Journal of Manufacturing Systems 83 (2025) 577–596

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Technical paper

A dual-arm robotic system for automated multi-branch wire harness assembly in automotive industry

Pablo Malvido Fresnillo [a,*], Saigopal Vasudevan [a], Wael M. Mohammed [a], Jose A. Perez Garcia [b], Jose L. Martinez Lastra [a]

[a] FAST-Lab, Faculty of Engineering and Natural Sciences, Tampere University, Korkeakoulunkatu 7, 33720 Tampere, Finland
[b] Design and Fabrication in Industrial Engineering, University of Vigo, Campus Lagoas Marcosende, 36310 Vigo, Spain

ARTICLE INFO

Keywords:
Dual-arm manipulation
Deformable linear objects
Wire harness assembly
Computer vision
Task-level programming

ABSTRACT

Wire harnesses are critical components in modern vehicles, responsible for transmitting electrical signals and power to sensors and actuators. Despite the high level of automation in the automotive industry, wire harness manufacturing still relies heavilylargely depends on manual assembly. This is due to the significant challenges posed by the process, such as the complexity of perceiving and manipulating flexible materials and the high degree of customization required. As a result, existing solutions only address specific assembly tasks, rather than the entire processare fragmented, unable to scale to full production, and remain economically unviable for high-mix scenarios. To bridge this gap, this paper presents a novel robotic system for fully automating wire harness assembly. The system adopts a task-level programming methodology that leverages process knowledge to enable fast and easy reconfiguration. Additionally, it incorporates specific solutions to address key challenges in multi-branch wire harness manipulation, such as cable separation and entanglement prevention. The system's performance was evaluated in two real-world assembly scenarios using a dual-arm robot. Experimental results demonstrate the system's effectiveness and ease of reconfiguration, achieving success rates of 55% and 73% in two complex multi-branch wire harness assembly processes, and highlight areas of improvement, which will be further investigated in future works. The system repository is openly available allowing other researchers to build their solutions upon the proposed methodology.

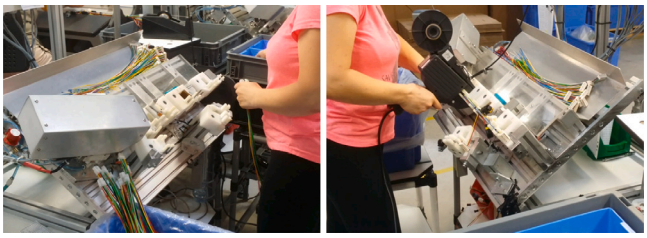
1. Introduction

A wire harness is an assembly of cables that transmit signals or electrical power. The cables are bundled together in a tree-like structure using straps, cable ties, or adhesive tape, and their ends are fitted with electrical terminals or connectors [1]. These products are essential in industries that require the interconnection of numerous electronic components, such as automotive, aerospace, military, medical, and telecommunications [2]. In particular, in the automotive industry, the increasing electrification of vehicle components over the past twenty years has significantly boosted the demand for wire harnesses. As a result, modern cars typically contain several thousand individual wires with a combined length of several kilometers [3], often making wire harnesses the most expensive electrical component in a vehicle [4,5]. Furthermore, with the growing interest in electric and autonomous vehicles, these figures are expected to rise exponentially in the next few years, with the global market projected to reach 74.5 billion USD by 2030 [6].

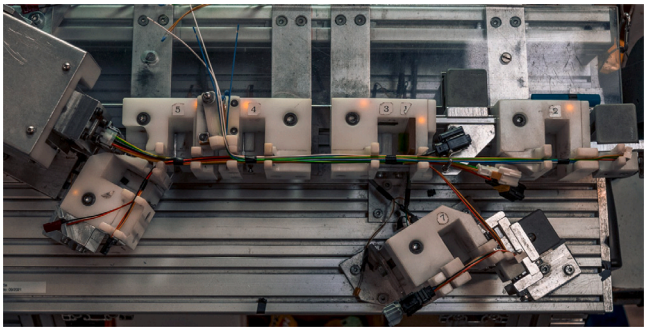
Automotive wire harnesses are complex products, typically composed of multiple individual wire harnesses. Their production process is generally divided into four phases: (i) *cutting*, where individual wires are cut to the desired length and terminals are attached; (ii) *pre-assembly*, which involves finalizing the individual wire harnesses through operations like welding, twisting, or tube-shrinking, as well as assembling small components; (iii) *final assembly*, where the individual wire harnesses are combined to form the final product, routing their cable groups along different guides and bundling them with adhesive tape or cable ties; and (iv) *testing*, where finally the assembled wire harness is electrically tested [7]. The third phase, or assembly phase (which can be observed in Fig. 1), is the most time-consuming stage of the process, taking around 80% of the total production time. However, it is also the least automated one, with an automation degree lower than 10% [7], indicating substantial room for improvement. Moreover, the manual tasks involved in this phase pose significant ergonomic risks to workers [8], further underscoring the strong need to automate the wire harness assembly process.

* Corresponding author.
E-mail address: pablo.malvidofresnillo@tuni.fi (P. Malvido Fresnillo).

https://doi.org/10.1016/j.jmsy.2025.10.008
Received 19 March 2025; Received in revised form 14 October 2025; Accepted 16 October 2025
0278-6125/© 2025 The Author(s). Published by Elsevier Ltd on behalf of The Society of Manufacturing Engineers. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).



(a)



(b)

Fig. 1. Manual wire harness assembly (a) and final product (b). *Source:* Adapted from [9,10] respectively.

While many authors have made contributions toward the automation of this task, the numerous challenges that it entails have so far prevented the development of a complete solution. The primary challenge lies in the robotic manipulation of wire harnesses, which are generally referred to as BDLOs (Branched Deformable Linear Objects) [11]. In addition to the typical difficulties associated with the manipulation and perception of simple DLOs (Deformable Linear Objects), such as their unpredictable shape and their lack of compression strength [12,13], BDLOs present further complexity. Their branched structure makes them even more prone to entanglement and demands more advanced perception and control strategies. Another significant challenge is the high product variety of wire harness manufacturing companies, a result of the high level of customization offered by automobile manufacturers [7]. Hence, the developed system must be able to adapt quickly to different harness configurations and assembly setups to handle the wide range of product specifications efficiently.

The research work presented in this paper aims to provide a comprehensive solution for the automation of wire harness assembly. However, due to the magnitude of the problem and the significant challenges already highlighted, the focus is not on delivering a fully robust and autonomous system. Instead, the goal is to propose a versatile and reconfigurable system that can serve as a foundation for future work. To support this, the project's repository has been made publicly available, allowing other researchers to build upon it. Additionally, this work aims to develop new perception and manipulation algorithms to help address some of the main problems involved in the assembly process (e.g., cable routing, cable separation, etc.) and integrate them within the proposed system architecture. To achieve this objective, the problem was broken down into three research questions:

RQ1 *How can different cable groups of a wire harness be separated when they must be routed along different paths?*

RQ2 *How can cables be routed in cluttered spaces without supporting structures and while avoiding entanglements?*

RQ3 *What kind of system is required to enable integration and seamless reconfiguration for a new assembly process?*

The rest of the article is structured as follows: Section 2 reviews related research, covering both complete and partial approaches to automating the various tasks involved in wire harness assembly. Section 3 presents the developed system, first introducing its structure and different modules, and then detailing how these modules work together to achieve the desired functionality (Section 3.4). Section 4 documents the experiments performed to validate the proposed methodology and analyzes the results. Finally, Section 5 concludes the article, highlighting the limitations of the proposed approach and suggesting possible directions for future work.

2. State of the art and related works

The automation of processes involving the manipulation of wire harnesses is one of the most challenging problems in robotics today. Several authors have proposed approaches for automating different phases of the wire harness lifecycle, from the early production stages [14] to their final installation inside the car's cockpit [15,16]. This section focuses on the assembly phase, which is the most time-consuming and least automated [7], reviewing the most relevant approaches proposed in the literature.

Due to the magnitude of the problem, most of the proposed solutions are partial, targeting a specific aspect of the assembly. In general, the existing approaches focus on one of the following categories: bin-wire harness picking, cable routing, taping and ties collocation, quality inspection, and connector mating. The remainder of the section analyzes the main research contributions in each of these categories, and finally, Table 1 summarizes the target application of all the reviewed approaches.

Bin-Wire harness picking: This is generally the first task in wire harness assembly and it involves extracting the individual wire harnesses from the initial bin, typically from a hanging rack, a bin or a container. In [17], a computer vision method was presented to distinguish different types of multi-branch wire harnesses hanging from a rack and to determine their optimal grasping point. First, the point cloud of the target wire harness is segmented using various clustering techniques and color similarity comparisons. Then, the picking point is identified based on the distances between its individual wires. A different approach was proposed by Kamiya et al. [18], enabling a robot to learn how to grasp wire harnesses from a bin after a single human demonstration. A Mask R-CNN model segments the optimal grasp region, after which the grasp pose and point cloud are extracted during the demonstration. In a new scenario, the grasp is transferred by aligning the new and demonstrated point clouds. However, this method assumes only a single wire harness in the bin and is not robust to entanglements.

Kamiya et al. [18] proposed a method for automating this task, which enables a robot to learn how to grasp wire harnesses correctly from a single human demonstration. To achieve this, first, the authors trained a Mask-RCNN model to segment the optimal grasp region of the wire harness, which has a characteristic structure. Afterward, the user performs the grasping demonstration, from which the system extracts the grasp region (using the trained model) and its corresponding point cloud, along with the hand's grasp pose. To transfer the learned grasp pose to a new scenario, the transformation field is obtained by aligning the point cloud of the new grasp region with the point cloud from the demonstration through point cloud registration. However, this approach is only valid when there is a single wire harness in the bin, as it is not robust against entanglements.

Zhang et al. [19] addressed this issue by proposing a set of disentangling actions. Their system first identifies potential grasp points and then uses a prediction model to foresee the performance of applying different disentangling actions at each

grasp point. The optimal action–grasp pair is selected based on the predicted scores. This solution was later extended in [20], where the authors introduced two new disentangling actions, more suited for long wire harnesses, that leverage bimanual and dynamic manipulation, along with haptic feedback to detect entanglements in real-time.

Cable routing: The next step after picking the wire harnesses is to route them. This task involves guiding their various branches along specific paths. These paths are typically made up of structural components like pins, clamps, or guides, which help to organize and secure the wire harness as it is being routed. While most of the approaches in the literature focus on routing individual cables, less attention is given to the specific challenges of multi-branch routing, such as entanglements and cable separation. Guo et al. [21] presented a path planning algorithm, based on Bi-RRT, for cable routing in narrow spaces. Huarte et al. [22] proposed a cable routing solution for an aeronautic wire harness assembly process. In this approach, individual wires are routed along different paths without relying on sensor data. The target paths and the workbench pin locations are automatically extracted from the CAD drawings of the assembled wire harness. Using this information and some predefined rules, the system is able to perform the offline computation of the robot's trajectory. A similar approach is followed by Fresnillo et al. in [23], where a set of parameterized bimanual routing operations are defined. These simple operations can be combined to perform complex routing tasks. As in the previous case, these operations do not rely on sensor data and are fully configured using information extracted from various files, including the CAD 3D model of the workbench.

The practice of breaking down cable routing into a sequence of simpler, predefined, and configurable robot operations is common and has been implemented by several authors. An example is the system proposed by Galassi et al. [24], which includes three routing operations for rotating the cable around a pin, inserting it in a clamp, or both. Unlike the previous approaches, these operations use sensor feedback to adjust the computed trajectories during manipulation. In particular, tactile sensing is employed to prevent the cable from slipping out of the finger pads, and force sensing is used to control the tension applied to the cable. Similarly, Süberkrüb et al. [25] proposed a comprehensive list of cable routing primitives with force feedback. These primitives consist of basic actions such as grasping, sliding, tensioning, or inserting a cable, which can be combined to perform a wide range of more complex operations, like those proposed in the previous approaches. The primitives are automatically configured based on a reduced DLO model, which estimates the cable's state using the measured forces.

Besides force sensing, computer vision is another commonly used perception technique in robotic cable routing approaches. Nguyen et al. [26] utilized a depth camera to detect a single-branch wire harness hanging from a clamp and compute its optimal grasp point. First, the clamp is detected using a Mask R-CNN model. From this starting point, the wire harness profile is roughly estimated sequentially using a CNN model. The accuracy of the prediction is then improved by analyzing the depth of the neighboring points around the predicted harness profile. This precise estimation of the wire harness shape enables the system to select the optimal grasp point for routing it into the next clamp. A different kind of computer vision application for cable routing is presented in [27]. This approach proposes a motion planning framework for cable routing involving environmental contacts. The framework divides the manipulation process into three phases based on an angular contact mobility index introduced by the authors: pulling, pre-contact rotation, and post-contact rotation. An eye-in-hand camera and computer vision processing are used to detect contact between the cables and the contact pins, adjusting the routing phase accordingly.

All the previously mentioned cable routing approaches offer valid solutions for single cables or single-branch wire harnesses. However, when applying them to multi-branch wire harnesses, additional considerations are necessary. In particular, if the harness branches need to follow different paths, they must first be detected and separated. This is typically addressed using computer vision techniques. For example, Nguyen et al. [28] proposed a system that utilizes an RGB-D camera to identify the profiles of the multiple branches of a wire harness and, based on this, define the optimal manipulation strategy. The procedure followed by the authors begins with the wire point cloud segmentation through a combination of background subtraction and color classification techniques. Then, the connectors at the end of each branch are identified and classified using a combination of the YOLO detection model and a 3D point-to-plane matching technique. Finally, a tracking algorithm iteratively reconstructs the 3D profile of each branch, starting from the position of its corresponding connector. Similar vision-based approaches for identifying different wire harness branches have been proposed in the literature, both for simple wire harnesses [29] and wire harness sub-assemblies [30].

Alternatively, Govoni et al. [31] proposed a method for separating wire harness branches that relies solely on data from a proximity sensor. In this approach, the wire harness connector is first secured in a fixture, and its exact pose is then determined using the sensor. Since the relative positions of the wires within the connector are known, the grasp point for separating any group of cables can be easily computed. To minimize uncertainty, the separation point should be as close as possible to the secured connector.Despite its lower complexity, this method depends on the cables being horizontally and neatly arranged; otherwise, their positions beyond the connector cannot be inferred. This limits its applicability in more cluttered environments, where external contacts can deform the cables. The approach was later extended in [32] to perform a complete multi-branch wire harness assembly, including routing, taping, and final inspection.

Unlike all these approaches, which propose different sensor-based manipulation policies to accomplish the robotic assembly, Lv et al. [33] presented a physics-based multi-branch wire harness assembly simulator. This tool models the joints of multi-branch cables, and considers their topological and anatomical features. This could potentially allow testing different manipulation strategies faster or directly learning them using techniques like reinforcement learning. Furthermore, besides the manipulation policy, multi-branch wire harness assembly has other challenges, such as planning the sequence of operations to be performed. Zhou et al. [34] proposed a systematic approach to this problem, which take into account both the topology of the harness structure and the nature of the assembly operations.

Finally, it is worth mentioning that due to the deformability and the size of the wire harnesses, most of the reviewed approaches utilize dual-arm robotic platforms for their manipulation [22,23,25,26,28]. Motivated by this, Kim et al. [35] developed a dual-arm path planning technique specifically for wire harness routing.

Taping and ties collocation: Once the wire harness branches have been routed, they have to be bundled together using adhesive tape or cable ties. Ibañez et al. [36] presented an approach to automate the taping task using a collaborative robot equipped

with an automatic taping gun. This robotic platform receives the wire harnesses already routed and it simply has to insert the taping gun at the required locations and actuate it. A similar solution was presented by Palomba et al. [37], with the key difference being that the robot inserts the taping gun from the back of the platform through dedicated holes. This leaves the front of the assembly platform free, allowing space for other activities, such as cable routing. Navas-Reascos et al. [8] applied the same concept for the robotic placement of cable ties from the back of the platform.

Quality inspection: For a robotic solution to autonomously handle wire harness assembly, it must be capable of accurately assessing the quality of the completed assembly. Nguyen et al. [38] proposed a methodology to visually evaluate the quality of the assembly using an RGB camera. In this method, first, the main components of an assembled wire harness are segmented (e.g., connectors, clips, taped cable branches, non-taped cable branches, etc.), and then some quality parameters are extracted, such as the routing path of the wire bundles, the component's color, and location, etc. By comparing the extracted parameters with the target ones, the system can classify the assembly as successful or not. This methodology was later adapted for point cloud images by the same authors [39], obtaining more accurate results. In this case, a PointNet++ model was used for the segmentation, which was trained with real and synthetic data.

The previous approaches focus on assessing the quality of the final assembly. However, some authors have proposed solutions to detect errors or anomalies during the assembly. For instance, the system presented by Galassi et al. [40] detects whether all cables have been successfully inserted into the clips during the routing stage. This information is derived from a point cloud using an encoder–decoder model. Another example is the approach proposed by Zürn et al. [41]. This system can identify topological features of the wire harnesses (e.g., cable ties, connectors, or branch points) during the manipulation, as it is robust against the occlusions caused by the operator's hands. In this case, the information is extracted from 2D images using a Mask R-CNN model.

Connector mating: Finally, after the wire harness has been assembled, it needs to be electrically tested by inserting its connectors into their counterparts in the testing board. This requires first, identifying the connectors and then, manipulating them. Multiple approaches have been proposed for connector detection, both utilizing traditional computer vision techniques [42] and Deep Learning models [43,44]. Regarding their manipulation, several solutions have been proposed. For instance, Zhou et al. [44] suggest the use of a dual-arm robot for this task to facilitate the correct alignment of the connector. Moreover, Yumbla et al. [45] characterized the mating tolerance of a large dataset of connectors, as well as the forces required for their insertion.

As can be seen, the state-of-the-art offers only partial solutions that address specific phases of the wire harness assembly process. This allows each operation to start from controlled initial conditions, which simplifies robotic handling—especially when dealing with highly deformable products. In contrast, this paper presents a comprehensive system that integrates solutions for multiple stages of the process. Specifically, the proposed system is capable of performing wire harness picking (under simplified conditions), cable routing, cable separation, taping, and quality inspection. To concatenate all these operations and prevent the accumulation of errors, the system relies on accurate perception (vision and force) and detailed information about the environment.

Table 1
Reviewed state-of-the-art wire harness robotic manipulation approaches classified by their application.

Application	Approaches
Cutting & pre-assembly	[14]
Wire harness picking	[17–20]
Cable routing (single branch)	[21–27,35]
Cable routing (multi-branch)	[28–34]
Taping and ties collocation	[8,36,37]
Quality inspection	[38–41]
Connector mating	[42–45]
Cockpit installation	[15,16]

Finally, the novelty of introducing a complete robotic system for wire harness assembly is highlighted in Table 2, where the different operations performed by other approaches in the literature are summarized and compared with our solution. As can be seen, only [28,32] present similar solutions. The functional differences, limitations, and performance of these two are therefore discussed and compared with our solution in Section 4.

As can be seen, the state-of-the-art only presents partial solutions for specific phases of the wire harness assembly process. This paper aims to present a comprehensive system that integrates solutions for all stages of the process, enabling the full automation of multi-branch wire harness assembly.

3. System description

As discussed in previous sections, automating wire harness assembly poses significant challenges. The primary difficulty lies in the complex manipulation of these products, which consist of multiple small DLOs prone to entanglement with each other and the workbench components. As a result, the robotic system requires advanced motion planning, control, and perception capabilities. Another key challenge is reconfigurability, as the system must be able to quickly adapt to different assembly models and setup configurations. Moreover, for practical deployment in industrial environments, the system must be user-friendly and operable by individuals with no robotics expertise while ensuring safety.

To address all these challenges a multi-module ROS-based system is proposed, which can be seen in Fig. 2. This system integrates various modules, each tackling specific challenges, by leveraging ROS (Robot Operating System) topics, services, and actions. It consists of the following modules: the *CAD platform*, which analyzes a set of files describing the assembly process and extracts relevant information; the *task planner*, which decides the operations to perform based on the information received from the CAD platform; the *robot operations*, containing the configurable definition of the entire list of operations that the robot can execute; the *perception* and *motion planning modules*, which extend the manipulation capabilities of the different robot operations; the *user interface*, adapted from [46], which allows the user to easily and intuitively monitor, control, and configure the system; and the *safety manager*, which guarantees the user's safety.

To enhance reconfigurability, the system follows a task-level programming methodology, discussed in detail in Section 3.4. This approach enables users to configure all modules and their interactions using high-level instructions to adapt to different wire harness assembly processes. The robotic assembly sequence consists of the following steps: first, a wire harness is picked from the cable-hanging platform. Next, its main connector (i.e., the one connected to all the wires) is secured in a designated fixture on the assembly platform. Then, its cable groups are routed through different sets of guides, separating them beforehand if there is more than one. This process is repeated for all assembled harnesses and, finally, they are bundled using an automatic taping gun.

Table 2
Wire harness assembly operations covered by state-of-the-art robotic approaches.

	Connector placement/ fixturing	Cable routing	Dual-arm cable routing	Multi-branch separation	Vision-based multi-branch separation	Taping	Assembly inspection
[22]		X	X				
[23]	X	X	X				
[24]		X					
[25]		X	X				
[38]		X	X				
[27]		X					
[28]		X	X	X	X		
[31]	X	X		X			
[32]	X	X		X		X	X
Ours	X	X	X	X	X	X	X[a]

[a] Cable separation success inspection

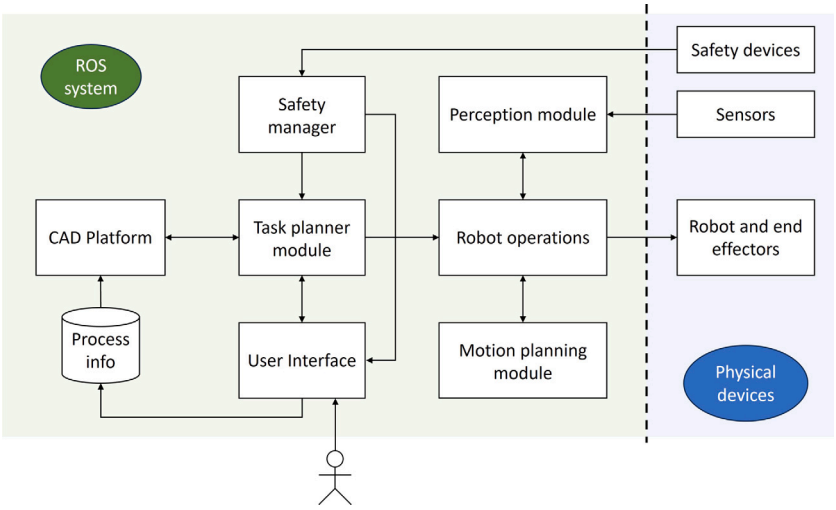


Fig. 2. Structure of the system.

To execute these operations, the system relies on a set of manipulation capabilities provided by the perception and motion planning modules. Specifically, the perception module includes a computer vision algorithm to identify the optimal grasp pose for cable separation (detailed in Section 3.2) and force data analysis to detect when the cables are taut. Meanwhile, the motion planning module incorporates a path-planning algorithm designed to prevent entanglements between the wire harnesses and the assembly platform (introduced in Section 3.3), and it integrates the extended MoveIt! framework from [47], which provides dual-arm coordination and automatic tool changing (ATC) capabilities.

The following subsections introduce the main contributions of this work: the CAD platform, the vision-based cable separation method, the tangle-free path planning algorithm, and the task-level programming methodology. Additionally, the system's repository is publicly available[1].

3.1. CAD platform

To allow the automatic reconfiguration of the proposed system, all the robot trajectories, the sequence of robot operations, the decisions, and the configuration of all the modules, depend on some key information about the assembly process, the physical layout, the manipulated wire harnesses, and the available robotic tools. The CAD Platform is the module responsible for extracting this key information from a set of configuration files, processing it, and making it available to the rest of the modules through ROS services. This enables the entire system to be reconfigured simply by updating these configuration files, which is crucial to answer RQ3.

Fig. 3 shows the complete pipeline of the CAD platform module. As can be seen, the first step consists of extracting relevant information from multiple configuration input files with different formats. These files can be classified into four groups:

(i) Physical platforms: A physical platform is defined as an element of the layout that is directly involved in the assembly process. These platforms can consist of one or multiple components with a common functionality. Three physical platforms are used in the proposed system, which can be seen in Fig. 4. These platforms are: the *assembly platform*, composed of a set of guides, connector fixtures, and cameras, used to route, tape, and assemble the individual wire harnesses; the *cable-hanging platform*, composed of a set of guides from which the wire harnesses are picked initially; and the *automatic tool changing (ATC) platform*, which has different slots for picking and placing the robot's tools. Each of them is completely defined with three files: the CAD 3D assembly model of the platform (which inspired the name of the module), in X3D format; an XML file that contains the dimensions and frames of interest of all the different components of the platform (i.e., one description per model); and a TXT file that specifies the model and the label name for each component of the CAD assembly. Additionally, the position and orientation of the platform within the scene are specified from the CAD Platform launch file with a static transform publisher. Besides these three files, the STL mesh of each platform component is provided to allow its visualization.

[1] https://github.com/pablomalvido/remodel_ws

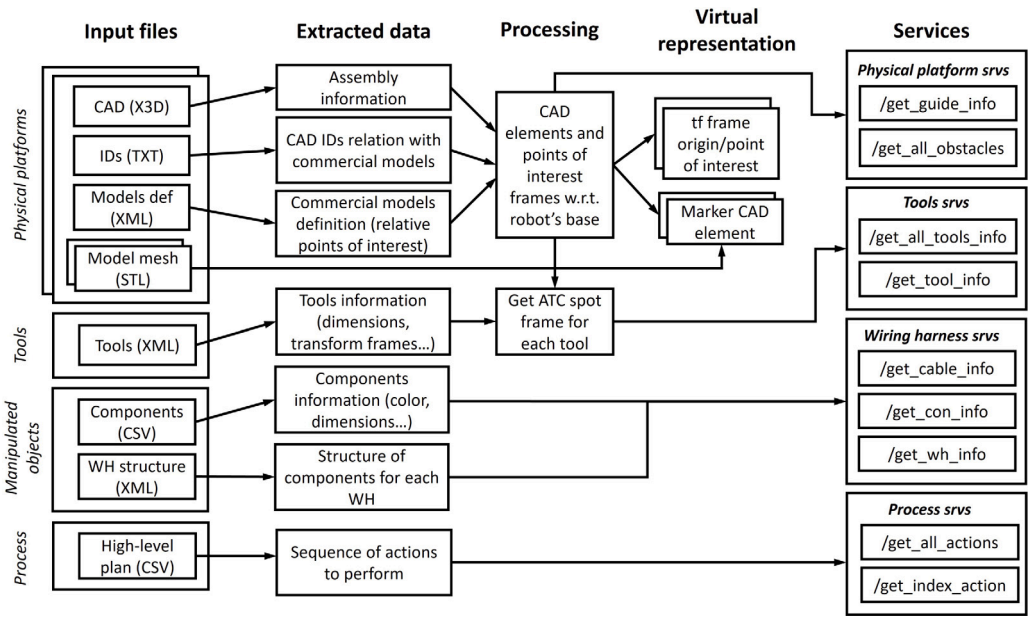
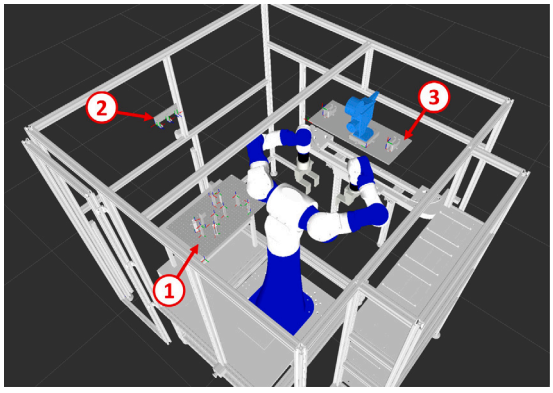
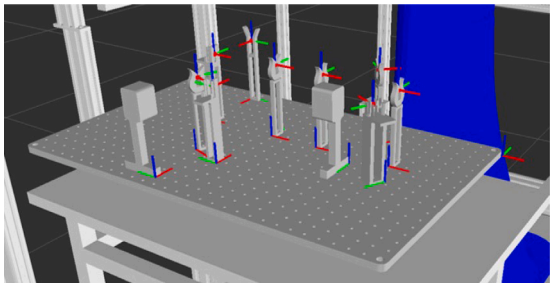


Fig. 3. CAD platform pipeline.



(a)



(b)

Fig. 4. Physical platforms of the system and their frames of interest visualized in RVIZ. (a) Global view of the cell. 1: Assembly platform, 2: Cable-hanging platform, 3: ATC platform; (b) Close view of the assembly platform.

(ii) Tools: One XML file is used to describe the robotic tools utilized for the process (i.e., an automatic taping gun, and two parallel grippers), specifying their dimensions, ATC platform slot, and frames of interest.. This description includes the tool's dimensions, the label identifier of its slot in the ATC platform, and the transformation frame from its tool changer frame to its actuation frame (i.e., the nozzle of the taping gun or the finger pads of the grippers).

(iii) Manipulated objects: Two files are used to describe the manipulated wire harnesses. A CSV file with information about all their cables and connectors, such as their color and dimensions; and an XML file that describes the structure of the harnesses, specifying how the previous cables and connectors are arranged in each of them.

(iv) Process: One CSV file defines the high-level plan of the assembly process. This plan contains the sequence of actions to be executed by the robot. For each action, it indicates its type (i.e., fixture connector, route cables, or tape), the component involved (e.g., a certain connector, or a group of cables), and the location (i.e., the label name of the platform elements where the actions must be performed). It is worth noting that the action types and the robot operations may not necessarily be the same, and the task planner is responsible for selecting the most appropriate operation in each case, as explained in Section 3.4.

After the relevant data of these files is extracted, it sometimes needs to be processed and merged, as they may provide complementary information. This is the case of the three files describing the physical platforms. The X3D file provides the relative pose of each component of the platform with respect to its parent component; the XML file contains the frames of interest of each component model with respect to its own origin frame; additionally,and the TXT file specifies the model number of each component in the CAD assembly. Therefore, to obtain all the frames of interest of the platform with respect to the robot base frame, the information of these three files must be combined, identifying and concatenating the hierarchical chain of transformations for each of them. To accomplish this, the hierarchical chain of transformations from the robot base frame to each component's origin frame is first identified. Then, for each component, its model is checked, and its relative frames of interest are added to the chain. Finally, for each point of interest, its frame with respect to the robot base is obtained by multiplying its entire transformation chain. Furthermore, the data extracted from the tools is also combined with the physical platforms' information, particularly to obtain the ATC frame of each tool with respect to the robot base frame.

Finally, the processed information is provided on request to any module of the system through ROS services. These services can provide, for instance, information about a specific guide, end effector, wire harness, or cable; or about all the obstacles in the assembly platform, or all the actions of the process plan. The entire list of provided services can be seen in Fig. 3. Additionally, the information from the CAD Platform is used to visualize the robot platform environment in RVIZ (see Fig. 4). The components of the physical platforms are displayed as Markers, using their STL files to define their geometry, and all their frames of interest are represented with TF frames. This can be observed in Fig. 4.

3.2. Vision-based cable separation

Cable separation is one of the most complicated operations in a robotic wire harness assembly process as the cables are small, adjacent, and could be entangled. All this makes the identification of the individual wire harness cables significantly challenging. Furthermore, in some cases, the maximum distance between the cable groups to be separated can be lower than one millimeter. Hence, the vision-based solution for the identification of the optimal grasp point to separate specific cable groups must be very robust and accurate. The methodology proposed to address this issue is presented in Fig. 5, which provides a solution for RQ1.

As can be seen, the input of the pipeline is an RGB image of the cables. This image is taken with a static camera when the harness connector is already fixed, hence, the position of the connector, as well as the initial position of every cable can be obtained from the CAD platform information (particularly, from the dimensions of the connector and the pose and dimensions of the connector fixture). Furthermore, as the camera orientation is perpendicular to the cables' plane, and at a fixed distance from it, the system can be calibrated to know the pixel-to-millimeter ratio within that plane. After this initial calibration and image pre-processing, the next step is the semantic segmentation of the image, removing the background and keeping only the cables. This segmentation is performed by a U-Net model trained on the synthetic cables dataset presented in [48].

The next step is to identify the cables' shape. This is done utilizing the approach presented in [29], which processes each cable individually with a pipeline that consists of three sequential steps: a color filter to isolate the analyzed cable (or at least reduce the noise in case there are several cables with the same color), an algorithm to identify discrete cable points, and a polynomial fitting of the identified points to estimate the cable shape. Finally, an algorithm is used to assess if there are errors in the shape estimation, discarding it in that case. In order to speed up the process, this pipeline is only applied to the two top cables of the lower group and the two bottom cables of the upper group (i.e., from cable index $s-2$ to $s+1$, being s the separation index, which is the index of the lowest cable of the upper group).

Then, the grasp point analysis limits (L) are identified. The analyzed interval starts at an offset distance from the connector fixture, consisting of half the robotic fingers' width plus a margin of safety to avoid potential collisions, and it extends either until the next obstacle in the assembly platform or for a maximum distance (in this case fixed to 60 mm). This maximum distance is considered because in most cases, the furthest from the connector fixture, the more unpredictable the cables' detection becomes, and the more prone to entanglements. After setting the limits, the next step is to check if the cable predictions extend for the entire interval, as there might be occlusions. Thus, if the detection does not reach the end of the interval, to be able to continue with the analysis, its estimated shape is extended with a horizontal line from the last detected point.

After calculating the cable shape functions for the entire interval L, the lower envelope function (g_t) of the cables from the top group and the upper envelope function (g_b) of the cables from the bottom group are computed. These functions are defined in [49] as:

$$g_b(x) = \sup_{i<s} f_i(x) \quad \forall\, x \in L$$

$$g_t(x) = \inf_{i\geq s} f_i(x) \quad \forall\, x \in L$$

A graphical explanation of these functions can be seen in Fig. 6, in which the second and third columns show the upper and lower envelope functions of the bottom and top cable groups of the first column plots respectively.

Once calculated, these two functions ($g_t(x)$ and $g_b(x)$), as well as the evaluation interval (L) and the number of analyzed points (n), are

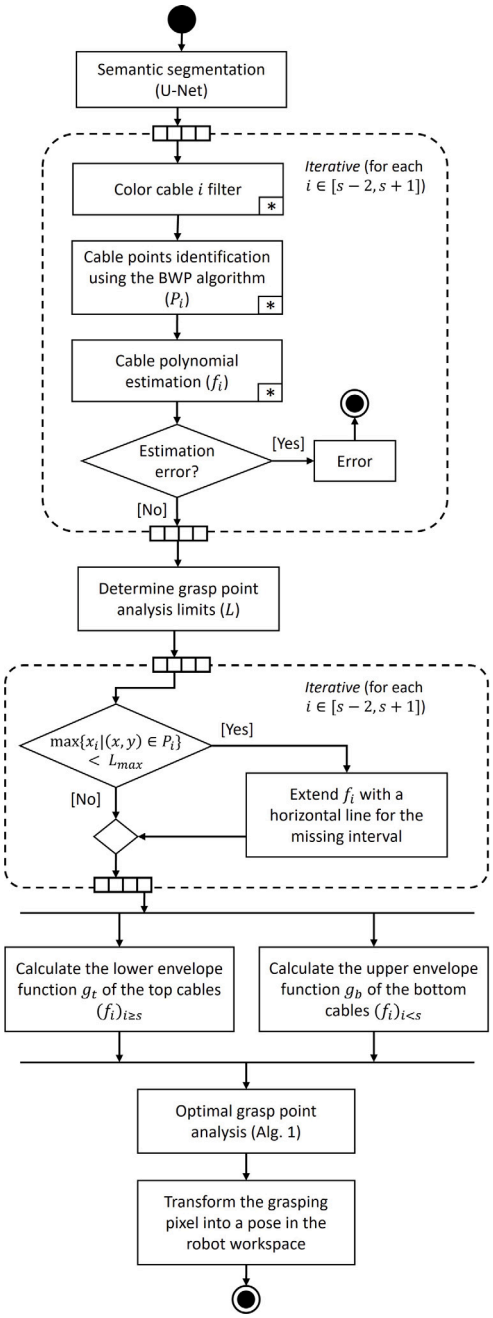


Fig. 5. UML activity diagram of the vision-based grasp point identification for cable separation methodology. The blocks with an * symbol on the lower right corner have been taken from [29]. Notation clarification: s is the lowest index of the separated cables.

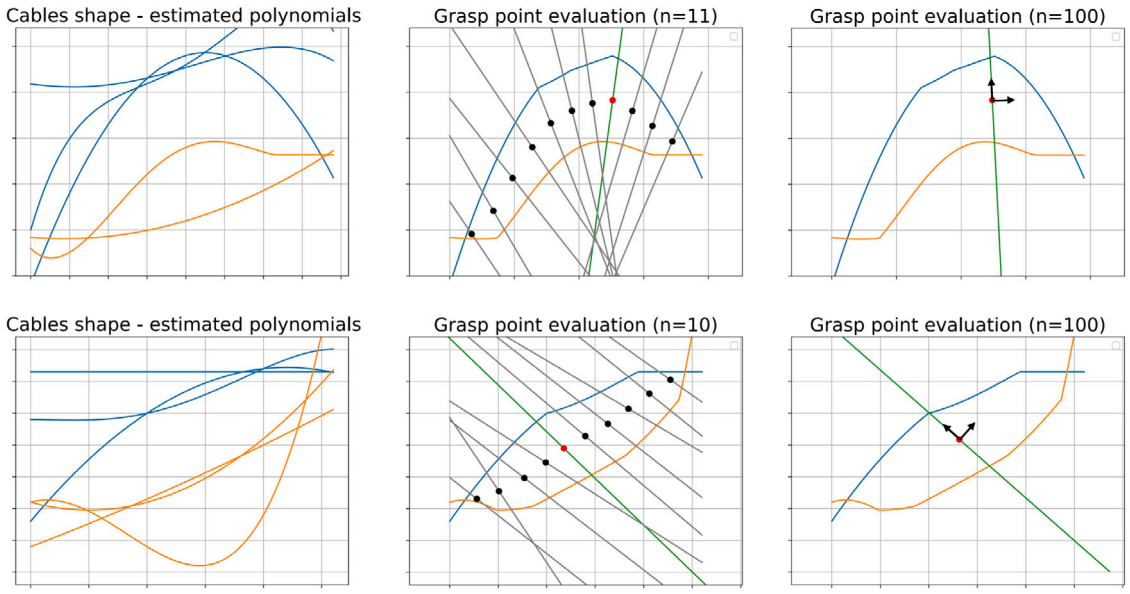


Fig. 6. Visual examples of the optimal grasp point analysis algorithm (Algorithm 1). Each row presents a different example. The first column shows the detected cable shapes, top cables are represented in blue, and bottom cables in orange. The second column shows all the analyzed points (black) and directions (gray), as well as the best grasp point (red) and direction (green), with a low n. Additionally, the blue and orange lines represent the lower and upper envelope functions of the top and bottom cables, respectively. The third column shows the best grasp point (red) and direction (green) with a high n.

Algorithm 1: Optimal grasp point analysis

Input : Top cables lower envelope function $g_t(x)$; bottom cables upper envelope function $g_b(x)$; x axis evaluation interval L; number of analyzed points n

Output: Grasp point p_g; grasp angle θ_g

1 $d_t(x) \leftarrow g_t'(x)$
2 $d_b(x) \leftarrow g_b'(x)$
3 $max_dist \leftarrow 0$
4 **for** $k \leftarrow 0$ *to* n **do**
5 $\quad x_k \leftarrow L_{\min} + \frac{k(L_{\max}-L_{\min})}{n}$
6 $\quad$ **if** $g_t(x_k) > g_b(x_k)$ **then**
7 $\quad\quad g_{\text{avg}} \leftarrow {}^{(g_t(x_k)+g_b(x_k))}/_2$
8 $\quad\quad d_{\text{avg}} \leftarrow {}^{(d_t(x_k)+d_b(x_k))}/_2$
9 $\quad\quad p \leftarrow [x_k, g_{\text{avg}}]$
10 $\quad\quad \vec{a}_\perp \leftarrow d_{\text{avg}}\hat{i} - \hat{j}$
11 $\quad\quad h(x) \leftarrow$ Line with direction $\vec{a}_\perp$ that passes through p
12 $\quad\quad x_t \leftarrow \underset{x_c \in \{x \mid g_t(x)=h(x)\}}{\text{argmin}} \|[x_c, g_t(x_c)] - p\|$
$\quad\quad$ //Intersection of $h(x)$ and $g_t(x)$
13 $\quad\quad x_b \leftarrow \underset{x_c \in \{x \mid g_b(x)=h(x)\}}{\text{argmin}} \|[x_c, g_b(x_c)] - p\|$
$\quad\quad$ //Intersection of $h(x)$ and $g_b(x)$
14 $\quad\quad$ **if** $\|[x_t, g_t(x_t)] - [x_b, g_b(x_b)]\| > max_dist$ **then**
15 $\quad\quad\quad p_g \leftarrow [{}^{(x_t+x_b)}/_2, {}^{(g_t(x_t)+g_b(x_b))}/_2]$
16 $\quad\quad\quad \theta_g \leftarrow \frac{\pi}{2} + \arctan(\frac{-1}{d_{\text{avg}}})$
17 $\quad\quad\quad max_dist \leftarrow \|[x_t, g_t(x_t)] - [x_b, g_b(x_b)]\|$
18 $\quad\quad$ **end if**
19 $\quad$ **end if**
20 **end for**

used as the inputs to call the optimal grasp point analysis algorithm presented in Algorithm 1, which returns the pixel coordinates (p_g) and orientation (θ_g) of the optimal grasp point to separate the cables. This algorithm has two premises: (i) the grasp point direction (which is the target for the gripper roll axis direction) must be perpendicular to the cables' direction and within the image plane, and (ii) the grasp point must maximize the distance between the two envelope functions along the grasp point direction. To achieve this, the distance between $g_t(x)$ and $g_b(x)$ along the grasp point direction is calculated at n points, which are evenly distributed in the x axis along the L interval, selecting the one with the maximum distance. For each of these analyzed points, the x value is fixed (i.e., x_k), and the y value is calculated as the average between $g_t(x_k)$ and $g_b(x_k)$ (i.e., the middle point between the envelope functions). Regarding the grasp point direction, it is calculated as perpendicular to the average direction of the two envelope functions in that point x_k (as can be seen in Line 10), which is given by the derivative of these functions (i.e., $g_t'(x_k)$ and $g_b'(x_k)$).

This process can be graphically seen in the graphs of the second column of Fig. 6, where the black and red points represent the n analyzed potential grasp points (i.e., $[x_k, y_k]$ for $k \in [0, n]$) and the gray and green lines represent their direction. Additionally, the red point and green line represent the optimal grasp point and its direction, as it presents the maximum distance between the envelop functions along its grasp point direction. Observing the graphs in the third column of the figure, it can also be seen that the higher the number of points analyzed is, the more precise the result will be.

Finally, after this algorithm provides the pixel and direction of the optimal grasp point within the image, the last step is to transform it into a pose in the robot workspace. This is done using the top corner of the connector fixture as a reference point, as both its pixel coordinates and its real pose are known, and the pixel-to-millimeter ratio within the cables' plane (previously calculated). All this information is provided by the CAD Platform.

Although this methodology is robust and highly accurate, cable separation errors can still happen since, in some cases, the distance between them is lower than a millimeter. Therefore, the proposed system also includes a vision-based algorithm to assess the result of the separation operation. The initial steps of this algorithm are common with the pipeline presented in Fig. 5, until the estimation of the polynomials fitting the cables' shape. After this, the current pose of the actuation frame of the gripper that is supposed to be grasping the separated group of cables (i.e., the pose of its finger pads) is converted into pixel coordinates within the image. As in the previous case, this transformation is done by utilizing the top corner frame of the connector fixture as the reference pose and the pixel-to-millimeter ratio. Next, it is analyzed if the predicted points of any of the cables (P_i) do not reach the x axis coordinate of the previously calculated gripper pixel. In that case, its estimated polynomial is extended with

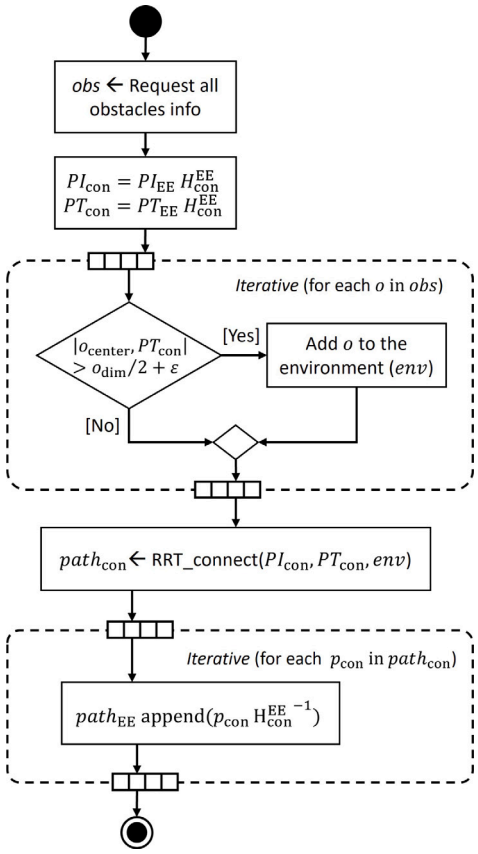


Fig. 7. UML activity diagram of the connector collision avoidance algorithm.

a straight line tangent to its final point. Finally, the algorithm checks which of the estimated cables pass closer than a threshold distance to the gripper actuation frame, considering them as grasped in case they do.

3.3. Tangle-free path planning

After picking each wire harness, the next step is to lock its main connector into a fixture. During the approach movement, the harness terminals hang freely as they move across the assembly platform. Given the length of the wire harness and the high density of guides and other components on the platform, there is a significant risk of the cables becoming entangled with platform elements during this movement. This issue could damage the cables and the platform components and, therefore, must be avoided. Thus, this function aims to find a path free of obstacles for the hanging harness while moving on top of the assembly platform, partially answering RQ2.

The steps of this algorithm are presented in Fig. 7. First, the pose and dimensions of all the obstacles of the assembly platform are requested to the CAD platform. Then, the initial (PI_{con}) and target (PT_{con}) poses of the hanging connectors are calculated from the end effector initial (PI_{EE}) and target (PT_{EE}) poses. This calculation is done using the homogeneous transformation matrix between the end effector and the hanging connectors (H_{con}^{EE}). Next, the obstacles environment is built, including only the obstacles further than a distance ε from PT_{con}. Then, a path free of obstacles from PI_{con} to PT_{con} is calculated using the RRT-connect algorithm [50]. Finally, the computed path for the hanging connectors is transformed into the path the end effector must follow.

3.4. System task-level programming

As mentioned in the Introduction (Section 1), the developed system must enable easy and rapid reprogramming to meet the highly volatile production demands of automotive wire harness manufacturers. Task-level programming offers an ideal solution, allowing complex tasks to be programmed as sequences of high-level, pre-programmed operations (or skills) [51]. The sequence of operations required to complete the task can be either planned automatically [52,53] or specified by the operator [54,55]. This approach separates programming responsibilities between the robot programmer, who develops the robot skills, and the task specialist, who combines these skills to achieve the task's goal without needing robotic expertise [56]. For instance, in [57], assembly tasks are programmed as a sequence of a set of predefined parameterized skills, such as pick, place, insert, or quality inspection.

Building on this approach, the developed system adopts a task-level programming methodology, which answers the final research question (RQ3). This enables the user to program the system by simply specifying the high-level plan of the assembly process and providing a set of files that describe the robotic platform layout and the manipulated objects, as explained in Section 3.1. To achieve this, the interaction between all the modules of the system is necessary. This interaction during the initialization and control of the robotic assembly process is shown in Fig. 8. As can be seen, the system is fully controlled from the UI, which is used to launch the modules and start and stop the robot execution. Upon initialization, the task planner requests all the necessary information from the CAD Platform, which is used to generate a detailed action plan for the robot, as can be seen in Fig. 10. This plan contains the sequence of robot operations, the cable groups involved in each of them, and the pose and dimensions of the platform components where the operations must be performed.

Once initialization is complete, the assembly process can be started. This begins by calling a ROS action server, which sequentially executes the robot operations defined in the action plan. Each operation is configured with the relevant information about its associated locations and cable groups, allowing it to generate optimal robot trajectories. In some cases, these operations require support from the perception and motion planning modules. For instance, the cable separation operation requests input from the perception module to determine the optimal grasp point for separating cable groups, while the routing operation relies on the motion planning module to compute robot trajectories that involve dual-arm synchronization, cartesian speed control, or force control. The process concludes when all operations are completed or when the operator stops or pauses it from the user interface. In the latter cases, the action server is preempted, with the current operation index either reset (for stopping) or stored (for pausing), allowing the process to be resumed later from the stored index.

The proposed task-level programming approach seeks to simplify the system programming as much as possible, enabling the user to define the process sequence with a very small and high-level list of actions. This list is composed of only four possible actions: (i) pick a wire harness from the cable-hanging platform, (ii) fixture the main wire harness connector, (iii) route a cable group through a set of guides, and (iv) tape together the cables between two routing guides. To achieve this, the high abstraction level of this input plan is reduced gradually, increasing the level of detail at various stages until the robot trajectories and end effector actions are fully defined, as shown in Fig. 9.

The first step is the generation of the robot action plan, which provides a more detailed sequence of the operations the robot must perform. These operations are predefined as parameterized functions, which are configured with information about the locations and the handled objects. Seven robot operations were developed for this process: pick wire harness (WH), fixture connector, grasp, separate cables, take separated, route, and tape. The pipeline for transforming the high-level process plan into the robot action plan is depicted in Fig. 10. As can be seen, the main contribution of this step is the increase in the level of detail of the cable routing actions. The structure of the wire harnesses is analyzed to determine when cable groups need to be separated,

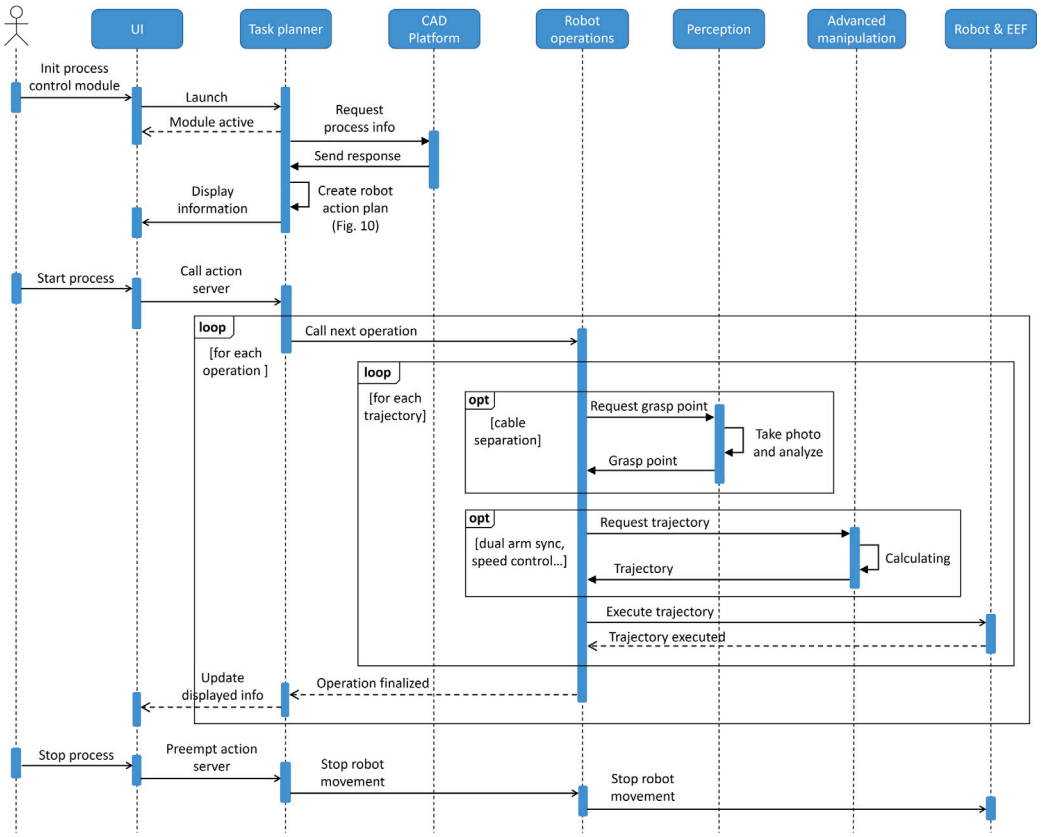


Fig. 8. UML sequence diagram of the system modules interaction for performing a wire harness assembly process.

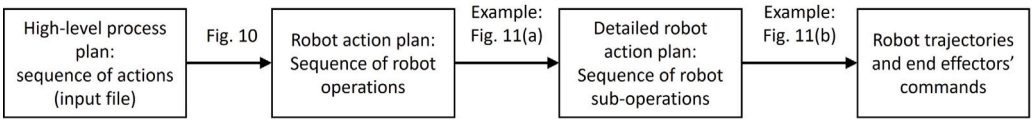


Fig. 9. Action plan progressive abstraction level reduction.

and these operations are introduced transparently for the user, along with other related operations (e.g., grasp or take separated) to ensure plan consistency. Additionally, detailed information about the handled elements and the platform components is gathered during this process to properly configure the operations.

The next step applies to complex robot operations with high variability, which are decomposed into smaller sub-operations. These sub-operations act as the building blocks that allow the generation of complex and highly configurable operations with minimal effort. In this case, this decomposition only applies to the cable routing operation. As can be seen in Fig. 11(a), the structure of this operation strongly depends heavily on the disposition of the involved routing guides. Thus, for each guide, there can be four different routing scenarios:

(i) The current guide is not aligned with the direction of the routed cables (i.e., the direction from the last guide to the current one). The alignment condition is met when the angle difference between them is below a threshold value (ε_θ). In this case, a pivoting sub-operation is needed, where the second robotic arm performs a coordinated movement to rotate the cables around it, aligning them with the guide for insertion. After this, the cables are routed toward the top of the next guide.

(ii) The next guide is aligned with the direction of the cables, but not with the current guide (i.e., their orientations differ more than ε_θ). In this scenario, the cables must first be inserted into the current guide, then rotated around it to point toward the next guide, and finally routed to its top.

(iii) The current guide is aligned with both the cable direction and the next guide, and the distance between the current and next guide exceeds a threshold distance (ε_d). Here, the cables are inserted into the current guide and routed to the top of the next guide.

(iv) Similar to the previous scenario, but the distance between the current and next guide is less than ε_d. In this case, since the guides are close together, the cables can be inserted into both guides simultaneously, saving time and reducing the risk of collisions. The cables are then just routed to the top of the next guide.

Additionally, special considerations are made for the initial and final guides of the routing sequence.

Finally, the last step of the proposed task-level programming system converts the previously selected operations and sub-operations into a sequence of robot trajectories, force-controlled movements, tool changes, and end effector actions. All these trajectories and actions are optimized based on the operation's input parameters, collected in the previous phases, and the information provided by the perception module. Three examples of this are discussed below, for the cable separation and connector fixturing operations, and for the cable routing sub-operations.

Fig. 11(b) shows the pipeline of the cable separation operation. This operation always happens after a connector fixturing operation, as the wire harness has just been locked so the cables are less likely to be entangled. Hence, the first action is to open the gripper, release the cables, and retract the robotic arm. Then,After releasing the cables and

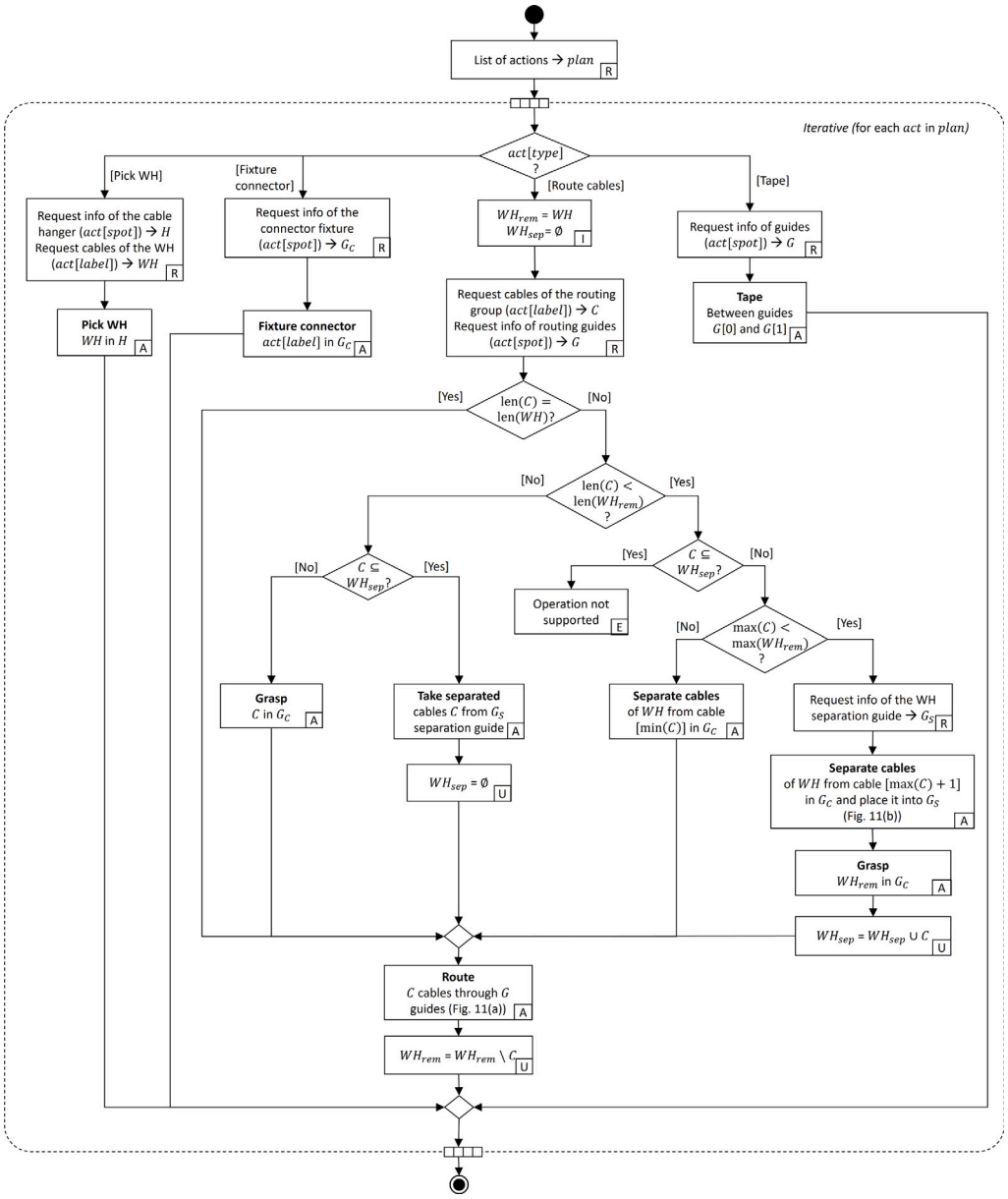


Fig. 10. UML activity diagram of the robotic action plan generation process. The letter in the bottom right corner specifies the purpose of each block (R: Request information from CAD Platform, I: Initialize a variable, A: Append information to the robotic action plan, U: Update a variable, E: Error). Notation clarifications: WH (list of cables of the handled wire harness), C (list of cables of the current routing group), WH_{sep} (list of cables inside the separation guide), WH_{rem} (list of remaining cables to be routed).

retracting the arm, the camera located in front of the connector fixture captures an image of the cables and analyzes it to determine the optimal grasp pose for separating the required cable group (using the algorithm depicted in Fig. 5). If the perception algorithm is able to find a grasp point the gripper moves to that pose and grasps the upper group. Otherwise, the resolution of the image used for its analysis is increased and the grasp pose computation is repeated. This approach maximizes the system's speed without sacrificing its robustness. After grasping the upper cable group, the robotic arm moves up and the camera takes a new picture. The image is analyzed, in this case with a higher resolution, and if an error in the separation is detected, the operator is called. Finally, if the separation was successful, the operation is either finalized, in case the upper cable group must be routed first, or inserted in a dedicated separation guide otherwise.

Regarding the connector fixturing operation, the first step is to determine a path free of obstacles toward the target fixture to avoid entanglements of the hanging cables and connectors with the assembly platform components. This path is provided by the motion planning module, using the wire harness tangle-free path planning algorithm presented in Fig. 7. After following this path, the connector is inserted into the fixture with a slow speed-controlled movement. Finally, the cables are pulled forward with a force-controlled movement until the connector is locked.

Finally, it is worth mentioning the cable routing sub-operations, which, in addition to computing the end effectors' trajectories, pay special attention to the actuation of the grippers. Thus, for each trajectory, the grasping mode must be specified as either sliding or pulling. In sliding mode, the gripper gently holds the cables within the small opening formed between its upper and lower nails. The grip is secure enough to keep the cables in place without restricting their movement, allowing them to slide freely as needed. This mode is used in sub-operations that involve guiding the cables, such as "lift to guide" or "route top". In pulling mode, the gripper firmly grasps the cables with the silicone pads located between its upper and lower nails.

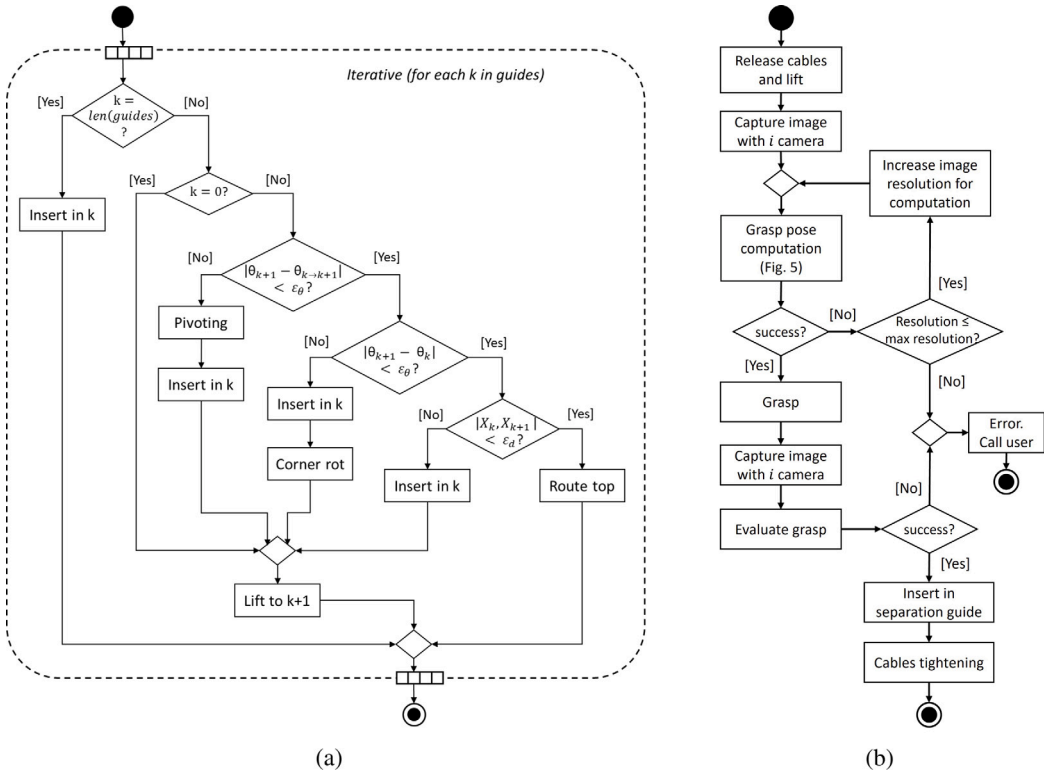


Fig. 11. UML activity diagram of the cable routing (a) and cable separation (b) robot operations. X_k and θ_k represent the position and orientation angle of the guide k, respectively.

Therefore, this grasping mode is employed in sub-operations that require applying or maintaining cable tension, such as "insert in guide", "pivoting", or "corner rotation". This firm grasp is often combined with force-controlled movements to ensure the cables are tight without damage.

Regarding the robotic arms trajectories, apart from the aforementioned force-controlled movements for tightening the cables, the rest consist of speed-controlled single or dual-arm movements. These movements follow either linear or circular paths, determined based on the points of interest of the guides and various offset values. Linear paths are used to slide the cables loosely toward the guides with the routing arm and to move the secondary arm to assist during the routing. The secondary arm's tasks include grasping the cables near the last routed guide to prevent them from slipping out beforewhen the routing arm is lifted toward the next guide, and pushing down the cables after insertion to ensure they are fully seated in the guide. Circular movements, on the other hand, are primarily used for inserting the cables into the guides while maintaining constant tension by keeping a consistent distance between the cable grasping points (i.e., maintaining a fixed radius). In these cases, the rotation axis must be perpendicular to the cable direction and the assembly platform's normal vector, passing through the position where the secondary gripper is holding the cables. Additionally, dual-arm coordinated circular trajectories are employed when it is necessary to pivot the cables around the secondary gripper to align them with the guide before insertion.

4. Experimental evaluation

4.1. Setup description

The proposed system was implemented and tested with the robotic platform shown in Fig. 12. As can be seen, thanks to the advanced manipulation capabilities of the system, there was no need to overengineer the experimental setup to suit the robot. Instead, the robot was able to operate effectively within a simple, human-centric design. To enable this, a dual-arm industrial robot was utilized (Motoman SDA 10F). Regarding the end effectors, two parallel grippers (Weiss WSG50) fitted with specialized fingers to either slide or firmly grasp the cables, and an automated taping gun were utilized. These end-effectors could be switched out by means of a pneumatic ATC. Additionally, the robot wrists were equipped with force/torque sensors (Nordbo NRS-6050-D50), which were used to measure the tension of the manipulated cables.

The experimental setup included also three physical platforms the robot could interact with to perform the assembly process: the *ATC platform*, used by the robot to dynamically change the utilized end effectors; the *cable-hanging platform*, from where individual wire harnesses are picked initially —avoiding complications such as entanglement; and the *assembly platform*, where the cables of the wire harnesses are routed and assembled. A detailed view of the assembly platform can be observed in Fig. 13. As illustrated, it consists of a modular station composed of a breadboard where multiple components can be mounted. These components include various connector head fixtures, separation and routing guides, and cameras. The connector head fixtures of wire harnesses with multiple cable groups (the Lumberg connector fixtures in this case) include a flexible step that mechanically locks the connector after insertion. This preempts the connectors from moving and allows to perform cable separation. Regarding the guides, routing guides have a flexible top that facilitates the insertion of cables while reducing their chances of accidentally slipping out, whereas separation guides have a wide opening as they aim to provide a tangle-free zone for the temporary storage of cable sub-branches. Finally, the cable separation cameras are compact 2D RGB cameras (OAK-1), and they are placed aligned with the connector fixtures and perpendicular to the cables' direction. These cameras capture the images used for the cable separation pose computation.

The entire system is controlled by a desktop computer with the following specifications: Intel core i7 12th generation processor, NVIDIA

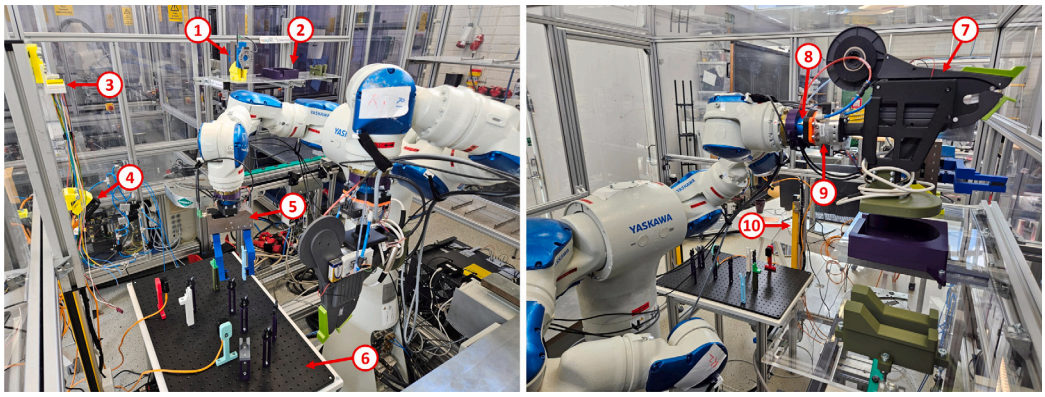


Fig. 12. Robotic platform setup. 1: Left gripper, 2: ATC platform, 3: Cable-hanging platform, 4: Global view camera, 5: Right gripper, 6: Assembly platform, 7: Automatic taping gun, 8: Force sensor, 9: Tool changer, 10: Safety devices (light curtain).

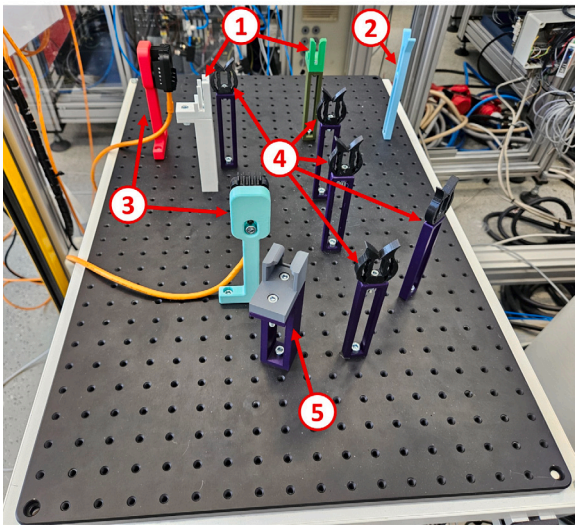


Fig. 13. Assembly platform. 1: Lumberg connector fixture (WH1-2), 2: Separation guide, 3: Cables separation camera, 4: Routing guide, 5: Molex connector fixture (WH3).

GeForce RTX 3080 GPU, and 32 GB RAM. This computer runs the ROS Noetic system and interfaces the robot controller, an Omron safety PLC, and the multiple sensors and actuators of the cell.

Three types of wire harnesses (WH1, WH2, and WH3) are assembled in the aforementioned setup, which are shown in Fig. 14. These harness models were provided by the automotive wire harness manufacturing company ELVEZ [10] and are currently among the products being manually assembled in their factory. WH1 consists of two cable groups, one with two cables and the other with eight, featuring cables in six different colors. It has a main Lumberg connector. The individual cables have a diameter of 1.25 mm and lengths averaging 560 mm. WH2 consists of two cable groups, one with three long cables and the other with eight short cables, featuring cables in eight different colors. It has a main Lumberg connector and the diameter of the wires is also 1.25 mm. Regarding their lengths, on average it is 475 mm for the longer group and 145 mm for the shorter one. WH3 contains a single cable group composed of six cables of five different colors, and it has a main Molex connector. The individual cables are slightly thicker than in the previous harnesses, with a diameter of 2 mm, and they have an average length of 400 mm.

Using these three wire harnesses, To demonstrate the industrial applicability of the proposed system, two different testing scenarios using these three wire harnesses are formulated, both of them representative of a real automotive wire harness manufacturing industry.

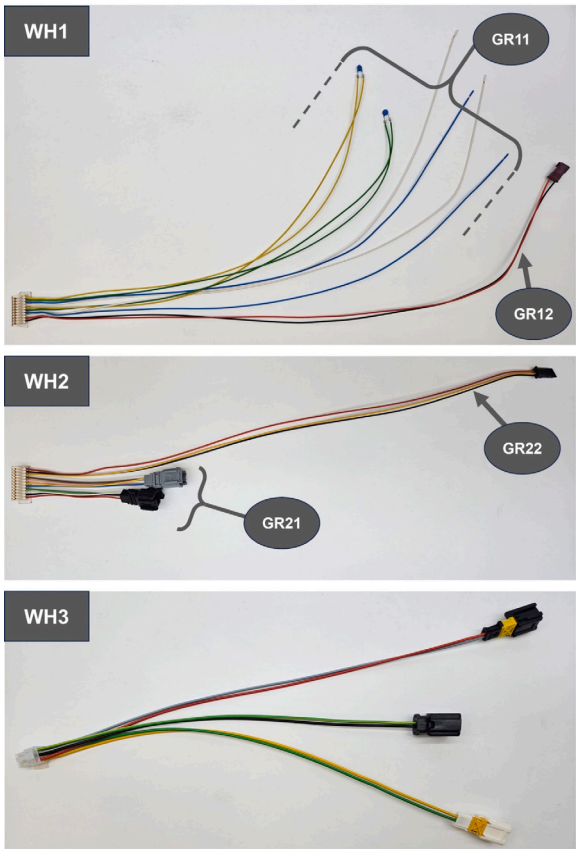


Fig. 14. Wire harnesses used for the assembly and their cable groups.

Specifically, both testing scenarios have been inspired by real assembly processes performed at the ELVEZ company [10]. The reduction of the assembly process time is not part of the scope of this study; therefore, the experiments are performed at low speed. Their implementation and the corresponding results are documented in Sections 4.2 and 4.3. A video of the robot performing the assemblies of both testing scenarios is available at this link.[2]

[2] https://youtu.be/v6aONASYWL4

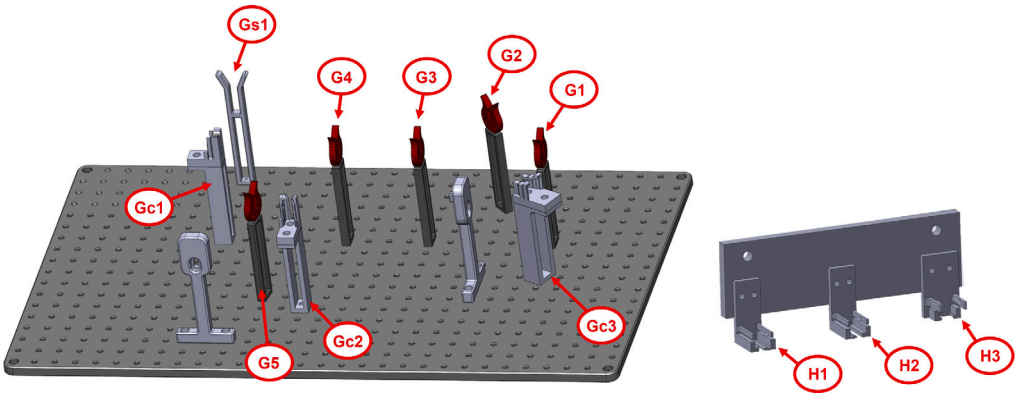


Fig. 15. Assembly platform (left) and cable-hanging platform (right) layout for testing scenario I.

Table 3
Testing scenario I action plan. Input sequence of actions provided by the user (left), and resultant sequence of robot operations and sub-operations generated by the task-planner module (right). Operation types abbreviations: P (Pick WH), FC (Fixture connector), RC (Route cables), SC (Separate cables), G (Grasp), L (Lift), I (Insert), TS (Take separated), P (Pivoting), C (Corner rotation). Locations are depicted in Fig. 15.

Input actions (type-component-location)	Robot operations/sub-operations (type-component[a]-location)
P-WH1-H1	P-WH1-H1
FC-WH1-Gc1	FC-WH1-Gc1
RC-GR11-G4,G3,G2	SC-GR12-Gc1,Gs1; G-GR11-Gc1; L-G4; I-G4; L-G3; I-G3; C-G2; L-G2; I-G2
RC-GR12-G4,G3,G1	TS-GR12-Gs1; L-G4; I-G4; L-G3; I-G3; L-G1; I-G1
P-WH2-H2	P-WH2-H2
FC-WH2-Gc2	FC-WH2-Gc2
RC-GR21-G5	SC-GR21-Gc2,Gs2; L,G5; I,G5
RC-GR22-G3,G1	G-GR22-Gc2; L-G3; P-G3; I-G3; L-G1; I-G1
P-WH3-H3	P-WH3-H3
FC-WH3-Gc3	FC-WH3-Gc3
RC-WH3-G3,G4	L-G3; P-G3; I-G3; L-G4; I-G4

[a] The handled component is not specified when it is already grasped.

4.2. Testing scenario I: Wire harnesses routing

This scenario aims to demonstrate the ability of the proposed system to arrange several wire harnesses according to a certain distribution. To achieve this, the robot must be able to pick the involved wire harnesses, fixture their connectors, separate their cable groups (if needed), and route them following the specified paths. To prove the robustness and adaptability of the system, this testing scenario was designed to entail a high diversity of robot operations. Thus, three different wire harnesses were utilized (WH1, WH2, and WH3, shown in Fig. 14), with a total of five cable groups, requiring two cable separation operations and five different sequences of cable routing sub-operations. The layout of the cable-hanging and assembly platforms as well as the input sequence of actions of this testing scenario are depicted in Fig. 15 and Table 3 respectively.

Besides the global assessment of the robotic system performance for arranging wire harnesses, this testing scenario also allowed the individual evaluation of different system capabilities. In particular, special attention was paid to the ability of the robot to fixture the wire harnesses' connectors avoiding entanglements with the assembly platform, the effectiveness of the force-controlled cable tightening approach, and the performance of the vision-based cable separation algorithm.

Regarding the first point, the system proved its ability to find collision-free paths for the hanging connectors toward the target connector fixtures. This is illustrated in Fig. 16, which shows the computed paths for the end effector (dashed black line) and the hanging connectors (red line) for the three wire harnesses. As can be seen, the connector's path is free of obstacles in all cases, theoretically guaranteeing safe movement without entanglements. However, even though the computed paths were always successful, in practice it was observed that entanglements could still occur in wire harnesses with many hanging cables and connectors, such as WH1, as the projected area enclosing the hanging elements is too big to be considered a point for the calculations. This issue was addressed by increasing the minimum distance to obstacles allowed for the computed path. Nevertheless, this issue will be studied carefully in future works.

Regarding the force-controlled cable tightening, the proposed solution has proven to be effective and harmless to the wire harnesses in the tests performed. This can be observed in Fig. 17, which shows the evolution of the forces exerted on the robot wrist during cable insertion and connector fixturing operations. Analyzing the cable insertion graph (left), it can be seen that around t = 1 s the cable is pulled until a certain threshold force is achieved. At this point, the grasped cables are considered tight and the robot can proceed with their insertion. The insertion starts around t = 5 s and it finishes approximately at t = 9 s. During this time, it can be seen that the pulling force (F_{xy}) is kept more or less constant thanks to the circular trajectory of the end effector. Moreover, it is noticeable that the vertical force (F_z) decreases during the insertion. This is the effect of, first, the insertion resistance offered by the flexible material of the routing guide (first peak at t = 7 s) and then, the robot pushing the cables against the guide (second peak at t = 8 s). On the other hand, the right graph shows how the cables are pulled until the connector is locked inside the fixture (when a certain threshold force is reached).

Finally, the performance of the proposed cable separation solution is discussed. As can be seen in Fig. 18, the methodology presented in Fig. 5 is used to determine the grasp point analysis limits (top row), estimate the analyzed cables shape with polynomials (second row), calculate the upper and lower envelope functions (third row), and compute the optimal grasp point and orientation (third and fourth rows). This methodology was employed to separate the cable groups of both WH1 and WH2 with a success rate of 84% and 80% respectively, after 50 experiments for each of them. The failures were mainly caused by two reasons: imprecise cables shape estimation due to considerable self-occlusions, and minor calibration issues for transforming the calculated grasp point from pixel coordinates to a pose in the workspace. Examples of both issues can be observed in the first column images of Fig. 18. On the one hand, self-occlusions from pixel 500 onwards (in the horizontal axis) occasioned inaccuracies in the estimation of the cables' shape polynomial functions, however, in this case, this did not affect the grasp point calculation as it happened mostly outside the analyzed area. On the other hand, as can be seen in the bottom image, even though the computed grasp point and orientation are correct, the distance between the cable groups to be separated is so small that a minor

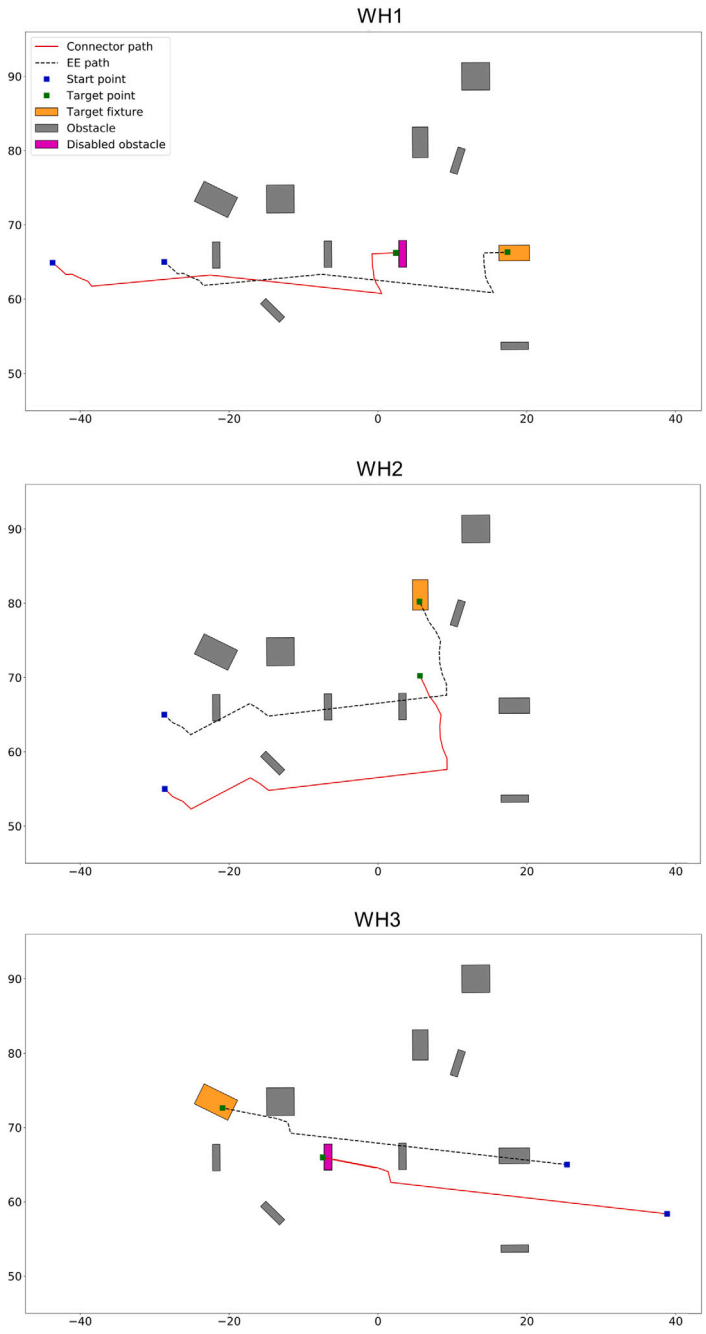


Fig. 16. Path planning algorithm for connector collision avoidance applied for the three wire harnesses.

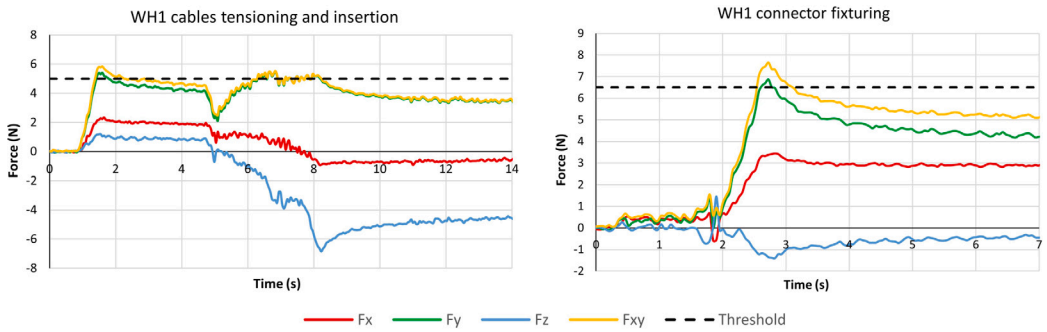


Fig. 17. Force sensor readings at different phases of WH1 manipulation. The force readings were tared right before the beginning of each graph.

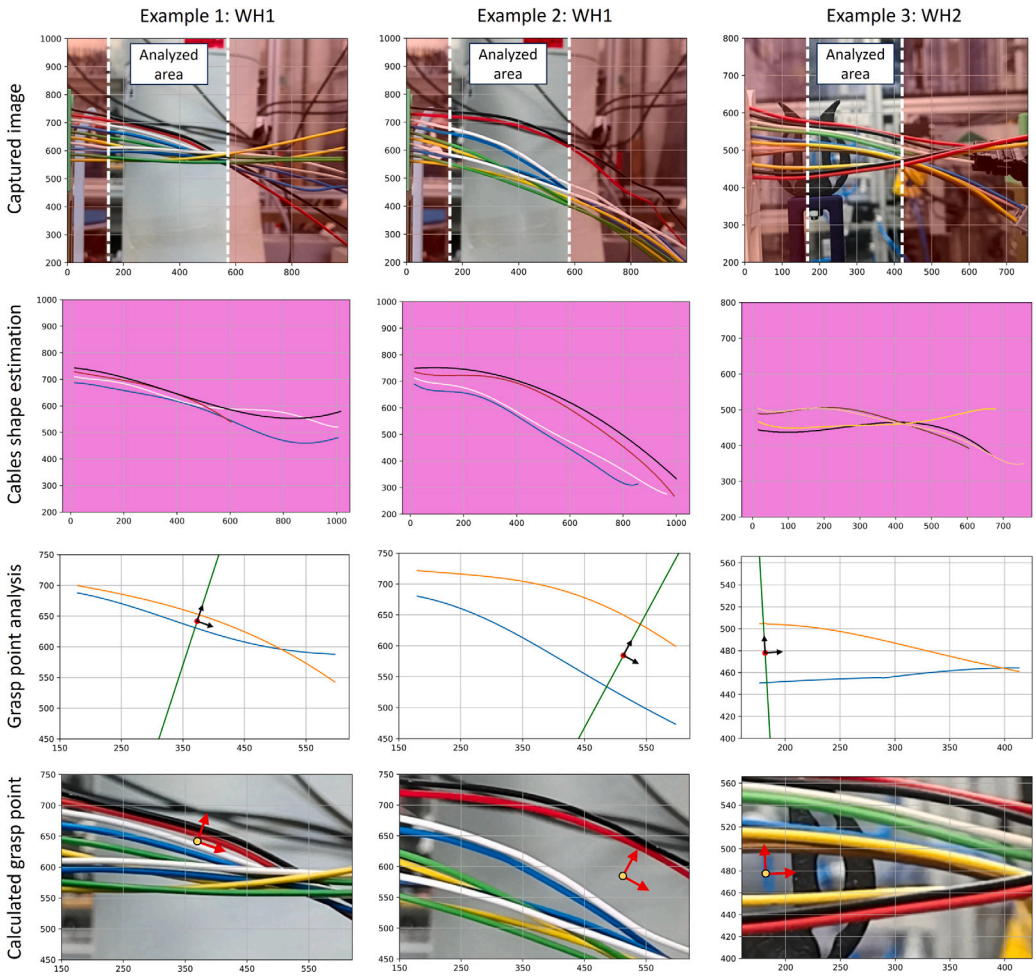


Fig. 18. Examples of optimal grasp point computation. For WH1 the aim is to separate the two top cables from the rest, while for WH2 the aim is to separate the three bottom cables from the rest. The axes show pixel values.

calibration error could lead to a failure. Nevertheless, in this case, the cable separation was still successful, demonstrating the robustness of the proposed method. As a reference, the only similar solution in the state-of-the-art [28] reported an average error of 3.4 mm in the wire harness cables shape estimation. Although this level of accuracy was sufficient for their setup, where the harnesses were hanging and the branches were spaced further apart, it would be insufficient for some of the denser cable arrangements addressed in this study.

Despite the high success rate achieved for cable separation, it is not enough to allow the system to operate autonomously. Therefore, after the separation, the result is captured and evaluated, as can be seen in Fig. 19. Currently, this was only implemented to test the performance of the cable separation assessment algorithm, however, in future works, it could be used to launch corrective actions when wrong separations are detected. This algorithm uses a higher image resolution for analysis than the grasp point estimation one to increase its robustness (at the expense of a higher computation time). Its performance was tested on WH1 with a total of 100 cable separation images—half of them successful and the other half including errors. Some of these errors were manually introduced to reach the desired number of samples. The obtained results were positive, detecting 94% of the separation errors. However, 14% of the successful grasps were incorrectly classified due to self-occlusions within the grasped cable group. This means that there is still work to do before the proposed system can be claimed as a fully autonomous solution for wire harness assembly, which will be the target of our future research.

The performance of all these aspects of the system was also analyzed holistically during the execution of the testing scenario (following the action plan specified in Table 3). To evaluate it, each of the three wire harnesses involved was routed 30 times according to the plan, and the results were classified as successful or not. Furthermore, the failure cases were categorized into six groups based on their cause: connector fixturing (when the connector does not get completely locked into the fixture), cables separation (when the cable separation is not successful, without considering the cable separation evaluation algorithm), cables insertion (when not all the grasped cables are inserted properly into a routing guide), entanglements (when the wire harness gets entangled with other cables or with any element of the assembly platform), trajectory generation (when the trajectory generated by the system is wrong), and other (encompassing any remaining failure scenarios).

The results are presented in Table 4, from which several conclusions can be extracted. First, the primary cause of failure was cables separation, accounting for 61% of all errors across the three wire harnesses.

Table 4
Results of testing scenario I. Failure causes abbreviations: CF (connector fixturing), CS (Cables separation), CI (Cables insertion), E (Entanglements), TG (Trajectory generation).

WH	Success	Failure					
		CF	CS[a]	CI	E	TG	Other
WH1	21/30	0/30	5/30	1/30	3/30	0/30	0/30
WH2	22/30	1/30	6/30	0/30	0/30	0/30	1/30
WH3	29/30	0/30	0/30	0/30	1/30	0/30	0/30

[a] The cables separation evaluation algorithm was not considered for these experiments.

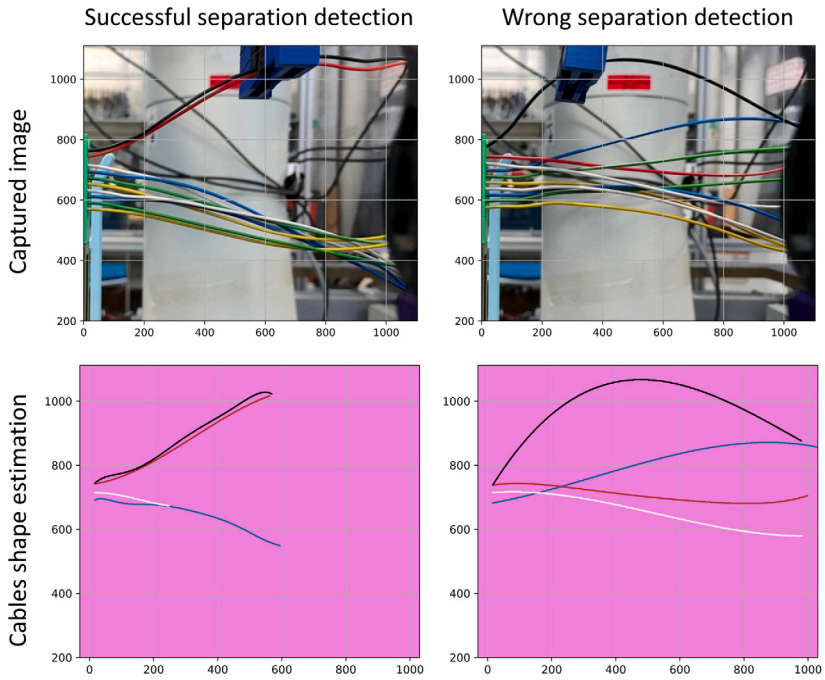


Fig. 19. Examples of cable separation evaluation for WH1. The aim is to separate the black and red cables from the rest. The axes show pixel values.

However, as previously mentioned, the cable separation evaluation algorithm was not used during these experiments. This suggests that the implementation of a recovery action for the detected errors could considerably increase the overall success rate of the system. The second major cause of failure were the entanglements, mainly of WH1, which could be expected due to its multiple long cables. As for the remaining failure causes, only a few errors occurred during connector fixturing and cables insertion operations, demonstrating the effectiveness of the force-controlled tightening approach; and no errors were due to wrong trajectory generation. Moreover, only one failure was classified in the "other" category, which happened during a pivoting sub-operation, in which the secondary arm missed the grasp of the cables. Finally, it is striking that the success rate of WH3 is considerably higher than the other two. However, this is not something surprising, as it had the simplest route, with only one cable group and no cable separations.

Finally, the entire assembly sequence was executed 20 times, achieving a successful result in 11 of them. The result of one of the successful experiments is shown in Fig. 20, and a video of the entire process can be seen at the link provided at the end of Section 4.1 (check 1:30–8:15). Unfortunately, these results cannot be compared with any of the state-of-the-art solutions, as none of them have reported a complete assembly of more than one multi-branch wire harness. However, these numbers can serve as a baseline for future studies. As reference, the most similar approach in the state-of-the-art [32] reported a success rate of 80% for an assembly process of three wire harnesses. However, several key differences and limitations should be considered for comparison. First, that approach employed a simpler yet less generalizable method for cable separation, where the separation point was computed based on the known connector position and the predefined relative positions of each cable. This restricts its applicability to structured scenarios, as the cables must be horizontally and neatly arranged to provide reliable information. As shown in Table 4, over 60% of the errors in our experiments originated from the challenging vision-based cable separation operation. Excluding these cases, the success rate would likely reach around 80%. Second, the solution in [32] does not support dual-arm manipulation, which would be necessary for layouts involving large directional changes that require pivoting the cables. Finally, the approach was validated on a single assembly scenario, so its reconfigurability has not been demonstrated.

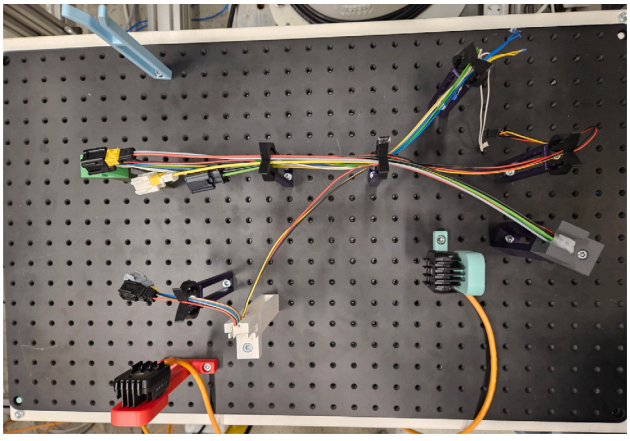


Fig. 20. Testing scenario I successful result.

4.3. Testing scenario II: Taping and system reconfiguration

This second testing scenario has two purposes: to evaluate the performance of the robotic taping operation and to demonstrate the ability of the system to automatically reconfigure for a new setup and assembly sequence. Hence, this scenario presents a new layout for both the assembly and cable-hanging platforms (see Fig. 21), involves only two wire harnesses (WH1 and WH3), and includes one taping operation. Additionally, the scenario has been designed to cover a high variety of routing sub-operations as can be seen in the action plan at Table 5. Finally, the ATC feature of the system is also tested, as the robot needs to change one of its grippers for the automatic taping gun in order to perform taping.

A preliminary taping experiment was conducted before designing this testing scenario to ensure its feasibility. In this experiment, the minimum and maximum distances between routing guides where taping could be performed were empirically determined. To do this, five WH1 taping attempts were executed at each distance (Δx_{guides}), starting from 50 mm and increasing until reaching failure in steps of 25 mm.

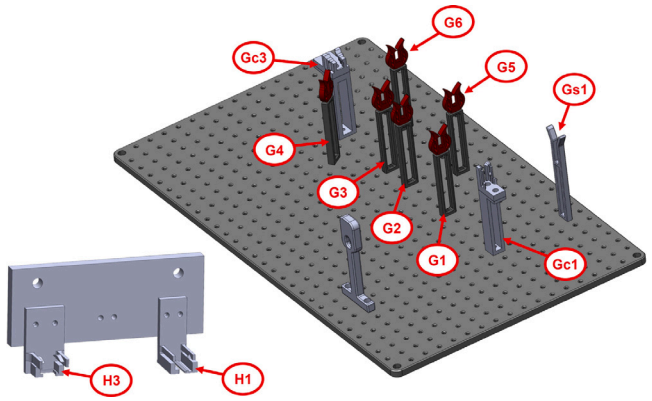


Fig. 21. Cable-hanging platform (left) and assembly platform (right) layout for testing scenario II.

Table 5
Testing scenario II action plan. Input sequence of actions provided by the user (left), and resultant sequence of robot operations and sub-operations generated by the task-planner module (right). Operation types abbreviations: P (Pick WH), FC (Fixture connector), RC (Route cables), T (Tape), SC (Separate cables), G (Grasp), L (Lift), I (Insert), RT (Route top), TS (Take separated), P (Pivoting), C (Corner rotation). Locations are depicted in Fig. 21.

Input actions (type-component*-location)	Robot operations/sub-operations (type-component[a]-location)
P-WH1-H1	P-WH1-H1
FC-WH1-Gc1	FC-WH1-Gc1
RC-GR11-G1,G2,G3,G4	SC-GR12-Gc1,Gs1; G-GR11-Gc1; L-G1; I-G1; L-G2; RT-G3; I-G3; C-G4; L-G4; I-G4
RC-GR12-G1,G5,G6	TS-GR12-Gs1; L-G1; I-G1; L-G5; P-G5; I-G5; L-G6; I-G6
P-WH3-H3	P-WH3-H3
FC-WH3-Gc3	FC-WH3-Gc3
RC-WH3-G3,G2,G1	L-G3; RT-G2; I-G2; L-G1; I-G1
T-G2,G1	T-G2,G1

[a] The handled component is not specified when it is already grasped or when it is not relevant.

Table 6
Results of the robotic taping experiments. Distances are in mm.

Δx_{guides}	$\Delta z_{\text{tape}_{\text{avg}}}$	$\Delta z_{\text{tape}_{\text{max}}}$	Success
50	–	–	0/0
75	18	21	5/5
100	25	27	5/5
125	–	> 40	2/5

Besides the success or failure of each attempt, the vertical displacement of the cables (Δz_{tape}) was also collected (see Fig. 22). Table 6 presents a summary of the results. A distance is only considered valid for taping if all its attempts were successful and the maximum displacement of the cables was lower than 30 mm. Based on this, the resultant minimum and maximum guides distance for taping were 75 mm and 100 mm respectively. Thus, to comply with this, in testing scenario II, taping is performed between guides G1 and G2, which are separated by 100 mm.

Once this was determined, the complete assembly was attempted 30 times, with successful results in 22 of them. As in testing scenario I, most of the failures were due to cable separation errors, for which the cable separation evaluation algorithm was not considered. Additionally, no errors occurred during the ATC or the taping operation. Regarding the reconfigurability, no errors were caused by wrong trajectory generation, which indicates a perfect reconfiguration of the system by simply updating the CAD Platform input files. A video of the entire assembly process can be seen at the link provided at the end

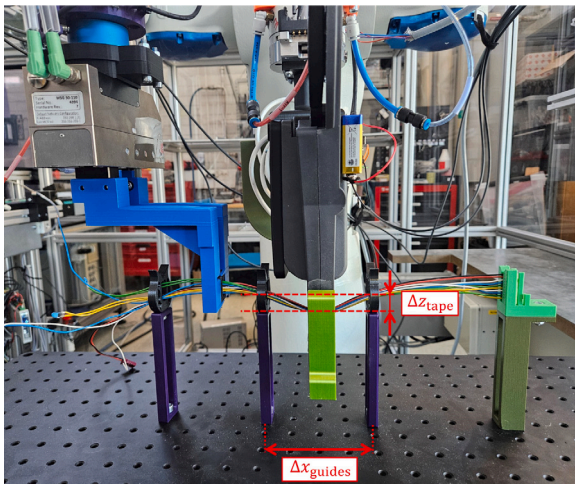


Fig. 22. Robotic taping preliminary experiment.



Fig. 23. Testing scenario II successful result.

of Section 4.1 (check 8:15–11:55). Furthermore, Fig. 23 shows the final result of one of the successful experiments.

5. Conclusions and future directions

This paper proposes a robotic system for the automatic assembly of multi-branch wire harnesses. The system implements a task-level programming methodology that enables its fast reconfiguration without the need for robotic expertise. To achieve this, the system counts with a list of predefined configurable basic robot operations, and a task planner module, which can decide how to combine and configure these operations to perform complex assembly tasks. To make this decision, the task planner module needs information about the process, the manipulated products, and the physical layout of the assembly platform. This information is provided by another module called CAD Platform, which extracts, processes, and combines relevant data from various configuration files, including CAD 3D models. This way, the system can be fully reprogrammed by simply updating these input files. Moreover, the system incorporates a set of advanced motion planning and perception capabilities, which can be utilized by the predefined robot operations to deal with the unpredictable behavior of DLOs. In particular, two novel capabilities were proposed for addressing two common multi-branch wire harness challenges: a computer vision technique to identify the optimal grasp pose to separate wire harness cable groups, and a path-finding algorithm to avoid entanglements of the wire harness with the assembly platform.

The system's functionality was validated in two real assembly scenarios involving three and two wire harnesses, respectively. These

scenarios were designed to cover the entire range of robot operations and utilize all the manipulation and perception capabilities of the system. Several key conclusions were drawn from the experiments. First, the task planner's selection of the robot operations, the generation of robot trajectories, and the reconfiguration of the system between scenarios were successful in all the tests carried out, demonstrating the efficacy of the CAD Platform module and the task-level programming approach. Second, cable separation proved to be the most challenging operation, and the primary source of errors, with an average success rate of 82%. However, not all these errors were due to incorrect grasp pose computation. Many were caused by minor calibration inaccuracies, as the distance between the cables to be separated was sometimes less than 1 mm. Third, a few failure cases were occasioned by cable entanglements with the assembly platform or other cables, highlighting the effectiveness of the entanglement avoidance algorithm, though there is still room for improvement. Lastly, despite the high success rates of individual assembly operations and capabilities, the cumulative effect of minor issues over the course of a long assembly process resulted in overall lower success rates, with 55% and 73% recorded for the two scenarios. This highlights that performing the entire assembly process as a continuous task is significantly more challenging than executing individual operations in isolation. It underscores one of the main differences and added complexities compared to the majority of the state of the art, which typically addresses only isolated steps under controlled initial conditions.

These results show the immense potential of the proposed system; however, it is still not robust enough to operate as a fully autonomous solution. To accomplish this, some limitations of the current system and key areas of future research have been identified from interpreting the experimental outcomes. First, the success rate of cable separation needs improvement. This could be achieved by refining different aspects of the current solution (e.g., algorithm, lighting, calibration) or by implementing corrective actions to autonomously address failure cases. Second, the proposed entanglement avoidance algorithm relies exclusively on the known position of the assembly platform obstacles and the expected, but uncertain, position of the hanging cables. To deal with these uncertainties, the use of sensor feedback and online trajectory planning should be explored. Finally, the development of online error detection techniques and recovery actions would be essential for the system to achieve full autonomy.

CRediT authorship contribution statement

Pablo Malvido Fresnillo: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Saigopal Vasudevan:** Writing – review & editing, Writing – original draft, Validation, Formal analysis. **Wael M. Mohammed:** Writing – review & editing, Formal analysis. **Jose A. Perez Garcia:** Supervision, Conceptualization. **Jose L. Martinez Lastra:** Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Pablo Malvido Fresnillo reports financial support was provided by Horizon 2020. Saigopal Vasudevan reports financial support was provided by Horizon 2020. Jose Luis Martinez Lastra reports financial support was provided by Horizon 2020. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement n° 870133, correspondent to the project entitled REMODEL, Robotic tEchnologies for the Manipulation of cOmplex DeformablE Linear objects.

References

[1] Wang H, Salunkhe O, Quadrini W, Lämkull D, Ore F, Johansson B, Stahre J. Overview of computer vision techniques in robotized wire harness assembly: Current state and future opportunities. Procedia CIRP 2023;120:1071–6. http://dx.doi.org/10.1016/j.procir.2023.09.127.

[2] Aguirre E, Ferrière L, Raucent B. Robotic assembly of wire harnesses: Economic and technical justification. J Manuf Syst 1997;16(3):220–31. http://dx.doi.org/10.1016/S0278-6125(97)88890-5.

[3] Trommnau J, Kühnle J, Siegert J, Inderka R, Bauernhansl T. Overview of the state of the art in the production process of automotive wire harnesses, current research and future trends. Procedia CIRP 2019;81:387–92. http://dx.doi.org/10.1016/j.procir.2019.03.067.

[4] Heisler P, Utsch D, Kuhn M, Franke J. Optimization of wire harness assembly using human–robot-collaboration. Procedia CIRP 2021;97:260–5. http://dx.doi.org/10.1016/j.procir.2020.05.235.

[5] Aguirre E, Raucent B. Economic comparison of wire harness assembly systems. J Manuf Syst 1994;13(4):276–88. http://dx.doi.org/10.1016/0278-6125(94)90035-3.

[6] Salunkhe O, Quadrini W, Wang H, Stahre J, Romero D, Fumagalli L, Lämkull D. Review of current status and future directions for collaborative and semi-automated automotive wire harnesses assembly. Procedia CIRP 2023;120:696–701. http://dx.doi.org/10.1016/j.procir.2023.09.061.

[7] Azmeh S, Nguyen H, Kuhn M. Automation and industrialisation through global value chains: North Africa in the German automotive wiring harness industry. Struct Change Econ Dyn 2022;63:125–38. http://dx.doi.org/10.1016/j.strueco.2022.09.006.

[8] Navas-Reascos GE, Romero D, Rodriguez CA, Guedea F, Stahre J. Wire harness assembly process supported by a collaborative robot: A case study focus on ergonomics. Robotics 2022;11(6):131. http://dx.doi.org/10.3390/robotics11060131.

[9] Gualtieri L, Palomba I, Merati FA, Rauch E, Vidoni R. Design of human-centered collaborative assembly workstations for the improvement of operators' physical ergonomics and production efficiency: A case study. Sustainability 2020;12(9):3606. http://dx.doi.org/10.3390/su12093606.

[10] ELVEZ — the company. ELVEZ; 2020, URL https://elvez.si/the-company/#gallery. [Online; Accessed 16 Sep 2024].

[11] Zürn M, Wnuk M, Lechler A, Verl A. Topology matching of branched deformable linear objects. In: 2023 IEEE international conference on robotics and automation. IEEE; p. 2023–02. http://dx.doi.org/10.1109/ICRA48891.2023.10161483.

[12] Sanchez J, Corrales J-A, Bouzgarrou B-C, Mezouar Y. Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. Int J Robot Res 2018;37(7):688–716. http://dx.doi.org/10.1177/0278364918779698.

[13] Zhou P, Zheng P, Qi J, Li C, Lee H-Y, Duan A, Lu L, Li Z, Hu L, Navarro-Alarcon D. Reactive human–robot collaborative manipulation of deformable linear objects using a new topological latent control model. Rob Comput Integr Manuf 2024;88:102727. http://dx.doi.org/10.1016/j.rcim.2024.102727.

[14] Tunstel E, Dani A, Martinez C, Blakeslee B, Mendoza J, Saltus R, Trombetta D, Rotithor G, Fuhlbrigge T, Lasko D, Wang J. Robotic wire pinning for wire harness assembly automation. In: 2020 IEEE/aSME international conference on advanced intelligent mechatronics. IEEE; p. 06–9. http://dx.doi.org/10.1109/AIM43001.2020.9158905.

[15] Hermansson T, Bohlin R, Carlson JS, Söderberg R. Automatic assembly path planning for wiring harness installations. J Manuf Syst 2013;32(3):417–22. http://dx.doi.org/10.1016/j.jmsy.2013.04.006.

[16] Kicki P, Bednarek M, Lembicz P, Mierzwiak G, Szymko A, Kraft M, Walas K. Tell me, what do you see?—interpretable classification of wiring harness branches with deep neural networks. Sensors 2021;21(13):4327. http://dx.doi.org/10.3390/s21134327.

[17] Lee JS, Kim D, Roh M, Kim JB. Vision based deformable wires recognition using point cloud in wire harness supply. In: 2022 13th international conference on information and communication technology convergence. 2022, p. 208–13. http://dx.doi.org/10.1109/ICTC55196.2022.9952859.

[18] Kamiya K, Wang Y, Lu J, Kondoh S, Kanda S, Honda Y, Mizoguchi H, Nishio M, Makino K, Ota J. Learning from human hand demonstration for wire harness grasping. In: 2024 IEEE international conference on advanced intelligent mechatronics. IEEE; p. 15–9. http://dx.doi.org/10.1109/AIM55361.2024.10637203.

[19] Zhang X, Domae Y, Wan W, Harada K. Learning efficient policies for picking entangled wire harnesses: An approach to industrial bin picking. IEEE Rob Autom Lett 2022;8(1):73–80. http://dx.doi.org/10.1109/LRA.2022.3222995.
[20] Zhang X, Domae Y, Wan W, Harada K. A closed-loop bin picking system for entangled wire harnesses using bimanual and dynamic manipulation. Rob Comput Integr Manuf 2024;86:102670. http://dx.doi.org/10.1016/j.rcim.2023.102670.
[21] Guo J, Zhang J, Wu D, Gai Y, Chen K. An algorithm based on bidirectional searching and geometric constrained sampling for automatic manipulation planning in aircraft cable assembly. J Manuf Syst 2020;57:158–68. http://dx.doi.org/10.1016/j.jmsy.2020.08.015.
[22] González Huarte J, Ortiz de Zarate M, Ibarguren A. CAD-based robot programming solution for wire harness manufacturing in aeronautic sector. Robotics 2023;12(5):130. http://dx.doi.org/10.3390/robotics12050130.
[23] Fresnillo PM, Vasudevan S, Mohammed WM. An approach for the bimanual manipulation of a deformable linear object using a dual-arm industrial robot: cable routing use case. In: 2022 IEEE 5th international conference on industrial cyber-physical systems. IEEE; p. 24–6. http://dx.doi.org/10.1109/ICPS51978.2022.9816981.
[24] Galassi K, Palli G. Robotic wires manipulation for switchgear cabling and wiring harness manufacturing. In: 2021 4th IEEE international conference on industrial cyber-physical systems. IEEE; p. 10–2. http://dx.doi.org/10.1109/ICPS49255.2021.9468128.
[25] Süberkrüb F, Laezza R, Karayiannidis Y. Feel the tension: Manipulation of deformable linear objects in environments with fixtures using force information. In: 2022 IEEE/RSJ international conference on intelligent robots and systems. IEEE; p. 23–7. http://dx.doi.org/10.1109/IROS47612.2022.9982065.
[26] Nguyen TP, Yoon J. A novel vision-based method for 3D profile extraction of wire harness in robotized assembly process. J Manuf Syst 2021;61:365–74. http://dx.doi.org/10.1016/j.jmsy.2021.10.003.
[27] Zhu J, Navarro B, Passama R, Fraisse P, Crosnier A, Cherubini A. Robotic manipulation planning for shaping deformable linear objects WithEnvironmental contacts. IEEE Rob Autom Lett 2019;5(1):16–23. http://dx.doi.org/10.1109/LRA.2019.2944304.
[28] Nguyen TP, Kim D, Lim H-K, Yoon J. Revolutionizing robotized assembly for wire harness: A 3D vision-based method for multiple wire-branch detection. J Manuf Syst 2024;72:360–72. http://dx.doi.org/10.1016/j.jmsy.2023.12.002.
[29] Malvido Fresnillo P, Vasudevan S, Mohammed WM, Martinez Lastra JL, Perez Garcia JA. An approach based on machine vision for the identification and shape estimation of deformable linear objects. Mechatronics 2023;96:103085. http://dx.doi.org/10.1016/j.mechatronics.2023.103085.
[30] Guo J, Zhang J, Gai Y, Wu D, Chen K. Visual recognition method for deformable wires in aircrafts assembly based on sequential segmentation and probabilisitic estimation. In: 2022 IEEE 6th information technology and mechatronics engineering conference. IEEE; p. 04–6. http://dx.doi.org/10.1109/ITOEC53115.2022.9734432.
[31] Govoni A, Laudante G, Mirto M, Natale C, Pirozzi S. Towards the automation of wire harness manufacturing: A robotic manipulator with sensorized fingers. In: 2023 9th international conference on control, decision and information technologies. IEEE; p. 03–6. http://dx.doi.org/10.1109/CoDIT58514.2023.10284109.
[32] Laudante G, Mirto M, Pennacchio O, Galassi K, Govoni A, Pasquali A. Mechatronic integration of a dual-arm robotic system for wiring harness manufacturing. IEEE/ASME Trans Mechatron 2025;1–12. http://dx.doi.org/10.1109/TMECH.2025.3536627.
[33] Lv N, Liu J, Ding X, Lin H. Assembly simulation of multi-branch cables. J Manuf Syst 2017;45:201–11. http://dx.doi.org/10.1016/j.jmsy.2017.09.007.
[34] Zhou H, Lu Q, Qian J. A graph-based framework for assembly sequence planning of a cable harness. J Manuf Syst 2024;73:39–51. http://dx.doi.org/10.1016/j.jmsy.2024.01.007.
[35] Kim J, Kim J-G, Park J, Han B-K, Kim S, Park DI. Dual-arm path-planning algorithm for wiring harness assembly using redundantly actuated robotic systems. IEEE Access 2023;11:98427–35. http://dx.doi.org/10.1109/ACCESS.2023.3306793.
[36] Román Ibáñez V, Pujol FA, García Ortega S, Sanz Perpiñán JM. Collaborative robotics in wire harnesses spot taping process. Comput Ind 2021;125:103370. http://dx.doi.org/10.1016/j.compind.2020.103370.
[37] Palomba I, Gualtieri L, Rojas R, Rauch E, Vidoni R, Ghedin A. Mechatronic re-design of a manual assembly workstation into a collaborative one for wire harness assemblies. Robotics 2021;10(1):43. http://dx.doi.org/10.3390/robotics10010043.
[38] Nguyen HG, Franke J. Deep learning-based optical inspection of rigid and deformable linear objects in wiring harnesses. Procedia CIRP 2021;104:1765–70. http://dx.doi.org/10.1016/j.procir.2021.11.297.
[39] Nguyen HG, Habiboglu R, Franke J. Enabling deep learning using synthetic data: A case study for the automotive wiring harness manufacturing. Procedia CIRP 2022;107:1263–8. http://dx.doi.org/10.1016/j.procir.2022.05.142.
[40] Galassi K, Caporali A, Laudante G, Palli G. Scalable shared encoding architecture for learning-based error detection in robotic wiring harness assembly. In: 2024 IEEE international conference on advanced intelligent mechatronics. IEEE; p. 15–9. http://dx.doi.org/10.1109/AIM55361.2024.10637054.
[41] Zürn M, Kienzlen A, Klingel L, Lechler A, Verl A, Ren S, Xu W. Deep learning-based instance segmentation for feature extraction of branched deformable linear objects for robotic manipulation. In: 2023 IEEE 19th international conference on automation science and engineering. IEEE; p. 26–30. http://dx.doi.org/10.1109/CASE56687.2023.10260646.
[42] Yumbla F, Abeyabas M, Luong T, Yi J-S, Moon H. Preliminary connector recognition system based on image processing for wire harness assembly tasks. In: 2020 20th international conference on control, automation and systems. IEEE; p. 13–6. http://dx.doi.org/10.23919/ICCAS50221.2020.9268291.
[43] Wang H, Johansson B. Deep learning-based connector detection for robotized assembly of automotive wire harnesses. In: 2023 IEEE 19th international conference on automation science and engineering. IEEE; p. 26–30. http://dx.doi.org/10.1109/CASE56687.2023.10260619.
[44] Zhou H, Li S, Lu Q, Qian J. A practical solution to deformable linear object manipulation: A case study on cable harness connection. In: 2020 5th international conference on advanced robotics and mechatronics. IEEE; p. 18–21. http://dx.doi.org/10.1109/ICARM49381.2020.9195380.
[45] Yumbla F, Yi J-S, Abayebas M, Shafiyev M, Moon H. Tolerance dataset: mating process of plug-in cable connectors for wire harness assembly tasks. Intel Serv Robot 2020;13(1):159–68. http://dx.doi.org/10.1007/s11370-019-00307-5.
[46] Fresnillo PM, Vasudevan S, Garcia JAP, Lastra JLM. An open and reconfigurable user interface to manage complex ROS-based robotic systems. IEEE Access 2024;12:114601–17. http://dx.doi.org/10.1109/ACCESS.2024.3445152.
[47] Malvido Fresnillo P, Vasudevan S, Mohammed WM, Martinez Lastra JL, Perez Garcia JA. Extending the motion planning framework — moveit with advanced manipulation functions for industrial applications. Rob Comput Integr Manuf 2023;83:102559. http://dx.doi.org/10.1016/j.rcim.2023.102559.
[48] MalvidoFresnillo P, Mohammed WM, Vasudevan S, PerezGarcia JA, MartinezLastra JL. Generation of realistic synthetic cable images to train deep learning segmentation models. Mach Vis Appl 2024;35(4):1–14. http://dx.doi.org/10.1007/s00138-024-01562-y.
[49] Choquet G. Topology. Academic Press; 1966, p. 129–31.
[50] Kuffner JJ, LaValle SM. RRT-connect: An efficient approach to single-query path planning. In: Proceedings 2000 iCRA. millennium conference. IEEE international conference on robotics and automation. symposia proceedings (cat. no. 00CH37065), Vol. 2, IEEE; 2000, p. 995–1001.
[51] Pedersen MR, Nalpantidis L, Andersen RS, Schou C, Bøgh S, Krüger V, Madsen O. Robot skills for manufacturing: From concept to industrial deployment. Rob Comput Integr Manuf 2016;37:282–91. http://dx.doi.org/10.1016/j.rcim.2015.04.002.
[52] Arrais R, Veiga G, Ribeiro TT, Oliveira D, Fernandes R, Conceição AGS, Farias PCMA. Application of the open scalable production system to machine tending of additive manufacturing operations by a mobile manipulator. In: progress in artificial intelligence. Cham, Switzerland: Springer; 2019, p. 345–56. http://dx.doi.org/10.1007/978-3-030-30244-3_29.
[53] Mayr M, Rovida F, Krueger V. SkiROS2: A skill-based robot control platform for ROS. In: 2023 IEEE/RSJ international conference on intelligent robots and systems. IEEE; p. 01–5. http://dx.doi.org/10.1109/IROS55552.2023.10342216.
[54] Gašpar T, Deniša M, Radanovič P, Ridge B, Savarimuthu TR, Kramberger A, Priggemeyer M, Roßmann J, Wörgötter F, Ivanovska T, Parizi S, Gosar Ž, Kovač I, Ude A. Smart hardware integration with advanced robot programming technologies for efficient reconfiguration of robot workcells. Rob Comput Integr Manuf 2020;66:101979. http://dx.doi.org/10.1016/j.rcim.2020.101979.
[55] Schou C, Andersen RS, Chrysostomou D, Bøgh S, Madsen O. Skill-based instruction of collaborative robots in industrial settings. Rob Comput Integr Manuf 2018;53:72–80. http://dx.doi.org/10.1016/j.rcim.2018.03.008.
[56] Malvido Fresnillo P, Vasudevan S, Mohammed WM, Martinez Lastra JL, Pérez García JA. A method for understanding and digitizing manipulation activities using programming by demonstration in robotic applications. Robot Auton Syst 2023;170:104556. http://dx.doi.org/10.1016/j.robot.2023.104556.
[57] Yin Y, Zheng P, Li C, Wan K. Enhancing human-guided robotic assembly: AR-assisted DT for skill-based and low-code programming. J Manuf Syst 2024;74:676–89. http://dx.doi.org/10.1016/j.jmsy.2024.04.016.