REMODEL - Robotic tEchnologies

for the Manipulation of cOmplex

DeformablE Linear objects

Deliverable 5.2
Cable grasping

Version 2021-02-28

Project acronym: REMODEL
Project title: Robotic tEchnologies for the Manipulation of cOmplex DeformablE Linear objects
Grant Agreement No.: 870133
ObjectsTopic: DT-FOF-12-2019
Call Identifier: H2020-NMBP-TR-IND-2018-2020
Type of Action: RIA
Project duration: 48 months
Project start date: 01/11/2019
Work Package: WP5
Lead Beneficiary: UCLV
Authors: All partners
Dissemination level: Public
Contractual delivery date: 28/02/2021
Actual delivery date: 28/02/2021
Project website address: https://remodel-project.eu

Table of contents

List of figures

1 Introduction

This Deliverable aims to describe the strategies and algorithms concerning the wire grasping developed during the first part of REMODEL project, by exploiting the data available from the sensing system. The latter is constituted by vision and tactile data.

The vision system is used to recognize the cables to grasp and to implement an approaching phase through a vision-based control algorithm. After the cable grasping based on vision data, the tactile sensor is used to evaluate the grasp quality, in order to release and re-grasp the cable if the first approaching phase did not allow to obtain the desired grasp. In particular, the tactile map is used to evaluate the position and the orientation of the wire with respect to the end-effector frame and to compare it with a desired pose.

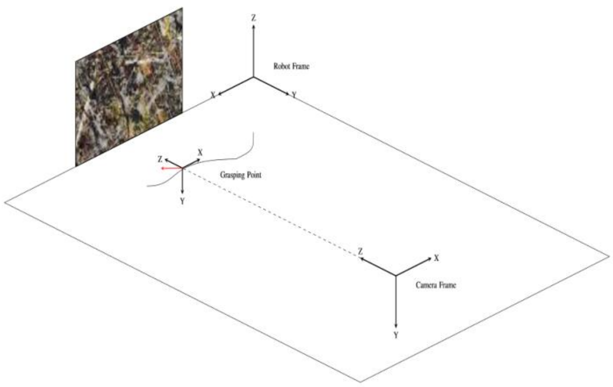
According to Technical Annex at the release date of this Deliverable, the REMODEL project reached the TRL4 level. At this level, the proposed procedures are validated in laboratory with different DLOs. ROS packages have been released for the developed algorithms.

Since the Deliverable expected some demonstrators, a set of videos have been prepared and uploaded to REMODEL website. The links to these videos and their description are reported in the document.

2 Use of vision data

2.1 Cable Detection Algorithm

The computer vision algorithm adopted to detect the cable in the scene is based on ARIADNE[1], a neural network for the recognition of deformable objects in cluttered environments. For the grasp test, a 2D camera is placed at a known distance from the cable, see Figure 1. Moreover, a fiducial marker is placed in a known position on one finger in order to estimate the camera position with respect to the robot: the marker is detected by the camera providing its position in the camera frame, as shown in Figure 2, then the knowledge of the marker position in the world frame thanks to the robot kinematics is exploited to compute the camera position with respect to the robot, see Figure 3. ARIADNE will look for the cable by detecting elements that are characterized by a long and narrow shape. Figure **1** shows how the camera and the grasping pose reference frame are arranged in space. Starting from the input image provided by the camera and reported in Figure **4**, ARIADNE creates a binary mask representing the region of the image in which the cable is present, as shown in Figure **5**. This mask highlights with a white colour the presence of this object with respect to the background represented with the black colour. This image is created for the user to understand the passages and the correctness of the algorithm. Once the mask is created, the wire is approximated with a spline, see Figure 6, parametrized by $s \in [0;1]$, more precisely $s=0$ on the left side of the image while $s=1$ on the right side. From this spline approximation, the grasping point on the cable is defined by selecting two specific values of s, i.e., $s=0.45$ and $s=0.55$. The first point represents the origin of the grasping pose reference frame, while the second is used to define the orientation of the x-axis. The y-axis is selected to be on the image plane normal to the x-axis and the z-axis is orthogonal to the image plane in the direction of the camera.



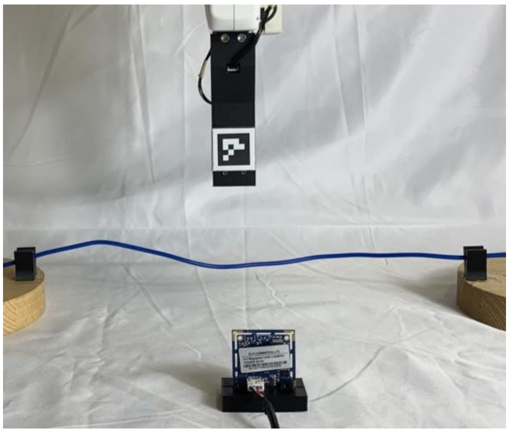


Figure 1: SetUp representation. A 2D camera is placed at a known distance from the wire positioned over two support. Behind the cable there is a Pollock image to show the capability of the net to distinguish an object from the background. It is also possible to see the different reference frames of the set-up.

[1] De Gregorio, D., Palli, G., Di Stefano, L.Let's Take a Walk on Superpixels Graphs: Deformable Linear Objects Segmentation and Model Estimation(2019) Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11362 LNCS, pp. 662-677.

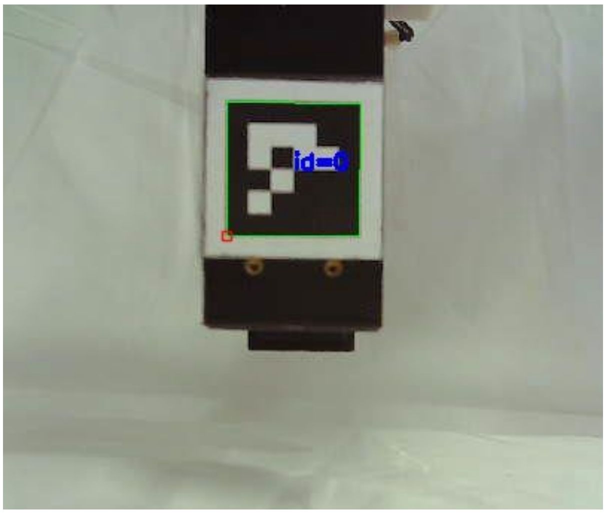


Figure 2: The marker placed on the finger used to detect the camera position.

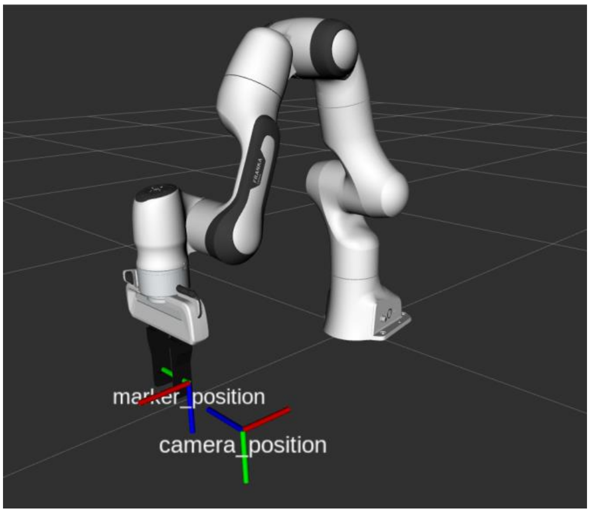


Figure 3: Relative position of the camera with respect to the robot.

The grasp reference frame shown in Figure 7 is then converted in the coordinate system of the end-effector, to be used for the actual grasp. The conversion can be applied knowing the rotation matrix R between the two systems and the relative position vector p.

$$\begin{bmatrix} x_{world} \\ y_{world} \\ z_{world} \\ 1 \end{bmatrix} = \begin{vmatrix} R_{11} & R_{12} & R_{13} & p_x \\ R_{21} & R_{22} & R_{23} & p_y \\ R_{31} & R_{32} & R_{33} & p_z \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{bmatrix} x_{camera} \\ y_{camera} \\ z_{camera} \\ 1 \end{bmatrix} \tag{1}$$

In a possible future development, the distance z can be easily computed by the introduction of a second camera looking at the same scene along a different direction. The process of using two cameras to capture a scene is denominated as stereophotography, using the additional information coming from the second camera and knowing the position of a camera with respect to the first one is possible to obtain the depth of a point, which in this case coincides with the distance of the cable to grasp. The addition of a stereo-view can be useful in situation in which the cable and the camera positions are variable.

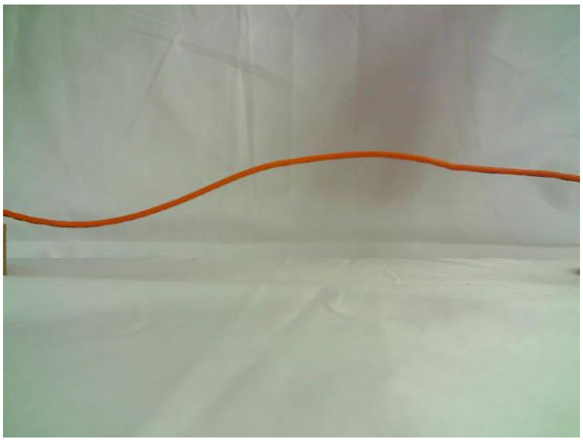


Figure 4: Input image taken from the camera.

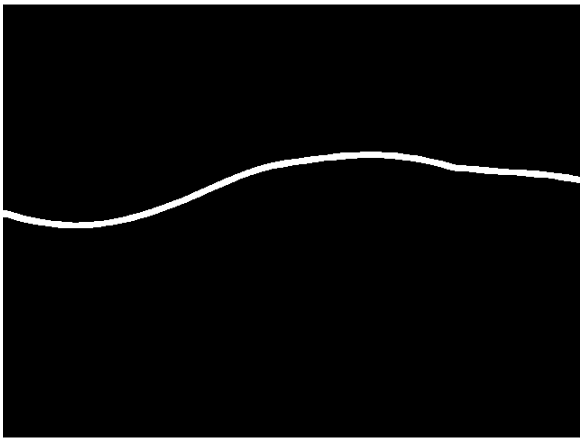


Figure 5: The mask highlights the object in the scene that has a DLO characteristic.

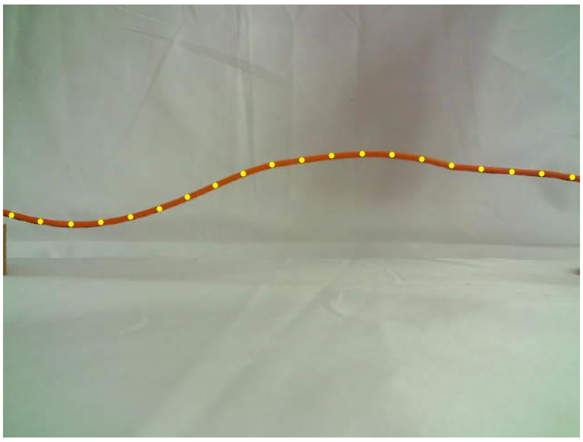


Figure 6: The approximation spline is represented in the image.

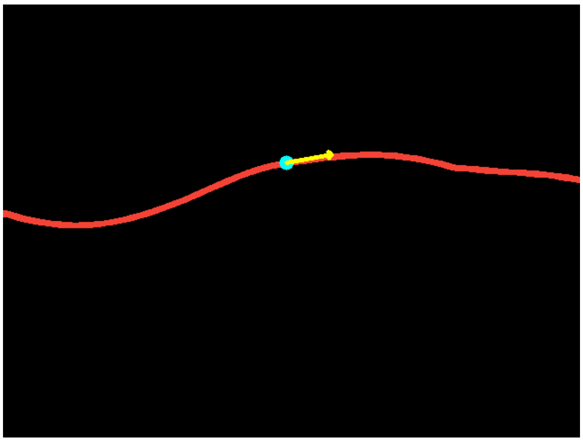


Figure 7: The final point used as reference to grasp the wire.

2.2 Experimental Setup and Results

The experiments are executed by using a 7-dof robotic arm, the Panda from Franka Emika, equipped with a Schunk PG70 parallel fingers electric gripper. The robot is controlled through MoveIt, a ROS plugin specifically designed to provide a high-level interface for trajectory and motion planning. This framework enables to control the robot by passing the desired trajectory to the low-level controller. To control the gripper instead, a ROS-service has been created to set the position of the fingers through an USB port connected with the gripper’s DMI controller. This gripper has been preferred with respect to the Franka Hand because of its characteristic. In particular, the original Panda gripper provides a series of action servers to control the basic movement of the gripper: these commands are blocking instructions and, as a consequence, when a command is executed, the control must wait for its completion before to accept another task. The blocking instruction does not permit a real-time control strategy for the position of the fingers: this is not directly used in these first tests, but it will be used in the future for the dynamic tests. Another important feature of the Schunk PG70 gripper is the presence of electrical brakes embedded in the mechanism of the gripper itself. The brake remains active whenever no command is received and does not permit to the fingers to slide, this means that once the requested position is reached, the fingers will not move anymore until a new command is issued.

The result of a sample grasp is shown in the following figures. In particular, Figure 8 shows the initial configuration of the cable before grasping, while Figure 9 shows the resting tactile map, i.e. in case no cable is grasped. Figure 10 shows the execution of the grasp by the robot and the corresponding tactile map in Figure 11.

Figure 8: Pre grasp – Camera image.

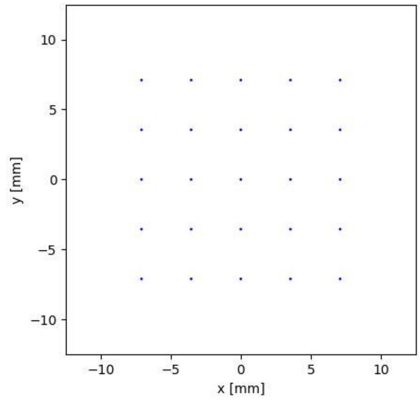


Figure 9: Pre grasp – Tactile map.

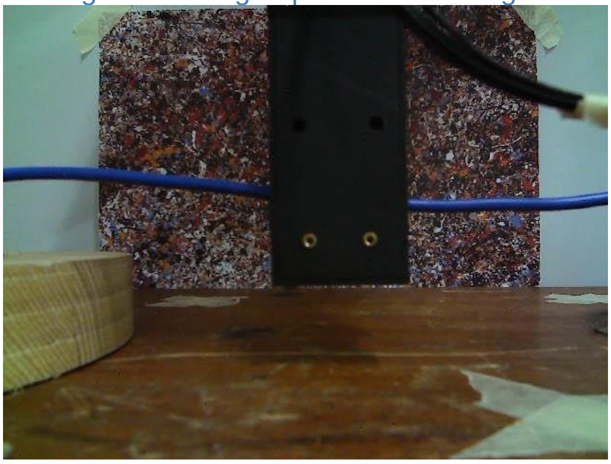


Figure 10: Grasp – Camera image.

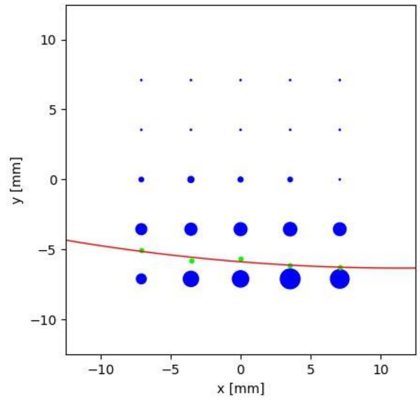


Figure 11: Grasp – Tactile map.

3 Use of tactile sensor data

The tactile sensor can be used to obtain some useful information for the cable grasping and manipulation tasks. For this reason, some ROS nodes have been developed to extract such information from the sensor.

In the following sections, first some technical details about the tactile sensor are given and then the different nodes composing the whole system are presented and explained. The overall scheme of the system is shown in Figure 12, where the ellipses are the ROS nodes and the rectangles represent the ROS topics which the nodes publish or are subscribed to.

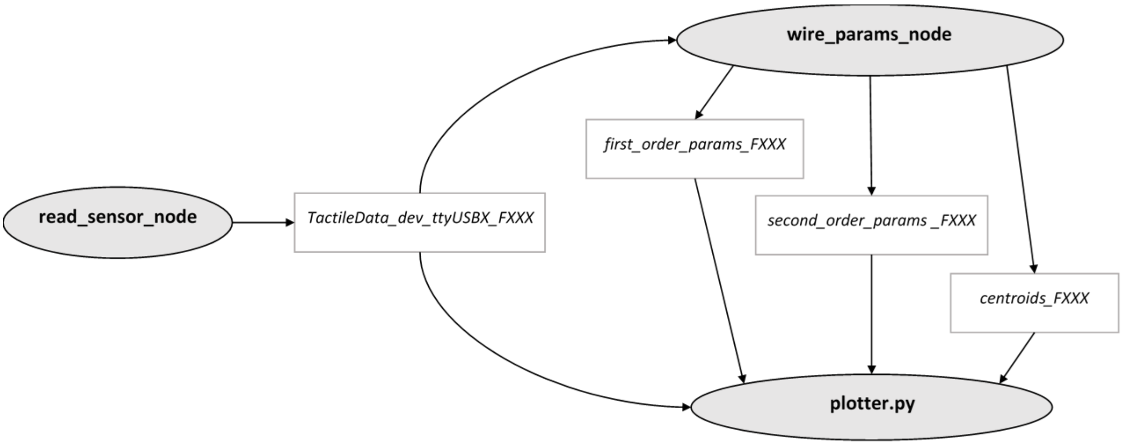


Figure 12: Overall scheme.

3.1 Tactile sensor

The tactile sensor has the shape of a “finger” (see Figure 13), since it is realized to be mounted on a typical commercial gripper.

Figure 13: Tactile sensor.

The sensing area is covered by a silicon layer of 25x25 mm and it consists of 25 photo reflectors positioned with a grid shape having 5 rows and 5 columns. Between two adjoining photo reflectors there is a distance of 3.55mm (see Figure 14). Each photo reflector gives in output a voltage signal depending on the quantity of light hitting the receiver.

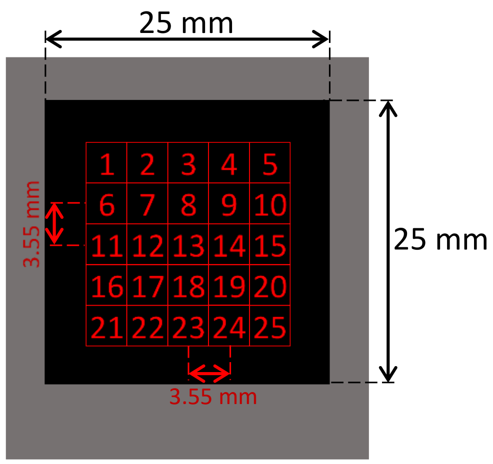


Figure 14: Sensing area.

The sensor needs to be connected to a computer using a USB cable (more tactile sensors can be connected using only one USB cable). In fact, the USB connection is necessary for both giving alimentation to the sensor and transferring data from the sensor to the computer.

Thus, data from the sensor are sent to the serial port of the computer and then the information is received and processed by a ROS node called *read_sensor_node*, which is explained in the next paragraph.

3.2 The ROS node "read_sensor_node"

This node, schematized in Figure 15, is responsible to make the data received from the serial port readable, and it publishes the information on two different ROS topics with a frequency of about 500 Hz ("about" because ROS does not assure that the frequency is strictly respected).

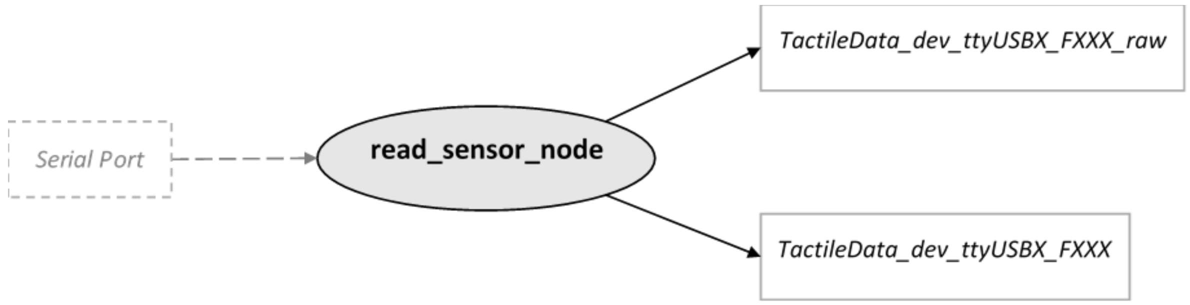


Figure 15: Scheme for read_sensor_node.

The first thing the node does is to scan the devices connected to the serial ports of the computer searching for the compatible ones, i.e., the tactile sensors. If one or more sensors are available, it asks the user to choose which one to use. Once the user makes the choice, the node starts to interrogate the selected serial port, converts received data from digital values to analog values and publishes the information on two ROS topics.

The first topic is named "*TactileData_dev_ttyUSBX_FXXX_raw*", where the X are numbers and depend on the ID of the serial port and the ID of the connected sensor. On this topic, data coming from the sensor are published without being processed, so it contains voltage values depending on how the light emitted by the photo reflector is reflected by the silicon layer. Since the silicone is molded manually, this value is not the same for each cell.

The second topic is named "*TactileData_dev_ttyUSBX_FXXX*" and it contains the same values published on the first topic after the removal of the "offset" for each cell, computed as the mean value of the voltage during the first 50 samples.

The message in these topics is a custom message called "*tactile_sensor_data*" which is formed by a Header, containing the timestamp, and a vector of Float32. The vector has 25 elements which correspond to the voltage values of the cells numbered as shown in Figure 14.

The code of this ROS node is available on GitLab, for all the partners, at the following link: https://dei-gitlab.dei.unibo.it/palli_group/sensorreadingrosapp

3.3 The ROS node "wire_params_node"

The purpose of this ROS node, schematized in Figure 16, is to use the information published by *read_sensor_node*, hence data from the tactile sensor, to estimate the pose of a DLO (in this case a wire) in contact with the sensing area of the sensor.

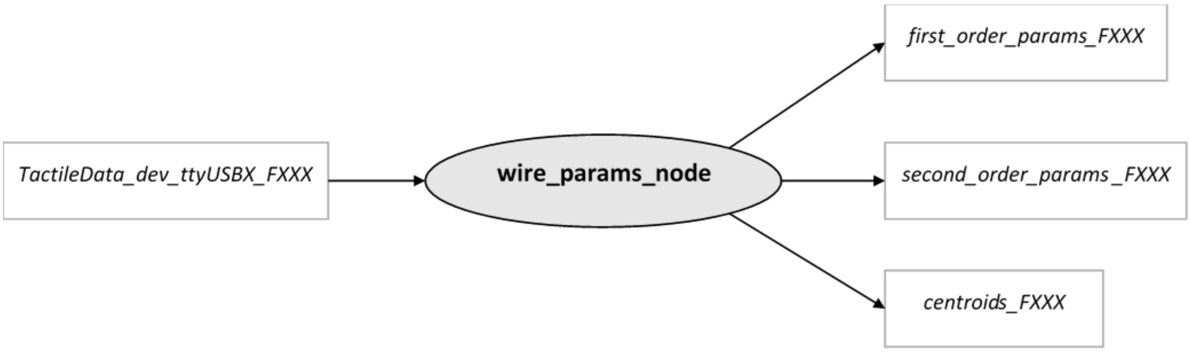


Figure 16: Scheme for *wire_params_node.*

The reference system used for the approximations is shown in Figure 17.

In the "*first_order_params_FXXX*" topic, the node publishes the parameters of a first order approximation of the wire shape (a straight line). Hence, the wire is approximated as:

$$y = mx + n \tag{2}$$

where m and n are the two parameters published on the topic.

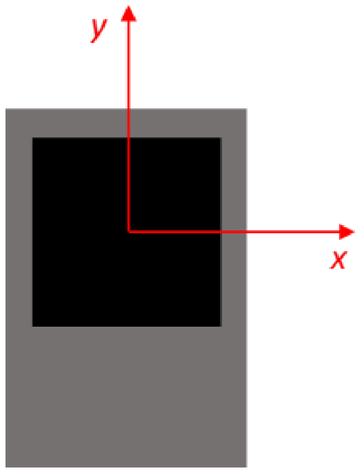


Figure 17: Reference system.

In the "*second_order_params_FXXX*" topic, instead, the node publishes the parameters of a second order approximation of the wire shape (a parabola). Hence, the wire is approximated as:

$$y = ax^2 + bx + c \quad \text{vertical axis of symmetry} \tag{3}$$

$$x = ay^2 + by + c \quad \text{horizontal axis of symmetry} \tag{4}$$

where a, b and c are the three parameters published on the topic.

To obtain these approximations, the node uses the voltage values received from the *read_sensor_node* and the mechanical coordinates of the 25 cells. The proposed algorithm (pseudocode in Algorithm 1) can be divided in three steps:

- Detection of the main direction of the grasped wire (horizontal or vertical)
- Computation of the centroid coordinates for each column (if horizontal) or for each row (if vertical) using the tactile data
- Computation of the parameters for the two approximations using the centroid coordinates

Algorithm 1 - Estimation of wire shape

Input: cell coordinates (x_i, y_j)
voltage variations Δv_{ij}
Output: parameters for (2)-(3)-(4)
 	
compute quantities $h,\ v$ from (5)-(6)
if $h > v$ **then**
compute column centroids from (7)
else
compute row centroids from (8)
end
compute a, b, c and m, n via a least square method

The main direction of the grasped wire is obtained by computing and comparing the two following quantities:

$$h = \min\left\{\sum_{i=1}^{5} \Delta v_{i1}, \sum_{i=1}^{5} \Delta v_{i2}, \sum_{i=1}^{5} \Delta v_{i3}, \sum_{i=1}^{5} \Delta v_{i4}, \sum_{i=1}^{5} \Delta v_{i5}\right\} \tag{5}$$

$$v = \min\left\{\sum_{j=1}^{5} \Delta v_{1j}, \sum_{j=1}^{5} \Delta v_{2j}, \sum_{j=1}^{5} \Delta v_{3j}, \sum_{j=1}^{5} \Delta v_{4j}, \sum_{j=1}^{5} \Delta v_{5j}\right\} \tag{6}$$

where Δv_{ij} are the voltage values with the offset removed and the cells (1,1) and (5,5) correspond to the cells 1 and 25 in Figure 14, respectively.

If $h > v$ the main direction is aligned with the x-axis, i.e., the wire is horizontal, while if $h < v$ the main direction is aligned with the y-axis and so the wire is vertical.

For the second step, if the main direction is horizontal, the y-coordinates y_j^c of the column centroids are computed as

$$y_j^c = \frac{\sum_{i=1}^{5} y_i \Delta v_{ij}}{\sum_{i=1}^{5} \Delta v_{ij}} \qquad j = 1, \dots, 5 \tag{7}$$

while, if the main direction is vertical, the x-coordinates x_i^c of the row centroids are computed as

$$x_i^c = \frac{\sum_{j=1}^{5} x_j \Delta v_{ij}}{\sum_{j=1}^{5} \Delta v_{ij}} \qquad i = 1, \dots, 5 \tag{8}$$

where x_j and y_i are the mechanical x-coordinate of the jth column and the mechanical y-coordinate of the ith row respecively.

The last step consists in estimating the parameters in the (2) and (3)-(4) by using a least squares method applied to the data obtained with the (7)-(8).

The topic named "*centroids_FXXX*" contains the positions of the centroids computed with (7)-(8) and the information on the wire main direction (vertical or horizontal). This topic is used for graphical representation by the node explained in the next section.

The code of this ROS node is available on GitLab, for all the partners, at the following link: https://dei-gitlab.dei.unibo.it/palli_group/wireestimationstatic

3.4 The ROS node "plotter.py"

This python script is a ROS node that reads from the topics published by the nodes explained above and represents the received information in a graphical way. It is available in the same package of the *wire_params_node*.

Once started, the node checks if there is at least one *read_sensor_node* publishing and if more than one sensor is connected, it asks the user to choose one of them. At this point, if for the selected sensor the topics related to the shape approximations are not published (*wire_params_node* is not running), the node shows the tactile map directly, otherwise it asks what type of information to show among three options: tactile map only, tactile map with straight-line approximation or tactile map with parabola approximation. The tactile map consists in a grid of dots whose size depends on the voltage value of the corresponding cell, the shape approximation is represented by using a red line and the positions of the centroids are indicated by green dots.

Two examples of the output of this node are given in Figure 18, where the image on the left shows the representation of a first order approximation of the wire while the image on the right shows a second order approximation.

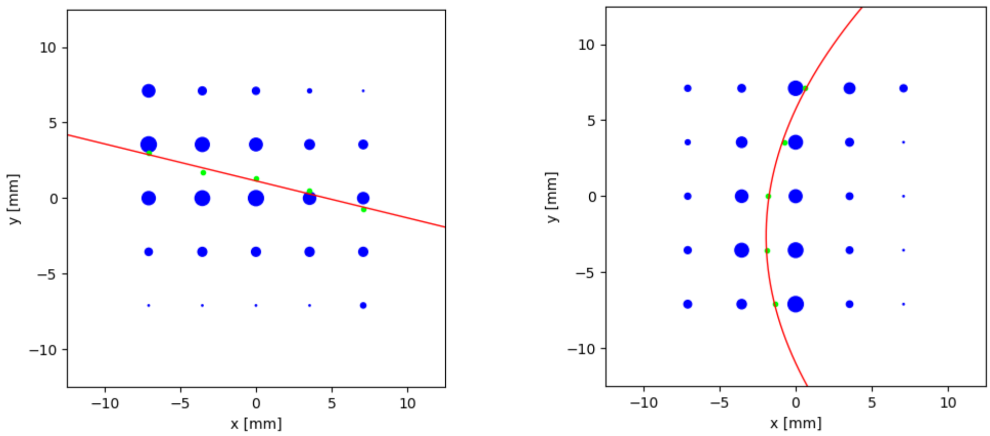


Figure 18: Graphical representation examples.

3.5 Shape reconstruction results

To test the quality of the computed shape approximations, the tactile sensor has been pressed using wires having different diameters and the scene has been recorded using an external camera. Thus, the graphical representation of the approximation and the image of the real wire have been compared.

Some frames from these comparisons are reported in Figure 19, Figure 20, Figure 21 and Figure 22 as examples.

3.6 Grasping correction experiment

This section shows an experiment to demonstrate how the information obtained by using the ROS nodes explained previously can be exploited. The objective is to use the parameters given by the linear approximation of the wire, eq. (2), to grasp it in a desired manner, i.e., with a certain position and orientation with respect to the sensorized finger.

The setup for this experiment is constituted by the tactile sensors mounted on a Schunk PG70 gripper and the robotic arm is the PANDA. The cable has a diameter of 2.5 mm and it is suspended between two supports to avoid eventual collisions during the grasping task, see Figure 10.

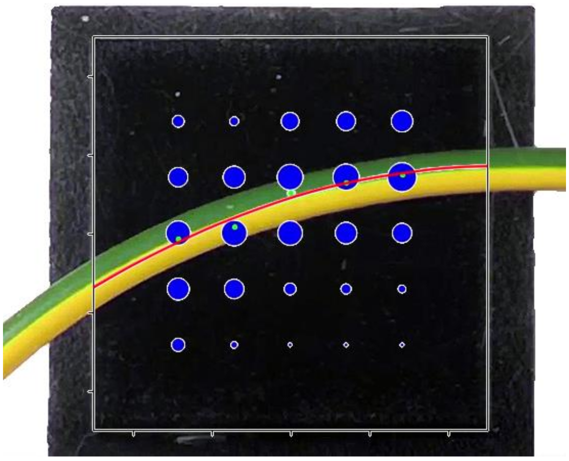


Figure 19: Wire 1 (2.5 mm).

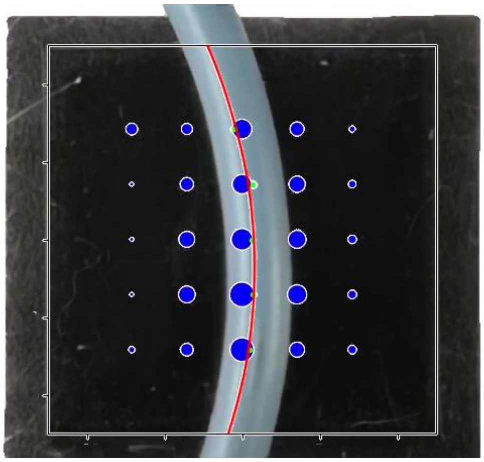


Figure 20: Wire 2 (4 mm).

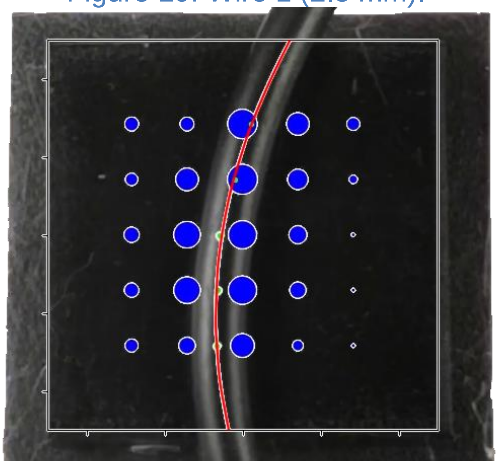


Figure 21: Wire 3 (3.5 mm).

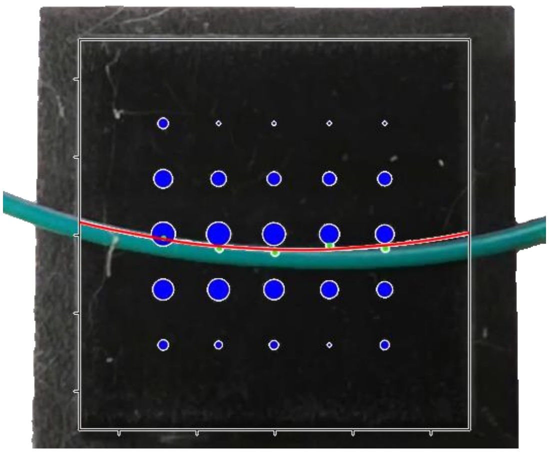


Figure 22: Wire 4 (1 mm).

The experiment consists in grasping the cable two consecutive times: the first time is a “trial” grasp and the second one is the final grasp, after the correction.

For the first trial, the cable location can be considered known or it can be obtained, for example, using the algorithm explained in Section 2.1. In both cases, the cable location may not be the real one for various reasons, e.g., the deformability of the cable, partial occlusions of the vision systems, errors in the cable detection algorithm and others. Thus, once the fingers are closed over the cable, the linear approximation obtained by the “wire_params_node” in Section 3.3 is used to compute the new robot pose for the final grasp. In particular, the desired cable pose for this experiment is the one with the cable passing through the center of the sensing area of the tactile fingers and aligned with the x-axis of the reference system in Figure 17. In terms of the parameters m and n in the equation (2), where m is angular coefficient of the straight line and n is its distance from the center of the sensor, the desired cable pose results in having both parameters equal to zero (see Figure 23).

Figure 24 shows the two grasps occurring during the experiment: the image on the left shows the “trial” grasp while the one on the right shows the final grasp, with the cable in the desired pose with respect to the fingers.

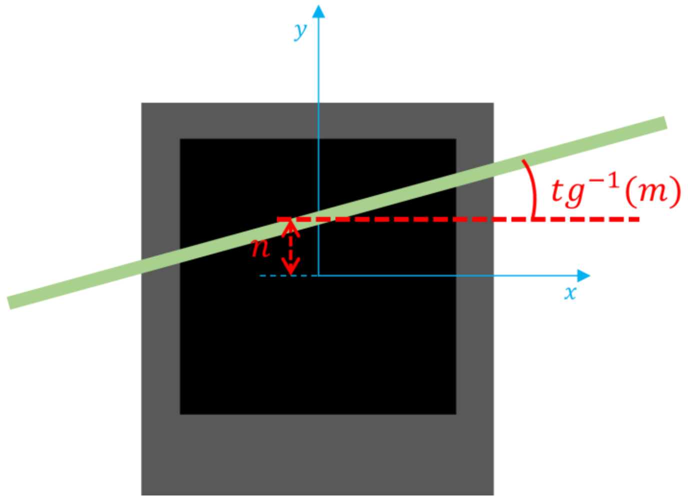


Figure 23: Parameters of linear approximation.

Figure 25 reports the results of 20 repetitions of the experiment, where the initial grasp has been done considering the actual cable location and adding small random offsets to it. From the charts it can be noticed that, in terms of position, the mean error and standard deviation at the first grasp are 4.0 mm and 2.2 mm respectively, while they are reduced to 1.9 mm and 1.5 mm respectively at the second grasp. In terms of angular error and standard deviation, instead, they are 0.32 rad and 0.14 rad respectively during the first grasp and boil down to 0.11 rad and 0.07 rad respectively at the second grasp.

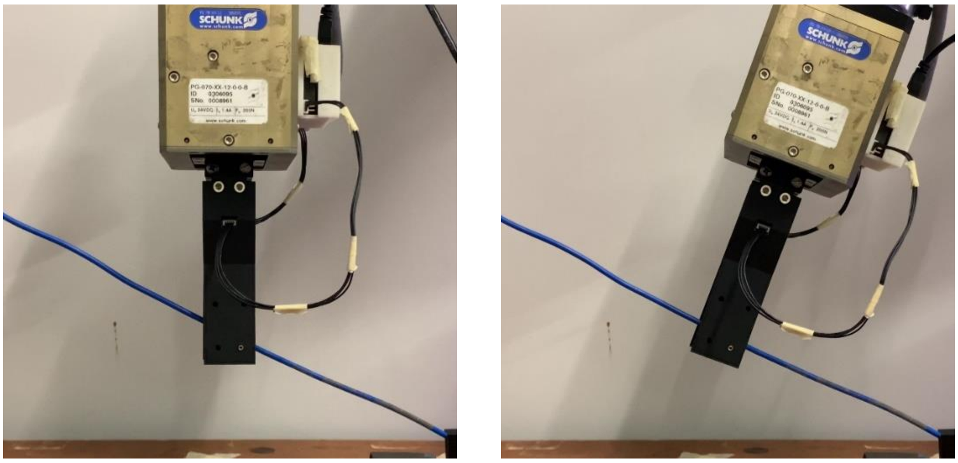


Figure 24: Grasping correction: trial grasp on the left; final grasp on the right.

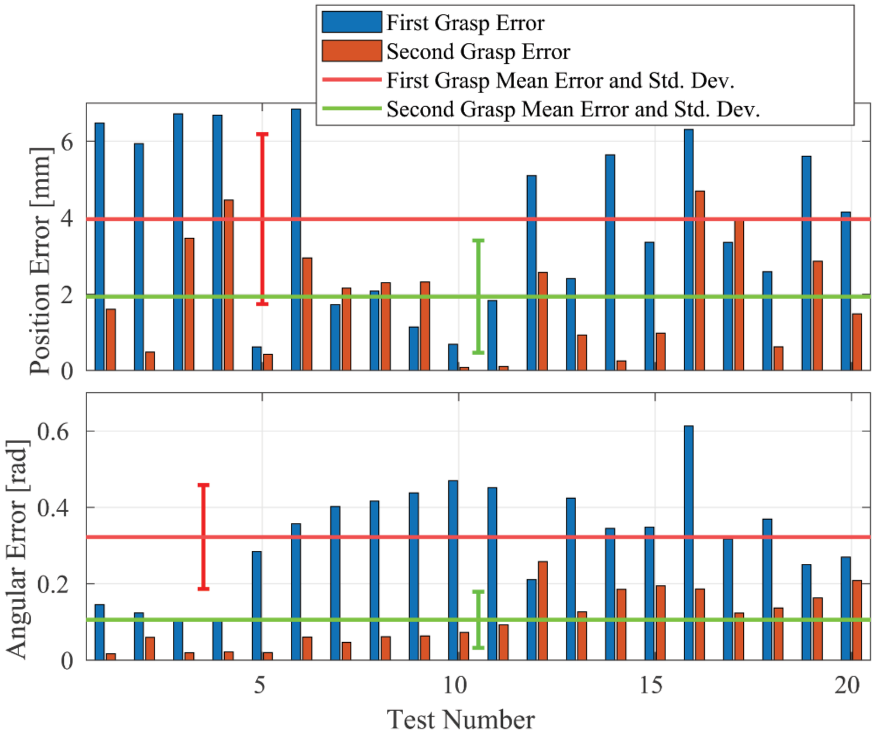


Figure 25: Errors before (blue) and after (orange) correction.

4 Conclusions

This document presented the approach proposed by REMODEL concerning the grasp of DLOs. As shown the proposed solution exploits both vision data coming from a standard 2D camera and tactile data available from tactile sensors suitably developed in REMODEL (WP6 activities). The images are elaborated by using a software package (ARIADNE) developed by UNIBO.

The vision data are used to recognize the wire position in order to define a frame used for the DLO initial grasp. After the initial grasp, the tactile data are used to evaluate the grasp quality by estimating the shape and the positioning of the wire with respect to a known reference frame.

The same data can be used to re-grasp the wire if the initial grasping pose is not satisfactory, by reaching a desired position and orientation for the grasped object. To demonstrate the effectiveness of the proposed method several videos have been prepared and attached to this document.

The first video[2] shows the localization of the cable by the vision sensor through the Ariadne software package. The vision sensor provides the grasp pose of the cable to the robot that then executes the grasp. The system has been tested with cables of different sizes and colours and with different backgrounds and light conditions.

The second video[3] shows side by side the tactile signals with the data related to their post processing with the estimated wire shape and the actual wire used during the experiments. Different diameters have been used.

In the third video[4], the same data reported in the second one are superimposed in order to allow the reader an immediate comparison between the estimated shape and ground truth.

The last video[5] reports an experiment of re-grasp on the basis of tactile data. After a first grasp where the wire shape is estimated, the wire is re-grasped in order to horizontally align the wire with respect to the tactile sensor frame.

[2] https://remodel-project.eu/sites/remodel.drupal.pulsartecnalia.com/files/ariadne_grasp.mp4

[3] https://remodel-project.eu/sites/remodel.drupal.pulsartecnalia.com/files/completo_final.mp4

[4] https://remodel-project.eu/sites/remodel.drupal.pulsartecnalia.com/files/completo_sovrapp_final.mp4

[5] https://remodel-project.eu/sites/remodel.drupal.pulsartecnalia.com/files/ReGrasp_crop_short.mp4