REMODEL - Robotic tEchnologies

for the Manipulation of cOmplex

DeformablE Linear objects

Deliverable 5.4 – Wiring harness manipulation

Version 2022-10-14

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: UNIBO
Authors: UNIBO, TUM, PUT, TAU
Dissemination level: Public
Contractual delivery date: 31/10/2022
Actual delivery date: 31/10/2022
Project website address: https://remodel-project.eu

1 Scope

The manipulation strategy developed to manage complex wiring harnesses composed by multiple cables and different branches is described in this document and shown in the linked videos. In the final implementation of the use case demonstrators, the manipulation strategy receives the data about the wiring harness morphology by the CAD interface developed in T3.1 and will exploit both the wire tracking ability of T4.2 and the wire manipulation strategy of T5.3 to deal with the complexity of the task. Dual arm manipulation is exploited to hold, group or to separate properly the branch as desired by the manufacturing task.

Laboratory evaluation of the wiring harness disentanglement and arrange/route of the branches along the desired path considering obstacles is shown in a linked video as laboratory scale evaluation of the developed technology. Manipulation of single wiring harness branches for the arrangement on supports is also evaluated. Bimanual manipulation will be used to hold or separate entangled branches.

The separation of entangled branches and their route along the desired path is based on the perception system and the testing is performed considering the wiring harness provided by ELVEZ and VWP.

2 Topological representation of the wiring harness

The manipulation and perception pipeline in UC3 rely on VW wiring harness. Due to fact that the type of wiring harness will not be changed during entire process, a general graph representation of such a harness has been developed. The specific harness elements hierarchy was formed considering all the key objects from perception point of view.

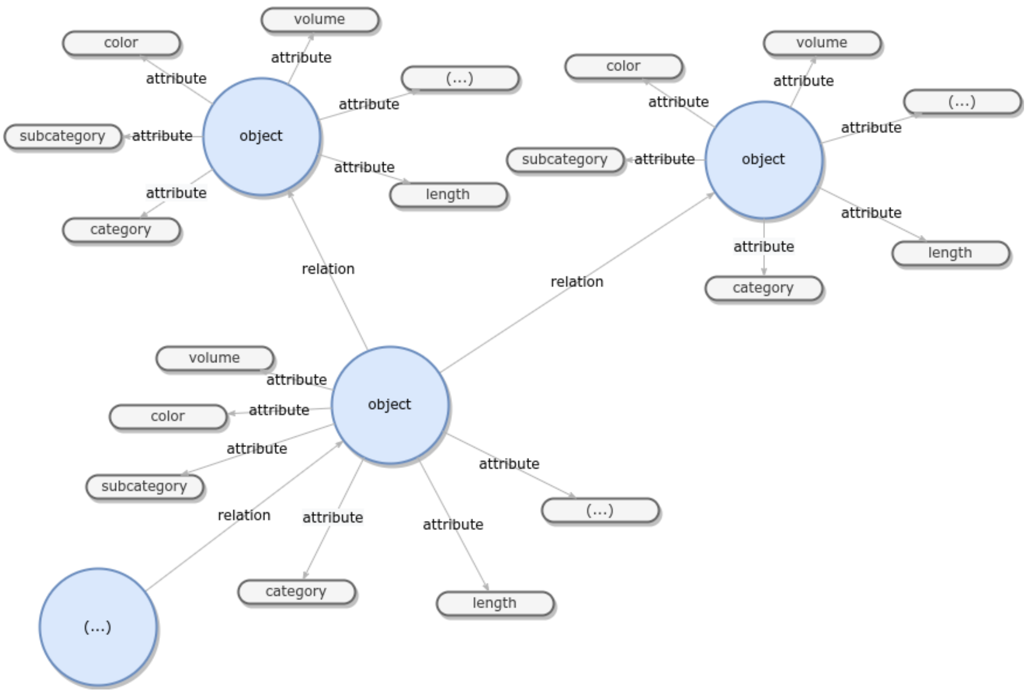


Figure 1: Concept of graph representation for VW wiring harness

The entire wiring harness consist of:

- branches
- bags
- connectors
- miscellaneous (plastic elements, root, box)

That hierarchy allows distinguishing subcategories with similar features by relationships between all the objected detected on scene. Moreover, apart from boosting perception pipeline, graph representation may include final placements of key objects with reference to the base link.

3 Wiring Harness Perception

3.1 UC2.2 (ELVEZ)

The system used for the harness manipulation in the ELVEZ use case is equipped with both vision and proximity perception.

3.1.1 Proximity-based perception

The proximity-based perception relies on the proximity sensor presented and detailed in the deliverable D6.2. The interface board used in the setup is the second version, the one with the MCU on board, and only one sensor module at the top of the finger is mounted (see Figure 2 - left). In addition, the 3D-printed case used to mount the proximity sensor on the back part of the fingers includes fingernails exploited for the cable separation task (see Figure 2 - right).

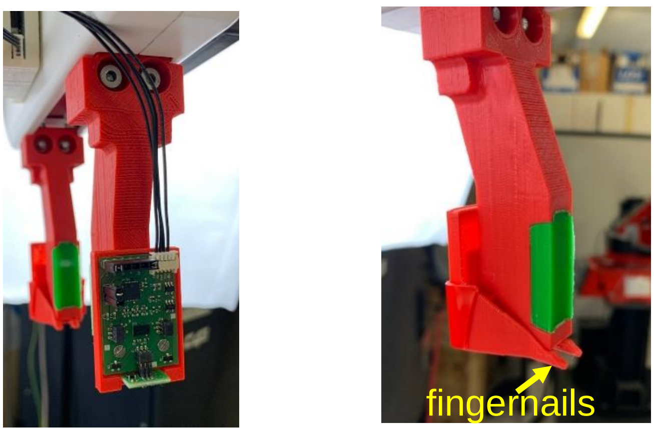


Figure 2: Proximity sensor mounted on the finger and detail of the fingernails

3.1.2 Tactile-based perception

The tactile-based perception relies on the tactile sensors presented and detailed in the deliverable D6.2. The same setup shown in Figure 2 has been realized for the tactile fingers, with the proximity sensor mounted on the back part of one finger and fingernails for the cable separation task (see Figure 3).

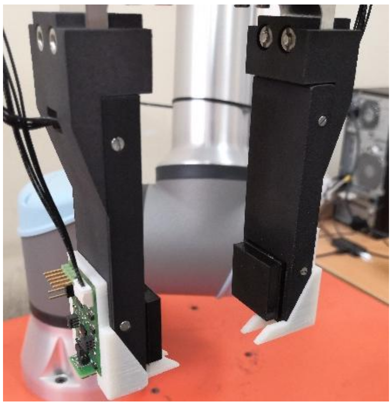


Figure 3 - Setup with tactile sensors

3.1.3 Vision-based perception

A perception algorithm has been developed for identifying and estimating the shape of the different wires composing the wiring harnesses of ELVEZ in the scene. Then, the output of this algorithm (i.e., the polynomial fit of every wire) is used for two applications: the determination of the most appropriate grasping point to separate a specific group of wires, and the evaluation of the cables' separation.

The perception algorithm is composed of four modules as can be seen in Figure 4. The first module is image preprocessing, where the scene and platform information provided by the CAD Platform is used to determine the position of the connector, as the position of the connector holder (see Section 4.1) is known, and to adjust the image resolution based on the number of pixels per cable diameter that the user wants to use. The higher this parameter is, the higher the accuracy, but also the longer the computation time.

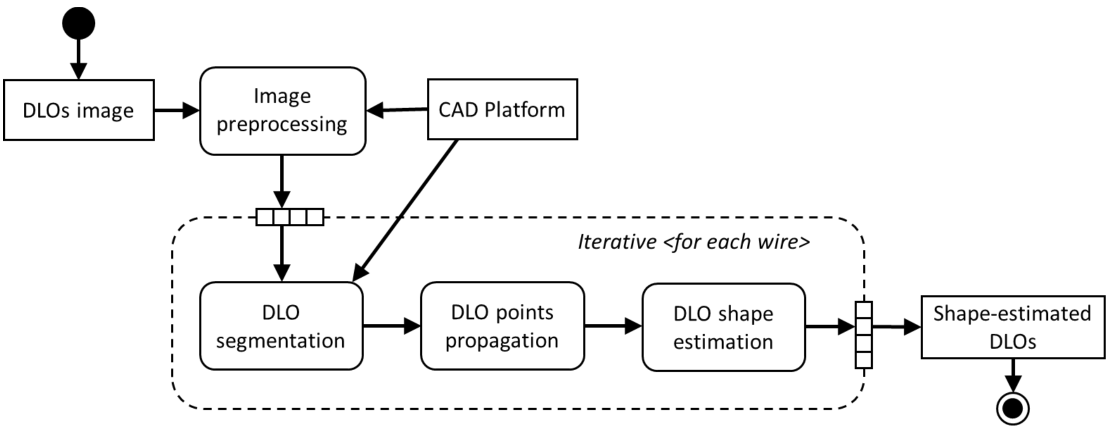


Figure 4. UML activity diagram of the vision-based perception algorithm used in UC2.2.

Then, as the cables of the ELVEZ wiring harnesses are adjacent and have a lot of entanglements, each wire is computed individually by the next three modules, reducing the noise of the rest of the cables. First, a segmentation of the wire in question is performed. This is done by combining a Canny edge detector, a color filter, and further analysis to differentiate cables from borders (see Figure 5).

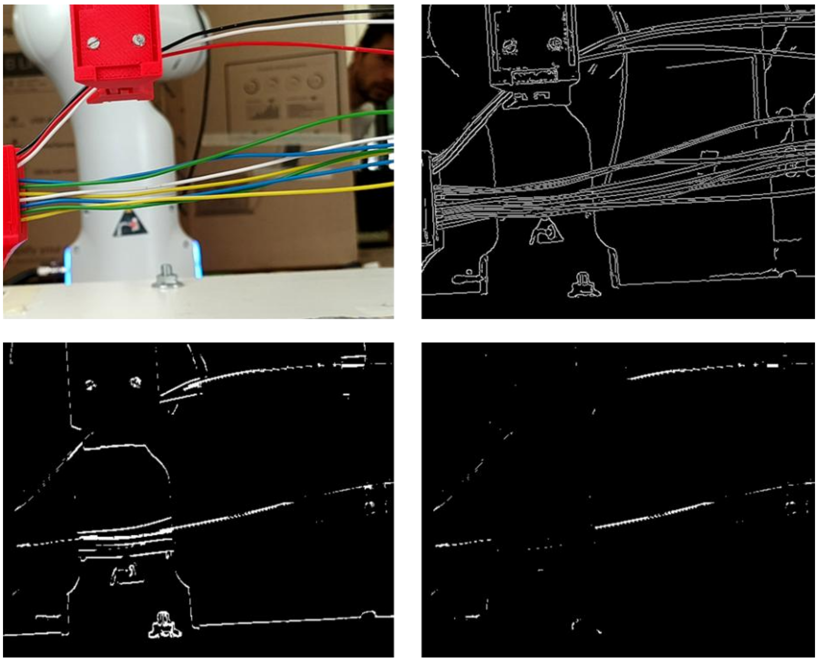


Figure 5. White wires segmentation steps: original image (top left), canny edge detector (top right) + color filter (bottom left) + cable/border classifier (bottom right)

After this, the cable points are ‘propagated’ in the segmented image, starting from the end of the image, until the connector position. This module evaluates all the segmented pixels and finds all the possible cable shapes. Then, all the cable candidates are evaluated based on different metrics and the most likely one is selected. Finally, the cable points sequence is used by the last module, to fit a polynomial that represents the cable shape. The order of this polynomial is determined by cross-validation. Figure 6 shows all the steps of the described process for the shape estimation of the top white wire, and finally the estimated shape of all the wires of the scene. The figure also shows the robustness of the system against occlusions (by the fingers), cable entanglements, and complex backgrounds.

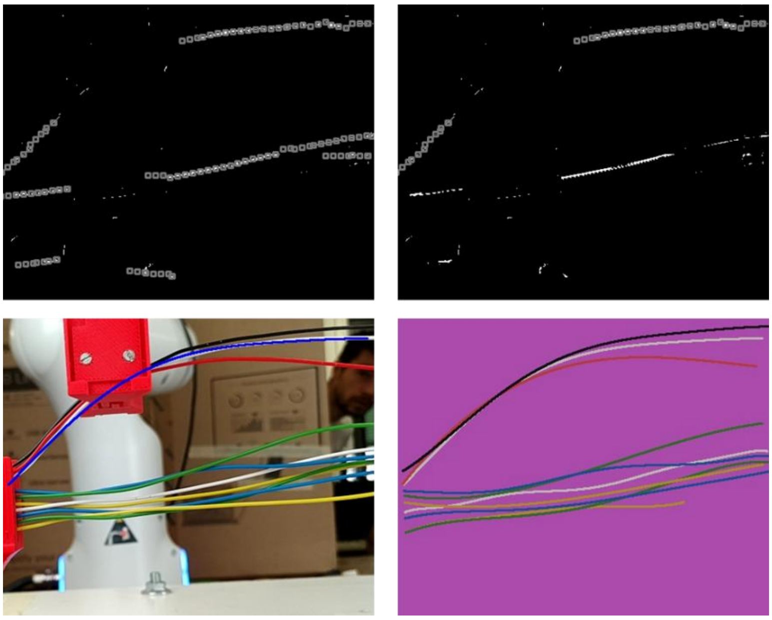


Figure 6. Top white wire shape estimation steps, given the segmentation of Figure 5: points propagation for all the cable candidates (top left), points propagation for the most likely cable (top right), cable polynomial fitting with a blue line (bottom left), estimated shape of all the cables in the scene (bottom right)

As specified at the beginning of this section, the output of this perception algorithm is used for two applications, as can be seen in the description of the experiments in Section 4.1.1. The first one is the determination of the grasp point. This is done by determining the coordinate at which the distance between the lower wire of the upper cable group and the higher wire of the lower cable group is maximum, given the polynomial fitting of every wire. Then, the selected image pixel has to be converted into a real pose for the robot. This conversion is performed based on the mm/pixel value in the plane of the connector holder, the location of the connector holder in the image (pixels), and its location in the real setup (mm), which is provided by the CAD Platform (T3.1). As the approach is implemented with a 2D camera, the depth coordinate is assumed to be the same as for the connector holder. This pose is then sent to the robot in a topic, to perform the wiring harness separation. Figure 7 shows an example of grasping point determination indicated with a green box.

Figure 7. Grasping point determination to separate the black and red cables

The second application is the cables' separation evaluation. Given the polynomial fitting of every wire, and knowing the current position of the gripper, the algorithm checks if the modeled cables pass through the grasping area or not. The information sent to the ROS system is the index of the grasped wires and the success or not of the cable separation operation, to either continue with the cables' routing or start a corrective action. Figure 8 shows some examples of correct and incorrect cable separations and their corresponding evaluation.

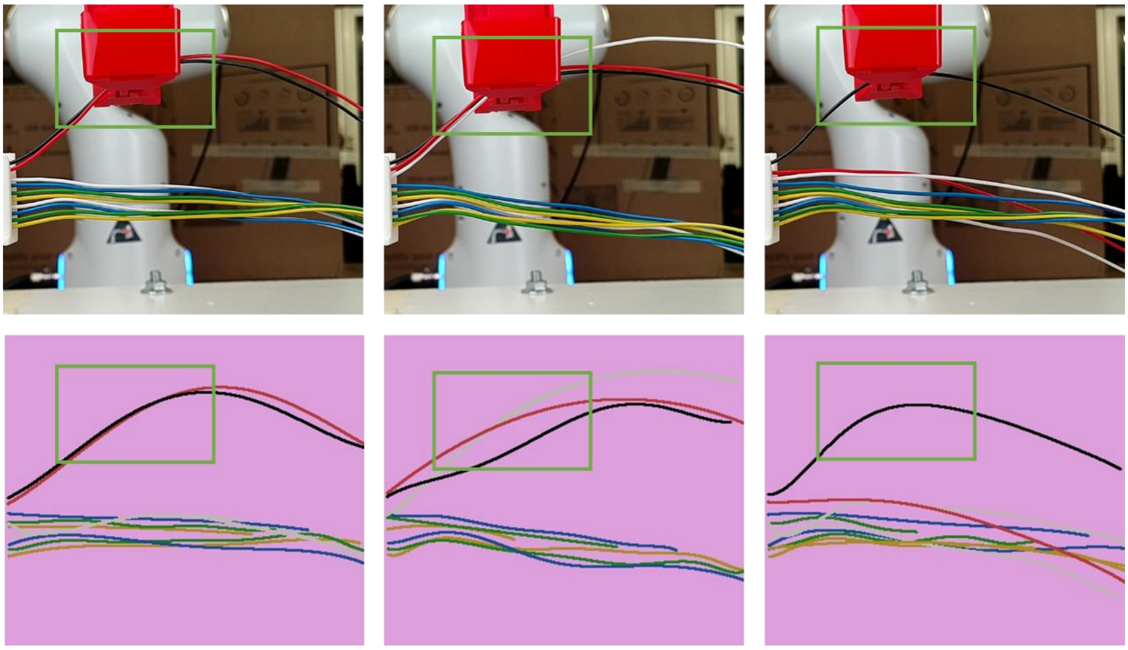


Figure 8. Cable separation evaluation. The robot must grasp just the black and red cables and the grasping area is indicated with a green rectangle. Correct grasping (left), incorrect grasping: one extra cable (middle), incorrect grasping: one cable missing (right)

3.2 UC3 (VWP)

A perception algorithm has been developed for detecting the wiring harness of VWP in the scene and identifying relevant key-points along the object for the further manipulation.

A pointcloud of the scene is captured using the fixed 3D camera (see Sec. 4.1). From the cloud, the plane is segmented employing the RANSAC (RANdom SAmple Consensus) algorithm. Then, a depth image of the scene is constructed by projecting the points of the cloud in the image plane assigning as depth value the distance from the segmented plane. As results, a mask image can be extracted from the depth image by simply considering as foreground (white) the pixels having a depth value greater than zero, and as background (black) the pixels with a depth value of zero. These processing steps from the source pointcloud to the depth and mask images are depicted in Fig. 3 for a sample wiring harness.

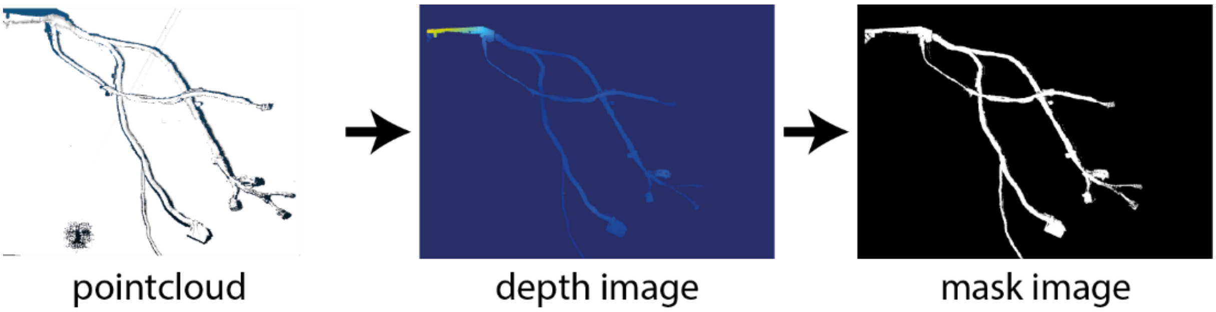


Figure 9: Pointcloud processing obtaining depth and mask images

Given the mask image, a graph representation describing the configuration of the wiring harness is computed following the method already exploited for the Deliverable 4.3 Cable realtime tracking. In particular, the approaches detailed in Sec. 2.2 *Nodes* and *Edges* of D4.3 are exploited for obtaining the graph nodes and edges.

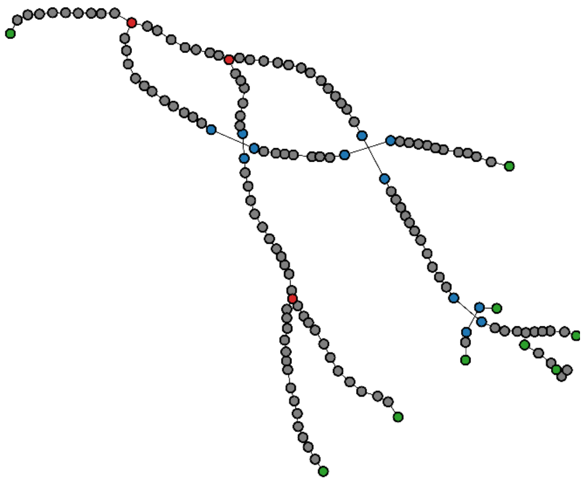


Figure 10: Graph generated from the mask image of Figure 9.

The obtained graph representation is then used for extracting relevant key-points on the wiring harness. These points denote important areas of the graph that can be exploited for assessing a manipulation strategy. Based on the number of edges (i.e., node degree) of each graph's node, it can be classified as one of the following:

- End-point: node with 1 edge;
- Segment-point: node with 2 edges;
- Branch-point: node with 3 edges;
- Intersection-point: node with more than 3 edges;

The intersection-point are solved aiming at disentangling the graph and thus the wiring harness branches. By collecting the neighbours' nodes of an intersection point and by evaluating the overall curvature (as in D4.3 Sec. 2.2) and depth for each candidate neighbour pair, the correct edges between neighbours' nodes can be inserted. These edges are obtained by evaluating, as mentioned, each node curvature and depth value, and exploiting the cosine similarly computation for assessing the best matches, see D4.3 Edges.

Thus, the graph edges are updated removing the original intersection-point node. In Fig. 4 is depicted the graph generated with the mentioned approach for the mask provided in Fig. 3. Notice that Fig. 4 shows the result of the nodes classification with the intersection-point cleaning already performed. The class of each node is denoted with distinct colours, in particular: grey (segment-point), green (end-point), red (branch-point) and blue (neighbours of intersection-point).

The mentioned classification allows also to label the different sections of the wiring harness. With *section,* the sequence of nodes between two particular key-points (either branch-point or end-point) is denoted. In Fig. 5 the same graph of Fig. 4 but with the *section* labelled by unique colours is depicted. This section-based view of the graph is exploited during the manipulation actions.

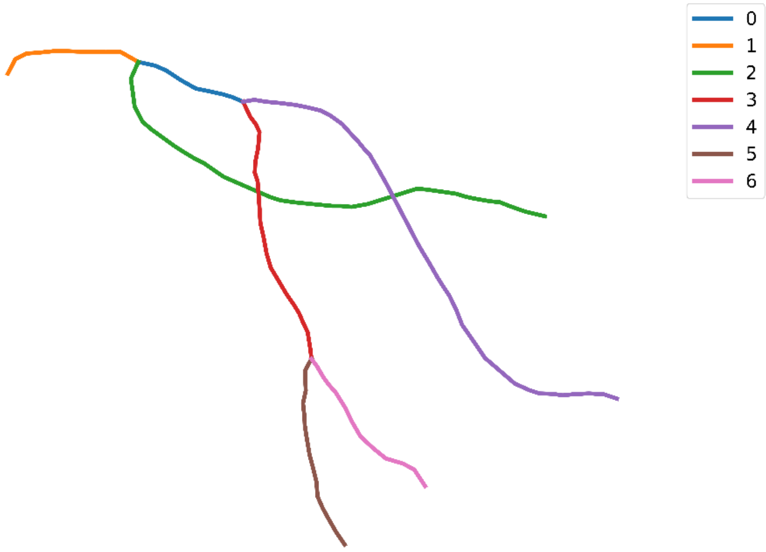


Figure 11: Graph sections obtained by grouping nodes between key-points

4 Task Description

4.1 UC2.2 use case (ELVEZ)

The motivation behind the participation of ELVEZ in REMODEL is to introduce robotic solutions to **primarily perform the task of cable manipulation for assembling the cable harnesses shown** in Figure 12.

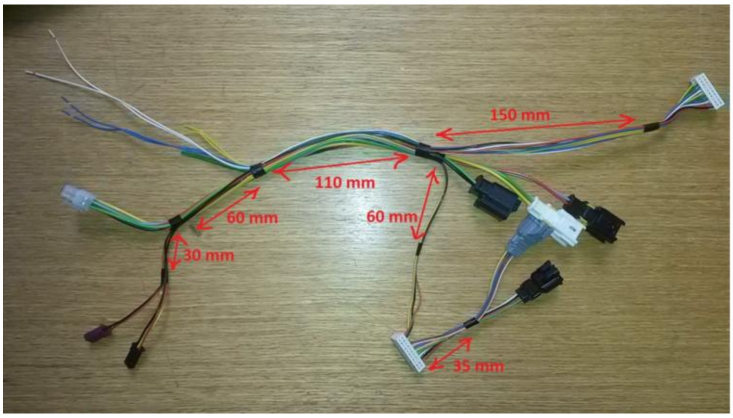
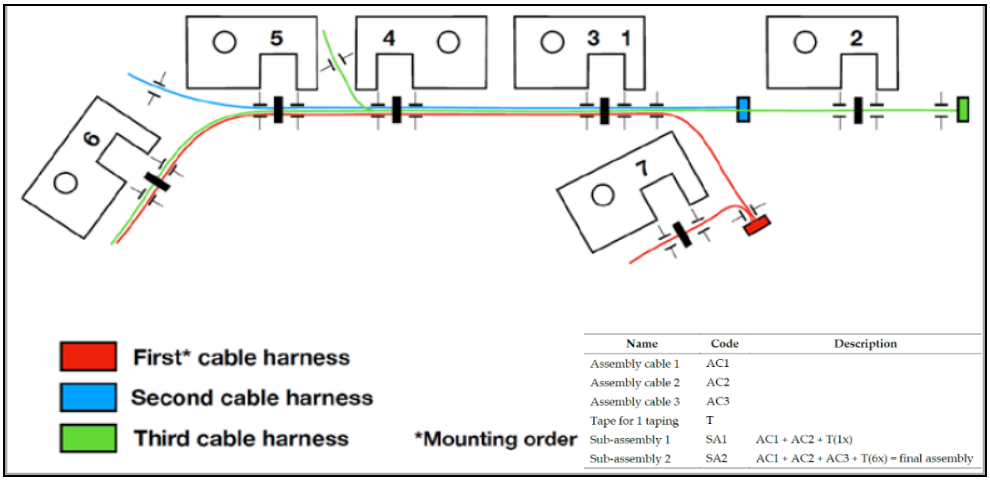


Figure 12: Cable Harness Assembly with the dimensions.

The whole use case description is reported in D2.1 and the product components and schematic assembly are shown in Figure 13.



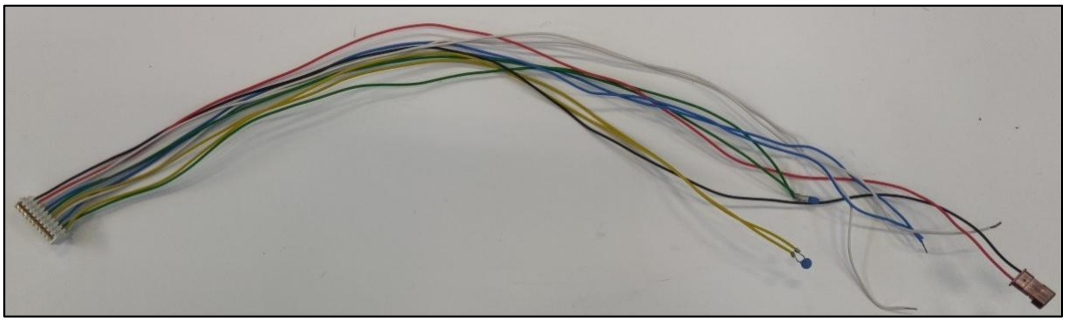


Figure 13: Pictorial representation of the assembly operations and sequence (top); and the wire harness of interest (bottom).

In the following, the main manipulation tasks involving the three cable harnesses constituting the product are described as tasks of interested for this deliverable:

1. Grasp the cable harness from the warehouse and insert the connector head into its designated mold in the assembly jig;
2. Separate the specific groups of cable that need to be routed separately;
3. Route the separated cable groups through the specified path of guides and insert them into the guides; where they are held in a semi-fixed state by means of support clips.

The experimental platform shown in Figure 14 used for the evaluation of these operations is composed by:

- A Panda robotic arm and gripper from Franka Emika;
- A Schunk Force/Torque sensor mounted between the arm and the gripper;
- Suitably designed fingers with integrated proximity sensors (Figure 2);
- A cable harness warehouse;
- An assembly jig equipped with cases to hold the connectors and clips to hold the cables along the desired path (Figure 13);
- A 2D camera to check the cable group separation (Figure 15).

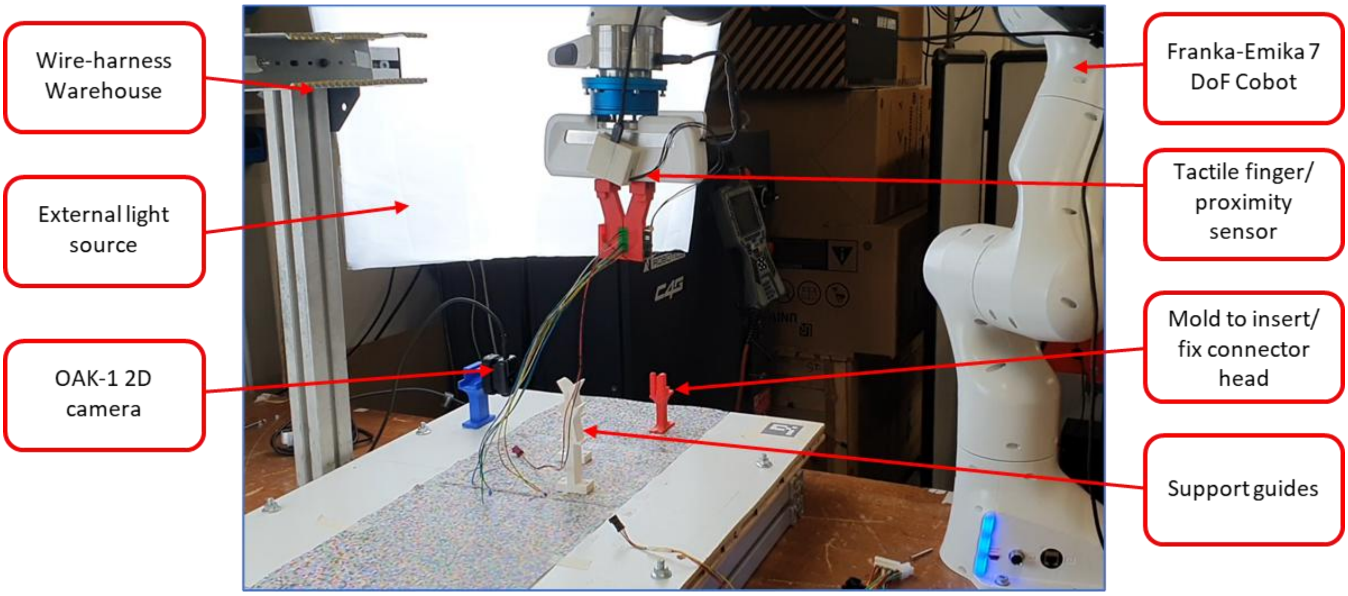


Figure 14: Robotic platform for the wiring harness manipulation tests

Figure 15: Oak-1 2D camera (left); and the proximity sensor integrated fingers (right)

Figure 16 Design of the cable clips and connector holders.

The tasks tested during the integration week were the wiring harness grasping from the warehouse, the connector insertion, and the wires separation executed on the cable harness highlighted in green in Figure 8, without considering the routing phase. In the considered harness, the red and black wires, placed in the top part of the harness in the experimental setup, have to be separated from the remaining wires. The whole task consists of two subtasks: the separation of the wires and the consequent check. For the separation, two strategies have been tested: one relying on proximity perception and the other on vision perception. The final check, instead, is vision based only.

4.1.1 Vision-based cable separation and evaluation:

Initially, the robot manipulator inserts the connector head into the designated mold (Figure 16, middle) and is routed through the support guides (Figure 16, left). The next phase is to perform the cable separation, where the stationary 2D camera captures the scene where the connector head is inserted, and the cable is routed. The '***algorithm'*** identifies the individual cable propagations from their fixed positions (Figure 17, left) in the connector head and determines the point of maximum distance between the two cables of interest (between the red and white cable for this particular scenario), as indicated by the green reticule in Figure 17 (right). This position is updated to the robot manipulator, which then proceeds to the specified location, closes the fingers of the gripper and raises the arm up, performing the cable separation action, as shown in Figure 18 (left). The previous steps conclude the vision-based perception and cable separation aspect of the wiring harness manipulation. The same operation can be performed by utilizing the proximity sensor integrated fingers developed for REMODEL by UCLV in section 4.1.2.

After the cables have been identified and separated, the vision 'algorithm' identifies the cable propagation again as shown in Figure 18 (right), to determine if the right cables have been separated. A video of the cable separation and evaluation utilizing vision system can be found in Alfresco[1].

Figure 17: Vision Perception- Cable perception (left); and grasp point estimation (right)

[1] http://intranet.remodel-project.eu/share/s/rDrzeiYbR1SXbzFz3GI4ZQ

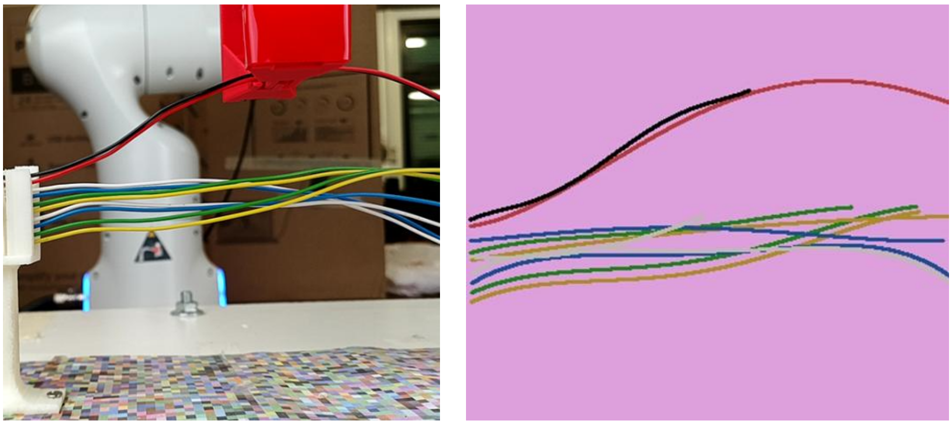


Figure 18: Cable physically separated by the manipulator

4.1.2 Proximity-based cable separation

For the proximity-based separation task, the proximity sensor is used to measure the distance from the top part of the harness connector and the measured distance, together with the known dimensions and the end effector pose, is then used to compute the separation point. The distance from the connector is estimated by placing the proximity sensor right above the connector and taking the mean value of 20 consecutive measurements. Images showing the execution of the separation subtask are reported in Figure 19. In case the exposed part of the connector is big enough, it is also possible to do a scansion of the area following a predefined trajectory to measure the distance in more than one point achieving more robustness. Finally, the evaluation of the cable separation is performed using the vision 'algorithm', as shown in Section 4.1.1. A video of the complete experiment based on the proximity sensor can be found on Alfresco[2]. The video shows all the steps of the experiment that are: grasping the harness from the warehouse; inserting the connector in the holder; measuring the distance from the connector; separating the wires; and checking the final separation.

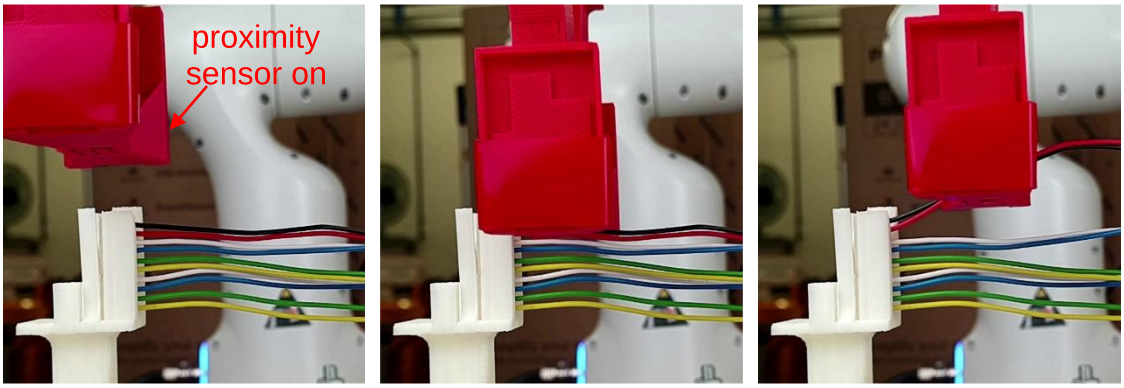


Figure 19 - Video frames from the cable separation task experiment: measuring connector distance (left); reaching computed separation point (middle); separating wires (right).

4.1.3 Tactile-based cable routing

[2] http://intranet.remodel-project.eu/share/s/rDrzeiYbR1SXbzFz3GI4ZQ

Tactile sensors are exploited in the cable routing task to correctly lock the connector into the specific holder (see Figure 21 on the left) and to put the wires in tension during the manipulation, useful to avoid entanglement and to insert the branches into the support clips.

To this aim, the following indicator computed from the tactile signals has been used to detect the force exerted on the wires. This indicator is obtained by using the formula:

$$Indicator = \left| \sum_{r=1}^{5} \sum_{c=1}^{5} (v_{rc}(t) - v_{rc}(t_0)) \right|$$

Where r, c are the indexes, respectively, of rows and columns of the tactile sensor taxels, and the $v_{rc}(t_0)$ are the values of tactile signals acquired as reference, with respect to which we want to compute on-line variations of the current values of tactile signals, i.e., in this context it is the tactile map right before the pulling operation. The indicator is, in practice, related to the tangential value of the grasping force.

A graph showing the value of the tactile indicator during an experiment of the routing task is reported in Figure 20, while some video frames corresponding to the moment when the wires are put in tension are in Figure 21. The indicator in compared to threshold values suitably defined on the basis of the number of wires to manipulate during a specific sub-task of the Elvez use case. In particular, a lower value of threshold is needed when a lower number of wires has to be routed. In the figure, the time ranges related to the routing of only two cables are those with the threshold values fixed to 0.1 (see Figure 21 on right), while a threshold equal to 0.3 has been fixed during the routing of the remaining 8 wires (see Figure 21 in the middle). The video of the whole experiment for the described task can be found on Alfresco[3].

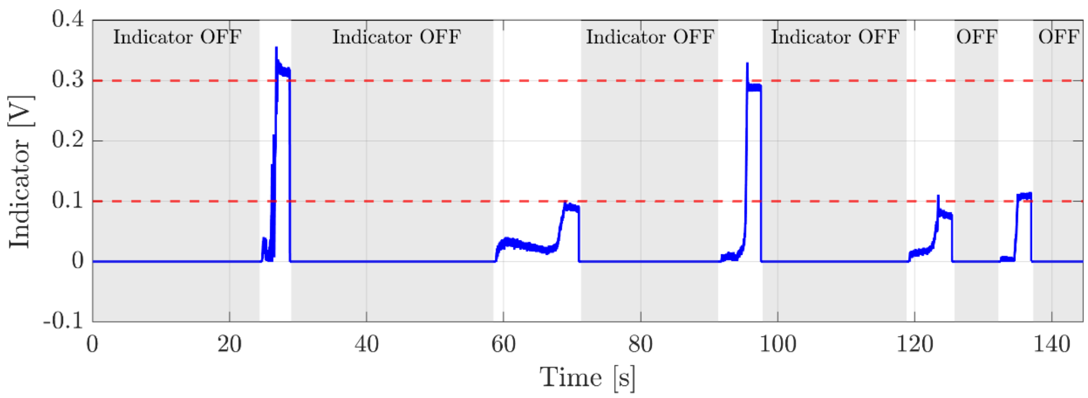


Figure 20 - Tactile indicator during the routing task

[3] http://intranet.remodel-project.eu/share/s/T3Mr6fEGSQ-_uKffw1twbA

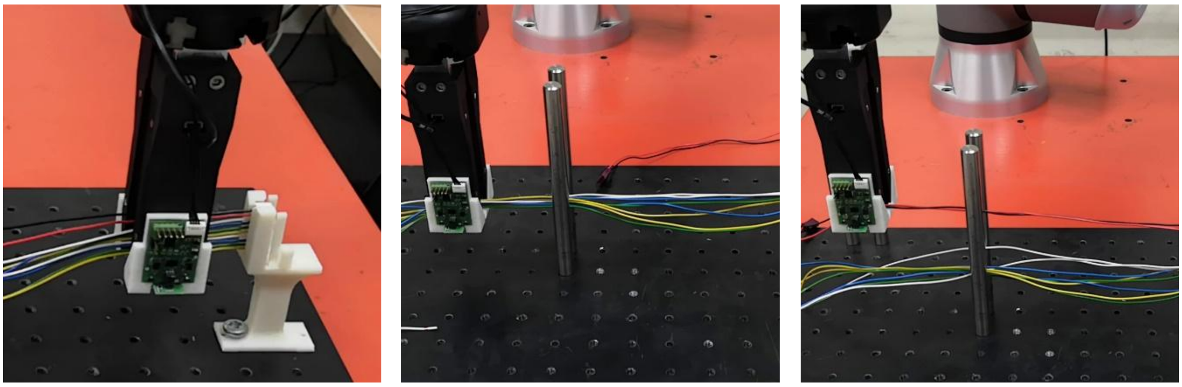


Figure 21 - Video frames from the routing task experiment: connector locking (left); 8 wires pulling (middle); 2 wires pulling (right)

4.2 UC3 use case (VWP)

The goal of the UC3 is to put the cockpit wiring harness on the metal rack, and position the specific parts of the harness (like branches and bags) to prepare it for the further assembly process. The desired state of the harness which have to be achieved is presented in Figure 22, while the initial state of the harness, folded in the transporting box, in Figure 23.

The task can be summarized by the following steps that one has to perform in order to properly prepare the harness:

1) Take harness out of the transportation box;
2) Position the whole wiring harness on the heater and the metal rack;
3) Put bags with wires in predefined areas;
4) Position the key branches with important connectors according to the requirements.

Figure 22: Overview of the cockpit with the wiring harness mounted. This is a desired state that has to be obtained after the manipulation.

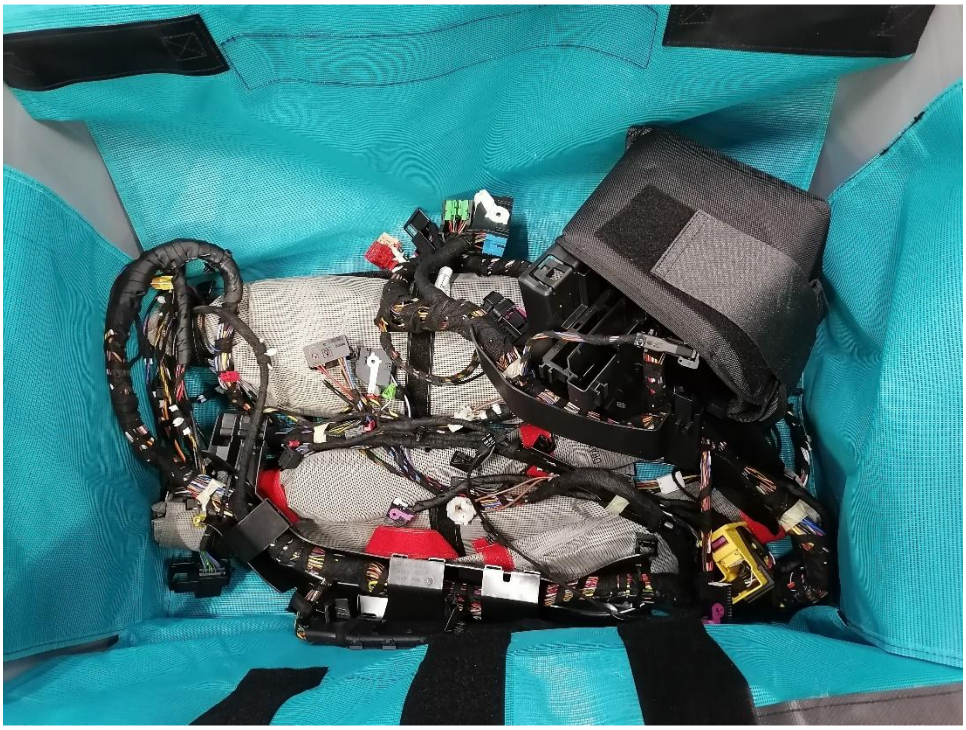


Figure 23: Cockpit wiring harness in the transportation box. This is an initial state of the harness before the manipulation.

The integration week setup is realized at UNIBO laboratories in Bologna with the purpose to test and evaluate all the technologies developed by the partners in a common platform.

It is composed by a collaborative robotic arm with a gripper that permits human presence without any other safety device. The robot used for the experiment is a 7-DoFs panda robot mounted on a side of the table. At the opposite site, a 3D camera has been mounted in front of the robotic arm at a distance of 90cm. The camera has been placed at a height of 75cm and a mounting angle of 15° with respect to the surface. The mounting angle has been decided to avoid having the camera aligned with the surface for improved 3D performances.

The resulting working surface collected from the camera is a rectangular region of 50 cm by 85 cm inside the robot workspace. The camera used is a PhoXI 3D Scanner M made by Photoneo and it the same used from the PUT and for UC3. It worth mentioning that the camera can be easily replace by another model, in particular the Zivid used instead by ELVEZ and TAU, in this way the technologies can be more easily changed and used among different partners.

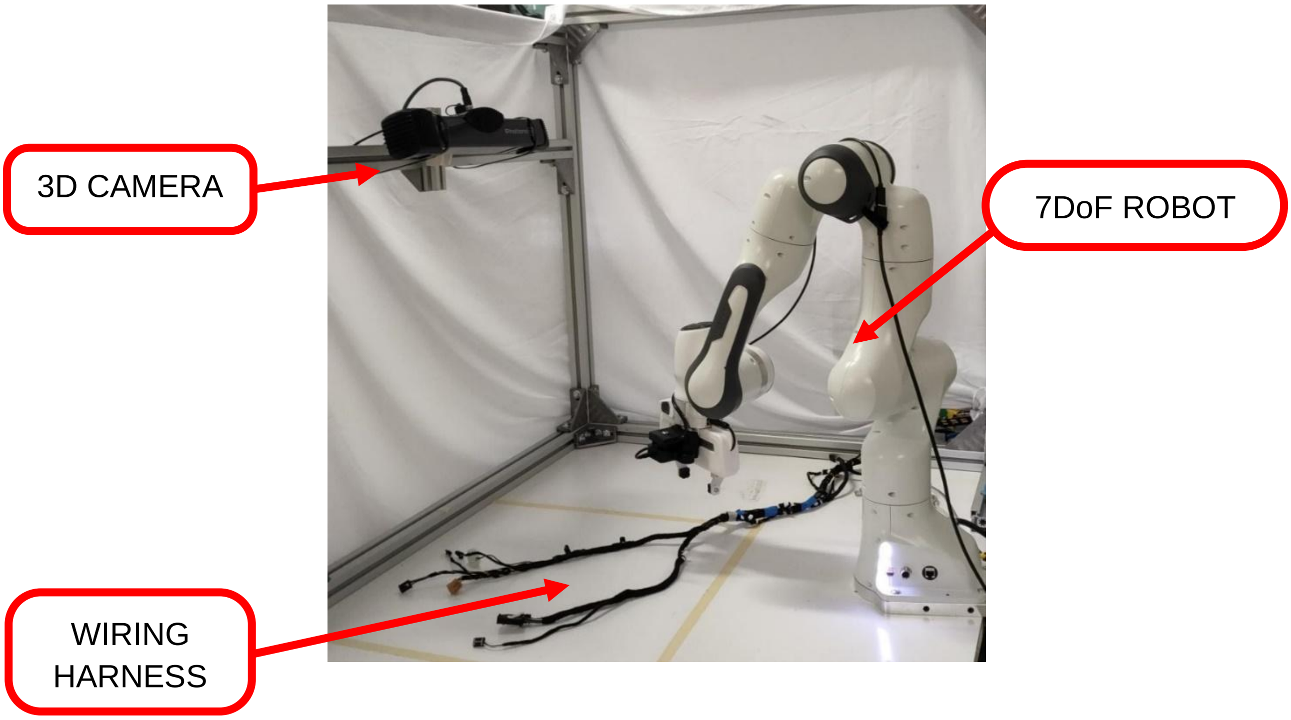


Figure 24: Evaluation setup for the integration week for user case 3

4.2.1 Harness manipulation strategy

The entanglement of wire harnesses can appear at any random configuration due to the unpredictable way of putting the wire harnesses into boxes and their movement during transportation. Moreover, the entanglement of wire harnesses can contain not only simple intersections of two branches but more complex intertwisted intersections as well. Hence, we need an unambiguous and robust way of determining which branch or section of the wire harness has to be manipulated first to start the disentanglement procedure.

4.2.1.1 Level based planner

A simple yet robust wire harness disentanglement planner (WHDP) has been developed. The WHDP outputs, in an iterative manner, the next branch for manipulation. Every section of the wire harness (a section is defined as a part of the wire harness which starts from a branch point and ends with an end point) gets a counter assigned. Then, we iterate over all intersections and increase the counter of the section which is on top. At the end, the section with the highest counter value is the one that must be manipulated first. These steps are repeated until there are no more intersections. Finally, the WHDP algorithm can be described with the following steps:

1. Set section counter to 0;

2. Iterate over all intersections and increment the counter of the section which lays on top;
3. The section with the highest counter value should be manipulated;
4. Repeat step 1. – 3. until there are no more intersections left.

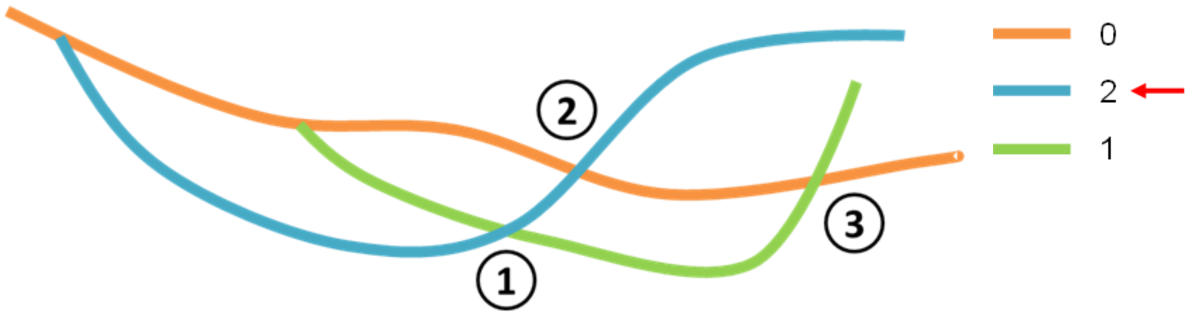


Figure 25: Left: Example scenario of an entangled wire harness. There are three sections (orange, blue and green) and three intersections. Right: Result after applying our WHDP algorithm to this scenario. The blue section has the highest counter value and thus should be manipulated first.

An example of how the WHDP algorithm works is shown in Figure 25. The scenario contains a part of a wire harness consisting of three sections. The sections are coloured with orange, blue and green, and are intersecting with each other. Applying our WHDP algorithm results in identifying the section which should be manipulated first, i.e., the section which has no other part/section laying on it and is therefore free of collisions during the manipulation.

4.2.1.2 Intersection solver

The intersection solver is responsible for computing the trajectory along which the targeted wiring harness branch, determined by the WHDP, needs to be manipulated.

Thus, the intersection solver should be able to deal with all the feasible cases. Moreover, it should be possible to recover from wrong branch placement due to unpredictable wire behavior during motion execution - wire bending depends on many physical factors which might be hard to describe mathematically. Also, the current branch state consists of several parameters. Besides shape, which has been well examined in **D4.3 Cable real-time tracking**, states like wire twist or tension seem to be challenging to estimate based on perception processing. With such knowledge some assumptions were considered. As it was noticed, all the branches of the wiring harness tend to keep a specific shape due to its stiffness. That “origin” shape is noticeable mostly on branching points - it is necessary to use much force to counter wire stiffness close to the branching points. Those forces usually do not affect the wiring harness without performing any action on it. Also, the gravity impact is not enough for that specific wiring harness.

Having considered the above reasoning, looking onto the beginning of branches seems to be a reasonable approach to retrieve the origin shape of the examined branch. Those origin shapes tend to straighten all the branches which provide locally an untangled state of the wiring harness. With resolved local intersections some further action can be performed to resolve intersection on global level - straighten branches in different wiring harness sections may affect each other.

Practically, the intersection solver works in the following way. For a given branch, all the other objects (branches, connectors, etc.) are considered as collision objects and an occupancy grid is constructed. Based on the described strategy, first the current error (branch overlap with occupancy grid) is calculated and the new virtually (without action performing) error is estimated which occurs in case of straighten branch, i.e. the direction of the edge between the considered branch-point and its neighbor in the branch is used as reference. Then, consecutive branch shifts in both directions are considered as actions which may decrease the error as well. For a set of those virtual branch configuration and corresponding errors, the best solution is chosen as the action corresponding to the minimal error, and robot motion is performed. As grasping point, the node half way between the intersection-point being processed and the downstream end-point is chosen.

4.2.1.3 Results

The perception system of Sec. 3.2 is combined with the planner and solvers of Sec. 4.2.1. Several experimental tests have been carried out at the UNIBO laboratory concerning the branches disentangling task. Some examples of the manipulation actions performed are available via a video hosted on Alfresco[4]. These main examples are also reported as Figure 26, Figure 27 and Figure 28. Notice that the before and after manipulation strategies are shown on the left. At the center the occupancy grid detailed in Sec. 4.2.1.2 is depicted with the graph generated by the perception system superimposed for ease of visualization and understanding. It is worth highlighting how the branch that is manipulated is not used for the generation of the occupancy grid. On the right, the pick and place motion is shown with the pick point addressed in the graph as the red node, the placing point as the red cross and the rotation point as the green node. As shown in Figure 27, the level-base planning and the intersection solver are able to cope also in case the branch being manipulated contains more than one intersection.

Although the results are only addressing a single-arm manipulation configuration, the extension to a dual-arm setup is quite straightforward, e.g. employing the second arm for holding the branch nearby the rotation point, and will be investigated in the near feature aiming toward the preparation of the final demos.

[4] http://intranet.remodel-project.eu/share/s/MdjyF1AZTouS8251jebCvg

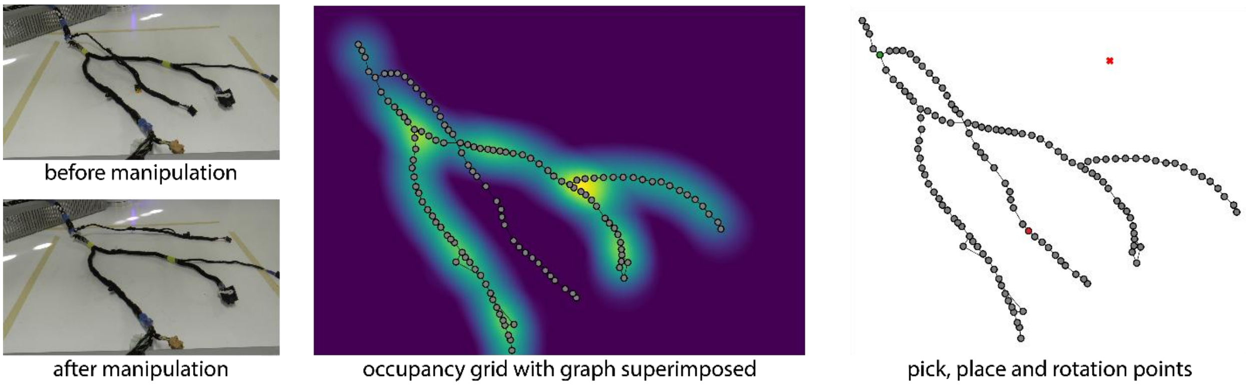


Figure 26: Manipulation example 1

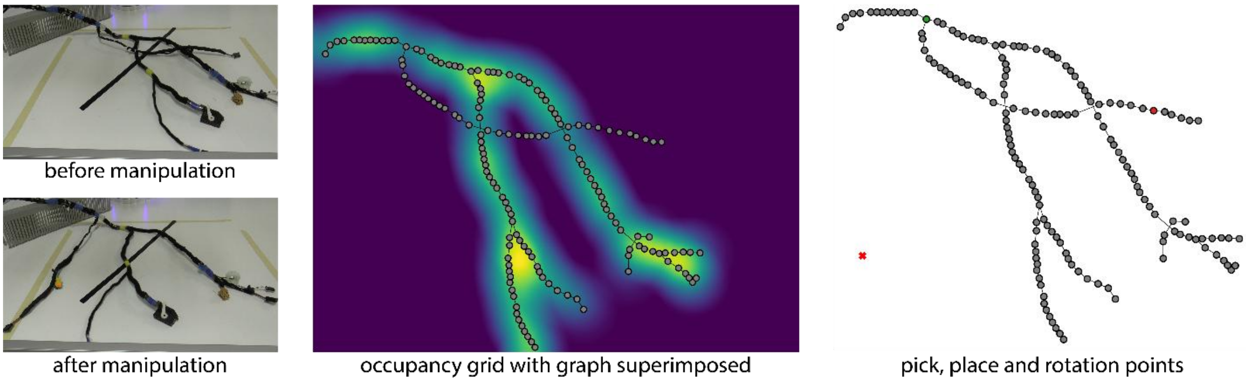


Figure 27: Manipulation example 2

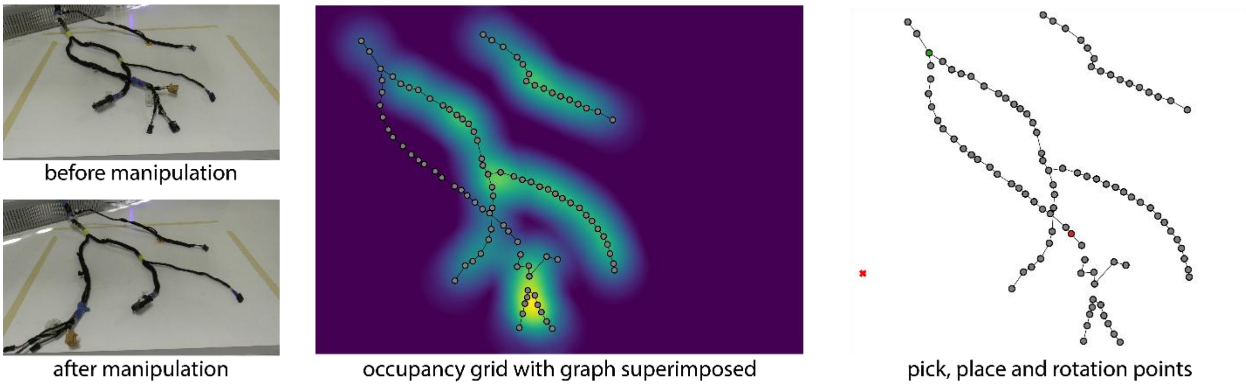


Figure 28: Manipulation example 3

5 Conclusions

This deliverable reports the manipulation strategy developed to manage complex wiring harnesses composed by multiple cables and different branches. In the proposed approach, the separation of entangled branches and their route along the desired path is based on the perception system and the testing is performed considering the wiring harness provided by ELVEZ and VWP.

The wiring harness assembly, disentanglement and arrange/route of the branches along the desired path considering obstacles are shown in laboratory scale demonstrators. Manipulation of single wiring harness branches for the arrangement on supports has been also evaluated. Bimanual manipulation will be used to hold or separate entangled branches. Dual arm manipulation is exploited also to hold, group or to separate properly the branch as desired by the manufacturing task.

These results represent the fundamental technology for the implementation of the final demonstrators for the wiring harness assembly and manipulation proper of ELVEZ and VWP use cases.