Knotting/Unknotting Manipulation of Deformable Linear Objects

Hidefumi Wakamatsu
Eiji Arai
Department of Materials and Manufacturing Science
Graduate School of Eng.
Osaka University
Suita, Osaka 565-0871, Japan
wakamatu@mapse.eng.osaka-u.ac.jp

Shinichi Hirai
Department of Robotics
Ritsumeikan University
Kusatsu, Shiga 525-8577, Japan

Abstract

Here, we propose a planning method for knotting/unknotting of deformable linear objects. First, we propose a topological description of the state of a linear object. Second, transitions between these states are defined by introducing four basic operations. Then, possible sequences of crossing state transitions, i.e. possible manipulation processes, can be generated once the initial and the objective states are given. Third, a method for determining grasping points and their directions of movement is proposed to realize derived manipulation processes. Our proposed method indicated that it is theoretically possible for any knotting manipulation of a linear object placed on a table to be realized by a one-handed robot with three translational DOF and one rotational DOF. Furthermore, criteria for evaluation of generated plans are introduced to reduce the candidates of manipulation plans. Fourth, a planning method for tying knots tightly is established because they fulfill their fixing function by tightening them. Finally, we report knotting/unknotting manipulation performed by a vision-guided system to demonstrate the usefulness of our approach.

KEY WORDS—linear objects, deformable, manipulation, planning, knotting, unknotting, tightening

1. Introduction

People have been using knots since the Paleolithic era. In the past, knots were needed to fix structures made of timber and stones. They have always been especially important in sailing, for raising and trimming sails, mooring, etc., and medieval European sailors had to know how to tie various knots. Knots were developed more than 1,000 years earlier in China than in Europe. The Chinese knots have a three-dimensional and symmetrical structure, and are not only practical but also decorative. Such Chinese knots were introduced into ancient Korea and Japan. Feudal warlords in Japan practiced tea ceremony, and their tea powder was stored in pots. To prevent the danger of poisoning by an assassin, complex knots were developed for sealing these pots. Only a trusted manager knew how to tie/untie the sealing knot and he/she would easily notice if the shape of the knot had been changed. Various decorative knots have since been derived from these secure knots, and are still used in tea ceremony or in incense burning. Knots were also used instead of letters or figures in some ancient civilizations, for example, in the Inca Empire.

Nowadays, we use several types of knot, to fasten clothes or shoes, to wrap gifts, for sewing, camping, angling, and climbing. In the apparel industry, knots are used to attach buttons, beads, or sequins onto clothes. In the medical field, they are essential for suturing or ligating organs and tissues. Moreover, in the Japanese food industry, they are used for binding *kinchaku*—a fried bean-curd pouch stuffed with rice cake or diced vegetables—or *kobumaki*—roe or filleted fish rolled with kelp—with gourd shavings.

Thus, knots are "tied" closely to our life and culture, and many types of knot have been developed in different parts of the world (Budworth 2002). Tying such knots requires dexterity; for example, to tie a bowknot, we manipulate a linear object using several fingers of both hands, bending, twisting, holding, and binding it. However, such procedure is dependent on our physical structure and experience, and many other procedures may exist. Modeling of knotting/unknotting processes of a linear object would be useful for analyzing the human dexterity involved in these processes. Furthermore, such modeling will be useful to design a knotting/unknotting

The International Journal of Robotics Research
Vol. 25, No. 4, April 2006, pp. 371-395
DOI: 10.1177/0278364906064819
©2006 SAGE Publications

system with mechanisms different from those used by the human arms/hands. In this paper, a planning method for knotting/unknotting of deformable linear objects is proposed.

First, a topological description of the state of a linear object is given. The topological states can be represented as finite crossing states, including the sequence of crossings and their properties. Second, transitions between the states are defined by introducing four basic operations. A state transition corresponds to a basic operation changing the number of crossings or permuting their sequence. Then, possible sequences of crossing state transitions, i.e., possible manipulation processes, can be generated once the initial and the objective states are given. Third, a method for determining grasping points and their directions of movement is proposed to realize the derived manipulation processes. Furthermore, criteria for evaluation of generated plans are introduced to reduce the number of candidate manipulation plans. Fourth, a planning method for tying knots tightly is established because knots fulfill their fixing function only when tight. Finally, we describe knotting/unknotting manipulation performed by a vision-guided system to confirm the usefulness of our approach.

1.1. Related works

There has been a great deal of research regarding the modeling of linear object deformation; FEM, a particle-based approach, the Cosserat theory, and differential geometry have been applied. The deformed shape of a thread suspended by two points has been analyzed using calculus of variations, and shown to be described by a catenary (Irvine 1981). The deformation of clothes has also been described using catenaries (Weil 1986). These approaches, however, ignore the material properties, and consider only the mass. The deformed shape of threads in a fabric has been described geometrically (Leaf 1960). In computer graphics, the particle-based approach has been applied to simulate the motion of hairs. Rosenblum et al. (1991) described flexure and extension of hairs, while Daldegan et al. (1993) described flexure and torsion of hairs, implying that flexure, torsion, and extension of a linear object can be described using the particle-based approach. Recently, a fast algorithm was introduced to describe linear object deformation using the Cosserat formulation (Pai 2002). Cosserat elements possess six degrees of freedom—three for translational displacement and three for rotational displacement. Flexure, torsion, and extension of a linear object can be described using Cosserat elements. In differential geometry, curved lines in 2D or 3D space have been studied to describe their shapes mathematically (Gray 1993). Moll et al. (2005) have proposed a method to compute the stable shape of a linear object under some geometrical constraints quickly based on differential geometry. It can be applied to path planning for flexible wires. Kinematic and dynamic modeling of a hyper-redundant/hyper-flexible manipulator, such as a cable, have been proposed using differential geometry (Chirikjian and Burdick 1994; Mochiyama and Suzuki 2003). Differential geometry can describe flexure of a linear object, but not extension along or torsion around the object. We established an alternative modeling method based on an extension of differential geometry (Wakamatsu et al. 2004), which describes linear object deformation according to four functions: flexure expressed by two functions, torsion, and extension.

In the medical field, modeling not only of organs/tissues but also of sutures, needles, and catheters is important for simulation of surgery. Spline-based modeling has been applied to the real-time simulation of soft tissues as well as sutures in surgery (Kühnapfel et al. 2000). Nienhuys and van der Stappen (2004) have proposed a computational method for simulating needle insertions considering stick-slip friction between the needle and tissue. As needles have beveled (i.e., asymmetrical) tips, they may bend during insertion. Webster et al. (2004) have modeled this needle bending using a bicycle with a fixed front wheel angle as a nonholonomic model.

Planning methods for deformable linear object manipulation have been proposed (Henrich and Wörn 2000). Insertion of a wire into a hole has been analyzed using a beam model of the wire to derive a strategy to perform the insertion successfully (Zheng et al. 1991; Nakagaki et al. 1997). Sensor-based dynamic insertion of a wire has been investigated (Yue and Henrich 2002). Inverse problems in the manipulation of a linear object have been solved using an object model computed in parallel on a cluster system (Remde and Henrich 2000). The majority of manipulative tasks, including grasping and assembly, are performed through mechanical contact. As rigid object manipulation can be represented as a sequence of finite contact states, planning methods using contact state graphs have been studied (Lozano-Pérez et al. 1984; Desai and Volz 1989). These methods have been applied to the planning of manipulation or assembly. Recently, there has been interest in the development of a systematic approach to the planning of deformable object manipulation. A qualitative representation method of thin object manipulation in 2D space was proposed based on the contact state of a thin object (Wakamatsu et al. 2001). This method can be applied to linear object manipulation in 2D space. Henrich et al. (1999) and Acker and Henrich (2005) classified the contact state of a linear object in 3D space to describe its assembly/disassembly tasks.

Knot tying is a linear object manipulation task with certain characteristics. Phillips et al. (2002) have simulated knot tying in a rope using a particle-based model of the rope. In the present study, the loosely knotted shape of a rope was given as the initial state and its tightening was simulated. Knots have been studied in the field of mathematics, and knot theory provides a topological classification and description of knots of a thread (Adams 1994). This classification and description are useful for the analysis of knots, but it considers only looped threads. Ladd and Kavraki (2004) developed an untangling planner for mathematical knots represented as closed piecewise linear curves. Knotting manipulation by robots has been studied. Inoue and Inaba (1984) reported tying a knot in a rope with a manipulator utilizing visual feedback. Hopcroft et al. (1991) devised an abstract language to express various knotting manipulations and performed knot-tying tasks with a manipulator without detailed trajectory input. Matsuno et al. (2001) realized a task consisting of tying a cylinder with a rope with a dual manipulator system identifying the rigidity of the rope from visual information. In these studies, the method for tying the knot is given in advance. To tie an overhand knot, we first form a loop in a linear object and then pass one endpoint through the loop. Such procedures for tying were given in the above studies. Morita et al. (2003) have been developing a system for knot planning from observation of human demonstrations. In this study, plans could not be derived automatically; human demonstrations were required for task planning, and derived plans were dependent on human physical structure and experience.

The planning method proposed here is based on knot theory, but we extend it to real knots. Mathematical knots are looped and do not have any endpoint. They are topologically categorized by the number of crossings and their properties. The topology of any knot never changes unless it is cut, generated/deleted crossings, and reconnected again. In the case of real knots, their shape also can be represented topologically using crossings. However, they have two endpoints. This means that they can deformed not only keeping but also changing their topology. Tying a knot in an unlooped rope corresponds to changing the topology of the rope. Therefore, we have to consider state transitions with topological change. Moreover, we focused not only on state transitions but also on actions that realize such state transitions. All possible knotting/unknotting plans can be derived on a computer once the initial and the objective states of a linear object are given, and are independent of human physical structure and experience. Thus, we can select a suitable plan for a manipulating system even if the system does not have the same mechanisms as the human body. In this paper, we also discuss planning for tightening knots. Some knots can be untied by simply pulling both endpoints. Tightening operations are also indispensable to complete knots. Consequently, our proposed method will be useful for actual planning of knotting/unknotting manipulation of linear objects.

2. Representation of Knotting/Unknotting Process

2.1. Crossing States

This section describes a topological representation of knotting/unknotting processes of a linear object. Let us investigate the process of tying an overhand knot in the linear object illustrated in Figure 1. The linear object in the initial state shown in Figure 1(a) should be tied in the overhand knot shown in Figure 1(e-1). The object has two endpoints, the left E_l and the right E_r. Successful tying of an overhand knot requires two consecutive operations. First, the object must be changed from the unlooped state into the looped state illustrated in Figure 1(b). Next, an endpoint must be passed through the loop as shown in Figure 1(c-1), and pulled to tie the object as illustrated in Figure 1(e-1). In each state, the object is crossed in a certain fashion, for example, the looped state in Figure 1(b) has one crossing, while the overhand knot in Figure 1(e-1) has three. In addition, the number of crossings increases during a successful tying process. Tying may fail due to inappropriate operations, for example, passing one endpoint of the object through the loop in Figure 1(c-2) yields the failed state in Figure 1(e-2). Note that the number of crossings may decrease during an unsuccessful tying process, for example, the intermediate state, in Figure 1(d-2), has two crossings, while the unsuccessful final state, in Figure 1(e-2), has only one. Two intermediate states, Figure 1(d-1) and Figure 1(d-2), have two crossings and the left part of the object is below the right part at crossing C_1. However, the right part of the object is below the left part at C_2 in Figure 1(d-1), while the right part is above the left part at C_2 in Figure 1(d-2). This suggests that it is necessary to specify which part is above/below at each crossing to describe the states during the knotting/unknotting process.

To describe the crossing states of a deformable linear object, it is first necessary to project the three-dimensional shape of the linear object onto a plane. The projected two-dimensional curve may cross itself. Crossings in the projected curve specify the crossing state.

Next, we trace the projected curve from the left endpoint E_l to the right endpoint E_r, which is defined as *counting direction*, and number the crossings passed first. The sequence of the crossings in the counting direction can then be described. Figure 2 shows an example of a linear object tied in a slipknot. The object has 5 crossings C_1 through C_5 and their sequence is denoted as:

$$E_l - C_1 - C_2 - C_3 - C_4 - C_5 - C_1 - C_2 - C_5 - C_4 - C_3 - E_r. \quad (1)$$

In addition, whether each crossing is involved in the upper or the lower part is specified. The symbols C_i^u and C_i^l denote the location of the i-th crossing point in the upper and the lower part, respectively. Furthermore, the crossings can be categorized as left-handed helical or right-handed helical, as in Figure 3(a) and Figure 3(b), respectively. In a left-handed helical crossing, the upper part overlaps first on the right side of the lower part and then overlaps on its left side. 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.

The sequence of symbols at individual crossing points determines the crossing states of a linear object. The crossing

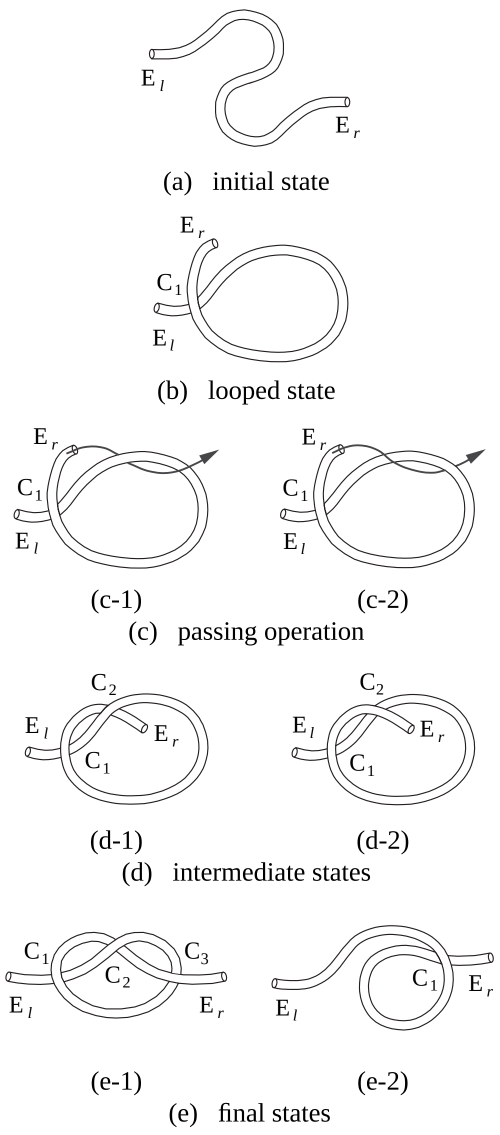


Fig. 1. Tying of overhand knot.

state of a knotted object shown in Figure 2(a) is described as:

$$E_l - C_1^{u-} - C_2^{l-} - C_3^{l+} - C_4^{u+} - C_5^{u-} - C_1^{l-} - C_2^{u-} - C_5^{l-} - C_4^{l+} - C_3^{u+} - E_r. \quad (2)$$

Let us describe a segment between two crossings C_i and C_j as ${}_i^pL_j^q$, where p and q indicate whether the segment at the crossings is the upper $(p, q = u)$ or lower part $(p, q = l)$. The knotted object shown in Figure 2(b) has 11 segments. For example, the second segment between crossing points C_1^{u-} and C_2^{l-} is denoted as ${}_1^uL_2^l$. Terminal segments adjoining the left and right endpoints are described as L_i^p and ${}_j^qL$, respectively.

Consequently, we can represent the crossing states of a knotted linear object by a sequence of crossing point symbols. This representation is topological; no geometric properties, such as length and thickness, or physical properties, such as weight and rigidity, are included.

2.2. Basic Operations for Crossing State Transitions

In this section, we introduce the basic operations that perform the transitions between crossing states of a knotted object. To change the crossing state of a linear object, an operation must be performed on the object. Therefore, a state transition corresponds to an operation that changes the number of crossings or permutes their sequence. In this paper, we introduce four basic operations described in Figure 4. Operations I, II, and III are equivalent to Reidemeister moves I, II and III in knot

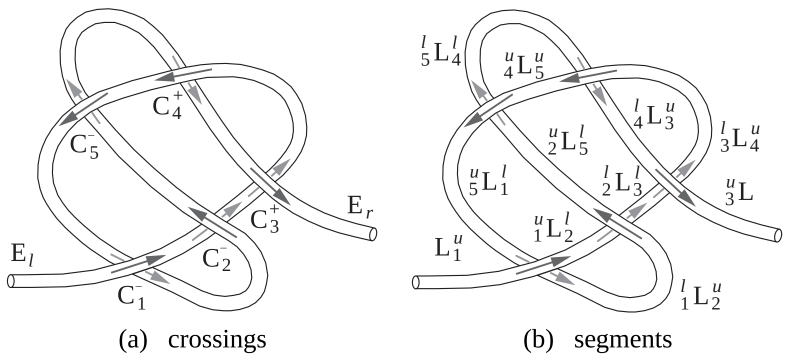


Fig. 2. Example of knotted linear object.

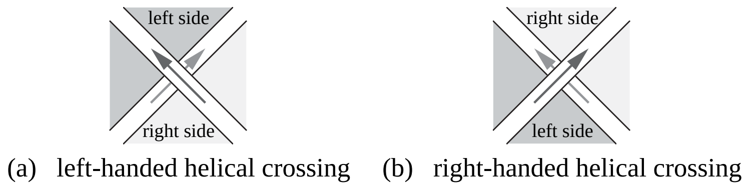

left side
right side
(a) left-handed helical crossing
right side
left side
(b) right-handed helical crossing

Fig. 3. Crossing type.

theory, respectively (Adams 1994). It has been demonstrated that any looped knot can be changed to topologically equivalent knots with these three moves alone (Reidemeister 1983). Operations I, II, and III are applied to intermediate parts of a linear object, while operation IV manipulates the endpoint of the object. Operation IV is needed because a linear object has two endpoints, while knot theory does not focus on any unlooped knot with endpoints. Operation IV corresponds to a topological change of a knot in the knot theory: cutting a looped knot, generating/deleting a crossing, and reconnecting one endpoint with the other. Consequently, any state transition corresponding to both a topologically equivalent change and a topological change can be realized by operations I through IV. Operations I, II, and IV increase or decrease the number of crossings. Let us divide operation I into crossing operation CO_I which increases the number of crossings and uncrossing operation UO_I which decreases the number. Crossing operations CO_{II} and CO_{IV} and uncrossing operations UO_{II} and UO_{IV} are also defined. Operation III does not change the number of crossings but permutes their sequence. Operation III is referred to as an arranging operation AO_{III}.

Each basic operation can be applied to specific subsequences of crossings. Let us investigate subsequences to which each operation is applicable. Operation UO_I is applicable to subsequences represented as follows:

$$\cdots - C_i^u - C_i^l - \cdots , \tag{3}$$
$$\cdots - C_i^l - C_i^u - \cdots . \tag{4}$$

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

$$\cdots - C_i^u - C_j^u - \cdots - C_i^l - C_j^l - \cdots , \tag{5}$$
$$\cdots - C_i^u - C_j^u - \cdots - C_j^l - C_i^l - \cdots , \tag{6}$$
$$\cdots - C_i^l - C_j^l - \cdots - C_i^u - C_j^u - \cdots , \tag{7}$$
$$\cdots - C_i^l - C_j^l - \cdots - C_j^u - C_i^u - \cdots . \tag{8}$$

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 UO_{IV} is applicable to subsequences represented as follows:

$$E_l - C_i^u - \cdots - C_i^l - \cdots , \tag{9}$$
$$E_l - C_i^l - \cdots - C_i^u - \cdots , \tag{10}$$
$$\cdots - C_i^l - \cdots - C_i^u - E_r , \tag{11}$$
$$\cdots - C_i^u - \cdots - C_i^l - E_r . \tag{12}$$

That is, a crossing adjacent to the endpoint can be deleted by operation UO_{IV}. Operation AO_{III} is applicable to subsequences represented as permutations of the following three subsequences: α_i, β_i, and γ_i, e.g., $\cdots - \beta_1 - \gamma_2 - \alpha_1 - \cdots$:

$$\alpha_1 : \cdots - C_i^u - C_j^u - \cdots ,$$
$$\alpha_2 : \cdots - C_j^u - C_i^u - \cdots , \tag{13}$$

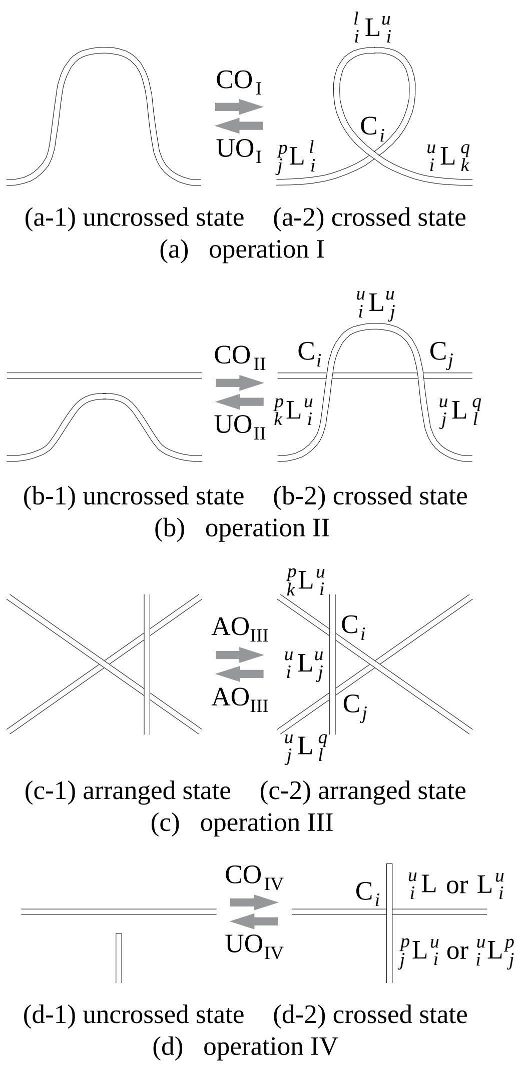


Fig. 4. Basic operations.

$$\begin{aligned}
\beta_1 &: \quad \cdots - \mathrm{C}_j^l - \mathrm{C}_k^u - \cdots , \\
\beta_2 &: \quad \cdots - \mathrm{C}_k^u - \mathrm{C}_j^l - \cdots , \qquad (14)\\
\gamma_1 &: \quad \cdots - \mathrm{C}_i^l - \mathrm{C}_k^l - \cdots , \\
\gamma_2 &: \quad \cdots - \mathrm{C}_k^l - \mathrm{C}_i^l - \cdots . \qquad (15)
\end{aligned}$$

That is, three crossings consisting of three segments one of which overlaps with 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. On the other hand, crossing operations $\mathrm{CO_I}$, $\mathrm{CO_{II}}$, and $\mathrm{CO_{IV}}$ can be applied to any sequence. Consequently, the number of possible crossing operations from one crossing state can be larger than that of possible uncrossing operations. For example, in the state shown in Figure 4(a-2), one $\mathrm{UO_I}$ operation deletes the crossing C_i, and three $\mathrm{CO_I}$ operations generate a new crossing, which is generated in segments $_j^p\mathrm{L}_i^l$, $_i^l\mathrm{L}_i^u$, or $_i^u\mathrm{L}_k^q$ where j and k are the previous and the subsequent crossing number, respectively. Thus, generation of possible transitions from a crossed to an uncrossed state is more effective than that from an uncrossed to a crossed state because larger numbers of possible states/operations must be considered in the latter. In this study, a state transition network was generated from a crossed to an uncrossed state. Note that all uncrossing operations applicable to each crossing state are included in the graph, while some possible crossing operations, which can be derived by inversing uncrossing operations, are not.

Each transition denotes a crossing/uncrossing operation. The knotting process can be represented as a sequence of transitions from an uncrossed to a crossed state, while the unknotting process can be described as the inverse—a sequence of transitions from a crossed to an uncrossed state. Once the initial crossing state and the objective crossing state are given, it is possible to generate possible sequences of transitions, i.e., possible knotting/unknotting processes.

In this paper, we assume that any linear object has two endpoints. Then, any crossing state description includes subsequences described in eqs. (9) through (12) whenever a linear object crosses itself. This implies that any unknotting process can be represented by iteration of operations $\mathrm{UO_{IV}}$ alone. It also means that any knotting process, i.e., the inverse of any unknotting process, can be represented by iteration of operations $\mathrm{CO_{IV}}$ alone. Recall that we often search for an endpoint and manipulate it to unravel a self-entwined rope. This leads to the following theorem:

THEOREM 1. Any knotting/unknotting manipulation of a linear object can be realized by iteration of operations $\mathrm{CO_{IV}}/\mathrm{UO_{IV}}$ alone.

3. Planning of Knotting/Unknotting Manipulation

3.1. Actions for Uncrossing Operations

In the previous section, the knotting/unknotting process of a linear object was represented as a sequence of crossing state transitions. Moreover, we found that possible knotting/unknotting processes can be generated once the initial and the objective states are given. To accomplish one of the possible processes, it is necessary to grasp, move, and release the object during knotting/unknotting. Whether the crossing state of the object changes as expected is dependent on the points grasped and their directions of movement, including the direction of approach of a manipulator to each grasping point. We investigated a qualitative plan for manipulation as a sequence of crossing state transitions, including grasping points, their directions of movement, and their directions of approach, which is referred to as a *qualitative manipulation plan*, to realize a possible process. In this section, we explain the procedure used to determine adequate grasping points, their directions of movement, and their directions of approach for one state transition.

Let us define crossings that are generated/deleted by a crossing/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. A crossed state shown in Figure 4(a-2) consists of one crossing, C_i^+, and three segments, ${}^p_jL^l_i$, ${}^l_iL^u_i$, and ${}^u_iL^q_k$. Operation $\mathrm{UO_I}$ deletes crossing C_i^+, which corresponds to a target crossing. Segment ${}^l_iL^u_i$ corresponds to a target segment because it exists between target points, C_i^{l+} and C_i^{u+}. 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 $\mathrm{UO_I}$ and $\mathrm{UO_{II}}$ can be realized by grasping the target segment or both its adjacent segments and that operation $\mathrm{UO_{IV}}$ can be realized by grasping the target segment or its adjacent segment. In the above case, segment ${}^l_iL^u_i$ or segments ${}^p_jL^l_i$ and ${}^u_iL^q_k$ should be grasped.

Next, let us consider the direction of movement of a grasping point required to realize each operation. Assume that a pair of fingertips grasps a linear object during its knotting/unknotting. Once the fingertip pair grasps the object firmly, it can be regarded as a rigid body. Generally, a rigid body in 3D space has 3 DOF in translation and 3 DOF in rotation. We apply 3 DOF in translation and 3 DOF in rotation to the knotting/unknotting of a linear object. Let us define the central axis of a linear object as axis 1, the axis perpendicular to the central axis and along the projection plane 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 direction of approach of a 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. For example, in Figure 4(d-2), we assume that a manipulator grasps a target segment, ${}^u_iL$, and translate it along axis-1 to perform operation $\mathrm{UO_{IV}}$. The manipulator cannot perform operation $\mathrm{UO_{IV}}$ when it grasps the target segment from the rear, while it can perform the operation when it grasps from the front. 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. Figure 5 shows an example of set of grasping points, their directions of movement, and their directions of approach to realize operation $\mathrm{UO_I}$. This set is referred to as an *action* in this paper. In Figure 5(a), both segments adjoining a target segment are grasped and pulled away from each other. Then, the object state can be changed into the state shown in Figure 5(b). Figure 6 illustrates possible actions for each uncrossing operation. In this figure, actions before performing uncrossing operations as shown in Figure 5(a) are enumerated. 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. We derived 34 actions to realize uncrossing operations as listed in Figure 6.

3.2. Actions for Crossing Operations

We assume that any crossing operation can be realized by grasping target points, target segments, or their adjacent

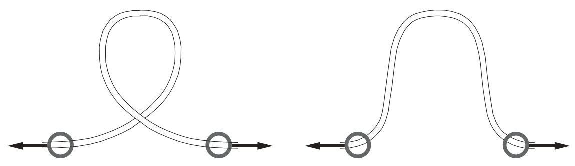


Fig. 5. Example of action for operation UO_I.

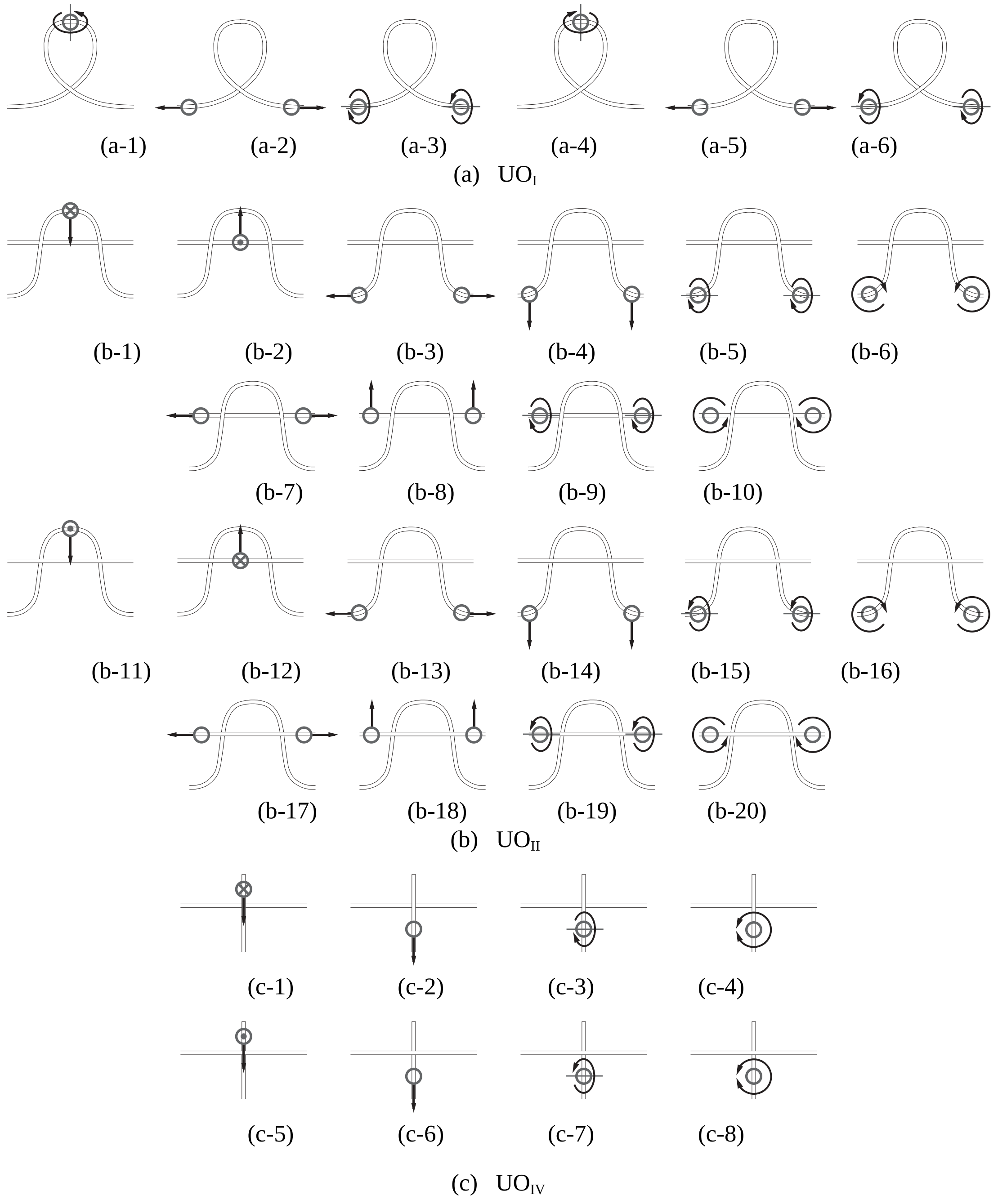


Fig. 6. Actions for uncrossing operations.

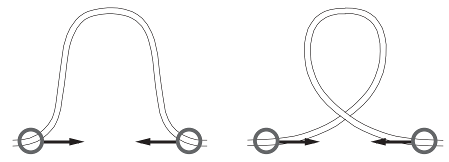


Fig. 7. Example of action for operation $\mathrm{CO_I}$.

segments. Figure 7 shows an action to realize operation $\mathrm{CO_I}$. In Figure 7(a), both ends of a segment are grasped and moved closer to each other. Then, the object is kinked and a target crossing can be generated as shown in Figure 7(b). Thus, we can derive 46 actions for crossing operations as enumerated in Figure 8. These actions correspond to those after performing crossing operations as shown in Figure 7(b).

3.3. Actions for Arranging Operation

An arranging operation does not generate/delete any crossing. Let us define a segment with two upper adjacent crossing points as an upper target segment, a segment with two lower adjacent crossing points as a lower target segment, and a segment with one upper and one lower adjacent crossing points as a middle target segment for operation $\mathrm{AO_{III}}$, respectively. Then, we can realize operation $\mathrm{AO_{III}}$ by moving one of these three target segments. Figure 9 shows all 16 actions for operation $\mathrm{AO_{III}}$. In Figure 9(a) through (f), an upper target segment is moved, while a lower target segment is moved in Figure 9(g) through (l). Moreover, in Figure 9(m) through (p), a middle target segment is moved to realize operation $\mathrm{AO_{III}}$.

Actions, i.e., adequate sets of grasping points, their directions of movement, and their directions of approach, 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.

3.4. One-Handed Knotting

Possible actions for each operation illustrated in Figure 6, Figure 8, and Figure 9 include those with only a single grasping point. Moreover, crossing operations can be realized by planar motions, i.e., T_1, T_2, and R_3, of a single grasping point. Relevant actions are: Figure 8(a-1), (a-2), (a-6), and (a-7) for operation $\mathrm{CO_I}$; Figure 8(b-3), (b-4), (b-15), and (b-16) for operation $\mathrm{CO_{II}}$; and Figure 8(c) except (c-5) and (c-11) for operation $\mathrm{CO_{IV}}$. Operation $\mathrm{AO_{III}}$ can also be performed by actions with planar motions of a single grasping point as shown in Figure 9(a) and (g). Now, let us assume that a linear object is placed on a table corresponding to the projection plane. Manipulators can approach the object only from the front of the projection plane. In this case, operations $\mathrm{CO_I}$, $\mathrm{CO_{II}}$, and $\mathrm{CO_{IV}}$ can be accomplished by a single manipulator approaching from the front. These actions are shown in Figure 8(a-1), (a-6), (b-3), (b-16), (c-1), (c-3), (c-4), (c-6), (c-7), (c-10), and (c-12). Operation $\mathrm{AO_{III}}$ can also be performed by a manipulator approaching from the front. Thus, any crossing operation and any arranging operation can be performed by grasping a single point/segment from the front of the projection plane and by imposing planar motions, T_1, T_2, and R_3. Translation along the projection normal, i.e., T_3, is needed to overlap the grasping point/segment with another point/segment. Consequently, the following theorem is derived.

THEOREM 2. Any knotting manipulation of a linear object placed on a table can be accomplished by a one-handed robot with three translational DOF and one rotational DOF.

Note that a SCARA robot is sufficient to impose motions T_1, T_2, T_3, and R_3 on an object. Thus, one SCARA robot can realize any knotting manipulation, and it is not necessary to use dual 6-DOF manipulators or anthropomorphic arms.

4. Example of Manipulation Planning

4.1. Planning of Unknotting Manipulation

In this section, we describe an example of possible unknotting processes generated by a computer system. Figure 10 shows a required manipulation, which corresponds to untying a slipknot. The initial state in Figure 10(a) is represented as $\mathrm{E}_l-\mathrm{C}_1^{u-}-\mathrm{C}_2^{l-}-\mathrm{C}_3^{l+}-\mathrm{C}_4^{u+}-\mathrm{C}_5^{u-}-\mathrm{C}_1^{l-}-\mathrm{C}_2^{u-}-\mathrm{C}_5^{l-}-\mathrm{C}_4^{l+}-\mathrm{C}_3^{u+}-\mathrm{E}_r$ and the objective state in Figure 10(b) is represented as $\mathrm{E}_l-\mathrm{E}_r$. The initial state includes a subsequence to which operation $\mathrm{UO_{II}}$ is applicable: $\cdots-\mathrm{C}_4^{u+}-\mathrm{C}_5^{u-}-\cdots-\mathrm{C}_5^{l-}-\mathrm{C}_4^{l+}-\cdots$. It also includes subsequences to which operation $\mathrm{UO_{IV}}$ is applicable: $\mathrm{E}_l-\mathrm{C}_1^{u-}-\cdots-\mathrm{C}_1^{l-}-\cdots$ and $\cdots-\mathrm{C}_3^{l+}-\cdots-\mathrm{C}_3^{u+}-\mathrm{E}_r$. Application of operation $\mathrm{UO_{II}}$ will cause deletion of crossings C_4^{+} and C_5^{-}. Then, the crossing state of the object is changed to $\mathrm{E}_l-\mathrm{C}_1^{u-}-\mathrm{C}_2^{l-}-\mathrm{C}_3^{l+}-\mathrm{C}_1^{l-}-\mathrm{C}_2^{u-}-\mathrm{C}_3^{u+}-\mathrm{E}_r$. Application of operation $\mathrm{UO_{IV}}$ results in deletion of crossing C_1^{-} or C_3^{+}. In the former case, the crossing state is changed to $\mathrm{E}_l-\mathrm{C}_1^{l-}-\mathrm{C}_2^{l+}-\mathrm{C}_3^{u+}-\mathrm{C}_4^{u-}-\mathrm{C}_1^{u-}-\mathrm{C}_4^{l-}-\mathrm{C}_3^{l+}-\mathrm{C}_2^{u+}-\mathrm{E}_r$ after renumbering crossings. In the latter case, the following crossing state is derived: $\mathrm{E}_l-\mathrm{C}_1^{u-}-\mathrm{C}_2^{l-}-\mathrm{C}_3^{u+}-\mathrm{C}_4^{u-}-\mathrm{C}_1^{l-}-\mathrm{C}_2^{u-}-\mathrm{C}_4^{l-}$

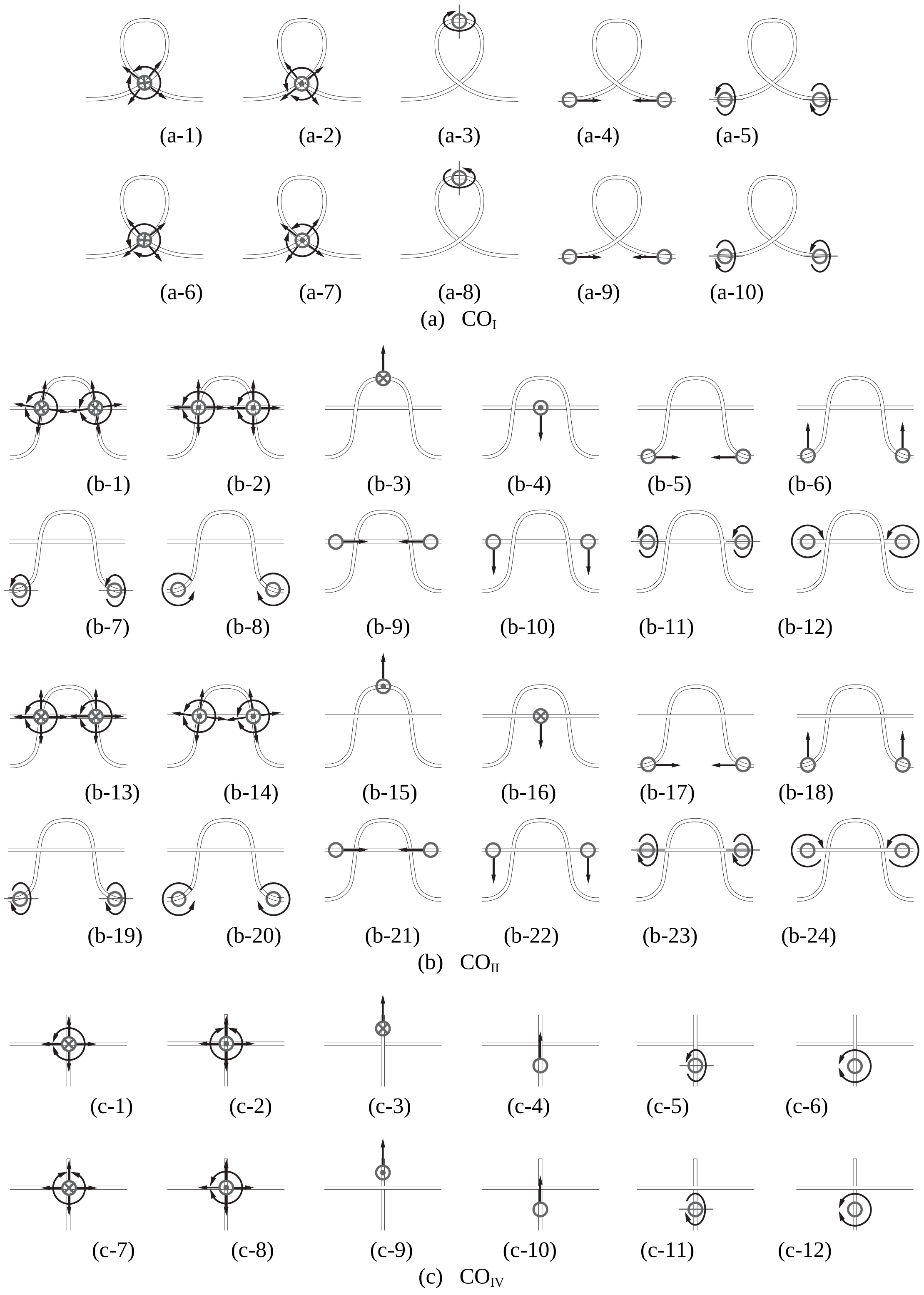


Fig. 8. Actions for crossing operations.

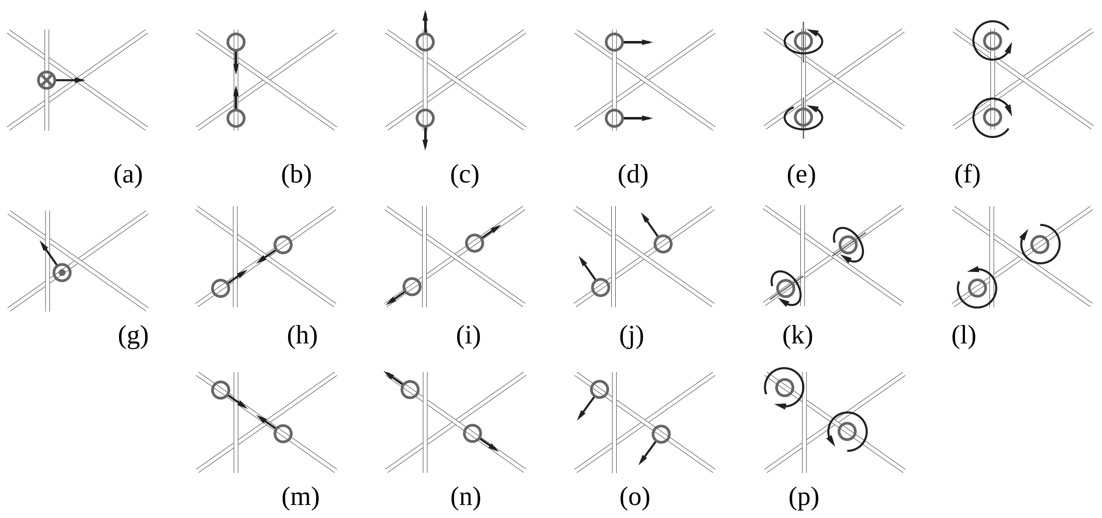


Fig. 9. Actions for arranging operation.

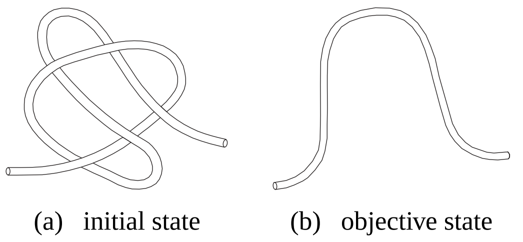


Fig. 10. Example of required manipulation—untying slipknot.

$-C_3^{l+}-E_r$. Thus, we can generate possible state transitions from each crossing state automatically. All the following computations were performed on an 833 MHz Alpha 21264 CPU with 1 GB memory operated by Tru64UNIX. Programs were compiled with Compaq C Compiler V6.4 with optimization option O4. Assuming that only uncrossing operations can be used, i.e., without operation AO_{III}, 14 crossing states and 39 state transitions are derived as shown in Figure 11. Computation was completed within 10 ms. Note that the state transition graph shown in Figure 11 includes sequences consisting of UO_{IV} operations only, confirming the validity of Theorem 1. Including operation AO_{III}, we can derive 21 crossing states and 68 state transitions. This computation takes about 10 ms. Figure 12 shows additional crossing states and uncrossing operations when operation AO_{III} is included. In this figure, states with the number less than 15 are equivalent to those with the same number in Figure 11. As shown in these figures, possible knotting/unknotting processes of a linear object can be generated automatically once the initial and the objective states are given.

In general, a crossing state graph becomes huge and complex when an object has many crossings and when operations AO_{III} are included. For example, let us investigate the bowknot shown in Figure 13(a), which has 11 crossings. When operation AO_{III} is not considered, a graph with 53 states and 153 operations is derived to untie this knot, requiring a computation time of about 15 ms. With the inclusion of operations AO_{III}, the generated graph has 932 states and 4282 operations, and it takes about 0.5 s to generate this graph. Figure 13(b) shows a traditional Japanese decorative knots, the *ume*-knot, the shape of which mimics that of “ume” or Japanese plum blossom. This knot is used for binding the mouth of tool bags for incense burning. The ume-knot has 54 crossings. A graph with 77796 states and 398841 operations was generated for its untying when operation AO_{III} was not considered, and computation time was about 22200 s (6 hours and 10 minutes). It was not possible to compute a graph including operations AO_{III} due to memory limitations. However, even if sufficient memory were available, a very long time would probably be needed to compute the graph including operations AO_{III}. Thus,

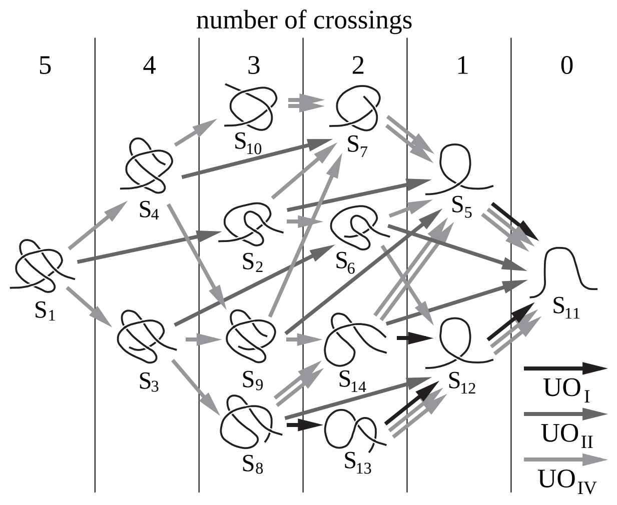


Fig. 11. Results of unknotting process planning.

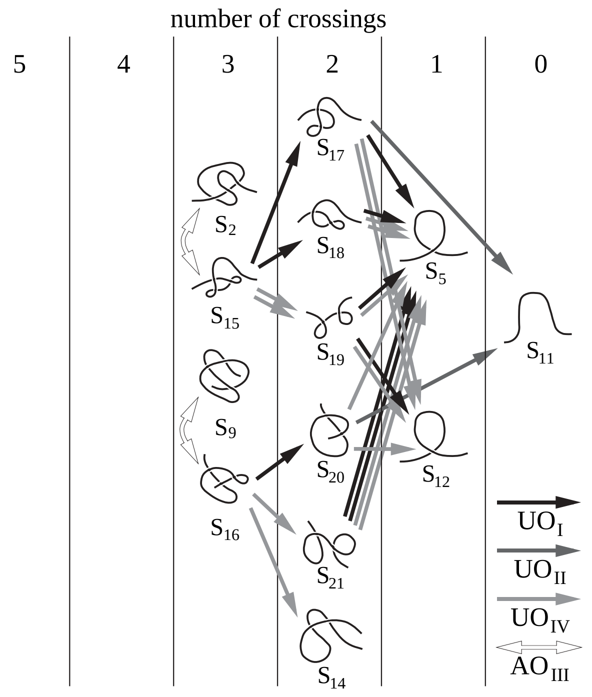


Fig. 12. Additional states and operations when AO_{III} is included.

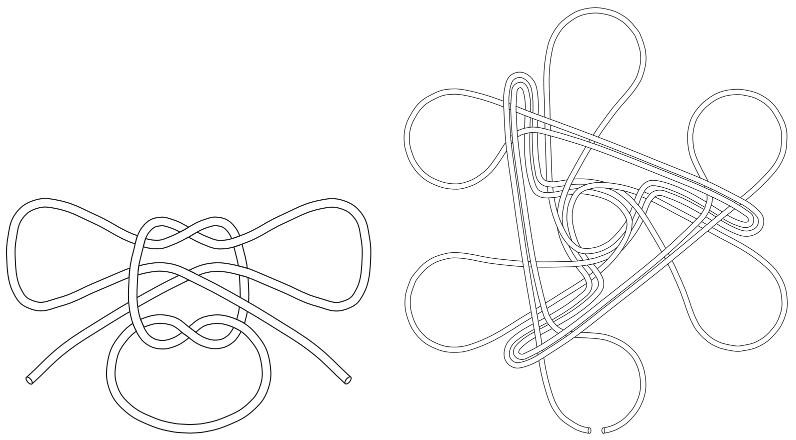

(a) Bowknot (b) *Ume*-knot

Fig. 13. Complex knots.

it is preferable not to include operations $\mathrm{AO_{III}}$ to allow the complete and efficient generation of a crossing state graph for tying/untying of complex knots.

Here, we introduce evaluation criteria to allow the selection of appropriate qualitative manipulation plans from among those generated. First, let N_t be the number of state transitions in one sequence. In this paper, we discuss a sequence including fewer intermediate states as it takes more time to complete the required manipulation when the selected sequence includes more state transitions. Note that a knotting/unknotting process corresponds to increase/decrease of crossings of a linear object. Recall that operation II generates/deletes two crossings, while operations I and IV generate/delete one crossing. Then, we find that sequence that includes more iterations of operation II has fewer intermediate states. This implies that we should select a sequence involving more operations II and fewer operations I and IV. Next, let N_r be the number of regraspings in one sequence. When a grasping point remains unchanged during manipulation, the position and direction of the linear object at the grasping point coincide with the position and direction of the fingertip of the manipulator. Thus, estimation of the object shape is not necessary once the manipulator grasps the object. However, if the grasping point changes during manipulation process, the position and direction of the segment to be grasped in the next operation must be estimated in detailed planning. Furthermore, it takes more time to change the grasping point. Therefore, a sequence with fewer regraspings is preferable.

In this example, the number of crossings in the initial state is five, and that in the objective state is decreased to zero. We can reduce the number of crossings from five to zero by applying two $\mathrm{UO_{II}}$ operations and one $\mathrm{UO_{I}}$ or $\mathrm{UO_{IV}}$ operation. Their possible sequences are as follows:

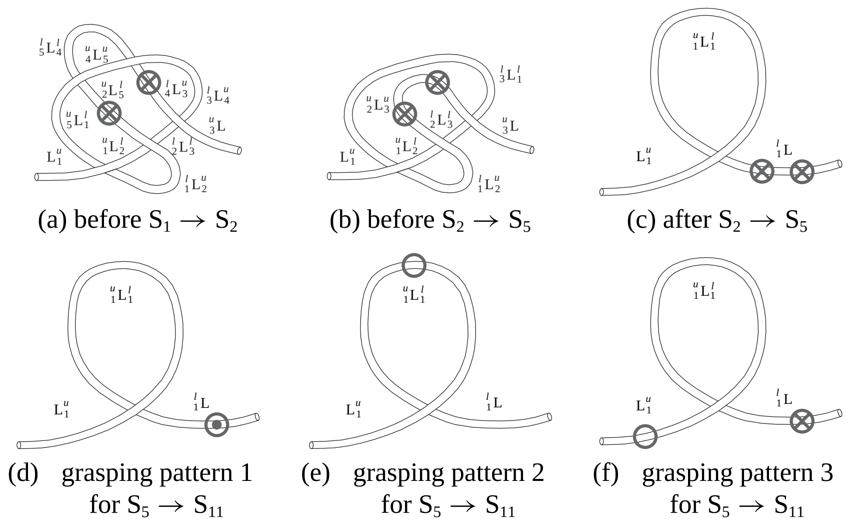
$$\begin{aligned}\mathrm{OQ_1} &: \mathrm{UO_{II}} \to \mathrm{UO_{II}} \to \mathrm{UO_{I}}/\mathrm{UO_{IV}},\\ \mathrm{OQ_2} &: \mathrm{UO_{II}} \to \mathrm{UO_{I}}/\mathrm{UO_{IV}} \to \mathrm{UO_{II}},\\ \mathrm{OQ_3} &: \mathrm{UO_{I}}/\mathrm{UO_{IV}} \to \mathrm{UO_{II}} \to \mathrm{UO_{II}}.\end{aligned}$$

Then, we check whether the above possible sequences can realize the required process. The three sequences, $\mathrm{OQ_1}$ through $\mathrm{OQ_3}$, correspond to the following state transition sequences, respectively:

$$\begin{aligned}\mathrm{SQ_1} &: \mathrm{S_1} \to \mathrm{S_2} \to \mathrm{S_5} \to \mathrm{S_{11}},\\ \mathrm{SQ_2} &: \mathrm{S_1} \to \mathrm{S_2} \to \mathrm{S_6} \to \mathrm{S_{11}},\\ \mathrm{SQ_3} &: \mathrm{S_1} \to \mathrm{S_3} \to \mathrm{S_6} \to \mathrm{S_{11}}.\end{aligned}$$

If the required process cannot be realized with two $\mathrm{UO_{II}}$ operations and one $\mathrm{UO_{I}}$ or $\mathrm{UO_{IV}}$ operation, we check whether one $\mathrm{UO_{II}}$ operation and three $\mathrm{UO_{I}}$ and/or $\mathrm{UO_{IV}}$ operations can realize the required process. Thus, we check repeatedly whether a knot with n crossings can be unknotted by applying ${}_{\alpha+\beta}\mathrm{C}_{\beta}$ combinations of α $\mathrm{UO_{II}}$ operations and β $\mathrm{UO_{I}}$ and/or $\mathrm{UO_{IV}}$ operations with decreasing α and increasing β so that they satisfy $2\alpha + \beta = n$ until a sequence of operations to unknot is found. Then, we can efficiently derive manipulation processes including fewer state transitions, i.e., processes with lower N_t without generating the whole graph including all possible processes. For example, from this algorithm, it is found that the ume-knot illustrated in Figure 11(b) can be unknotted by applying 27 $\mathrm{UO_{II}}$ operations. The graph has 1512 states and 5113 operations, and it takes about 15 s to generate it with the same specification of computation.

Next, we select adequate actions so that the manipulation process has fewer regraspings, N_r. Let us consider the sequence $\mathrm{SQ_1}$. For the first transition from state $\mathrm{S_1}$ to state $\mathrm{S_2}$, assume that segments ${}^{u}_{2}\mathrm{L}^{l}_{5}$ and ${}^{l}_{4}\mathrm{L}^{u}_{3}$ are grasped from the front



(a) before $S_1 \rightarrow S_2$ (b) before $S_2 \rightarrow S_5$ (c) after $S_2 \rightarrow S_5$

(d) grasping pattern 1 for $S_5 \rightarrow S_{11}$ (e) grasping pattern 2 for $S_5 \rightarrow S_{11}$ (f) grasping pattern 3 for $S_5 \rightarrow S_{11}$

Fig. 14. Grasping points and their changing times.

as shown in Figure 14(a) to perform operation UO_{II}. Then, the grasped segments become equivalent to segment ${}^u_2L^u_3$ in state S_2 as shown in Figure 14(b). State S_2 can be changed to state S_5 by moving segment ${}^u_2L^u_3$. Then, segment ${}^l_1L$ in state S_5 is grasped from the front by two manipulators as shown in Figure 14(c). There are three choices to change the state to the final state, S_{11}. The first is to regrasp segment ${}^l_1L$ from the rear for operation UO_{IV} as shown in Figure 14(d). The second is to release segment ${}^l_1L$ and to grasp segment ${}^u_1L^l_1$ as shown in Figure 14(e) for operation UO_I or UO_{IV}. The third is to grasp segment L^u_1 while still grasping segment ${}^l_1L$ for operation UO_I as shown in Figure 14(f). It is necessary to regrasp for the last transition from state S_5 to state S_{11} in all choices. Consequently, in the above plans to perform sequence, SQ_1, the minimum number of $N_r = 1$. We can also derive the minimum N_r for sequence SQ_2 and sequence SQ_3; $N_r = 2$ in SQ_2, while $N_r = 1$ in SQ_3. This implies that sequence SQ_2 should be eliminated from the manipulation plans. Thus, we can select appropriate candidate manipulation plans by evaluating N_t and N_r. Then, quantitative analysis should be performed to check whether a selected manipulation can be realized practically by taking the physical properties of the linear object, such as rigidity, into consideration.

4.2. Planning of Knotting Manipulation

Next, we present an example of knotting process generation. Figure 15 shows a required manipulation corresponding to tying a slipknot, i.e., the inverse of the previous manipulation example. The system generates state transition sequences from the objective state to the initial state corresponding to the state transition graph shown in Figure 11. The state transition graph shown in Figure 16 is derived by inversing the uncrossing operations to corresponding crossing operations.

5. Planning for Tying Tightly

In general, knots fulfill their fixing function after they are tied tightly by pulling on various parts of the knot. Which part of a knot should be pulled for tightening is dependent on the topological state of the knot.

The knot shown in Figure 17(a) corresponds to an overhand knot and the knot shown in Figure 17(b) corresponds to a slipknot. The overhand knot can be tied tightly by grasping both terminal segments of the knot, L^u_1 and ${}^l_3L$, and pulling them in opposite directions. In contrast, the knot shown in Figure 17(c) is released if both terminal segments are pulled. Now, let us grasp an intermediate segment in addition to the terminal segments when pulling. For example, by grasping segment ${}^l_5L^l_4$ in addition to terminal segments L^u_1 and ${}^u_3L$ and pulling these segments in opposite directions, the slipknot shown in Figure 17(b) is then tied tightly. Let us cut the object at a point in segment ${}^l_5L^l_4$ as shown in Figure 18(a) and divide the object into two parts as shown in Figure 18(b) and Figure 18(c). Let E'_r be the new right endpoint of the left part shown in Figure 18(b) and let E'_l be the new left endpoint of the right part shown in Figure 18(c). Then, the left part corresponds to an overhand knot as shown in Figure 17(a). That is, the left part can be tightened when terminal segment, L^u_1, and segment ${}^l_5L^l_4$ are pulled. Then, terminal segment, ${}^u_3L$ must be fixed against the pull of segment ${}^l_5L^l_4$ to maintain the crossing state. Consequently, we can tie the slipknot by pulling three segments, L^u_1, ${}^l_5L^l_4$, and ${}^u_3L$. The knot shown in Figure 17(c) does not include such tightenable parts even if all segments are cut and the object is divided into many parts. This suggests that the knot shown in Figure 17(c) can never be tied tightly. Thus, knots can be categorized as tightenable or untightenable. The knots shown in Figure 17(a) and Figure 17(b) are tightenable, while that shown in Figure 17(c) is untighten-

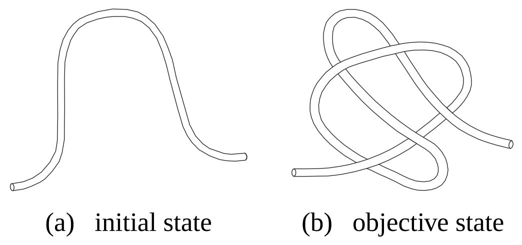


Fig. 15. Example of required manipulation—tying slipknot.

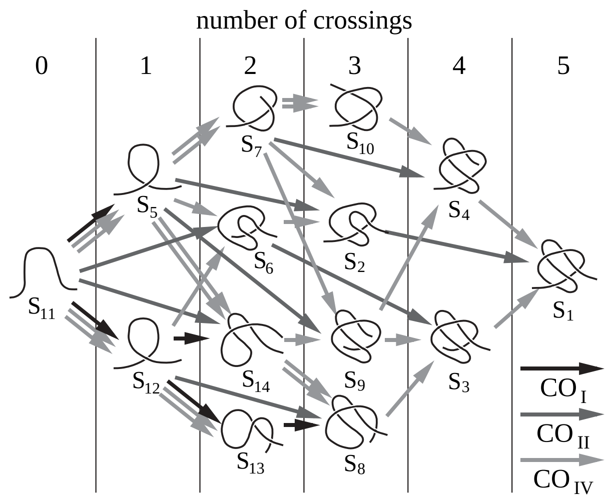


Fig. 16. Results of knotting process planning.

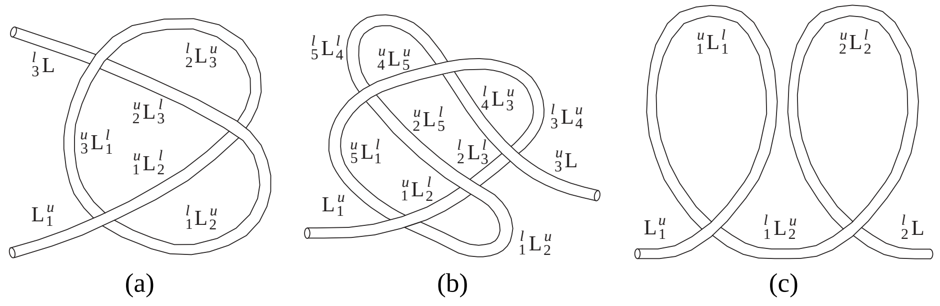


Fig. 17. Completely tightenable, partially tightenable, and untightenable knots.

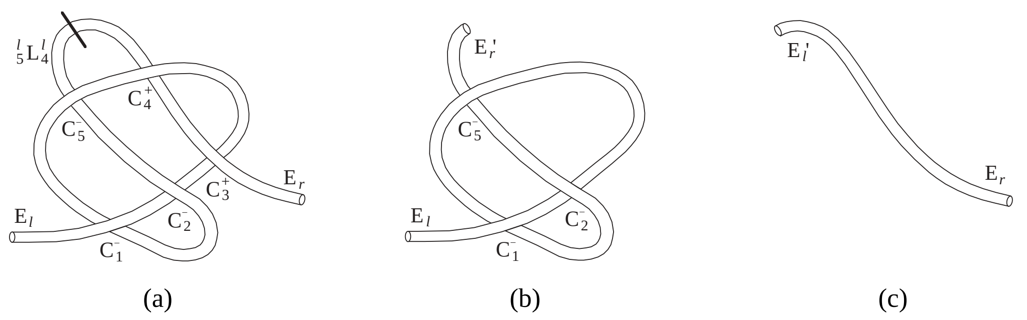


Fig. 18. Dividing of slipknot.

able. Furthermore, some tightenable knots can be tied tightly by pulling their terminal segments alone, while others cannot. This section describes how a knot can be tied tightly by pulling its segments in opposite directions.

First, we define that *tying tightly* or *tightening* is to infinitely shorten the length of the ungrasped segments without changing the crossing state of the object by pulling the grasped segments. Then, the knots shown in Figure 17(a) and Figure 17(b) can be defined as *tightenable* knots, and that shown in Figure 17(c) can be defined as an *untightenable* knot. Moreover, we define a knot that can be tightened by pulling its terminal segments alone as a *completely tightenable* knot. Recall that intermediate segments in addition to terminal segments must be pulled away to tighten the knot shown in Figure 17(b). This knot is referred to as a *partially tightenable* knot. For example, a figure-of-eight knot is a completely tightenable knot, while a bowknot is a partially tightenable knot.

Let us define a set of closed regions surrounded by a linear object as the *inner region*, and the other region in the projection plane is defined as the *outer region*. Some segments of a knot touch the outer region, while other segments do not. The former are referred to as *outer segments*, while the latter are referred to as *inner segments*. Figure 19 shows an example of inner and outer regions/segments. Note that we have to pull both terminal segments of a knot as far away as possible to achieve tightening. If the terminal segments are inner segments, it may not be possible to pull them away sufficiently without changing the crossing state. Thus, in this section, we assume that both terminal segments are outer segments. Cut parts of the knot as shown in Figure 18(b) and Figure 18(c) must also have outer terminal segments for tightenability. Therefore, segments to be pulled for tightening a linear object can be determined by repeating a procedure consisting of cutting some outer segments, dividing the object into various parts, and checking tightenability of each part, after identifying the outer segments.

First, we will explain the procedure used to detect outer segments from the crossing state description. All outer segments are included within the boundary of the outer region and are connected to one another continuously. Therefore, we can detect outer segments by tracing a segment connected to the previous outer segment and making contact with the outer region from the left terminal segment. Let us define *tracing direction* such that any segment touches the outer region on its right side. At the start of tracing from the left endpoint, the tracing direction is the same as the counting direction. We trace the object, and when we encounter an upper/lower crossing point, we switch to the other crossing point so that the following segment touches the outer region on its right side. The tracing direction after switching is dependent on the crossing. If a crossing is a right-handed helical and after switching from the upper to the lower point, we should trace segments in the opposite of the counting direction. Table 1 shows the rule for switching at each crossing. If we arrive at the right endpoint of the object, we turn back and continue this examination. The procedure finishes when we return to the left endpoint. Then, all segments traced in this procedure correspond to outer segments. For example, the knot shown in Figure 17(b) is described as follows:

$$\begin{aligned} &E_l - C_1^{u-} - C_2^{l-} - C_3^{l+} - C_4^{u+} - C_5^{u-} - C_1^{l-} - C_2^{u-} \\ &\quad - C_5^{l-} - C_4^{l+} - C_3^{u+} - E_r . \qquad (16) \end{aligned}$$

Then, we can trace segments L_1^u, ${}_1^{l}L_2^u$, ${}_2^{l}L_3^l$, ${}_3^{u}L$, ${}_3^{l}L_4^u$, ${}_5^{l}L_4^l$, and ${}_5^{u}L_1^l$ as outer segments, and it is possible to confirm that these are outer segments from Figure 19. Thus, we can detect outer segments of a linear object from the crossing state description. To check for partial tightenability, we cut these outer segments alone.

Next, let us describe a procedure to check tightenability. Note that a completely tightenable knot can be tied by pulling both terminal segments alone. If it can be unknotted by the operation UO_I or UO_{II}, its crossing state can change even if both terminal segments are grasped. This implies that a completely tightenable knot does not include any subsequence given in eqs. (3) through (8). We can check whether a knot is completely tightenable from its crossing state description. If the knot is not completely tightenable, we then check whether it is a partially tightenable knot.

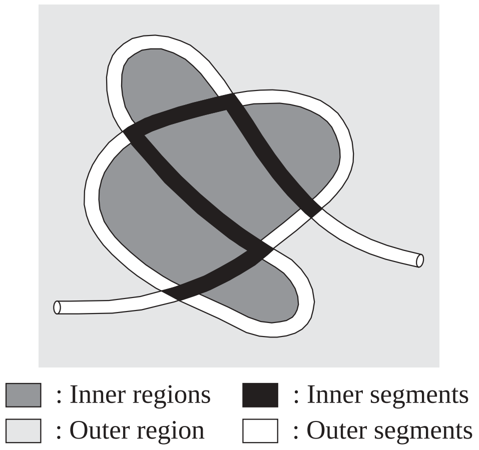


Fig. 19. Example of inner and outer region/segments.

Table 1. Rule for Switching at Crossing

Previous Direction	Crossing		Next Direction
Same	Right-handed helical	lower → upper	same
		upper → lower	opposite
	Left-handed helical	lower → upper	opposite
		upper → lower	same
Opposite	Right-handed helical	lower → upper	opposite
		upper → lower	same
	Left-handed helical	lower → upper	same
		upper → lower	opposite

We cut outer segments of the knot and divide it into various parts. Upper/lower crossing points in one part can be deleted if their corresponding crossing points are included in another part. Then, if at least one part has crossings with itself and if any part does not include any subsequence to which operation $\mathrm{UO_I}/\mathrm{UO_{II}}$ can be applied, the knot is partially tightenable and cut segments are pulling segments. The knot is untightenable if the above conditions are not satisfied even if all segments are cut. For example, the knot shown in Figure 17(b) includes a subsequence, $\cdots -\mathrm{C}_4^u-\mathrm{C}_5^u-\cdots-\mathrm{C}_5^l-\mathrm{C}_4^l-\cdots$, which corresponds to a subsequence given in eq. (6). Thus, the knot is not completely tightenable. Recall that outer segments of this knot are L_1^u, ${}^{l}_{1}\mathrm{L}_2^u$, ${}^{l}_{2}\mathrm{L}_3^l$, ${}^{u}_{3}\mathrm{L}$, ${}^{l}_{3}\mathrm{L}_4^u$, ${}^{l}_{5}\mathrm{L}_4^l$, and ${}^{u}_{5}\mathrm{L}_1^l$. The two parts, P_{11} and P_{12}, obtained after cutting segment ${}^{u}_{5}\mathrm{L}_1^l$ are described as follows:

$$\begin{aligned}\mathrm{P}_{11}\ :\ & \mathrm{E}_l - \mathrm{C}_1^{u-} - \mathrm{C}_2^{l-} - \mathrm{C}_3^{l+} - \mathrm{C}_4^{u+} - \mathrm{C}_5^{u-} - \mathrm{E}_r' \\ & \Rightarrow \mathrm{E}_l - \mathrm{E}_r', \end{aligned} \tag{17}$$

$$\begin{aligned}\mathrm{P}_{12}\ :\ & \mathrm{E}_l' - \mathrm{C}_1^{l-} - \mathrm{C}_2^{u-} - \mathrm{C}_5^{l-} - \mathrm{C}_4^{l+} - \mathrm{C}_3^{u+} - \mathrm{E}_r \\ & \Rightarrow \mathrm{E}_l' - \mathrm{E}_r. \end{aligned} \tag{18}$$

Neither part has any crossing. Thus, segment ${}^{u}_{5}\mathrm{L}_1^l$ is not a pulling segment for tightening. The two parts, P_{21} and P_{22}, produced by cutting segment ${}^{l}_{1}\mathrm{L}_2^u$ can be described as follows:

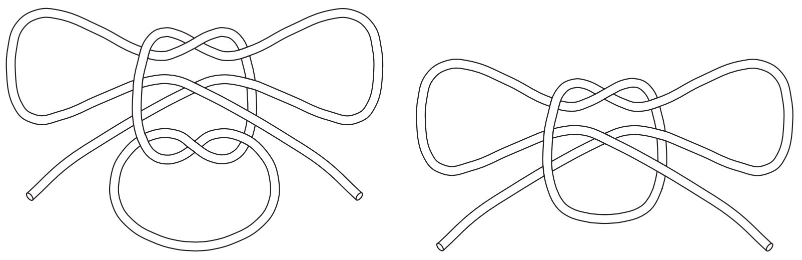
$$\begin{aligned}\mathrm{P}_{21}\ :\ & \mathrm{E}_l - \mathrm{C}_1^{u-} - \mathrm{C}_2^{l-} - \mathrm{C}_3^{l+} - \mathrm{C}_4^{u+} - \mathrm{C}_5^{u-} - \mathrm{C}_1^{l-} - \mathrm{E}_r' \\ & \Rightarrow \mathrm{E}_l - \mathrm{C}_1^{u-} - \mathrm{C}_1^{l-} - \mathrm{E}_r', \end{aligned} \tag{19}$$

$$\begin{aligned}\mathrm{P}_{22}\ :\ & \mathrm{E}_l' - \mathrm{C}_2^{u-} - \mathrm{C}_5^{l-} - \mathrm{C}_4^{l+} - \mathrm{C}_3^{u+} - \mathrm{E}_r \\ & \Rightarrow \mathrm{E}_l' - \mathrm{E}_r. \end{aligned} \tag{20}$$

As part P_{21} can be unknotted by applying operation $\mathrm{UO_I}$ to segment ${}^{u}_{1}\mathrm{L}_1^l$, segment ${}^{l}_{1}\mathrm{L}_2^u$ is not a pulling segment either. The two parts produced by cutting segment ${}^{l}_{5}\mathrm{L}_4^l$ are described as follows:

$$\begin{aligned}\mathrm{P}_{31}\ :\ & \mathrm{E}_l - \mathrm{C}_1^{u-} - \mathrm{C}_2^{l-} - \mathrm{C}_3^{l+} - \mathrm{C}_4^{u+} - \mathrm{C}_5^{u-} - \mathrm{C}_1^{l-} \\ & - \mathrm{C}_2^{u-} - \mathrm{C}_5^{l-} - \mathrm{E}_r' \Rightarrow \mathrm{E}_l - \mathrm{C}_1^{u-} - \mathrm{C}_2^{l-} - \mathrm{C}_5^{u-} \\ & - \mathrm{C}_1^{l-} - \mathrm{C}_2^{u-} - \mathrm{C}_5^{l-} - \mathrm{E}_r', \end{aligned} \tag{21}$$

$$\begin{aligned}\mathrm{P}_{32}\ :\ & \mathrm{E}_l' - \mathrm{C}_4^{l+} - \mathrm{C}_3^{u+} - \mathrm{E}_r \\ & \Rightarrow \mathrm{E}_l' - \mathrm{E}_r. \end{aligned} \tag{22}$$



(a) normal shape (b) simplified shape

Fig. 20. Bowknot.

Part P_{31} does not include any subsequence described by eqs. (3) through (8), suggesting that the knot is partially tightenable and can be tied tightly by pulling segments L_1^u, ${}_5^lL_4^l$, and ${}_3^uL$. Moreover, part P_{31} is tightenable and part P_{32} is regarded as a simple segment. This means that both terminals of part P_{31}, i.e., segments L_1^u and ${}_5^lL_4^l$ must be pulled away from each other to tighten this knot, while segment ${}_3^uL$ is only fixed so that the crossing state does not change. Thus, we can derive how to tighten a knot from the crossing state description using the above procedure.

In general, complex knots consist of a combination of simple knots. For example, a bowknot shown in Figure 20(a) includes an overhand knot. Let us define a continuous subsequence the start and end segments of which are both outer segments and to which operation UO_I/UO_{II} cannot be applied as a *tightenable sub-knot*. We can identify a tightenable sub-knot as one segment in the tightenability check. A bowknot can be simplified as shown in Figure 20(b). The knot shown in Figure 20(b) is a partially tightenable knot with four pulling segments—both bow parts and both terminal segments—which correspond to pulling segments for tightening a bowknot.

Completely tightenable knots or tightenable sub-knots correspond to looped knots in knot theory when both endpoints are connected without changing their crossing states. Figure 21 shows examples of completely tightenable knots and corresponding looped knots in knot theory, which classifies looped knots according to their equivalence. This implies that we can make a database of completely tightenable knots/tightenable sub-knots. Thus, looped knots in knot theory are useful for the tightenability check.

6. Experiments

We proposed methods to represent manipulation processes of a linear object and to determine actions to perform these processes in Sections 2 and 3, respectively. Thus, we can plan knotting/unknotting manipulation of a linear object qualitatively when its initial and objective state are given. Moreover, in Section 5 we proposed a method for tying knots tightly.

In this section, we discuss the effectiveness of our proposed methods. Figure 22 shows an overview of our developed system consisting of a PC, a 6 DOF manipulator, and a CCD camera. A chemical fiber rope, the physical properties of which were unknown, was placed on a table and its shape captured by the camera fixed above the table, corresponding to the projection plane.

The first experiment involved an unknotting manipulation and the second a knotting manipulation.

6.1. Unknotting Manipulation

Figure 23 shows the required manipulation corresponding to untying of an overhand knot. The initial state shown in Figure 23(a) can be represented as $E_l - C_1^- - C_2^- - C_3^- - C_1^- - C_2^- - C_3^- - E_r$ and the objective state shown in Figure 23(b) can be represented as $E_l - E_r$. The assumptions in this case study were as follows:

1. One endpoint of the object is fixed during manipulation; the open square in Figure 23 indicates the position of fixation.

2. One manipulator can be used.

3. The manipulator can approach the object only from the front of the projection plane.

4. The manipulator releases the object whenever one operation is finished.

One manipulation process shown in Figure 24 was derived by our proposed method; it consisted of three transitions, i.e., $N_t = 3$. Furthermore, the manipulation plans illustrated in Figure 25 were selected by considering criterion, N_r, with respect to each operation.

Next, the system determined the current crossing state of the object from a grayscale image. The position of individual crossings were identified by analyzing the image. In this experiment, information regarding which point was upper/lower at each crossing was given manually for simplicity. However, this information can also be obtained automatically using a

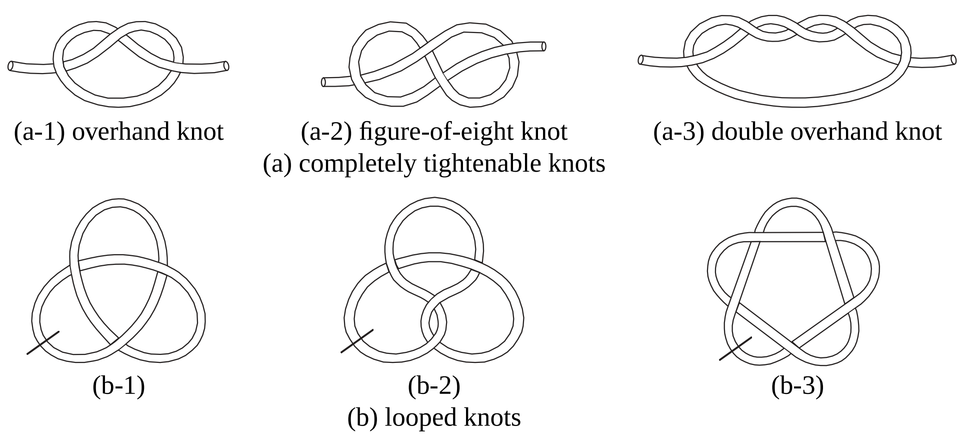

(a-1) overhand knot
(a-2) figure-of-eight knot
(a-3) double overhand knot
(a) completely tightenable knots
(b-1)
(b-2)
(b-3)
(b) looped knots

Fig. 21. Completely tightenable knots and looped knots in knot theory.

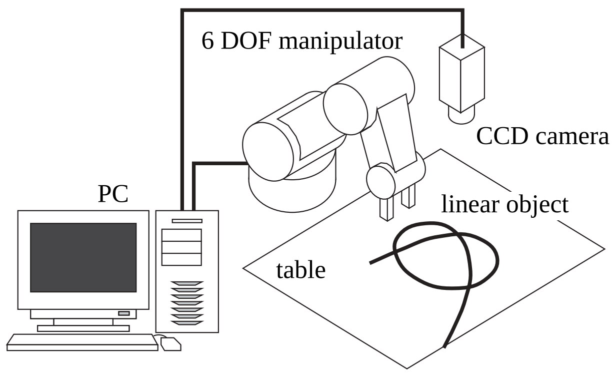


Fig. 22. Overview of developed system.

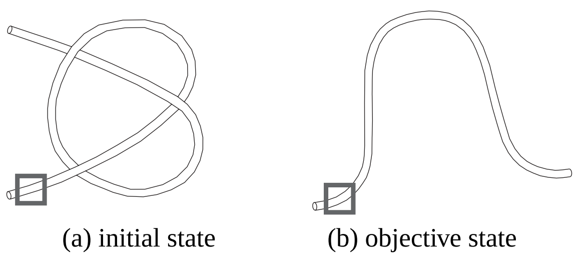


Fig. 23. Required manipulation—untying overhand knot.

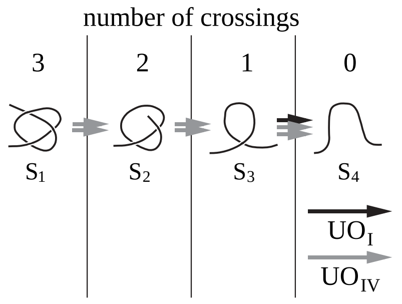


Fig. 24. Generated processes for untying overhand knot.

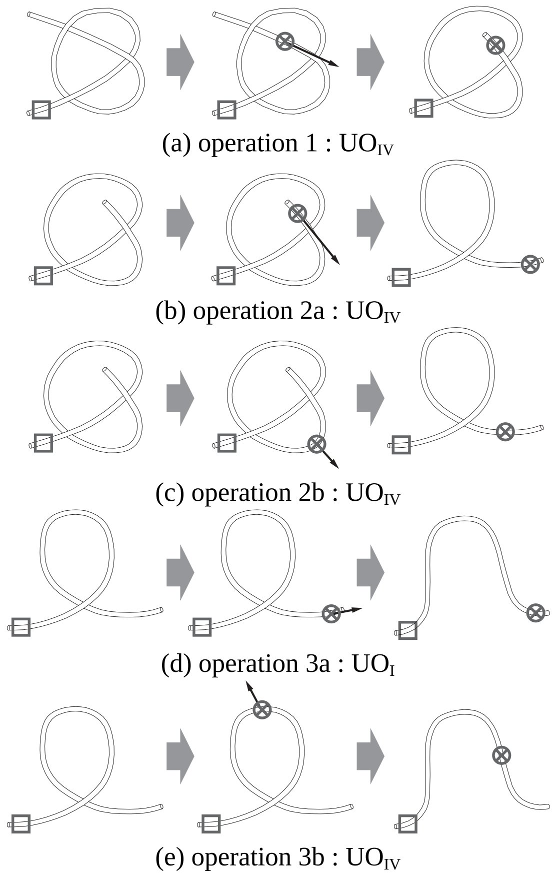


Fig. 25. Generated manipulation plans for untying overhand knot.

stereo camera. We regarded the position of each grasping point as the midpoint of each segment. The directions of axis-1 and axis-2 were determined from the tangent at a grasping point. As the adequate distance of movement for a state transition was unknown, the system checked whether its crossing state was changed after moving the object. Thus, the manipulator grasped, moved, and released the object according to the generated qualitative plan.

In this case study, the system selected operation 2b and operation 3a by considering the space requirements for insertion of the gripper and the motion range of the manipulator. Figure 26 shows the results of this manipulation. The manipulator first pulled the lower terminal in the loop, then took the terminal off the loop, and pulled the terminal away to remove the loop. In this case, the loop decreased in size gradually and was eventually removed. A less flexible object may become kinked before removing the loop. In such cases, the action for operation 3a shown in Figure 25(d) is not adequate. Whether kinking occurs is dependent on physical properties of the linear object. As they are not considered in this qualitative planning method, quantitive analysis is required to verify the validity of the derived actions.

6.2. Knotting Manipulation

The second experiment involved tying a slipknot as shown in Figure 15. The assumptions in this case study were the same as in the first experiment. The possible manipulation processes as shown in Figure 16 were derived by our proposed method. Considering criteria N_t, we selected process $S_{11} \rightarrow S_5 \rightarrow S_2 \rightarrow S_1$ in Figure 16. Next, the actions to be taken were determined. From theorem 2, we selected three actions shown in Figure 8(a-1) for transition $S_{11} \rightarrow S_5$, Figure 8(b-3) for transition $S_5 \rightarrow S_2$, and Figure 8(b-16) for transition $S_2 \rightarrow S_1$. The manipulation plans illustrated in Figure 27 were then derived. Next, we examined tightening of this slipknot. Recall that a slipknot can be tied tightly by pulling segments L_1^u, ${}_5^lL_4^l$, and ${}_3^uL$, which correspond to both terminal segments and the bow segment as shown in Figure 28. From assumption 1, the right terminal segment was fixed. Therefore, we had to grasp and pull away the left terminal segment and the bow segment. However, from assumption 2, two segments could not be grasped simultaneously. In this experiment, the left terminal segment was fixed with a weight as shown in Figure 29(h). Then, we tightened the slipknot by pulling the bow segment alone. Figure 29 shows the results of this knotting manipulation. As shown in this figure, a slipknot could be tied by repeated overlapping of one part of the linear object with its other part. These observations indicated that we can make several knots using a simpler mechanism than the human arm/hand and in a manner different to that performed by a human. Thus, we concluded that our proposed method is useful for automatic planning and execution of linear object manipulation.

7. Conclusions

In this paper, we proposed a planning method for knotting/unknotting of deformable linear objects. First, we proposed a description of the topological state of a linear object. Such objects can be represented as finite crossing states including three properties: number/sequence of crossings, which point is upper/lower at each crossing, and whether each crossing is left-handed helical or right-handed helical. Second, four basic operations were introduced: three Reidemeister moves and an additional operation to cross/uncross a terminal segment of a linear object. A state transition between crossing states corresponds to a basic operation changing the number of crossings or permuting their sequence. Then, possible sequences of crossing state transitions, i.e., possible manipulation processes can be generated once the initial and the objective states are given. Third, actions were defined to realize the derived manipulation processes. They correspond to adequate sets of grasping points, their directions of movement, and the direction of approach of the manipulator to each grasping point. Then, qualitative manipulation plans, i.e., possible processes and actions, can be generated on a computer. These proposals and definitions suggested that theoretically any knotting manipulation of a linear object can be realized by a one-handed robot with three translational DOF and one rotational DOF if the object is placed on a table. Furthermore, criteria for evaluation of manipulation plans were introduced to reduce the number of candidate manipulation plans. In general, knots fulfill their fixing function after they are tied tightly. Which part of a linear object should be pulled away for tightening depends on the topological state of the object. Therefore, we established a planning method for tying knots tightly. Using this method, it is possible to determine the parts that should be pulled to tighten knots. Finally, we demonstrated that our system can plan and execute both knotting and unknotting manipulations of a linear object.

Application of the method proposed in this paper allows the planning of linear object manipulation in a qualitative manner. However, this method is not suitable for determining the grasping points of manipulators and their trajectories in detail. Previously, we developed methods for static modeling (Wakamatsu and Hirai 2004) and dynamic modeling (Wakamatsu et al. 2005) of linear object deformation. These modeling methods can be applied to detailed planning as they can be used to estimate the geometrical shape of a knotted linear object. Figure 30 shows the computed shape of an overhand knot. Once a qualitative plan is selected by using our proposed method, the number of crossings after executing one operation is determined. The number of grasping points to perform the operation is also determined. Then, inputting them as constraints, both position of grasping points and their appropriate trajectories in each operation can be calculated using our physical simulation and deformation path planning. Thus, a manipulation strategy will be derived by combining qualitative planning with quantitative analysis.

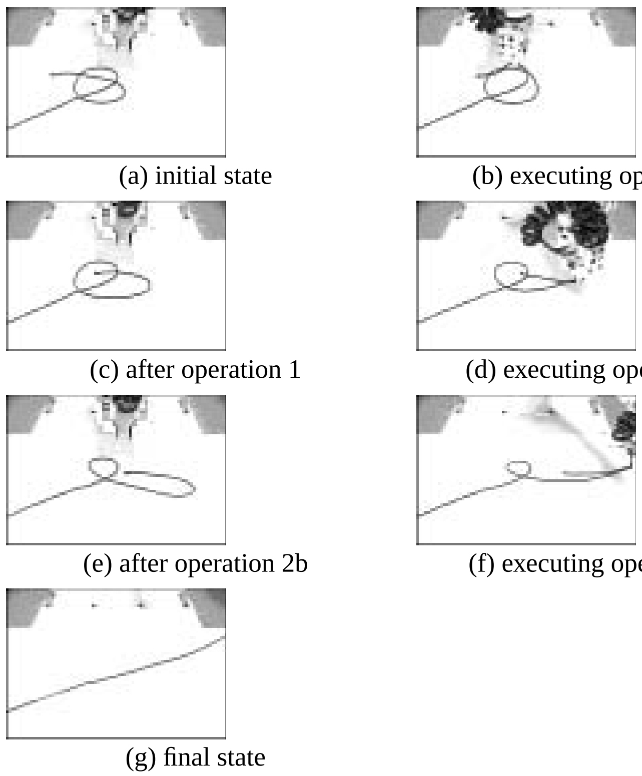

(a) initial state
(b) executing operation 1
(c) after operation 1
(d) executing operation 2b
(e) after operation 2b
(f) executing operation 3a
(g) final state

Fig. 26. Results of experiment for untying overhand knot.

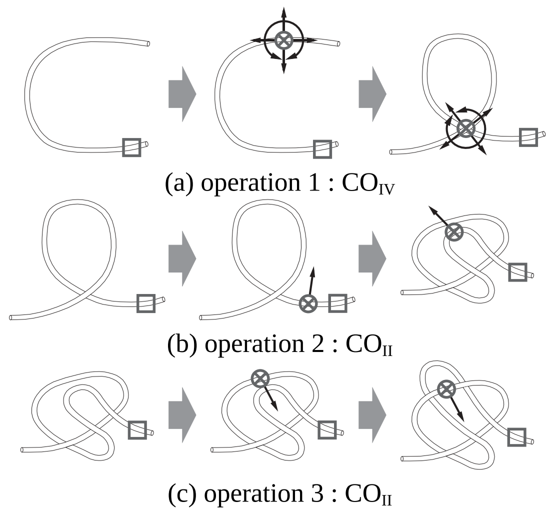

(a) operation 1 : CO_{IV}
(b) operation 2 : CO_{II}
(c) operation 3 : CO_{II}

Fig. 27. Generated manipulation plans for tying slipknot.

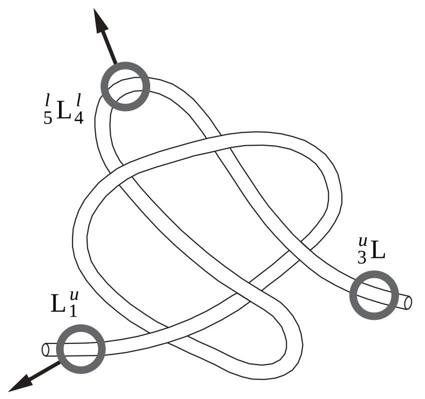


Fig. 28. Pulling segments for tightening slipknot.

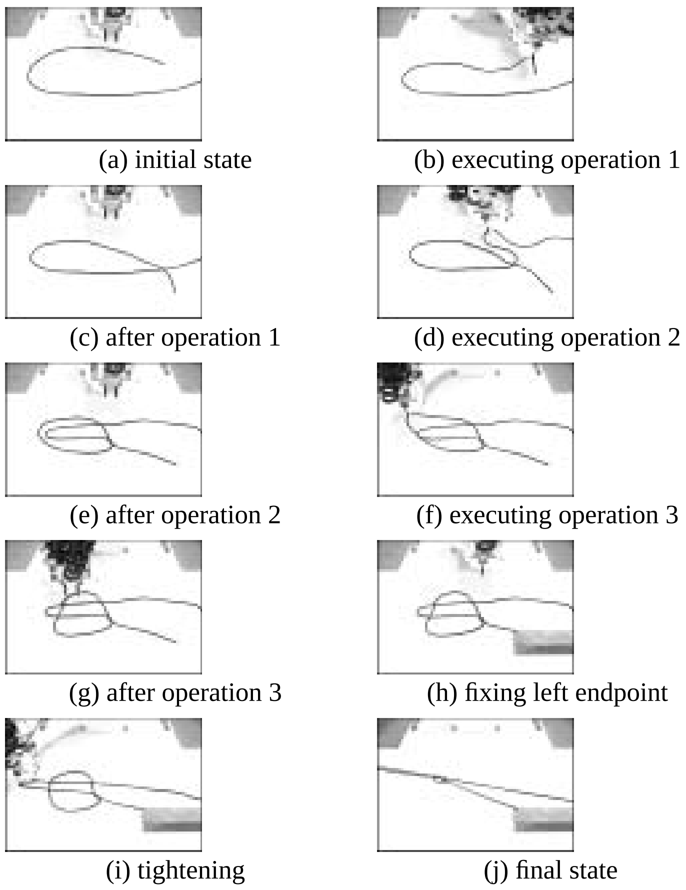


Fig. 29. Results of manipulation for tying slipknot.

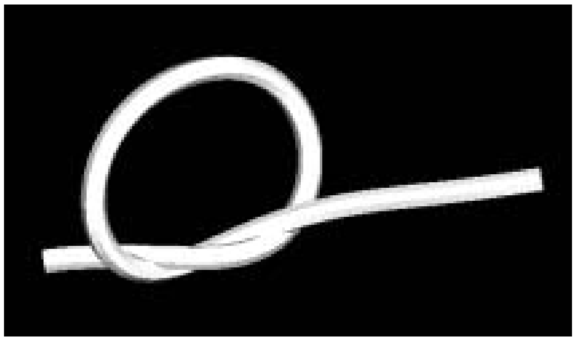


Fig. 30. Computed shape of overhand knot.

References

Acker, J. and Henrich, D. 2005. Manipulation of Deformable Linear Object; From Geometric Model Towards Program Generation. *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 1553–1559.

Adams, C. C. 1994. *The Knot Book: An Elementary Introduction to the Mathematical Theory of Knots*, Henry Holt & Co.

Budworth, G. 2002. *The Illustrated Encyclopedia of Knots*, The Lyons Press.

Chirikjian, G. S. and Burdick, J. W. 1994. A Modal Approach to Hyper-Redundant Manipulator Kinematics. *IEEE Trans. on Robotics and Automation* 10(3):343–354.

Daldegan, A., Thalmann, N. M., Kurihara, T., and Thalmann, D. 1993. An Integrated System for Modeling, Animating and Rendering Hair. *Computer Graphics Forum (Eurographics '93)* 12(3):211–221.

Desai, R. S. and Volz, R. A. 1989. Identification and Verification of Termination Conditions in Fine Motion in Presence of Sensor Errors and Geometric Uncertainties. *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 800–807.

Gray, A. 1993. *Modern Differential Geometry of Curves and Surfaces*, CRC Press.

Henrich, D., Ogasawara, T., and Wörn, H. 1999. Manipulating Deformable Linear Objects – Contact States and Point Contacts. *Proc. IEEE International Symposium on Assembly and Task Planning*, pp. 198–204.

Henrich, D. and Wörn, H. eds. 2000. *Robot Manipulation of Deformable Objects*, Springer–Verlag, Advanced Manufacturing Series.

Hopcroft, J. E., Kearney, J. K., and Krafft, D., B. 1991. A Case Study of Flexible Object Manipulation. *International Journal of Robotics Research* 10(1):41–50.

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

Irvine, H. M. 1981. *Cable Structures*, MIT Press.

Kühnapfel, U., Çakmak, H. K., and Maass, H. 2000. Endoscopic surgery training using virtual reality and deformable tissue simulation. *Computers & Graphics* 24(5):671–682.

Ladd, A. M. and Kavraki, L. E. 2004. Using Motion Planning for Knot Untangling. *Int. J. of Robotics Research* 23(7–8):797–808.

Leaf, B. G. A. V. 1960. Models of the Plain-Knitted Loop. *Journal of the Textile Institute Transactions*, pp. 49–58.

Lozano-Pérez, T., Mason, M. T., and Taylor, R. H. 1984. Automatic Synthesis of Fine Motion Strategies for Robots. *Int. J. of Robotics Research* 3(1):3–24.

Matsuno, T., Fukuda, T., and Arai, F. 2001. Flexible Rope Manipulation by Dual Manipulator System Using Vision Sensor. *Proc. Int. Conf. Advanced Intelligent Mechatronics*, pp. 677–682.

Mochiyama, H. and Suzuki, T. 2003. Kinematics and Dynamics of a Cable-like Hyper-flexible Manipulator. *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 3672–3677.

Moll, M. and Kavraki, L. E. 2005. Path Planning for Variable Resolution Minimal-Energy Curves of Constant Length. *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2142–2147.

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

Nakagaki, H., Kitagaki, K., Ogasawara, T., and Tsukune, H. 1997. Study of Deformation and Insertion Tasks of a Flexible Wire. *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2397–2402.

Nienhuys, H.-W. and van der Stappen, A. F. 2004. A Computational technique for Interactive Needle Insertion in 3D Nonlinear Material. *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2061–2067.

Pai, D. K. 2002. STRANDS: Interactive Simulation of Thin Solids using Cosserat Models. *Computer Graphics Forum* 21(3):347–352.

Phillips, J., Ladd, A., and Kavraki, L. E. 2002. Simulated Knot Tying. *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 841–846.

Reidemeister, K. 1983. *Knot Theory*, BCS Associates.

Remde, A. and Henrich, D. 2000. Direct and Inverse Simulation of Deformable Linear Objects. In Henrich, D. and Wörn, H. eds., *Robot Manipulation of Deformable Objects*, Springer–Verlag, Advanced Manufacturing Series, pp. 43–70.

Rosenblum, R. E., Carlson, W. E., and Tripp, E. 1991. Simulating the Structure and Dynamics of Human Hair: Modelling, Rendering and Animation. *Journal of Visualization and Computer Animation* 2(4):141–148.

Wakamatsu, H., Tsumaya, A., Shirase, K., and Arai, E. 2001. Representation of Deformable Thin Object Manipulation Based on Contact State. *Proc. 5th Japan-France Congress & 3rd Asia-Europe Congress on Mechatronics*, pp. 519–524.

Wakamatsu, H. and Hirai, S. 2004. Static Modeling of Linear Object Deformation Based on Differential Geometry. *Int. J. Robotics Research* 23(3):293–311.

Wakamatsu, H., Takahashi, K., and Hirai, S. 2005. Dynamic Modeling of Linear Object Deformation based on Differential Geometry Coordinates. *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 1040–1045.

Webster III, R. J., Cowan, N. J., Chirikjan, G., and Okamura, A. M. 2004. Nonholonomic Modeling of Needle Steering. *9th International Symposium on Experimental Robotics*.

Weil, J. 1986. The Synthesis of Cloth Objects. *Computer Graphics* 20(4):49–54.

Yue, S. and Henrich, D. 2002. Manipulating Deformable Linear Objects: Sensor-Based Fast Manipulation during Vibration. *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2467–2472.

Zheng, Y. F., Pei, R., and Chen, C. 1991. Strategies for Automatic Assembly of Deformable Objects. *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2598–2603.