Title:	Self-reconfiguration of a robotic workcell for the recycling of electronic waste
Acronym:	ReconCycle
Type of Action:	Research and Innovation Action
Grant Agreement No.:	871352
Starting Date:	01-01-2020
Ending Date:	31-07-2024

Deliverable Number:	D4.3
Deliverable Title:	Reconfigurable/adaptable soft effector control methods
Type of Deliverable:	Report
Dissemination Level:	Confidential
Authors:	Riccardo Persichini, Manuel G. Catalano, Vinicio Tincani
Contributing partners:	IIT, QBR

Estimated Date of Delivery to the EC: 31-07-2023

Actual Date of Delivery to the EC: 04-08-2023

Executive Summary

This deliverable provides a detailed report on the low-level control strategies and user interfaces of modular end-effectors developed within the ReconCycle project. These controllers are integrated into the firmware layer of the devices and made accessible to users through both the middle-level layer for integration in C++ or ROS environments and specific graphical user interfaces. These controllers form the foundation for integration with high-level trajectory and task planners developed within the project.

Table of Contents

1 Introduction

The two novel end-effectors developed within the ReconCycle project exhibit peculiarities in the low-level control of electric prime movers. These peculiarities need to be addressed at the firmware level to provide users with proper and easy-to-use control commands that can be utilized in high-level planning and programming. Both the SoftHand 2 and VS Gripper feature two motors, enabling the synergistic, multi-shape grasping modality of the SoftHand 2 and the variable stiffness behavior of the VS Gripper. Moreover, the VS Gripper is also equipped with tactile sensing systems that must be integrated in the control framework of the system and exposed as well to users to allow the use of the perceived contact information.

This deliverable offers a comprehensive report on the low-level control strategies and user interfaces of modular end-effectors developed as part of the Reconcycle project. These controllers are integrated into the firmware layer of the devices. Users can access them through both the middle-level layer for integration in C++ or ROS environments and specific graphical user interfaces. These controllers serve as the foundation for integration with high-level control planners developed within the consortium.

2 SoftHand 2

IIT and QBR explored new technical solutions about robotic hands, in particular, taking over the design of the *SoftHand 2* and bring the prototype (TRL 4) to become a product marketed in May 2022 (TRL 9) with the name *qb SoftHand 2*.
The introduction of the two-motors actuation allows the qb SoftHand2 Research to exploit the principles of first and second synergies in an intrinsically intelligent design, which is not only safe w.r.t. unexpected human-robot interaction but also adaptable to grasp and manipulate different shaped objects, showing an unparalleled level of simplicity and flexibility. Combining the two synergies and/or the movements of motors, qb SoftHand2 Research can perform different gestures, as shown in Figure 1.

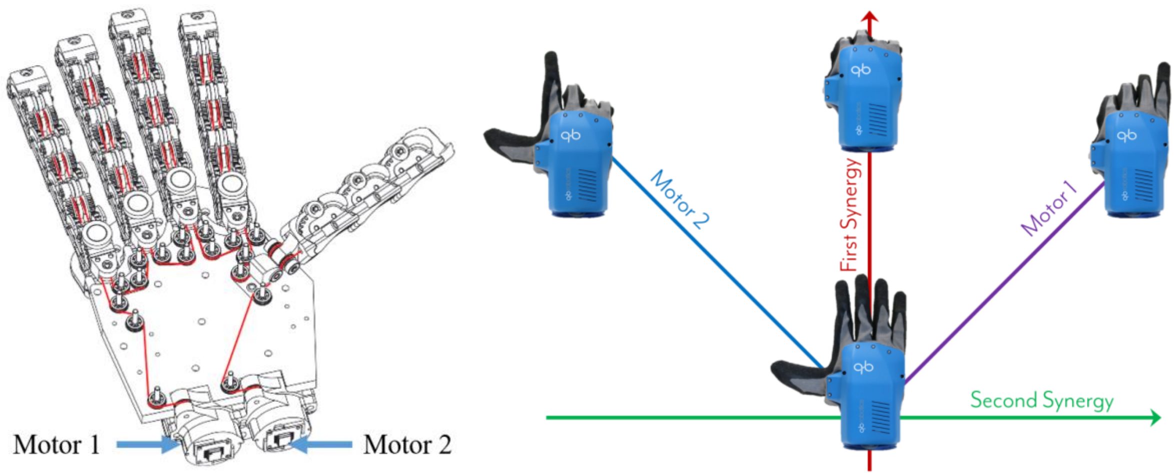

Figure 1 Transmission system with two motors. There is a single tendon passing though the hand, starting from a motor pulley up to the other one. The disposition of free pulleys on the palm, i.e. the layout of the transmission system, determines the kinematics of the whole hand.

The new grasping skills, especially the more precise "pinch grasp", allow the hand to better grasp small objects such as batteries, which are often installed in electronic devices that need to be disassembled. The new features of qb SoftHand 2 Research can be observed via the following link: https://www.youtube.com/watch?v=tKnWyoWBwac.

2.1 SoftHand 2 Control Interface

The SoftHand2 GUI is a simple application to quickly test the basic functionalities of the device through a personal computer. It may be also useful to diagnose troubles about hardware or software components. When a serial port is found, the ***"Connect"*** button becomes available. By clicking on it, the user can scan the serial resource to find every qbrobotics device connected to the system. If there is at least one device found, this operation enables all of the GUI features and it shows a green Connected label next to the button.

The button ***"Activate"*** activates or deactivates the motor driver on the qb SoftHand2 Research; the current activation status is shown next to the Active/Inactive button.

Figure 2 shows the **Basic tab** of the application, with the following features:

- *Measurements and Currents*: The three buttons "Get Measurements", "Get Currents" and "Get Synergies" enable auto-refreshing threads that respectively show the encoder measurements, the motor current (in milliamperes) and the actual synergies.
- *Sliders*: Here one can set the synergies of the device.
- *Inputs*: Set directly reference signal to the device's motors.
- *Home*: Moves the device to the start position.

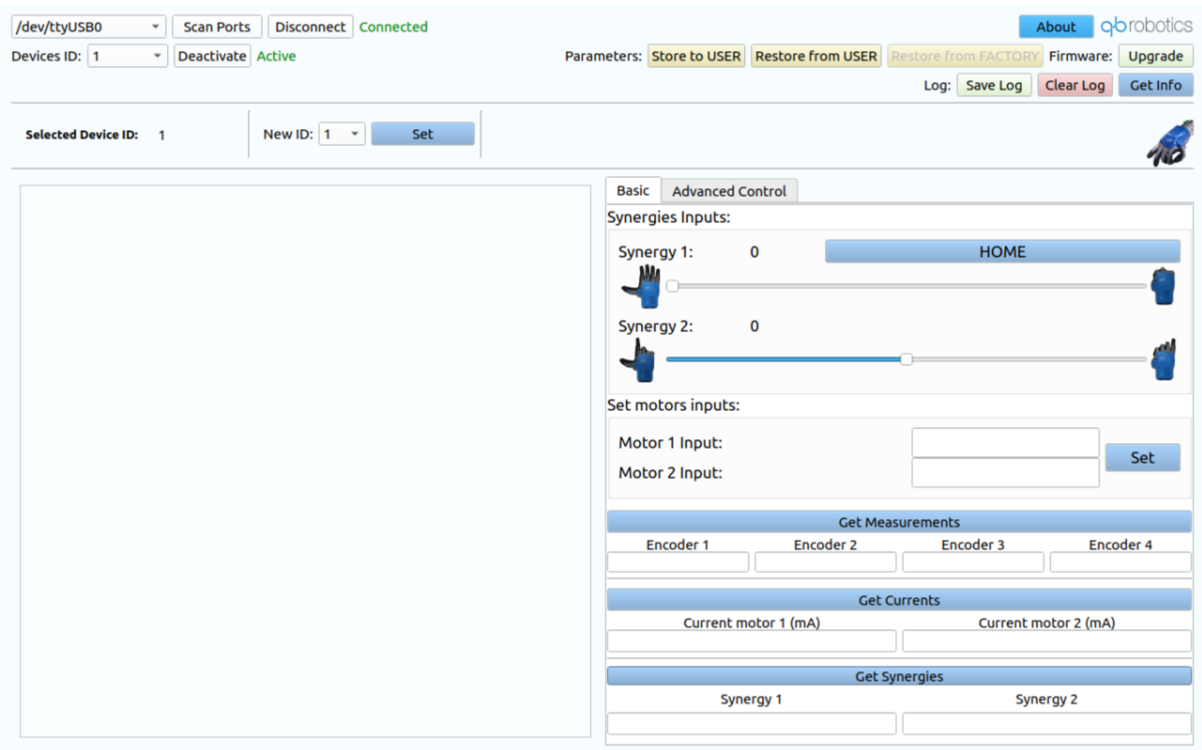

Figure 2 SoftHand2 GUI - Basic tab

Another interactive tab is the Advanced control tab, shown in Figure 3

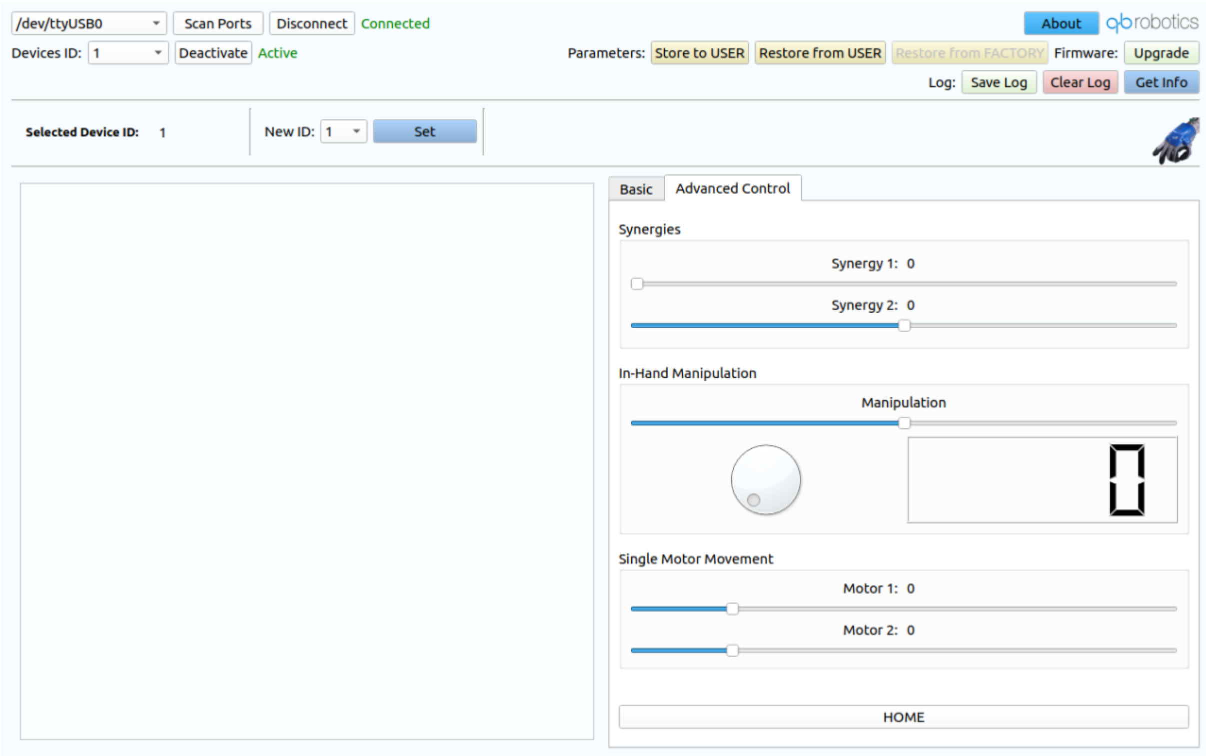

Figure 3 SoftHand2 GUI – Advanced control tab

In this interaction board there are:

- *Synergies sliders*: Here one can set the synergies of the device.
- *Manipulation slider and dial*: This is a simple example of how to control motors in order to perform object manipulation. Using this slider, one motor moves in the opposite direction of the other but not by the same amount.
- *Motor sliders*: It is possible to set the motor references to the device.
- *Home*: Moves the device to start position.

3 Variable Stiffness Gripper

The VS gripper (Figure 4) is based on the use of variable stiffness actuators systems that implement in their architecture an agonistic-antagonist non-linear elastic mechanism which allows the implementation of the physical change of the grasping stiffness. The two motors can be controlled to allow a couple of behaviours, as explained hereafter.

We implemented the following modalities:

Position: This control mode most closely approaches the operation of a classic gripper. In this modality, the user can control both position and stiffness. The first parameter is used to set the angular position of the movable finger, while the second one allows the elasticity of the finger itself to be adjusted. Higher values of stiffness indicate greater stiffness of the gripper during the grasp; on the other hand, lower values allow soft grips.

Deflection: This mode allows the user to regulate the gripping force during the grasp. In this modality, the movable arm is controlled to close completely towards the fixed part for each commanded value. Small values enable the grasping of fragile and low-weight objects; on the other hand, higher values can be used for heavy and rigid objects.

Figure 4 The first VS Gripper with DEFLECTION control: soft grasp on the left and strong grasp (high deflection reference) on the right side.

3.1 VS Gripper Control Interface

The VS Gripper GUI (Figure 5) is a simple application to quickly test the basic functionalities of the device through a personal computer.

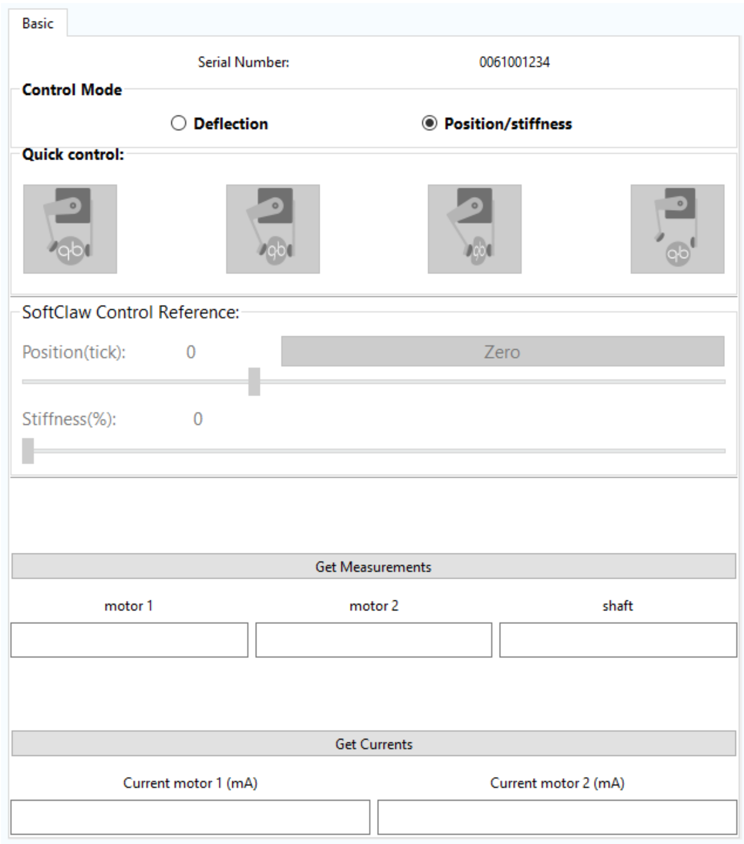

Figure 5 VS Gripper GUI

- *Control Mode*: choice of the deflection or position/stiffness control modality.
- *Quick control*: smart buttons for deflection control with preconfigured soft, medium, strong grasp, and opening.
- *Input and sliders*: depending on the control modality set and using sliders, it is possible to set the strength (position command to motors) and the stiffness of qbmove building block actuator, which drives the actuated finger.
- *Measurements and Currents*: the two buttons "Get Measurements" and "Get Currents" enable an auto-refreshing thread to respectively show the encoder measurements and the motor current (in milliamperes) in the fields below.

3.2 Tactile Sensing Control Interface

The new version of VS Gripper allows the replacement and customization of the two fingers (Figure 6). This leaves many application possibilities open, including the integration of tactile sensors to increase haptic capabilities useful for the project's tasks.

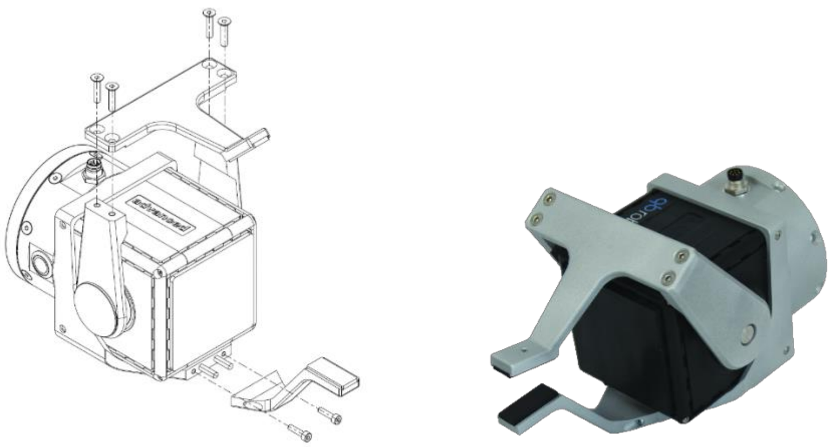

Figure 6 New finger substitution system

3.2.1 Contactile Sensor

Contactile is a Sydney based start-up, founded in 2019 https://contactile.com/.

The sensor that can be integrated in the gripper is the tactile sensor array PapillaArray (shown in Figure 7). Each PapillArray can measure 3D displacement, 3D force, and vibration on each sensing element, as well as global 3D force and torque, slip initiation, and friction. The controller supplies power directly to the sensors.

Supplied hardware:

- Tactile sensor x2 – tactile sensor with 9 PapillaArrays.
- Picoblade 12pin molex cable L600mm x2.
- Tactile sensor controller – control box with micro USB input and 2 Molex PicoBlade 12-pin outputs.
- USB to micro USB cable.

3.2.2 Mechanical Integration

The goal was to integrate a single sensor on the fixed finger of the gripper and position its control box so that it does not constitute an obstacle to the use of the gripper itself.

Figure 7 Mechanical integration of tactile sensor array on VS Gripper

The customizations for mechanical integration are:

- Design and production of customized finger.
- Design and production of a flange for control box.
- New cable of suitable length.

3.2.3 Software Integration

The sensor kit offers an application with an interface (Figure 8) that consists of:

Figure 8 Papillarray Contactile Interface

- Buttons for sensor connection and calibration.
- Visual signals of sensors in contact and center of pressure.
- Numerical values of 3D forces and torques for each individual cell, as well as the result values.
- Graphs with numerical values of 3D forces and torques for each single cell, as well as of the results.
- Algorithm for creep detection.

Contactile sensors give measurements about the force on each sensor in the matrix and the total force/torque acting on the entire surface. The following APIs are provided to use this information and be able to integrate it into the control of the VS Gripper:

- C++ API: they are supplied as pre-compiled libraries. It is possible to modify the source code of the device interface. The API allows reading the force/torque data described above and implements some algorithms, e.g. slip-detection.
- ROS: the same functions described for the C++ API are also available in ROS.

This data can be used to implement gripping strategies, detect slips, and provide force feedback.

3.3 Testing of the Integrated System

The integrated system has been used in some preliminary tests. Videos are available from the following link: https://www.dropbox.com/t/ZDlKBWhG412hTpFd.

The video named BALANCED GRASP shows the system lifting an object with a gripping position not in a vertical line with the center of gravity of the object. By reading the torque (Mz) and weight (Fx) through sensors, the force closure of the gripper is properly settled in order to maintain the grasp.

The video named GRIP SURFACE shows the detection of contact points of the sensor.

The video named DEFORMATION USING DIFFERENT DEFLECTION PRE-SETS shows the different stiffness behaviours of the gripper grasping a soft bal. From the sensor GUI it is possible to see the effects on the exerted force.

4 Soft End-Effectors ROS Control

QBR developed a ROS node to integrate the qb SoftHand and the VS Gripper into the recycling workcell control software. The communication is handled by a single central Node that manages the shared resources for the serial communication (in this case two USB ports) and provides several ROS services through which it is possible to interact with the connected devices (e.g. get info or measurements, activate or deactivate motors, set commands, etc.). This node is called Communication Handler. On startup, it scans the serial communication resources connected and initialize the services. Figure 9 shows the overall architecture.

The control node exploits the ROS control Controller Manager, which loads and runs the device controllers. Each controller provides an Action Server which, together with the Hardware Interface structure, allows the user to send commands to the specific device and get its measurements, exploiting the services provided by the Communication Handler Node. It is implemented as an Action Client that provides a method to send goals, i.e. command references, directly to each device. It is subscribed to a topic (*_controller/command) that can be used to send reference commands from outside the code. We used this ROS topic to send the control references to each device.

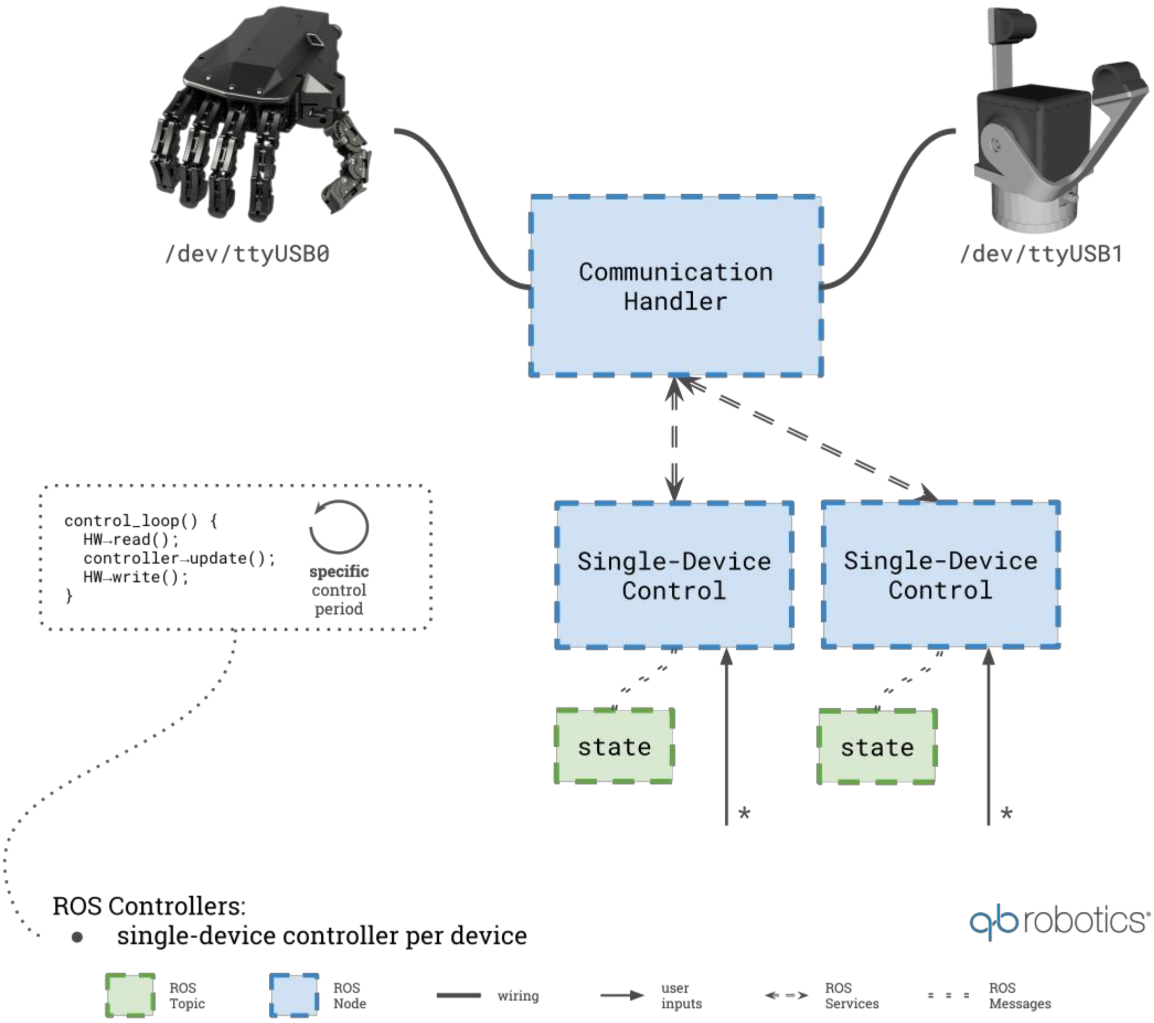

Figure 9 Synchronous and Asynchronous control nodes for VS gripper.

The ROS packages are available in these repositories: https://wiki.ros.org/Robots/qbhand and http://wiki.ros.org/Robots/qbmove.

The following figures (Figure 10) show an example of the pick and place sequence and levering action in ROS, performed by the VS Gripper and Franka Emika robot.

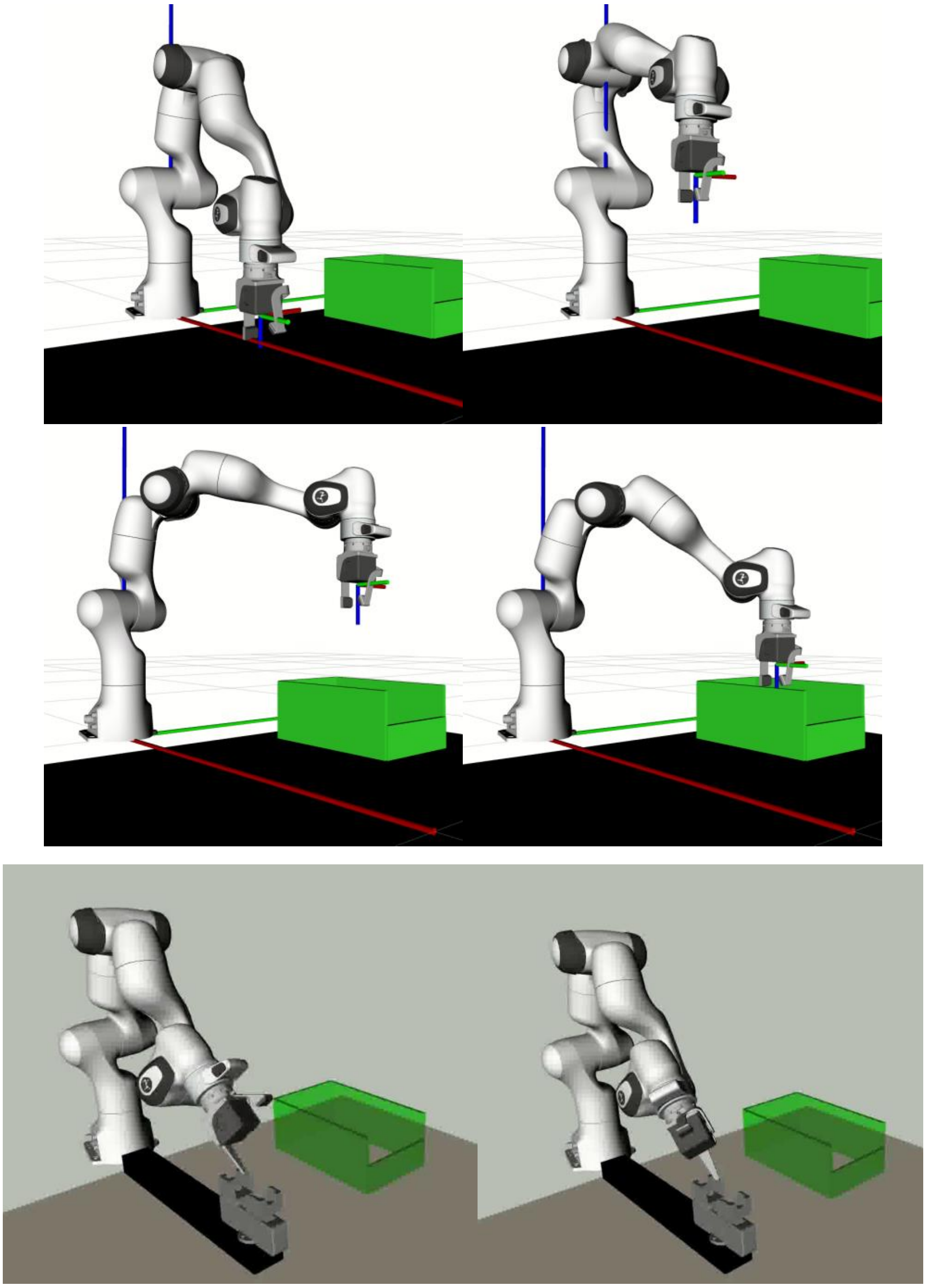

Figure 10 Grasp sequence and levering actions with VS Gripper in ROS

4.1 ROS Control Interface

For ROS environment, GUIs for SoftHand 2 and VS Gripper are available. Figure 11 shows the SoftHand2 GUI, where you can find two sliders to manage the synergies (or the motors, depending on the running controller) of the hand. It is also possible to vary the speed through the bottom speed scaling slider in order to set a faster or slower motion.

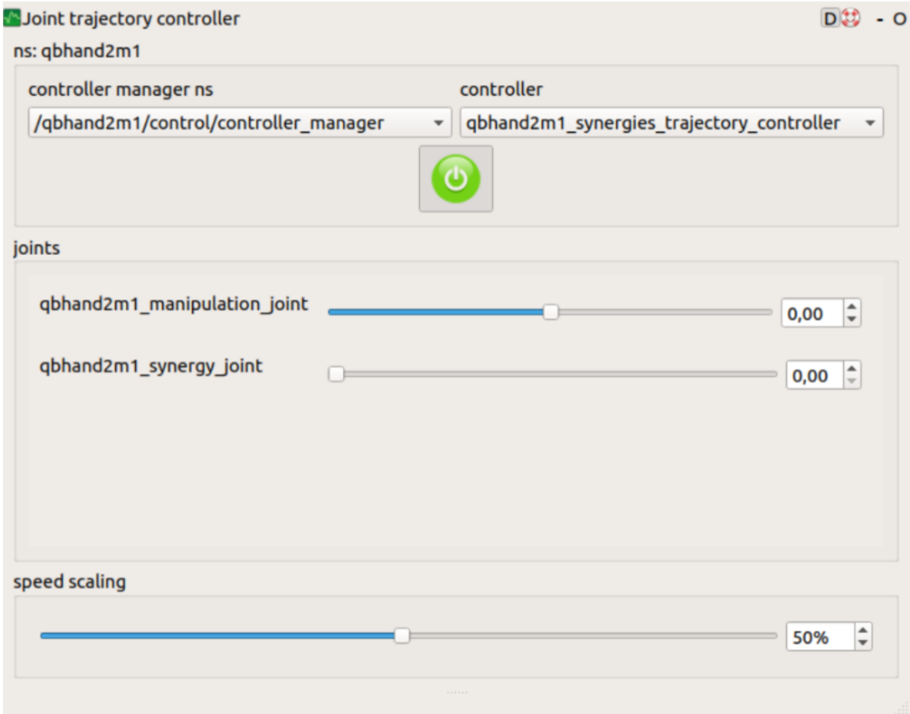

Figure 11 qb SoftHand 2 ROS GUI control

Figure 12 shows the GUI control for VS Gripper. A single slider will appear in the GUI to control the closure of the gripper. It takes both positive values (grip strength) and negative values (to open the finger). It is also possible to vary the speed through the bottom speed scaling slider in order to set a faster or slower motion.

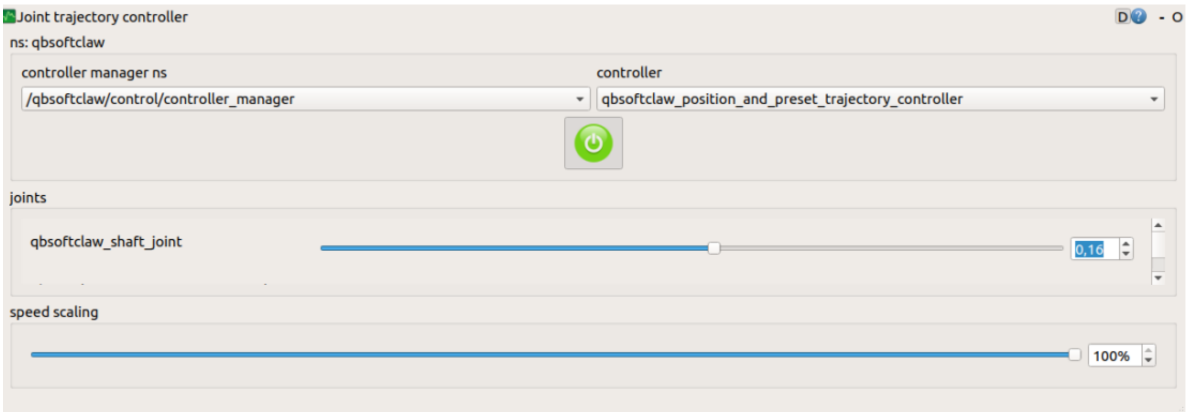

Figure 12 VS Gripper ROS GUI control