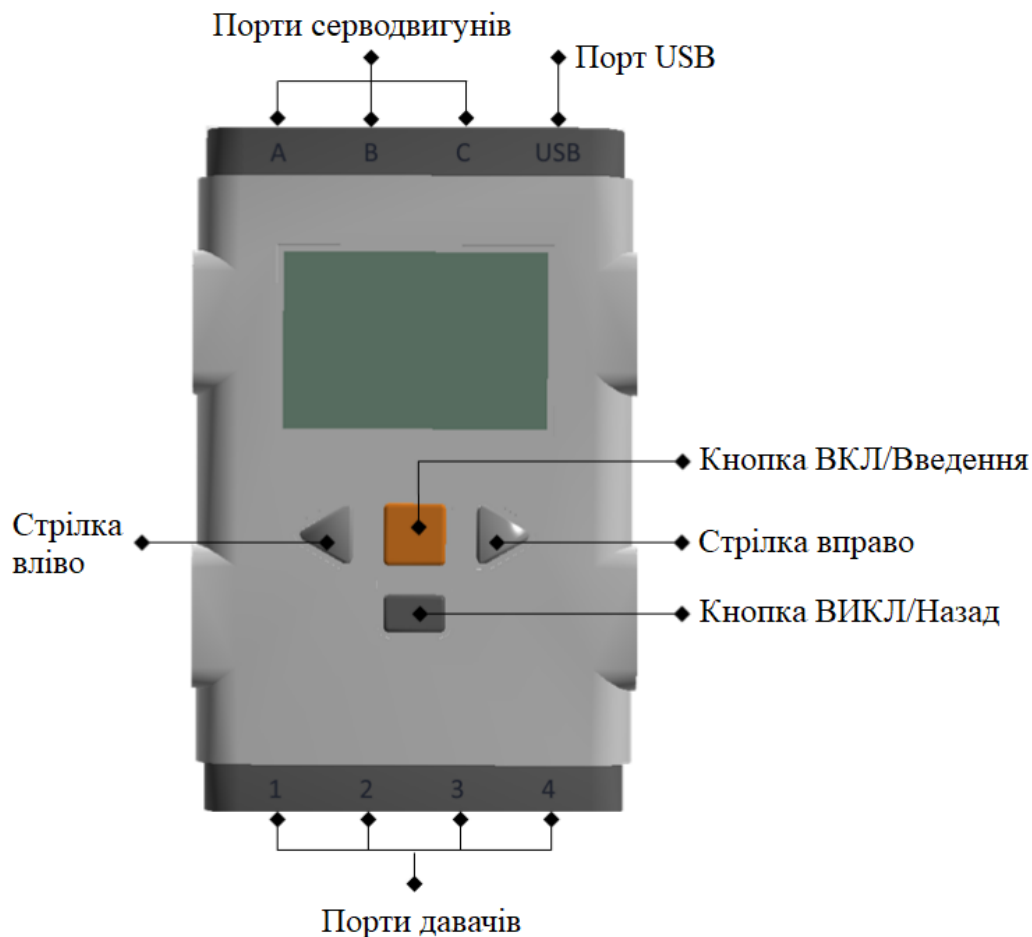


Рис. 3.4.1. Зовнішній вигляд блоку NXT і його роз'єми

У даному навчальному курсі ми дотримуємось цих рекомендацій. Насправді сенсори можна підключати до довільних портів. Дотримуватися стандартних портів обов'язково необхідно при роботі із меню **Try Me** (див. нижче).

Блок NXT працює від шести елементів живлення типу AA. До складу навчальної версії набору входить акумуляторна батарея.

Робота із меню NXT здійснюється за допомогою чотирьох кнопок (рис. 3.4.1):

- Сірі трикутники – кнопки "вперед" (вправо) і "назад" (вліво) – дозволяють переміщуватися всередині меню до необхідного пункту;
- Помаранчевий квадрат відповідає кнопці вводу; вона ж використовується для включення робота;
- Темно-сірий прямокутник – "відміна" або перехід назад до попереднього пункту. Ця ж кнопка використовується для відключення робота: для цього необхідно натискати кнопку до того часу, коли на екрані не з'явиться надпис **Turn off?**, після чого для підтвердження відключення необхідно натиснути помаранчеву кнопку (рис. 3.4.2). Також можна натиснути і утримувати темно-сіру кнопку до повного вимкнення NXT.

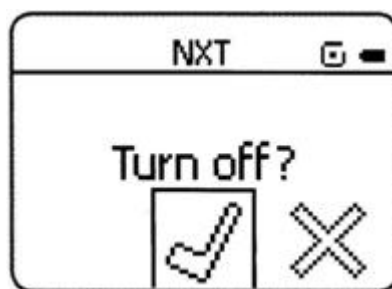


Рис. 3.4.2. Екран відключення блоку NXT

Завдання 1.1. Навчіться включати та відключати свій NXT. Спробуйте відключити NXT двома способами, які були описані вище.

Уся інформація на дисплеї NXT відображається англійською мовою. При включенні екран NXT має вигляд, який зображений на рис. 3.4.3:

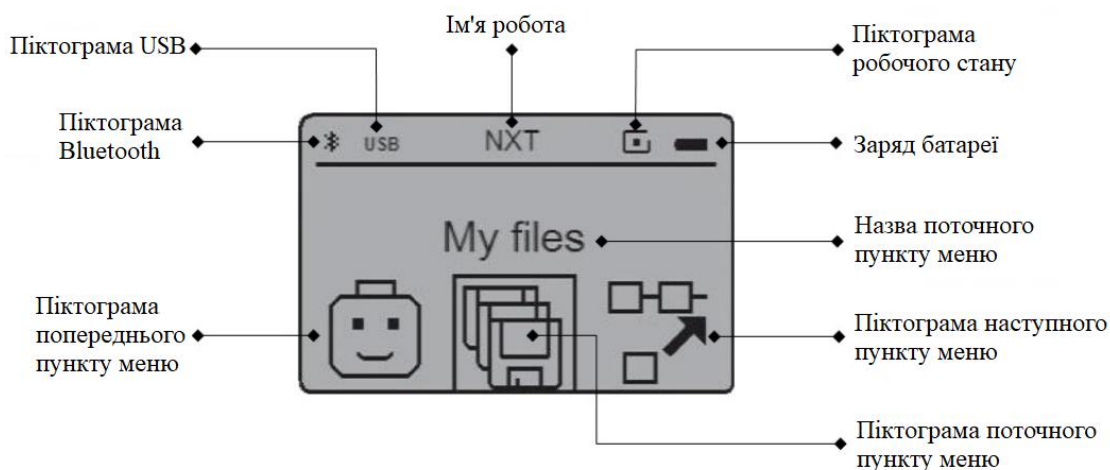







Рис. 3.4.3. Екран блоку NXT після включення

-  Bluetooth увімкнений, але NXT не розпізнається іншими пристроями Bluetooth.
-  Bluetooth увімкнений, NXT розпізнається іншими пристроями Bluetooth.
-  Bluetooth увімкнений, NXT підключений до пристрою Bluetooth.
-  USB підключений і працює нормально.
-  USB підключений, але наявні несправності

Якщо NXT працює нормально, то іконка робочого стану обертається. Зупинка обертання іконки означає, що NXT "завис" і вимагає перезавантаження.

За замовчуванням роботу надане ім'я NXT. Ім'я можливо змінити (як це зробити – описано в п. 3.4.2).

Завдання 1.2. Увімкніть NXT. Ознайомтеся із інформацією, яка відображається на екрані NXT. Переміщуйтеся по головному меню NXT за допомогою стрілок. Відключіть NXT.

Розглянемо структуру меню NXT.

- **My Files (Мої Файли).** Місце, де зберігаються усі файли NXT.
 - **Software files (Файли програм).** Місце, де зберігаються скопійовані файли програм, що завантажені користувачем. Із цього місця здійснюється запуск програм на виконання.
 - **NXT files (Файли NXT).** Службові файли NXT.
 - **Sound files (Звукові файли).** Місце для зберігання різноманітних звукових файлів, як вбудованих, так і записаних користувачем (користувач може записати власні звуки для їх подальшого використання у програмах).
 - **Datalog files (Файли даних).** Тут містяться дані, що зібрані NXT за допомогою меню **NXT Datalog**. Дані можна зчитувати під час роботи програми або передавати по Bluetooth іншим пристроям.
- **NXT Program (Програма NXT).** Цей розділ дозволяє програмувати NXT прямо на блоці, без допомоги комп'ютера. Можна створювати короткі програми із п'яти команд (не більше п'яти). Зручний для початкового знайомства із можливостями NXT. Однак більш складні задачі у цьому розділі не вирішуються.
- **NXT Datalog (Дані NXT).** Отримує, відображає на екрані та записує дані від зовнішніх пристроїв (давачів) у файли, які у подальшому зберігаються у меню **Datalog files**.
- **View (Огляд).** Дозволяє відображати на екрані дані, отримані від зовнішніх пристроїв. Але, на відміну від **NXT Datalog**, не записує ці дані у файли. Може використовуватися для тестування і калібрування давачів.
- **Bluetooth.** Дозволяє створити канал безпроводного зв'язку між NXT та іншими пристроями із підтримкою Bluetooth. Призначений для завантаження програм без допомоги USB-кабелю, обміну програмами з іншими NXT, дистанційного керування іншими NXT (не більше трьох) або керування NXT за допомогою інших пристроїв, наприклад, мобільного телефону.
- **Settings (Налаштування).** Розділ налаштування різноманітних параметрів блоку NXT.
 - **Volume (Гучність).** Встановлює гучність динаміків.
 - **Sleep (Сон).** Налаштовує параметри автоматичного відключення при простої.
 - **NXT Version.** Дозволяє отримати інформацію про поточні версії програмних і апаратних засобів.
 - **Delete files (Видалення файлів).** Призначений для видалення файлів.
Будьте обережні: видаляє усі файли із обраного каталогу!

- Try Me (Спробуй). Після підключення датчиків і моторів до відповідних портів, можна виконати декілька готових програм та ознайомитися із можливостями конструктора.

Приклад 1.1. Налаштуємо NXT таким чином, щоб він автоматично відключався через дві хвилини простою. Для цього виберемо із головного меню Settings\Sleep, після чого за допомогою стрілок обираємо необхідне значення (поточне значення відображається на екрані) і натискаємо на помаранчеву кнопку введення. Якщо тепер не працювати із NXT протягом двох хвилин, він вимкнеться сам. Зверніть увагу на варіант Never (Ніколи): у цьому режимі NXT не буде включений до того часу, доки користувач сам цього не зробить. Але це може призвести до більш швидкого розрядження батареї.

Приклад 1.2. Спробуємо відтворити звукові файли на блоці NXT. Головне меню\My Files\Sound files\Good Job. У вікні відображається ім'я обраного файлу і варіанти роботи з ним, які можна переглянути за допомогою стрілок:

- Run – запустити обраний файл на виконання (піктограма робочого стану). Якщо обрати цей варіант, то звуковий файл буде відтворений. У процесі роботи файлу на екран буде виведене повідомлення Running, а після завершення – Done.
- Delete – видалити обраний файл (піктограма корзини). Якщо обрати цей варіант, з'явиться повідомлення із попередженням Are you sure? (Ви впевнені?). Варіант за замовчуванням - піктограма у формі хрестика (Ні). В тому випадку, якщо обраний файл все ж необхідно видалити, то за допомогою стрілки слід обрати піктограму із галочкою (Так).
- Send – переслати (піктограма листа). Для того, щоб переслати файл іншим пристроєм, слід спочатку налаштувати канал зв'язку Bluetooth.

Завдання 1.3. Встановіть максимальну гучність динаміків. Перевірте гучність, відтворивши довільний звуковий файл NXT. Встановіть комфортну для себе гучність.

Завдання 1.4. Визначіть різні варіанти автовідключення NXT. Встановіть таймер автовідключення на 10 хв.

Завдання 1.5. Протестуйте давачі дотику, ультразвуку, серводвигун за допомогою меню Try Me (не слід забувати про стандартні порти підключення давачів).

Робота з іншими пунктами меню буде розглянута в подальшому матеріалі за необхідністю.

3.6.2. Середовище програмування NXT-G

Запускаємо програмне забезпечення Mindstorms NXT 2.0. У вікні, що відкрилося (рис. 3.4.4) можна переглянути відео "Getting Started Guide" (Керівництво до швидкого старту) та "Software Overview" (Огляд програмного забезпечення).

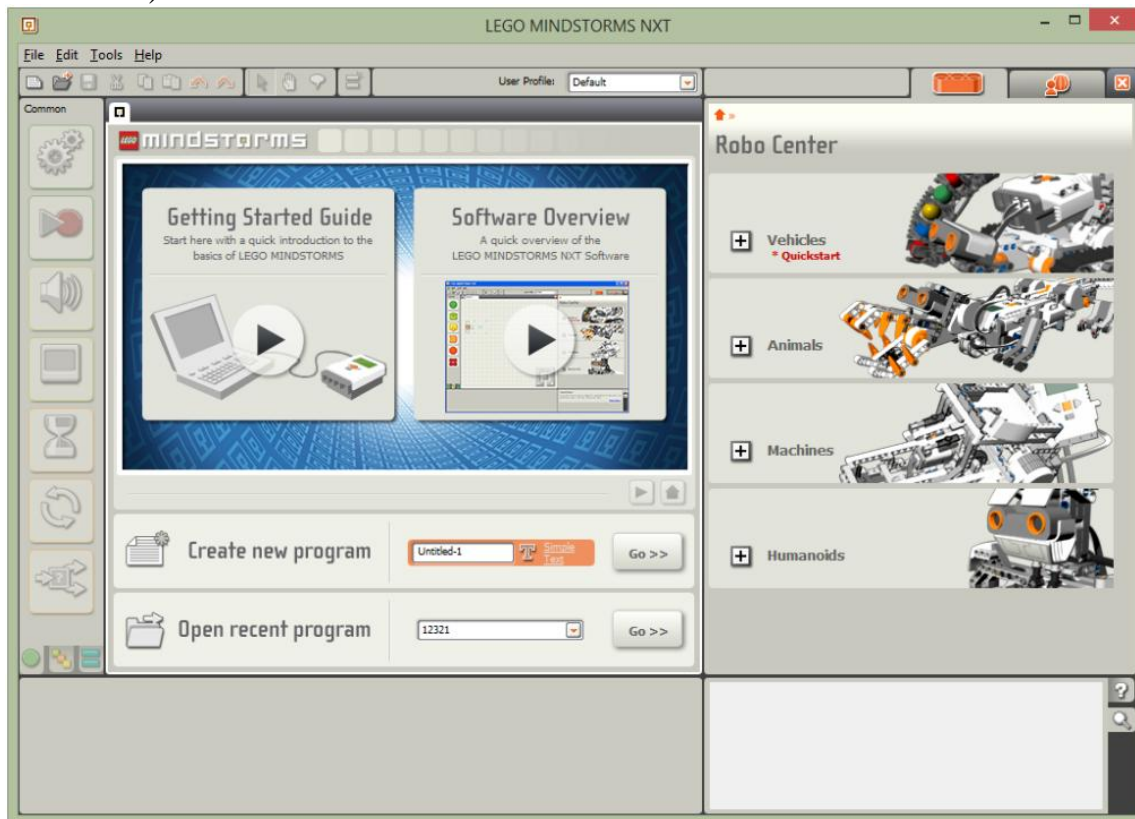


Рис. 3.4.4. Вікно середовища програмування NXT-G після запуску

3.6.2.1. Профілі

Якщо при розробці програм на NXT-G один комп'ютер використовується декількома користувачами, зручно створити для кожного розробника свій унікальний профіль, у якому будуть зберігатися програми тільки цього розробника. Після встановлення Lego Mindstorms NXT 2.0 автоматично створюється один профіль з ім'ям **Default** (за замовчуванням) (рис. 3.4.5).



Рис. 3.4.5. Профіль за замовчуванням

Приклад 1.3. Створимо новий профіль **Student**. Для цього виберемо пункт меню **Edit\Manage Profiles**. У вікні, що відкрилося (рис. 3.4.6), послідовно

натискаємо на кнопку **Create** (за замовчуванням для нового профілю буде запропоноване ім'я **Profile-1**), заповнюємо поле **Name** і натискаємо кнопку **Close**.

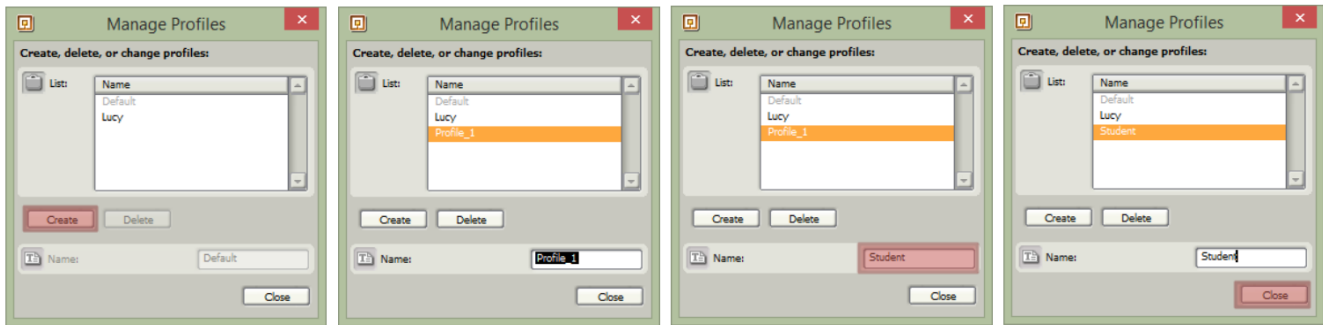


Рис. 3.4.6. Створення власного профілю

Тепер можна вибрати свій унікальний профіль зі списку (рис. 3.4.7):

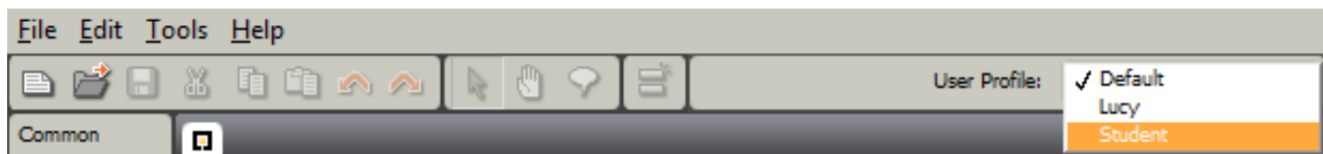


Рис. 3.4.7. Вибір власного профілю

Завдання 1.6. Видаліть профіль **Student** і створіть профіль із власним ім'ям. Перед подальшою роботою оберіть свій власний профіль.

3.6.2.2. Створення і відкриття програм

Інтерфейс середовища користувача за замовчуванням встановлює для першої нової програми ім'я **Untitled-1** (Безіменна-1). У подальшому номер кожної нової програми, створеної користувачем, автоматично збільшується на одиницю. Однак більш доцільним є використання для власних програм унікальних характерних імен.

У полі **Create new program** (Створити нову програму) введіть **Hello!** і натисніть кнопку **Go** (Вперед).

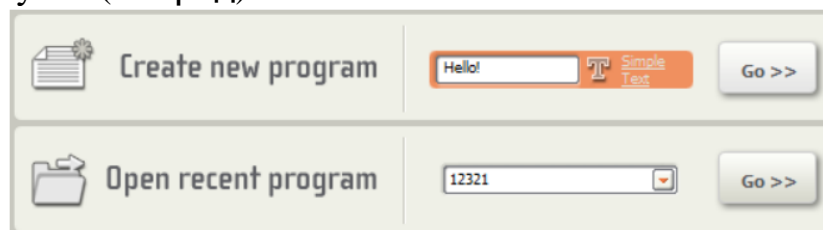


Рис. 3.4.8. Створення або відкриття програми

Увага! Екран **NXT** відображає текст виключно англійською мовою, тому використання символів кирилиці для створення імені файлу є неприпустимим.

У подальшому відкрити вже створену програму можна шляхом вибору її зі списку у полі **Open recent program** (Відкрити існуючу програму) (рис. 3.4.8).

Для створення і відкриття файлів можна використовувати також і стандартні засоби – комбінації клавіш **Ctrl+N** і **Ctrl+O** відповідно або скористатися меню **File**.

Вкладки у лівій верхній частині вікна середовища дозволяють відкрити **Robo Center** і **My Portal** (рис. 3.4.9). Ці можливості ми залишаємо за межами цього посібника і рекомендуємо читачу самостійно з ними ознайомитися. Для збільшення області програмування можна просто натиснути на хрестик.

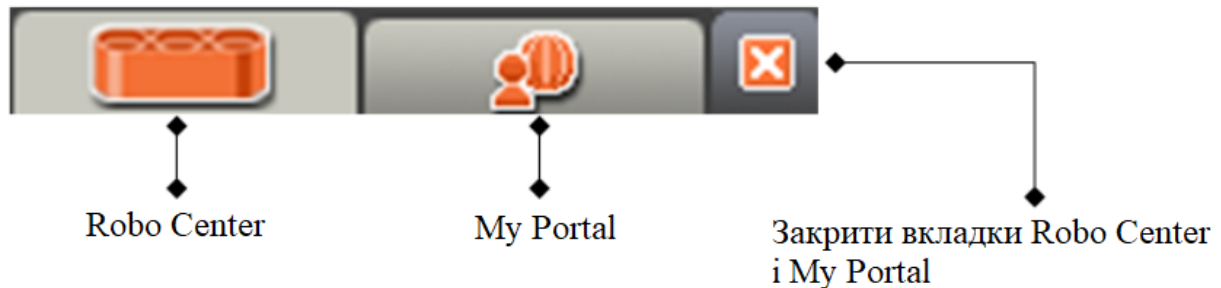


Рис. 3.4.9. Доступ до Robo Center і My Portal

При виникненні необхідності ці вкладки завжди можна ввімкнути.

3.6.2.3. Доступ до програмних блоків

В середовищі **NXT-G** реалізований візуальний спосіб проектування програм, що дуже зручно для навчання. Програма складається із блоків. Кожен блок представляє різні типи дій. Окремі блоки можна налаштовувати. Усі блоки організовані і представлені у трьох палітрах програмування (рис. 3.4.10) - "**Common**" (Загальна), "**Complete**" (Повна) и "**Custom**" (Палітра користувача). Одночасно можна працювати тільки з однією палітрою.

Блоки мають різне забарвлення, яке залежить від функціонального призначення блоку. Блоки, функціональне призначення яких є подібним, мають однаковий колір. Наприклад, усі блоки, які відповідають за виконання певної дії (**Action**), зображені зеленим кольором, а блоки, які відображають поточне значення давачів – жовтим кольором. Кольорове забарвлення дозволяє легко знаходити необхідні блоки у палітрі.

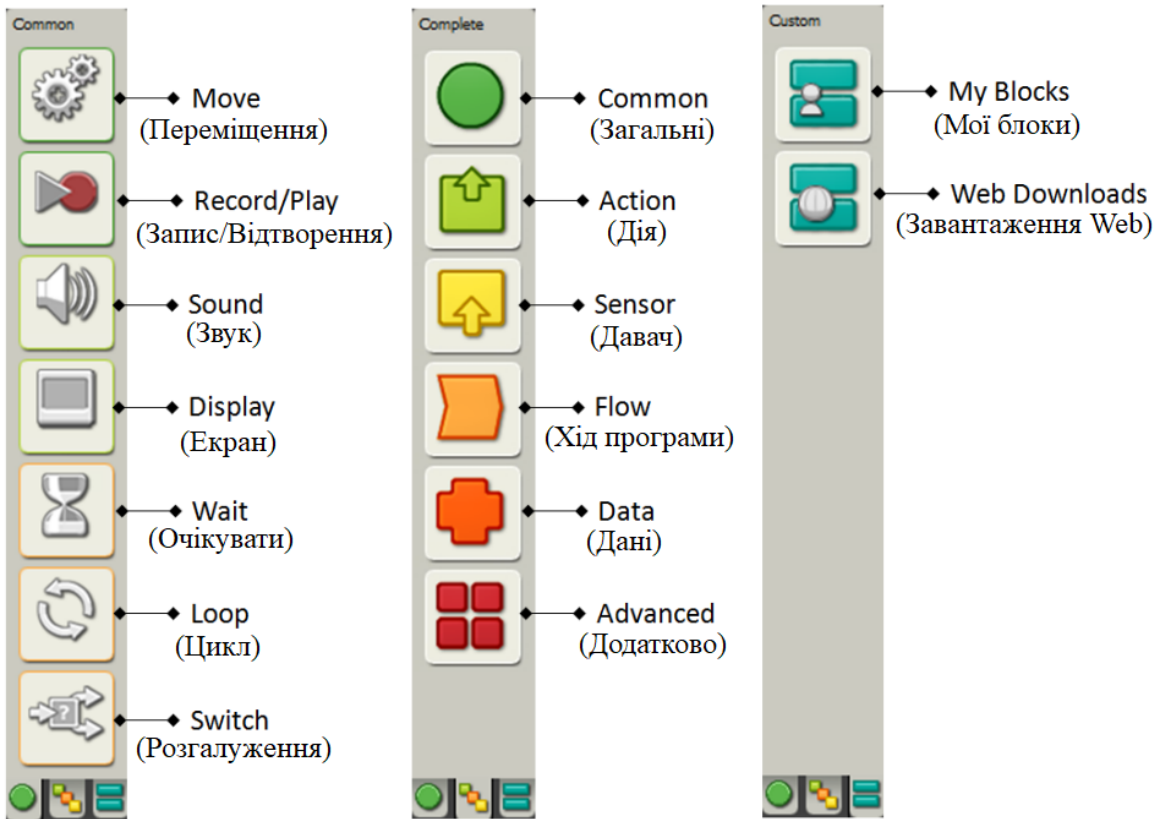


Рис. 3.4.10. Палітри блоків NXT-G

Блоки, що використовуються найчастіше, згруповані у палітрі "Common". Кожна піктограма цієї палітри є окремим програмним блоком, окрім піктограми **Wait**, яка показує шість варіантів цього блоку (рис. 3.4.11):

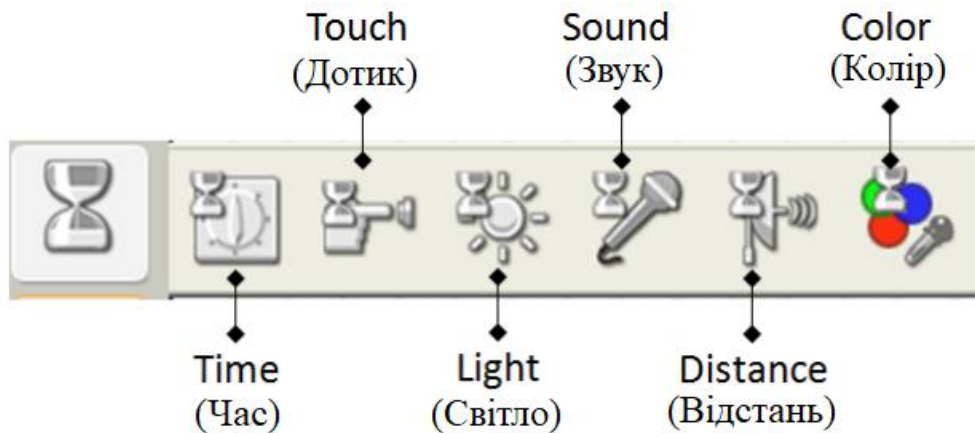


Рис. 3.4.11. Список варіантів блоку **Wait**, що випадає при виділенні

Будь-який із варіантів блоку **Wait** можна отримати шляхом налаштування блоку.

Завдання 1.7. Вивчіть спливаючі підказки блоків палітри "Common" за допомогою наведення на них курсору миші.

Повна палітра відкриває доступ до 39 блоків, які організовані у шість категорій. Перша категорія повторює загальну палітру. Із рештою категорій ми ознайомимося пізніше на конкретних прикладах. Палітра користувача складається з блоків, які користувачі завантажили і створили самостійно (робота із цією палітрою розглядається у подальшому матеріалі).

3.6.3. Перша програма

Продовжимо створення програми **Hello!** і тим самим завершимо огляд інтерфейсу середовища NXT-G. З правого боку від палітри програмування знаходиться велика область програмування. Блоки переміщуються мишею із палітри на спеціальну напрямну у вигляді балки Lego, яка збільшується із черговим доданим блоком.



Рис. 3.4.12. Вихідний вузол для побудови програми на NXT-G

Приклад 1.4. Складемо класичну для входження у мову програмування програму, яка буде виводити на екран повідомлення: **Hello, world!**

Додаємо на лист програми блок **Display** із загальної палітри. У нижній частині вікна відображається панель налаштування поточного (виділеного) блоку. За замовчуванням блок налаштований на відображення зображень (**Action: Image**), а саме, смайлика із посмішкою. Заміна картинки можлива у властивості **File**. Властивість **Display** дозволяє очищувати (**Clear**) перед виводом нового зображення.

У даному випадку на екран необхідно вивести текст, тому замінюємо властивість **Action** на **Text** (рис. 3.4.12), замінюємо текст за замовчуванням на **Hello, world!** і обираємо позицію відображення **Line - 5** (можна просто клацнути мишею в необхідному місці властивості **Position** або задати необхідні координати).

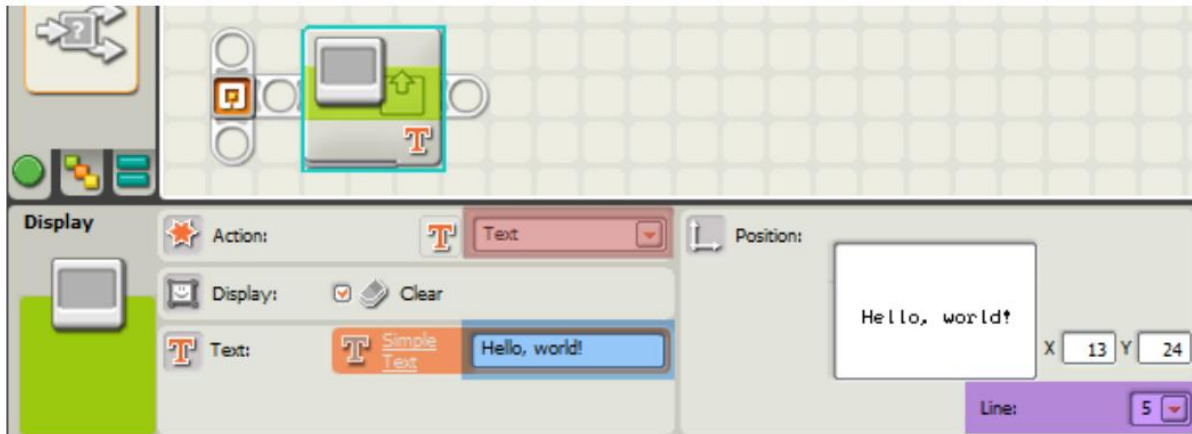


Рис. 3.4.13. Налаштування блоку Display

Зверніть увагу на те, що зміна налаштування блоку змінює і зовнішній вид блоку.

Наша програма вийшла дуже компактною, вона повністю вміщується на екрані. В подальшому ми зустрінемо більш довгий (в прямому сенсі цього слова) код. Для цього, щоб побачити код, який не помістився на екран, можна скористатися курсорними стрілками. Крім того, можна клацнути мишею по області коду у правому нижньому куті екрану (стає доступною, якщо клацнути по вкладці із зображенням лупи).

Для завантаження і запуску програм на блок NXT у середовищі NXT-G використовуються кнопки керування, іншими словами, контролер (рис. 3.4.14):

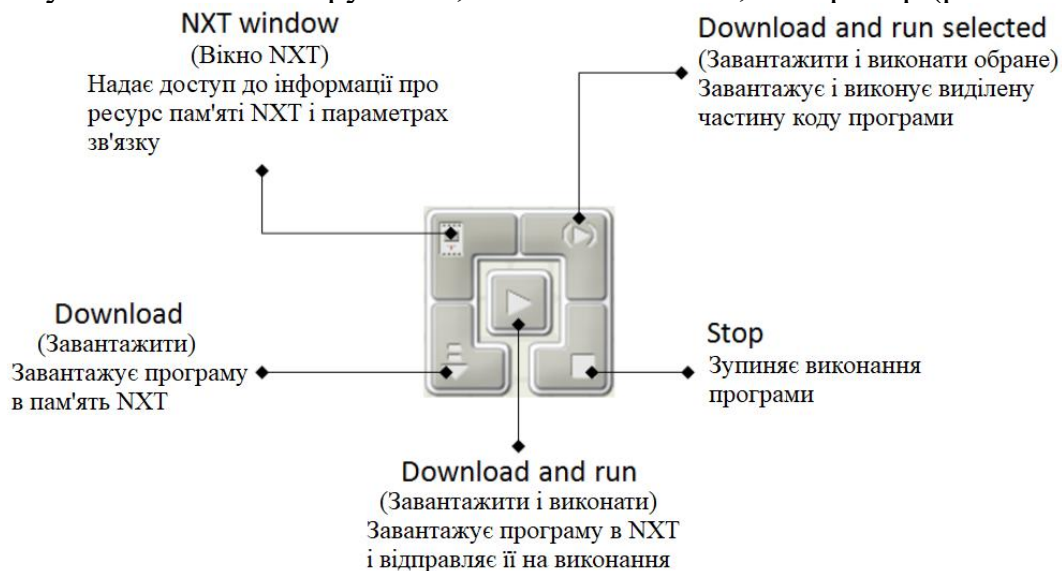


Рис. 3.4.14. Контролер NXT-G

Підключаємо робот USB-кабелем до комп'ютера. Вмикаємо робот натисканням на помаранчеву кнопку. Запускаємо програму на виконання за допомогою центральної кнопки контролера. Що ми бачимо? На екрані майнув надпис Hello! І потім одразу Done. Це означає, що програма Hello! вже завершила

свою роботу, а ми цього так і не побачили. **Зверніть увагу**, що зараз на екрані NXT відображається меню *Software files*.

Для того, щоб встигнути побачити текст на екрані NXT до завершення програми, необхідно використати затримку часу. Додамо в програму новий блок **Wait** і налаштуємо його на відлік двох секунд (рис. 3.4.15):

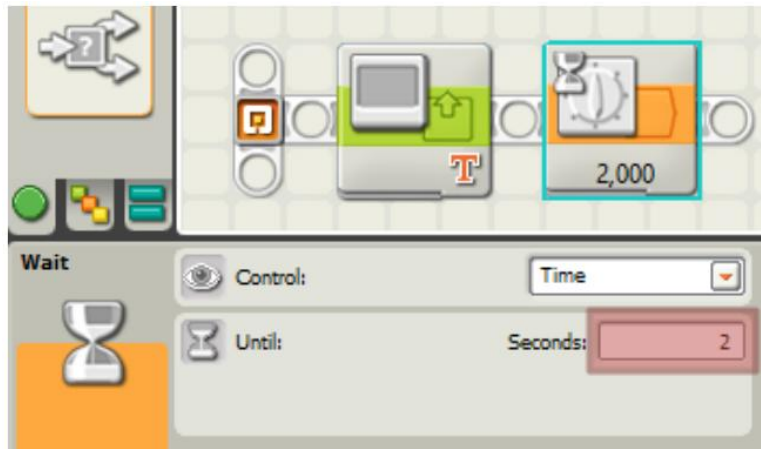


Рис. 3.4.15. Налаштування блоку **Wait**

Ще раз запускаємо програму на виконання за допомогою центральної кнопки контролера. Тепер текст залишається на екрані протягом двох секунд. Оскільки зараз на екрані NXT активний надпис **Run**, то можна повторити виконання програми натисканням на помаранчеву кнопку вводу.

Завдання 1.8. Створіть нову програму **Eyes**, у якій на екрані NXT будуть відображені очі, які споглядають вліво-вправо. Використовуйте файли **Looking left** и **Looking right** для імітації погляду вліво/вправо. Зміна напрямку погляду повинна здійснюватися чотири рази.

Завершимо тему присвоювання нового ім'я Вашому NXT-блоку.

Приклад 1.5. Зміна ім'я NXT. Натисніть на кнопку NXT Window контролера. У результаті відкриється вікно, права частина якого NXT Data містить наступну інформацію (рис. 3.4.16):

- Name (Ім'я)
- Battery (Заряд батареї)
- Connection (З'єднання)
- Free Storage (Вільна пам'ять)
- Firmware Version (Версія програмного забезпечення, так званої "прошивки")

Введіть у поле **Name** нове ім'я Вашого NXT і натисніть на кнопку вводу поруч із цим полем. На екрані NXT автоматично відобразиться нове ім'я.

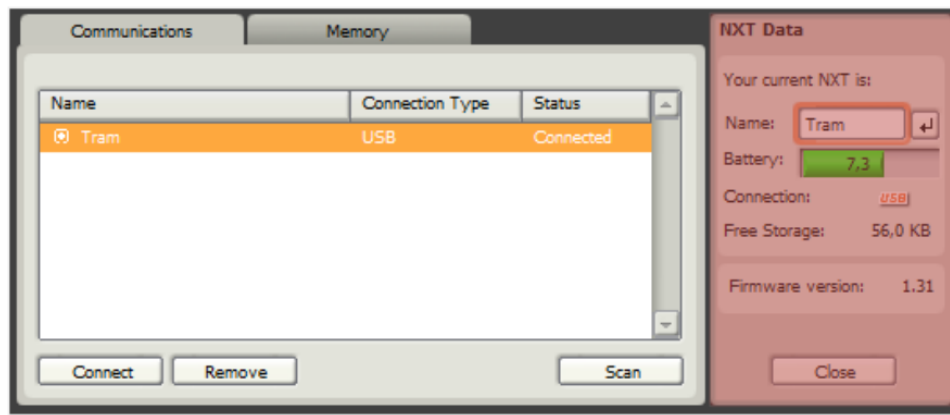


Рис. 3.4.16. Вікно NXT

3.6.4. Програмування серводвигунів: команда Move

Однією із основних функцій робота є рух (переміщення у просторі). Рухатися може як весь робот як такий, так і окремі його частини. За рух (переміщення) відповідають серводвигуни (або сервоприводи). У комплекті Lego Mindstorms серводвигуни оснащуються давачами обертів, які підраховують кількість градусів повороту головної осі. Важливим компонентом сервомотора є редуктор, який через систему шестерней перетворює надзвичайно швидкий обертальний рух внутрішнього електродвигуна у більш повільний. Наявність датчика обертів і редуктора дозволяють сервомотору здійснювати прецизійні рухи головної осі. Сервопривод Lego може бути повернутий із точністю 1°. Внутрішнє улаштування сервоприводу зображене на рис. 3.4.17.



Рис. 3.4.17. Сервопривод (зліва) і його внутрішнє улаштування (справа)

Програмування руху (переміщення) відбувається за допомогою блоку Move (Рух), який знаходиться у загальній і повній палітрах (рис. 3.4.18).



Рис. 3.4.18. Блок Move у загальній (Common - зліва) і повній (Complete - справа) палітрах

Блок Move містить велику кількість налаштувань, які дозволяють керувати поведінкою серводвигуна (рис. 3.4.19).

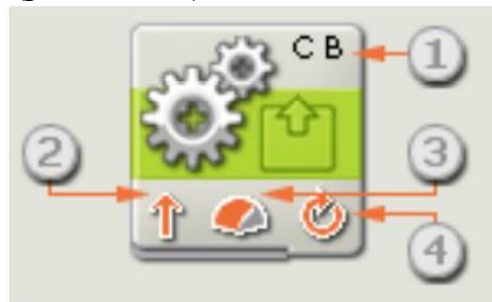


Рис. 3.4.19. Налаштування блоку Move

На рис. 3.4.19 цифрами відмічені:

1. Мотори, якими управляє цей блок.
2. Напрямок обертання моторів.
3. Рівень потужності двигуна (швидкість). ***Зауважимо**, що реальна швидкість робота буде залежати також і від його конструкції, типу поверхні (слизька, шорстка та ін.), нахилу поверхні, маси робота і т.п.*
4. Параметр тривалості руху: без обмеження, в градусах, обертах або секундах.

Налаштування блоку Move, як і інших блоків, здійснюється у нижній частині екрану після вибору блоку (рис. 3.4.20):



Рис. 3.4.20. Панель налаштувань блоку Move

На рис. 3.4.20 цифрами відмічені:

1. Вибір двигунів, якими необхідно керувати (**Port**). Може бути А, В або С. Один блок **Move** може керувати одразу двома серводвигунами (для зупинки – навіть час); на рисунку зображений вибір двигунів В і С. **Зауважимо**, що при створенні колісних або гусеничних роботів краще ліві і праві колеса (гусениці) призначати моторам літери В і С, а допоміжні операції (маніпулятори, поворотні бапти та ін.) призначати мотору літеру А. Це пов'язано із улаштуванням блоку **NXT** і дозволить реалізувати більш точне переміщення робота.
2. Вибір напрямку обертання двигуна/двигунів (**Direction**). Може бути "вперед", "назад" або "стоп".
3. При керуванні двома серводвигунами слугує для повороту: перемістіть слайдер (бігунок) **Steering** вліво або вправо. Для прямолінійного руху встановіть його посередині. Якщо використовувати проміжні значення цього параметра, то робот буде їхати по дугоподібній траєкторії. Чим далі слайдер від центра, тим крутішим буде поворот.
4. **Power** задає рівень потужності у межах від 0 до 100%. Потужність не може виходити за межі цього діапазону.
5. **Duration** – тривалість роботи серводвигуна задається у:
 - **Rotations** – кількості обертів двигуна;
 - **Degrees** – градусах, на які повернеться вал двигуна;
 - **Seconds** – секундах;
 - **Unlimited** – без обмеження. Це дуже важливий параметр; його вибір передбачає, що двигун буде працювати до того моменту, коли відбудеться певна подія. Наприклад, можна запустити двигун і зупинити його тоді, коли робот потрапить на червоне поле.
6. **Next Action** поведінка двигунів після відключення. Тут існують два варіанти:
 - **Brake** – гальмування. В цьому випадку при зупинці двигун негайно припинить своє обертання. На виконання цієї задачі буде витрачена енергія. Зокрема, при постійних різких гальмуваннях батарея розряджається швидше.
 - **Coast** – після команди на зупинку просто вимикається живлення від двигуна і робот котиться по інерції. Цей режим корисно використовувати у випадку, коли точна зупинка не вимагається.
7. Поле зворотного зв'язку. Тут відображається поточне значення повороту двигунів при наявності з'єднання робота із комп'ютером.

Завдання 2.1. Зміна налаштувань.

- Запустіть середовище для програмування роботів Lego Mindstorms NXT.
- Створіть новий файл. Додайте у програму блок **Move**.

- Змінюючи налаштування блоку **Move**, спостерігайте за змінами на зображенні блоку в програмі.
- Не створюючи програму на рис. 3.4.21, прочитайте її. Які налаштування відповідають кожному блоку програми?

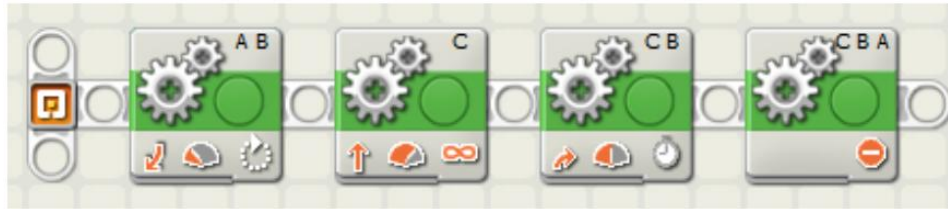


Рис. 3.4.21. Приклад програми до завдання 2.1

Для подальшої роботи нам знадобиться тестова модель робота.

Завдання 2.2. Збирання тестової моделі.

- Складіть робота **Five Minute Bot** за інструкцією: http://www.nxtprograms.com/NXT2/five_minute_bot/index.html
- **Зверніть увагу** на назву портів, до яких підключені двигуни.

Завдання 2.3. Програма пересування робота.

- Створіть програму, яка зображена на рис. 3.4.22.
- Уважно розгляньте налаштування блоку **Move**. Що, на Вашу думку, буде робити робот у відповідність із цією програмою?
- Перевірте власні припущення, завантаживши програму на робот і запустивши її. Чи виконав робот ті дії, які Ви очікували?
- Не вимикаючи USB-кабель обертайте один із моторів рукою і одночасно спостерігайте за показами у полі зворотного зв'язку. У яких одиницях видається інформація у цьому полі?

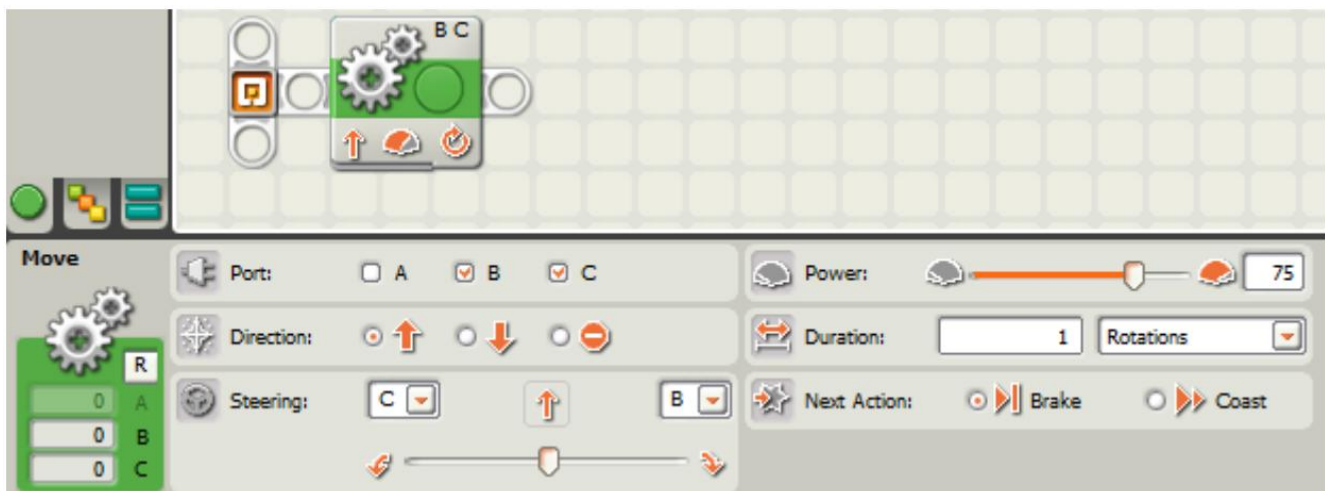


Рис. 3.4.22. Приклад програми до завдання 2.3

Завдання 2.4. Рух одним серводвигуном.

Змініть попередню програму таким чином, щоб робот обертав тільки один із двох підключених моторів. Встановіть тривалість обертання, яка дорівнює восьми обертам. Спостерігайте, як змінюється характер руху робота в залежності від напрямку руху.

Майте на увазі, що довжини USB-кабелю може не вистачити на вісім обертів, тому не забудьте від'єднати його перед запуском програми на виконання!

Заповніть таблицю:

Серводвигун	Напрямок	Характер руху
Лівий	Вперед	
Лівий	Назад	
Правий	Вперед	
Правий	Назад	

При виконанні подальших завдань Вам знадобиться тестове поле, яке входить до складу комплекту із роботом.

Завдання 2.5. Рух двома двигунами.

- Вивчіть доступні параметри тривалості руху, змінюючи їх у полі **Duration**.
- Встановіть **Next Action у Brake** і виконайте вимірювання відстані, на яку переміщується робот при різних значеннях параметра **Duration** и заповніть таблицю:

Кількість	Тип	Шлях (см)	Кількість	Тип	Шлях (см)
1	Seconds		360	Degrees	
2,5	Seconds		720	Degrees	
1	Rotations			Unlimited	
2	Rotations				

Зверніть увагу, що при встановленні тривалості руху в **Unlimited** робот не пересувається нескінченно, як це можна було очікувати. Навпроти, як тільки починається рух, він зупиняється і програма завершується. Щоб зрозуміти таку поведінку, необхідно зрозуміти, як працює блок **Move** у режимі **Unlimited**. При запуску програми першим запускається блок **Move** (він у нас єдиний). Якщо у цьому блоці встановлена тривалість руху, яка відрізняється від **Unlimited**, блок відпрацьовує необхідний час або кількість обертів, і програма завершується. Якщо встановлений рух **Unlimited**, то двигуни запускаються у паралельному режимі. Це означає, що після включення двигунів програма не очікує, доки завершиться виконання блоку **Move**, а переходить до наступного блоку програми; передбачається, що двигуни пізніше будуть зупинені "вручну".

Отже, після запуску серводвигунів, програма переходить до наступного блоку. Але його у нашій програмі немає, тому програма завершується: вона досягла

свого кінця. Разом із завершенням програми автоматично зупиняються усі двигуни. Цим і пояснюється така незвична, на перший погляд, поведінка нашого робота.

Слід мати на увазі і те, що завдання тривалості руху в секундах не зовсім коректне, оскільки при зниженні заряду батареї за той самий час двигун здійснить меншу кількість обертів.

Завдання 2.6. Повороти за допомогою **Steering**.

Встановіть слайдер **Steering** у будь-яке положення, відмінне від центрального, і завантажте програму на робот. Як поводить себе робот? Змінюючи положення слайдера і спостерігаючи за поведінкою робота, дайте відповідь на наступні запитання.

- У який бік повертає робот?
- Від чого залежить "крутизна" повороту?
- При якому положенні слайдера поворот є найбільш крутим?
- У якому напрямку обертаються ведучі колеса при поворотах різної крутизни?
- Де розташований центр повороту при поворотах різної крутизни?

Отже, Ви повинні були помітити, що найшвидший поворот здійснюється двома серводвигунами при переміщенні слайдера **Steering** до упору. При цьому колеса обертаються у протилежних напрямках, центр повороту знаходиться в середині осі, яка з'єднує колеса.

Коли другий двигун вимкнений (слайдер **Steering** у цей момент є недоступним) робот здійснює плавний повільний поворот одним двигуном; центр повороту – нерухоме колесо. Рух робота при цьому схожий на роботу циркуля.

Проміжні положення слайдера **Steering** змушують робот переміщуватися по дузі. При цьому внутрішнє колесо, тобто те, у напрямку якого відбувається поворот, описує дугу меншого радіусу, ніж зовнішнє. Центр повороту співпадає із центром тих кіл, які описують колеса.

Завдання 2.7. Повороти на місці.

Напишіть програму для повороту робота (не серводвигуна!) на місці на 90° ; на 180° ; на 270° ; на 360° . Назвемо спосіб повороту швидким, якщо робот обертає два колеса у протилежних напрямках (слайдер **Steering** переміщений до упору); назвемо його плавним, якщо поворот здійснюється одним колесом (друге є нерухомим).

Для вимірювання кутів поворотів встановіть робот у центрі тестового поля таким чином, щоб заднє колесо "дивилося" на 180° , а вісь ведучих коліс співпадала із червоною смугою.

Занесіть у таблицю кути поворотів двигунів В і С.

Кут повороту робота	Спосіб повороту	Кут повороту серводвигунів
90°	Швидкий	
180°	Швидкий	
270°	Швидкий	
360°	Швидкий	
90°	Плавний	
180°	Плавний	
270°	Плавний	
360°	Плавний	

Спробуйте використати отримані дані для повороту на задану кількість градусів в інший бік. Вдалося?

Завдання 2.8. Перегони.

Напишіть програму для переміщення робота навкруги коробки від набору Lego Mindstorms. Влаштуйте змагання у групі по швидкісному об'їзду коробки: один заїзд – три кола.

3.6.5. Стани і події. Давачі

Поведінка робота – це низка станів, які змінюють один одного (стан руху, стан пошуку і т.п.). Зміна станів відбувається у залежності від зовнішніх подій. Робот реєструє події за допомогою різноманітних давачів. Всю поведінку робота можна схематично зобразити у вигляді діаграми переходів між різними станами.

Приклад 3.1. Діаграма переходів "Стан 1 → Подія → Стан 2", що відповідає руху робота до чорної лінії (рис. 3.4.23):

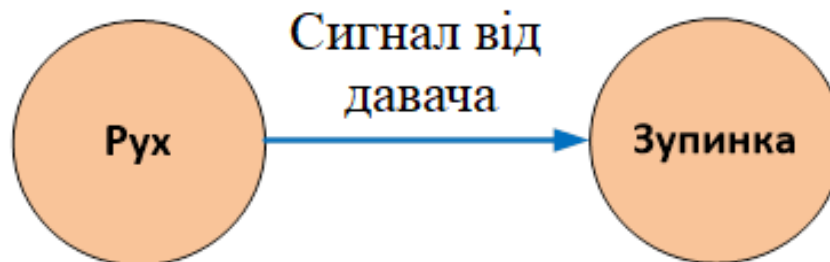


Рис. 3.4.23. Діаграма переходів: рух робота до чорної лінії

1. Робот знаходиться у стані руху.

2. Давач світла реєструє зниження яскравості до певного межового значення – це нова подія, яку можна назвати "робот доїхав до чорної лінії".
3. В результаті реєстрації події робот змінює свій стан на бездіяльність (зупинка).

Таким чином, можна сказати, що робот повинен весь час очікувати виникнення подій. Для цього у середовищі NXT-G присутній спеціальний блок Wait (Очікувати):



Рис. 3.4.24. Варіанти блоку Wait у загальній палітрі

Джерелами подій можуть бути:

- різноманітні давачі (сенсори): кольору, освітленості, дотику, та ін.;
- таймер;
- кнопки NXT-блоку;
- повідомлення від інших роботів, які передані по Bluetooth з'єднанню.

У подальшому у цій темі ми розглянемо роботу із деякими давачами, які використовуються найчастіше.

Давач кольору (Color Sensor)



Рис. 3.4.25. Зовнішній вигляд давача кольору

Давач кольору (RGB-давач) суміщає три функції:

1. Дозволяє роботу розрізняти кольори.
2. Дозволяє роботу розрізняти рівень освітленості шляхом вимірювання яскравості світла, що відбивається.
3. Кольорове підсвічування.

Робота у режимі визначення кольору

Приклад 3.2. Очікування події від давача кольору.

Робот рухається по тестовому полю вперед до того часу, доки не зареєструє синій колір.

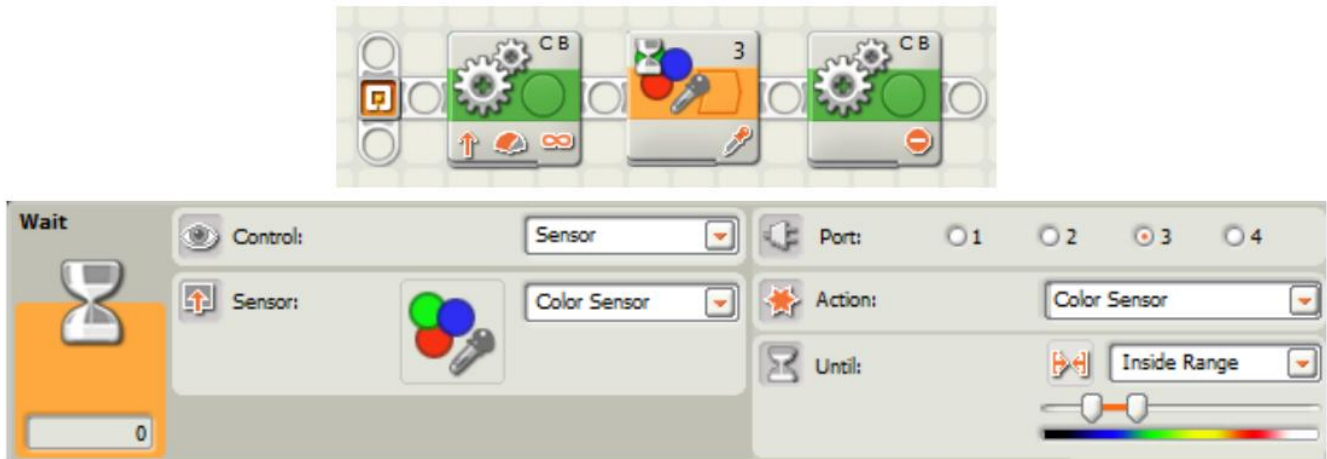


Рис. 3.4.26. Приклад програми "Рух до синього"

У попередньому матеріалі "**Програмування серводвигунів: команда Move**" ми відмітили, що параметр **Unlimited (Без Обмеження)** у налаштуваннях блоку **Move** дозволяє роботу рухатися без будь-яких обмежень до появи чергової події. Після цього можна змінити стан робота. Зміна стану не відбувається автоматично: його необхідно запрограмувати! У нашому прикладі після настання події (реєстрація давача синього кольору) запрограмована зупинка моторів В і С.

Зверніть увагу:

- *поле **Until** вимовляється англійською [анті:л], а перекладається наступним чином: "до того часу, доки не настане...";*
- *у полі **Until** можливий один із двох варіантів:*
 - **Inside Range (Всередині діапазону)** – *очікується подія "потрапляння реєстрованого кольору всередину вказаного діапазону кольорів";*
 - **Outside Range (Назовні діапазону)** – *очікується подія "потрапляння реєстрованого кольору назовні вказаного діапазону кольорів";*
- *колір у полі **Until** задається за допомогою двох слайдерів.*

Завдання 3.1. Протестуйте давач кольору:

- Додайте до конструкції робота **Five Minute Bot** давач кольору, розташований попереду і направлений вниз, на висоті приблизно 1–1,5 см від поверхні підлоги (столу).
- Додайте на полі програми блок **Wait Color Sensor** і випробуйте його на предметах різних кольорів:
 1. Підключіть блок NXT до комп'ютера через USB-з'єднання.
 2. Виділіть блок **Wait** і натисніть на кнопку **Download and run selected контролера NXT** (рис. 3.4.27).
 3. Не від'єднуючи USB-кабель, піднесіть предмети різного кольору до давача на відстань приблизно 1 см.

4. Слідкуйте за результатом у полі зворотного зв'язку на панелі налаштувань давача (лівий нижній кут панелі налаштувань).
5. Здійсніть таке саме тестування за допомогою меню View NXT. Зробіть висновок, чи співпадають результати тестів?



Рис. 3.4.27. Компіляція і виконання виділеного фрагменту

Завдання 3.2. Складіть програму для переміщення робота по *тестовій смугі* до виявлення червоного (синього, зеленого та ін.) кольору (*тестова смуга* являє собою таблицю різних кольорів, зображена по краю тестового поля. Містить чорний, синій, зелений, червоний, жовтий кольори та їх градації). Після виявлення заданого кольору робот повинен вимовити назву цього кольору.

Завдання 3.3. Складіть програму для переміщення робота по тестовій смугі до виявлення зеленого або синього кольору. Запускаючи робот з різних боків тестової смуги, перевірте роботу складеної програми.

Робота у режимі вимірювання освітленості

Давач кольору із навчального набору може працювати у двох режимах – як давач кольору і як давач освітленості. Переведіть у налаштуваннях блоку **Wait Color Sensor** давач кольору у режим давача освітленості, для чого у полі **Action** оберіть **Light Sensor** (рис. 3.4.28).

Зверніть увагу, що зовнішній вид блоку **Wait** змінився.

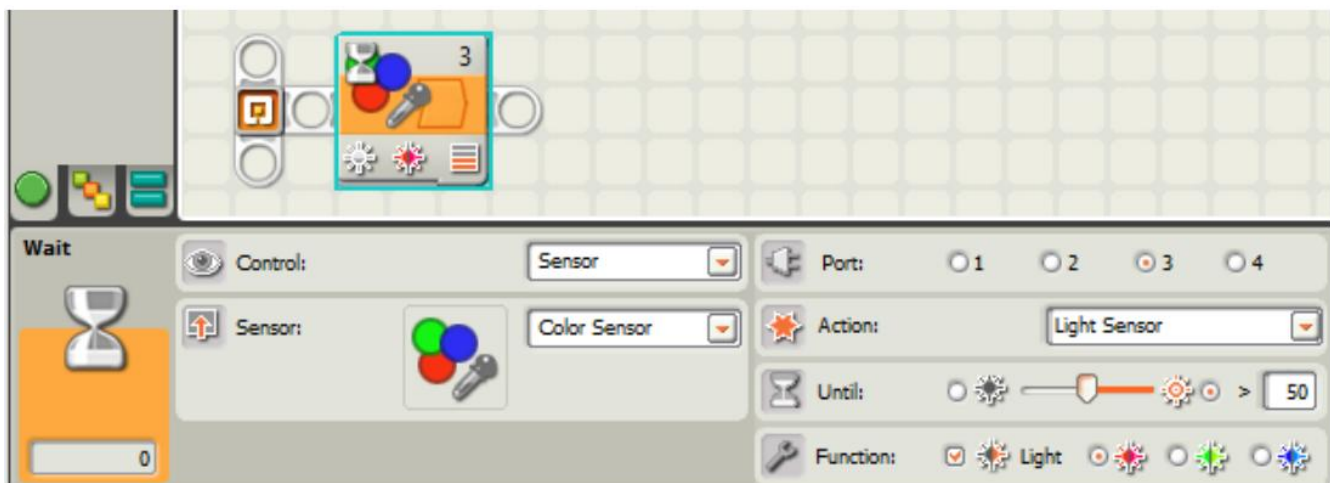


Рис. 3.4.28. Налаштування блоку **Wait Color Sensor** у режимі вимірювання освітленості

Для кращої уяви, що саме бачить давач освітленості, уявіть його переміщення над чорно-білою фотокарткою. Результатом вимірювання буде яскравість у діапазоні від 0 (чорний колір) до 100 (білий), хоча межових значень на практиці зафіксовано не було.

Зверніть увагу:

- *рівень освітленості у полі Until задається за допомогою слайдера або вводу числового значення;*
- *знак у події, що очікується, перемикається натисканням на кнопку із яскравим або затемненим "сонечком" (за замовчуванням встановлений знак ">");*
- *значення поля Until вимовляється як: "до того часу, доки освітленість не стане...". Наприклад, на рис. 3.4.28 очікується подія "рівень освітленості > 50", що означає виконання дії, яка встановлена попереду блоку Wait до того часу, поки освітленість не стане більша за 50;*
- *поле Function дозволяє використовувати давач кольору як джерело червоного, зеленого або синього світла (увімкнений прапорець Light). За допомогою вимірювання інтенсивності світла, яке відбивається від поверхні, можна оцінити її яскравість: від темної поверхні відбивається менше світла, ніж від світлої поверхні. Взагалі, якщо заміри освітленості відбуваються при гарному денному світлі, підсвічування можна вимкнути (зняти прапорець Light). Однак виробник радить у режимі вимірювання освітленості використовувати червоне підсвічування. Також при вимкненому світлі можна вимірювати загальну освітленість в оточуючому середовищі.*

Завдання 3.4. Дослідіть роботу давача кольору в режимі вимірювання освітленості на предметах різного кольору аналогічно тому, як це здійснювалось у попередньому завданні:

- Запишіть ті значення освітленості, які давач визначає для чорного, синього, блакитного, жовтого, зеленого, червоного кольорів на тестовому полі.
- Використовуйте для підсвічування лампи різних кольорів. Як змінюються покази давача?
- Вимкніть підсвічування і порівняйте покази давача з підсвічуванням і без.
- Здійсніть кілька експериментів із предметами, що Вас оточують.

Завдання 3.5. Використовуючи RGB-давач у режимі вимірювання освітленості, складіть програму для переміщення робота по білому полю до виявлення чорного кольору. З цією метою у полі **Until** зазначте необхідний поріг чорного кольору. Після зупинки робот повинен вивести на екран відповідне повідомлення (наприклад, "I Found Black Color").

Давач ультразвуку (Ultrasonic Sensor)

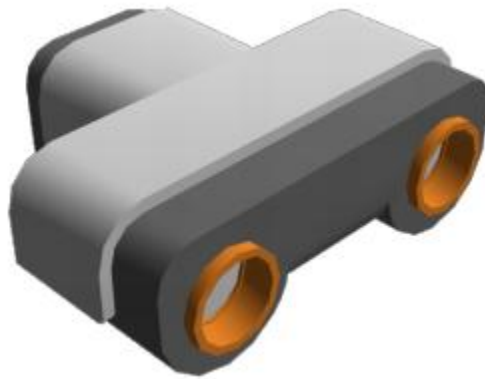


Рис. 3.4.29. Зовнішній вигляд давача ультразвуку

Давач ультразвуку у певному сенсі замінює роботу зір, оскільки працює за тим самим принципом, як ехолот у летючих кажанів: відстань вимірюється за рахунок визначення часу, що необхідний для звукової хвилі для повернення після відбиття від об'єкту. Налаштування цього давача показують, що із його допомогою можна виміряти відстань від 0 до 255 см із точністю ± 3 см. Однак на практиці мінімальна відстань, на якій ультразвуковий давач видає більш-менш достовірний показ, складає 5 - 7 см.

Крім того, слід мати на увазі, що у порівнянні із давачами кольору, світла, дотику та ін., давач ультразвуку є доволі повільним. Це пояснюється тим, що, по-перше, швидкість розповсюдження звуку є значно меншою за швидкість світла, а по-друге, у цього давача повільний протокол обміну даними.

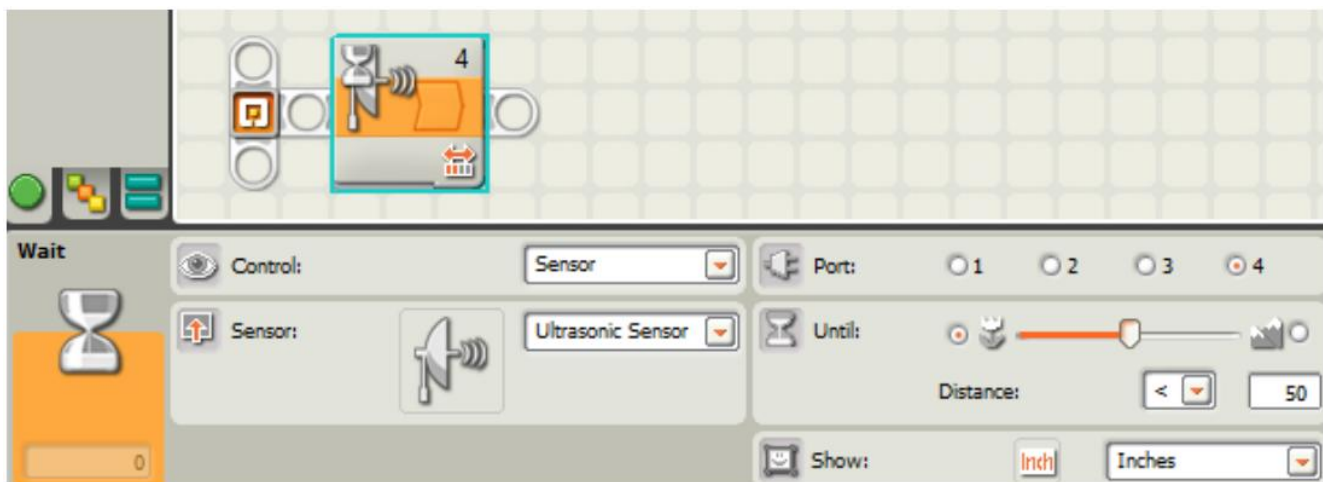


Рис. 3.4.30. Налаштування блоку Wait Ultrasonic Sensor

Додайте до складу конструкції робота давач ультразвуку, який спрямований вперед по ходу руху робота. Налаштування давача наведені на рис. 3.4.30.

Зверніть увагу:

- у полі **Until** так само, як і для **Color Sensor**, задається умова виходу із блоку;
- знак із "<" на ">" або навпаки у події, що очікується, перемикається натисканням на кнопку і переміщенням вказівника на **Farther Than** (Більше, ніж...) (зображення "гори") або **Nearer Than** (Менш, ніж...) (зображення квітки – встановлене за замовчуванням). Також можна скористатися списком;
- поле **Show** задає одиниці вимірювання відстані. За замовчуванням відстань вимірюється в дюймах (*inches*). Не забувайте перемикати цей параметр на сантиметри.

Завдання 3.6. Проведіть тестування давача ультразвуку, наближуючи до нього предмети на різну відстань і спостерігаючи за результатами:

- за допомогою меню **View**;
- у полі зворотного зв'язку блоку **Wait**.

Завдання 3.7. Складіть програму, за допомогою якої робот рухається в напрямку перешкоди і зупиняється на відстані 40 см від неї.

Давач дотику (**Touch Sensor**)

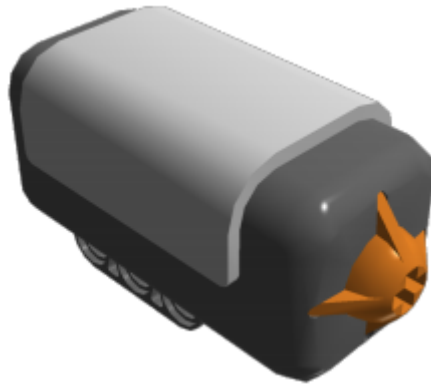


Рис. 3.4.31. Зовнішній вигляд давача дотику

Додайте до конструкції робота **Five Minute Bot** кнопку в якості давача дотику, яку можна підключити за допомогою довгого кабелю із подальшим використанням як пульт дистанційного керування.

У налаштуваннях блоку **Wait Touch** у ролі **Action** доступними є такі варіанти:

- **Pressed** – давач натиснутий;
- **Released** – давач відпущений;
- **Bumped** – виконане клацання (тобто кнопка натиснута і одразу відпущена).

Завдання 3.8. Складіть програму для гри "Хто точніше?", сенс якої полягає в тому, щоб зупинити робот точно на заданій лінії. Після запуску програми робот

повинен початим рух у напрямку до чорної лінії. Як тільки колеса робота торкнуться лінії, гравець повинен натиснути кнопку, і при цьому робот має зупинитися. Перемагає той гравець, у якого відстань від передніх коліс до заданої лінії є найменшим. Налаштування блоку **Move**: необмежений рух, потужність 80%. Оберіть найкращий варіант для вирішення поставленої задачі.

3.6.6. Цикли

При написанні програм часто виникає необхідність у циклічному повторенні певної групи команд. Для цього використовується мовна конструкція "цикл", якій в NXT-G відповідає блок **Loop**.

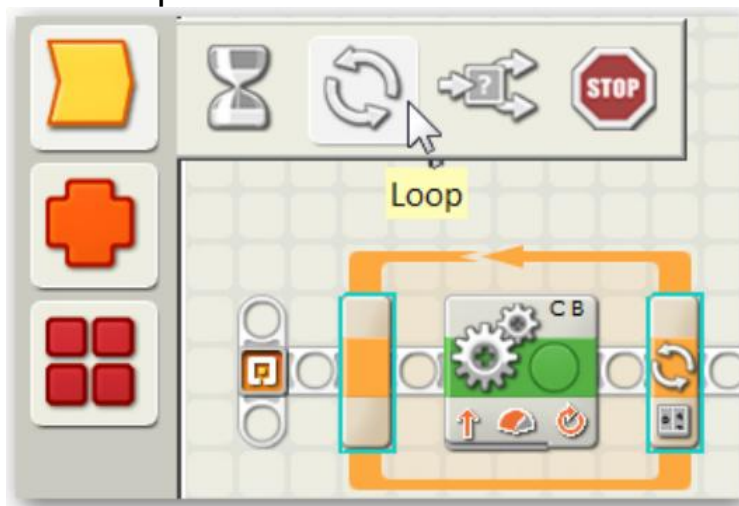


Рис. 3.4.32. Блок **Loop** у меню **Flow** повної палітри та приклад його використання

Всередині циклу розміщуються блоки, які необхідно виконувати багаторазово. Зауважимо, що необхідно з розумінням (із обережністю) розміщувати у циклі блоки, яким самим у певному сенсі характерна циклічність, наприклад **Move**, **Sound** та деякі інші.

Програмуючи цикл, необхідно знати заздалегідь, у якому випадку він зупинить свою роботу. Завершення циклу (вихід із циклу) може відбуватися по різним причинам. У NXT-G можливі наступні способи виходу із циклу, які доступні у полі **Control**:

1. **Forever** – нескінченно (за замовчуванням);
2. **Sensor** – при спрацюванні датчика;
3. **Time** – через визначений час;
4. **Count** – після виконання циклу певної кількості разів;
5. **Logic** – при виконанні деякої логічної умови, що задається програмістом.

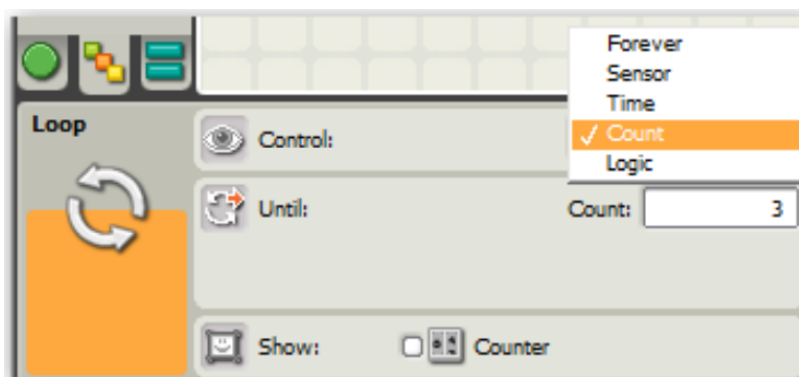


Рис. 3.4.33. Вибір варіанту виходу із циклу

У процесі вивчення мови NXT-G нами поступово будуть засвоєні усі вказані способи. Почнемо із нескінченного циклу.

Найпростіші види циклів

Завдання 4.1. Підготуйте робот із давачем кольору у передній частині, при чому давач повинен бути спрямований вниз і знаходитися на відстані близько 1 см від столу. Розташуйте робот перед чорним полем (або перед чорною лінією) на відстані від неї близько 20 см.

Завдання 4.2. Рух до лінії. Використовуючи давач кольору в режимі вимірювання освітленості, складіть програму, за допомогою якої робот доїде до чорної смуги і зупиниться (див. завдання 3.3).

Завдання 4.3. Рух до лінії і назад. Змініть програму із завдання 4.2 так, щоб після досягнення чорної лінії робот від'їжджав на біле поле.

Завдання 4.4. Нескінченний циклічний рух до лінії і назад. Змініть програму із завдання 4.3 таким чином, щоб робот виконував рух до лінії і назад нескінченну кількість разів. Для цього переведіть попередню програму в нескінченний (**forever**) цикл.

Завдання 4.5. Циклічний рух до лінії і навпаки за лічильником, де необхідно змінити програму пересування до лінії і назад за лічильником. Наприклад, переналаштуємо програму із завдання 4.4 таким чином, щоб рух до лінії і назад здійснився три рази. Зміна налаштувань у цьому випадку виглядає наступним чином (рис. 3.4.34).

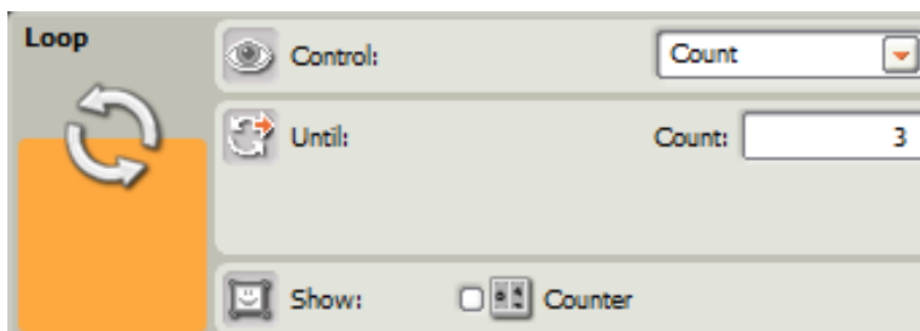


Рис. 3.4.34. Налаштування циклу із лічильником

Переміщення роботу по лінії

Приклад 4.1. Розглянемо класичну задачу про переміщення робота вздовж лінії. Паперовий полігон із такою лінією входить до складу кожного набору.

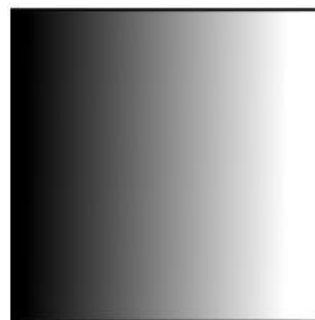
Знайдемо відповідь на питання, чи може робот переміщуватися по лінії, тобто таким чином, щоб давач завжди знаходився би на чорному полі. Для цього уявімо, що через певний час після початку руху давач "з'їхав" із чорного поля і опинився на білому. Зрозуміло, що в цьому випадку робот повинен скоригувати свій рух і повернутися на чорну лінію. Але у який бік робот має "підрулювати": вправо чи вліво? Справа у тому, що білий колір з обох боків від чорної лінії є однаковим, і коли робот потрапляє у таку ситуацію, він не може прийняти однозначне правильне рішення.

Для виходу із такої ситуації необхідно змінити постановку задачі: робот повинен переміщуватись не по лінії, а по межі чорного і білого кольору.

Що при цьому зміниться? Людина і робот сприймають межу між чорним і білим кольором по-різному (рис. 3.4.35).



Сприйняття межі між
чорним і білим
кольорами людиною



Сприйняття межі між
чорним і білим
кольорами роботом

Рис. 3.4.35. Різниця у сприйнятті людиною і роботом межі між чорним і білим кольором

Бачимо, що давач кольору при своєму русі вправо або вліво чіткої межі між білим і чорним кольором не спостерігає. Це відбувається тому, що при його переміщенні, наприклад зліва направо (рис. 3.4.35), кількість світла, яка потрапляє на чутливий елемент давача, поступово збільшується. Отже, покази давача також будуть плавно збільшуватися.

Більш за все нас влаштовує середнє положення давача, тобто таке, коли він знаходиться над межею. У цьому випадку давач спостерігає не чітко білий або чітко чорний колір, а певний усереднений – сірий. Для визначення показу давача для сірого кольору необхідно визначити середньоарифметичне між чорним і білим кольорами.



Рис. 3.4.36. Визначення середньоарифметичного "сірого" між чорним і білим кольорами

При пересуванні робота зміщення вліво (в чорну область) призведе до зменшення показів давача, а зміщення вправо (у білу область) – до збільшення показів. За таким принципом можна розрізнити ліве від правого.

Отже, програма для переміщення робота вздовж лінії виглядатиме наступним чином (рис. 3.4.37).

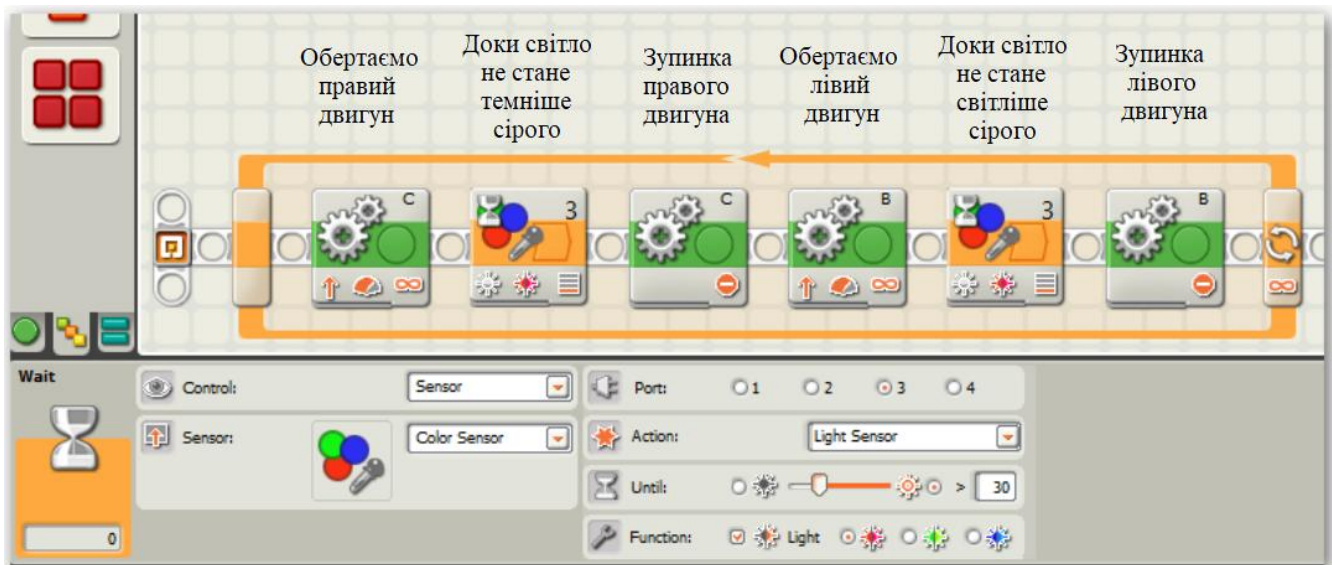


Рис. 3.4.37. Програма переміщення робота вздовж лінії

Зверніть увагу: за описаних вище умов робот розташовується на полі таким чином, щоб давач кольору "дивився" на межу лінії: чорний колір з лівого боку, білий – з правого.

Завдання 4.6. Знайдіть відмінності між запропонованим алгоритмом і програмою із завдання 4.4.

Завдання 4.7. Складіть програму для переміщення робота вздовж чорної лінії, змінивши код програми 4.4 та перевірте її працездатність.

Завдання 4.8. Запустіть робот вздовж тієї ж лінії, але у зворотний бік. Як це можна зробити без змін у програмі?

Цикл із лічильником. Передача даних між блоками

Розглянемо наступний приклад.

Приклад 4.2. Необхідно скласти переміщення робота із прискоренням.

Для початку складемо програму переміщення робота із постійною швидкістю за наступних умов:

- Кількість повторів – 20 разів;
- Напрямок руху – вперед.

У налаштуваннях блоку **Move** необхідно відмітити **Coast** (без гальмування наприкінці виконання блоку).

У полі **Duration** встановлюємо значення **1 Rotation**.

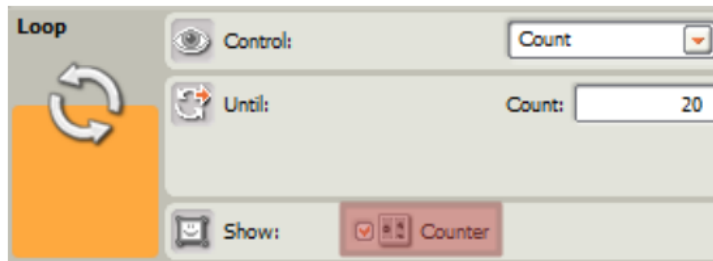
Перевіряємо роботу програми. Робот переміщується прямо із постійною швидкістю.

Для створення прискорення скористаємося передачею даних із циклу на блок руху. Вочевидь, оскільки прискорення передбачає поступове збільшення швидкості робота, воно буде реалізоване за рахунок збільшення потужності мотора.

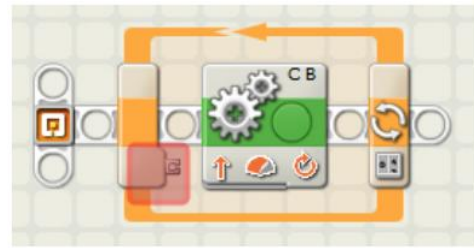
Внесемо зміни у програму.

У налаштуваннях блоку **Loop** встановимо прапорець **Show Counter** (Показати Лічильник).

Зверніть увагу на зміну зовнішнього виду циклу: з'явився роз'єм **Loop Count**.



а)



б)

Рис. 3.4.38. Відображення лічильника циклу (а) і доступ до нього (б)

Відкриємо *висувну панель концентратора даних* блоку **Move**, клацнувши на кнопці у лівій нижній частині блоку (рис. 3.4.39).

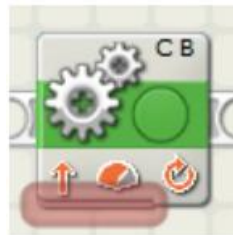


Рис. 3.4.39. Висувна панель концентратора даних блоку **Move** у згорнутому стані

У результаті цього відкриваються вхідні роз'єми (лівий бік концентратора) та вихідні роз'єми (правий бік концентратора), які призначені для підключення шини даних від роз'ємів інших блоків (рис. 3.4.40). Закрити висувну панель концентратора можна, клацнувши на тому самому місці блоку. Отже, в такий спосіб можна передавати дані всередині програми на NXT-G без фактичного використання змінних.

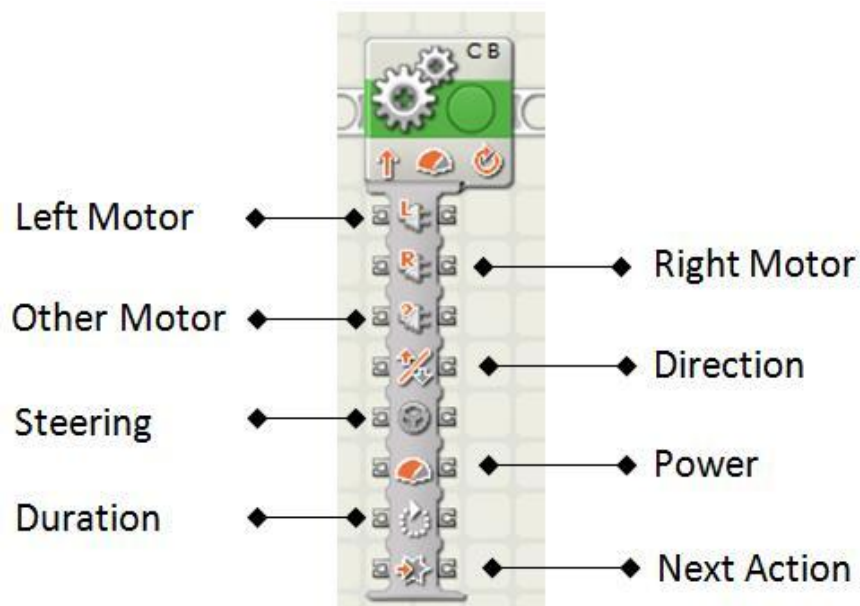


Рис. 3.4.40. Висувна панель концентратора даних блоку **Move** в розгорнутому стані

Поеднаємо шиною роз'єм лічильника циклу із роз'ємом Power (Потужність) блоку Move (при наведенні вказівника миші на роз'єм з'являється відповідна спливаюча підказка). Для виконання з'єднання клацаємо лівою кнопкою миші на початок, проводимо мишу у кінець шляху і клацаємо ще раз. По ходу траєкторії можна здійснювати додаткові клацання у тих місцях, де бажано зробити перегин. У полі Duration зазначаємо Unlimited (без обмеження тривалості). Далі згортаємо концентратор даних – клацнемо по висувній панелі блока таким чином, ніби хочемо її згорнути. Усі роз'єми при цьому, до яких не підходять шини, зникають. Виконання таких дій допомагає помітно зменшити площу, яку займає програма.

Отримуємо програму, яка має наступний вигляд (рис. 3.4.41).

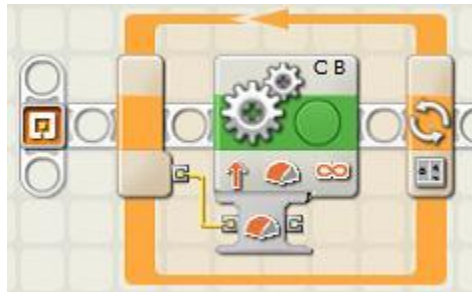


Рис. 3.4.41. Програма переміщення робота із прискоренням

Перевіряємо роботу програми, і для відображення прискорення більш наочно, додаємо у цикл невелику затримку, як це показано на рис. 3.4.42.

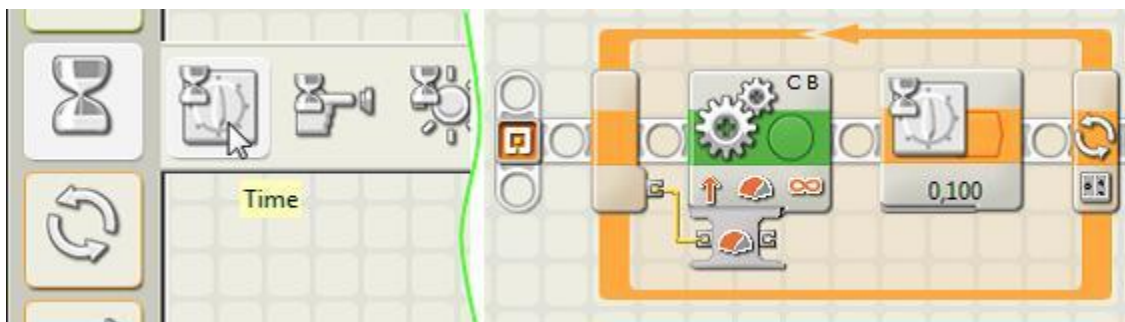


Рис. 3.4.42. Переміщення робота із прискоренням та із затримкою часу для більшої наочності

Приклад 4.3. Виведення інформації на екран блоку NXT.

Перевіримо, що насправді надходить із циклу на вхід блоку Move. Для цього складемо спрощену програму, за допомогою якої значення лічильника циклу виводиться на екран блоку NXT. Нагадаємо, що для виводу використовується блок Display з панелі Common.

Оскільки нам необхідно виводити на екран числову інформацію, її спочатку необхідно перетворити на текстового тип. Це можна здійснити за допомогою блоку Number to Text панелі Advanced.

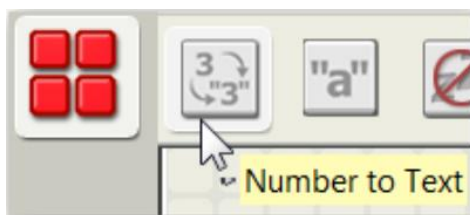


Рис. 3.4.43. Доступ до блоку **Number to Text** панелі **Advanced**

Схема програми виглядатиме наступним чином:

- у налаштуваннях блоку **Loop** значення лічильника встановлюємо таким, що дорівнює 10;
- значення лічильника циклу передаємо до перетворювача "Число → Текст";
- отриманий текст передаємо на вхід блоку **Display**;
- створюємо затримку для того, щоб встигнути зафіксувати значення на екрані;
- після циклу встановлюємо затримку до натискання на кнопку **Start**, щоб побачити останнє значення, яке було виведене на екран.

Графічна інтерпретація програми наведена на рис. 3.4.44.

Зверніть увагу: шини різних даних забарвлені різними кольорами. Всього у *NXT-G* використовуються три типи даних, і кожний тип має власне забарвлення:

- числовий тип даних – жовтий колір;
- текстовий тип – червоний колір;
- логічний тип даних – зелений колір.



Рис. 3.4.44. Програма виводу числових значень лічильника на екран

На рис. 3.4.45 наведені налаштування останнього блоку:

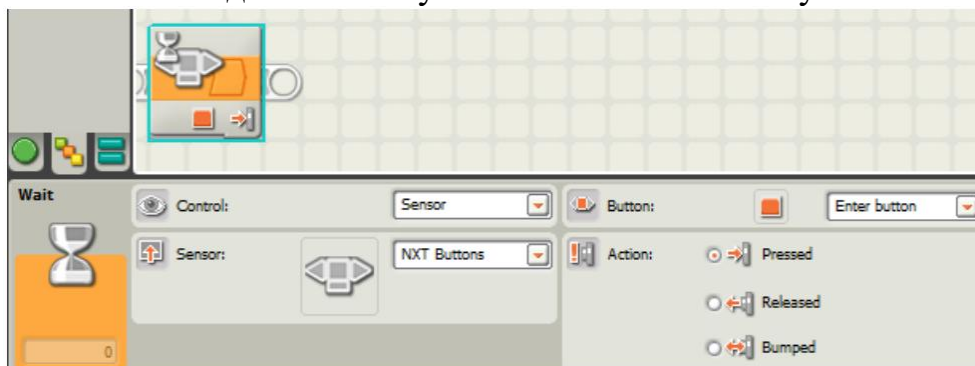


Рис. 3.4.45. Налаштування блоку **Wait**

Цикл із виходом за значенням датчика

В NXT-G існує варіант використання циклу, в якому вихід відбувається у відповідності з певним значенням якогось давача. Налаштування такого циклу повністю співпадає із налаштуванням блоку **Wait**, а різниця полягає в тому, що у параметрах циклу встановлений прапорець **Show Counter**. Пустий цикл із виходом за значенням давача подібний відповідному блоку **Wait**, але всередині циклу можна розмістити блоки, які повинні багаторазово повторюватися до спрацювання давача.

Приклад 4.4. Необхідно перетворити програму, зображену на рис. 3.4.44, на найпростіший секундомір. Для цього також необхідно відлічувати час, але зупинка має здійснитися після натискання на кнопку. Графічна інтерпретація такого перетворення зображена на рис. 3.4.46).

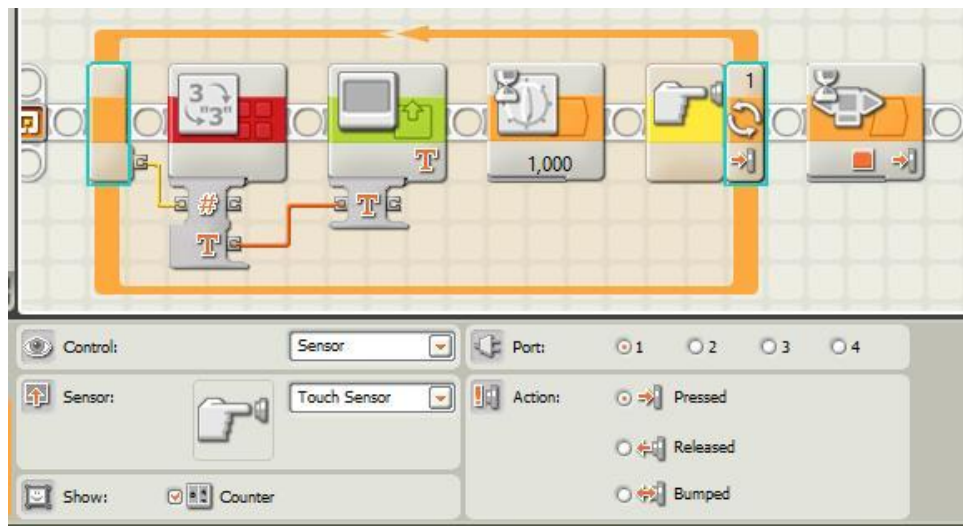


Рис. 3.4.46. Найпростіший секундомір із використанням циклу із виходом по значенню давача

Завдання 4.9. Складіть програму, за якою робот, оснащений ультразвуковим давачем, повинен під'їхати до стінки і зупинитися на відстані 30 см від неї, при цьому поточну відстань до стінки робот повинен під час руху виводити на екран.

Цикл із виходом за умовою

Існують задачі, у яких доцільно або зручно використати цикл, вихід із якого відбувається при виконанні певної логічної умови. За своєю структурою він подібний до циклів "While", "Do, While" або "Repeat-Until" і т.п. різних мов програмування. Налаштування його здійснюється так, як показано на рис. 3.4.47. На цій схемі роз'єм №1 (Loop Condition) з'являється тоді, коли у полі №2 (Control) обране значення **Logic**. На такий порт циклу необхідно підвести *зелений* провід із

логічним значенням. Поле №3 (Until) дозволяє визначити умову, за якою виконання циклу зупиняється: коли умова, що була подана на роз'єм, набуває значення "істинне" (True), або коли вона стає "неправдива" (False).

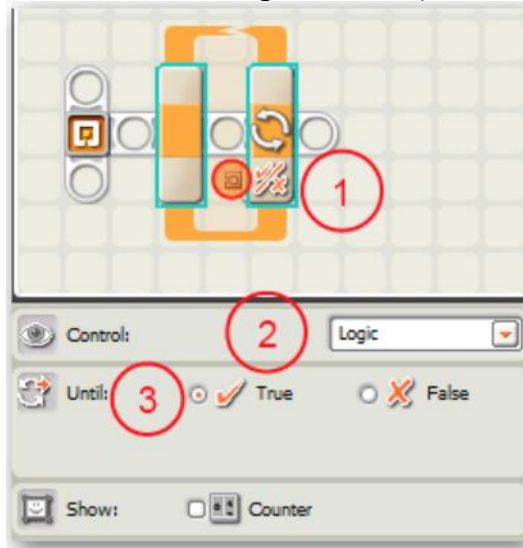


Рис. 3.4.47. Налаштування циклу із виходом за логічною умовою

Приклад 4.5. Необхідно скласти програму для очікування в n секунд за допомогою таймера на 1 секунду. (Приклад не є типовим і наведений лише для знайомства із циклом "вихід за умовою").

У даному циклі (рис. 3.4.48), що налаштований на вихід за певною умовою, виконуються такі дії:

1. здійснюється відлік 1 секунди за допомогою відповідного блоку;
2. номер ітерації циклу, який подається із порту **Counter**, надходить на блок порівняння **Compare** з меню **Data (Дані)** повної палітри (рис. 3.4.49);
3. тут він порівнюється із числом n , і результат цього порівняння зеленим проводом подається на другий роз'єм циклу;
4. якщо умова, задана у блоці порівняння, виконується, цикл припиняє свою роботу, і з динаміка лунає звуковий сигнал.

Блок **Compare** надає доступ до наступних логічних операцій:

- **Less than** - "менш, ніж";
- **Greater than** - "більш, ніж";
- **Equal** - "дорівнює".

Дані у поля блоку **Compare** можуть потрапляти як по зовнішнім роз'ємам, так і шляхом вводу у відповідні поля. Так, у прикладі 4.5 (рис. 3.4.50) дані в поле А потрапляють із роз'єму лічильника циклу (для введення у поле в даному випадку доступ відсутній), а у поле В дані задає користувач.

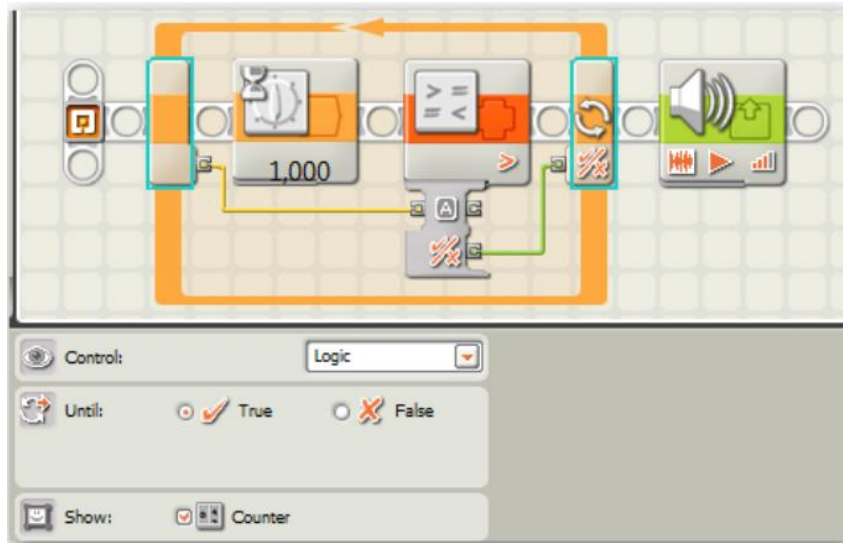


Рис. 3.4.48. Програма для моделювання очікування в n секунд за допомогою таймера на 1 секунду



Рис. 3.4.49. Блок Compare у повній палітрі

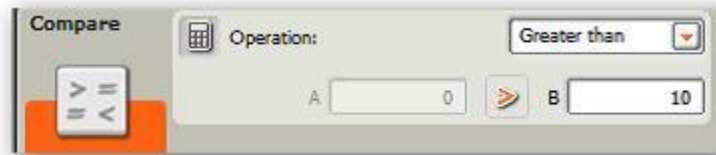


Рис. 3.4.50. Налаштування блоку Compare у прикладі 4.5 (в даному випадку $n = 10$)

Завдання 4.10. Здійсніть дослідження коду програми для прикладу 4.5. Складіть програму і перевірте її роботу при різних значеннях n .

Завдання 4.11. Складіть програму, яка після запуску очікує натискання на помаранчеву кнопку NXT, після чого видає звуковий сигнал (наприклад, лунає коротка фраза) і завершує роботу. Для опрацювання натискання на помаранчеву кнопку необхідно використати програмний блок NXT Buttons з меню Sensors та циклом із виходом за умовою. Використовувати блок Wait при виконанні цього завдання забороняється! Нагадаємо, що блоки жовтого кольору групи Sensors дозволяють отримати поточні покази датчиків.

Завдання 4.12. Змініть попередню програму таким чином, щоб вона завершувалася після n -го натискання на помаранчеву кнопку. Використайте вкладені цикли.

3.6.7. Розгалуження в NXT-G

Мова NXT-G, окрім послідовного і циклічного виконання програм, передбачає також розгалуження, яке реалізується за допомогою оператора Switch. Названий оператор перевіряє виконання певної логічної умови і запускає програму по одній з двох або декількох гілок.

У мові NXT-G оператор розгалуження використовується рідше, ніж у звичайних мовах програмування, оскільки наявний більш зручний оператор Wait. Однак існують випадки, коли необхідно використати саме розгалуження.

Приклад 5.1. Складіть програму, за якою робот повинен почати рух після натискання на помаранчеву кнопку Start. Якщо давач кольору опиниться над червоним об'єктом, робот має видати звуковий сигнал "Red" і зупинитися, а програма повинна завершитися. Інакше робот повинен переміщуватися і періодично видавати звуковий сигнал "Detect".

Вирішення поставленої задачі виглядає приблизно так, як зображено на рис. 3.4.51.

Зверніть увагу на налаштування блоку Switch.

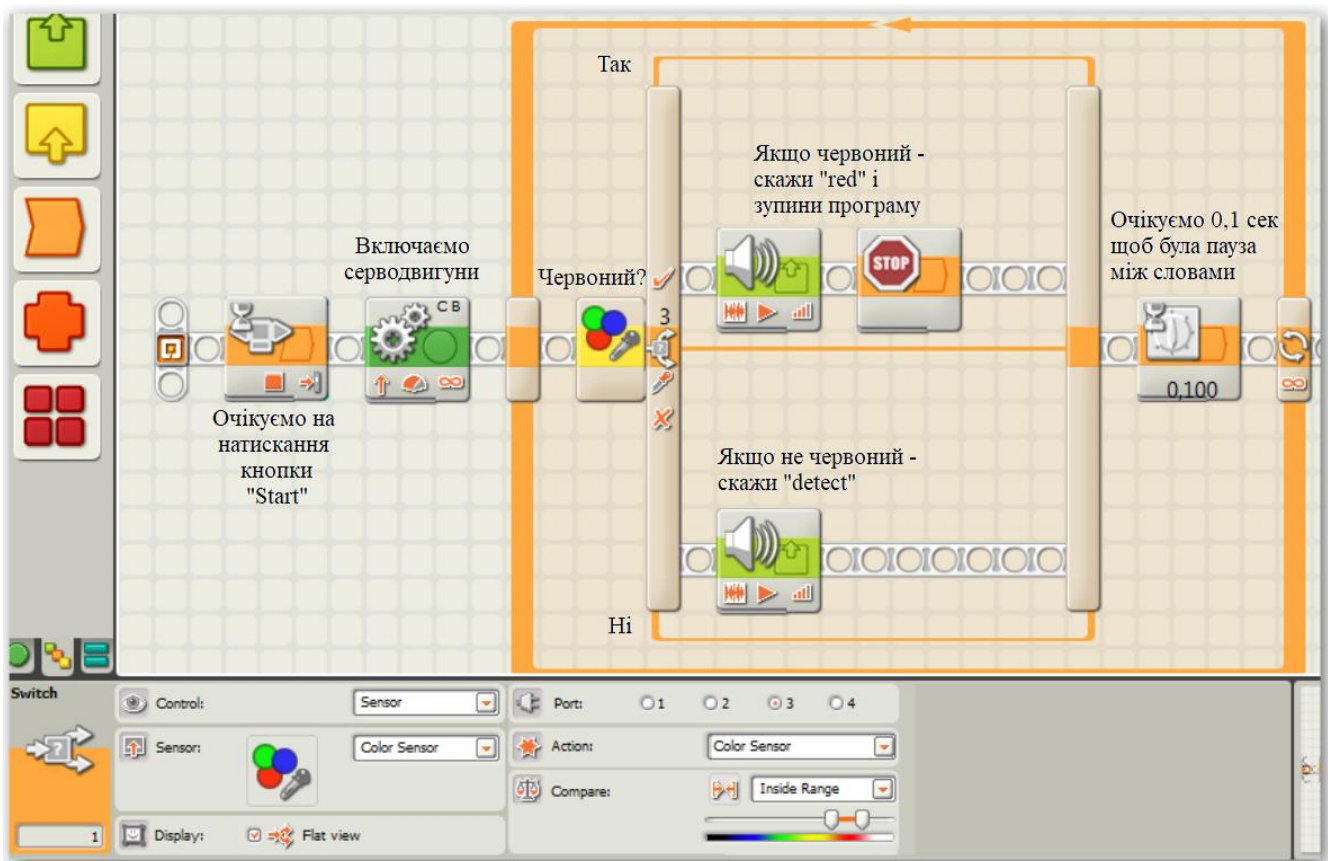


Рис. 3.4.51. Графічна інтерпретація програми до прикладу 5.1 у мові NXT-G

За допомогою блоку **Wait**, що розташовується на першому місці наведеної програми, робот запускається кнопкою **Start** на блоці. Звукові сигнали, які повинен "вимовляти" робот згідно із завданням, програмуються за допомогою блоку **Sound**, налаштування якого зображені на рис. 3.4.52. Налаштування **Sound** у другій гілці розгалуження здійснюється аналогічно.



Рис. 3.4.52. Налаштування блоку **Sound** для прикладу 5.1

Зверніть увагу на прапорець **Wait for completion** у налаштуваннях блоку **Sound**. Якщо він встановлений, спочатку відтворюється заданий набір звуків, і тільки потім управління передається наступному блоку. Якщо прапорець знятий, вмикається звук із одночасною передачею управління до наступного блоку.

Наприкінці нескінченного циклу встановлена затримка на 0,1 секунди.

Отже, у полі **Control** блоку **Switch** встановлюється один із двох варіантів:

- **Sensor** – поведінка робота залежить від логічної умови, що пов'язана із поточним значенням давача;
- **Value** – поведінка робота залежить від результату порівняння логічного, числового або текстового значення, яке надійшло ззовні, із заданим значенням.

У полі **Display** прапорець **Flat view** вмикає або вимикає повне відображення блоку **Switch** (складається з двох гілок розгалуження). У вимкненому стані обидві гілки відображуються у вигляді вкладок, при чому на даний момент можна побачити тільки одну з них (рис. 3.4.53). Відмічена "галкою" вкладка відповідає гілці, по якій піде робота програми в тому випадку, якщо умова, що перевіряється, є істинною, а вкладка із "хрестиком" – неправдивому значенню умови.

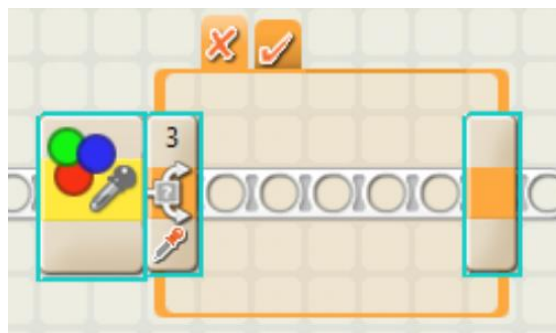


Рис. 3.4.53. Компактне відображення блоку **Switch**

Приклад 5.2. Робот "вимовляє" "No" у тому випадку, якщо давач дотику не натиснутий, а якщо натиснутий - "Yes".

Зверніть увагу на можливість зміни того значення, що порівнюється, за допомогою поля Condition блоку Switch (рис. 3.4.54). У блоці Touch Sensor поля Action обрана позиція Pressed.

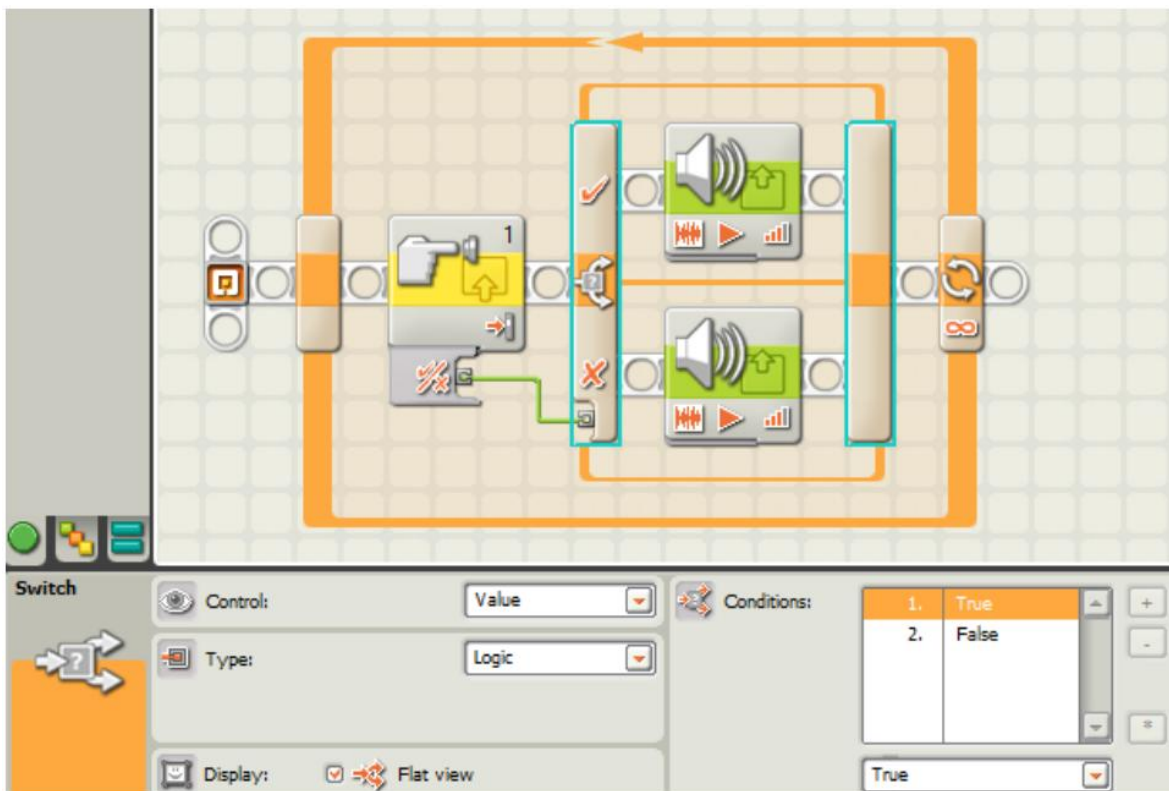


Рис. 3.4.54. Блок Switch: приклад використання даних логічного типу

Приклад 5.3. На екрані NXT здійснюється послідовна зміна п'яти зображень годинника (використовуються файли Time00 - Time04) залежно від лічильника циклу. Для того, щоб створити додаткові вкладки значень блоку Switch, необхідно використати кнопку "+" у полі Condition (рис. 5.5).

Зверніть увагу, що прапорець Flat view для цього повинен бути вимкнений. Вкладки, які відповідають різним значенням, мають власні підписи, які з'являються при наведенні на них курсору миші. У тому випадку, коли кількість оброблюваних значень більша п'яти і виникає необхідність переглянути усі додані вкладки, блок Switch необхідно "розтягнути" вниз або вправо. Для цього курсор миші необхідно навести на балку Lego всередині блоку, натиснути ліву кнопку миші і потягнути її вниз, або аналогічно вправо.

Описане використання блоку Switch із чисельними вкладками відповідає оператору вибору Case (Switch) у текстових мовах програмування.

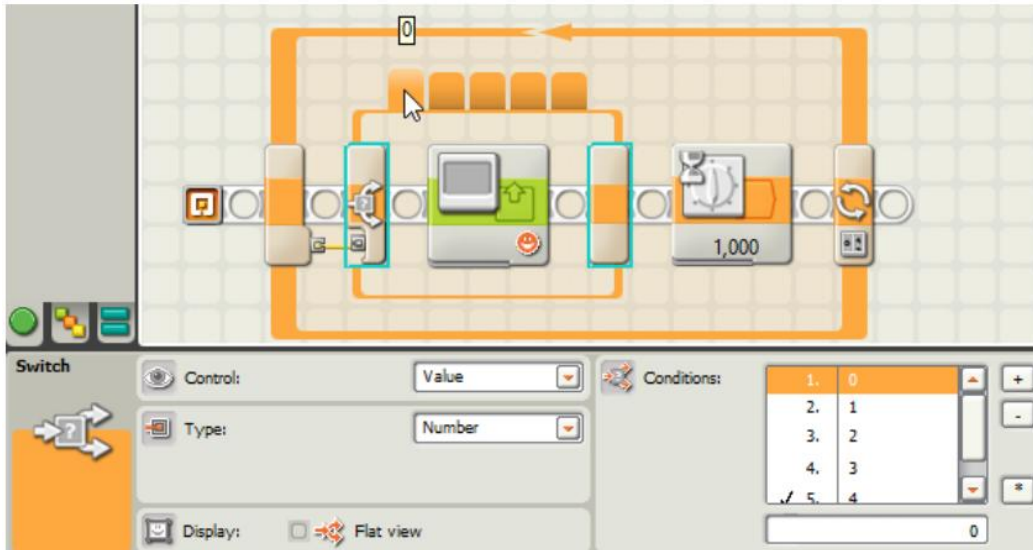


Рис. 3.4.55. Приклад використання блоку Switch із чисельним розгалуженням

При обробці складних логічних умов може виникнути необхідність у застосуванні вкладених розгалужень.

Приклад 5.4. У приклад 5.1 необхідно додати умову, за якою робот повинен зупинитися також і при виявленні зеленого кольору.

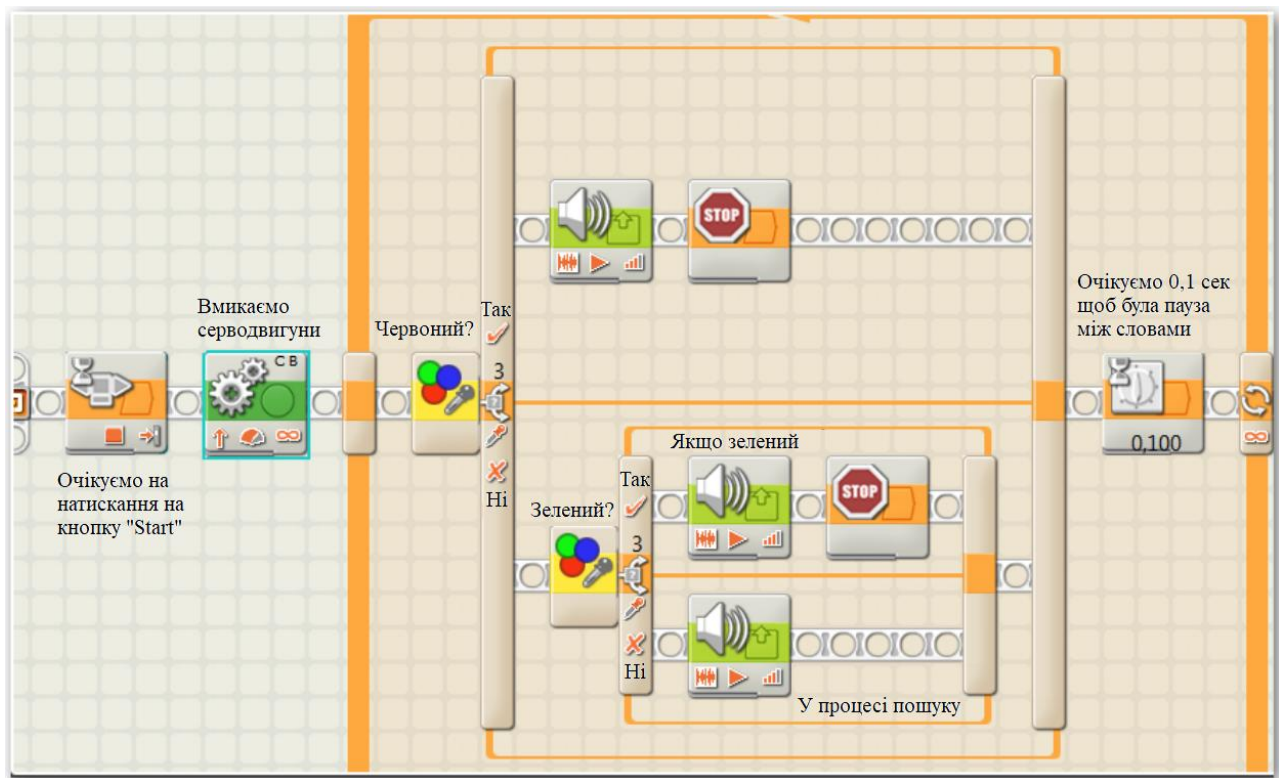


Рис. 3.4.56. Графічна інтерпретація програми для прикладу 5.4

Завдання 5.1. Необхідно скласти програму для управління роботом за допомогою давача дотику (однією кнопкою без фіксації). При запуску програми робот повинен очікувати на натискання кнопки **Start**, після чого розпочати рух вперед. При натисканні на кнопку робот повинен зупинитися і почати рух назад, при відпусканні кнопки – зупинитися і почати рухатися вперед знову.

Завдання 5.2. Необхідно скласти програму для управління роботом за допомогою двох давачів дотику (двома кнопками без фіксації), які використовуються у якості пульта управління. Алгоритм руху наступний. При натисканні на дві кнопки одночасно робот повинен переміщуватися вперед. Якщо ліву відпустити, а права залишається натиснутою – робот повинен повернути вправо, якщо залишається натиснута ліва – повернути вліво. Якщо відпустити обидві кнопки, робот має зупинитися.

Завдання 5.3. Необхідно скласти програму для робота-дослідника, який пересувається різнокольоровим полем і озвучує назву того кольору, над яким здійснює рух. Робот повинен "вимовляти" тільки основні кольори, для озвучування яких у пам'яті зберігається відповідний звуковий файл. Він має зупинитися тоді, коли доїде до червоного кольору. У якості оператора вибору необхідно використати блок **Switch**.

Використайте підказку: усі основні кольори закодовані цифрами 0...5.

Завдання 5.4. Необхідно скласти програму, за допомогою якої робот буде переміщуватися і залишатися на заданій відстані від рухомої вертикальної площини (дошка, книга, журнал і т.п.). Комплектація робота – два мотори і давач ультразвуку, який спрямований вперед. Рухома вертикальна площина розміщується перед роботом, який при постійному русі повинен зберігати відстань до цієї площини приблизно 20 см.

Використайте підказку: скористайтеся завданням 4.4 і прикладом 4.1.

3.6.8. Створення блоків користувача

Підпрограми: My block

Мова NXT-G дозволяє програмісту створювати власні блоки, схожі на блок Move. Це дозволяє значно скоротити той програмний код, який знаходиться в області перегляду. Для мови NXT-G це дуже важливо, оскільки на екрані можна відобразити тільки невелику частину коду і часто необхідно використовувати прокрутку.

Блоки користувача можна використовувати багаторазово і за участі створювати нові блоки. У матеріалі цього розділу буде розглянуто побудову найпростішого варіанту блоків – без вхідних і вихідних параметрів, а також розробка більш складних структурних елементів.

Завдання 6.1. Необхідно визначити і записати, на скільки градусів необхідно повернути колеса робота, щоб сам робот повернувся би на 90° .

Завдання 6.2. Необхідно скласти програму переміщення робота по квадрату за наступною схемою:

- Відтворити 4 рази:
 - Рух вперед;
 - Поворот робота на 90° (необхідно використати результати завдання 6.1).

Завдання 6.3. Скласти програму для переміщення робота за траєкторією, яка зображена на рис. 3.4.57.

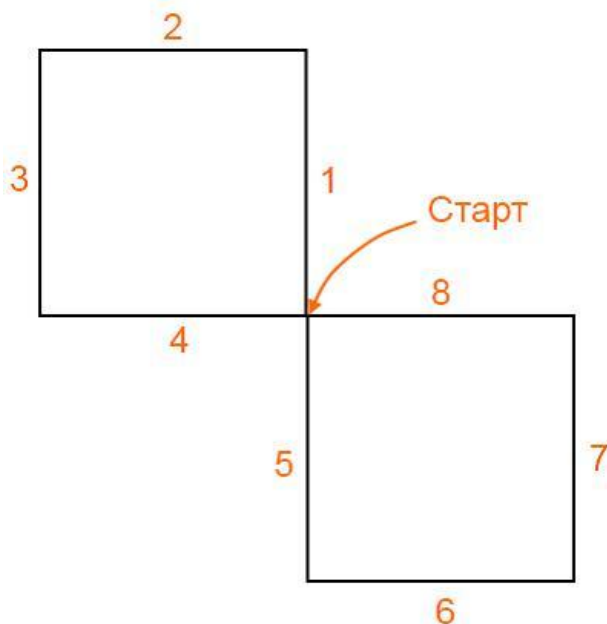


Рис. 3.4.57. Траєкторія переміщення робота до завдання 6.3

Умови завдання: робот розпочинає рух у центрі фігури із порядком проходження сторін за вказаними номерами. Схема програми:

- Відтворити 2 рази
 - Квадрат
 - Поворот на N°. (для визначення N° скористайтесь результатами завдання 6.1).

Приклад 6.1. Необхідно скласти аналогічну програму із використанням підпрограми (My block). Для цього:

1. виділити мишею цикл переміщення роботу по квадрату;
2. обрати пункт меню **Edit \ Make A New My Block** (Редагування \ Створити новий блок);
3. у вікні **My Block Builder** (Конструктор блоків) у полі **Block Name** ввести ім'я блоку **Square** (*виключно буквами англійського алфавіту*) і опис блоку у полі **Block Description** (можна українською мовою), наприклад "Алгоритм переміщення по квадрату";
4. натиснути кнопку **Next** (Далі);
5. на цьому кроці необхідно створити значок для Вашого блоку за допомогою переміщення готових піктограм у верхню область вікна (можна використати декілька піктограм);
6. натиснути кнопку **Finish** (Кінець);
7. у результаті описаних дій на листі програми замість циклу буде зображений Ваш блок під назвою **Square**. Якщо здійснити на ньому подвійний клік, то на сусідній вкладці (на новому листі) відкриється вміст блоку, який можна редагувати за необхідністю. Власні блоки можна обрати з палітри **Custom palette** (Моя палітра).

Зверніть увагу: обов'язковими з перелічених кроків створення підпрограми є 1, 2, 6. Однак виробник **NXT-G** радить виконувати всю послідовність дій і особливу увагу приділяти опису підпрограми. Це зазвичай вважається правилом гарного тону, допомагає програмісту швидко зрозуміти або нагадати призначення програми та її параметрів. Тому в даному навчальному матеріалі така дія вважається необхідною.

Отже, програма переміщення за "вісімкаю" (рис. 3.4.57) може виглядати так, як це зображено на рис. 3.4.58 або 3.4.59.

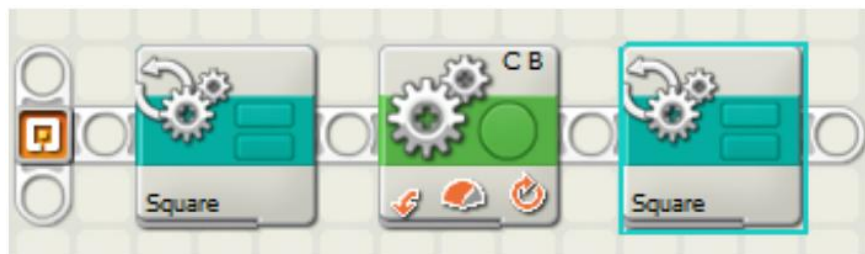


Рис. 3.4.58. Переміщення робота по квадрату (використовується підпрограма)

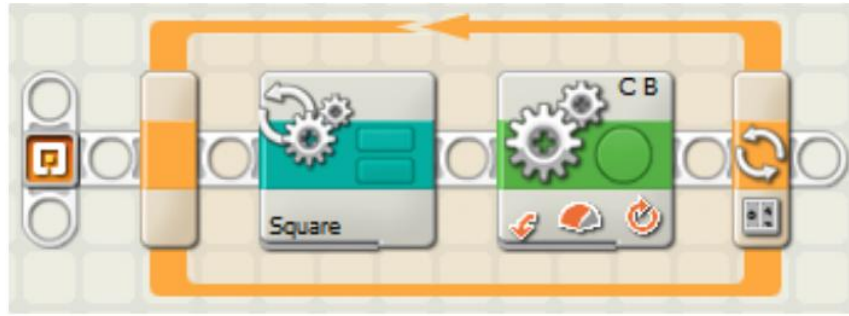


Рис. 3.4.59. Переміщення робота по квадрату (використовується підпрограма із циклом)

Завдання 6.4. Порівняйте роботу двох варіантів програми.

Зверніть увагу: отримати доступ до конструктора блоків можна також за допомогою панелі інструментів (рис. 3.4.60).

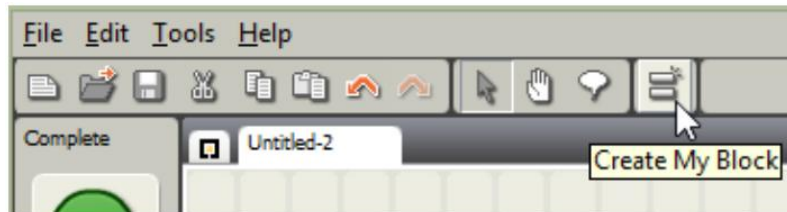


Рис. 3.4.60. Кнопка Create My Block на панелі інструментів

Завдання 6.5. Необхідно скласти програму для переміщення робота за траєкторією, що являє собою рівнобічний (правильний) трикутник (рис. 3.4.61).

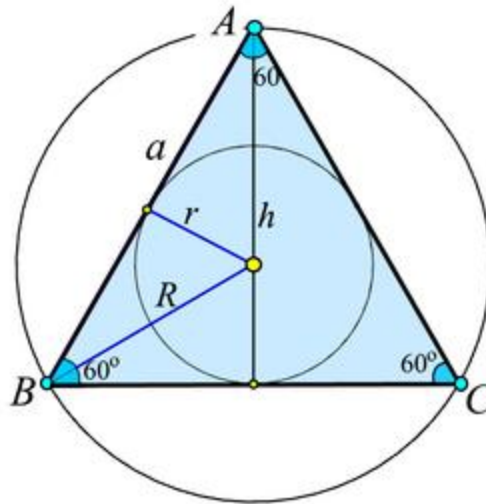


Рис. 3.4.61. Рівнобічний трикутник

Завдання 6.6. Необхідно скласти програму для переміщення робота, за якою він при русі "змальовує" мозаїку (рис. 3.4.62). Для наочності необхідно підкласти на стіл листи паперу і закріпити на роботі маркер, який спрямований вниз. Використайте результати завдання 6.5.

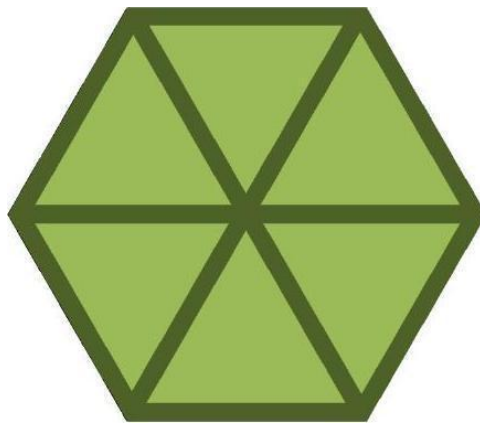


Рис. 3.4.62. Траєкторія "мозаїка" для завдання 6.6

Процедури із параметрами у NXT-G

З метою збільшення гнучкості коду у програми, які реалізуються мовою NXT-G, можна передавати як вхідні, так і вихідні параметри. Вхідні (початкові) параметри програма отримує для використання їх у своїй роботі. Вихідні параметри являють собою результат виконання підпрограми, вони використовуються для подальшої роботи і можуть бути вхідними даними для інших підпрограм.

Приклад 6.2. Необхідно скласти програму для переміщення робота по квадрату. Вхідним параметром є довжина сторони квадрату, яку необхідно задати випадковим числом у межах від 1 до 5 обертів валу серводвигуна.

1. Для початку складіть самостійно просту програму, за якою робот пересуватиметься по квадрату із стороною в 1 оберт валу серводвигуна (процедура в даному випадку не використовується).
2. Після цього додаємо у програму новий блок **Random** (Випадкове значення) із меню **Data** (Дані) повної палітри (рис. 3.4.63).



Рис. 3.4.63. Блок генерування випадкових чисел із меню **Data**

3. Встановлюємо цей блок перед основним циклом програми (рис. 3.4.64).



Рис. 3.4.64. Створення процедури із вхідним параметром: крок 1

- У налаштуваннях блоку **Random** задаємо діапазон генерованих випадкових чисел у межах від 1 до 5 (рис. 3.4.65).

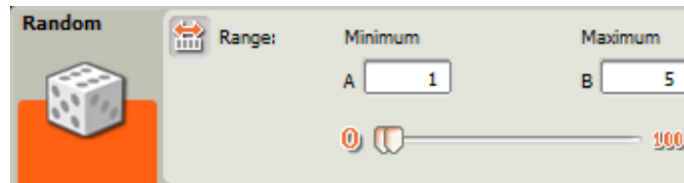


Рис. 3.4.65. Налаштування блоку **Random**

Для того, щоб робот переміщувався на таку кількість обертів, яку генерує блок **Random**, необхідно передати згенероване число на роз'єм **Duration** (Тривалість) блоку **Move** (при наведенні вказівника миші на роз'єм з'являється спливаюча підказка).

Зверніть увагу:

- При передачі даних на роз'єм **Duration** в налаштуваннях блоку **Move** не можна вказувати тип **Unlimited**. Тут задаються дані типу **Degrees** (градуси), **Rotations** (оберти) або **Seconds** (секунди). При цьому варіанти **Degrees** та **Rotations** обробляються однаково – як тип **Degrees**, тому обираємо **Degrees**.
- Кількість обертів, які повинен здійснити вал мотору, необхідно перерахувати в градуси із розрахунку, що 1 оберт валу дорівнює 360° .
- Встановлюємо у цикл блок **Math** (Математика) із меню **Data**, і налаштовуємо його на добуток (**Multiplication**) на 360° (рис. 3.4.66).

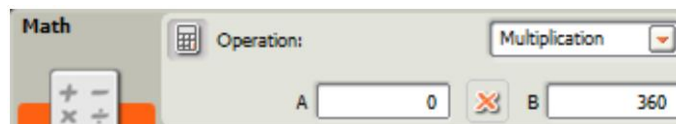


Рис. 3.4.66. Налаштування блоку **Math**

У блоці **Math** можна використати наступні шість операцій:

- **Addition** – додавання;

- Substraction – віднімання;
 - Multiplication – добуток;
 - Division – відношення;
 - Absolute Value – абсолютна величина;
 - Square Root – корінь квадратний.
5. З'єднуємо шиною даних вихідний роз'єм блоку **Random** із вхідним роз'ємом **A** блоку **Math**.
 6. Далі з'єднуємо шиною даних вихідний роз'єм **Result** блоку **Math** із вхідним роз'ємом **Duration** блоку **Move** (рис. 3.4.67), який власне відповідає за довжину грані квадрату (числові вихідні роз'єми позначаються символом "#").
 7. Перевіряємо роботу програми.

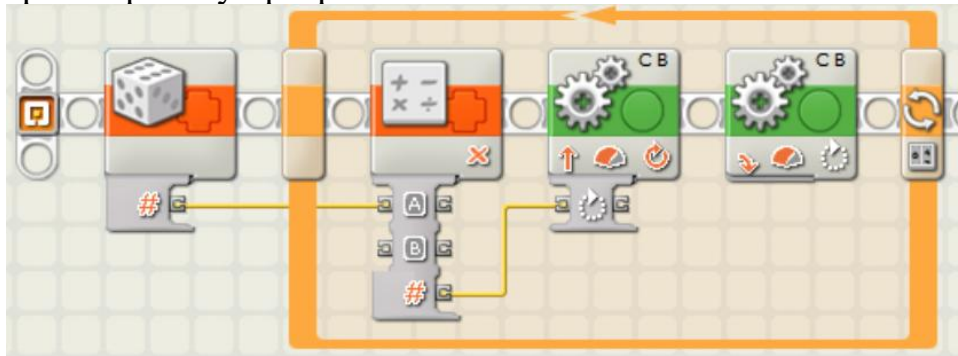


Рис. 3.4.67. Створення процедури із вхідним параметром: крок 2

8. Змінимо частину програми у вигляді процедури (підпрограми) із параметром. З цією метою необхідно виділити усе, що має потрапити у процедуру, окрім параметра (в даному випадку – окрім блоку **Random**), і натиснути кнопку **Create My Block** (Створити власний блок) на панелі інструментів або скористатися меню **Edit** (Редагувати) \ **Make A New My Block** (Створити новий власний блок).

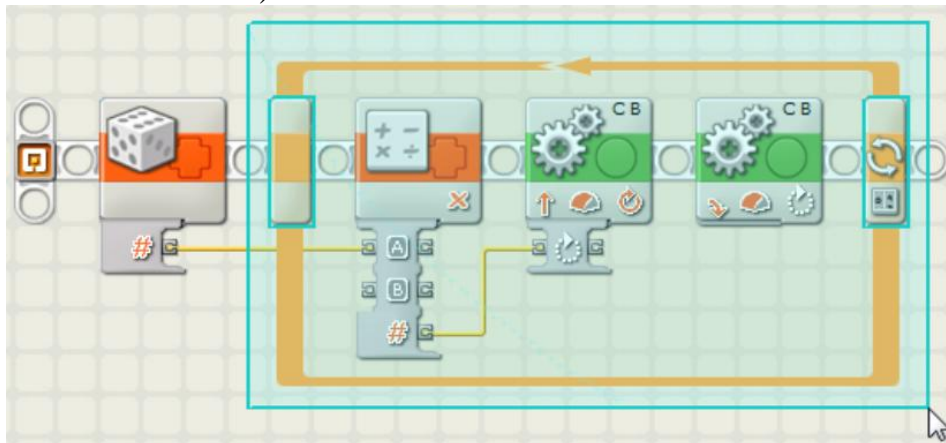


Рис. 3.4.68. Створення процедури (підпрограми) із вхідним параметром: крок 3

9. У вікні, що з'явилося, у відповідні поля вводимо назву програми (наприклад, **sq_param**, пам'ятаючи, що назва має бути записана виключно буквами латинського алфавіту) і опис програми українською мовою. Далі створюємо піктограму блоку і натискаємо кнопку **Finish**.

Зверніть увагу: у нового блоку з'явилися вхідний та вихідний роз'єми для передачі даних (рис. 3.4.69).

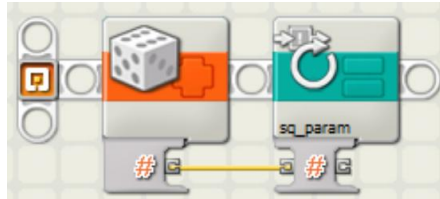


Рис. 3.4.69. Вхідний та вихідний роз'єми підпрограми

10. Здійснюємо подвійний клік на блоці **sq_param** – у новій вкладці з'являється вміст процедури. Змінюємо назву вхідного параметра на зручну, наприклад "Довжина сторони квадрату" (рис. 6.14).

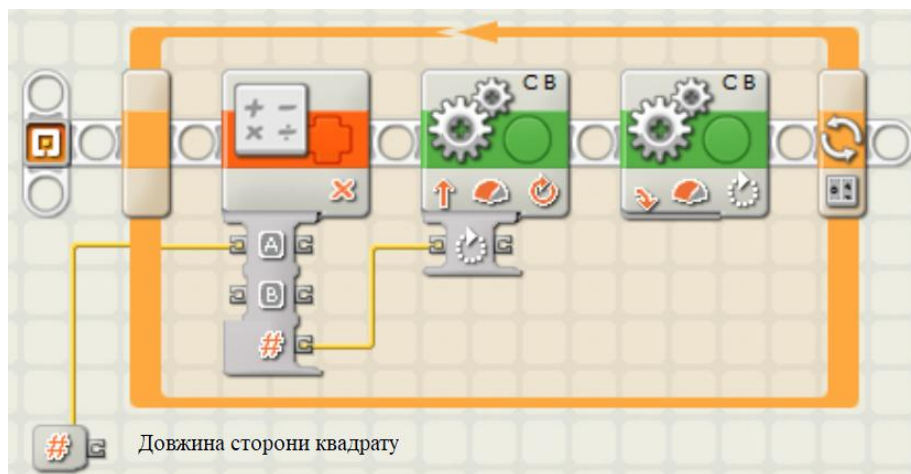


Рис. 3.4.70. Зміна назви вхідного параметра підпрограми

11. Зберігаємо файл процедури і закриваємо його.

12. Видаляємо блок **sq_param** із програми і знову додаємо його з палітри **My Blocks**.

Зверніть увагу: при наведенні вказівника миші на роз'єм блоку **sq_param** змінилася спливаюча підказка (рис. 3.4.71).

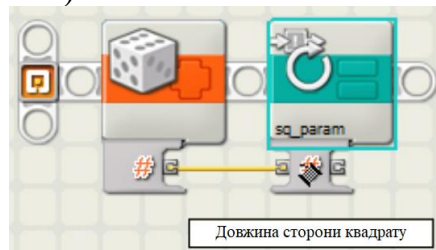


Рис. 3.4.71. Спливаюча підказка вхідного параметра підпрограми

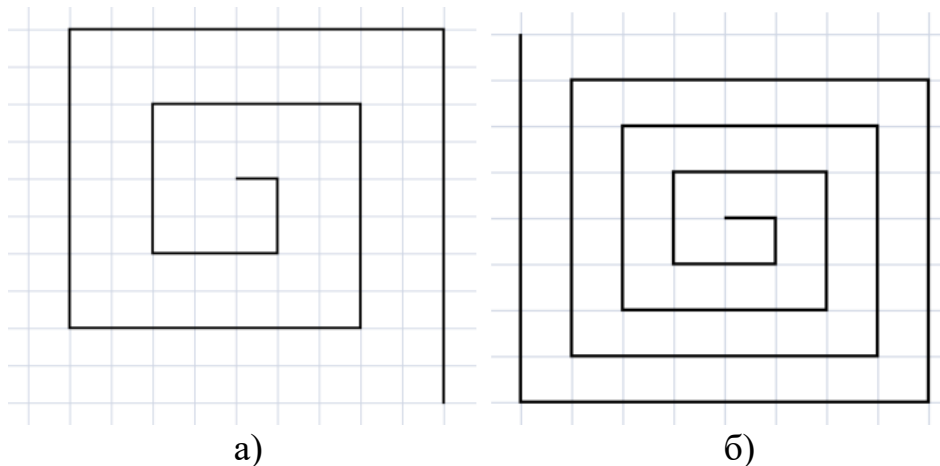


Рис. 3.4.74. Траєкторії "квадратної" спіралі для завдань 6.7 та 6.8

Завдання 6.8. Скласти програму для переміщення траєкторією "квадратної" спіралі, але із дещо зміненим алгоритмом: дві гілки довжиною 1 крок, потім дві гілки довжиною в два кроки, і т.д. (рис. 3.4.74, б). При цьому робот має виводити на дисплей повідомлення про довжину (у кроках) поточної гілки спіралі.

Завдання 6.9. Скласти програму для переміщення робота траєкторією правильного N-кутника. Вхідними параметрами є, по-перше, кількість сторін правильного N-кутника, і по-друге, довжина сторони вказаної фігури. Довжину сторони можна задати у кроках (1 крок – це повний оберт валу мотору на 360°) або у сантиметрах. Вказані параметри повинні бути згенеровані як випадкові значення і виводитися на дисплей блоку NXT у різних строках.

Давайте розглянемо, як створюється власний блок користувача із вхідними і вихідними параметрами.

Приклад 6.3. Необхідно створити блок для переведення величини в градусах у радіани за наступними умовами:

1. повинно бути наявним джерело даних;
2. результат повинен бути візуалізований;
3. необхідно скласти програму для обчислення радіан.

Отже блок, який розробляється, повинен мати вхідну і вихідну шину даних. Прийmemo, що градуси (початкові дані) будуть отримані від генератора випадкових чисел, а радіани (результат) буде виводитися на дисплей NXT. Отримаємо програму, із якої найбільшу цікавість виявляють математичні блоки добутку та відношення, налаштування яких зображене на рис. 3.4.80.

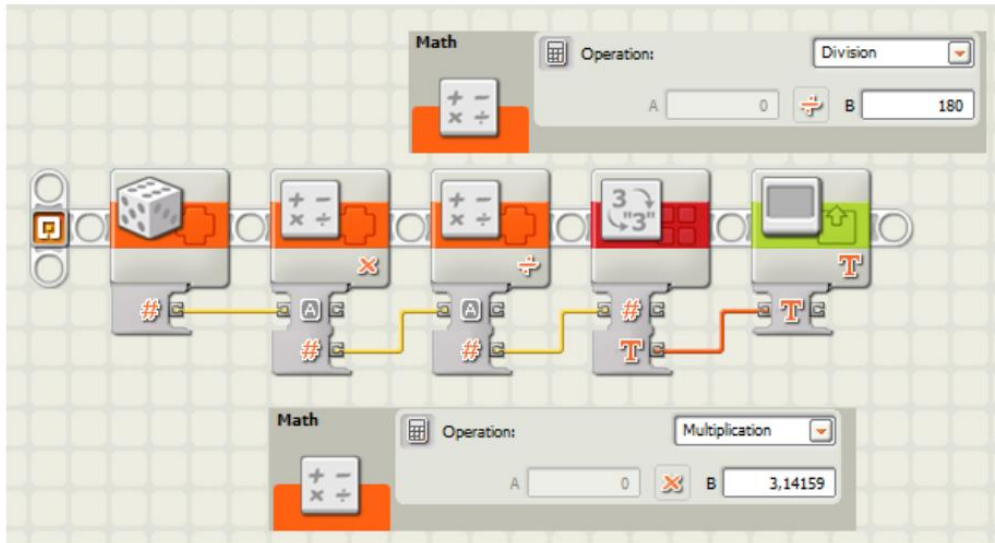


Рис. 3.4.80. Програма для переведення градусів у радіани

Побудуємо власний блок користувача із використанням блоків добутку і відношення. З цією метою виділимо два вказаних блоки і натиснемо на кнопку Create My Block на панелі інструментів, після чого вказуємо у відповідних полях ім'я блоку та його опис (рис. 3.4.81).

Зверніть увагу: тепер тіло блоку містить Result – вихідний параметр.

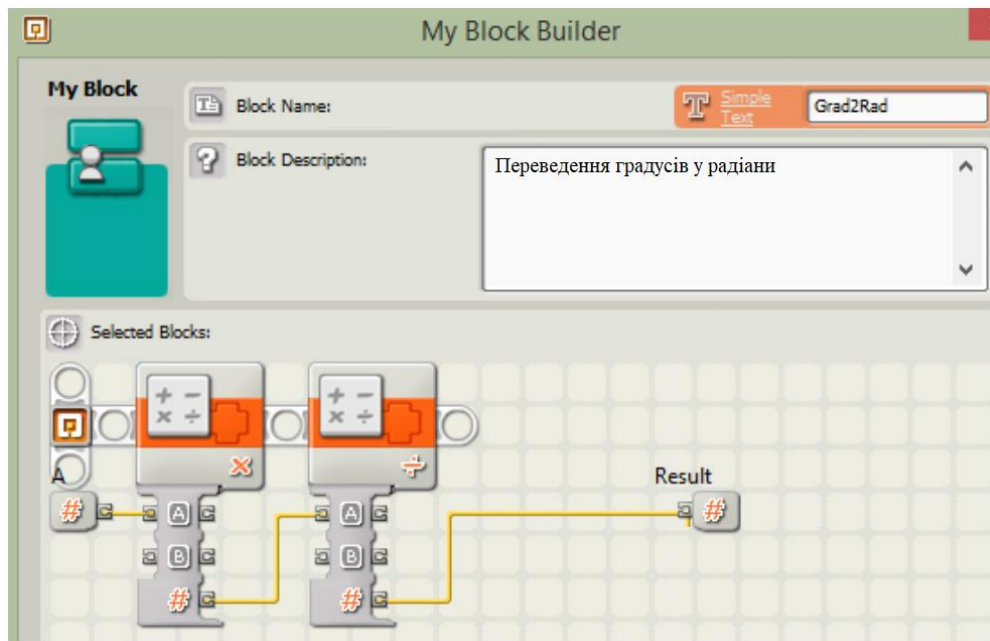


Рис. 3.4.81. Початок створення власного блоку користувача для переведення градусів у радіани

Після зміни назв вхідних і вихідних параметрів отримаємо кінцевий вигляд створеного блоку (рис. 3.4.82).

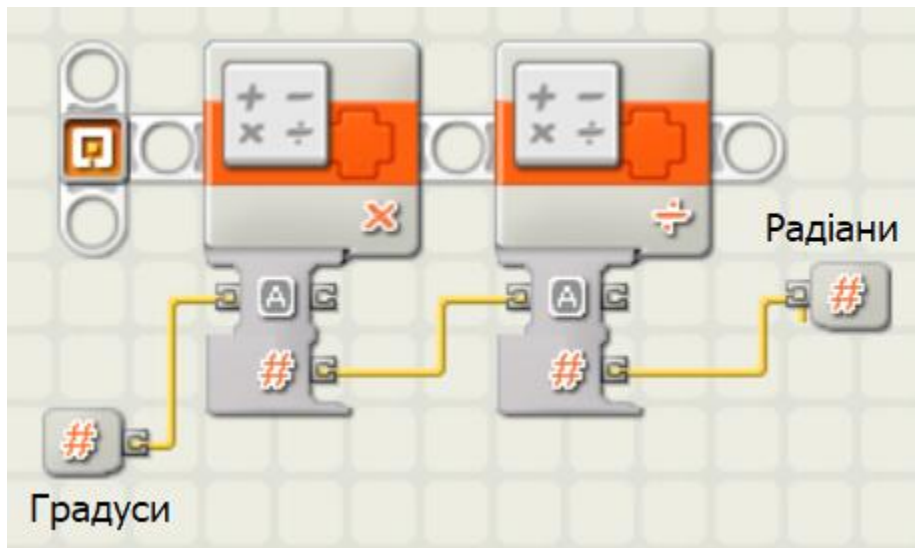


Рис. 3.4.82. Створений блок для переведення градусів у радіани

3.6.9. Змінні та константи

Змінні і константи використовуються у програмі для зберігання даних. Змінна являє собою область пам'яті, якій надане певне ім'я. Визначальною характеристикою змінної є її тип – змінна може містити дані виключно власного типу. Так, до складу NXT-G входять три основних типи: текстовий (**Text**), числовий (**Number**) та логічний (**Logic**), тому кожна змінна матиме один з названих типів. Відповідно, тип однозначно визначає, по-перше, розмір області пам'яті, яка відводиться для зберігання змінної, і по-друге, кількість операцій, яка може бути здійснена з цією змінною.

Змінні використовуються як для передачі даних від головної програми до власних блоків користувача, так і навпаки. Для цього і у власному блоці, і в головній програмі, необхідно визначити змінні однакового типу із однаковою назвою.

Константа, як і змінна, являє собою область пам'яті певного типу із певним ім'ям. Однак, якщо значення змінних можуть змінюватись у процесі відпрацювання програми, то константи визначаються перед запуском програми і під час її відпрацювання не можуть бути зміненими. Візуально у мові NXT-G різниця між змінними і константами реалізована таким чином, що змінні мають як вхідні так і вихідні роз'єми, а константи – тільки вхідні.

У більшості стандартних блоків наявні наскрізні роз'єми, що дозволяє складати програми без використання змінних і констант (як ми робили це до цього часу). Але існують такі задачі, вирішення яких не може бути здійснено без використання змінних.

Спочатку докладно опишемо послідовність роботи із змінними, для чого розглянемо наступний приклад.

Приклад 7.1. Необхідно скласти програму для підрахунку кількості паралельних чорних ліній, які зображені на білому полі і які робот перетинає при своєму русі. Нижче наведений покроковий алгоритм вирішення цієї задачі.

Крок 1. Використовуючи **завдання 3.3 (п. 3.4.5.1)** з наведеного вище матеріалу, складаємо програму переміщення робота до чорної лінії.

Крок 2. Перед початком руху до наступної лінії, робот повинен зберегти інформацію про вже знайдену чорну лінію, тобто "порахувати" її. Для зберігання цього підрахунку створимо змінну під назвою "Лінія" за допомогою блоку **Variable (Змінна)** із пункту меню **Data** повної палітри (рис. 3.4.83).

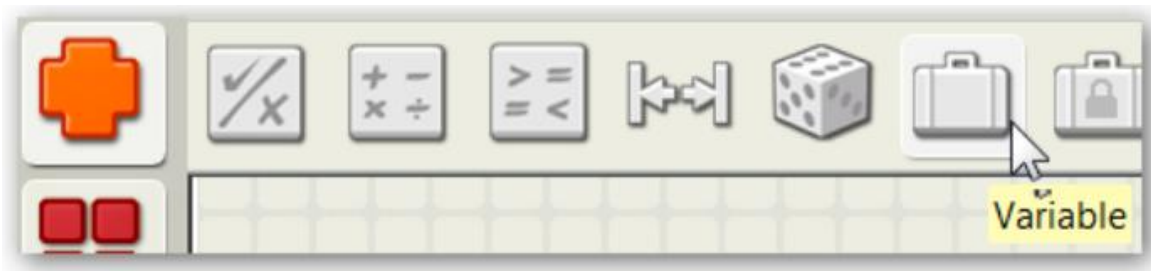


Рис. 3.4.83. Вибір блоку Variable із меню Data

У налаштуваннях блоку **Variable** за замовчуванням (рис. 3.4.84) середовище NXT-G пропонує створення трьох типів змінних: числового, текстового і логічного типу. Назви змінних задаються відповідно до їх типу, що часто є незручним, оскільки назва не пояснює призначення і сенс даних, які зберігаються у змінній. Тому доцільно спочатку визначити власні змінні, надати їм зрозумілі назву і тип, і вже потім використовувати.

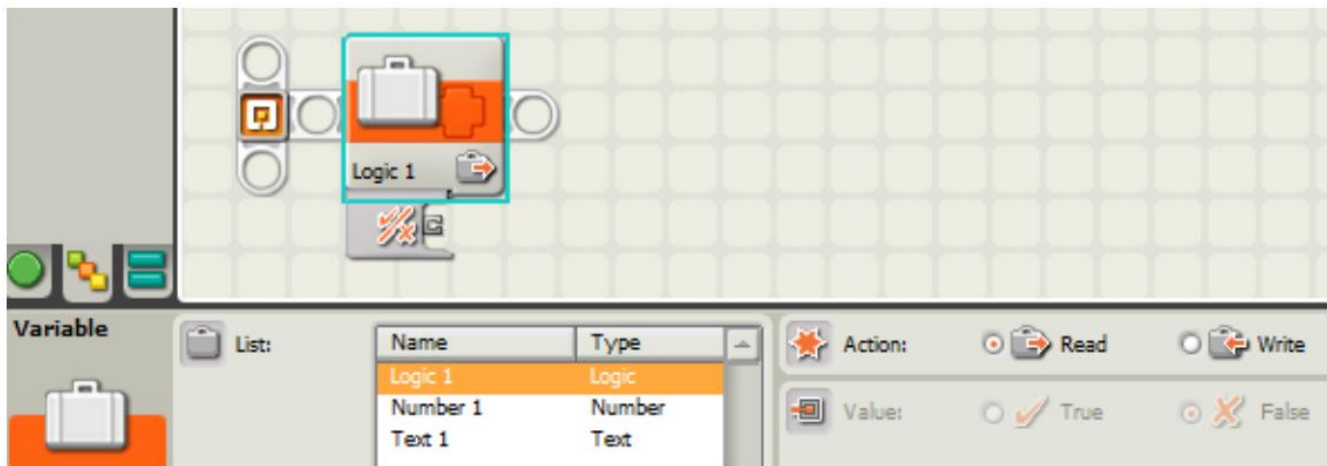


Рис. 3.4.84. Налаштування блоку Variable за замовчуванням

Відповідно до вказаного, створимо змінну числового типу під назвою "Лінії".
Для цього:

- у головному меню заходимо в пункт **Edit (Редагувати)**, і далі у **Define variables (Визначити змінну)**;
- відкривається вікно редактора змінних **Edit Variables** (рис. 3.4.85), в якому натискаємо кнопку **Create (Створити)**;
- хоча назву змінної можна задати довільно (як латинськими буквами, так і кирилицею), доцільно називати змінні у відповідності з метою їх використання в даному конкретному випадку. Отже, у полі **Name (Ім'я)** вводимо "Лінії" і задаємо **Data type (Тип даних)** - "Number" (Числовий);
- натискаємо кнопку **Close**.

Знову натискаємо на блок **Variable** у програмі і спостерігаємо за зміною налаштувань. Розглянемо їх більш детально.

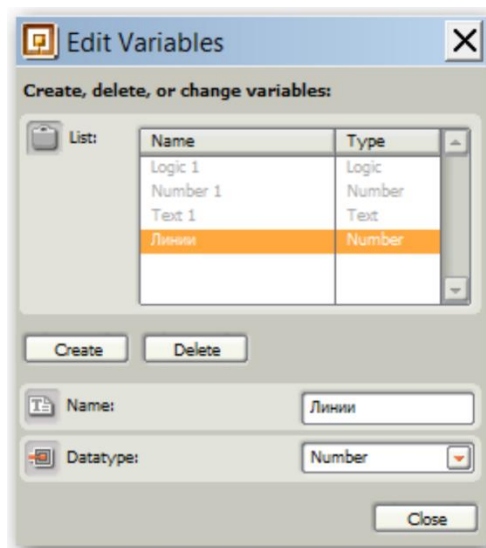


Рис. 3.4.85. Вікно редактора змінних

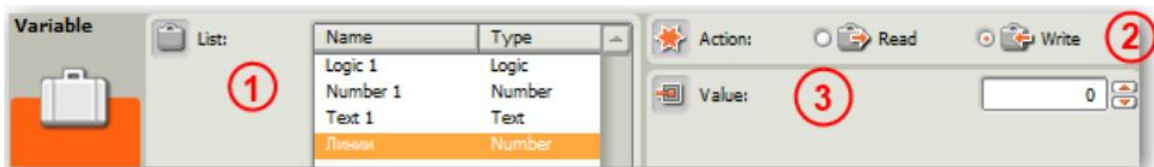


Рис. 3.4.86. Налаштування блоку **Variable** після створення змінної "Лінії" числового типу

Налаштування змінної (рис. 3.4.86) складаються з трьох полів:

1. Поле **List (Список)** містить список наявних змінних, де вказана їх **назва (Name)** і **тип (Type)**. Бачимо, що новостворена назва "Лінії" числового типу вже є у цьому списку.

2. **Action (Дія).** Це поле може набувати значення **Read (Зчитувати)** або **Write (Записувати)**. При виборі режиму **Read** існує можливість лише зчитувати попередньо записані у змінну дані. При виборі режиму **Write** дані можна як зчитати, так і записати у змінну. Для однієї змінної програма може містити декілька блоків, при чому в одному блоці дані записуються, в іншому зчитується, і/або навпаки.
3. У полі **Value (Значення)** можна задати початкове значення змінної. В залежності від обраного типу значення можуть змінюватися.

Для нашої програми у полі **Value** необхідно задати нульове початкове значення, оскільки робот ще "не знайшов" жодної лінії.

Крок 3. Додаємо у програму блок **Math**. Його необхідно налаштувати на додавання одиниці до змінної "Лінії" із наступним зберіганням нового значення у тій же змінній. Далі необхідно вивести це значення на екран (можна використати результати, отримані у *прикладі 4.3, п. 3.4.6.3*), а для продовження підрахунку і виводу нових значень необхідно додати у програму цикл. Отримаємо наступний код (рис. 3.4.87).

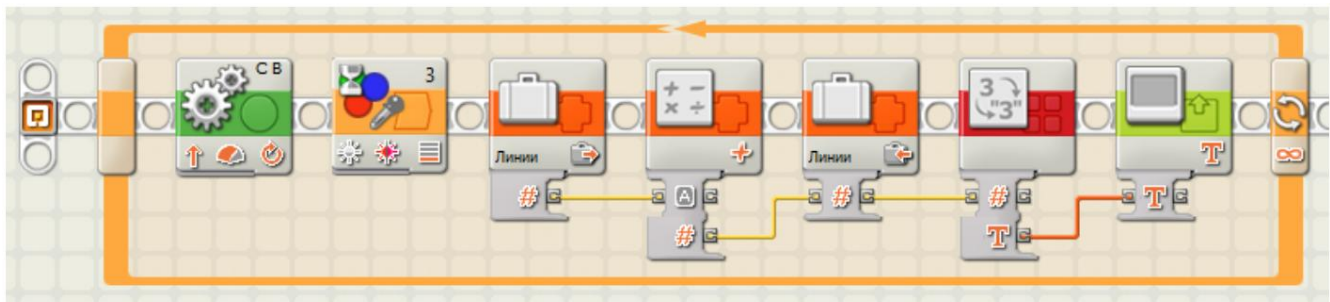


Рис. 3.4.87. Програма для підрахунку кількості чорних ліній

Завдання 7.1. Уважно розберіть приклад 7.1, побудуйте розроблену програму в середовищі NXT-G і запустіть її виконання. Перевірте, чи правильно робот підраховує кількість чорних ліній?

Зверніть увагу: іноді розробники стикаються із ситуацією, коли ширина лінії є такою, що робот за той час, коли давач кольору переміщується над цією лінією, встигає поррахувати її кілька разів. Отже виникає необхідність певним чином призупинити підрахунок. Одним із варіантів є використання затримки часу, але виникає питання про її тривалість (чим лінія є ширшою, тим триваліша має бути затримка). Бачимо, що цей варіант не є універсальним. Інший варіант базується на очікуванні закінчення ширини чорної лінії, тобто переміщення до більш світлої ділянки, і програма (рис. 3.4.87) набуде наступного вигляду (рис. 3.4.88).

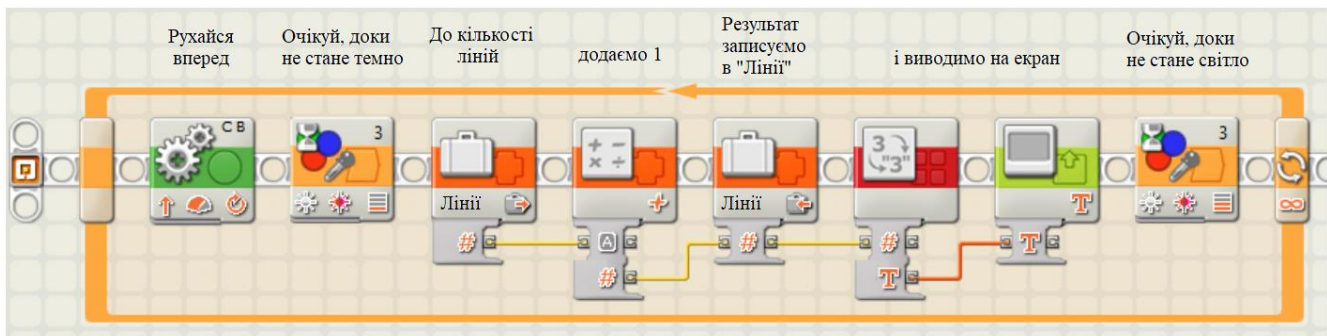


Рис. 3.4.88. Кінцевий варіант програми для підрахунку кількості чорних ліній

Завдання 7.2. Задайтеся питанням: коли саме робот зупиниться? Знайдіть спосіб програмної зупинки робота і спробуйте його реалізувати програмно.

Завдання 7.3. Складіть програму для зупинки робота на п'ятій лінії.

Завдання 7.4. Приєднайте кнопку (давач дотику) до вільного порту і складіть програму для підрахунку роботом кількості натискань на неї. Вигадайте власний спосіб зупинки програми за певною умовою. Поточна кількість натискань повинна виводитись на екран.

Для застосування констант в середовищі NXT-G передбачений блок **Constant** із меню **Date**.



Рис. 3.4.89. Звернення до блоку **Constant** із меню **Date**

Зверніть увагу на зображення блоку: замок на чемодані означає неможливість змінювати внутрішнє наповнення цього блоку.

Розглянемо налаштування блоку **Constant**:

- **Action – Choose from list (Вибрати зі списку)** або **Custom (Задається користувачем)**. За замовчуванням встановлений другий варіант, тобто користувач визначає константу в самому блоці за допомогою заповнення решти полів. Для відкриття доступу до варіанту **Choose from list**, необхідно визначити константу в редакторі констант (**Edit \ Define Constants**) заздалегідь, і зробити це так само, як ми робили це для змінної. Такі дії сприяють багаторазовому використанню тієї ж самої константи у різних частинах програми.

- **Data Type** – по аналогії зі змінною, для константи також можна обрати текстовий, числовий або логічний тип даних.
- **Value** – значення константи.
- **Name** – ім'я (назва) константи; після вводу саме воно автоматично відображається на блоці.

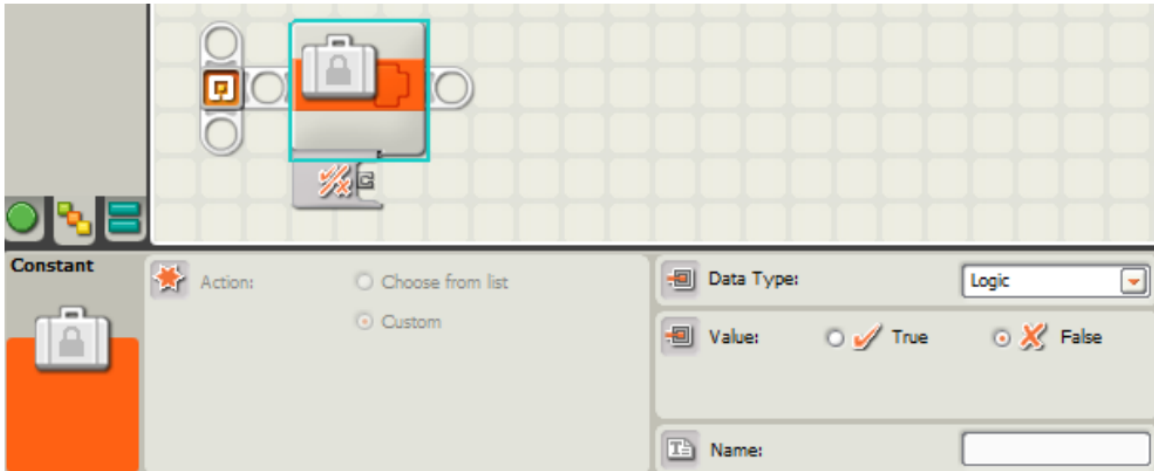


Рис. 3.4.90. Налаштування за замовчуванням блоку Constant

Робота із блоком Constant багато в чому аналогічна роботі із блоком Variables.

3.6.10. Потоки

У повсякденному житті люди часто виконують декілька справ одночасно. Так, наприклад, ми часто:

- йдемо певним маршрутом, оглядаємо краєвиди і розмовляємо по телефону;
- зустрічаючись із друзями, п'ємо каву, розмовляємо і переглядаємо контент із соціальних мереж;
- під час командних спортивних ігор взаємодіємо із, наприклад, м'ячем індивідуально і одночасно плануємо сумісні дії з членами команди в цілому, і т.п.

Для нас така поведінка є звичайною, і, якщо замислитися, насправді ми можемо виконувати ще більше дій одночасно. Це називають паралельним виконанням завдань, або роботою в режимі багатозадачності. Більшість систем людського організму пристосовані до режиму багатозадачності.

Сучасні комп'ютери працюють у режимі багатопоковості: програми складаються з декількох потоків (threads), що виконуються паралельно. Робот із навчального набору Lego Mindstorms NXT 2.0 із програмованим блоком NXT також передбачає виконання багатопотокових програм.

Насправді, багатопотоковість нами вже використовувалася раніше, наприклад, коли запускаються серводвигуни, вони продовжують свою роботу, а управління негайно передається далі на наступний блок програми. Таким чином, серводвигуни працюють у паралельному режимі, і одночасно з цим можуть відтворюватись звуки, опитуватись давачі і т.п.

Важливо розуміти і пам'ятати, що паралельність не тільки надає гнучкість програмі, але й потенційно може бути джерелом численних помилок. Більшість із них виникає тоді, коли у різних потоках (двох або декількох) виникає спроба одночасного використання одного і того ж самого джерела даних або ресурсу. Наприклад, один потік записує деяке значення у певну змінну, а інший потік одночасно намагається записати інше значення в ту саму змінну. Або в одному потоці ми запускаємо двигун вперед, а в іншому одночасно запускаємо той самий двигун назад. Ще одна поширена проблема виникає під час несинхронного запуску або зупинки одного з паралельних потоків. Описаним вище випадкам характерна загальна проблема синхронізації, а саме: як організувати сумісний доступ із різних потоків до спільного ресурсу. Тому, для запобігання подібних ситуацій, складання програм вимагає акуратності і уважності.

Приклад 8.1. Розглянемо задачу для робота, яку людською мовою можна назвати "Рухаємося і розмовляємо": він повинен доїхати до чорної лінії і зупинитися, сповістивши про це. Під час руху робот повинен постійно вимовляти певну фразу, наприклад, "Tracking object". А для того, щоб поведінка робота була більш подібною до людської, він має вимовляти обрану фразу не через однакові проміжки часу, а через випадкові.

Для вирішення поставленої задачі програму оформимо у вигляді двох потоків команд і використаємо блок **Timer** із меню **Sensor**, за допомогою якого здійснюється відлік часу у мілісекундах.



Рис. 3.4.91. Звернення до блоку Timer із меню Sensor

Після його додавання на лист програми здійснимо його налаштування (рис. 3.4.92). Із порту ① зчитується кількість мілісекунд, які пройшли з моменту перезапуску таймера. Поле ② (Timer) вказує на номер таймера (усього в одній програмі можна використати до трьох таймерів). Поле ③ (Action) показує дію, яку можна виконати: зчитати дані із таймера (Read) або скинути таймер в нуль (Reset). Інші порти і поля поки що не знадобляться.

Крім того, для розуміння програми необхідно нагадати провила роботи із циклом, вихід з якого відбувається за логічною умовою (п. 3.4.6.5).

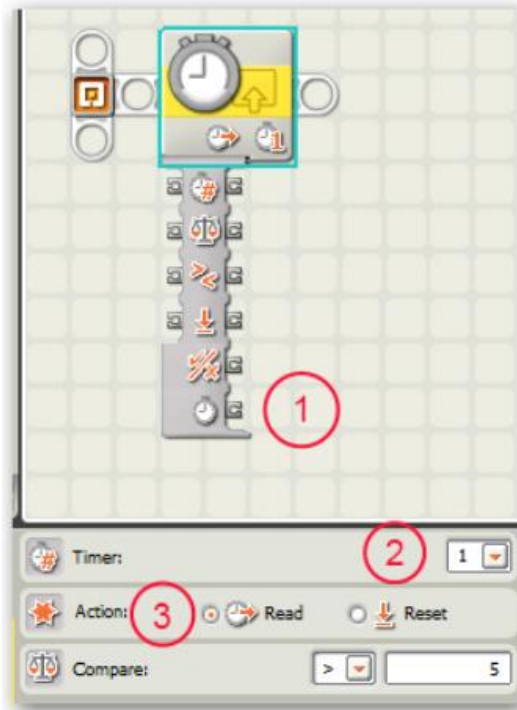


Рис. 3.4.92. Налаштування блоку Timer із меню Sensor

Розглянемо будову програми.

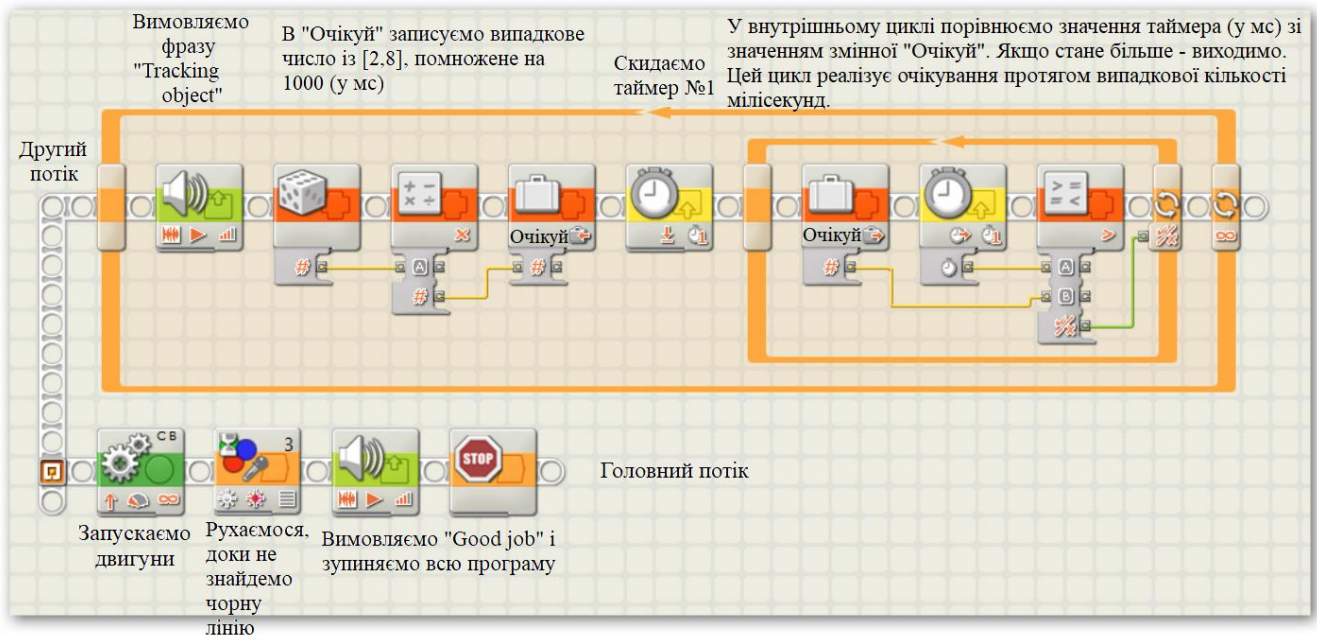


Рис. 3.4.93. Програма до прикладу 8.1

У прикладі 6.2 (п. 3.4.8.2) була розглянута можливість створення нового потоку. Для створення нового потоку програм, який не виходить із початкової точки, можна встановити вказівник миші на необхідній ділянці основного ланцюга, натиснути клавішу Shift і перемістити вказівник миші вгору або вниз. Після цього можна розміщувати нові блоки (рис. 3.4.94).



Рис. 3.4.94. Створення нового потоку програм

Слід також зазначити, що інколи буває зручно спочатку створити підпрограми для нового потоку, і тільки після цього підвести до нього сіру смугу.

Завдання 8.1. Прочитавши коментарі, уважно проаналізуйте роботу програми (рис. 3.4.93) взагалі та налаштування кожного окремого блоку. Наберіть програму в середовищі NXT і запустіть її виконання. Дайте обґрунтовану відповідь на запитання: чи можна було реалізувати цей алгоритм в одному потоці?

Завдання 8.2. Підключіть дві кнопки (давачі дотику) до робота. Складіть програму, яка у двох потоках (один потік на кожну кнопку) підраховує сумарну кількість натискань на кнопки і виводить її на екран (тобто якщо на одну з кнопок натискали 4 рази, а на іншу 3, на екрані повинне з'явитися число 7). Відображення числа на екрані необхідно реалізувати у третьому потоці.

Завдання 8.3. Змініть попередню програму таким чином, щоб замість суми на екран виводився би добуток кількості натискань на кнопки (тобто $3 \times 4 = 12$).

Завдання 8.4. Додайте у попередні програми умову: при перевищенні певного значення (для суми у задачі 8.2 та добутку в задачі 8.3) робота програми повинна бути завершена.

3.6.11. Керування переміщенням робота за допомогою системи із від'ємним зворотнім зв'язком

В сучасних науці, техніці та виробництві задача керування є надзвичайно актуальною. Наприклад, існує об'єкт керування (ОК) або система, стан яких необхідно підтримувати на певному рівні. Для цього використовується регулятор (Р), який, по-перше, отримує інформацію у часі t про стан об'єкту або системи від давача (давачів) (Д), і по-друге, розраховує і формує керуючий сигнал $U(t)$ для виконавчого механізму (ВМ), який за допомогою регулюючого органу (РО) змінює кількість енергії або матеріального потоку, що впливають на стан об'єкту (системи). Така схема керування називається системою із від'ємним зворотнім зв'язком, оскільки при відхиленні від рівноваги регулятор намагається повернути систему або об'єкт у заданий стан.

У даному пункті ми розглянемо два алгоритми керування – позиційний і неперервний, і відповідно два види регуляторів – більш простий релейний і більш стійкий пропорційний.

Релейний регулятор

Розглянемо улаштування системи керування автономним роботом на базі найпростішого релейного регулятора.

Приклад 9.1. Переміщення робота вздовж стінки.

Необхідно скласти програму для переміщення робота, який оснащений давачем ультразвуку, вздовж стінки із нерівностями (виступами і заглибленнями) на певній заданій відстані.

Для реалізації поставленої задачі необхідно, по-перше, заздалегідь продумати розташування давача (як зазначалося у попередніх пунктах, відгук цього давача є доволі повільним) – його необхідно встановити попереду корпусу робота. По-друге, робот необхідно встановити подалі від стінки (рис. 3.4.95), оскільки на незначних відстанях (5...7 см) покази давача ультразвуку мають значну похибку. У подальшій роботі ми також побачимо, що необхідно скоригувати і напрямок давача.

Алгоритм переміщення робота (рис. 3.4.96) може бути озвучений словами приблизно наступним чином:

1. рухатися прямо;
2. якщо відстань до стінки більша заданого значення, необхідно повернути до стінки;
3. якщо відстань до стінки менша заданого значення – повернути від стінки;
4. повторювати кроки 2 – 3 нескінченно або до настання певної події.

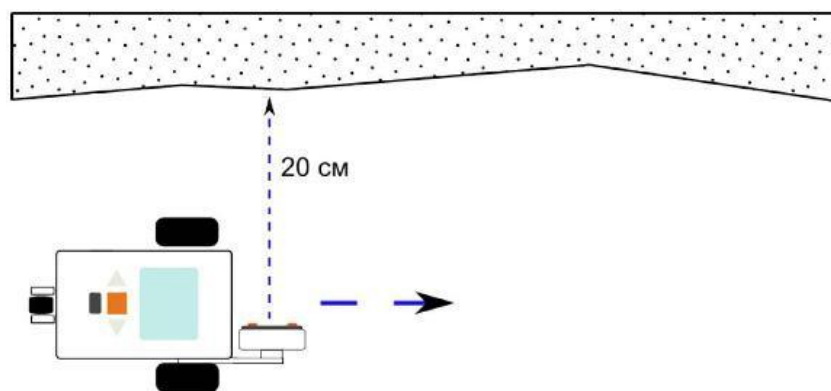


Рис. 3.4.95. Приклад розташування датча ультразвуку

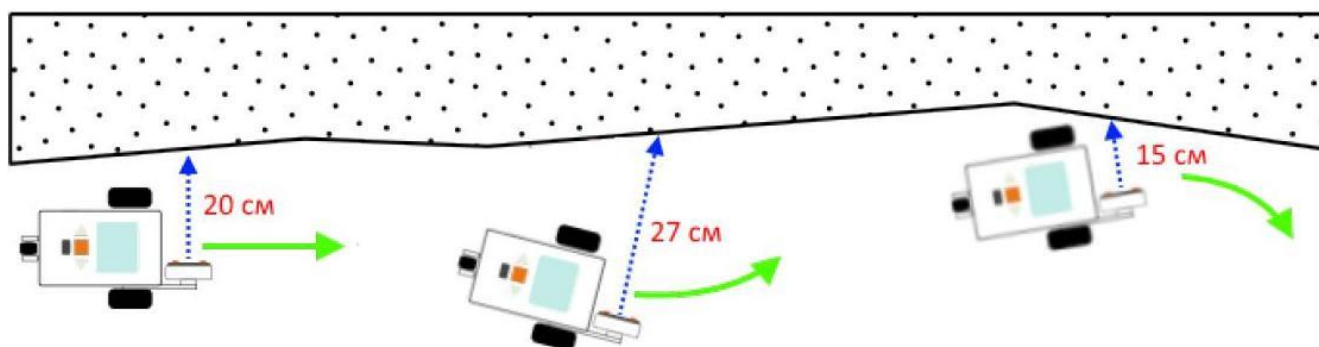


Рис. 3.4.96. Графічна ілюстрація руху робота вздовж стінки

Вочевидь, найпростішим рішенням поставленої задачі є використання позиційного (релейного) алгоритму управління роботом.

В електротехніці реле – це пристрій, який замикає або розмикає свої контакти із автоматичним поверненням у вихідне положення за певною умовою (реле напруги, струму, часу і т.д.). Наприклад, проміжне реле напруги із нормально розімкненими контактами замикає свої контакти при подачі напруги на котушку реле, і розмикає їх, коли напруга зникає. По відношенню до регулятора поняття "релейний" означає зміну поведінки системи лише для двох випадків – значення сигналу більше заданого (1) або менше заданого (2).

Якщо задати відстань до стінки 20 см, отримаємо програму, зображену на рис. 3.4.97. Бачимо, що після вмикання обох моторів запускається нескінченний цикл, за допомогою якого реалізується описаний словами алгоритм.

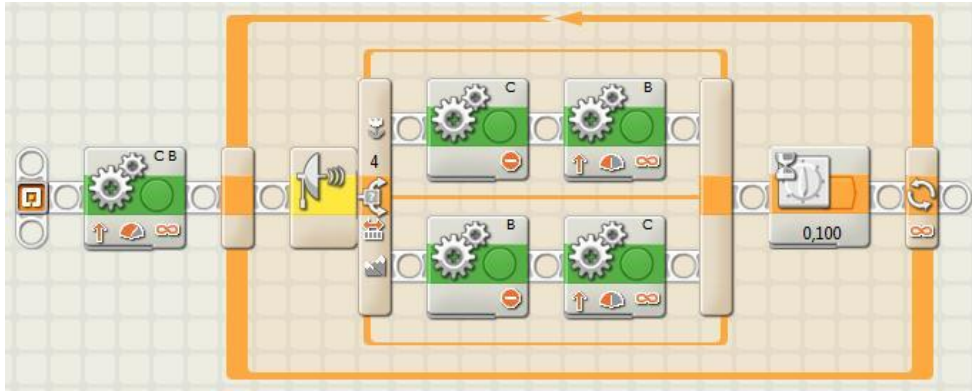


Рис. 3.4.97. Реалізація релейного алгоритму управління рухом роботу вздовж стінки

Характерною особливістю такого алгоритму вирішення поставленої задачі є "пульсуючий", "нишпорячий" рух: робот постійно повертає із однаковою інтенсивністю, незалежно від того, наскільки близько або далеко він опинився по відношенню до стінки. Тому траєкторія завжди буде зигзагоподібною, оскільки під час повороту робот завжди буде "коливатись" навколо середнього значення 20см.

Оскільки, як було зазначено, за швидкодією давач ультразвуку відноситься до "повільних", у циклі використовується певна затримка (0,1 с) для того, щоб інформація від давача встигала оброблятися блоком NXT. Фактично величина затримки визначає той час, на протязі якого робот рухатиметься незмінною траєкторією. Якщо збільшити час затримки, отримаємо більші колювання траєкторії. Раніше відмічалось, що затримку менше 0,06 с встановлювати недоцільно, оскільки давач ультразвуку не встигне здійснити вимірювання.

При застосуванні релейного (позиційного) алгоритму часто можуть виникати сходження з дистанції. Одна із причин полягає в обраному положенні встановлення давача ультразвуку: робот не зможе розрізнити положення, які симетричні відносно нормального положення. Тобто два положення (рис. 3.4.98) будуть ідентичними, і відстань до стінки в обох випадках буде більша за 20 см.

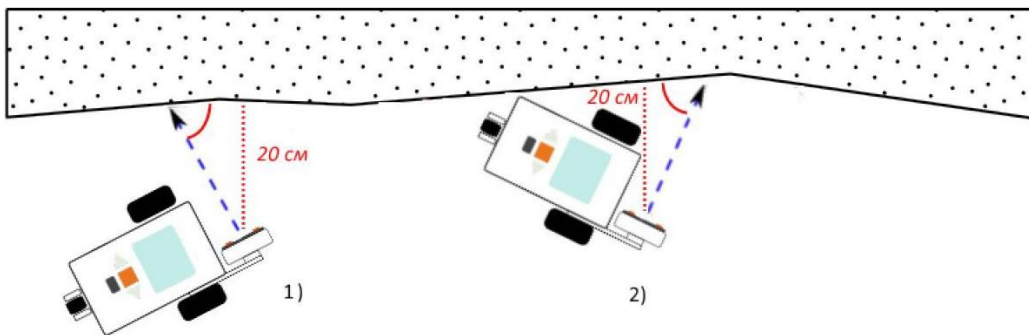


Рис. 3.4.98. Графічна інтерпретація проблематики прийняття рішення

У відповідності до другого кроку алгоритму робот повинен повернути до стінки (вліво), і якщо для позиції (2) це вірно, для позиції (1) це буде помилкове твердження, і робот зійде з траєкторії.

Вирішення цієї проблеми полягає у зміні способу встановлення датчика: його можна встановити не під кутом 90° суворо вліво (не перпендикулярно до напрямку руху робота), а під кутом 45° до напрямку руху (рис. 9.5).

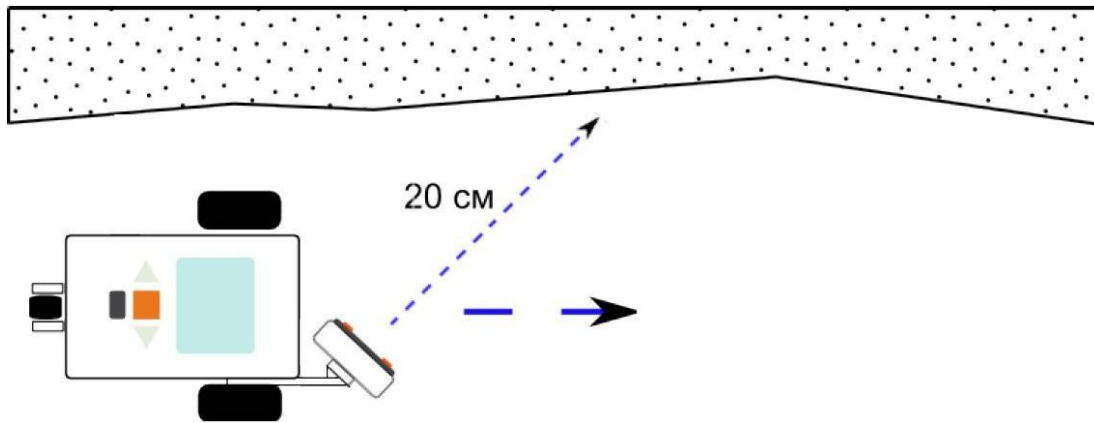


Рис. 3.4.99. Зміна способу встановлення датчика ультразвуку під кутом 45° до напрямку руху

Таким чином, можна запобігти виникненню ситуації, яка зображена на рис. 3.4.98. При "підрулюванні" роботом вліво відстань до стінки буде зменшуватися (при кутах повороту менше 45°), при "підрулюванні" вправо, навпаки, збільшуватися.

Слід зазначити, що не варто розраховувати на максимальну ефективність релейного алгоритму для переміщення робота вздовж стінки будь-якої форми: здійснення простого повороту на 90° може викликати у нього неочікувані труднощі. Отже, найбільшою перевагою релейного регулятора для поставленої задачі є простота його алгоритму. У наступному пункті цього розділу ми будемо розглядати інший за змістом алгоритм керування.

Завдання 9.1. Складіть програму переміщення робота вздовж стінки при різних способах встановлення датчика ультразвуку. Дайте відповідь на питання: у якому випадку рух є найбільш стійким?

Завдання 9.2. Порівняйте вирішення задачі 9.1 з прикладом 4.1. Спробуйте реалізувати релейний алгоритм переміщення вздовж стінки без використання розгалуження.

Завдання 9.3. Складіть програму переміщення робота за релейним алгоритмом із використанням розгалуження.

Пропорційний регулятор (П-регулятор)

З курсу "Теорія автоматичного керування" відомо, що до одним із алгоритмів керування є неперервний; він реалізується за допомогою пропорційних (П-), пропорційно-інтегральних (ПІ-), пропорційно-інтегрально-диференціальних (ПІД-регуляторів) і т.д.

Труднощі, із якими ми стикнулися при застосуванні релейного регулятора у попередньому пункті цього розділу, підштовхують нас до пошуку іншого прийняттого варіанту вирішення задачі керування. Одним із таких варіантів є пропорційний регулятор, коли керуючий вплив на мотори робота не є незмінним, а змінюється пропорційно до відхилення від заданої відстані до стінки. Іншими словами, чим більше відхилення, тим з більшою швидкістю повинні обертатись вали моторів для коригування траєкторії робота. Якщо давач реєструє задану відстань до стіни, робот переміщується прямо; невелике відхилення викликає незначне "підрулювання", відповідно збільшення відхилення викликає інтенсифікацію роботи моторів.

Приклад 9.2. Визначимо, як саме реалізується П-алгоритм переміщення робота вздовж стіни.

Для П-регулятора керуючий вплив $U(t)$ на серводвигуни робота в момент часу t розраховується за виразом

$$U(t) = K_P \cdot E(t), \quad (9.1)$$

де

- $E(t) = X_0 - X(t)$ – відхилення робота (давача) від заданого положення (інколи називають похибкою, Error);
- X_0 – задана відстань до стінки (за поставленою задачею 20 см);
- $X(t)$ – поточне положення давача;
- K_P – коефіцієнт підсилення (коефіцієнт пропорційності), значення якого завжди позитивне (більше 0).

Вочевидь, графічною залежністю для визначення керуючого впливу U в залежності від похибки E буде пряма лінія (рис. 3.4.100).

За графічною залежністю (рис. 3.4.100) легко визначається коефіцієнт підсилення K_P . Якщо вибрати на цьому графіку довільну точку, наприклад А(1;2), коефіцієнт K_P дорівнюватиме

$$K_P = U_A / E_A = 2 / 1 = 2.$$

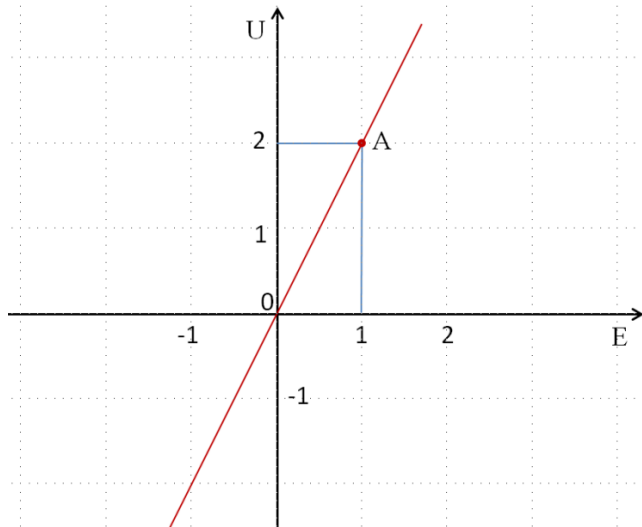


Рис. 3.4.100. Графік залежності $U(E)$ для П-регулятора

Похибка може бути як із знаком "плюс" (у тому випадку, якщо давач робота ближче до стінки, ніж задано), так і зі знаком "мінус" (якщо робот віддалився від стінки на більшу за задане значення відстань). Таким чином, керуючий вплив $U(t)$ може бути як позитивним, так і від'ємним.

Для робота, що переміщується вздовж стінки, за умови використання П-регулятора, потужність двигунів $PowerB(t)$ і $PowerC(t)$ у момент часу t визначається за виразами

$$PowerB(t) = Nm + U(t), \quad (9.2)$$

$$PowerC(t) = Nm - U(t), \quad (9.3)$$

де

- Nm – "нормальна" потужність двигунів, тобто така потужність, із якою повинні обертатися обидва двигуна при нульовому відхиленні від заданого курсу (в нашій програмі можна задати значення $Nm = 50$);
- $U(t)$ – керуючий вплив на двигуни, величина якого визначається за виразом (9.1).

Знак перед $U(t)$ для конкретного робота може змінюватись залежно від того, який саме двигун знаходиться з лівого боку, а який із правого.

Давайте визначимо сенс коефіцієнта пропорційності K_P . Вочевидь, поворот робота здійснюється в результаті того, що від потужності одного з двигунів керуючий вплив віднімається, а до потужності іншого, навпаки, додається. Таким чином, коефіцієнт K_P може підсилювати або ослаблювати вплив регулятора на двигуни: якщо $K_P > 1$ – здійснюється підсилення, якщо $K_P < 1$ – послаблення. Велике значення K_P зробить робот дуже чутливим до похибок, що призведе до різких рухів для коригування траєкторії. Мале значення K_P додасть плавності руху робота, але на крутих поворотах робот може "загубити" стінку і зійти з траєкторії.

Значення K_p , які якнайкраще підходять для кожного окремого випадку залежать від датчиків, які використовуються, складності траси, конструктивних особливостей робота, швидкості його переміщення (потужності двигунів). Величини $PowerB(t)$ і $PowerC(t)$ задаються у межах $[0; 100]$. Тому при значних відхиленнях (похибках) (відповідно, при великих керуючих впливах) потужність двигунів буде обмежуватися таким чином, щоб вона не виходила за межі вказаного діапазону. В цьому випадку П - регулятор не функціонуватиме коректно, оскільки не зможе компенсувати великі відхилення (похибки).

Отже, одна із особливостей П-регуляторів полягає в тому, що вони адекватно працюють тільки при невеликих похибках. І для врахування даної особливості необхідно забезпечити стінку без різких поворотів і задати невелику швидкість пересування. Конкретну складність траєкторії і швидкість переміщення можна отримати лише експериментально для кожного конкретного робота і конкретних умов оточуючого середовища.

Розглянемо, як у середовищі NXT-G реалізується алгоритм роботи П-регулятора за допомогою *вкладених процедур* (My blocks) відповідно наведених вище формул. Блок розрахунку відхилення (похибки) від заданої відстані, який можна назвати Error, має такі параметри:

- вхідні – поточне значення від датчика відстані $X(t)$;
- вихідні - похибка $E(t)$.

Блок **Error** у середовищі NXT-G виглядає наступним чином (рис. 3.4.101).

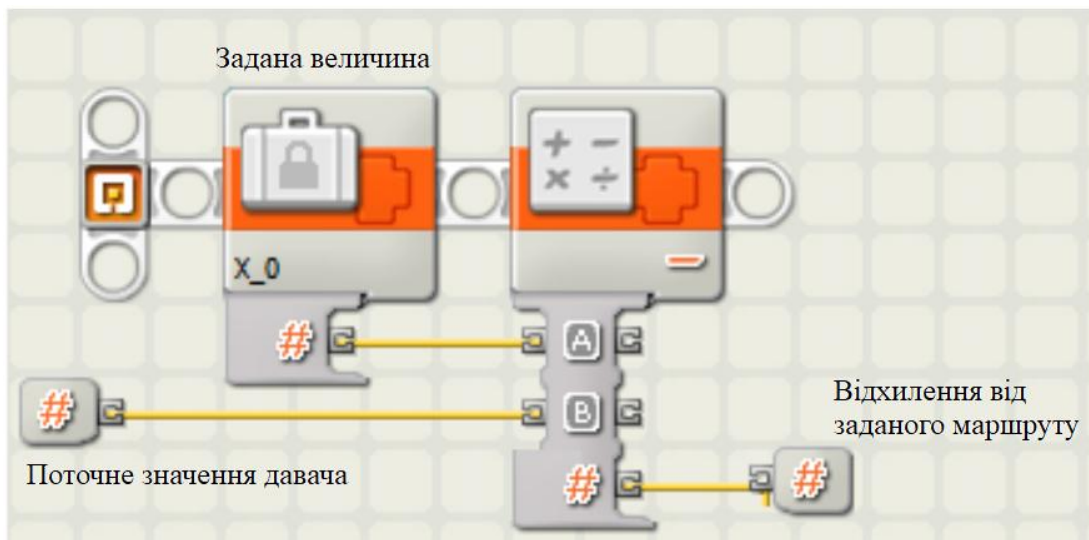


Рис. 3.4.101. Підпрограма для розрахунку похибки відхилення від заданої відстані

Блок буде використаний у підпрограмі для обчислення керуючого впливу $U(t)$. Вона має такі параметри:

- вхідні – поточні значення, виміряні датчиком $X(t)$;
- вихідні – керуючий вплив на двигуни.

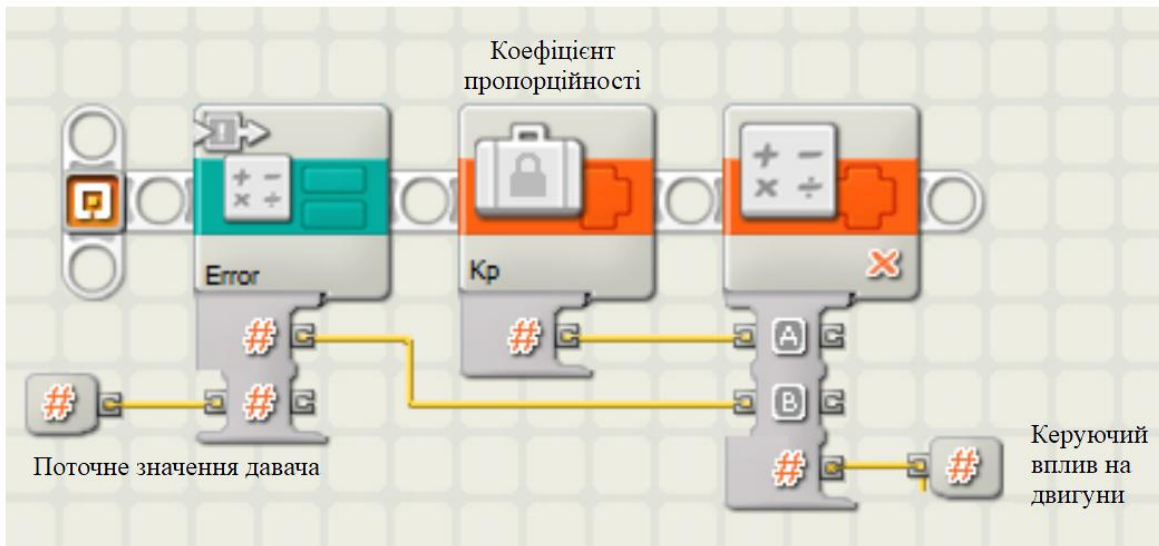


Рис. 3.4.102. Підпрограма розрахунку керуючої дії $U(t)$

Цей блок буде використаний для розробки блоку П-регулятора, який матиме такі параметри:

- вхідні – поточне значення давача;
- вихідні – потужність моторів В і С.

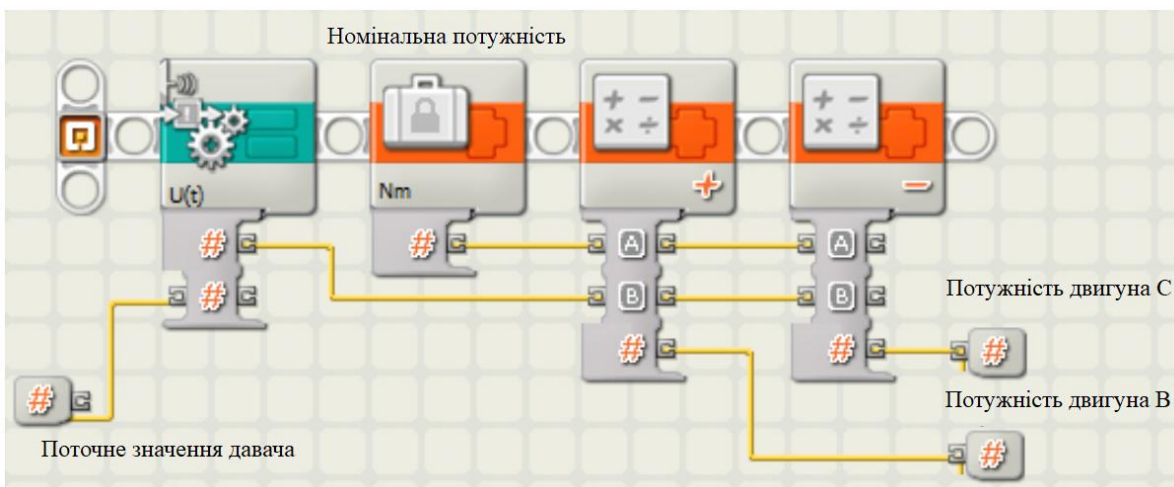


Рис. 3.4.103. Підпрограма обчислень П-регулятора

Повна програма, використання якої дозволяє реалізувати нескінченний рух вздовж стінки із використанням блоку П-регулятора і формул (9.2), (9.3), наведена на рис. 3.4.104.

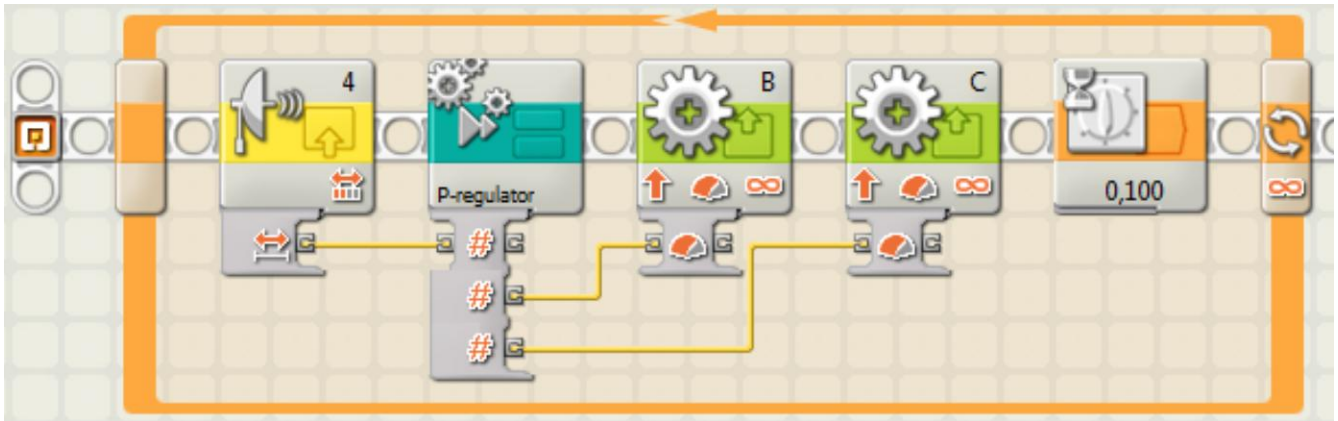


Рис. 3.4.104. Приклад програми переміщення вздовж стінки із застосуванням П – регулятора

Завдання 9.4. Реалізуйте та перевірте роботу розробленого алгоритму переміщення вздовж стіни із використанням П-регулятора. Для налаштування регулятора використайте підбір значення коефіцієнта K_p . Якщо виникне така необхідність, зменшіть значення нормальної потужності. Можна також змінити з горизонтального на вертикальне положення давача ультразвуку: гарний результат дає таке встановлення, коли "очі" давача розташовані один над одним.

Завдання 9.5. Складіть програму, де реалізований алгоритм переміщення вздовж лінії з використанням П-регулятора. Налаштуйте регулятор таким чином, щоб робот впевнено долав круті повороти.

Завдання 9.6. У своїй групі проведіть змагання на швидкісне проходження траси.

Завдання 9.7. Проведіть дослідження впливу збільшення або зменшення вдвічі підбраного коефіцієнту K_p на зміну роботи у програмі, отриману при виконанні завдання 9.5. Як ці зміни впливають на характер переміщення робота? Чи зміг робот подолати весь маршрут? Які ділянки маршруту робот долав із найбільшими складнощами?

Завдання 9.8. Складіть програму, де реалізований алгоритм слідування за рукою із використанням П-регулятора (використайте завдання 5.4). Регулятор необхідно налаштувати таким чином, що рух робота був якомога більш плавним.

Підводячи підсумок розгляду застосування різних алгоритмів керування і регуляторів можна відзначити, що використання П-регулятора не дозволяє розв'язати задачу повністю: рух робота сильно залежить від його конструкції, налаштування і т.п. Для отримання більш стабільного руху необхідно використовувати більш інтелектуальні алгоритми, до яких належать пропорційно-диференційний (ПД-) і пропорційно-інтегрально-диференційний (ПІД-) регулятори.

Кегельринг

У галузі робототехніки широкою популярністю користуються різноманітні змагання. Зокрема, велика їх кількість проводиться полях всевітньої олімпіади роботів (*англ.* World Robot Olympiad, WRO), де перед роботами постають складні інтелектуальні задачі. Найбільш відомими і видовищними змаганнями є перегони, танці роботів, пошук виходу із лабіринтів, сумо роботів та ін. Розглянемо один із видів змагань початкового рівня складності – "Кегельринг".

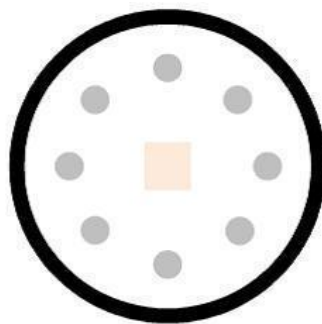


Рис. 3.4.105. Зовнішній вигляд рингу для змагання "Кегельринг"

Сутність змагання полягає у наступному. У центрі рингу (рис. 3.4.105) розміщений робот, оснащений давачами ультразвуку і кольору (або світла), і кеглі (алюмінієві банки). Робот повинен виштовхнути усі кеглі за межі рингу, і перемагає той робот, який виконає цю задачу із мінімальними затратами часу. Максимальний час на вирішення задачі – дві хвилини.

Вочевидь, поставлену задачу можна розділити на дві підзадачі:

1. пошук кеглі;
2. виштовхування кеглі за межі рингу.

Пошук кеглі.

Завдання 9.9. Скласти програму пошуку кеглі.

Для пошуку кеглі необхідно встановити на корпус робота направлений вперед давач ультразвуку. Алгоритм поведінки робота буде наступним:

1. робот, який знаходиться у центрі рингу, розвертається на місці до того часу, поки давач не зафіксує відстань до об'єкту, меншу за задане значення;
2. зупинка.

Виштовхування кеглі.

Якщо кегля знайдена, алгоритм її виштовхування наступний:

1. рухатися вперед до чорної лінії;
2. повернутися назад.

Постає питання: на скільки робот має повернутися назад? Вочевидь, на таку ж саму відстань, яку подолав до лінії. Тоді як визначити цю відстань?

У серводвигуни комплекту Lego Mindstorms вбудований датчик обертів, інформацію від якого можна за допомогою блоку **Rotation Sensor** із меню **Sensor** повної палітри. Вказаний блок здійснює підрахунок кількості градусів (із точністю до 1°) або кількість повних обертів валу двигуна.



Рис. 3.4.106. Звернення до блоку **Rotation Sensor** меню **Sensor**

Блок **Rotation Sensor** оснащений багатьма вхідними та вихідними роз'ємами. Для розв'язання даної задачі знадобиться лише один роз'єм, за допомогою якого можна отримувати інформацію про поточну кількість градусів або обертів, здійснених валом мотора.

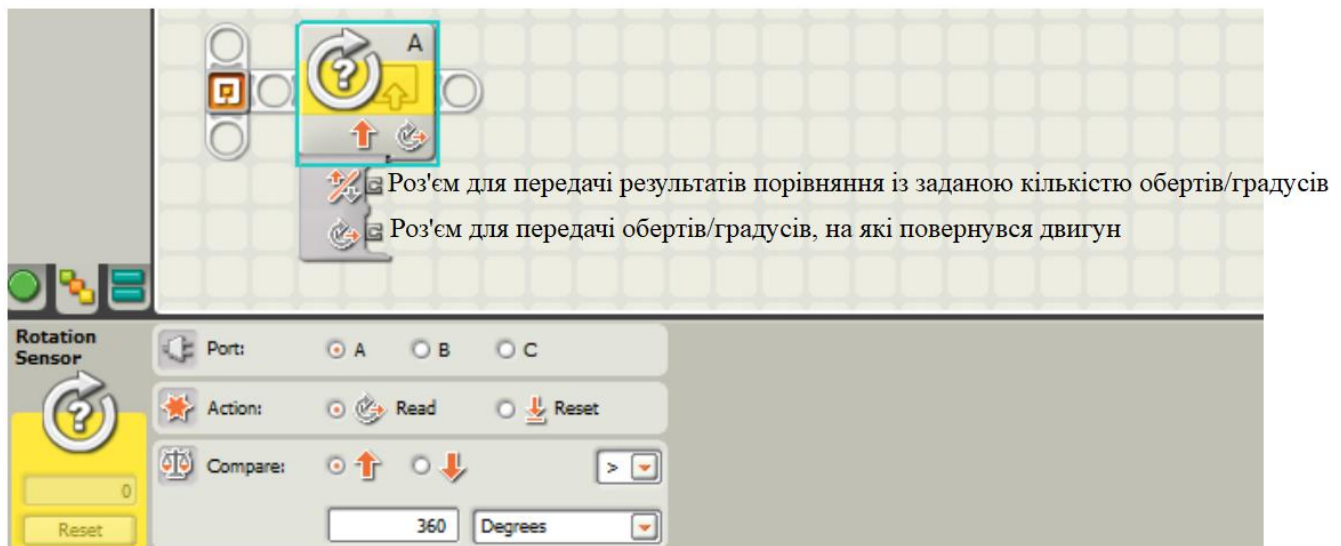


Рис. 3.4.107. Налаштування блоку **Rotation Sensor**

Поле **Action** задає виконувану дію: **Read** – зчитування даних з датчика; **Reset** – скидання в нуль показів датчика.

Приклад 9.3. Продемонструємо роботу датчика обертів. Для цього:

- складіть програму для роботи, зображену на рис. 3.4.108.
- **Зверніть увагу:** перший блок **Rotation Sensor** скидає в нуль покази датчика обертів, а другий функціонує в режимі зчитування;
- завантажте програму на програмований блок за допомогою USB-кабелю та кнопки **Download and Run**;

- не від'єднуючи USB-кабель, зробіть пару обертів колеса робота вперед/назад і спостерігайте за інформацією на екрані блоку;
- порівняйте покази на екрані з показами в полі зворотного зв'язку давача обертів.

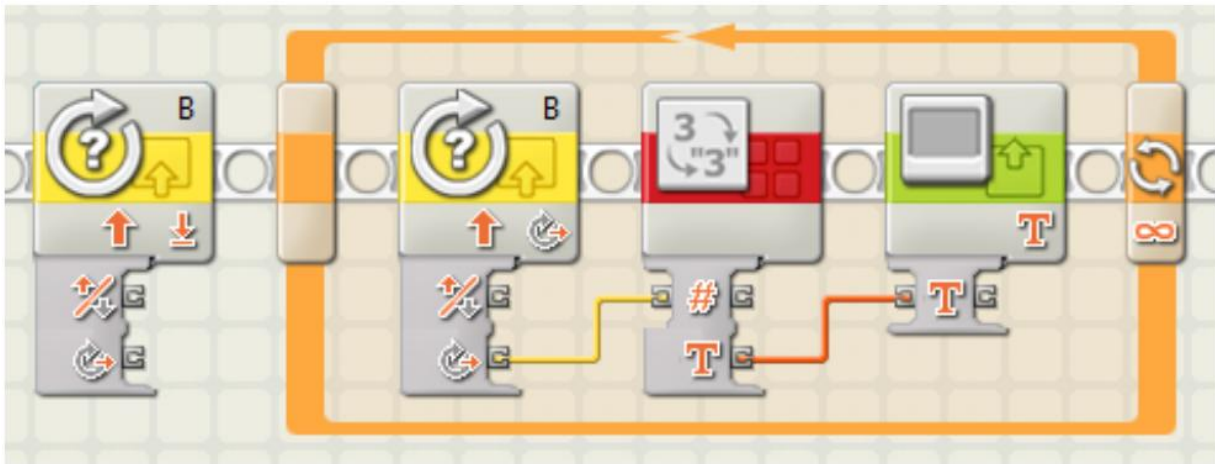


Рис. 3.4.108. Програма для демонстрації роботи давача обертів

Завдання 9.10. Внесіть зміни у програму (рис. 3.4.108) таким чином, щоб інформація виводилася б у кількості обертів.

Загальний алгоритм реалізації змагання з Кегельрингу.

Отже, загальний алгоритм кегельрингу виглядає наступним чином:

1. Робот розвертається на місці до моменту, поки давач ультразвуку не виявить кеглю.
2. Необхідно переміщуватися вперед до чорної лінії.
3. Повернутися назад на таку саму кількість обертів, на яку він проїхав вперед, виштовхуючи чергову кеглю. (нагадаємо, що на концентратор даних блоку Move можна передати тільки градуси, але не оберти).
4. Перейти до пункту 1.

Завдання 9.11. Складіть програму, яка реалізує описаний алгоритм кегельрингу в цілому і влаштуйте змагання за наведеними правилами.

Завдання 9.12. Складіть програму, за допомогою якої можна імітувати роботу класичного дискового телефону. Користувач набирає номер з трьох цифр (від 0 до 9). Після набору кожної цифри диск повинен повернутися у вихідне положення. По завершенню набору робот вимовляє в голос набраний номер (кожну цифру).

3.7 Питання для самоперевірки

1. Які основні функції виконує ROS у робототехнічних системах?
2. Яка архітектура ROS і які її ключові компоненти?
3. Яку роль відіграють вузли (nodes) та топіки (topics) у ROS?
4. Як здійснюється обмін повідомленнями між компонентами в ROS?
5. Що таке Microsoft Robotics Developer Studio і які основні компоненти входять до його складу?
6. Як створювати проекти в Microsoft Robotics Developer Studio?
7. Які типи роботів підтримує Microsoft Robotics Developer Studio?
8. Як налаштувати віртуальне середовище для тестування роботів у Microsoft Robotics Developer Studio?
9. Опишіть принцип роботи Simulation Environment у Microsoft Robotics Developer Studio.
10. Як працювати з Services в Microsoft Robotics Developer Studio? Яка їх роль в робототехнічних системах?
11. Як створюються та підключаються порти (ports) у Microsoft Robotics Developer Studio для взаємодії між різними компонентами системи?
12. Як використовувати Visual Programming Language (VPL) для розробки програм у Microsoft Robotics Developer Studio?
13. Як реалізувати паралельні процеси в Microsoft Robotics Developer Studio за допомогою ConcurrentTask?
14. Що таке Contract в Microsoft Robotics Developer Studio та як його використовувати?
15. Як налаштувати сенсори і актуатори в Microsoft Robotics Developer Studio для реальних роботів?
16. Як використати Service Manifests для визначення поведінки компонентів робототехнічної системи в Microsoft Robotics Developer Studio?
17. Як працювати з Microsoft Robotics Visual Simulation і для яких цілей він використовується?
18. Які основні елементи програмування в NXT-G?
19. Яка основна концепція програмування в NXT-G?
20. Як виглядає структура програми в середовищі NXT-G?
21. Які типи блоків використовуються для програмування в NXT-G?
22. Як реалізуються цикли та умовні оператори в NXT-G?
23. Як у NXT-G реалізується управління моторами?
24. Які типи датчиків підтримуються у NXT-G і як вони використовуються?
25. Як запрограмувати робота на рух у відповідь на сигнали з датчиків?
26. Які типи датчиків підтримує середовище NXT-G?
27. Як працюють умовні оператори (if-else) у NXT-G?
28. Як використовуються цикли (while, for) у NXT-G?
29. Як створювати та використовувати змінні в NXT-G?
30. Як передавати дані між блоками у програмі?

31. Як реалізувати збереження та використання даних під час виконання програми?
32. Як реалізувати паралельне виконання завдань у NXT-G?
33. Які є обмеження NXT-G у порівнянні з іншими середовищами для LEGO Mindstorms?
34. Як можна розширити функціональність NXT-G за допомогою стороннього програмного забезпечення?
35. Як налаштувати бездротовий зв'язок між двома LEGO NXT за допомогою Bluetooth?
36. Які можливості розширення надає NXT-G для складних робототехнічних завдань?
37. Як можна імпортувати або створювати власні блоки для NXT-G?
38. Яка основна концепція програмування в LabVIEW?
39. Які основні переваги графічного програмування в LabVIEW?
40. Що таке віртуальні прилади (Virtual Instruments, VI) у LabVIEW?

3.8 Література до 3 розділу

- 3.1. Bill M., Muller C., Kraus W., Executive Summary World Robotics 2019 Industrial Robots, International Federation of Robotics, 2019.
- 3.2. Spong M. W., Hutchinson S., Vidyasagar M., Robot modeling and control. John Wiley & Sons, 2020.
- 3.3. Siciliano B., Khatib O., Springer handbook of robotics. Springer Science & Business Media, 2008.
- 3.4. Початковий курс із програмування NXT. Частина I. Навчальний посібник науково-технічної конференції LEGO «Інженерна культура: від школи до виробництва». – Науково-Методичний Центр Університету TUFTS, 2012. – 40 с.
- 3.5. Початковий курс із програмування NXT. Частина II. Навчальний посібник науково-технічної конференції LEGO «Інженерна культура: від школи до виробництва». – Науково-Методичний Центр Університету TUFTS, 2012. – 34 с.
- 3.6. Промислові роботи MELFA Mitsubishi Electric Europe B.V. [Електронний ресурс]. Режим доступу до ресурсу: <https://emea.mitsubishielectric.com/fa/plp/rbt/robot>
- 3.7. NXT motor internals // Philo's Home Page. [Електронний ресурс]. Режим доступу до ресурсу: <http://www.philohome.com/nxtmotor/nxtmotor.htm>
- 3.8. ROS (Robot Operating System). Операційна система для роботів [Електронний ресурс]. Режим доступу до ресурсу: <https://www.ros.org/>
- 3.9. Siciliano, B., & Khatib, O. (Eds.). (2016). Springer Handbook of Robotics (2nd ed.). Springer. DOI: 10.1007/978-3-319-32552-1

- 3.10 Bräunl, T. (2003). *Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems*. Springer-Verlag. <https://doi.org/10.1007/978-3-540-00580-6>
- 3.11 Corke, P. (2011). *Robotics, Vision and Control: Fundamental Algorithms In MATLAB*. Springer. <https://doi.org/10.1007/978-1-4471-2508-4>
- 3.12 Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Ratti, C., & Ng, A. Y. (2009). ROS: an open-source Robot Operating System. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1-4). <https://doi.org/10.1109/ROBOT.2009.5152481>
- 3.13 Khusainov, R., & Frolov, A. (2018). *Robotics Programming: Build Robotic Projects with Arduino, Raspberry Pi, and BeagleBone Black*. Apress. <https://doi.org/10.1007/978-1-4842-3591-8>