

Wiring connector-terminated cables based on manipulation planning with collision-free EMD net

Kimitoshi Yamazaki^{a,1,*}, Kyoto Nozaki^b, Yuichiro Matsuura^c, Solvi Arnold^{a,1}

^a Mech. Sys. Engineering, Shinshu University, Nagano, Japan

^b Grad. Sch. of Science and Technology, Shinshu University, Nagano, Japan 380-8553, Wakasato 4-17-1, Nagano, Nagano, Japan

^c Seiko Epson Corp. Nagano, Japan 399-0295, 1010 Fujimi, Fujimi-machi, Suwa-gun, Nagano, Japan

ARTICLE INFO

Keywords:

Wiring
Manipulation planning
Deformable linear object
EMD net

ABSTRACT

In this paper, we propose a manipulation planning method for cable wiring in the assembly of electric appliances etc. We address a scenario where a robot grasps a connector attached to the end of a cable and has to bring the connector to a socket. To accomplish this automatically, we propose a novel manipulation planning method. The method extends the Encode-Manipulate-Decode network (EMD net), which can predict shape changes of deformable objects and generate robot motion sequences for producing desired shape transitions. This enables us to find connector trajectories that avoid collision between the cable and the surrounding environment. We conducted experiments with several different cable lengths. We also introduce some functions required for real-world wiring, such as online cable shape modification. Experimental results show that the proposed method can achieve stable manipulation of real cables.

1. Introduction

In the assembly of electrical appliances, automating cable wiring is a challenging task. This is because cables are flexible objects that deform when manipulated. Furthermore, motors and circuit boards are often incorporated into electrical appliances before cable wiring, presenting obstacles that must be avoided during the wiring process. The wiring process requires consideration of all the forces and deformations applied to the cable during manipulation, and avoidance of any snagging on surrounding structures. Consequently, much of cable wiring is currently performed manually. However, assembling electrical appliances is a burdensome task for humans, and automation is necessary. Such automated systems should require minimal human intervention, and should be robust enough to adapt to changes in cable length and type, as well as to changes in the work environment.

The purpose of this study is to establish a motion planning method for wiring a connector-terminated cable. Fig. 1 shows a simplified diagram of the wiring scenario addressed in this study. Challenges to be considered are also presented. The manipulation target is a cable with a connector attached to its end. The root of the cable is attached to a circuit board. The actions required of the robot in automating this task is to grasp the connector, move it to a predetermined socket position, and

then insert the connector into the socket. At this point, it is desirable to avoid interference of the cable, which deforms during connector transport, with surrounding structures. One important challenge for achieving this is to generate a path along which the connector can safely be transported.

Various studies on the automation of cable wiring are reported in the literature. Many studies target the task of connector insertion [1,2]. Others have proposed manipulation planning methods for applying arbitrary deformations to cables [3,4]. In the former, force sensing and control for proper mating of connectors and sockets are central subjects. Since the goal of the present study is to move the connector to the vicinity of the socket, potential for compatibility with these studies is high. On the other hand, the latter studies address similar subject matter as the present study. However, the goal of the present study is to avoid interference in situations where the cable may get caught on surrounding structures, by actively deforming the cable through appropriate manipulation of the connector. This is a practical function in wiring work, but has not been considered much in previous studies.

The method proposed in this paper makes it possible to predict the deformation of cables when the connector is picked and moved, and uses the prediction result as a cue to generate an appropriate manipulation. This results in a trajectory for transporting the connector that avoids

* Corresponding author.

E-mail address: kyamazaki@shinshu-u.ac.jp (K. Yamazaki).

¹ SEIKO-EPSON

interference between the cable and the surrounding environment. The prediction and generation of manipulation trajectories are performed using the EMD net (Encode-Manipulate-Decode network) [5], which has a modular neural network architecture. However, since interference avoidance was previously not considered in this architecture, we here extend the method and propose a new Collision-Free EMD net (CF-EMD net).

The contributions of the present study are as follows:

- We propose a manipulation planning method to move a connector attached to a cable fixed at one end on a circuit board to a desired position while avoiding interference of the cable with surrounding obstacles.
- As a strategy for efficient use of the method, we show how to generate training data using physical simulation of Deformable Linear Objects (DLOs) and train the neural network using this simulation data.
- We confirm that the proposed method works in a problem setting that simulates the internal structure of an electrical appliance through experiments on actual equipment. Furthermore, we show how to manipulate real cables while bridging the gap between simulation and reality by correcting the difference between the predicted and real cable geometry on the spot.

The structure of this paper is as follows. The next section introduces related work. Section 3 describes the problem setup, our approach, and an overview of our methodology. Section 4 describes our extended modular neural network for interference avoidance. Section 5 describes data collection and neural network training using physical simulation. In Section 6, we describe a method for correcting the results of physical simulations to match the geometry of real cables in order to apply the proposed method in real-world cable manipulation. Section 7 presents our verification experiments, and Section 8 summarizes the study.

2. Related work

2.1. Cable manipulation

In R&D aimed at product assembly automation, there are several existing works on the topic of cable manipulation. As mentioned in the previous section, one representative task is robotic insertion of connectors into sockets. Huang et al. [1] successfully inserted a connector into a socket using force control. Chen et al. [2] investigated the relationship between the force sensor signal and the relative positions of a connector pair, in order to avoid jamming during the connector mating process. Song et al. [6] used visual servo and impedance control to minimize position errors in the mating process. Yumbla et al. [7] made a dataset of mating tolerances between 70 types of connectors and sockets commonly used in manufacturing. Romeres et al. [8] proposed a method based on Gaussian process regression to learn the profile of forces during successful insertion and quantify the deviations of the tolerances obtained from it.

Various methods have also been proposed for high-level automation involving 3D pose estimation [9,10]. Sumi et al. [11] developed a robot assembly system with a 3D vision sensing system and demonstrate servo amplifier assembly with manipulation of a connector-terminated cable. Yumbla et al. [12] successfully detected the position of a connector. Zhou et al. [13] estimated the pose of a connector and used this pose to insert the connector into a socket using a dual-armed robot. Ying et al. [14] proposed a method to accurately estimate the 3D pose of a connector and grasp it. They combined PointNet++ [15] for connector part detection with ICP for accurate pose estimation. As described above, there are many studies of recognition and manipulation for wiring tasks, but these studies did not explicitly consider cable shape change.

On the other hand, there have been studies that consider a cable as a DLO subject to direct manipulation. Sano et al. [16] demonstrate wiring of a flexible flat cable onto an electronic substrate. The main contributions of this study were novel manipulation procedures and dedicated end-effectors, but cable shape estimation was not addressed. She et al.

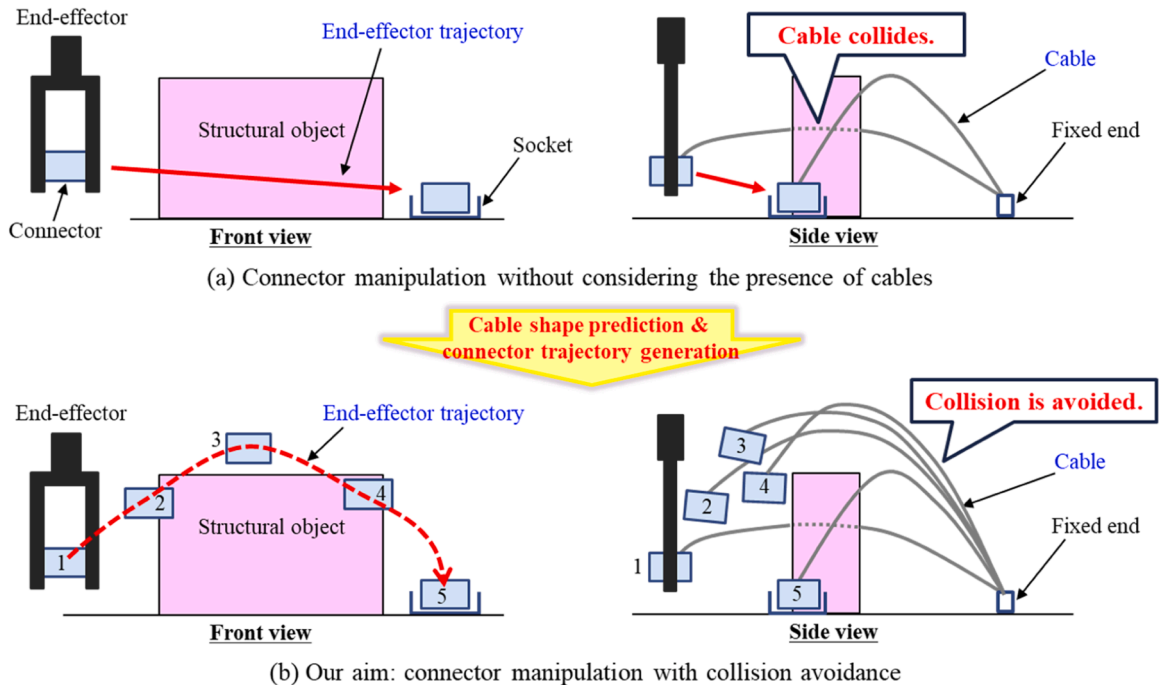


Fig. 1. If only the robot and the connector are considered in motion generation, the cable may interfere with surrounding structures, as in (a). This study aims to generate manipulations that avoid interference of the cable with surrounding structures, as shown in (b). The challenge here is to generate a connector trajectory that accounts for the time-series of the cable shape.

[17] focused on sliding hand motions for clamping a cable, and detected cable posture and frictional force between cables using tactile sensors. Zhu et al. [3] proposed a motion planning method for transitioning a cable from its current shape, estimated from color images, to a target shape, and succeeded in deforming a cable grasped at both ends by a dual-armed robot into a desired shape.

2.2. DLO modeling and manipulation learning

Outside the context of cable assembly as well, DLOs have been an active topic of research in recent years. We can find excellent work in modeling and manipulation learning. With regard to DLO simulation, one class of methods builds on the finite element model (FEM) [18]. FEM simulation results have been used to control the shape of physical DLOs. Position-based dynamics (PBD) is another representative recent simulation method, which has been applied to DLO manipulation as well. Liu et al. [19] proposed a modeling method for rope-like objects based on PBD. These models make it possible to compensate for the shortcomings of physical parameter estimation, and improve the matching of rope physics to real-world scenarios. Ying et al. [20] proposed a differentiable physics simulation method and demonstrated shape-control on rope-like objects, with consideration of the effects of contacts with surrounding environment.

Several methods have been proposed that tightly couple simulation and learning. Yang et al. [21] combined an interaction network and a recurrent neural network to learn a dynamic DLO model. They used the obtained model in a model predictive controller to achieve a given goal shape. Huo et al. [22] proposed an approach for training a network from a synthetic dataset to encode key points on DLOs. Yu et al. [4] proposed an efficient method for obtaining a deformation model by learning the relationship between a manipulation and a DLO shape change in a simulated environment, and then learning the same thing in the real world. Yamazaki et al. [23] proposed the use of GANs as a method for generating DLO shape transitions. They showed that it is possible to generate smooth shape transitions for motions of the endpoints of DLOs, and also to automatically generate transitions between different shapes. Chang et al. [24] proposed a model-based DLO manipulation method, called Sim2Real2Sim, to perform the task of inserting a plug into a socket. These studies are useful for applying specific manipulations to cables, but their applicability in other settings, such as wiring in the presence of obstacles, is unknown. The distinguishing feature of the method presented here is its ability to generate appropriate manipulation sequences while predicting the DLO's shape transitions over a long horizon. This makes it possible to obtain a manipulation sequence that does not break down even when parts of the cable part have to traverse through the vicinity of obstacles in the scene.

3. Problem setting and our approach

3.1. Problem setting

We assume a task environment consisting of the circuit board of an electrical appliance, with a cable extending from the circuit board and terminating in a connector. Hence, the connector part is floating freely. It is assumed that the circuit board has elements and other structures attached to it, and that these structures are of non-negligible height. That is, it is possible for cables or the robot itself to collide with them during wiring. Such structures are assumed to extend upward from the board, and the structures are assumed not to protrude from above or from the side.

Under these conditions, we consider the task of grasping such a connector and inserting it into a socket on the board, as shown in Fig. 1. We adopt an industrial robot manipulator equipped with a gripper-type end-effector as our robot platform. Such simple systems are commonly employed in the manufacturing of electronics products because of their lower cost compared to multi-arm systems. Here, to manipulate the

cable, the robot should grasp the connector. We let the end-effector grasp the connector, because grasping the cable would make the subsequent insertion of the connector into the socket difficult. Therefore, we will consider manipulation planning under the assumption that only the connector is directly manipulated and that the cable deforms pursuant to the connector's movements. However, we assume that the cable has a certain level of stiffness, and that as long as the connector part is grasped, gravity will induce no significant downward deformation in the cable.

3.2. Issues and approach

A possible problem in the above setup is that the cable gets caught on parts of the surrounding environment when the robot moves the connector, preventing the connector from moving to its goal position or damaging the cable. Therefore, motion planning must be able to predict the state of not only the robot but also the cable appropriately and detect when a motion would cause such snagging in advance. Furthermore, it is desirable to generate manipulations for transporting the connector to the goal position that avoid snagging. However, cables are flexible objects and deform as they are manipulated. Therefore, it is necessary to know how the cable deforms when a manipulation is applied. If this is not adequately addressed, the likelihood of the cable interfering with its surroundings increases.

To solve this problem, this study implements neural network modules for predicting how the cable geometry changes and for generating connector trajectories. We construct a manipulation planner for DLOs by learning both of these functions simultaneously. The base of this system is the EMD net [5]. However, since the EMD net does not provide collision avoidance, we extend the method here to add this functionality. In this study, we refer to the extended method as the Collision-Free EMD net (CF-EMD net).

To train the CF-EMD net, a certain amount of training data is required. Collecting these data in the real world is a heavy burden. Therefore, we implement a DLO physics simulation and collect training data in virtual space. However, there are subtle differences in the behavior of flexible objects between the real and virtual environments. Our approach for bridging this gap is to use a range image sensor to measure the real-world cable shape, and adjust the CF-EMD net's output on basis of the measured shape. The details of the method are explained in the next section.

4. CF-EMD net

4.1. EMD net [5]

The EMD net is an action planner that primarily targets deformable object manipulation. It takes the state of a deformable object as input, performs a virtual manipulation on it, and outputs the predicted state after the manipulation. The network consists of a modular structure: an encoder, manipulation network, and decoder. This modular structure can be connected recursively to predict state changes over long horizons and generate appropriate manipulations accordingly. The encoder and decoder use fully-connected layers to compress and decompress data, respectively. Although there are several implementation patterns for manipulation networks [25], this study adopts the method of processing minute manipulations continuously. The manipulation network combines fully connected layers and an LSTM (Long Short-Term Memory). LSTM [26] is a type of recurrent neural network that is known to have high prediction performance for medium-term time series data. Therefore, it is suitable for our study, which requires smooth time-series of changes in the cable shape.

Fig. 2 shows the structure and data flow of the EMD net. Let s_n be the cable shape at the n -th time step and let m_n be a minute manipulation applied to the cable. We let $\hat{s}$ indicate the shape of the cable predicted by EMD net. The EMD net's encoder converts the shape s into a low-

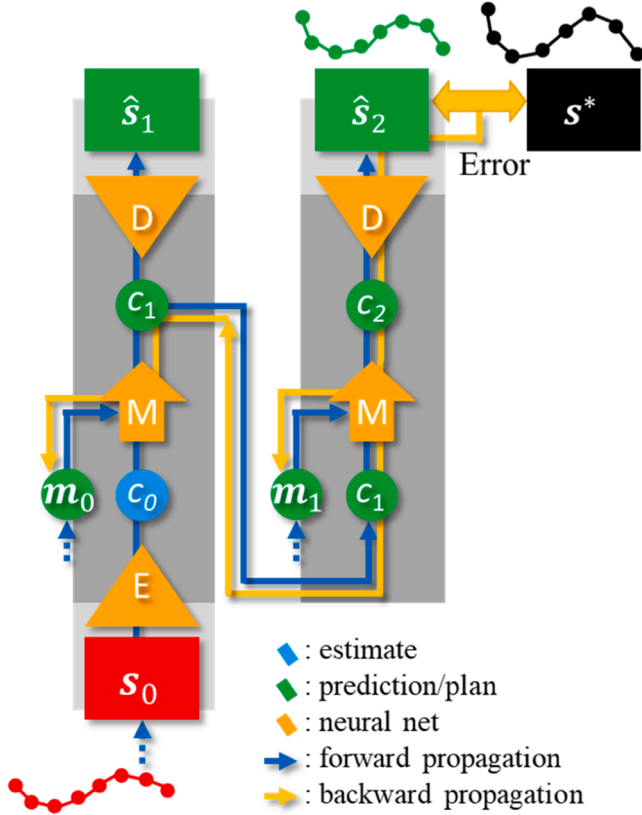


Fig. 2. EMD net structure and data flow. The case shown here is one in which the goal shape is reached in two operations. In actual wiring task, dozens of module instances are connected to generate long-horizon manipulation sequences.

dimensional compressed representation c . Then, c_n and the manipulation value m_n are input to the manipulation network to obtain a compressed representation c_{n+1} of the shape after manipulation. This network is used recursively to predict how the cable will be deformed by a sequence of micro-manipulations specified as a time series of m values.

Next we explain how to use the EMD net. First, the target DLO shape $s_n (= s^*)$ and the current shape s_0 are given. Also, a sequence of minute manipulations $m_{0:n-1}$ is given. Then, as indicated by the blue arrows in Fig. 2, virtual operations are successively applied to s_0 , yielding a series $\hat{s}_{1:n}$ of the resulting shape predicted for each operation. Then we compare the final shape prediction $\hat{s}_n$ with s_n , and update $m_{0:n-1}$ by back propagation of the difference, as illustrated by the orange arrows in Fig. 2. This is repeated until the difference between $\hat{s}_n$ and s_n is deemed sufficiently small. The resulting manipulation time-series $m_{0:n-1}$ is used as the planning result.

4.2. Representation of cable, robot and environment

To use the above EMD net, the shape s must be given in an appropriate representation. Additionally, since our problem setting requires collision avoidance, we need to represent the robot hand and the surrounding environment in a form that is suitable for collision checking. These representation formats are described below.

First off, we consider the cable as a Deformable Linear Object (DLO) represented by a point chain model. That is, the cable is divided into segments of a given length, with equally spaced nodes set at the division points. Assuming that adjacent nodes are connected by straight line segments, the resulting collection of line segments approximates the shape of the cable. However, only the node positions are observed by EMD net, and the line segments between nodes are not used in practice.

Hence, s can be expressed as $\{x_k\}_{k=1}^K$, where x denotes three-dimensional node coordinates and K is the total number of nodes. Meanwhile, the robot hand and the surrounding structures are first approximated as a set of rectangular cuboids, and then represented as a set of three-dimensional points sampled at equal intervals on the cuboids' surfaces. This simplified representation allows for simple and fast distance-based calculation of collision risk during manipulation planning.

4.3. CF-EMD net

We modify the EMD net as shown in Fig. 3 to meet the requirements of this study. In order to use this network for manipulation planning, a new loss function is defined as follows.

$$\argmin_{m_i} L(\hat{s}_i, h_i, h_n, h^*, m_i, o),$$

where h_i denotes the set of points that approximates the shape of the hand during manipulation, h_n denotes the same set of points in their final posture at the end of the manipulation, and h^* is the same set of points for the hand posture obtained when the connector is in its goal pose. Set o contains the points representing the surrounding structures. Loss L is composed of four terms as follows.

$$L(\hat{s}_i, h_i, h_n, h^*, m_i, o) =$$

$$MSE(h^*, h_n) + \omega_0 \cdot LEN(m_i) + \omega_1 \frac{1}{AVO(\hat{s}_i, h_i, o)} + \omega_2 \cdot DIS(s_i, h_i),$$

where ω_s are weight coefficients. The first term on the right-hand side, MSE, calculates the Mean Squared Error between the hand shape at the goal pose and the predicted final hand shape. The purpose of the second term on the right-hand side is to minimize the length of the manipulation trajectory. This function calculates the lengths of the translation vectors for each manipulation m_i and adds them up.

The third term AVO is defined as follows:

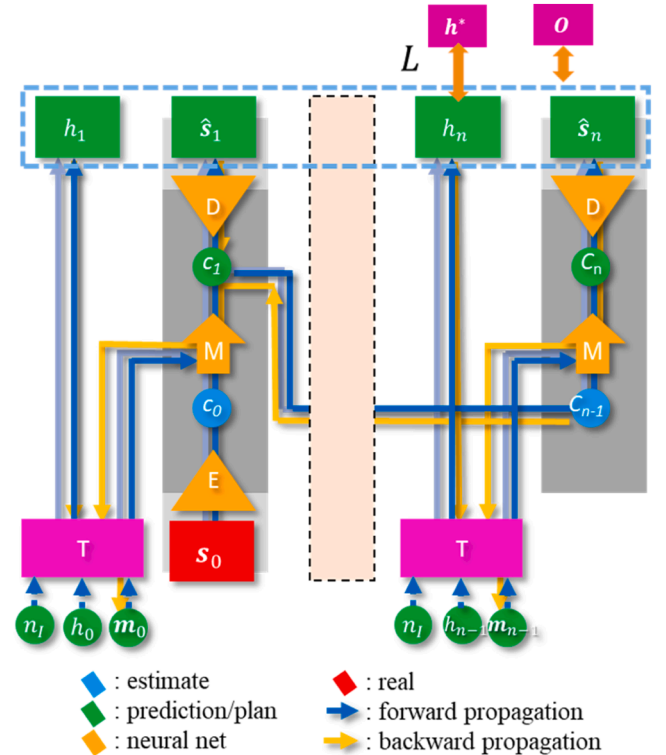


Fig. 3. CF-EMD net structure and data flow. Box T applies rigid body transformations (rotation and translation) on basing of manipulation input m_i .

$$AVO(\hat{s}_i, h_i, o) = \begin{cases} d_a(\hat{s}_i, h_i, o) & \text{if } d_a \leq r \\ \infty & \text{otherwise,} \end{cases}$$

where

$$d_a(\hat{s}_i, h_i, o) = \min_{x \in \hat{s}_i, h_i, y \in o} \|x - y\|.$$

Fig. 4 illustrates the role of this function. Given the set of points describing the hand-cable complex and the set of points describing the surrounding structures, the function searches for the pair of points at which the distance between the two point sets is smallest. The smaller this distance, the larger the output value of the function. In the above equation, r is a threshold value. There are two reasons for setting a threshold. One is to prevent the generation of paths that keep excessive distance from the surrounding structures. The other is to avoid leaving distance between h_n and h^* in cases where the DLO or hand necessarily approach an obstacle in their final pose.

The fourth term DIS is defined as follows

$$DIS(s_i, h_i) = \begin{cases} -d_d(s_i, h_i, o) & \text{if } d_d < 0 \\ 0 & \text{otherwise,} \end{cases}$$

where

$$d_d(s_i, h_i) = \min_x x$$

In this study, x is effectively the z -coordinate, since only the height direction needs to be checked.

Fig. 5 illustrates the role of this function. This term is responsible for avoiding contact of the hand or cable with the circuit board, and is set to 0 when the minimum distance to the circuit board is greater than or equal to 0, in order to prevent generation of trajectories at unnecessarily large heights.

Using the above configuration of the loss function, a path can be generated for the DLO and the hand to move the connector around without interfering with the surroundings. Adam [27] is used as the optimization algorithm. As a reminder, the abovementioned loss function is used for manipulation planning. For training the neural network a different loss function is used. The training process is discussed in the next section.

5. Data collection and learning

5.1. Strategies for using learning-based planners

The CF-EMD net is a neural network. Therefore, as is the case with other such methods, it needs to be trained with training data in advance. In other words, it is necessary to collect a large amount of numerical data representing the deformation of DLOs by manipulation. There are two options for this: collecting the data through real-world manipulation, or using simulation. However, since there are an infinite number of possible cable states, collecting data using an actual robot is a very burdensome task. In this study, data collection is performed using physical simulation.

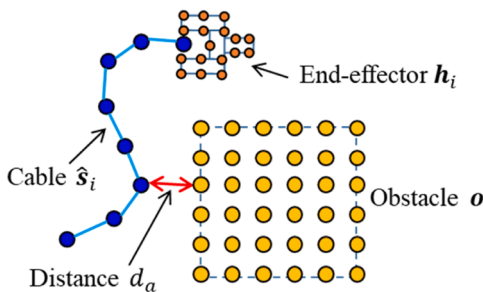


Fig. 4. Conceptual diagram of collision avoidance function with structural objects.

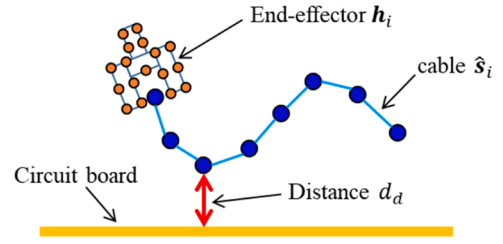


Fig. 5. Conceptual diagram of the function for maintaining proper height of the cable.

One key issue is how to handle the difference between the DLO shape estimated by the planner and the real cable shape when cable manipulation is performed using robot hardware. In this study, when performing real-world cable manipulation, the initial DLO shape input to the CF-EMD net is corrected using the results of in-situ cable measurements. When running the CF-EMD net, the initial shape s_0 of the DLO is generated by physical simulation, using the poses of the cable's root and connector as inputs. This is because the root's pose is known, and the connector's pose can be calculated given the assumption that the robot hand is grasping the connector normally. However, there may be a difference between the result of the physics simulation and the actual cable geometry. If this difference is left unchecked, all predictions of the DLO geometry during operation will be affected.

The initial shape of the actual cable is then measured with a distance sensor or similar device. However, while it would be good if the measurement could be done perfectly, in our cable routing setting we have to contend with problems such as a small workspace and a tendency for the cable to be occluded by parts of the surrounding structures, the robot hand, and the cable itself. Therefore, it is not always possible to measure the cable shape completely. In consideration of these facts, we proceed as follows. We obtain point cloud data from a distance sensor before manipulation planning, and use it to bridge the gap between simulation and reality, under the assumption that only partial measurements can be obtained. A detailed description is given in the next section.

5.2. Physics simulation

Physics simulation is employed to collect training data. We apply the method of Lv et al. [28], which falls into the category of methods that represent DLOs using mass-spring elements [29]. An overview of the method is given below.

As shown in Fig. 6, we let each mass point be connected by three types of springs, corresponding to tension-compression, bending, and torsion. The mass points we refer to here correspond to the nodes of the DLO model. The coordinates x_i^p of each mass point are obtained by solving the following equation using the Euler method:

$$m_i \frac{\partial^2 x_i^p}{\partial t^2} + k^d \frac{\partial x_i^p}{\partial t} = F_i = -\frac{\partial E}{\partial x_i^p} + F_i^e,$$

where m_i is the mass of point i , F_i^e indicates the external force exerted on it, and E is the sum of the energy due to the tension-compression,

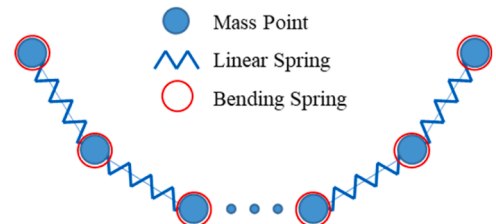


Fig. 6. Physical model of cable.

bending, and torsion springs comprising the cable. The damping coefficient k_d prevents excessive vibration of the masses in the calculation process. The derivative $-\partial E/\partial \mathbf{x}_i^p$ is given by the following equation:

$$-\frac{\partial E}{\partial \mathbf{x}_i^p} = \mathbf{F}_i^s + \mathbf{F}_i^b + \mathbf{F}_i^t = -\frac{\partial E^s}{\partial \mathbf{x}_i^p} - \frac{\partial E^b}{\partial \mathbf{x}_i^p} - \frac{\partial E^t}{\partial \mathbf{x}_i^p}$$

Here, $\mathbf{F}_i^s$, $\mathbf{F}_i^b$, and $\mathbf{F}_i^t$ are the forces due to the tension-compression, bending and torsion springs, respectively, and E^s , E^b , and E^t are their respective energies.

In the process of wiring a cable by its connector, the torsion applied to the cable is small and hence torsion is less important than tension-compression and bending. Therefore, we omit torsion from consideration, in order to reduce the simulation time costs and avoid unnecessary instability in the behavior of the simulated DLO.

5.3. Data collection for cable wiring

DLO Manipulation data is collected using the simulation method described in the previous subsection. Each data example consists of the sequence of manipulations $m_{0:n-1}$ applied to DLO, the DLO's initial geometry s_0 , and the sequence of DLO geometries $s_{1:n}$ traversed over the course of the manipulation sequence. Let one manipulation $m = (\delta x, \delta y, \delta z, \theta_x, \theta_y, \theta_z)$, where the first three parameters specify a minute displacement of the DLO's end point, and the remaining three parameters specify its rotation.

At the start of the simulation, the length of the DLO is set to a given value and the DLO's root is placed in its fixed pose. Then the following procedure is conducted and the obtained data is registered:

1. Set the initial connector pose p and the length of the DLO,
2. Obtain the DLO geometry s_0 at pose p ,
3. Randomly initialize a manipulation sequence $\tilde{m}_{0:n-1}$,
4. Generate manipulation sequence $m_{0:n-1}$ from $\tilde{m}_{0:n-1}$ via the modulation procedure described below,
5. Obtain the sequence of geometries $s_{1:n}$ that results as the manipulations in $m_{0:n-1}$ are iteratively applied to p .

Item 4 requires further explanation. In this study, it is assumed that the connector should avoid surrounding structures, so the required connector path will not always be a straight line. Therefore, it is necessary to have training data that evokes the kind of movements that are to be avoided. For this reason, we modulate the initial manipulation sequence $\tilde{m}_{0:n-1}$ using the following rule.

$$y = \begin{cases} m \\ 0 \\ m \cdot \sin\left(\frac{2\pi\omega i}{L-1} + \psi\right) \end{cases},$$

where y is the modulated manipulation and L is the number of manipulations. Coefficients ψ and ω control phase and wavelength, respectively, and are randomly generated. By modifying each element of the initial $m_{0:n-1}$ as described above, a variety of manipulation and DLO shape data are generated.

5.4. CF-EMD net training

From the data collected in the procedure explained in the previous subsection, we extract triplets consisting of a DLO shape s_i , a manipulation m_i applied to it, and the resulting shape s_{i+1} . An appropriate number of such triplets are prepared as training data, and used to train the CF-EMD net. The following loss function is used for training.

$$Loss = MSE(s_{i+1:n}, \hat{s}_{i+1:n})$$

Since the DLO geometry is represented by a point chain model, the

above equation calculates the average of the square of the distances between corresponding points in s_{i+1} and $\hat{s}_{i+1:n}$. Adam is used as the optimization algorithm.

6. Real-World cable detection and geometry correction

6.1. Overview

The method described in Section 4 does not guarantee consistency between the DLO shape maintained internally by the planner and the cable in the real world. Therefore, we introduce a process that reduces the gap between the planner's internal DLO representation and the real cable, by fitting the initial DLO geometry s_0 to the real cable's observed shape. For this purpose, we construct our manipulation setup as shown in Fig. 7. In addition to the manipulator that grips the connector, a manipulator equipped with a 3D range image sensor at its tip is prepared to enable highly accurate 3D measurement from arbitrary viewpoints. Ideally, this system would allow us to obtain an accurate point cloud measurement of the cable. However, high-precision sensors generally have trade-offs between the field of view and depth of field, and narrow measurement ranges. Therefore, it is realistic to assume that only a portion of the cable can be measured in our task setup. In order to perform manipulation planning under assumption of such incomplete data, an effective strategy is to recover the cable's overall geometry from the partial data using physical simulation.

We must also account for the possibility that part of the surrounding structures may be included in the measurements. We propose the following procedure. We assume that the root position of the cable is known, and that the robot has already grasped the connector. This means that the system is in a state where the geometry of the DLO can be specified in the physical simulator. The first step is to point the range image sensor at the cable and measure it. Points originating from the surroundings may be obtained at the same time, so we isolate those points in the measurement that originate from the cable. Next, we decide which part of the simulated DLO geometry s_0 corresponds to the resulting point cloud. Then, s_0 is corrected using this information as a constraint. The resulting DLO geometry is then used as s_0 .

This procedure allows the manipulation planning to be performed taking into account both the real cable geometry and the physical constraints of the DLO. In the following subsections, we will explain how we extract the point cloud of the cable and how to correct s_0 , respectively.

6.2. Cable point cloud extraction

As we assume that cable geometry is highly variable, it is difficult to determine which of the measured points correspond to the cable by means of simple pattern matching. Therefore, we adopt a classification approach using PointNet++ [30]. Specifically, the measured point cloud is input to the PointNet++ classifier, to obtain a binary classification of the points that divides the cloud into points originating from the cable and points originating from the surrounding environment.

For this purpose, it is necessary to train PointNet++ in advance. As in the case of manipulation planning, the data collection burden should be reduced as much as possible. However, in our experience, a certain degree of realism is required for the cable point cloud. Therefore, the

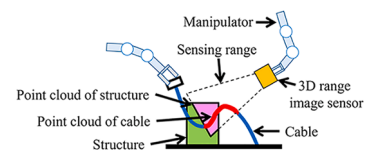


Fig. 7. Configuration of our robotic cable wiring system. The system consists of two manipulators: one for cable-handling and one for range sensing from an appropriate viewpoint.

following method is used to generate data. First, a real cable is straightened, and measured with the range image sensor to obtain point cloud data. This point cloud is then divided into segments of predetermined length. Then, the DLO is generated in various shape geometries using physical simulation. Then, the point cloud segments are arranged along the geometry of each virtual DLO. The resulting data mimics the point cloud data obtained from real cables. A point cloud data set of the surrounding environment is generated via simulation as well. PointNet++ is then trained as a binary classifier for distinguishing the cable from other objects in its input point cloud. Using this neural network, the system performs inference on the unknown point clouds obtained from the 3D range image sensor to extract the points originating from the cable.

6.3. Geometry correction using partial cable data

Suppose that the geometry s_0 of the DLO is initialized via physical simulation. The following procedure corrects the shape of the DLO. First, based on the DLO geometry obtained from the physics simulation, the range image sensor is moved so that at least the middle part of the cable is in the field of view, and the measurement data is obtained as point cloud P_O . Then, the point originating from the cable are isolated using the method described in the previous section. Then, using the ICP algorithm [31], a part of the DLO is matched with the point cloud data. Since a nearest neighbour search is required here, we map the point cloud to the node points of the DLO using Euclidean distance. Then, a rigid body transformation H is obtained to fit the point cloud to the simulated DLO geometry. Let P_R be the transformed point group.

Next, for each node point in the DLO, select the point in P_R with the closest distance. If the distance is less than a predefined threshold, they are considered to be associated. Let Q be the set of the node points for which such correspondence has been obtained. Next, Q is subjected to the inverse transformation of H and moved to the vicinity of P_O (the point set before the rigid body transformation). Then, physical simulation is performed with the points in Q fixed in place, thereby correcting the positions of the remaining nodes. Fig. 8 shows a schematic diagram of the above.

The above procedure corrects the DLO geometry by constraining the intermediate node points in addition to the cable ends, thereby approximating the actual current shape of the real cable.

7. Experiments

7.1. Experimental settings

To confirm the effectiveness of the proposed method, we conducted verification experiments in simulation, and real-world experiments on robot hardware. The pose of the cable root and the geometry of the surrounding structures were given as prior knowledge. The robot used to manipulate the connector was SEIKO-EPSON's N2 [32], a ceiling-suspended six-axis serial link manipulator. The end effector of

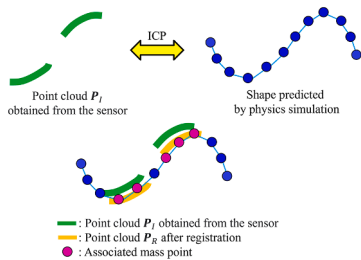


Fig. 8. Correction of DLO geometry using measured point clouds. First, the point cloud is matched to the DLO by looking for corresponding node points. Then, an inverse transformation is performed to obtain a fixed area for the cable correction process.

the robot was a parallel gripper. The tip of the gripper was designed as a jig that can clamp the connector from both sides. In our hardware experiments, the workpiece, which mimicked a laptop PC circuit board, was placed on a horizontal platform. Fig. 9 shows an overview of the experimental platform.

A 3D range image sensor was attached to the wrist of the robot. The specifications of this sensor were as follows: 720×540 pixel resolution, 79.1×59.3 mm measurement range, 10 mm effective measurement height, and 20 mm measurable height. The point cloud data obtained from the sensor was used for connector detection and pose estimation. We implemented Ying et al.'s [14] method to determine the connector pose with sufficient accuracy to be grasped by the robot.

Next, we describe the cables and connectors used for verification. Five cables with lengths of 150 mm, 200 mm, 250 mm, 300 mm, and 350 mm were prepared. Each cable was made by twisting together three wires. The conductor diameter of each wire was 0.65 mm, the outer diameter (including the sheath) was 1.45 mm, and the cross section of the resulting stranded cables was approximately $3 \text{ mm} \times 4.2 \text{ mm}$. In physical simulation, these stranded cables were approximated as single DLOs.

7.2. Appropriateness of the physics simulation

The proposed method collects DLO data using physical simulation, and trains the CF-EMD net on the resulting dataset. Before reporting on real-world cable manipulation, we assess the gap between the physical simulation and reality, and evaluate the correction methods described in Section 6. The parameters for the DLO physics simulation were experimentally selected following [28]. A cable diameter of 4 mm, Young's modulus of 126 MPa, and Poisson's ratio of 0.3 were used.

The verification setup is as follows. Measuring targets points are located at equal spacings along the cable length, dividing it into 10 sections. We bring the real cable in a shape configuration that features substantial bending as well as torsion. With both ends of the cable fixed, we measure the 3D coordinates of the target points manually. We then simulate the cable with the same connector pose and cable length, and calculate the difference in the 3D positions of the target points on the real and simulated cable. The left panel of Fig. 10 shows the definition of the coordinate system for the cable root and connector. In this experiment, the position of the connector was set as $(x, y) = (60, -60)$ mm and a rotation of $\pi/2$ was added along the y axis. The right panel of Fig. 10 shows this example.

Table 1 summarizes the measured position error against cable length as a percentage of the total length. Since both ends of the cable are fixed, the error tended to be larger at larger distances from both ends; for a 350

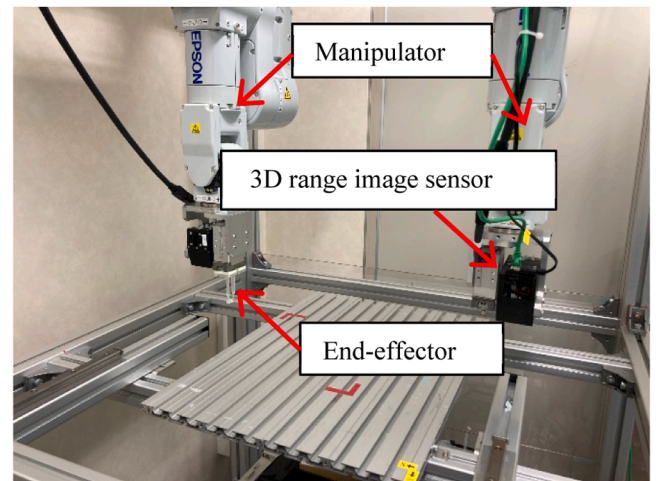


Fig. 9. Appearance of the robot system.

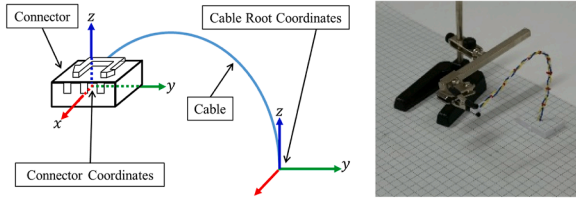


Fig. 10. Left: Definition of the connector coordinate system with respect to the root of the cable. Right: A fixed cable with bending and twisting.

Table 1

Measured differences between real and simulated cables.

Length [mm]	Average[mm(%)]	Max[mm(%)]
150	3.6 (2.4)	7.2 (4.8)
200	6.5 (3.3)	15.5 (7.7)
250	12.0 (4.8)	26.3 (10.5)
300	16.4 (5.5)	36.7 (12.2)
350	24.1 (6.9)	51.2 (14.6)

mm cable, the maximum error was about 50 mm, which is not negligible. The left panel of Fig. 11 plots the error between the simulated and real shape for the set of measurement targets on our five cables, with a maximum misalignment of 45 mm for the 350 mm geometry. The right graph in Fig. 11 shows the error after applying correction as described in Section 6. For the cables up to 300 mm, the error is reduced to less than 10 mm, and even for the 350 mm cable, the error is about 20 mm.

Note that this setup results in a fairly large cable twist. The authors have also verified the difference between simulation and reality under simpler conditions [33]. In this case, the error was less than 5% of the total cable length, and the maximum error was about 15 mm without the corrections described in Section 6. Although this value is not negligible, it can be adequately accommodated by providing a safety margin during manipulation planning.

7.3. Implementation and training of CF-EMD net

The specific structure of the CF-EMD net is as follows: For the encoder, the input is 30×3 dimensional, the output is 30 dimensional, and in between are three fully-connected layers with 30 neurons each. For the decoder, the input has 30 dimensions, the output has 30×3 dimensions, and in between are fully-connected layers with 30, 30, and 90 neurons. The 30×3 dimensionality derives from the fact that the number of DLO nodes was set to 30 and the coordinates of each node are three-dimensional. Meanwhile, the manipulation module accepts 6-dimensional manipulation inputs and outputs 30-dimensional compressed shape predictions. The LSTM consists of three layers with 30-dimensional input each.

Training data for the CF-EMD net was generated using physical simulation as described in Section 5.3. Manipulation values m are

randomly generated using ranges of ± 0.01 m for the displacement quantities (δx , δy , δz) and $\pm \pi/20$ rad for the rotation quantities (θ_x , θ_y , θ_z), respectively. Twenty manipulations were performed on each initial DLO geometry, and the DLO geometries and the manipulation were recorded as a set. The total number of data was 80,000 for each cable length, of which 64,000 were used for training and 16,000 for testing.

Training was performed separately for each cable length. Fig. 12 shows two example learning curves. Although there was a tendency for the learning curve to converge to a larger value as the cable length increased, a straightforward learning curve was obtained in each case, and the final loss was found to be sufficiently small. The test data showed similar loss values as the training data, indicating that no overfitting occurred.

7.4. Wiring experiment

Verification experiments were conducted using the robot system. As shown in Fig. 13, three layouts, (A) to (C), were prepared for the placement of structures, the root of the cable, and the position of the target socket. In all of these layouts, the cables would catch on work space structures if the connectors were simply moved to the sockets in straight lines. The approximate initial pose of the connector side of the cable was also determined, and the experiment was initiated from a pose in the vicinity thereof. The number of manipulations for generation of the connector trajectory was set to 30, and the number of cable node points in the simulation was set to 30.

The experimental procedure was as follows:

1. The 3D range image sensor is used to acquire a point cloud measurement of the workspace, from which we detect the connector and estimate its pose. We then grasp the connector with the manipulator.
2. Using the connector pose estimated in step 1 as input, we perform physical simulation of the cable, determine suitable viewpoints for the sensor on basis of the obtained geometry, and acquire a point cloud of the cable accordingly.
3. Based on the connector pose and the obtained point cloud of the cable, we correct the DLO geometry using the method described in Section 6.
4. Based on the connector pose obtained in step 1 and the DLO geometry obtained in step 3, we generate a manipulation trajectory for the connector using the method described in Section 5.5.
5. Let the manipulator perform the generated manipulation.

Table 2 shows results for the manipulation experiments. The ‘Result’ column shows the outcome classification, which is defined as follows.

- (a) [Success] Wiring succeeded without snagging of the cable
- (b) [Success] Wiring was successful despite some contacts
- (c) [Failure] Cable snagging occurred
- (d) [Failure] The connector was dislodged from the end-effector

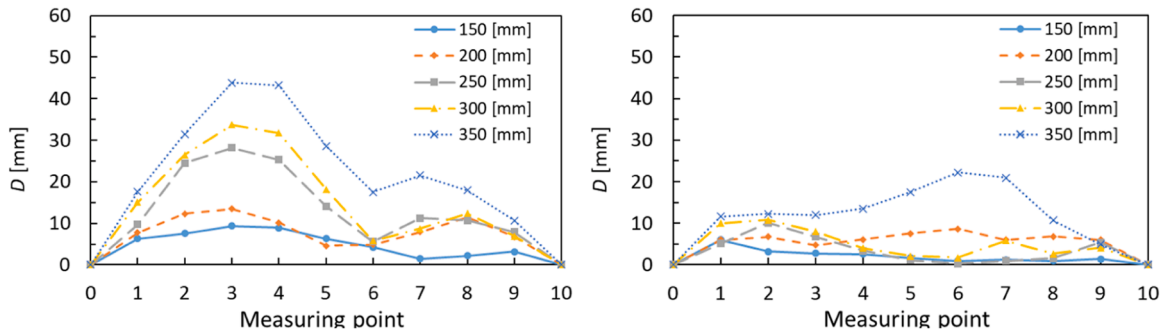


Fig. 11. Difference between measured and simulated values at each measurement point. Left: before correction, right: after correction.

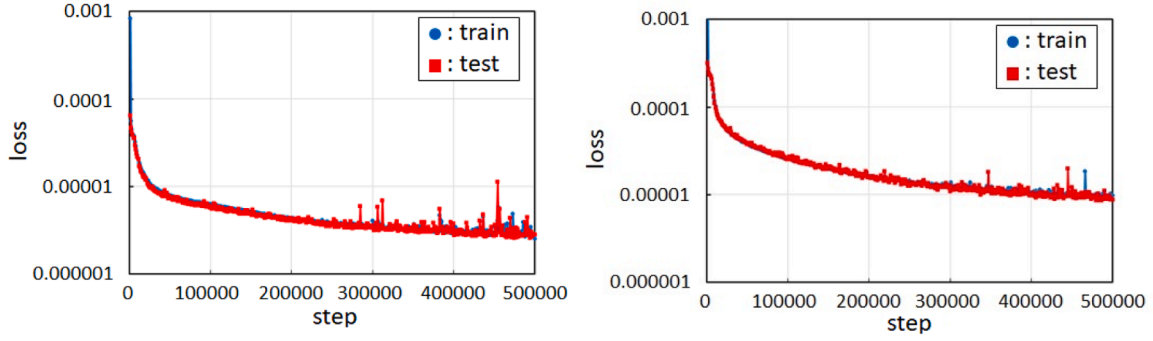


Fig. 12. Learning results of the CF-EMD net. Left: Learning curve for a 150mm-long cable, Right: Learning curve for a 350mm-long cable.

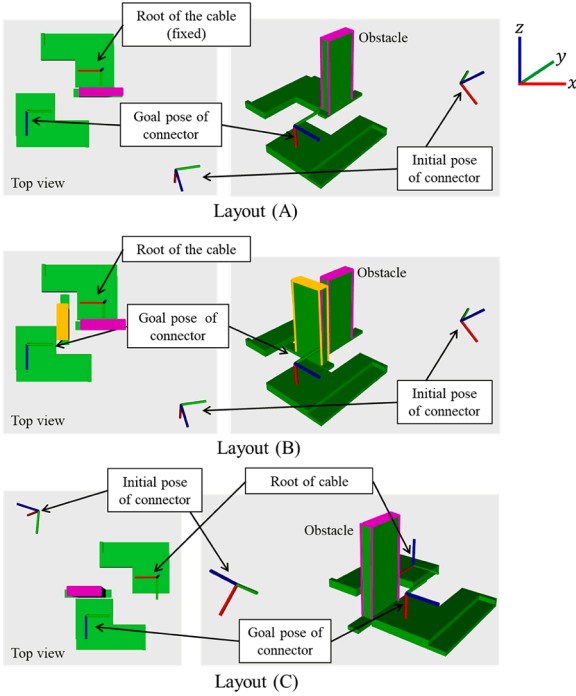


Fig. 13. Three task layouts for the cable wiring experiments.

Table 2
Result of the cable wiring experiment.

Layout	Cable Length	Result	Trajectory Length (proposed)	Trajectory length [24]
(A)	150	(a)	301	357
	200	(a)	245	351
	250	(d)	296	380
	300	(a)	302	471
	350	(a)	332	448
(B)	300	(a)	302	471
	350	(a)	344	457
(C)	150	(a)	194	254
	200	(a)	229	297
	250	(a)	176	309
	300	(a)	202	421
	350	(a)	188	181

We consider a trial a success if the cable did not interfere with task space structures or only touched them. On the other hand, if the cable caught on the structures or the connector got dislodged from the hand, we considered a trial a failure.

As can be seen from the table, in 11 of the 12 cases, the cable was

manipulated without any contact. In the remaining one case, the connector separated from the hand during cable manipulation, although the trajectory was appropriate. This means that in all cases, the system was able to avoid getting the cable caught on any surrounding structures. The reason why there are no results for cables of 150 to 250 mm in layout (B) is that this layout requires longer cables in the first place, a fact we determined in advance using existing methods [33]. The two columns on the right side of the table show the length of the connector trajectory in the generated manipulation plan for the proposed method and an existing method [24]. We observe that the proposed method generated shorter paths than the existing method in all cases but one. As for processing time, the average time taken to isolate the cable using PointNet++ was 3.33 s with a standard deviation of 0.15. The average time for connector trajectory generation using the CF-EMD net was 12.10 s with a standard deviation of 2.88. This processing time consisted of 0.18 s for shape estimation, 1.76 s for computation of the objective function, 3.32 s for calculation of the hand poses, and 6.84 s for updating the manipulation.

Fig. 14 shows data concerning the identification of the cables at step 3 above. PointNet++ was trained on the composite data as shown in the first row and output the appropriate identification results as shown in the second and third rows. Fig. 15 shows examples of generated trajectories. It can be seen that smoother and shorter connector trajectories

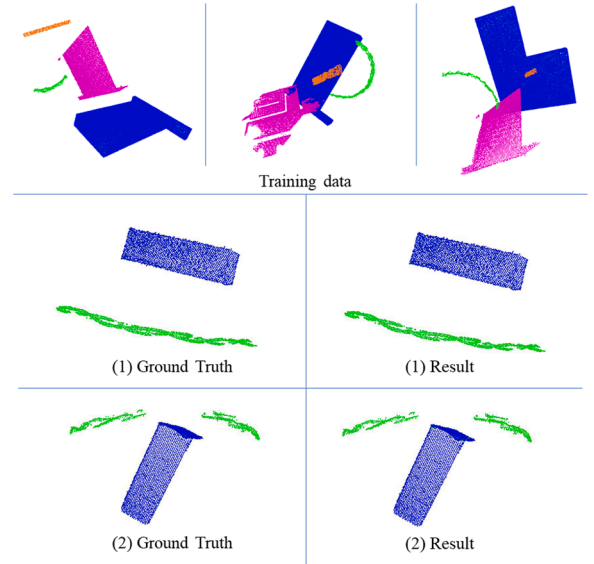


Fig. 14. Example of data concerning the identification of the cables. Top row: Training data, a composite of actual cable data (green), actual data of surrounding environment (pink), simulated data of surrounding components (blue), and simulated noise data (orange). 2nd and 3rd rows: Results of cable identification; in both cases, cable parts (green) were properly separated from other parts (dark blue).

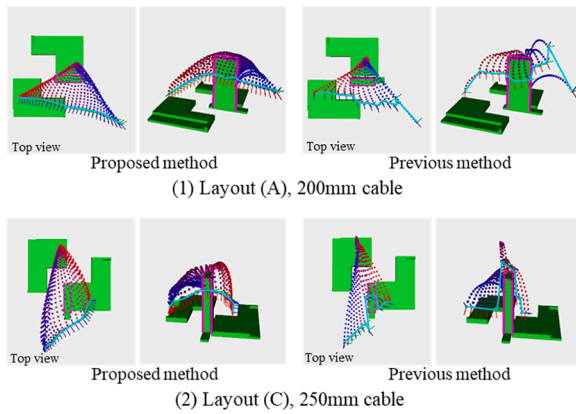


Fig. 15. Examples of generated connector trajectories. The light-blue broken line shows a trajectory to manipulate the connector output by the planner; the previous method produced a feasible but choppy trajectory, while the proposed method produced a smooth trajectory that did not move too far away from obstacles.

are generated compared to the existing method. This is due to the fact that the manipulation trajectory is determined in consideration of the nearby cable geometry and connector location at each point in time, which is characteristic for CF-EMD-based planning.

In other attempts to investigate the behavior of the proposed method, the effect of the number of nodes in the DLO was examined. The number was set to 10, 15, 20, and 30, respectively, and CF-EMD net was trained at each DLO, and manipulation planning were performed using the trained network. The length of the DLO was set to 250 mm, and the two verification environments were Layout (A) and Layout (C) shown in Fig. 13. Ten plans were executed for each environment. The results were as follows. Regardless of the number of nodes, manipulation trajectories were generated without any problems. In Layout (A), the computation time of the objective function decreased by only a few percent when the number of nodes was small, whereas there was almost no difference in Layout (C). Connector path length obtained did not change with the number of nodes in both layouts. These results suggest that the number of nodes does not significantly change the manipulation.

Finally, Fig. 16 shows an example of moving an actual connector to the vicinity of a socket. The robot system succeeded in automatically generating a motion that brings the connector close to the socket position, while avoiding a tall structure. Note that this experiment is automated from the detection of the connector; no human intervention is involved in the series of movements. The experimental movie can be viewed on our project web page: <http://www.ais.shinshu-u.ac.jp/cfemdnet/>

8. Conclusions

In this paper, we described a manipulation planning method for automated wiring of connector-terminated cables with one end mounted on a circuit board. We presented a method for generating a transport trajectory for the floating connector. We let the robot grasp the floating connector, and transport it to its socket while avoiding collision with surrounding structures. Core of the proposed method is the CF-EMD net, a modular neural network. We described its internal structure, motion planning flow, and learning method in detail. We also presented our policy of using physical simulation for training data collection and introduced a method for bridging the gap between reality and simulation. The effectiveness of the proposed method is demonstrated by manipulating stranded cables with lengths ranging from 150 mm to 350 mm in a test environment using ceiling-suspended robot arms. The maximum positional error between real and simulated cable was about 20 mm, indicating that the operation plans produced by the proposed

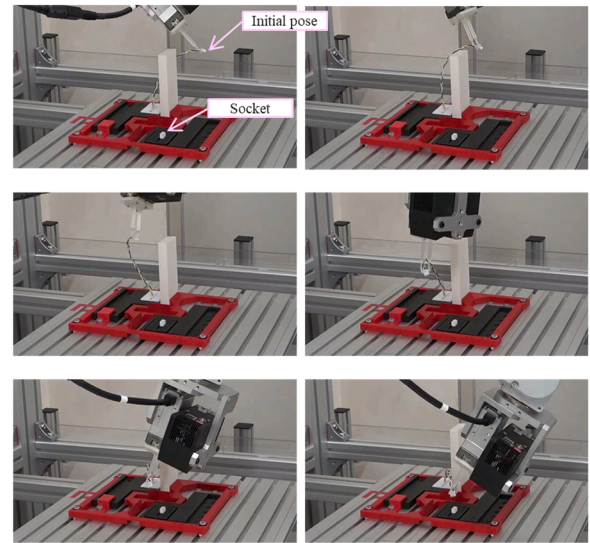


Fig. 16. Cable wiring example.

method are realistically executable. In our hardware experiments, the proposed method was integrated with existing connector detection and pose estimation methods, demonstrating that many parts of the connector wiring task can be automatically performed by a robot.

Future work includes improving the processing speed of CF-EMD net. The results of the proposed method depend on which parts of the cable can be measured. We believe that the introduction of a mechanism for estimating the degree of shape compensation could further improve robustness for real-world cable manipulation. We also plan to extend the simulation environment and CF-EMD net to accommodate cases where cable deformation is affected by gravity. Other prospects include extension to multi-point grasping. Lastly, in order to meet the demand for wiring in more complex environments, we are modifying the proposed method to accommodate grasping of the cable itself in addition to its terminal connector.

CRediT authorship contribution statement

Kimitoshi Yamazaki: Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **Kyoto Nozaki:** Methodology, Software, Validation. **Yuichiro Matsuura:** Conceptualization, Validation. **Solvi Arnold:** Methodology, Software, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Kimitoshi Yamazaki reports financial support was provided by Seiko Epson Corporation.

Data availability

Data will be made available on request.

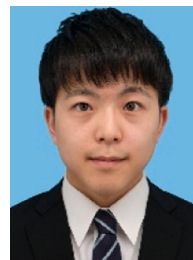
References

- [1] J. Huang, P. Di, T. Fukuda, T. Matsuno, Fault tolerant mating process of electric connectors in robotic wiring harness assembly systems, in: in Proc. of the 7th World Congress on Intelligent Control and Automation, 2008, pp. 2339–2344, <https://doi.org/10.1109/TCST.2009.2034735>.
- [2] F. Chen, F. Cannella, J. Huang, H. Sasaki, T. Fukuda, A study on error recovery search strategies of electronic connector mating for robotic fault-tolerant assembly, *J. Intelligent Robotic Sys.* 81 (2016) 257–271, <https://doi.org/10.1007/s10846-015-0248-5>.

- [3] J. Zhu, B. Navarro, P. Fraisse, A. Crosnier, A. Cherubini, Dual-arm robotic manipulation of flexible cables, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 479–484.
- [4] M. Yu, K. Lv, H. Zhong, X. Li, “Global model learning for large deformation control of elastic deformable linear objects: an efficient and adaptive approach,” *arXiv preprint arXiv:2205.04004*, 2022.
- [5] D. Tanaka, S. Arnold, K. Yamazaki, Disruption-resistant deformable object manipulation on basis of online shape estimation and prediction-driven trajectory correction, *IEEE Robot. Autom. Lett.* 6 (2) (2021) 3809–3816.
- [6] H.C. Song, Y. Kim, D.H. Lee, J. Song, Electric connector assembly based on vision and impedance control using cable connector-feeding system, *J. Mech. Sci. Techn.* 31 (2017) 5997–6003, <https://doi.org/10.1007/s12206-017-1144-7>.
- [7] F. Yumbla, J.S. Yi, M. Abayebas, M. Shafiyev, H. Moon, Tolerance dataset: mating process of plug-in cable connectors for wire harness assembly tasks, *Intell. Serv. Robot.* 13 (2020) 159–168, <https://doi.org/10.1007/s11370-019-00307-5>.
- [8] D. Romeres, D. Jha, W. Yezazunis, D. Nikovski, H. Dau, Anomaly detection for insertion tasks in robotic assembly using gaussian process models, in: *2019 18th European Control Conference (ECC)*, 2019, pp. 1017–1022, <https://doi.org/10.23919/ECC.2019.8795698>.
- [9] Y. Domae, H. Okuda, Y. Kitaaki, Y. Kimura, H. Takauji, K. Sumi, S. Kaneko, 3-D sensing for flexible linear object alignment in robot cell production system, *J. Robotics Mechatron.* 22 (2010) 100–111, <https://doi.org/10.20965/jrm.2010.p0100>.
- [10] Y. Kitaaki, R. Haraguchi, K. Shiratsuchi, Y. Domae, H. Okuda, A. Noda, K. Sumi, T. Fukuda, S. Kaneko, T. Matsuno, A robotic assembly system capable of handling flexible cables with connector, in: *2011 IEEE International Conference on Mechatronics and Automation*, 2011, pp. 893–897, <https://doi.org/10.1109/ICMA.2011.5985708>.
- [11] K. Sumi, Development of production robot system that can handle flexible goods/project for strategic development of advanced robot element technologies /Robot Assembly System for FA equipment, in: *2009 IEEE Workshop on Advanced Robotics and its Social Impacts*, 2009, pp. 42–46, <https://doi.org/10.1109/ARSO.2009.5587079>.
- [12] F. Yumbla, A. Meseret, T. Luong, J. Yi, H. Moon, Preliminary connector recognition system based on image processing for wire harness assembly tasks, in: *2020 20th International Conference on Control, Automation and Systems*, 2020, pp. 1146–1150, <https://doi.org/10.23919/ICCAS50221.2020.9268291>.
- [13] H. Zhou, S. Li, Q. Lu, J. Qian, A practical solution to deformable linear object manipulation: a case study on cable harness connection, *ICARM (2020)* 329–333, 2020.
- [14] C. Ying, Y. Mo, Y. Matsuura, K. Yamazaki, Pose estimation of a small connector attached to the tip of a cable sticking out of a circuit board, *Internat. J. Automat. Techn.* 16 (2) (2022) 208–217.
- [15] R. Qi, L. Yi, H. Su, L. Guibas, PointNet++: deep hierarchical feature learning on point sets in a metric space, *Adv. Neural Informat. Process. Sys.* Vol. 30 (2017).
- [16] K. Sano, S. Iijima, K. Yamazaki, A case study on automated manipulation for hooking wiring of flexible flat cables, in: *2019 IEEE International Conference on Mechatronics and Automation*, 2019, pp. 793–798, <https://doi.org/10.1109/ICMA.2019.8816286>.
- [17] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, E. Adelson, Cable manipulation with a tactile-reactive gripper, *Int J Rob Res* 40 (12–14) (2021) 1385–1401.
- [18] S. Duenser, J.M. Bern, R. Poranne, S. Coros, Interactive robotic manipulation of elastic objects, in: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3476–3481.
- [19] F. Liu, E. Su, J. Lu, M. Li, M.C. Yip, Robotic manipulation of deformable rope-like objects using differentiable compliant position-based dynamics, in: *IEEE Robotics and Automation Letters* 8, 2023, pp. 3964–3971.
- [20] C. Ying, K. Yamazaki, Motion generation for shaping deformable linear objects with contact avoidance using differentiable simulation, in: *Proc. of the 2023 IEEE International Conference on Robotics and Biomimetics*, 2023, pp. 628–635.
- [21] Y. Yang, J.A. Stork, T. Stoyanov, Online model learning for shape control of deformable linear objects, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 4056–4062.
- [22] S. Huo, et al., Keypoint-based planar bimanual shaping of deformable linear objects under environmental constraints with hierarchical action framework, *IEEE Robot. Autom. Lett.* 7 (2) (2022) 5222–5229.
- [23] K. Yamazaki, R. Matsuura, S. Arnold, Generating shape transitions of deformable linear objects using generative adversarial networks, in: *Proc. of the 2022 IEEE International Conference on Mechatronics and Automation (ICMA)*, Guilin, China, 2022, pp. 538–543, August 7–10.
- [24] P. Chang, T. Padif, Sim2Real2Sim: bridging the gap between simulation and real-world in flexible object manipulation, in: *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, Taichung, Taiwan, 2020, pp. 56–62, <https://doi.org/10.1109/IRC.2020.00015>.
- [25] S. Arnold, K. Yamazaki, Fast and flexible multi-step cloth manipulation planning using an encode-manipulate-decode network (EM²D Net), *Frontiers Neurorobotics* 13 (2019) 22, <https://doi.org/10.3389/fnbot.2019.00022>.
- [26] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, *Neural Comput* 9 (8) (1997) 1735–1780, 11.
- [27] Diederik Kingma, Jimmy Ba, Adam: a method for stochastic optimization, *Internat. Conf. Learn. Representat.* 12 (2014).
- [28] N. Lv, J. Liu, X. Ding, J. Liu, H. Lin, J. Ma, Physically based real-time interactive assembly simulation of cable harness, *J. Manuf. Syst.* 43 (3) (2017) 385–399.
- [29] A. Looock, E. Schömer, A virtual environment for interactive assembly simulation: from rigid bodies to deformable cables, in: *Proc. of the 5th World Multiconference on Systemics*, 2001, pp. 325–332.
- [30] C. Qi, L. Yi, H. Su, L. Guibas, PointNet++: deep hierarchical feature learning on point sets in a metric space, *Adv Neural Inf Process Syst* 30 (2017).
- [31] P.J. Besl, N.D. McKay, A method for registration of 3-D shapes, *IEEE Trans. Pattern Anal. Mach. Intelligence* 14 (2) (1992) 239–256.
- [32] Industrial Robot N series, <https://www.epson.jp/products/robots/lineup/6axis/n/> (Accessed on March 7, 2024).
- [33] K. Nozaki, Y. Changjian, Y. Matsuura, K. Yamazaki, Manipulation planning for wiring connector-attached cables considering linear object’s deformability, *Internat. J. Automat. Tech.* 17 (8) (2023) 399–409.



Kimitoshi Yamazaki received BE, ME and Ph.D. degrees from University of Tsukuba in 2002, 2004 and 2007, respectively. From 2006 to 2007, he was a JSPS research fellow. From 2007 to 2012, he was a project assistant professor in the University of Tokyo. From 2010 to 2014, he was also a researcher of PRESTO, JST. He joined Shinshu University in 2012 and is currently a professor. His research interests are in robot vision and motion planning. He is a member of RSJ, SICE, JSME and IEEE.



Kyoto Nozaki received BE and ME degrees from Shinshu University in 2020 and 2022, respectively. He currently works for SEICO-EPSON Corp.



Yuichiro Matsuura currently works for SEICO-EPSON Corp.



Solvi Arnold obtained her Ph.D. degree from the University of Nagoya in 2015. She was a JSPS research fellow from 2013 to 2015. In 2015 she joined Shinshu University, where she is currently a project associate professor. Her research interests include Artificial Intelligence, Intelligent Robotics, and Representation Learning. She is a member of the Robotics Society of Japan.