Proceedings of the 2006 IEEE International Conference on Robotics and Automation
Orlando, Florida - May 2006

Manipulation Planning for Unraveling Linear Objects

Hidefumi Wakamatsu, Akira Tsumaya, and Eiji Arai
Dept. of Materials & Manufacturing Science, Osaka Univ.
2-1 Yamadaoka, Suita, Osaka 565-0871, Japan
{wakamatu, tsumaya, arai}@mapse.eng.osaka-u.ac.jp

Shinichi Hirai
Dept. of Robotics, Ritsumeikan Univ.
1-1-1 Noji Higashi, Kusatsu, Shiga 525-8577, Japan
hirai@se.ritsumei.ac.jp

***Abstract*— A planning method for unraveling manipulation of deformable linear objects is proposed. In manipulation of a linear object, its raveling must be avoided. It takes much time to unravel it once it is raveled. Therefore, it is important to generate unraveling plans efficiently. First, a manipulation process of a linear object including its unraveling is represented as a sequence of its crossing state transitions. Then, possible manipulation processes can be generated once the initial and the objective crossing states are given. Second, qualitative actions to realize manipulation processes are determined. Third, a method for unraveling a linear object as far as possible when its crossing state can not be identified completely is proposed. Finally, an example of unraveling process generation is demonstrated.**

***Index Terms*— linear objects, manipulation, planning, unknotting, unraveling**

I. Introduction

Deformable linear objects such as tubes, cords, cables, wires, and threads are used widely; not only for data transmission or object transportation but also for fixing, fastening, wrapping, packing, suturing, and ligating of objects including themselves. In such manipulative tasks, knotting of linear objects is required. At the same time, raveling of them must be avoided. If unexpected ravel occurs, it takes much time to unravel. For example, raveling of earphone/headphone cord of a portable audio player as shown in Fig.1 would puzzle you sometimes. So, efficient unraveling is important as well as avoidance of such raveling.

Knotting manipulation by robots has been studied. Inoue et al. reported tying a knot in a rope with a manipulator utilizing visual feedback[1]. Hopcroft et al. devised an abstract language to express various knotting manipulations and performed knot-tying tasks with a manipulator[2]. Matsuno et al. realized a task consisting of tying a cylinder with a rope using a dual manipulator system[3]. Morita et al. have been developing a system for knot planning from observation of human demonstrations[4]. Unknotting manipulation, *i.e.*, the inverse of knotting manipulation, has been also studied. We have realized automatic planning and execution of knotting/unknotting manipulation[5]. Ladd et al. developed an untangling planner for mathematical knots represented as closed piecewise linear curves[6].

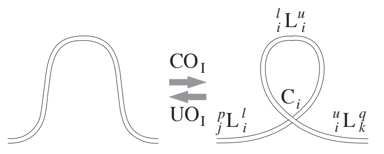
Unraveling is equivalent to unknotting. However, the state of a raveled object can be more complex than that of a knotted object as shown in Fig.1. Moreover, it is difficult to recognize the state of a reveled object completely because it may twine itself. Therefore, recognition of the object state and manipulation planning are both important for unraveling. In this paper, a planning method for unraveling manipulation of linear objects is proposed. First, a manipulation process is represented as a sequence of crossing state transitions. The object state is categorized according to three properties with respect to self-crossings of the object. State transitions are defined by introducing four basic operations. Then, possible manipulation processes can be generated once the initial and the objective crossing states are given. Second, qualitative actions to realize manipulation processes are determined. They consist of grasping points, their directions of movement, and the approach direction of the manipulator to each grasping point. Third, an additional operation is proposed to unravel an object even if its crossing state is not identified completely. By applying it, all identified parts of the object can be unraveled. Finally, an example of unraveling process generation with our developed system is demonstrated.

Fig. 1. Raveled earphone cord and strap

II. Unknotting Manipulation

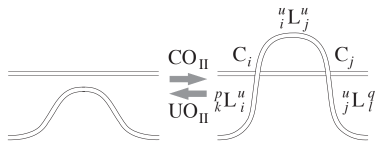
A. Unknotting Processes

In this section, we briefly explain a method to generate possible qualitative manipulation plans for unknotting a linear object. First, the state of a linear object can be topologically represented using three properties after projecting its shape on a projection plane. The first property is the sequence of crossings. It is determined by numbering a crossing met first



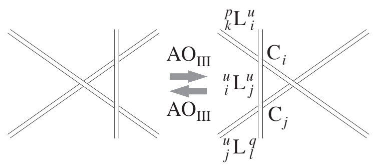
(a-1) uncrossed state (a-2) crossed state

(a) operation I



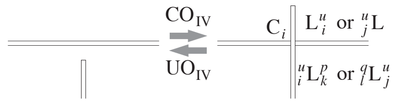
(b-1) uncrossed state (b-2) crossed state

(b) operation II



(c-1) arranged state (c-2) arranged state

(c) operation III



$\mathrm{CO_{IV}}$
$\mathrm{UO_{IV}}$
C_i
L_i^u or $_j^u\mathrm{L}$
$_i^u\mathrm{L}_k^p$ or $_l^q\mathrm{L}_j^u$

(d-1) uncrossed state (d-2) crossed state

(d) operation IV

Fig. 2. Basic operations

with tracing along the projected curve from one endpoint to the other. The i-th crossing is represented as symbol C_i. One endpoint where tracing starts is referred to as the left endpoint E_l and that where tracing ends as the right endpoint E_r. The second property is the location of a pair of points at each crossing, that is, which point is upper/lower. The upper point of i-th crossing is described as symbol C_i^u and the lower point of that as symbol C_i^l. The third property is the helix of each crossing. Let us define a crossing where the upper part overlaps first on the right side of the lower part and then overlaps on its left side as a *left-handed helical crossing*. Conversely, in a *right-handed helical crossing*, the upper part first overlaps on the left side of the lower part and then overlaps on its right side. The symbols C_i^- and C_i^+ represent left- and right-handed helical the i-th crossing, respectively.

Next, we introduce basic operations described in Fig.2, corresponding to state transitions. Crossing operations $\mathrm{CO_I}$, $\mathrm{CO_{II}}$, and $\mathrm{CO_{IV}}$ increases the number of crossings, while uncrossing operations $\mathrm{UO_I}$, $\mathrm{UO_{II}}$, and $\mathrm{UO_{IV}}$ decrease the number. Arranging operation $\mathrm{AO_{III}}$ does not change the number of crossings but permutes their sequence.

Each basic operation can be applied to specific subsequences of crossings. Let us investigate subsequences to which each operation is applicable. Operation $\mathrm{UO_I}$ is applicable to a subsequence represented as follows:

$$\cdots\text{-}\mathrm{C}_i^{u/l}\text{-}\mathrm{C}_i^{l/u}\text{-}\cdots. \tag{1}$$

That is, two crossing points corresponding to one crossing C_i, should be adjacent to each other in applying $\mathrm{UO_I}$. Operation $\mathrm{UO_{II}}$ is applicable to subsequences described as follows:

$$\cdots\text{-}\mathrm{C}_i^{u/l}\text{-}\mathrm{C}_j^{u/l}\text{-}\cdots\text{-}\mathrm{C}_i^{l/u}\text{-}\mathrm{C}_j^{l/u}\text{-}\cdots, \tag{2}$$

$$\cdots\text{-}\mathrm{C}_i^{u/l}\text{-}\mathrm{C}_j^{u/l}\text{-}\cdots\text{-}\mathrm{C}_j^{l/u}\text{-}\mathrm{C}_i^{l/u}\text{-}\cdots. \tag{3}$$

That is, two upper crossing points C_i^u and C_j^u, should be adjacent to each other and the corresponding lower crossing points C_i^l and C_j^l, should also be adjacent to each other. Operation $\mathrm{UO_{IV}}$ is applicable to subsequences represented as follows:

$$\mathrm{E}_l\text{-}\mathrm{C}_i^{u/l}\text{-}\cdots\text{-}\mathrm{C}_i^{l/u}\text{-}\cdots, \tag{4}$$

$$\cdots\text{-}\mathrm{C}_i^{u/l}\text{-}\cdots\text{-}\mathrm{C}_i^{l/u}\text{-}\mathrm{E}_r. \tag{5}$$

That is, a crossing adjacent to the endpoint can be deleted by operation $\mathrm{UO_{IV}}$. Operation $\mathrm{AO_{III}}$ is applicable to a subsequence represented as permutation of the following three subsequences: α, β, and γ, e.g., $\cdots\text{-}\beta\text{-}\gamma\text{-}\alpha\text{-}\cdots$:

$$\alpha: \quad \cdots\text{-}\mathrm{C}_{i/j}^u\text{-}\mathrm{C}_{j/i}^u\text{-}\cdots, \tag{6}$$

$$\beta: \quad \cdots\text{-}\mathrm{C}_{j/k}^{l/u}\text{-}\mathrm{C}_{k/j}^{u/l}\text{-}\cdots, \tag{7}$$

$$\gamma: \quad \cdots\text{-}\mathrm{C}_{i/k}^l\text{-}\mathrm{C}_{k/i}^l\text{-}\cdots. \tag{8}$$

That is, three crossings consisting of three segments one of which overlaps the others can be permuted by operation $\mathrm{AO_{III}}$. Uncrossing operations $\mathrm{UO_I}$, $\mathrm{UO_{II}}$, and $\mathrm{UO_{IV}}$ and arranging operation $\mathrm{AO_{III}}$ are applicable to their specific crossing subsequences indicated above. Once the initial and the objective crossing states of a linear object are given, we can generate possible sequences of crossing state transitions, that, is, possible processes of unknotting manipulation by repeating detection of applicable subsequences of individual operations and deletion/permutation of relevant crossings.

B. Actions to Realize Processes

To realize generated manipulation processes, we have to consider grasping points, their directions of movement, and the approach direction of the manipulator to each grasping point. In this paper, a set of them is defined as an *action*.

First, let us define crossings that are deleted by an uncrossing operation as *target crossings*. A single target crossing consists of two *target points*: an upper target point and a lower target point. We define a segment between two target points in operations $\mathrm{UO_I}$ and $\mathrm{UO_{II}}$ as *a target segment*. In the case of the operation $\mathrm{UO_{IV}}$, a segment between the target point and the endpoint is defined as a target segment. In general, it is difficult for a manipulator to grasp only the upper/lower point at a crossing, which corresponds to the target point. Instead, the target segment is moved by grasping it or its adjacent segment. Consequently, we assume that operations UO_{I} and UO_{II} can be realized by grasping the target segment or its both adjacent segments and that operation UO_{IV} can be realized by grasping the target segment or its adjacent segment.

Next, let us consider the direction of movement of a grasping point. We apply 3 DOF in translation and 3 DOF in rotation to unknotting. Let us define the central axis of a linear object as axis-1, the axis perpendicular to the central axis and along the projection plan as axis-2, and the axis parallel to the projection normal as axis-3. Translations along axis-1, axis-2, and axis-3 are referred to as T_1, T_2, and T_3, respectively. Rotations around axis-1, axis-2, and axis-3 are referred to as R_1, R_2, and R_3, respectively.

Furthermore, we define the approach direction of the manipulator with respect to the projection plane: from the front or from the rear. Whether each operation can be performed is dependent on this direction. Some operations can be performed if the manipulator approaches the appropriate segment from the front and grasps the segment, while others can be performed if the segment is grasped from the rear. Still other operations are unaffected by the direction of approach of the manipulator. Fig.3 illustrates actions for each uncrossing operation. The circles with a cross, circles with a dot, and open circles represent points to be grasped from the front, the rear, and either, respectively.

An arranging operation does not delete any crossing. Let us define a segment with two upper adjacent crossing points as an upper target segment for operation AO_{III} and a segment with two lower adjacent crossing points as a lower target segment for operation AO_{III}. Moreover, we can define a segment with one upper and one lower adjacent crossing points as a middle target segment for operation AO_{III}. Then, we can realize operation AO_{III} by moving an upper, a lower, or a middle target segment. Fig.4 shows all 16 actions for operation AO_{III}. In Fig.4-(a) through (f), (g) through (l), and (m) through (p), an upper, a lower, and a middle target segment is moved, respectively.

Thus, actions to realize each operation can be determined. Consequently, possible qualitative manipulation plans, *i.e.*, sequences of crossing state transitions and actions for each state transition, can be generated on a computer system once the initial and the objective crossing states of a linear object are given.

C. Example of Process Generation

We show an example of planning of unknotting process generation by our developed system. Required manipulation is untying a bowknot with 11 crossings shown in Fig.5. All the following computations were performed on an 833MHz Alpha 21264 CPU with 1GB memory operated by Tru64UNIX and program was compiled with Compaq C Compiler V6.4 with optimization option -O4. When operation AO_{III} is not considered, a graph with 53 states and 153 operations was derived to untie the bowknot, requiring a computation time of about 15 msec. With the inclusion of operations AO_{III}, the generated graph has 932 states and 4282 operations, and it took about 0.5 sec to generate this graph. Thus, we can generate all possible unknotting processes by the system. After that, one process is selected and its actions are determined according to manipulation

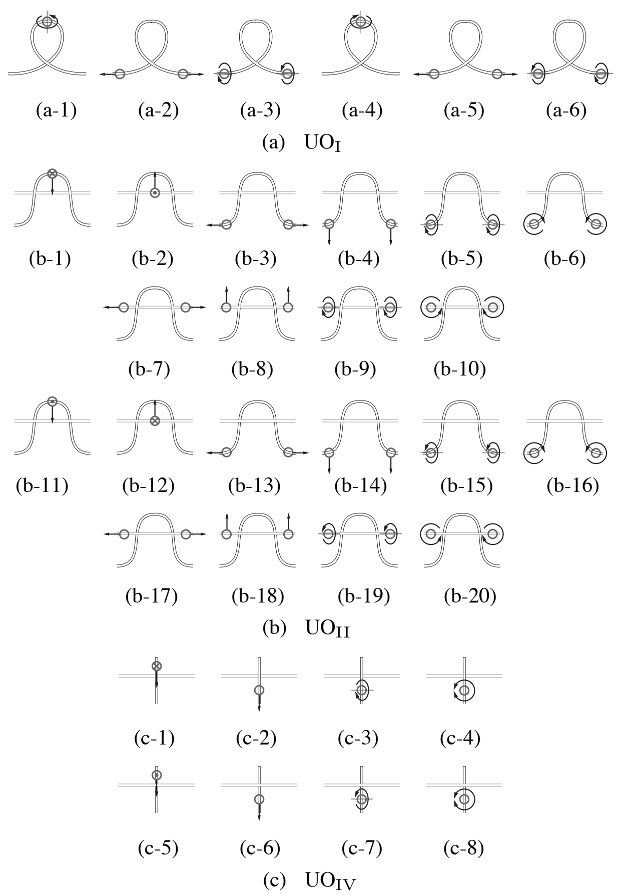


Fig. 3. Actions for uncrossing operations

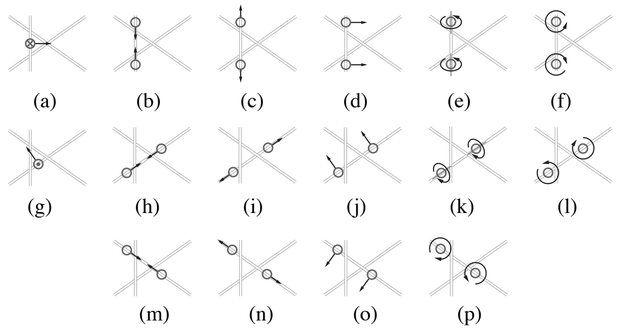


Fig. 4. Actions for arranging operation

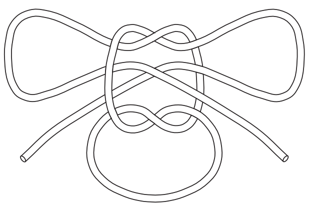
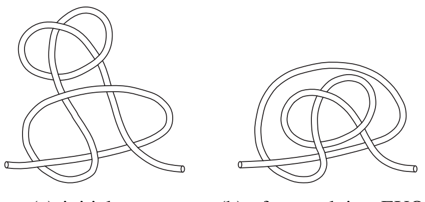


Fig. 5. Bowknot


(a) initial state (b) after applying EUO$_{\rm II}$
Fig. 6. Example of extended uncrossing operation

conditions such as the number of manipulators, DOF of the manipulators, and/or manipulation environment.

III. Unraveling Manipulation

A. Definition of Subknots

In general, the crossing state of a raveled object is more complex than that of a knotted object. It means that the number of possible unraveling processes, *i.e.*, sequences of crossing state transitions becomes very large and it takes much time to generate them. A raveled object sometimes includes some knots, for example, overhand knots in its intermediate parts. Such parts might be unknotted individually and independently of unraveling of the whole object. Moreover, it should be avoided that they are tightened during unraveling. Therefore, we introduce the subknot and the additional operation in this section.

Let us define a continuous subsequence which does not include any crossing with another subsequence as a *subknot*. It is denoted by K$_x$. Note that a subknot includes more than one pair of an upper and a lower crossing point at the same crossing. A procedure to detect subknots in the crossing state description of a knot is as follows. First, search both the upper and the lower crossing points with the same crossing number, *i.e.*, C_i^u and C_i^l. They are referred to as *boundary points*. After that, search the counterpart of crossing points existing between boundary points. If all counterparts exist between them, a subsequence from C_i^u to C_i^l corresponds to a subknot. If counterpart $C_j^{u/l}$ does not exist between boundary points, let it be a new boundary point and search again the counterpart of crossing points existing between the old and the new boundary points. If a detected subknot is equivalent to the knot itself, this indicates that the knot does not include any subknot.

Regarding a subknot as a single segment, uncrossing operations can be applied to subsequences including subknots. In this paper, such uncrossing operations are referred to as *extended uncrossing operations*. Extended uncrossing operation I, *i.e.*, EUO$_{\rm I}$ is applicable to the following subsequence:

$$\cdots\text{-}C_i^{u/l}\text{-}K_x\text{-}C_i^{l/u}\text{-}\cdots. \tag{9}$$

Operation EUO$_{\rm II}$ is applicable to subsequences described as follows:

$$\cdots\text{-}C_i^{u/l}\text{-}K_x\text{-}C_j^{u/l}\text{-}\cdots\text{-}C_i^{l/u}\text{-}K_y\text{-}C_j^{l/u}\text{-}\cdots, \tag{10}$$

$$\cdots\text{-}C_i^{u/l}\text{-}K_x\text{-}C_j^{u/l}\text{-}\cdots\text{-}C_j^{l/u}\text{-}K_y\text{-}C_i^{l/u}\text{-}\cdots. \tag{11}$$

Operation EUO$_{\rm IV}$ is applicable to subsequences represented as follows:

$$E_l\text{-}\ K_x\text{-}C_i^{u/l}\text{-}\cdots\text{-}C_i^{l/u}\text{-}\cdots, \tag{12}$$

$$\cdots\text{-}C_i^{u/l}\text{-}\cdots\text{-}C_i^{l/u}\text{-}\ K_x\text{-}E_r. \tag{13}$$

Operation EAO$_{\rm III}$ is applicable to a subsequence represented as permutation of the following three subsequences: α', β', and γ':

$$\alpha':\ \cdots\text{-}C_{i/j}^u\text{-}K_x\text{-}C_{j/i}^u\text{-}\cdots, \tag{14}$$

$$\beta':\ \cdots\text{-}C_{j/k}^{l/u}\text{-}K_y\text{-}C_{k/j}^{u/l}\text{-}\cdots, \tag{15}$$

$$\gamma':\ \cdots\text{-}C_{i/k}^l\text{-}K_z\text{-}C_{k/i}^l\text{-}\cdots. \tag{16}$$

Note that operation EUO$_{\rm I}$ counterchanges only the location of all crossing points included in a subknot, while operations EUO$_{\rm II}$, EUO$_{\rm IV}$, and EAO$_{\rm III}$ do not change the crossing state of the subknot. Fig.6-(a) shows an example of a knot including a subknot. Its initial crossing state is described as E_l-C_1^{u-}-C_2^{l-}-C_3^{l+}-C_4^{u+}-C_5^{u-}-C_1^{l-}-C_2^{u-}-C_5^{l-}-C_6^{u+}-C_7^{l+}-C_8^{u+}-C_6^{l+}-C_7^{u+}-C_8^{l+}-C_4^{l+}-C_3^{u+}-E_r. Subknot K$_1$ corresponds to C_6^{u+}-C_7^{l+}-C_8^{u+}-C_6^{l+}-C_7^{u+}-C_8^{l+}. If only uncrossing operations are considered, two operations UO$_{\rm IV}$ can be applied to the initial state. Including extended uncrossing operations, operation EUO$_{\rm II}$ also can be applied. Fig.6-(b) shows the crossing state after applying operation EUO$_{\rm II}$.

B. Introduction of Arranging Operation V

It is difficult to move a target segment including a subknot without changing the crossing state of the subknot. So, it is prefer to grasp and move segments which does not include any subknot to perform extended uncrossing operations. In this paper, the fifth operation is introduced, which is referred to as *arranging operation V*, denoted by AO$_{\rm V}$. Operation AO$_{\rm V}$ permutes crossing point $C_i^{u/l}$ and subknot K$_x$ as follows:

$$\cdots\text{-}C_i^{u/l}\text{-}K_x\text{-}\cdots\ \Rightarrow\ \cdots\text{-}K_x\text{-}C_i^{u/l}\text{-}\cdots. \tag{17}$$

This means that segments adjacent to crossing point $C_i^{l/u}$, which corresponds to the counterpart of crossing point $C_i^{u/l}$, pass over/under the subknot as shown in Fig.7. By applying operation $\mathrm{AO_V}$, a subknot in a target segment can be moved into its adjacent segments. If operation $\mathrm{AO_V}$ is applied to a subsequence described by (9), it is permuted as follows:

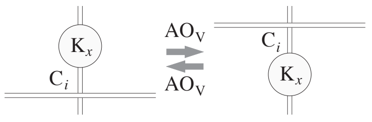
$$\cdots\text{-K}_x\text{-C}_i^{u/l}\text{-C}_i^{l/u}\text{-}\cdots. \tag{18}$$

This subsequence is equivalent to that for operation $\mathrm{UO_I}$ described by (1) because subknot K_x is negligible. This implies that operation $\mathrm{EUO_I}$ corresponds to the combination of operation $\mathrm{AO_V}$ and operation $\mathrm{UO_I}$. Moreover, in the latter case, the crossing state of subknot K_x is not changed. Operations $\mathrm{EUO_{II}}$, $\mathrm{EUO_{IV}}$, and $\mathrm{EAO_{III}}$ are also equivalent to the combination of operation $\mathrm{AO_V}$ and operations $\mathrm{UO_{II}}$, $\mathrm{UO_{IV}}$, and $\mathrm{AO_{III}}$, respectively. Fig.8 shows an example of combination of operations $\mathrm{AO_V}$ and $\mathrm{UO_{II}}$. Its initial state is equivalent to that shown in Fig.6-(a). First, segments adjacent to crossing point C_5^{u-} are passed over subknot K_1 by operation $\mathrm{AO_V}$. Then, a region to which operation $\mathrm{UO_{II}}$ is applicable appears as shown in Fig.8-(b). After applying $\mathrm{UO_{II}}$, the crossing state becomes that after applying $\mathrm{EUO_{II}}$ shown Fig.6-(b). Thus, we can unravel knots without moving subknots by applying operations I through V instead of extended operations I through IV.

By introducing the subknot and operation $\mathrm{AO_V}$, the number of possible state transitions can be decreased. For example, the crossing state graph for untying a bowknot has 932 states and 4282 operations as mentioned before. The bowknot includes subknot K_1, which corresponds to an overhand knot. Regarding it as a segment, a graph with 346 states and 1505 operations was derived. In this graph, the final state includes subknot K_1. A graph to untie subknot K_1 consists of 4 states and 7 operations. Consequently, 349 states and 1512 operations were derived when subknots are considered. Thus, we can reduce possible state transitions if a knot includes some subknots.

Moreover, operation $\mathrm{AO_V}$ can change the crossing state of a linear object into that including subknots alone, that is, the state where all crossings are included in one of subknots. A subknot has geometrically two segments connected its outside, which are referred to as *boundary segments*. When a vision system recognizes the current crossing state of a linear object by tracing segments of its binarized and thinned image, there may be some parts which can not be identified. If such part has two boundary segments, we can regard it as a subknot. Then, using five basic operations proposed in this paper, the object can be unraveled until it includes subknots alone even if the state of those subknots is unidentified. For example, let us assume that subknot K_1 included in a knot shown in Fig.6-(a) can not be identified. Then, this knot can be incompletely unraveled as follows:

$$\begin{aligned}
&\text{E}_l\text{-C}_1^{u-}\text{-C}_2^{l-}\text{-C}_3^{l+}\text{-C}_4^{u+}\text{-C}_5^{u-}\text{-}\\
&\qquad\text{-C}_1^{l-}\text{-C}_2^{u-}\text{-C}_5^{l-}\text{-K}_1\text{-C}_4^{l+}\text{-C}_3^{u+}\text{-E}_r
\end{aligned}$$



(a) arranged state (b) arranged state

Fig. 7. Arranging operation V

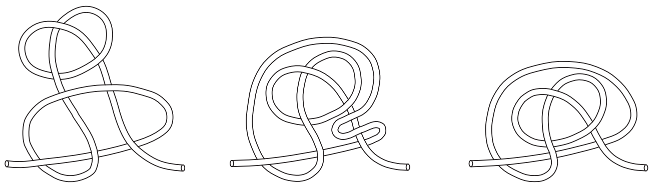

(a) initial state (b) after applying $\mathrm{AO_V}$ (c) after applying $\mathrm{UO_{II}}$

Fig. 8. Combination of $\mathrm{AO_V}$ and $\mathrm{UO_{II}}$

$$\begin{aligned}
&\Rightarrow \text{E}_l\text{-C}_1^{u-}\text{-C}_2^{l-}\text{-C}_3^{l+}\text{-C}_4^{u+}\text{-C}_5^{u-}\text{-}\\
&\qquad\text{-C}_1^{l-}\text{-C}_2^{u-}\text{-K}_1\text{-C}_5^{l-}\text{-C}_4^{l+}\text{-C}_3^{u+}\text{-E}_r\\
&\Rightarrow \text{E}_l\text{-C}_1^{u-}\text{-C}_2^{l-}\text{-C}_3^{l+}\text{-C}_1^{l-}\text{-C}_2^{u-}\text{-K}_1\text{-C}_3^{u+}\text{-E}_r\\
&\Rightarrow \text{E}_l\text{-C}_1^{u-}\text{-C}_2^{l-}\text{-C}_3^{l+}\text{-C}_1^{l-}\text{-C}_2^{u-}\text{-C}_3^{u+}\text{-K}_1\text{-E}_r\\
&\Rightarrow \text{E}_l\text{-C}_1^{u-}\text{-C}_1^{l-}\text{-K}_1\text{-E}_r\\
&\Rightarrow \text{E}_l\text{-K}_1\text{-E}_r,
\end{aligned} \tag{19}$$

where operations $\mathrm{AO_V}$, $\mathrm{UO_{II}}$, $\mathrm{AO_V}$, $\mathrm{UO_{II}}$, and $\mathrm{UO_I}$ are applied in this order.

C. Consideration of Tightenability

Let us define a subknot which can be unknotted completely by applying operations $\mathrm{UO_I}$, $\mathrm{UO_{II}}$, and/or $\mathrm{AO_{III}}$ as an *untightenable subknot* [5]. As shown in Fig.3-(a-2) and (a-5), operation $\mathrm{UO_I}$ can be performed by actions grasping both segments adjacent to the target segment and applying translation T_1, *i.e.*, translation along the central axis of a linear object, in opposite directions. There also be such actions for operation $\mathrm{UO_{II}}$ as shown in Fig.3-(b-3), (b-7), (b-13), and (b-17) and those for operation $\mathrm{AO_{III}}$ as shown in Fig.4-(c), (i), and (n). This indicates that an untightenable subknot can be unknotted completely by pulling both boundary segments in opposite directions regardless of its applying order of operations $\mathrm{UO_I}$, $\mathrm{UO_{II}}$, and/or $\mathrm{AO_{III}}$ for unknotting. Consequently, generation of unknotting processes for untightenable subknots can be omitted.

Contrary, a subknot in which some crossings remain even if all possible operations $\mathrm{UO_I}$, $\mathrm{UO_{II}}$, and $\mathrm{AO_{III}}$ are applied is defined as a *tightenable subknot*. The tightenable subknot is tied tightly by pulling both boundary segments in opposite directions. We must avoid to tighten such subknot for unraveling a knot. For unraveling a tightenable subknot completely, operation $\mathrm{UO_{IV}}$ has to be applied it. This means that the tightenable subknot must be moved to the left or the right terminal of a knot for unraveling. We can move such subknot to the terminals by applying operation $\mathrm{AO_V}$. If a knot consists of multiple tightenable subknots, we unknot such subknots beginning with the subknot nearest to the terminals. Thus, operation $\mathrm{AO_V}$ is quite useful for unraveling a knot including subknots.

IV. Case Study

In this section, we demonstrate an example of unraveling process generation. Fig.9 shows a linear object to unravel. Its crossing state is described as follows:

$$\begin{aligned}&\mathrm{E}_l\text{-}\mathrm{C}_1^{l-}\text{-}\mathrm{C}_2^{l+}\text{-}\mathrm{C}_3^{u-}\text{-}\mathrm{C}_4^{u-}\text{-}\mathrm{C}_5^{l-}\text{-}\mathrm{C}_6^{u+}\text{-}\mathrm{C}_7^{l+}\text{-}\mathrm{C}_5^{u-}\text{-}\\ &\text{-}\mathrm{C}_4^{l-}\text{-}\mathrm{C}_7^{u+}\text{-}\mathrm{C}_6^{l+}\text{-}\mathrm{C}_8^{u+}\text{-}\mathrm{C}_2^{u+}\text{-}\mathrm{C}_9^{u+}\text{-}\mathrm{C}_{10}^{l+}\text{-}\mathrm{C}_1^{u-}\text{-}\mathrm{C}_8^{l+}\text{-}\\ &\text{-}\mathrm{C}_3^{l-}\text{-}\mathrm{C}_{11}^{u+}\text{-}\mathrm{C}_{12}^{u+}\text{-}\mathrm{C}_{13}^{l+}\text{-}\mathrm{C}_{14}^{u+}\text{-}\mathrm{C}_{12}^{l+}\text{-}\mathrm{C}_{13}^{u+}\text{-}\mathrm{C}_{14}^{l+}\text{-}\mathrm{C}_{15}^{l+}\text{-}\\ &\text{-}\mathrm{C}_{16}^{u+}\text{-}\mathrm{C}_{10}^{u+}\text{-}\mathrm{C}_9^{l+}\text{-}\mathrm{C}_{16}^{l+}\text{-}\mathrm{C}_{17}^{u+}\text{-}\mathrm{C}_{18}^{l+}\text{-}\mathrm{C}_{19}^{l-}\text{-}\mathrm{C}_{20}^{u-}\text{-}\mathrm{C}_{21}^{l-}\text{-}\\ &\text{-}\mathrm{C}_{19}^{u-}\text{-}\mathrm{C}_{18}^{u+}\text{-}\mathrm{C}_{17}^{l+}\text{-}\mathrm{C}_{20}^{l-}\text{-}\mathrm{C}_{21}^{u-}\text{-}\mathrm{C}_{15}^{u+}\text{-}\mathrm{C}_{11}^{l+}\text{-}\mathrm{E}_r.\end{aligned} \tag{20}$$

Then, a graph with 2502 states and 12028 operations was generated. This computation took about 16 sec. This knot includes three subknots as follows:

$$\mathrm{K}_1 : \mathrm{C}_4^{u-}\text{-}\mathrm{C}_5^{l-}\text{-}\mathrm{C}_6^{u+}\text{-}\mathrm{C}_7^{l+}\text{-}\mathrm{C}_5^{u-}\text{-}\mathrm{C}_4^{l-}\text{-}\mathrm{C}_7^{u+}\text{-}\mathrm{C}_6^{l+}, \tag{21}$$

$$\mathrm{K}_2 : \mathrm{C}_{12}^{u+}\text{-}\mathrm{C}_{13}^{l+}\text{-}\mathrm{C}_{14}^{u+}\text{-}\mathrm{C}_{12}^{l+}\text{-}\mathrm{C}_{13}^{u+}\text{-}\mathrm{C}_{14}^{l+}, \tag{22}$$

$$\begin{aligned}\mathrm{K}_3 : &\mathrm{C}_{17}^{u+}\text{-}\mathrm{C}_{18}^{l+}\text{-}\mathrm{C}_{19}^{l-}\text{-}\mathrm{C}_{20}^{u-}\text{-}\mathrm{C}_{21}^{l-}\text{-}\\ &\text{-}\mathrm{C}_{19}^{u-}\text{-}\mathrm{C}_{18}^{u+}\text{-}\mathrm{C}_{17}^{l+}\text{-}\mathrm{C}_{20}^{l-}\text{-}\mathrm{C}_{21}^{u-}.\end{aligned} \tag{23}$$

Subknots K_1, K_2, and K_3 correspond to a figure of eight knot, an overhand knot, and a slipknot, respectively. Regarding these subknots as segments, the system generated a graph with 477 states and 1950 operations. Computation time was about 2 sec. Then, the final state has these subknots alone as shown in Fig.10. Thus, we can unravel the object until this state even if all subknots can not be identified completely. Next, the system checked their tightenability. Subknots K_1 and K_2 do not include any subsequence described by (1) through (3) or that represented as permutation of (6) through (8). So, they are tightenable. Subknot K_3 can be unknotted completely by applying operations $\mathrm{UO_{II}}$, $\mathrm{UO_{II}}$, and $\mathrm{UO_I}$ in this order, for example. This implies that subknot K_3 is untightenable and it can be omitted. Then, a graph with 23 states and 59 operations was derived to untying the knot consisting of subknots K_1 and K_2, requiring a computation time of about 0.9 sec. Thus, we can generate possible unraveling processes more efficiently when the subknot, operation $\mathrm{AO_V}$, and the tightenability are considered.

In our proposed method, only the topology of a linear object is considered and its physical properties are ignored. This means that the object can be stretched, bent, and twisted infinitely to perform operations, especially, operation $\mathrm{AO_V}$. To verify the feasibility of generated plans, static/dynamic simulation of linear object deformation is required. Moreover, state recognition is also important. To unravel an object completely, all subknots included in it must be identified completely. They are our future works.

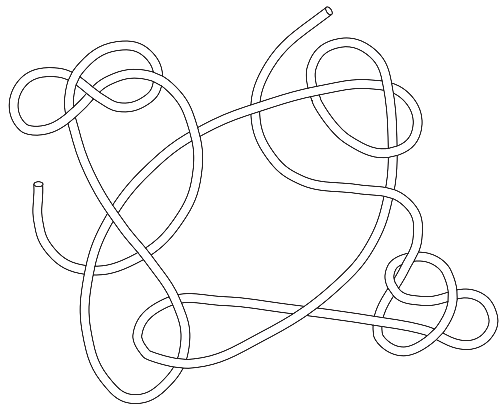


Fig. 9. Example of raveled object

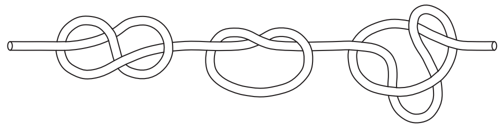


Fig. 10. State consisting of subknots alone

V. Conclusions

A planning method for unraveling manipulation of linear objects was proposed. First, a qualitative manipulation plan is represented as a sequence of crossing state transitions and actions to realize each transition. Possible manipulation plans can be generated when the initial and the objective crossing states and manipulation conditions are given. Next, the additional arranging operation was proposed. It is useful for unraveling a linear object of which crossing state is not identified completely. Finally, a planning example with our developed system demonstrated that we can generate unraveling processes efficiently.

References

[1] H. Inoue and M. Inaba, "Hand-eye Coordination in Rope Handling", Robotics Research: The First International Symposium, MIT Press, pp.163–174, 1984.

[2] J. E. Hopcroft, J. K. Kearney, and D. B. Krafft, "A Case Study of Flexible Object Manipulation", Int. J. of Robotics Research, Vol.10, No.1, pp.41–50, 1991.

[3] T. Matsuno, T. Fukuda, and F. Arai, "Flexible Rope Manipulation by Dual Manipulator System Using Vision Sensor", Proc. of International Conference on Advanced Intelligent Mechatronics, pp.677–682, 2001.

[4] T. Morita, J. Takamatsu, K. Ogawara, H. Kimura, and K. Ikeuchi, "Knot Planning from Observation", Proc. of IEEE Int. Conf. Robotics and Automation, pp.3887–3892, 2003.

[5] H. Wakamatsu, A. Tsumaya, E. Arai, and S. Hirai, "Manipulation Planning for Knotting/Unknotting and Tightly Tying of Deformable Linear Objects", Proc. of IEEE Int. Conf. Robotics and Automation, pp.2516-2521, 2005.

[6] A. M. Ladd and L. E. Kavraki, E, "Using Motion Planning for Knot Untangling", Int. J. of Robotics Research, Vol.23, No.7–8, pp.797–808, 2004.