

Аналіз мовної складової в програмних продуктах колаборативної робототехніки для вирішення технологічних завдань

(Представлено: д.т.н., проф. Кирилович В.А.)

На фоні всезростаючого попиту на колаборативну робототехніку змістом статті є висвітлення загальних проблем, які виникають під час програмування колаборативних промислових роботів. Виконано стислий аналітичний огляд мов програмування, які можуть забезпечити розв'язання ряду задач колаборативної робототехніки в цілому і в тому числі в механоскладальних виробництвах машино- та приладобудування. Проведено загальний аналіз мов програмування C++, Java та Python щодо їх переваг, недоліків та особливостей застосування в процесі програмування колаборативних роботизованих систем. Узагальнено представлено вказану вище інформацію у вигляді таблиці, де стисло вказано аналізовані особливості зазначених мов програмування та розділено їх на підгрупи для кращого сприйняття інформації. Результати проведеного аналізу мов програмування C++, Java та Python на основі доступних інформаційних джерел показали, що на сьогодні не існує універсальної мови програмування, яка б могла забезпечити виконання найпоширеніших вимог у колаборативній робототехніці у виробничій сфері. Проте розвиток колаборативних технологій та вказаних вище мов програмування невпинно продовжується завдяки інженерним та дослідницьким спільнотам, open source проєктам та стартап-проєктам, які активно засвоюють та розвивають нові інформаційні технології і впроваджують їх в колаборативні технології.

Ключові слова: робототехніка; колаборативність; промисловий робот; програмування; мехатроніка; кінематика; автоматизація; гнучке виробництво.

Актуальність теми. На сучасному етапі розвитку промислового виробництва промислові роботи (ПР) стають все більш затребуваними на різногалузевих виробництвах завдяки своїй високій функціональності, продуктивності, надійності тощо. Цей тип технологічного обладнання позитивно зарекомендував себе в автоматизованих технологічних процесах у різних галузях, в тому числі в приладо- та машинобудуванні. Хоча застосування ПР та автоматизація забезпечує високу ефективність, продуктивність та безпеку виробництва, існують технологічні процеси, де замінити людину ПР неможливо. Саме тому зараз набуває популярності особливий вид ПР, який розроблений для спільної роботи з людиною, що дає можливість зайняти вільну нішу в різноманітних технологіях, де класичні ПР не задовольняють вимоги, а людина малоефективна (непродуктивна). Такий вид ПР називається колаборативний промисловий робот (КПР). Тому парадигма сучасної промислової робототехніки у частині її колаборативності передбачає все більш зростаючу роль людини при реалізації відповідних технологічних процесів [1, 2]. Популярність КПР щорічно зростає, а статистичні дані Міжнародної федерації робототехніки (International Federation of Robotics – IFR) (рис. 1) вказують на зростання їх попиту на світовому ринку близько 50 % у період з 2020 до 2021 року [3].

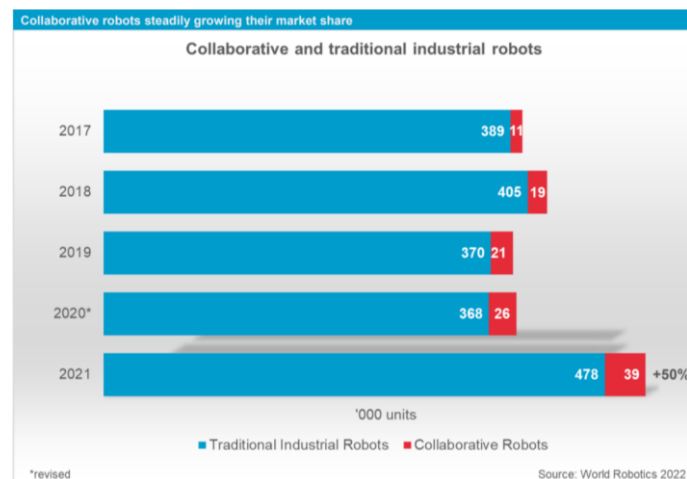


Рис. 1. Діаграма зростання кількості продажів ПР та КПР на світовому ринку [3]

Колаборативні роботизовані технології перш за все знаходять широке застосування у невиробничій сфері, в так званій сфері обслуговуючої робототехніки (*service robotics*) [4], яка позитивно проявила себе в сфері медицини, маркетингу (реклами), в соціальних сферах тощо. Проте саме в сучасних галузях приладо- та машинобудування колаборативна робототехніка найповніше розкриває серйозні перспективи широкого впровадження та подальшого розвитку. Однією з перспективних технологій з використанням КПП у співпраці з людиною є складальні або механоскладальні технології, які поєднують високу ефективність, гнучкість, надійність тощо. У колаборативній робототехніці, яка орієнтована на співпрацю між людиною та роботом, окрім спеціально розробленої апаратної складової, вирішальним фактором для досягнення ефективного функціонування системи є використання відповідних мов програмування та алгоритмів. Однак при виборі мови програмування для колаборативної робототехніки з'являється ряд проблем, що передбачає вирішення ряду техніко-технологічних завдань.

Метою статті є стислі узагальнення змісту доступних інформаційних джерел щодо використання найбільш поширених мов програмування, що використовуються в наявних проблемно орієнтованих програмних продуктах з метою їх потенційного використання під час розв'язування задач промислової колаборативної робототехніки.

Викладення основного матеріалу. Одною з головних проблем у колаборативній та класичній робототехніці є несумісність мов програмування між різними моделями ПР та КПП. Робототехніка на сьогоднішній день ще не має єдиної стандартизованої мови програмування, яка була б загальноприйнятою для всіх типів роботів. У найближчі роки навряд щось зміниться в цьому напрямі через бізнес, політику та ряд юридичних аспектів компаній-виробників. Майже кожна компанія-виробник ПР та КПП використовує власну мову програмування або власне програмне середовище, які функціонально та апаратно адаптовані під конкретні моделі чи модельні ряди роботів. Також є зворотний ефект: конкретні моделі ПР та КПП відомих компаній-виробників працюють лише з офіційними програмними продуктами. Тобто, створена так звана *закрита технічна система*. Така ситуація ускладнює розробку програмного забезпечення, пов'язаного з колаборативною робототехнікою в різних її проявах. Зазвичай це призводить до використання декількох мов програмування або до використання специфічних інженерних рішень, наприклад, конвертація команд між різними мовами, використання додаткового комп'ютерного обладнання тощо.

Іншою особливістю є необхідність інтеграції колаборативних технологій з іншими технологіями та системами. Колаборативна робототехніка може вимагати взаємодії з іншими системами, такими як комп'ютерні візуальні системи, штучний інтелект або хмарні сервіси [5–7]. Це ставить вимоги до мов програмування, які здатні до інтеграції та обміну даними з різними технологічними рішеннями.

Окрім того, колаборативна робототехніка вимагає гнучкості та можливості легко адаптуватися до змінних умов та вимог. КПП повинні бути здатні до швидкої реконфігурації та зміни своїх функцій для виконання нових завдань або співпраці з новими користувачами. Тому мови програмування мають підтримувати гнучкість, модульність, легкість впровадження змін тощо. Ще одним важливим аспектом є доступність навчання та розробки програмного забезпечення для КПП. Мови програмування повинні мати зручні інструменти для розробників, які дозволяють швидко навчитися та впроваджувати програми для КПП. Це сприяє залученню більш широкого кола фахівців до розробки КПП.

Узагальнюючи вказане вище, можна стверджувати, що використання мов програмування в колаборативній робототехніці має ряд невирішених взаємопов'язаних проблем. Несумісність мов програмування між роботами, складність використання, інтеграція з іншими технологіями, гнучкість та здатність адаптації до змінних умов – це лише кілька з проблем, з якими доводиться стикатися на практиці в контексті колаборативної робототехніки.

Вказане вище дає можливість для програмування КПП виокремити три перспективні та популярні серед інженерів і дослідників колаборативної робототехніки мови програмування, а саме: C/C++, Python та Java. У будь-якому разі дані мови програмування використовуються в різних галузях колаборативної робототехніки, починаючи від розробки *open source* (англ. open-source software – програмне забезпечення з відкритим початковим кодом) різномістовних проєктів КПП для навчання та досліджень, закінчуючи локальними конкретними промисловими інженерними рішеннями. Всі ці мови мають високий поріг входження користувача, особливо в сфері колаборативної робототехніки, складний графічний інтерфейс користувача та обмеженість підтримки з боку компаній-виробників ПР та КПП. Проте вони надають абсолютну волю розробнику в створенні алгоритмів роботи КПП, інтеграції технічного зору, використанні штучного інтелекту, застосуванні гнучких методів керування двигунами КПП, синхронізації з хмарними технологіями тощо. Варто зазначити, що ці мови програмування, як і сама колаборативна робототехніка, на даному етапі розвитку застосовуються частіше для досліджень або відпрацьовування конкретних технологій у виробничій сфері, проте широко застосовується в *service robotics*, особливо в open source або стартап-проєктах [8].

Нижче стисло наведено основні переваги вказаних мов програмування C/C++, Python та Java.

Використання мови C++ у програмуванні ПР та КПП має кілька важливих ключових аспектів. Перш за все використанням C++ дозволяє точно контролювати рух ланок маніпуляційної системи КПП за допомогою робототехнічних бібліотек та фреймворків (Framework – інфраструктура програмних рішень, що полегшує розробку складних систем. Спрощено таку інфраструктуру можна вважати своєрідною комплексною бібліотекою), таких як ROS (Robot Operating System) [9]. Розробники можуть використовувати C++ для програмування кінематики, навігації та планування траєкторії компонентів КПП. Це дозволяє КПП ефективно взаємодіяти з людиною та виконувати різноманітні завдання, що відповідає вимогам сучасної колаборативної робототехніки [10].

Взаємодія з датчиками та периферійними пристроями: C++ надає доступ до апаратного забезпечення робота, такого як датчики, камери, мікрофони та інші периферійні пристрої. Розробники можуть використовувати C++ для збору, обробки та аналізу даних з цих пристроїв. Це дозволяє КПП сприймати навколишнє середовище, реагувати на зміни та взаємодіяти з людьми або іншими об'єктами.

Також C++ надає потужні можливості для обробки та аналізу даних, зокрема, для виявлення образів, розпізнавання голосу, машинного навчання та іншого. Розробники можуть використовувати бібліотеки, такі як OpenCV, TensorFlow або Eigen, для реалізації «розумових» функцій робота, що дозволяє йому взаємодіяти з оточенням та виконувати складні завдання.

C++ є мовою з високою продуктивністю, що дозволяє розробникам оптимізувати код для досягнення максимальної швидкодії. Ще однією особливістю є те, що C++ надає доступ до низькорівневих можливостей, таких як пряма маніпуляція пам'яттю та асемблерні вбудовані функції. Це особливо корисно при роботі з вимогливими на ресурси додатками, де контроль над використанням пам'яті та оптимізація роботи з апаратними ресурсами є критично важливими. Також ця мова є кросплатформенною, що дозволяє розробникам писати програми, які можуть працювати на різних платформах та операційних системах. Це важливо для розробки КПП, які можуть бути використані в різних середовищах, умовах та з різним технологічним обладнанням.

Одна з найважливіших можливостей такої мови програмування – інтеграція з існуючим кодом. C++ часто використовується для інтеграції з існуючими системами та кодом, особливо з бібліотеками, які написані на C++. Це дозволяє розробникам використовувати вже наявний функціонал і взаємодіяти з іншими добре відпрацьованими компонентами системи.

Узагальнюючи вказане вище, вирішено переваги, недоліки та особливості застосування C++ в колаборативній робототехніці. Однією з переваг C++ є відносна швидкодія, оскільки C++ є компільованою мовою, це дозволяє генерувати оптимізований машинний код, що працює швидко та ефективно. Вказане особливо важливо для КПП, які мають швидко реагувати на зміни у своєму технологічному оточенні та взаємодіяти з людьми. Ще однією перевагою є низькорівневий доступ до апаратного забезпечення. C++ надає потужні можливості для безпосередньої взаємодії з апаратурою робота, такими як датчики, актуатори та інші пристрої. Також C++ підтримує об'єктно орієнтоване програмування, що спрощує структурування та організацію коду. Розробники можуть створювати класи та об'єкти, що представляють різні аспекти робота, це сприяє розширенню та повторному використанню коду. Останнього з висвітлених, але не менш вагомим переваг, є те, що C++ має велику та активну спільноту розробників, яка створює багато фреймворків, бібліотек та інструментів для розробки робототехніки.

Окрім переваг, C++ має ряд особливостей застосування та недоліків. Перш за все C++ вважається мовою високого рівня, що має високий поріг входження користувача, який іноді досягає років. C++ є складною мовою для вивчення та використання. Для ефективної роботи з мовою C++ необхідно мати глибокі знання з алгоритмів, структур даних та роботи з пам'яттю, що може зайняти значний час для вивчення та розвитку навичок. Також C++ є мовою, яка надає велику свободу розробникам, що веде до того, що можливості допустити помилки в коді значно збільшуються, наприклад, помилки у використанні пам'яті, які можуть призвести до зупинки програми, які важко відстежити та виправити. У мові програмування C++ не існує стандартного API (Application Programming Interface) для робототехніки, що означає, що розробники мають створювати власні інтерфейси та драйвери для взаємодії з роботами. Це може затримати розробку та призвести до виникнення проблем з сумісністю та інтеграцією з іншими системами.

Незважаючи на ці недоліки, C++ все ще залишається популярним вибором для програмування КПП завдяки своїм потужним можливостям, широким інструментаріям і спільнотою розробників.

Використання мови Java у програмуванні КПП має ряд особливостей. Java надає потужні інструменти для розробки програмного забезпечення, яке керує КПП. Розробники можуть використовувати Java для створення алгоритмів роботів для оцінки простору навколишнього середовища, планування маршрутів, керування рухом ланок маніпуляційної системи, обробки даних з сенсорів та взаємодії з оточенням тощо [11–12]. Мережева комунікація в Java має розширені можливості, що є важливим аспектом для систем моніторингу роботи КПП. Розробники можуть використовувати Java для створення мережевих протоколів, передачі даних, здійснення зовнішнього зв'язку та обміну інформацією між роботами та іншими системами.

Обробка даних з датчиків та сенсорних модулів також є сильною стороною Java. Вона має багатий набір бібліотек для обробки даних з сенсорів, таких як відеокамери, мікрофони, лазерні сканери тощо. Розробники можуть використовувати Java для отримання та аналізу даних з різних типів сенсорів, що дозволяє роботам більш ефективно взаємодіяти з оточенням. Також Java пропонує розширені можливості для розподіленої обробки даних, що може бути корисним для застосування КІР. Розробники можуть використовувати Java для створення розподілених систем, в яких кожен робот виконує свої завдання, співпрацюючи з іншими роботами у реальному часі.

Одна з вагомих особливостей Java – наявність вбудованої підтримки щодо безпеки даних, що робить її відмінним вибором для програмування КІР. Розробники можуть використовувати різні механізми безпеки Java, такі як шифрування даних, аутентифікація користувачів та керування доступом, щоб забезпечити захист від несанкціонованого доступу та зловживання використанням КІР та іншого обладнання. Також Java має добре розвинуту платформу та інструменти для масштабування програмного забезпечення, це дозволяє розробникам легко масштабувати програми для керування кількома КІР і забезпечувати їх ефективну роботу.

Враховуючи особливості використання Java у програмуванні КІР, можливо виокремити переваги, недоліки та особливості застосування. Перш за все це платформна незалежність. Java використовує віртуальну машину Java (JVM), що дозволяє виконувати код на будь-якій платформі, яка підтримує JVM. Це робить мову програмування Java зручним для розробки програмного забезпечення, яке працює на різних типах обладнання, в тому числі і КІР. Також Java має широкий вибір бібліотек, фреймворків та інструментів, які спрощують розробку колаборативного програмного забезпечення. Це дозволяє розробникам використовувати готові компоненти і зосередитися на основних функціях роботів. Java має вбудовану підтримку розподіленої обробки і може працювати в режимі мультипотокості. Це дозволяє створювати системи, в яких кожен робот виконує свої завдання, співпрацюючи з іншими роботами у реальному часі. Перевагою також є наявність розширених засобів безпеки, таких як шифрування даних, аутентифікація та керування доступом. Java є однією з найпопулярніших мов програмування, що веде до наявності широкого співтовариства розробників, готових надати підтримку, поради та вирішення проблем.

Серед недоліків використання Java у програмуванні КІР можна виокремити.

- швидкодія: Java, хоч і володіє багатьма можливостями, може мати обмежену швидкість;
- витрата пам'яті: JVM використовує значну кількість оперативної пам'яті для виконання програми. Це може бути недоліком у випадках, коли обмежений об'єм доступної пам'яті на роботі;
- обмежені можливості на вбудованих пристроях: на крайній ресурсомістких пристроях з обмеженими обчислювальними та пам'ятевими ресурсами використання Java може бути неефективним через великий об'єм пам'яті та обробки, потрібний для виконання JVM;
- відсутність деяких низькорівневих можливостей: Java як мова високого рівня може мати обмежені можливості у роботі з деякими низькорівневими функціями або прямим керуванням апаратною, що може бути важливим у деяких сферах робототехніки.

Мова програмування Python є відносно новою та поширеною мовою програмування, яка використовується для розробки програмного забезпечення для КІР. Нижче наведено основні переваги та особливості використання Python у програмуванні КІР.

Перш за все Python має багато можливостей для розробки програмного забезпечення для керування КІР. Розробники можуть використовувати мову програмування Python для створення алгоритмів обробки даних з сенсорів та взаємодії з оточенням. Python має багатий набір бібліотек для обробки даних з різних сенсорів, таких як відеокамери, мікрофони, лазерні сканери тощо. Інженери можуть використовувати Python для аналізу образів, виявлення об'єктів, обробки звуку та інших завдань, що пов'язані з обробкою сенсорних даних.

Ще одна перевага Python – це мережева комунікація. Python надає простий та потужний інтерфейс для мережевої комунікації, що є важливим аспектом колаборативних технологій. Розробники можуть використовувати Python для створення мережевих протоколів, передачі даних, здійснення зовнішнього зв'язку та обміну інформацією між КІР та іншими пристроями і системами.

Також Python має велику кількість бібліотек та фреймворків, які спрощують розробку колаборативних технологій. Наприклад, бібліотеки NumPy, OpenCV, SciPy, які дозволяють виконувати обробку даних, машинне навчання та застосування технічного зору [9, 13, 14].

Мова Python відомий своєю простотою та легкістю використання, що робить його популярним вибором для розробки колаборативних технологій та програмування КІР. Синтаксис Python є лаконічним і зрозумілим, що полегшує розробку, тестування та налагодження коду. Крім того, наявність розширених функцій мови, таких як генератори, декоратори та управління контекстом, дозволяє розробникам писати більш компактний та елегантний код.

Вагомою перевагою є те, що мова програмування Python має активну та велику спільноту розробників, яка надає допомогу, документацію та відповіді на питання. Вказане вище дає можливість програмувати КІР в Python більш доступно, швидко та зручно. Розробники можуть швидко знайти рішення для питань

або проблем, що виникають, а також спілкуватися з іншими експертами для обміну досвідом та ідеями. В цілому Python є потужним та гнучким інструментом для програмування КІР, завдяки своїй легкості використання та багатству бібліотек. Головними перевагами використання Python у програмуванні КІР є простота використання, велика кількість підтримуваних бібліотек. Наявність бібліотек для обробки сигналів, комп'ютерного зору, машинного навчання та інших областей дозволяє розробникам швидко впроваджувати потрібні функціональні можливості. Також Python має велику кількість інструментів, таких як інтерактивні середовища розробки (наприклад, Jupyter Notebook [15]), що дозволяють розробникам експериментувати, тестувати та налагоджувати код у реальному часі онлайн з браузера. Це допомагає прискорити процес розробки та зменшити час, необхідний для впровадження нових функцій.

Незважаючи на переваги, використання Python у програмуванні КІР також має недоліки:

- швидкодія: Python є інтерпретованою мовою програмування, що може вплинути на швидкість виконання деяких комплексних завдань. У випадку потреби високої швидкодії, такої як обробка великих обсягів даних або виконання складних алгоритмів, Python може бути менш ефективним порівняно з іншими мовами, такими як C++ або Java;
- зайнятість пам'яті: Python може використовувати більше пам'яті порівняно з іншими мовами, що може бути проблемою у випадку, коли ресурси обмежені;
- залежність від зовнішніх бібліотек: використання Python у програмуванні КІР може вимагати використання зовнішніх бібліотек, що може призводити до проблем з управлінням, швидкодії, забезпечення сумісності між різними версіями бібліотек тощо.

Враховуючи переваги, недоліки та особливості застосування, вибір мови програмування для програмування КІР має бути обґрунтованим залежно від конкретних потреб проєкту. Наприклад, якщо швидкодія, обмежена пам'ять пристроїв або доступ до низькорівневих функцій є критичними, то варто розглянути такі мови програмування, як C++ або Python, які краще забезпечують ці вимоги.

На основі викладеної вище аналітичної інформації можна стверджувати, що на сьогодні не існує універсальної мови програмування для колаборативних роботизованих технологій, що обумовлено рядом причин. Перш за все це фінансова складова: компанії-виробники КІР та ПР не зацікавлені у виготовленні продукції, яка може бути експлуатована програмними продуктами інших компаній та конкурентами. Також компанії-виробники не можуть гарантувати повну функціональну працездатність у випадках, коли користувач використовує сторонні програмні продукти, тому що в таких випадках втрачається гарантія. З технічної точки зору використання універсальної мови програмування є проблемним через використання компаніями-виробниками КІР різного апаратного забезпечення: двигунів, драйверів, контролерів тощо. Opensource проєкти частково вирішують цю проблему, надаючи підтримку в програмному питанні та з точки зору юридичних аспектів, проте також не надають гарантію користувачу. Тому їх застосування на великих підприємствах є недоцільним та ризиковим, бо у випадку аварійної ситуації, причиною якої буде обладнання, підприємство не отримає відшкодування або гарантійного обслуговування.

Однак існує декілька стратегій, які можуть допомогти подолати вказані вище проблеми з технічного боку. Перш за все це стандартизація бібліотек та модулів у мовах програмування для КІР може сприяти зниженню несумісності між різними системами та моделями КІР. Друга стратегія – розробка спеціалізованих мов програмування, які були б зорієнтовані саме на потреби колаборативної робототехніки. Ці мови можуть враховувати особливості співпраці між людиною та роботом, містити зручний інтерфейс для користувача та механізми безпеки. Розробка таких мов дозволить розробникам зосередитися на специфічних завданнях колаборативної робототехніки, забезпечуючи ефективність та безпеку системи.

Таблиця 1

Систематизація аналізованих особливостей мов програмування C++, Java та Python щодо можливостей застосування в колаборативних технологіях

| Мова | Вагомні аспекти в механоскладальних роботизованих виробництвах, автоматизованих виробництвах тощо | | | | | Відмітні особливості в програмній реалізації | | | | |
|--------|---|--|--|--|-----------------------|--|----------------------|--------------------------------|--|---------------|
| | взаємодія з датчиками та периферійними пристроями | планування траєкторії руху затиських пристроїв КІР | підтримка роботи з розподіленими системами | спеціалізовані бібліотеки для роботи з ПР та КІР | простота використання | швидкодія роботи програмного | мережева комунікація | підтримка різних платформ (ОС) | наявність універсальних фреймворків для роботи з КІР | безпека даних |
| C++ | ● | ● | ● | ● | ● | ● | ● | ◎ | ● | ● |
| Java | ◎ | ● | ● | ● | ◎ | ● | ● | ● | ● | ● |
| Python | ● | ● | ● | ● | ● | ◎ | ● | ● | ● | ● |

Вказану вище аналізовану інформацію про переваги, недоліки та особливості застосування мов програмування C++, Java та Python у колаборативній робототехніці стисло та узагальнено висвітлено в таблиці 1. Аналізовані можливості або особливості мов програмування умовно розділено на дві підгрупи, які описують особливу цінність будь-яких мов програмування в практичній реалізації у колаборативній робототехніці. В таблиці прийнято такі умовні позначення: ● означає повну підтримку або наявність аналізованої особливості чи твердження; ● означає часткову підтримку або відповідність аналізованої особливості чи твердження; © означає повну непідтримку або невідповідність аналізованої особливості чи твердження. Варто зазначити, що мови програмування розвиваються та оновлюються з часом, тому ця таблиця визначає загальну ситуацію в колаборативній робототехніці на сьогодні.

Особливо варто наголосити на тому, що ряд відомих виробників КПП (КУКА, АВВ та інші [16, 17]) забезпечує експлуатацію своєї продукції за рахунок програмних продуктів, перепрограмувати які неможливо через юридичні та технічні обмежень, але які є обов'язковою умовою експлуатації відповідних КПП.

Висновки та перспективи подальших досліджень. Виконано узагальнення аналізу доступних інформаційних джерел щодо використання мов програмування C++, Java, та Python з метою їх потенційного використання під час розв'язування задач промислової колаборативної робототехніки. Визначено їх переваги, недоліки та особливості застосування щодо програмування КПП. Останнє систематизовано та стисло викладено в табличній формі. Її зміст може розглядатись як основа вибору «потрібної» мови програмування для використання в колаборативній робототехніці, навіть якщо обрана мова не є комфортною для користувача та розробника.

Результати аналізу дають можливість розглядати мови програмування як один з критеріїв вибору програмних продуктів та мов програмування за умови їх схожої функціональності та за умови можливості використання. Вказане є показником професійної комфортності для дослідників та користувачів під час розробки та використання певних програмних додатків для існуючих програмних продуктів.

Список використаної літератури:

1. Кирилович В.А. Проблеми розподілу завдань між людиною та роботом у колаборативних механоскладальних технологіях / В.А. Кирилович, А.Р. Кравчук // Тези Всеукраїнської науково-практичної онлайн-конференції здобувачів вищої освіти і молодих учених, присвяченої Дню науки, 15–26 травня. – Житомир : Житомирська політехніка, 2023.
2. Кирилович В.А. Проблеми промислової колаборативної робототехніки в гнучких механоскладальних виробництвах / В.А. Кирилович, А.Р. Кравчук // Тези Всеукраїнської науково-практичної онлайн-конференції здобувачів вищої освіти і молодих учених, присвяченої Дню науки, 16–20, 26 травня. – Житомир : Житомирська політехніка, 2022. – С. 72 [Електронний ресурс]. – Режим доступу : <https://conf.ztu.edu.ua/wp-content/uploads/2022/06/5-2.pdf>.
3. The International Federation of Robotics // Order World Robotics Reports, 2022 [Electronic resource]. – Access mode : <https://ifr.org/free-downloads/>.
4. The International Federation of Robotics // Service Robot Definition, 2022 [Electronic resource]. – Access mode : <https://ifr.org/service-robots>.
5. Daniel P. Brogan Deep learning computer vision for robotic disassembly and servicing applications / P. Daniel Brogan, Nicholas M. DiFilippo, Musa K. Jouaneh // Array. – 2021. – Vol. 12, December, 100094 [Electronic resource]. – Access mode : <https://www.sciencedirect.com/science/article/pii/S2590005621000394>.
6. Review and application of Edge AI solutions for mobile collaborative robotic platforms / Aswin K. Ramasubramanian, Robins Mathew, Inder Preet, Nikolaos Papakostas // Procedia CIRP. – 2022. – Vol. 107. – P. 1083–1088 [Electronic resource]. – Access mode : <http://surl.li/homze>.
7. Framework and Enabling Technologies of Cloud Robotic Disassembly / Liping Yuan, Jia Cui, Xinyu Zhang, Jiayi Liu // Procedia Computer Science. – 2020. – Vol. 176. – P. 3673–3681 [Electronic resource]. – Access mode : <https://www.sciencedirect.com/science/article/pii/S1877050920319116>.
8. Documentation // Google Open Source, 2022 [Electronic resource]. – Access mode : <https://opensource.google/documentation/reference>.
9. Open Source Computer Vision Library : Official site [Electronic resource]. – Access mode : <https://opencv.org/>.
10. Lentin J. Fundamentals of C++ for Robotics Programming / J. Lentin. – 2018 [Electronic resource]. – Access mode : https://www.researchgate.net/publication/325339177_Fundamentals_of_C_for_Robotics_Programming.
11. The Destination for Java Developer : official site of Java [Electronic resource]. – Access mode : <https://dev.java/>.
12. A Flexible Java Class Library for Simulating and Teleoperating Robots / F.A. Candelas, F. Torres, S. Puente and other // IFAC Proceedings Volumes. – 2004. – Vol. 37, Issue 4, April. – P. 539–544 [Electronic resource]. – Access mode : <https://www.sciencedirect.com/science/article/pii/S1474667017361700>.
13. Open Source NumPy Library : official site [Electronic resource]. – Access mode : <https://numpy.org/doc/stable/>.
14. Open Source SciPy Library : official site [Electronic resource]. – Access mode : <https://scipy.org/>.
15. Jupiter Project Documentation : official site [Electronic resource]. – Access mode : <https://docs.jupyter.org/en/latest/>.
16. Kuka Download Center : official site [Electronic resource]. – Access mode : <http://surl.li/homzr>.
17. ABB Robotics : official site [Electronic resource]. – Access mode : <https://new.abb.com/products/robotics>.

References:

1. Kyrylovych, V.A. and Kravchuk, A.R. (2023), «Problemy rozpodilu zavdan mizh liudynoiu ta robotom u kolaboratyvnykh mekhanoskladalnykh tekhnolohiiakh», *Tezy Vseukrainskoi naukovo-praktychnoi onlain-konferentsii zdobuvachiv vyshchoi osvity i molodykh uchenykh, prysviachenoi Dniu nauky*, 15–26 travnia, Zhytomyrska politekhnika, Zhytomyr.
2. Kyrylovych, V.A. and Kravchuk, A.R. (2022), «Problemy promyslovoi kolaboratyvnoi robototekhniki v hnuchkykh mekhanoskladalnykh vyrobnytstvakh», *Tezy Vseukrainskoi naukovo-praktychnoi onlain-konferentsii zdobuvachiv vyshchoi osvity i molodykh uchenykh, prysviachenoi Dniu nauky*, 16–20, 26 travnia, Zhytomyrska politekhnika, Zhytomyr, pp. 72, [Online], available at: <https://conf.ztu.edu.ua/wp-content/uploads/2022/06/5-2.pdf>
3. «The International Federation of Robotics» (2022), *Order World Robotics Reports*, [Online], available at: <https://ifr.org/free-downloads/>
4. «The International Federation of Robotics» (2022), *Service Robot Definition*, [Online], available at: <https://ifr.org/service-robots>
5. Daniel P. Brogan, Nicholas M. DiFilippo and Musa K. Jouaneh (2021), «Deep learning computer vision for robotic disassembly and servicing applications», *Array*, Vol. 12, December, 100094 [Online], available at: <https://www.sciencedirect.com/science/article/pii/S2590005621000394>
6. Aswin K. Ramasubramanian, Robins Mathew, Inder Preet and Nikolaos Papakostas (2022), «Review and application of Edge AI solutions for mobile collaborative robotic platforms», *Procedia CIRP*, Vol. 107, pp. 1083–1088, [Online], available at: <http://surl.li/homze>
7. Liping Yuan, Jia Cui, Xinyu Zhang and Jiayi Liu (2020), «Framework and Enabling Technologies of Cloud Robotic Disassembly», *Procedia Computer Science*, Vol. 176, pp. 3673–3681, [Online], available at: <https://www.sciencedirect.com/science/article/pii/S1877050920319116>
8. «Documentation» (2022), *Google Open Source*, [Online], available at: <https://opensource.google/documentation/reference>
9. *Open Source Computer Vision Library*, Official site, [Online], available at: <https://opencv.org/>
10. Lentin, J. (2018), *Fundamentals of C++ for Robotics Programming*, [Online], available at: https://www.researchgate.net/publication/325339177_Fundamentals_of_C_for_Robotics_Programming
11. The Destination for Java Developer, official site, [Online], available at: <https://dev.java/>
12. Candelas, F.A., Torres, F., Puente, S. et al. (2004), «A Flexible Java Class Library for Simulating and Teleoperating Robots», *IFAC Proceedings Volumes*, Vol. 37, Issue 4, April, pp. 539–544, [Online], available at: <https://www.sciencedirect.com/science/article/pii/S1474667017361700>
13. *Open Source NumPy Library*, official site, [Online], available at: <https://numpy.org/doc/stable/>
14. *Open Source SciPy Library*, official site, [Online], available at: <https://scipy.org/>
15. *Jupiter Project Documentation*, official site, [Online], available at: <https://docs.jupyter.org/en/latest/>
16. *Kuka Download Center*, official site, [Online], available at: <http://surl.li/homzr>
17. *ABB Robotics*, official site, [Online], available at: <https://new.abb.com/products/robotics>

Кравчук Антон Романович – аспірант, асистент кафедри робототехніки, електроенергетики та автоматизації ім. проф. Б.Б. Самотока Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0002-8305-2492>.

Наукові інтереси:

- промислова та мобільна робототехніка;
- САПР;
- вбудовані системи.

Kravchuk A.R.

Analysis of the language component in software products of collaborative robotics for solving technological tasks

Against the background of the ever-growing demand for collaborative robotics, the content of the article is to highlight the general problems that arise when programming collaborative industrial robots. A brief analytical review of programming languages that can provide solutions to a number of problems of collaborative robotics in general and including in the mechanic assembly productions of machine and instrument engineering is performed. A general analysis of C++, Java, and Python programming languages was conducted regarding their advantages, disadvantages, and application features in the process of programming collaborative robotic systems. The above information is summarized in the form of a table, which summarizes the analyzed features of the above programming languages and divides them into subgroups for a better understanding of the information. The results of the analysis of C++, Java and Python programming languages based on available information sources showed that at the moment there is no universal programming language that could ensure the fulfillment of the most common requirements in collaborative robotics in the production sphere. However, the development of collaborative technologies and the above-mentioned programming languages continues unceasingly thanks to engineering and research communities, open source projects and startup projects that actively learn and develop new information technologies and implement them into collaborative technologies.

Keywords: robotics; industrial robot; collaboration; programming; mechatronics; kinematics; automation; flexible production.

Стаття надійшла до редакції 29.03.2023.