

Державний університет «Житомирська політехніка»
Кафедра А та КІТ ім. проф. Б. Б. Самотокіна

Використання robot operating system (ROS) для
створення напівнатурного моделюючого комплексу
маніпуляційних роботів

Кирилович В.А.

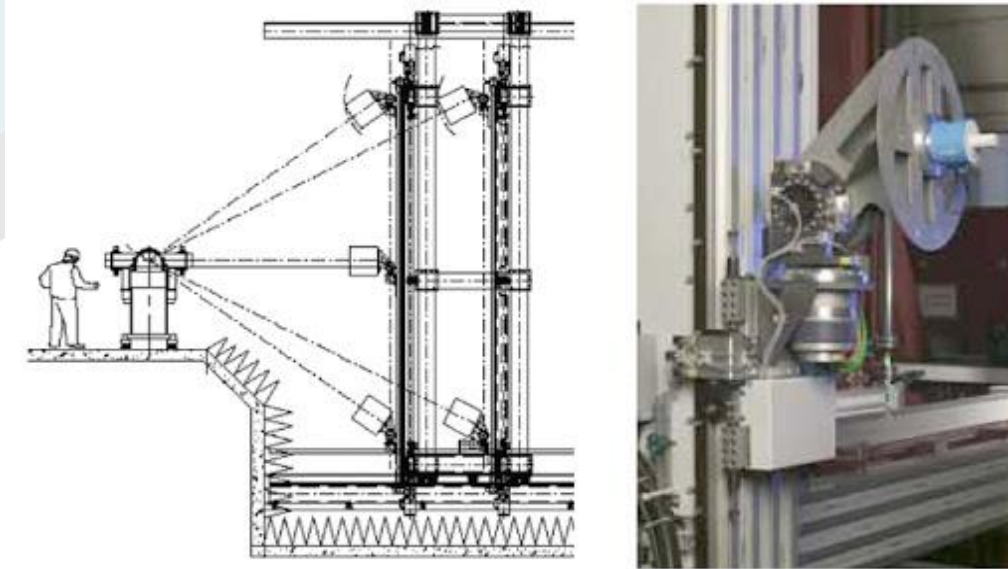
Мета та задачі проекту

Метою даного проекту є : ознайомлення з структурою та складовими напівнатурного моделюючого комплексу маніпуляційних роботів з використання robot operating system (ROS)

Для досягнення мети, було поставлено такі завдання:

1. Дослідити robot operating system (ROS) ;
2. Розглянути основні складові напівнатурного моделюючого комплексу маніпуляційних роботів;
3. Зробити висновки стосовно технічної складової маніпуляційних роботів та їх використання в теперішньому часі.

Актуальність теми

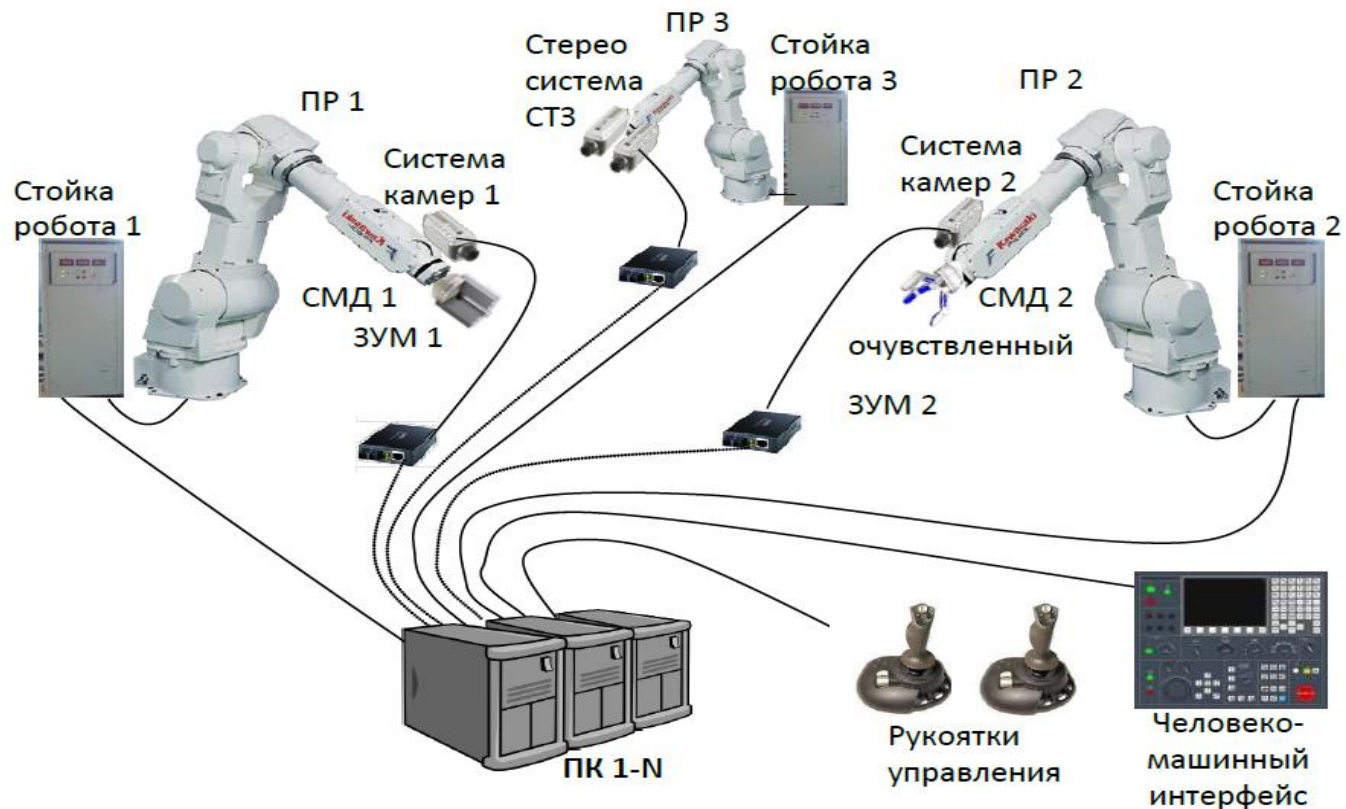


Напівнатурні моделюючі комплекси (НМК) є ефективним засобом, що дозволяють проводити дослідження нових робототехнічних систем (РТС) без необхідності виготовлення дослідних зразків, а також дослідження РТС, функціонування яких в наземних умовах неможливо (космічні маніпуляційні роботи).

До складу НМК може входити різні моделюючі і управляючі програмного забезпечення, датчики зовнішнього середовища (камери, СМД, лазерні далекоміри і інше), а також виконавчі і захватні пристрої. Таким чином, НМК сам по собі є складною розподіленої РТС.



Склад НМК маніпуляційних роботів



Комплекс включає в себе промислові роботи (ПР), СМД, захватні пристрої маніпулятора (ЗУМ), системи технічного зору (СТЗ) і багато іншого.

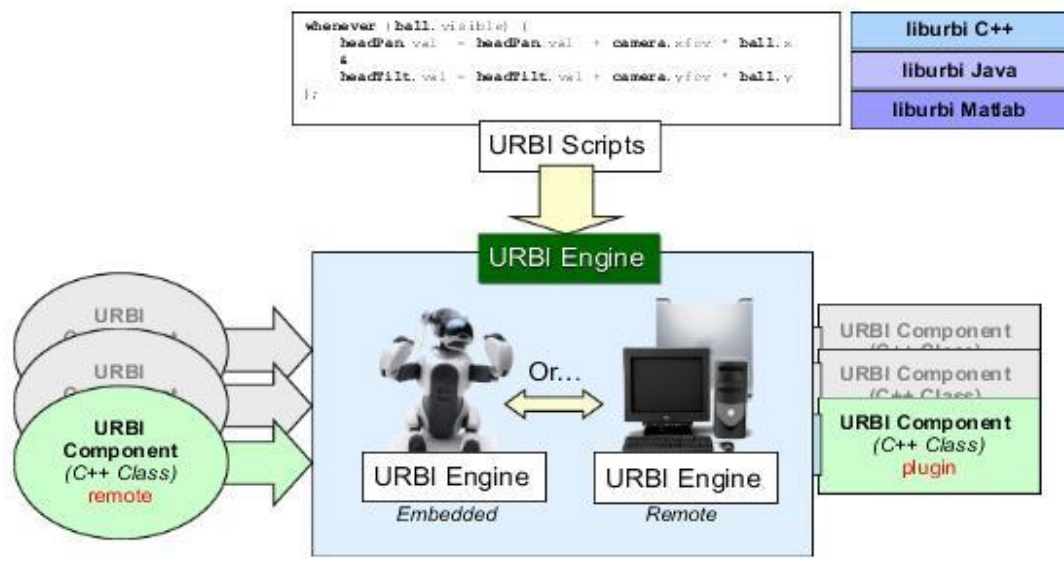
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РОБОТОТЕХНІКИ

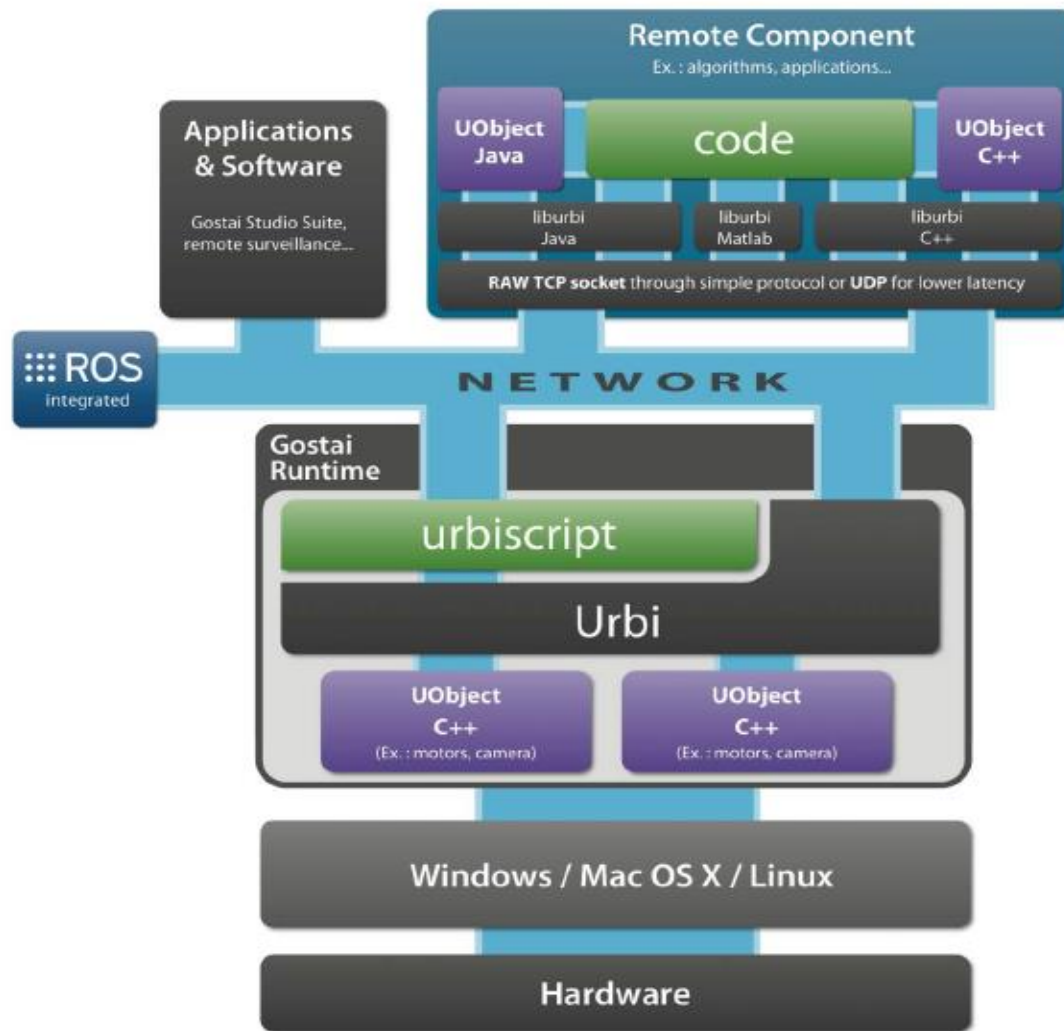
В даний час не існує якогось загальноприйнятого стандарту, якої-небудь однієї платформи для створення програмного забезпечення для роботів. Однак існує кілька рішень, які широко використовуються в академічних колах і "робототехнічних співтовариствах" і можуть лягти в основу такого стандарту. Далі розглянуті деякі універсальні крос-платформні рішення з відкритим вихідним кодом.

URBI

Urbi є повнофункціональною крос-платформеною (Windows, Linux, Mac OS) середою з відкритим вихідним кодом (ліцензія GNU AGPL v3) для організації взаємодії компонентів в складних системах.

Urbi забезпечує всі необхідні функції для координації роботи різних незалежних паралельних компонентів (приводи, датчики, програмне забезпечення пристроїв, які забезпечують такі функції, як перетворення тексту в мову, розпізнавання осіб і так далі).





Архітектура Urbi



Середовище Urbi дозволяє використовувати компоненти, написані на мові C++, для взаємодії з обладнанням там, де потрібна ефективність і доступ до низькорівневих функцій. Компоненти повинні використовувати API бібліотеки UObject C++ для інтеграції в Urbi (для Java і Matlab також існує свій API).

Вона також має urbiscript - мова сценаріїв для опису поведінки високого рівня, схожий на Python або LUA, але з вбудованою семантикою для опису паралельних і подієво-керованих програм.

Player

Player - крос-платформенне (Linux, Solaris, BSD, Mac OSX) програмне забезпечення для дослідження робототехнічних систем з відкритим вихідним кодом (ліцензія GNU GPL). Воно складається з трьох частин:

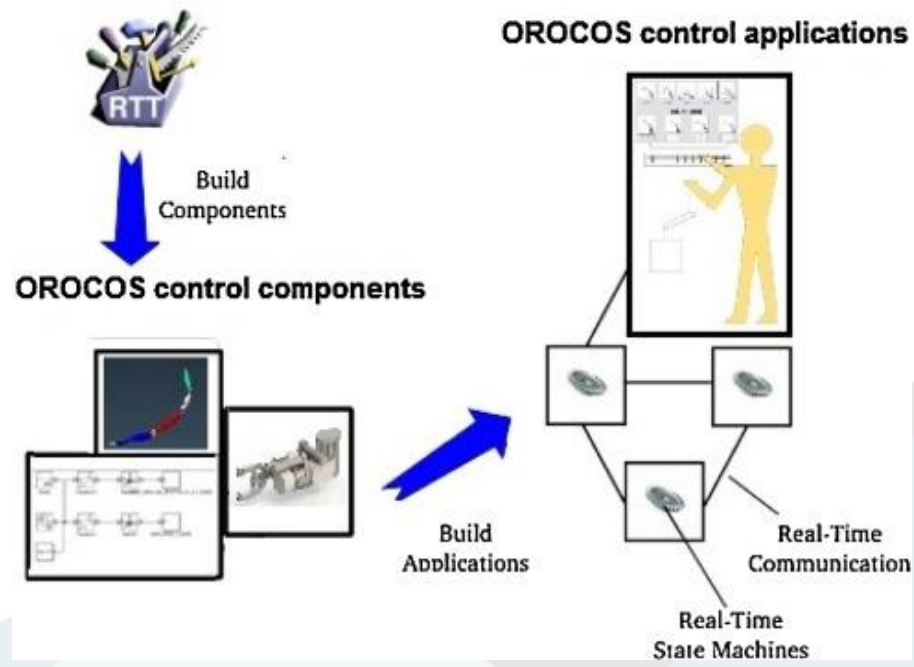
1. Player - сервер, який забезпечує мережевий інтерфейс для взаємодії між модулями-драйверами устаткування і керуючими програмами.
2. Stage - двовимірний симулятор мобільних роботів.
3. Gazebo - симулятор роботів / тренажер в тривимірному просторі.



Сервер *Player* взаємодіє з конкретними апаратними пристроями за допомогою драйверів, а своїм клієнтам надає стандартний абстрактний інтерфейс пристроїв. Це дозволяє керуючим програмами прозоро взаємодіяти з новим обладнанням. Нові драйвери можуть бути додані будь-яким розробником. Драйвери зберігаються у вигляді модулів, порядок їх завантаження та параметри описуються в конфігураційному файлі.

OROCOS

OROCO (Open Robot Control Software) - крос-платформенне (Linux / RTLinux, Windows, Mac OS) вільне програмне забезпечення (ліцензія GNU LGPL) для управління робототехнічними системами. OROCOS містить кілька незалежних частин. OROCOS RTT Class



У тому числі Orocos toolchain - основний інструментарій для створення розподілених робототехнічних додатків реального часу. З його допомогою можна створювати модулі на C ++, які взаємодіють з рештою системи яка надається через Real-Time Toolkit API.

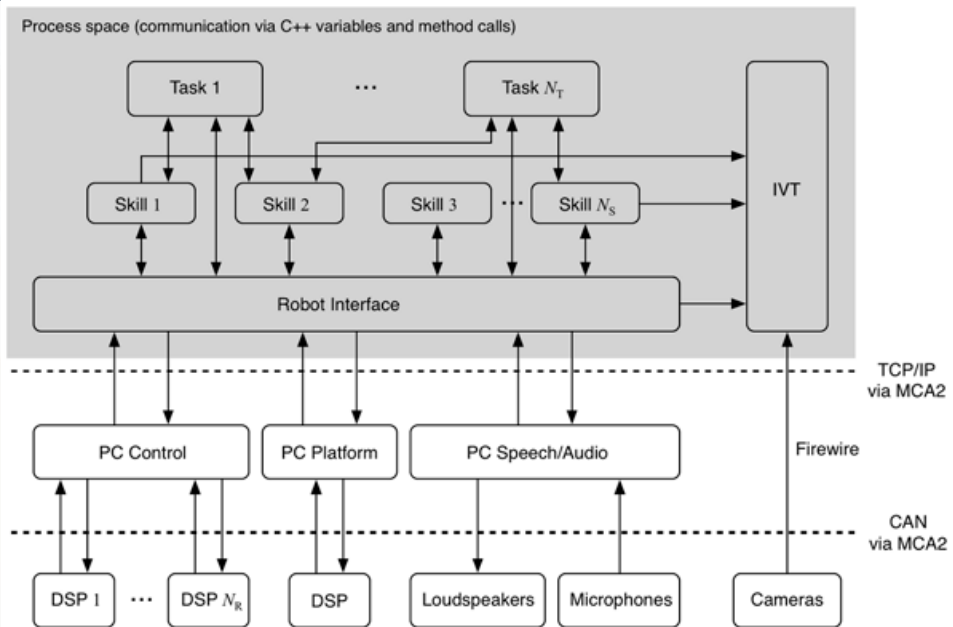
Відмітна особливість системи OROCOS в тому, що вона спеціально розроблена для створення додатків реального часу. Система OROCOS також інтегрована в ROS.

MCA2

MCA2 (Modular Controller Architecture 2.0) – крос-платформне (Linux / RTLinux, Win32, Mac OS / X) вільне (ліцензія GNU GPL) програмне забезпечення для управління роботами. MCA2 являє собою каркас для створення програмного забезпечення на основі модулів зі стандартними інтерфейсами. Він надає клієнтську бібліотеку для створення нових модулів і інфраструктуру, що забезпечує їх роботу, синхронізацію і взаємодію в процесі виконання, в тому числі в режимі реального часу.



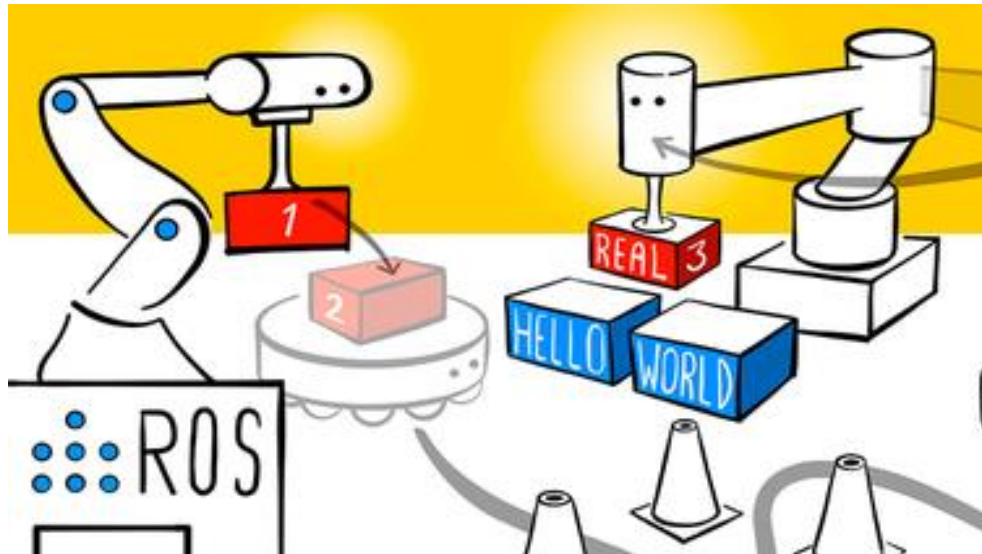
Спочатку архітектура МСА розроблялася спеціально для автономних роботів серії ARMAR. Розглянемо для прикладу архітектуру програмного забезпечення робота-гуманоїда ARMAR-III, яке побудоване з використанням МСА2.



На нижчому рівні розташовані сигнальні мікропроцесори виконавчої системи управління та інші апаратні засоби, такі як мікрофони, гучномовці, камери і так далі. Вони пов'язані з комп'ютерами середнього рівня або безпосередньо, або через CAN-шину. На середньому рівні здійснюється управління більш високого рівня: розрахунок прямої і інверсної кінематики, планування безпечних траєкторій, обробка мови і так далі. Склад модулів на перших двох рівнях залишається незмінним. Програмування робота здійснюється тільки на верхньому рівні. Тут програмний інтерфейс робота забезпечує зручний доступ до датчиків і виконавчих пристроїв за допомогою C ++ змінних і методів. На додаток до можливості прямого доступу до датчиків і приводів, визначені два рівні абстракції: завдання і навички. Навички реалізують можливості, такі як навігація, візуальний пошук об'єкта, захоплення / переміщення / установка об'єкта, відкриття / закриття дверей і так далі.

ROS

ROS (Robot Operating System) - мета-операційна система для роботів з відкритим вихідним кодом (ліцензія BSD). Вона не замінює операційну систему, вона розширює її набором модулів, необхідних для управління РТС. Спочатку ROS розроблена для UNIX-сумісних операційних систем, але ведуться роботи по її перекладу і на Windows. Основною метою ROS є підтримка повторного використання коду в робототехнічних дослідженнях і розробках.



ROS складається з двох частин:

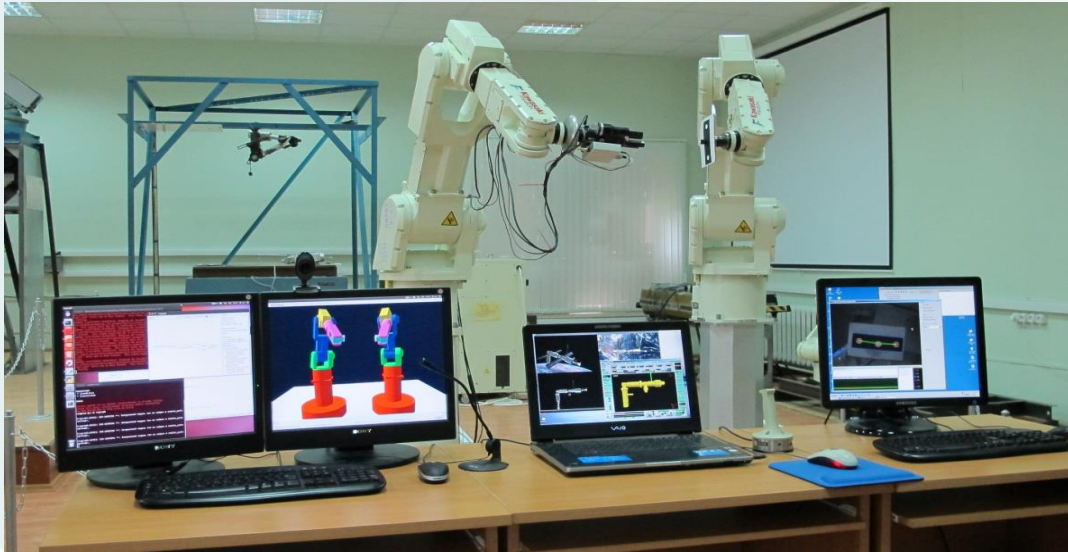
- ❖ ядро ROS - мінімально необхідний для роботи набір інструментів, модулів і бібліотек;
- ❖ набір розвитку користувачами пакетів, які реалізують різні функції робототехніки: робота з обладнанням, планування траєкторій, обробка інформації, моделювання та багато іншого.



Покращення НМК маніпуляційних роботів

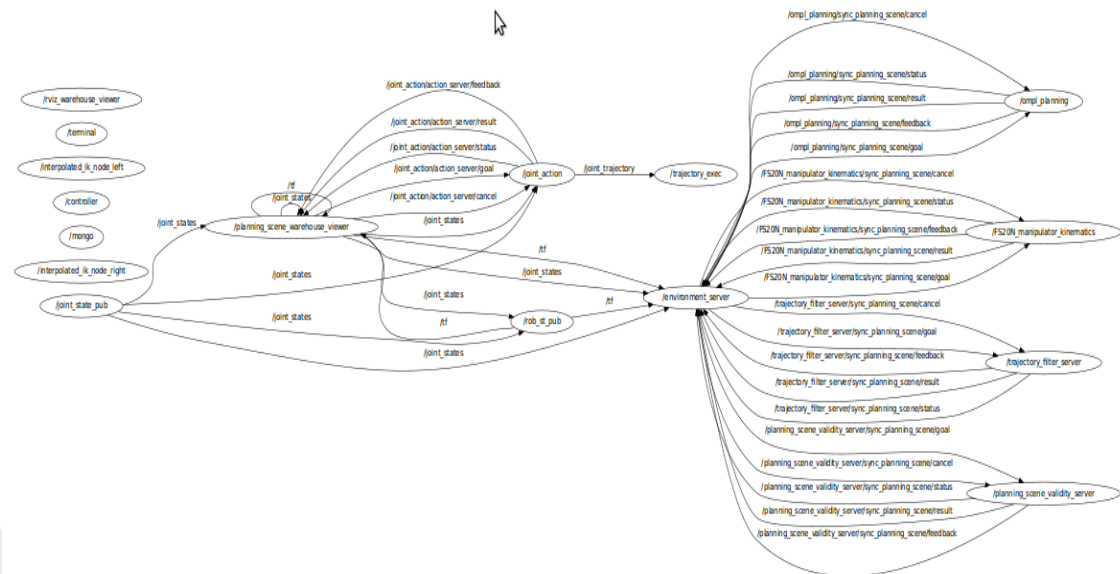
Після аналізу поточного стану робототехнічного програмного забезпечення варто вибрати ROS в якості основи для побудови НМК. Через такі причини: зручність розробки і використання, універсальність, модульність, гнучкість, можливість легкого масштабування, інтеграція з іншими системами робототехнічного програмного забезпечення, відкритий програмний код, велика кількість готових драйверів, алгоритмів, рішень для стандартних завдань робототехніки, можливість незалежно розробляти і додавати в систему власні модулі, легкість налагодження, велике активне співтовариство користувачів.





Зовнішній вигляд комплексу під управлінням ROS

Обчислювальна мережа ROS однієї з конфігурацій НМК



Висновок

- Для прикладу наведена обчислювальна мережа ROS при управлінні одним з роботів, яка включає вузли, які здійснюють взаємодію з користувачем, планування і фільтрацію траєкторій з урахуванням об'єктів робочого середовища, систему безпеки, обмін командами та інформацією з промисловим контролером маніпулятора. Перехід на дослідження дворукого робота може бути здійснений буквально однією командою, без необхідності тривалого налаштування і тим більше перекомпіляції.

Використана література

- <http://robocraft.ru/blog/robosoft/721.html>
- <https://www.generationrobots.com/blog/en/ros-robot-operating-system-2/>
- <http://docs.voltbro.ru/starting-ros/ros-about.html>
- <http://engsi.ru/file/out/505941>

ДЯКУЮ ЗА УВАГУ !