Exploiting ROS2 to facilitate end-effectors integration and control

Liana Bertoni$^{1,3}$, Davide Torielli$^{1,2}$, Nikos Tsagarakis$^1$, and Luca Muratore$^1$

1 Humanoids and Human Centered Mechatronics (HHCM), Istituto Italiano di Tecnologia, Genova, Italy
2 Department of Informatics, Bioengineering, Robotics, and Systems Engineering (DIBRIS), University of Genova, Genova, Italy
3 Dipartimento di Ingegneria Informatica (DII), University of Pisa, Pisa, Italy
• High Number of different End-Effectors

• Large variety of structures (kinematic, actuators, interfaces)

• Complex integration
• High Number of different End-Effectors

Develop a **framework** aimed to abstract each end-effector from its mechanical instructions and **plan at higher level**

• Complex integration
• ROS-based hardware-agnostic control framework

• Simplify integration of new End-Effectors
  *(few model description information required)*

• Automatic extraction of End-Effector capabilities
  *(Primitive Grasping Actions)*
FRAMEWORK

EXTRACTION PHASE

OFFLINE

EXECUTION PHASE

ONLINE
FRAMEWORK

OFFLINE

URDF
SRDF
FRAMEWORK

OFFLINE

PRIMITIVE GRASPING ACTIONS EXTRACTOR

URDF SRDF

Primitive Grasping Action
FRAMEWORK

ONLINE
 FRAMEWORK

Initialization

Grasping Action Command

ROS End-Effector GUI
Other Nodes

ROS End-Effector Executor

ONLINE
FRAMEWORK

Initialization

Grasping Action
Command

Motor Position
Reference

Low-Level
Reference

ROS End-Effector
GUI

Other
Nodes

ROS End-Effector
Executor

ROS End-Effector
HAL

ONLINE
FRAMEWORK

Initialization

Grasping Action Command

Motor Position Reference

Low-Level Reference

Grasping Action Feedback

Motor Position Feedback

Robot Feedback

ROS End-Effector GUI

Other Nodes

ROS End-Effector Executor

ROS End-Effector HAL

URDF SRDF

Primitive Grasping Action

Custom Grasping Action

ONLINE
ROS 2 INTEGRATION
ROS 2 INTEGRATION

Topics, Services, Actions

Launch Files

<xml/> catkin

<xml/> colcon ament
ROS 2 INTEGRATION

Topics, Services, Actions

<xml/> Launch Files
<xml/>

catkin ➔ colcon

Parameter Server ➔ ❌
ROS 2 INTEGRATION

**ROS** ➔ **ROS 2**

- Topics, Services, Actions
  - `<xml>` Launch Files
  - `<xml>`

- **Parameter Server** ➔ **X**

- **MoveIt**
  - Collisions (FCL)

- **MoveIt 2**
  - Collisions (FCL)

- Standard C methods (`dlopen`)
  - For HAL dynamic loading ➔ **👍**
  - ROS2 PluginLib
  - For HAL dynamic loading
The ROS End-Effector HAL abstracts the low-level details of the end-effector.
• ROS-Industrial Focused Technical Project (ROSIN-FTP)  
  https://www.rosin-project.eu/ftp/ros-end-effector

• ROS End-Effector source code  
  https://github.com/ADVRHumanoids/ROSEndEffector  
  https://github.com/ADVRHumanoids/ROSEndEffector2

• ROS End-Effector documentation  
  https://advrhumanoids.github.io/ROSEndEffectorDocs/

• Available in the ROS official repository :  
  ros-melodic-end-effector ros-noetic-end-effector (ROS)  
  ros-foxy-end-effector (ROS2)


THANK YOU FOR YOUR ATTENTION!

ANY QUESTIONS?

Exploiting ROS2 to facilitate end-effectors integration and control

Davide Torielli, Liana Bertoni, Nikos Tsagarakis, and Luca Muratore