STRATEGIE DI INTEGRAZIONE E STANDARDIZZAZIONE DELLA ROBOTICA NELL'INDUSTRIA 4.0

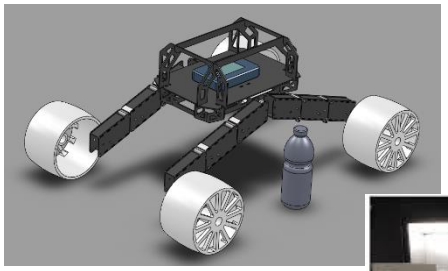
Paolo Gallina

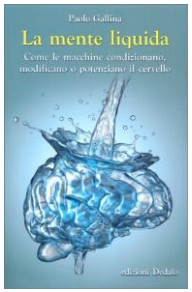
31 maggio 2019 - ore 15.00

Il Fondo Sociale Europeo in Friuli Venezia Giulia
Programma Operativo Regionale 2014-2020
Unione europea Fondo sociale europeo
REGIONE AUTONOMA FRIULI VENEZIA GIULIA
UN INVESTIMENTO PER IL TUO FUTURO
in rete fvg
seminari per l'innovazione in rete

Paolo Gallina

- Incardinato all'Università di Trieste
- Professore di Robotica e Interazione uomo-macchina
- Ricerca nel settore della:
  - Robotica mobile (rover);
  - Sistemi human-robot;
  - Robotica e neuroestetica
- Saggi pubblicati:
  - Anima delle Macchine, Edalo Ed.
  - La mente liquida, Edalo Ed.





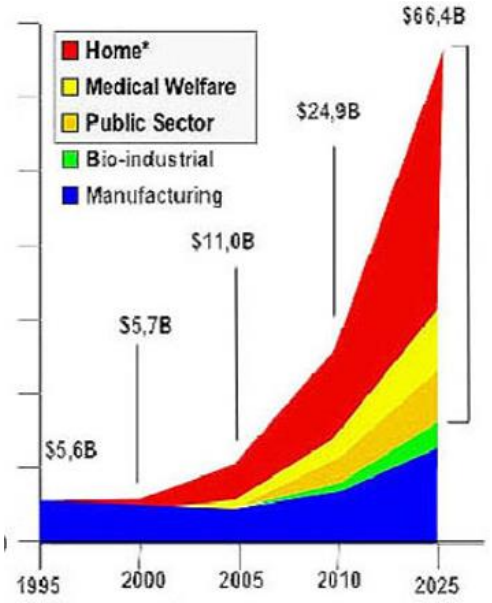
Topics trattati

- Robot collaborativi;
- Software di integrazione per la robotica;
- ROS (Robot Operating System);
- Casi applicativi.

Perché la necessità di una standardizzazione?

- Robot trend in crescita;
- I robot sono integrati in sistemi complessi;
- Necessità di intelligenza
  (difficilmente reperibile dal produttore);

Credit: www.ifr.org



Robot collaborativi

- Meccanicamente sicuri;
- Leggeri;
- Lenti;
- Cedevolezza;

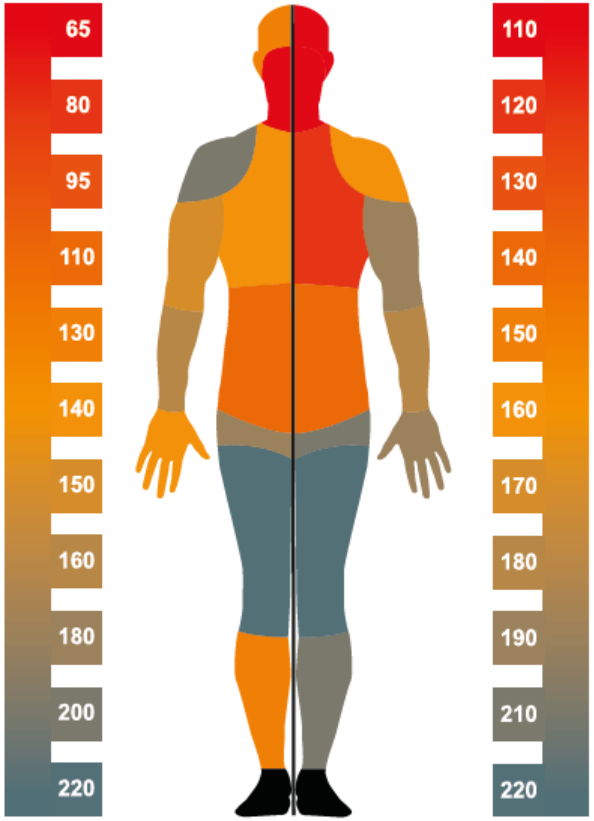
Capacità di percezione e cognitive

Diverse tipologie di Cobot

- Coesistenza
- Cooperazione
- Collaborazione

Credit:ZDNet

Sicurezza



Credit: Whitepaper II
Mensch-Roboter-Kollaboration
TUV Austria, Fraunhofer Austria and JOANNEUM RESEARCH, Vienna 2017

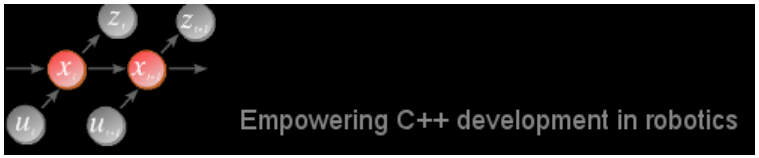
Necessità di standardadizzazione

- Robot richiedono integrazione
- Robot necessitano moduli intelligenti

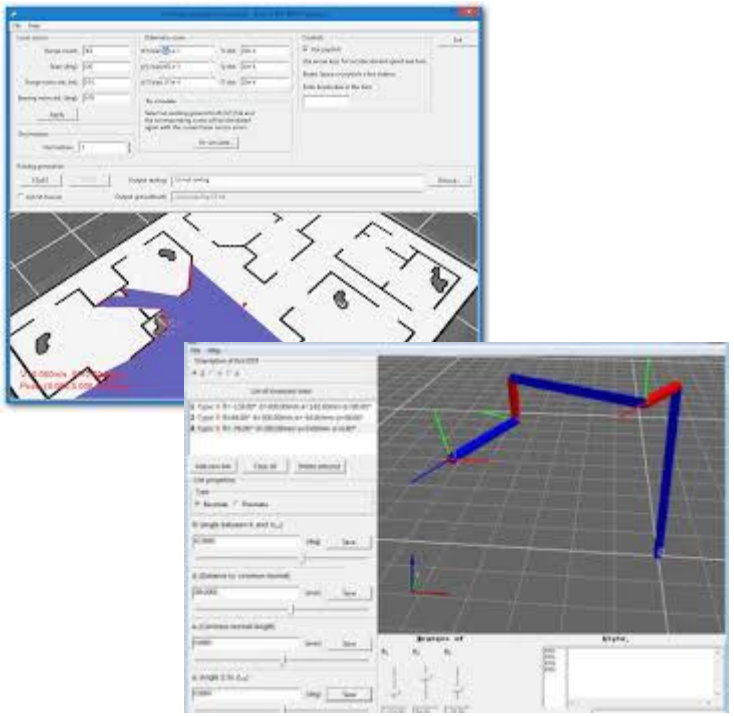
  (image processing, AI, perception)
- Sviluppare know how

Standardizzazione:
- Comunicazione;
- Analisi dei dati
- Sensoristica;

Soluzioni



- **Mobile Robot Programming Toolkit**
  - https://www.mrpt.org/



1. 3D range cameras
2. 2D laser scanners
3. 3D Laser scanners / 3D LIDARs
4. Cameras
5. Inertial Sensors / Gyroscopes
6. GPS receivers
7. Activemedia robotic bases (All ARIA-compatible bases)
8. Rovio mobile robot/webcam
9. Joysticks
10. Pan and Tilt Units
11. Range-only or RFID sensors
12. Generic I/O boards
13. Gas and Wind sensing devices

Microsoft Robotics Developer Studio(MRDS)

- a Windows-based environment for robot control and simulation
- visual programming tool
- web-based and windows-based interfaces
- 3D simulation
- easy access to a robot's sensors and actuators.
- programming language is C#.

ZeroMQ

- Funziona su linguaggi diversi e diverse piattaforme
- Messaggi su duiversi protocolli di trasmissione: IPC, TCP, TIPC
- High-speed asynchronous
- Active open source community.
- Diverse architetture: centralized, distributed, small, or large.
- Free software

ROS (Robotyics Operting system)

- Sistema operativo (gira su Linux)
- Librerie per la robotica
- Si tratta di una rete di nodi (programmi residenti nei robot o sensori)
- I nodi comunicano con messaggi

Perché ROS

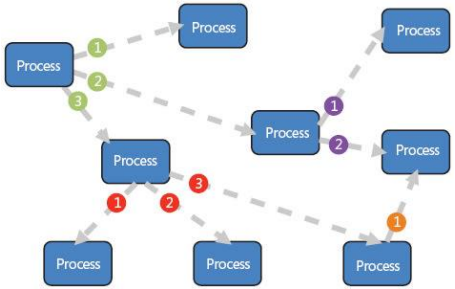
- Si allontana da una programmazione sequenziale
- Svantaggi della programmazione sequenziale
  - Non adatta per una gestione delle azioni in maniera asincrona
  - Non adatta a inviare streaming di dati a moduli multipli
- Come interviene ROS?
  - I nodi comunicano tra di loro senza coinvolgere gli altri nodi
  - Questo è realizzato tramite topics, services e actions
  - Ogni nodo ha processi autonomi (separati dagli altri)

Vantaggi nell'hardware

- Tradizionalmente, ogni volta che un nuovo modulo hardware deve essere 'collegato' agli altri, le rispettive librerie devono essere importate e deve essere costruito uno standard di comunicazione
- In ROS, dato che i nodi comunicano tra di loro attraverso messaggi standardizzati, lo standard dei dati è già realizzato.
- Si lavora a un livello di astrazione più elevato senza occuparsi dei dettagli dell'hardware.

Riassunto caratteristiche

- MODULARE
  - Event driven

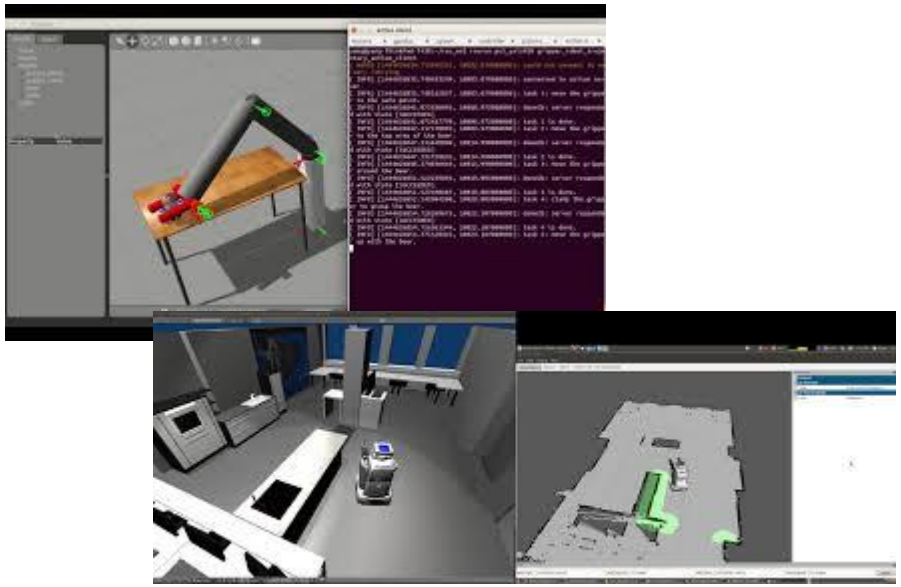


  - Asincrono
  - Nodi indipendenti

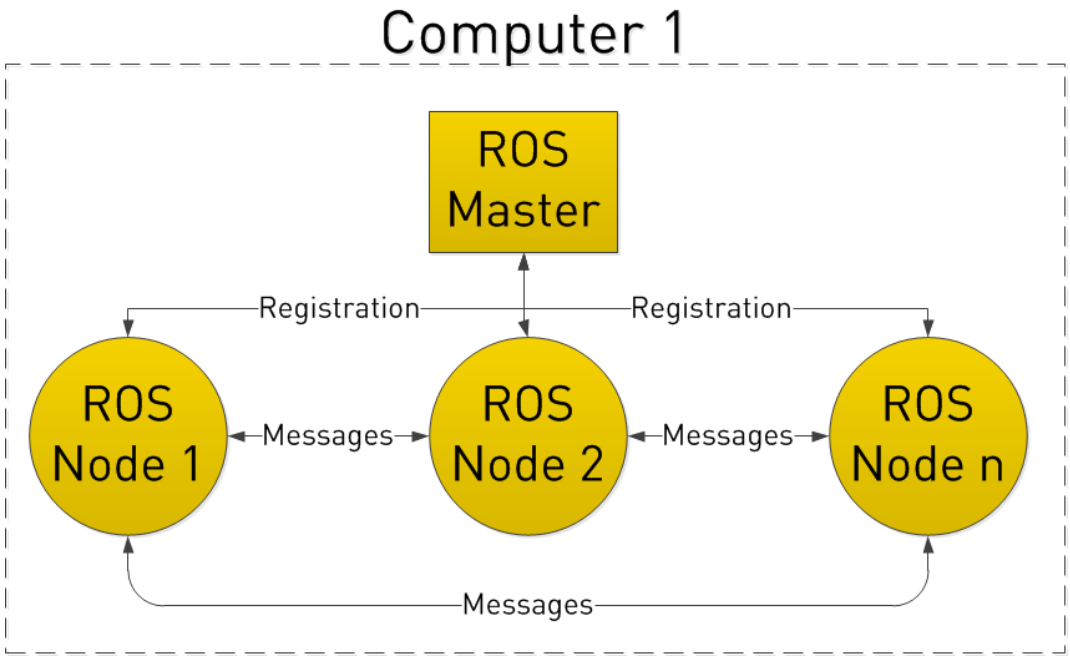
- ASTRAZIONE ELEVATA
  - L’interfaccia consiste in messaggi che abilitano la comunicazione tra diversi hardware
  - Per gestire la cumunicazione di dati non serve condividere informazioni sui driver hardware

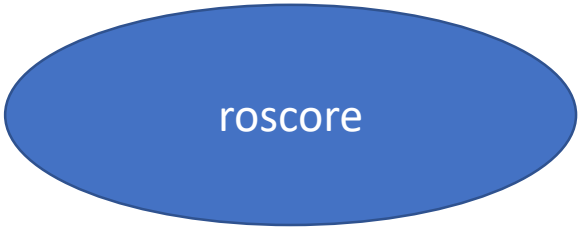
Tipici packetti

- Simulazione
- Navigazione
- SLAM
- Sensori e analisi immagine

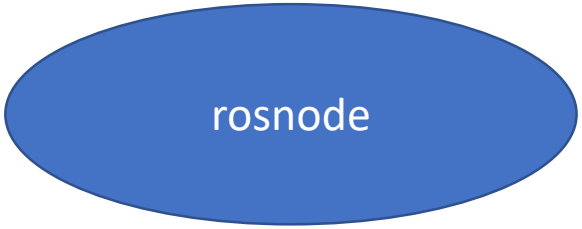


NODES


Computer 1
ROS Master
Registration
Registration
ROS Node 1
Messages
ROS Node 2
Messages
ROS Node n
Messages



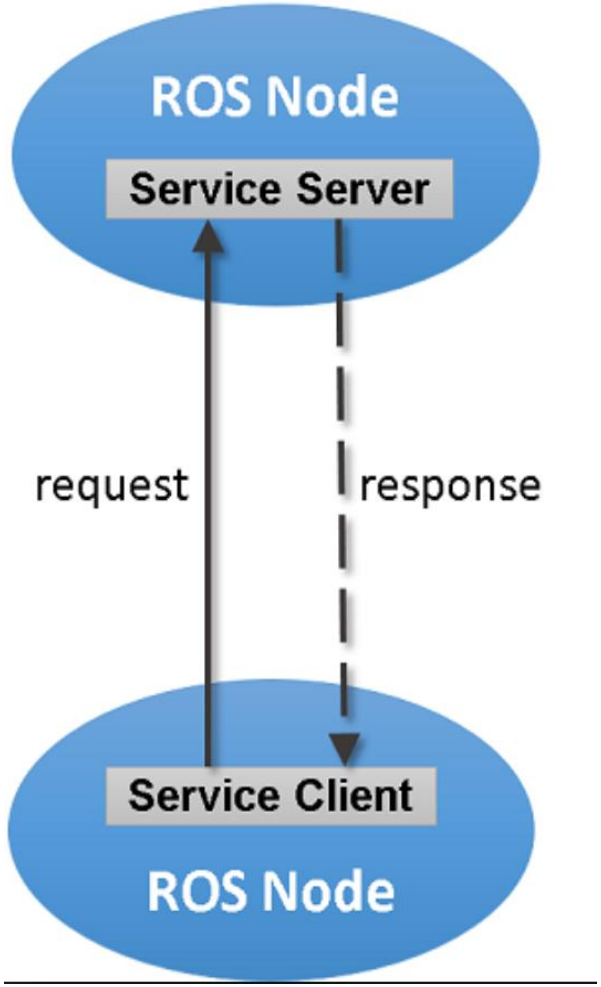
- «registra» i nodi presenti nel sistema e imposta la comunicazione tra essi



- Software implementato dall’utente o sviluppato in ROS

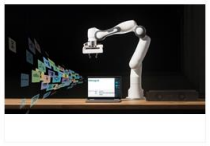
Subscriber e publisher

Servizi


ROS Node
Service Server
request
response
Service Client
ROS Node

MoveIt!

Librerie che incorporano:
motion planning, manipulation, 3D perception, kinematics, control e navigation

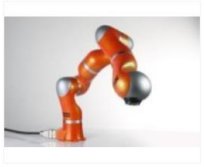


Panda
Franka Emika

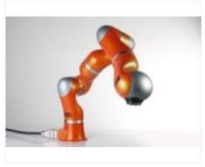
MARA
Acutronic

UR3
Universal Robots

UR5
Universal Robots



LBR
KUKA



LWR
KUKA

OmniRob
KUKA

Youbot
KUKA

IRB 1600
ABB

IRB 2400
ABB

IRB 2600
ABB

IRB 4400
ABB

Esempio (Matlab)

```
% si inizializza la rete
rosinit

% crea un publisher
pointpub = rospublisher('/miotopic','geometry_msgs/Point')


% crea un messaggio
msg = rosmessage(pointpub);

% riempie il messaggio
msg.X = 1;
msg.Y = 2;
msg.X = 0;

% spedisce il messaggio
send(pointpub,msg);

% chiude il tutto
% rosshutdown
```

Problemi di sicurezza (un nodo ha un in dirizzo IP)

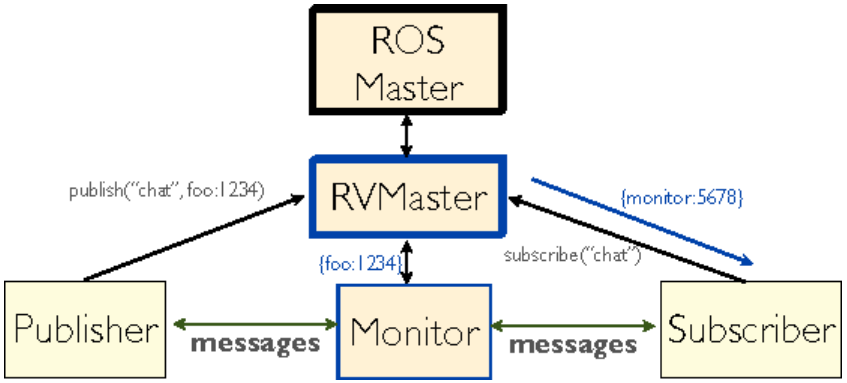
Un acker può spegnere / uccidere il nodo e sostituirlo, semplicemente eseguendo un nodo con lo stesso nome del nodo di destinazione.

Un acker può pubblicare liberamente messaggi ad un topic senza previa autorizzazione.

Un nodo non autorizzato che pubblica messaggi dannosi ad un argomento può causare un movimento imprevisto da parte di un robot che danneggia l'ambiente circostante e/o colpisce le persone vicine.

ROSRV: Runtime Verification for Robots

(le informazioni non sono criptate)



International Conference on Runtime Verification
RV 2014: Runtime Verification pp 247-254 | Cite as

ROSRV: Runtime Verification for Robots

Authors | Authors and affiliations

Jeff Huang, Cansu Erdogan, Yi Zhang, Brandon Moore, Qingzhou Luo, Aravind Sundaresan, Grigore Rosu

CryptoROS

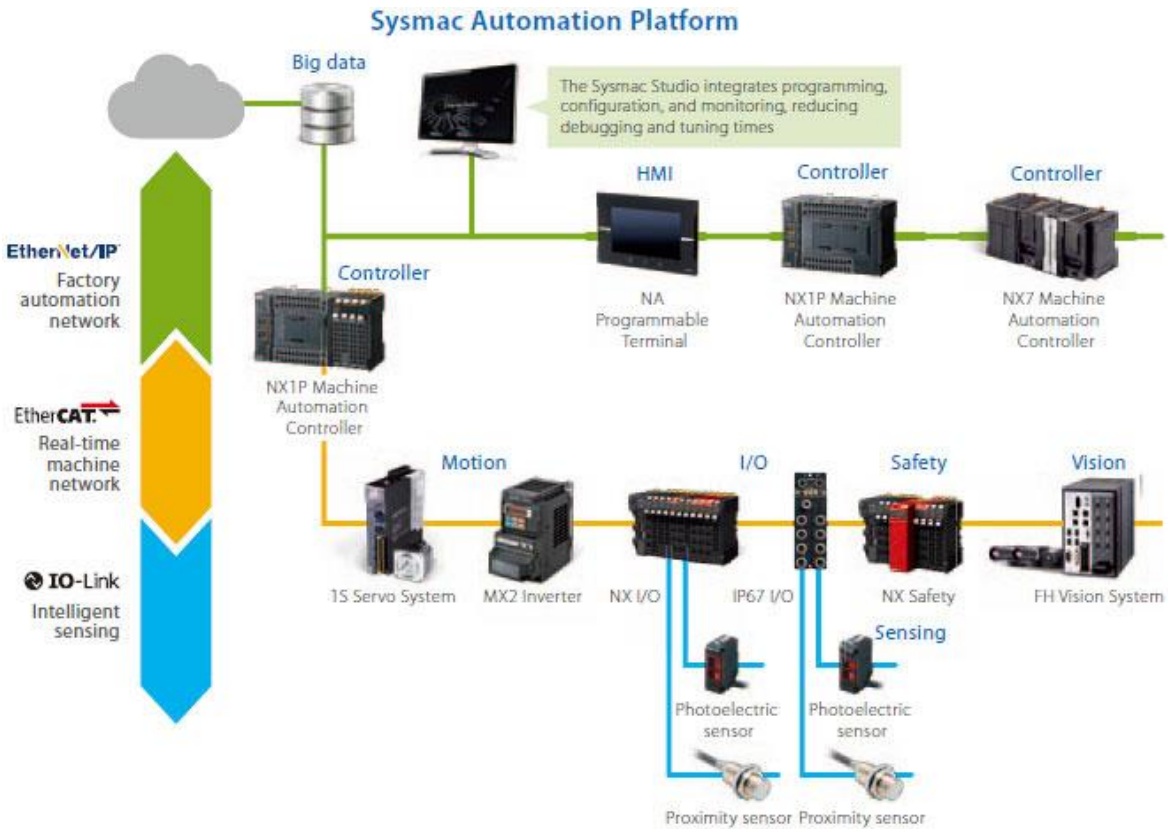
CryptoROS: A Secure Communication Architecture for ROS-Based Applications

Roham Amini[1*], Rossilawati Sulaiman[2], Abdul Hadi Abd Rahman Kurais[3**]
Faculty of Information Science & Technology
Universiti Kebangsaan Malaysia
Bangi Selangor, MALAYSIA

ROS non è real-time
è difficile implementare task sincronizzati

Credit: Chicago Automation

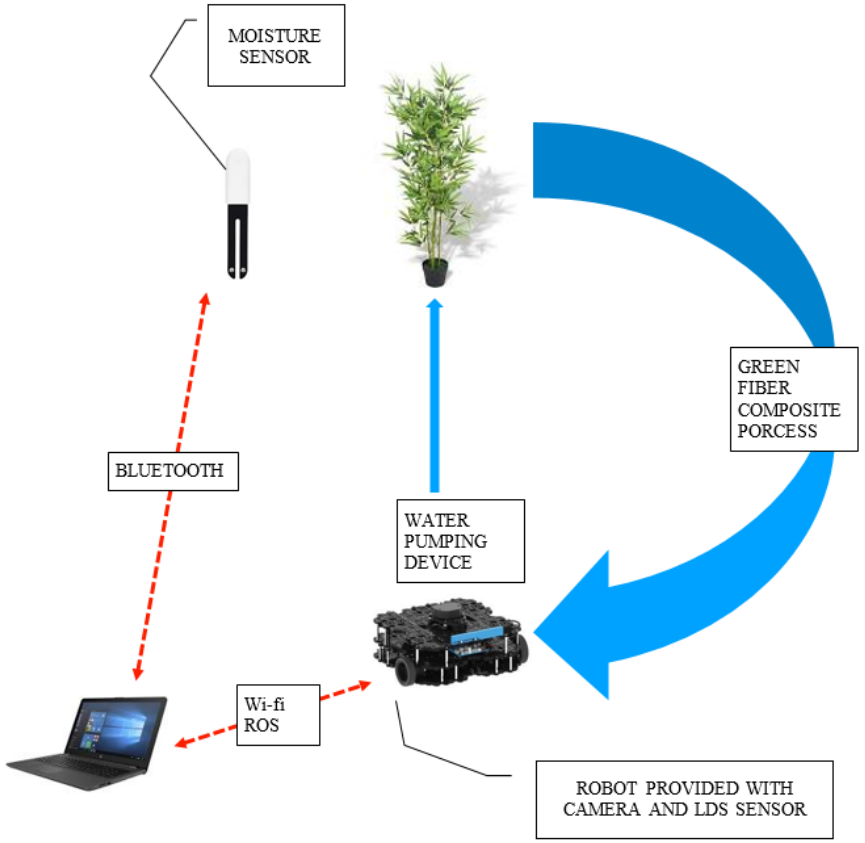
Comunicazione a basso livello

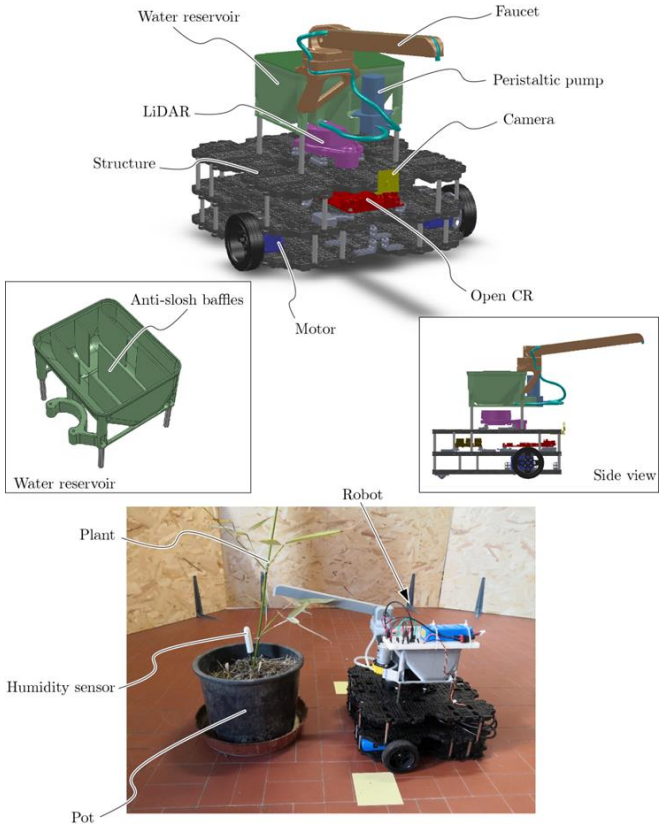


ROS 2

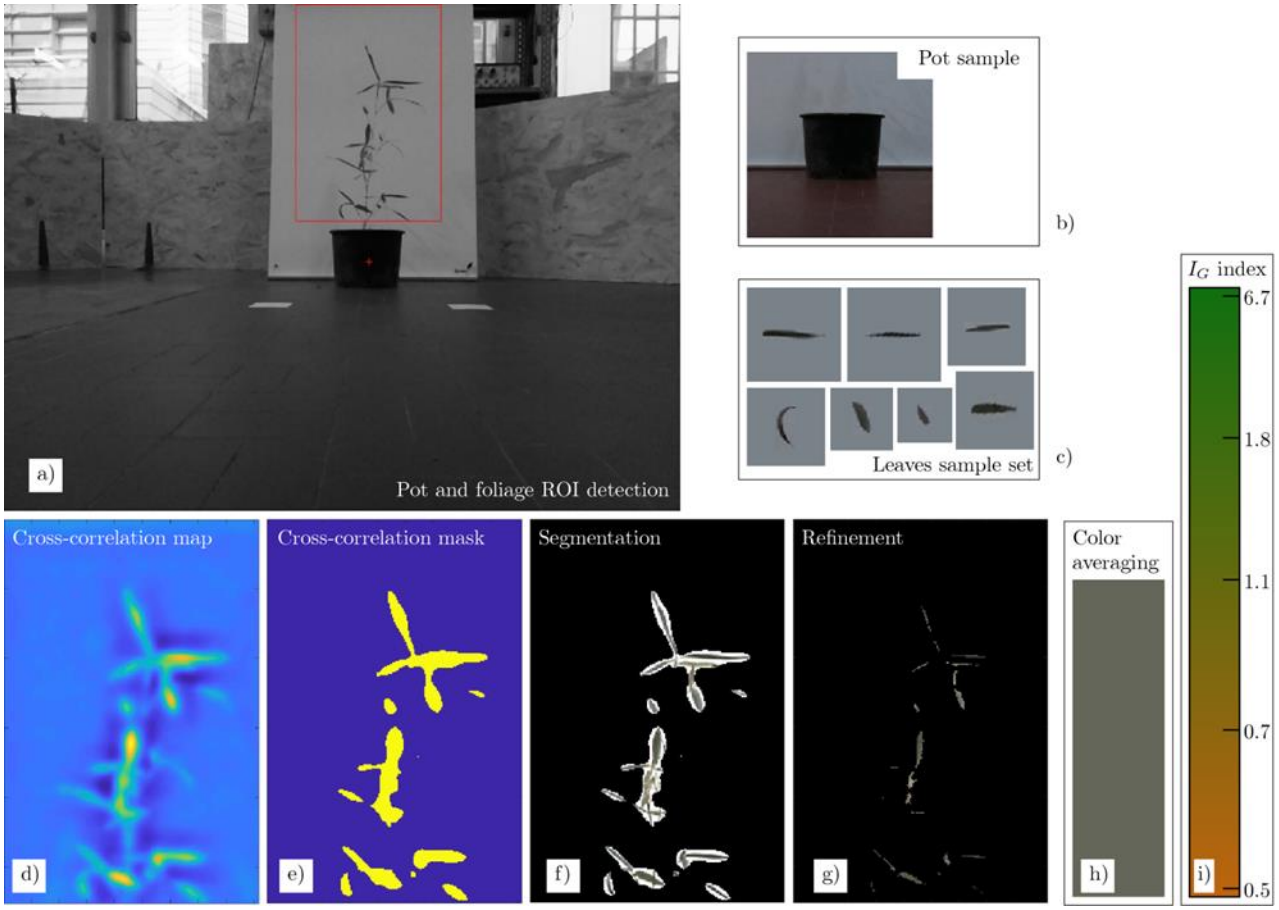
- A real-time ROS project
- Synchronization
- Scheduling

Case study1: Self Replicating Robot: a robot from cultivate plants.

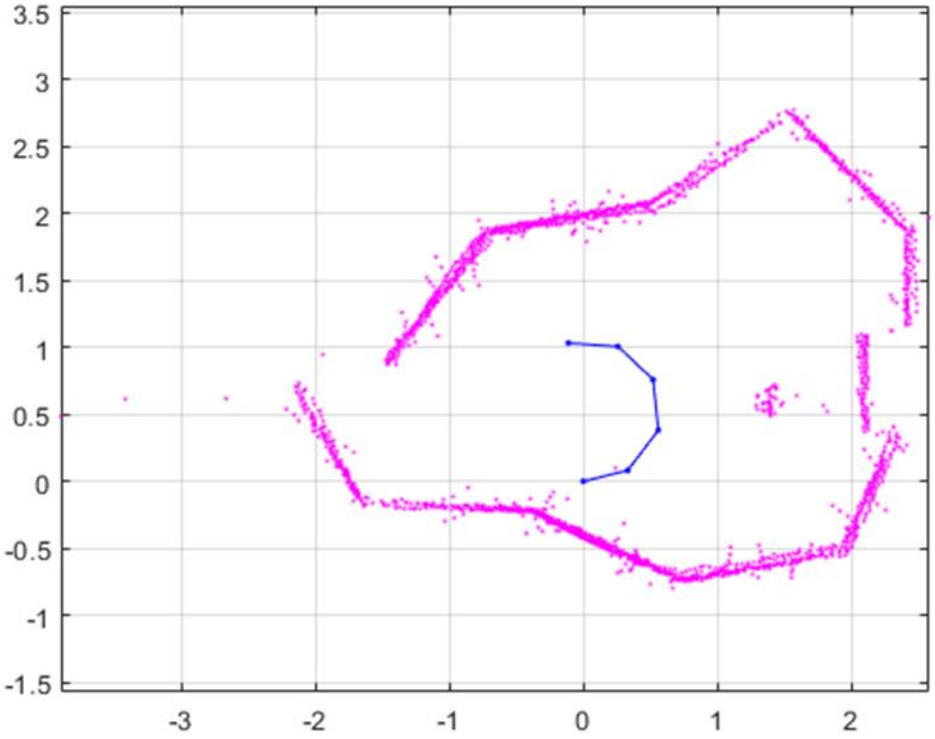




Vision system

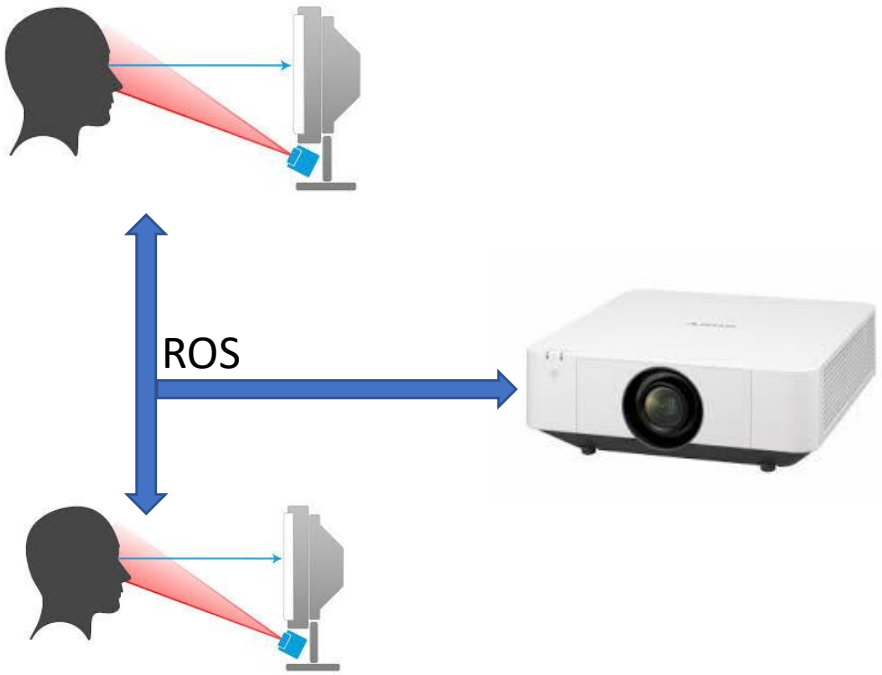


SLAM(Simultaneous Localization and Mapping) and ROS



Case study 2:

- 2 PC sono dotati di eye-tracker



Human-Human collaboration art
di Paolo Gallina & Pier Giorgio De Pinto
In collaborazione con DIA Dipartimento di Ingegneria ed Architettura della Università di Trieste e IED Milano per Milano Digital Week.
17 marzo 2019, Base, Milano, Italia.