Robotic Systems integration for cleanroom SRF assembly at Fermilab

Alessandro Ciaramella

Italian Summer Students Final Report

September 28, 2023

Supervisors:

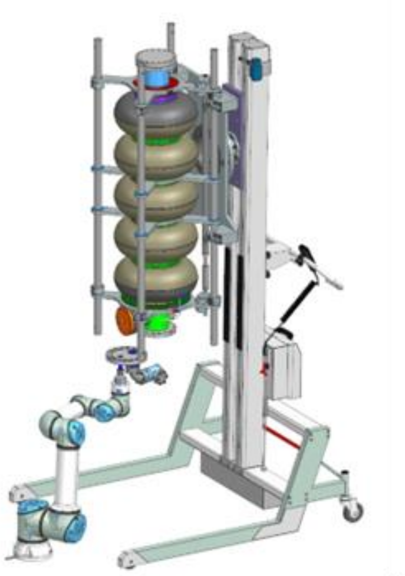
Dr. Donato Passarelli

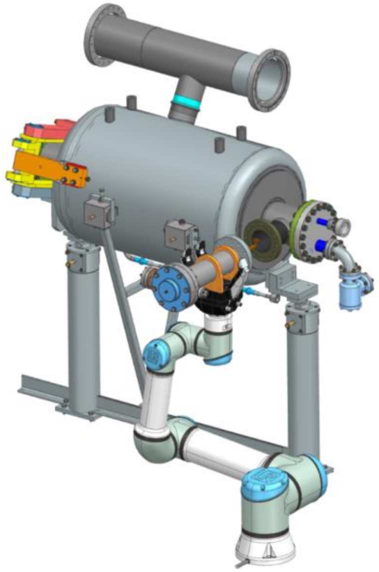
Dr. Genfa Wu

Current state of assembling procedure in cleanroom

Manual operations

- Human-driven activities remain the major source of uncontrolled particulate
- Precise manual alignment of components often exceeds expected timeframes





Robotically assisted operations

- Reduce the risk of chemical and particulate contamination during critical assembly steps
- Enhance efficiency and mitigate existing challenges for better product quality, performances repeatability and workers safety

Incremental development

Sources of risk

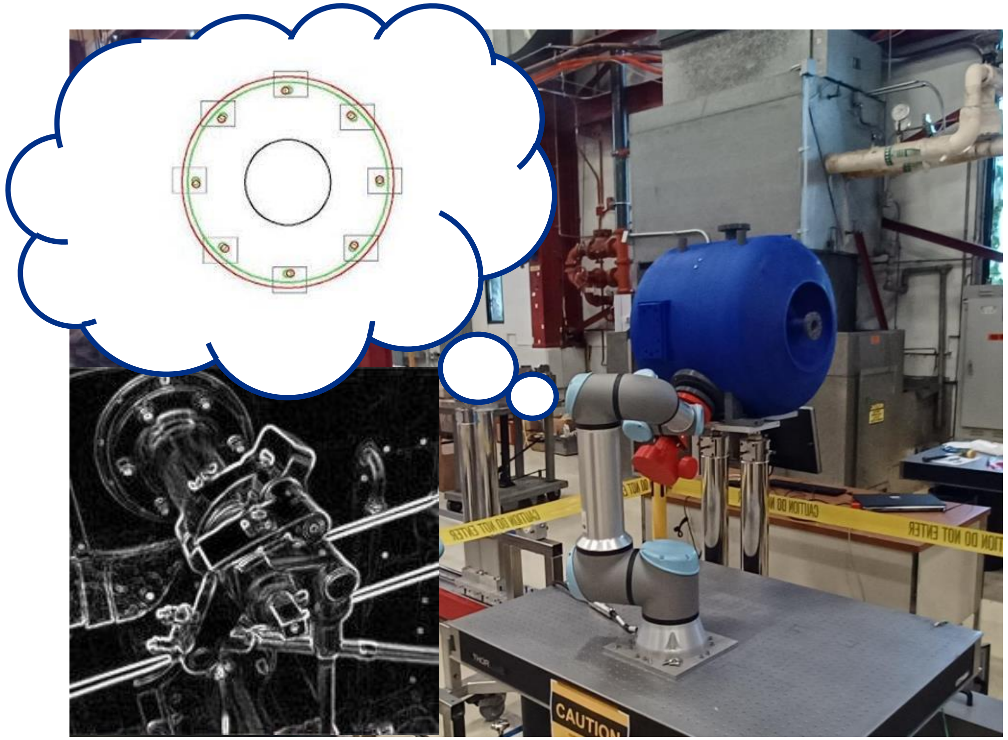
A) Proximity of humans to beamline aperture
B) Lack of precision in manual alignment

Mitigation strategy

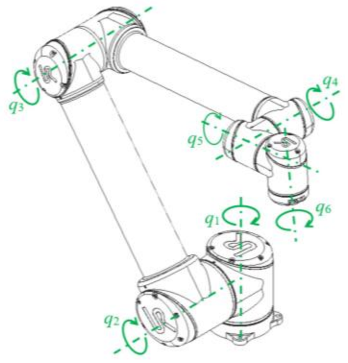
A) Robot as a technicians' tool for acting at distance
B) Robot performing autonomous alignment

A)

B)



A) Remote control

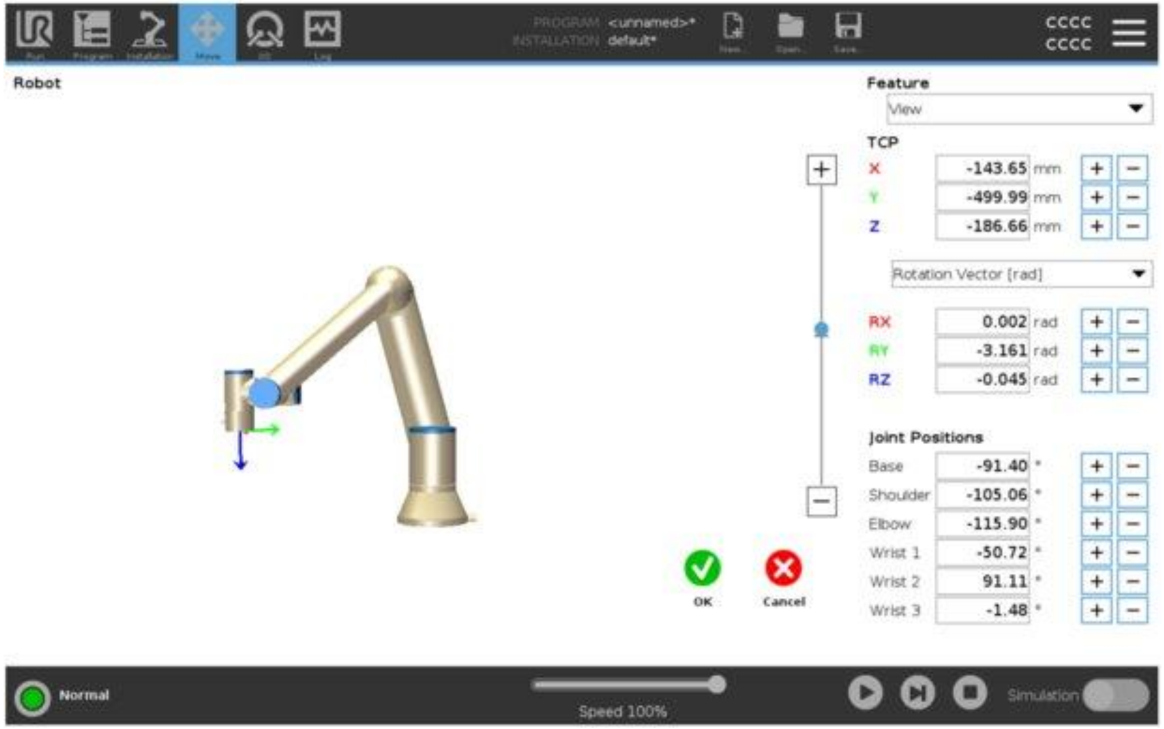


Teach pendant

Pros: Short-term feasible, plug and play, certified safe, easy to learn

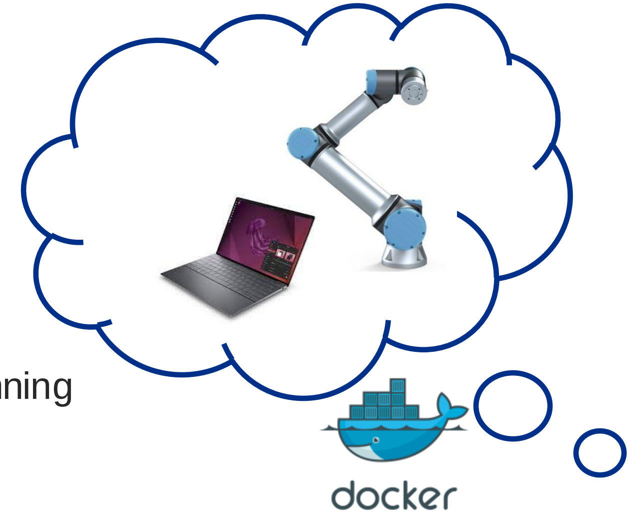
Cons: limited functionalities (or not direct to achieve)

Remote control touch screen interface

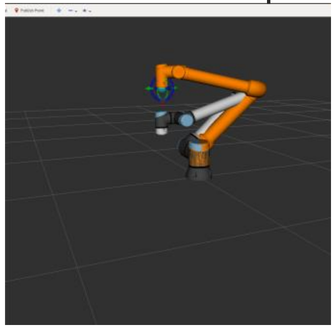


B) Robot System Integration

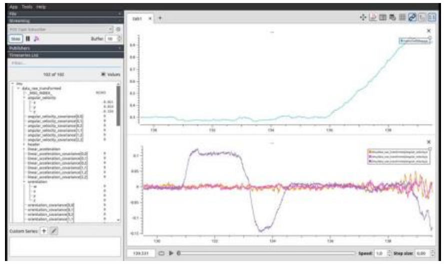
Containerized full system simulator



Collision avoidance planning



Live sensors monitoring



ros2

Pros: Modular, easier to maintain, possibility to use open-source code written by others

Cons: harder learning curve, long-run

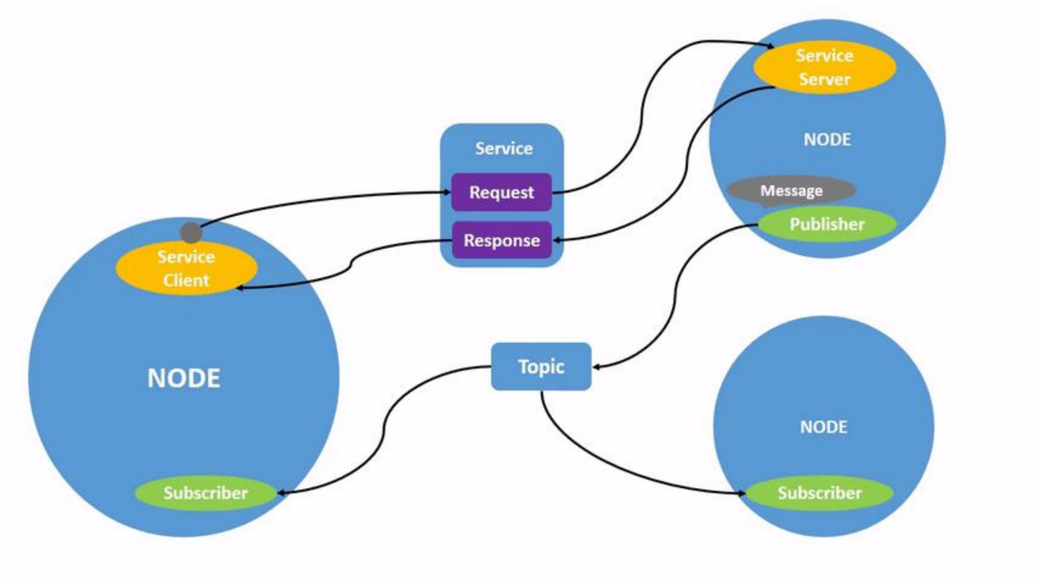
Additional packages:

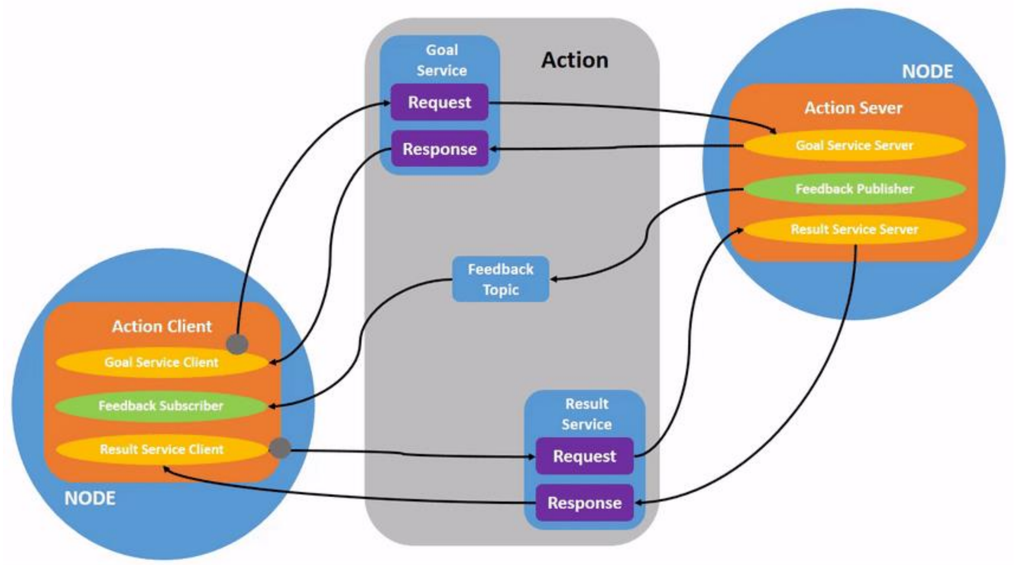
- Compliance controller
- Teloperation controller
- Computer vision

ROS in a nutshell

The Robot Operating System (ROS) is a set of software libraries and tools for building robot applications. From drivers to state-of-the-art algorithm.

Computation is performed in a peer-to-peer network of processes, the ROS Graph.
The fundamental concepts of ROS are nodes, messages, parameters, topics, services and actions.





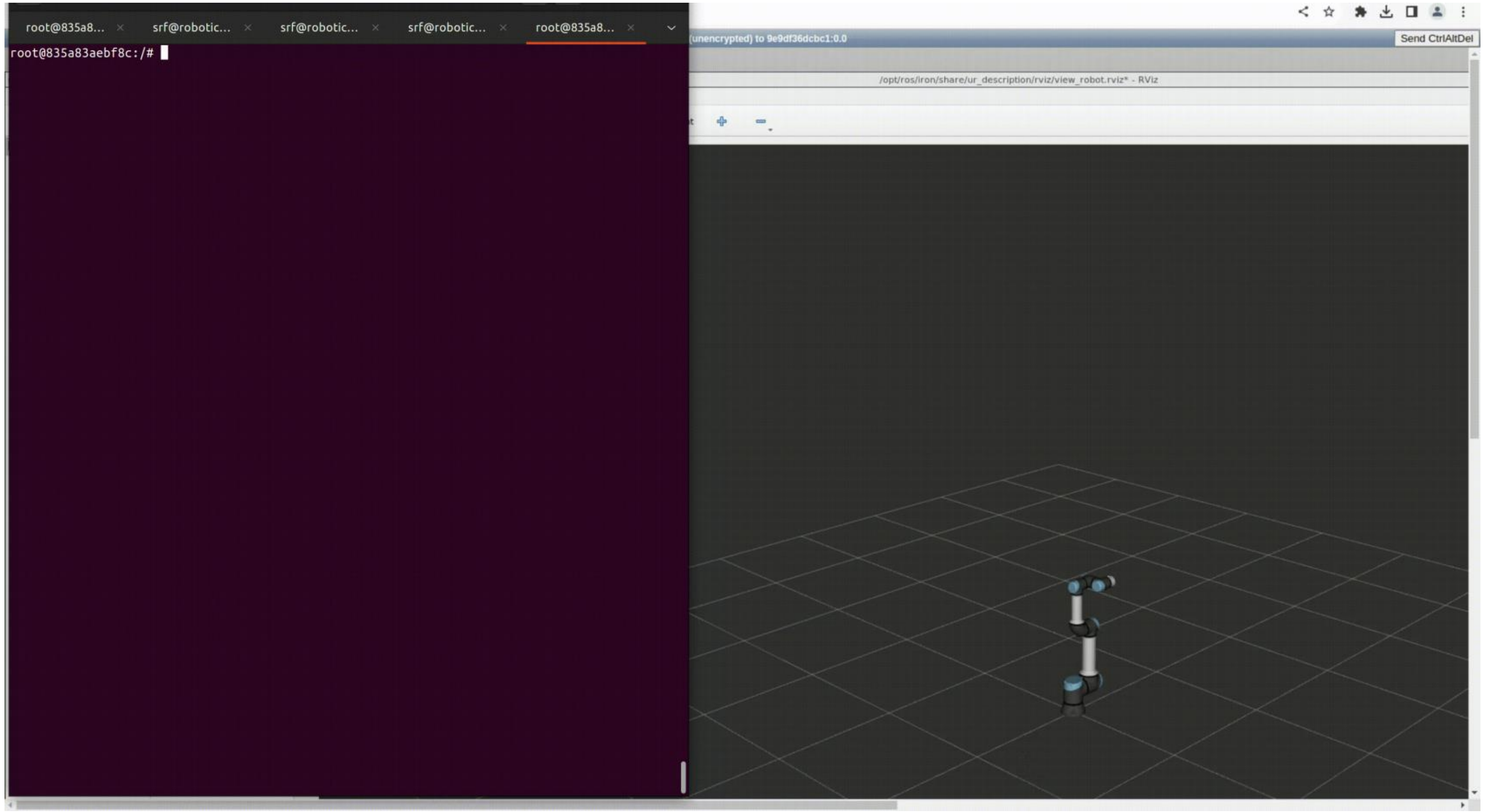
Interfaces for high-level control

Input interface

Read the sensors and acquire information about the actual state of the robot (ur_driver topics subscribing)

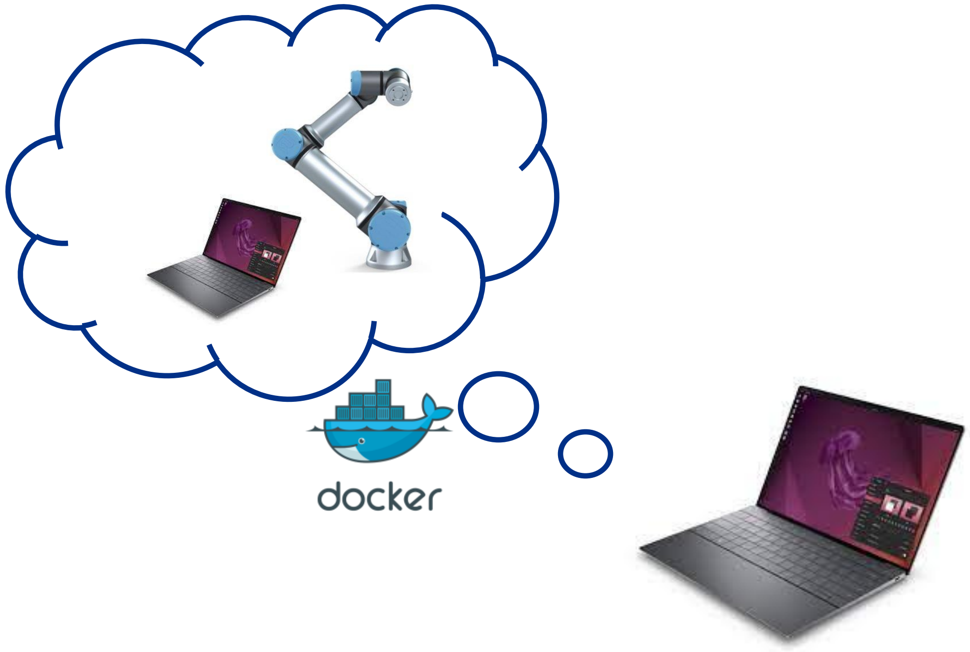
Output interface

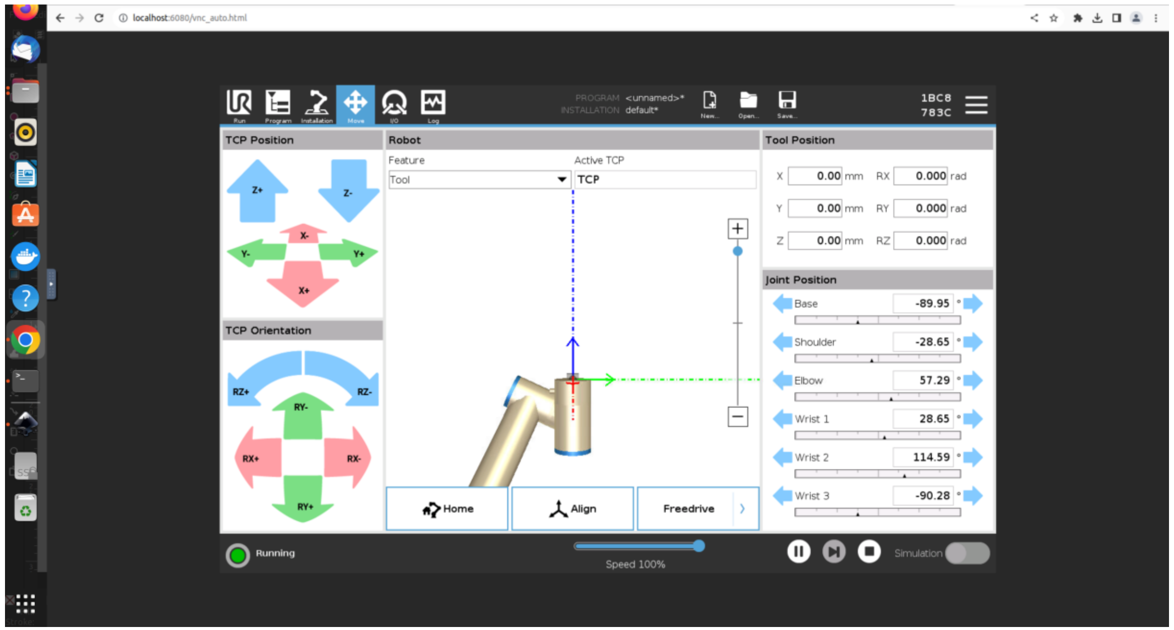
Command the robot to reach desired states, following polynomial trajectories (ros2_control action goal sending)



Simulator

The simulator system consists of 5 docker containers connected through a bridge network.
There is no difference between real robot driver interface and simulator one.
All the tools developed for simulator are directly available for real robot too.





Fermilab

Procedure for coupler-cavity assembly

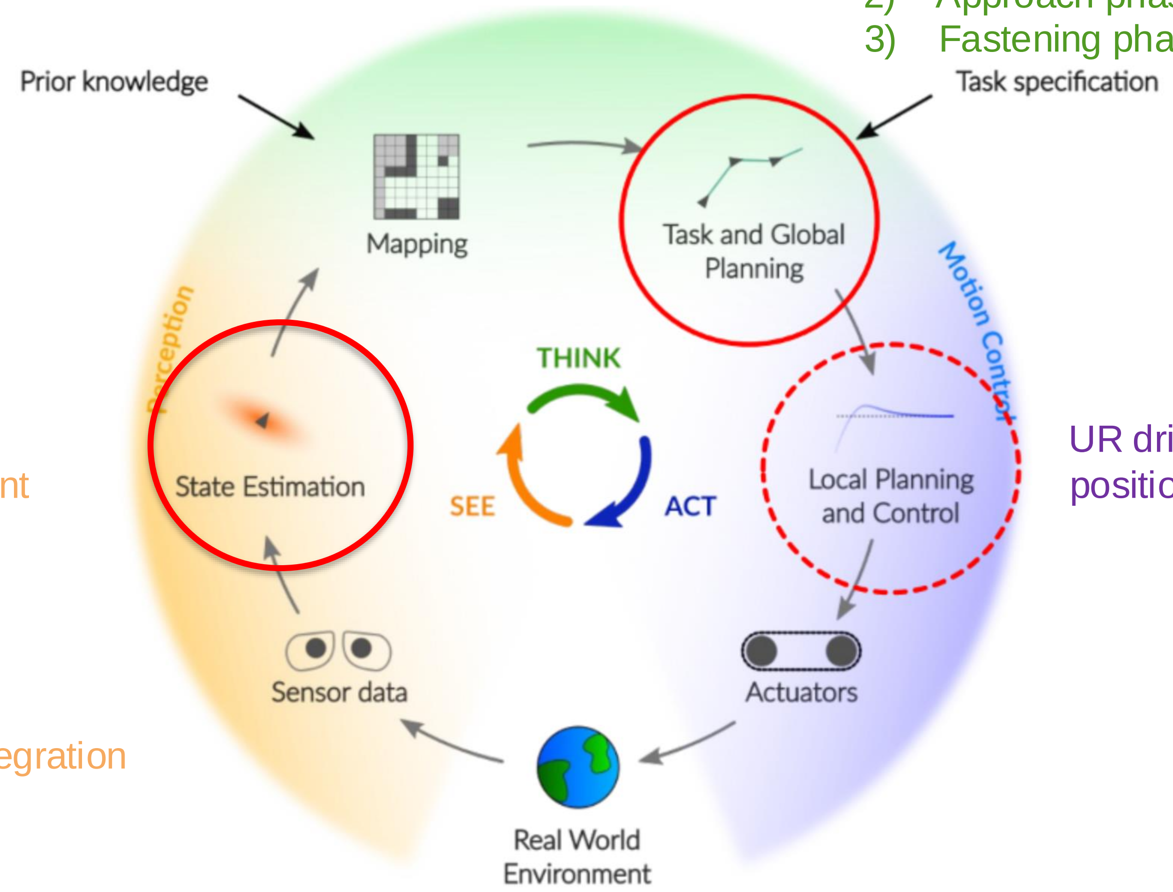
1) Move to arm-coupler installation predefined position
2) Activate brake, perform installation and change payload accordingly, deactivate brake
3) Move in the predefined approach position
4) Perform alignment using remote control or computer vision, save target position
5) Move to predefined cleaning position
6) Clean (dry blow with filtered N2 using a particle counter until a certain threshold).
7) Move back to target position
8) Push flanges against each other using force control or compliance control
9) Use fasteners for seal, change payload according to tool load cell data and uninstall coupler from arm tool.
10) Move to home position

Cleanroom demonstration

The robot will be moved in the cleanroom

Future tasks

1) General movement: predefined waypoints and trajectory interpolation
2) Approach phase: visual servoing
3) Fastening phase: compliance control



Pose estimation
online confidence assessment

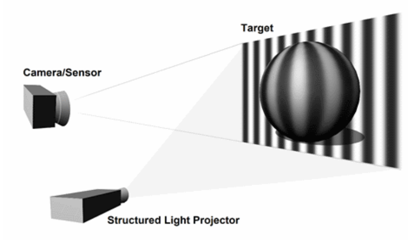
UR driver and Real-Time Data Exchange for position, velocity and compliance controllers

Camera integration

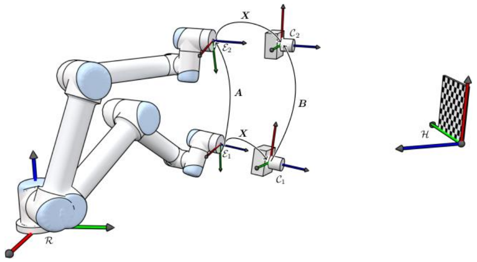
1) real hardware: cleanroom setup
2) simulated hardware: environment description and integration in simulator

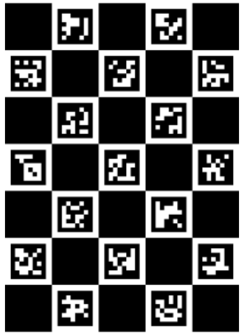
Computer Vision System for Cleanroom SRF Assembly

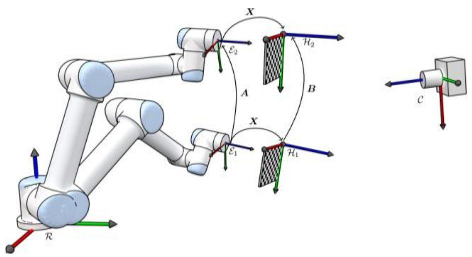
1) Choose camera (Intel D405? Different technology?)



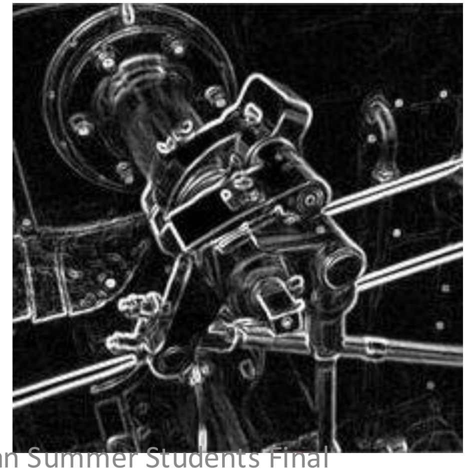
2) Develop calibration setup and procedure to measure both intrinsic and extrinsic parameters (start using ArUco and/or Chessboard)

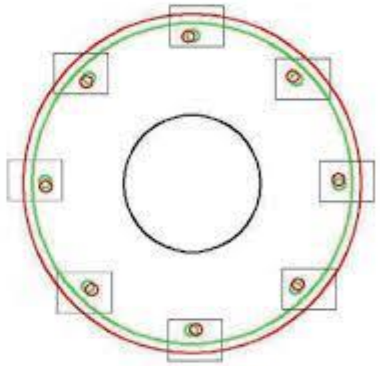






3) Program an effective algorithm for feature-based extraction (start with Canny algorithm, then use more robust model-based one), matching and pose estimation (PnP + RANSAC). No constraint on execution time, maximize accuracy.

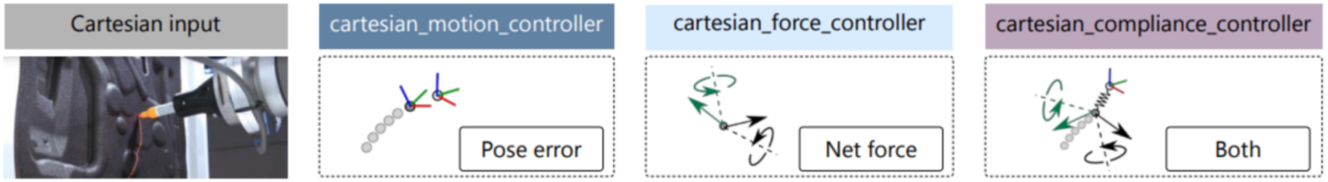




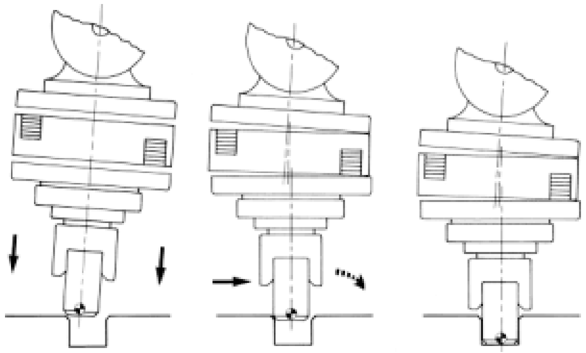
Optional) write code for ros2 nodes that: a) fetch camera data and publish on "/image_topic", b) subscribe to "/image_topic", run vision algorithm and publish on "/estimated_pose_topic" c) compute relative transformation using tf2 tools)

Safe and reliable motion control for Cleanroom SRF assembly

1) Integrate cartesian controllers in the system (use cartesian_controllers package by FZI)



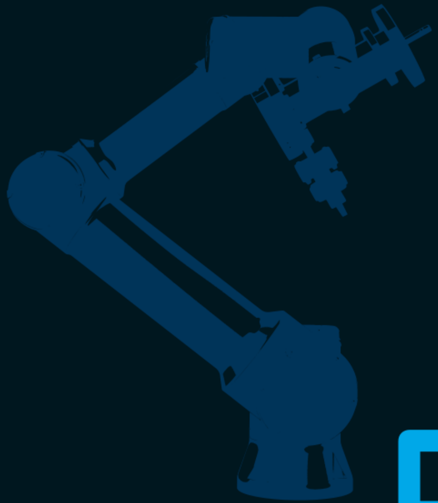
2) Fine tune all the stiffness parameters for optimal response during sealing and fastening



3) Prepare a user-friendly interface (safe joystick) for technicians training and operation

QUESTIONS?

Fermilab



SRF ROBOTICS