Manipulation of Deformable Linear Objects: From Geometric Model Towards Program Generation

Jürgen Acker and Dominik Henrich
Lehrstuhl Angewandte Informatik III
Fakultät für Mathematik, Physik und Angewandte Informatik
Universität Bayreuth, D-95445 Bayreuth, Germany
[Juergen.Acker | Dominik.Henrich] @ uni-bayreuth.de

Abstract - This paper discusses the handling of deformable linear objects (DLOs), such as hoses, wires, or leaf springs in a polyhedral environment. It investigates the formulation of assembly or disassembly tasks based on contact states. The result is an approach that facilitates the automatic extraction of robot programs from demonstrations in virtual reality and provides a base for the parameterization of detection algorithms. For this purpose, a contact state model for the description of assembly or disassembly tasks of DLOs is presented. It is described how the contact states can be derived from a geometric model of both the DLO and the environment. Such a model may be obtained by a simulation of the manipulation tasks in virtual reality. Further, the possible transitions between the contact states are classified into general transition classes. Those transition classes enable the selection of algorithms to detect such contact state transitions.

Index Terms - Keywords: Contact states, deformable linear objects, manipulation, assembly

I. Introduction

In industrial assembly tasks, rigid as well as non-rigid objects must be handled. However, industrial robot systems work only with rigid objects. Ropes, wires, hoses, steel springs or other deformable linear objects (*DLO*s) are usually handled by human workers only. For robots, the main problem in handling DLOs is coping with inherent uncertainties concerning the exact shape of an individual DLO. Additionally, the shape of a DLO also changes during the manipulation process due to gravity and contact forces. The obvious approach is the use of sensors to compensate for these uncertainties.

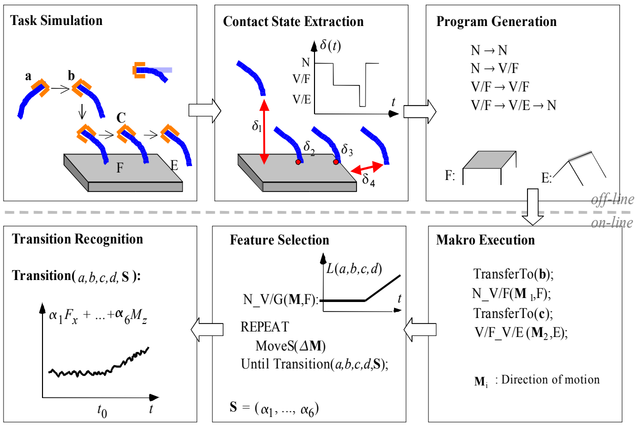
Often, research involving DLOs, e.g. [7], only investigates single tasks like “peg-in-hole” and it remains unclear how these special solutions can be applied to more general cases. Therefore, a systematic approach for manipulation of DLOs is needed.

The manipulation of rigid objects has been studied for years. Thus, systematic approaches exist in this domain, such as [2] and [5], where assembly or disassembly tasks are described based on contact states. The transitions between contact states are carried out by robust and flexible routines (*skills*). Those skills encapsulate the programming-intensive sensor data processing and can be used for solving complex assembly or disassembly problems, like the disassembly of a valve [2]. Further, the description of manipulation tasks was employed for planning such manipulation tasks.

Since contact states proved to be useful in the domain of rigid objects, [10] and especially [3] applied this approach to deformable objects. Both groups concentrated on DLOs but the representations differ. The contact state model in [3] regards only single contact points between a DLO and convex environment objects, whereas [10] uses a set of contact patterns and transitions between them for the purpose of planning manipulation tasks.

In general, planning is a difficult task for computers, since it is NP-hard. Therefore, other methods have been considered, such as programming by demonstration. In the domain of manipulating deformable objects, [4] realizes a skill controller based on programming by demonstration in combination with contact states. The design strategy of this controller depends on a data profile generated by a human worker. The recorded sensor data profile is further processed to identify the control parameters. The mapping of the continuous data profile to contact states and discrete transitions respectively has to be done manually, according to [4]; this reveals two disadvantages of this approach. First, such a demonstration has to be carried out with the same sensors that are later used for the manipulation task. Second, a human is not only necessary for the demonstration but for the identification of control parameters as well.

Therefore, in [6] the programming by demonstration paradigm is performed in virtual reality and the human worker uses a haptic input device to demonstrate the task. Fig. 1 shows the complete system concept, from the demonstration of the task to the sensor based detection of contact state transitions. The virtual reality covers the task simulation and portions of the contact state extraction. Here, the extraction is performed automatically based on the limited contact state model presented in [3]. The sensor based on-line part of this system was investigated in [9] and [1].



```
TransferTo(b);
N_V/F(M 1,F);
TransferTo(c);
V/F_V/E (M2,E);
```

$\mathbf{M}_i$: Direction of motion

```
N_V/G(M,F):
REPEAT
   MoveS(ΔM)
Until Transition(a,b,c,d,S);
```

$\mathbf{S} = (\alpha_1, \ldots, \alpha_6)$

Fig. 1 Illustration of the overall process for programming and execution of sensor-based robot programs taken from [6].

This papers covers the contact state extraction and program generation steps of the system shown in Fig. 1. We thereby refine and extend the contact state model from [3] to a polyhedral environment of convex and non-convex objects and to include multiple contacts. Further, we show how topological contact states can be extracted from the geometrical model used in virtual reality. This topological model is transferred to a more abstract discrete contact state model that provides a base for the program generation.

In Section II, the underlying geometrical models of the environment and the DLO are described as used in virtual reality. Section III introduces the topological model that we derive from the geometrical model. The next step towards program generation is the transformation of the topological model to the contact state model in Section IV. In Section V the transitions between contact states are investigated. Based on Sections III through V, we describe a manipulation task as an example in Section VI. Finally, Section VII provides a summary and gives an overview of further research.

II. Geometrical Model and Basic Assumptions

Because the *geometrical model* is the most concrete and it is used in the virtual reality portion of the complete system (Fig. 1), we start from this point. First, we define every deformable object with a diameter much smaller than its length as "linear". We assume elastic deformation and do not consider plastic or extensional deformation. Further, we assume torsion is negligible, thus we do not state it explicitly. In the ideal case, the diameter of the DLO is zero. Therefore, we model DLOs as a single curved line. More precisely, the DLO is modelled as a sequence of points $\left(P_i\right)_{i=0}^{n}$, where each point P_i is defined by its position vector $\vec{p}_i$. The number n of points is assumed to be high enough for a sufficient precision (Fig. 2, left). Therefore, n may depend on the shape of the DLO. The first point of the sequence is also called the *start point* (V_S) and the last point is the *end point* (V_E) of the DLO. Further, we define the *DLO vector* $\vec{D}_i$ of a given point P_i (1).

$$\vec{D}_i = \vec{p}_{i+1} - \vec{p}_i \qquad (1)$$

Thus, $\vec{D}_i$ can be used as the tangent along the DLO for each point of th DLO.

Now that we have described the geometric model of the DLO, we now describe the environment. We assume a polyhedral environment of rigid objects as in [3] but we extend the environment from convex objects only to include non-convex objects. Each non-convex object is split up into the smallest possible number of convex objects.

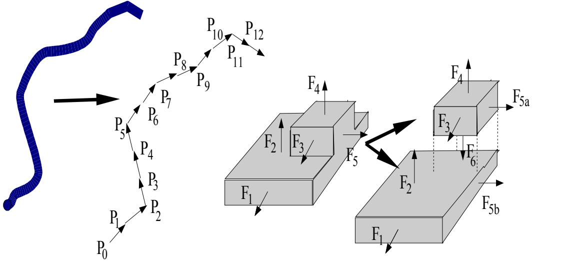


Fig. 2 Geometrical Models of DLO (left) and non-convex environment objects (right) and its decomposition into two convex objects.

Each convex object is modelled as a set of surfaces. For each surface F_i the normal vector $\vec{N}(F_i)$ and one point $P(F_i)$ within the surface are given. All normal vectors point away from the object´s surface, as shown on the right in Fig. 2. Due to the geometric model of the ideal DLO, we postulate that two environment objects intersect if the smallest distance between both objects is less or equal to the diameter of the DLO.

Recapitulating, the geometrical model contains all environmental objects and the exact shape of the DLO. Additionally, all angles and all distances are known in this model.

III. Topological Contact Model

From the geometric model we can derive the topological model. This model describes the *topological contact situation* i.e. all contacts between the DLO and environmental objects. Thus, in this model the environment and the DLO are only well defined at contact zones (Fig. 3).

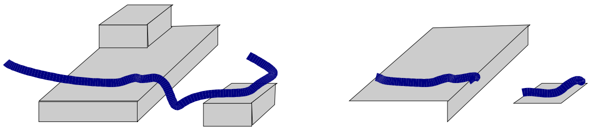


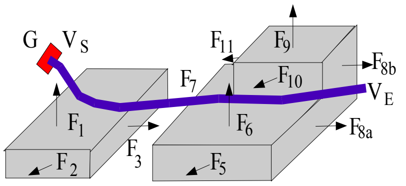
Fig. 3 Geometrical model of a contact situation (left); and contact zones of the situation from the left which remain in the topological model (right).

In order to describe those contact zones, we first introduce some *environment primitives EP*. The first environment primitives are the surfaces of the environment objects, further called faces F_i. Further, the intersection of of n directly adjacent faces F_i, F_j, ... F_n belonging to the same convex object are denoted as $F_i \cap F_j \cap ... \cap F_n$. The intersection $F_i \cap F_j$ may be abbreviated as edge $E_{i,j}$ and the intersection $F_i \cap F_j \cap .. \cap F_n$ may be abbreviated as $V_{i,\,j,\,..,n}$. Since the topological contact model is restricted to contact zones (Fig. 3), we introduce the empty object N representing empty space. For convenience, we also introduce the object $G_{\vec{z}}$, representing the gripper of the robot. The annotated vector $\vec{z}$ represents the normal vector of the gripper and is parallel to the z-axis of the tool center point (TCP). Thus, the environment primitives are the set $EP := \{N, G_{\vec{z}}, F_i, E_{i,j}, \mathrm{V}_{i,\,j,\,...,\,n}\}$.

Corresponding to the environment primitives, we introduce V and E as DLO primitives. The primitive vertex V is used for V_S and V_E. For all other points along the DLO, the primitive edge E is used. DLO primitives are indexed by their DLO vectors.

Now let us define the different types of contacts in order to define a contact situation. A *basic contact* is a pair P/Q where P is a DLO primitive and Q is a DLO or an environment primitive. Since for assembly tasks contacts between DLOs and the environment are the most important, we will more closely examine on contacts between DLO and environment primitives.

If one point of the DLO is in contact with more than one geometric primitive except N, the contact is called a *multiple contact,* otherwise it is a *single contact.* The basic contacts of a multiple contact are concatenated by the operator “+”. The *contact situation* is now the ordered sequence of all single or multiple contacts from V_S to V_E (Fig. 4). Here, the contacts are concatenated with logical *AND* (^). If *m* adjacent points of the DLO P_i, P_{i+1}, .., P_{i+m} and ($0<i<i+m<n$) share the same multiple or simple contact, then both contacts are contracted to one contact. At this point, the DLO vectors may differ due to the bending of the DLO. Therefore, the contracted contact is indexed by the DLO vector of the point $P_{i+½m}$. Any non-empty sequence between two contacts with N that does not contain further contacts with N is denoted as a *successional contact* (Fig. 4).



$$V_S/G_{\vec{z}}\wedge E_{\vec{a}}/N\wedge \underbrace{E_{\vec{b}}/F_1\wedge E_{\vec{c}}/(F_1\cap F_3)}_{successional\ \ contact}\wedge E_{\vec{d}}/N\wedge E_{\vec{e}}/(F_7\cap F_6)\wedge E_{\vec{f}}/F_6\wedge$$
$$E_{\vec{g}}/F_6+E_{\vec{g}}/(F_{11}\cap F_{10}\cap F_{12})\wedge E_{\vec{h}}/F_6+E_{\vec{h}}/(F_{10}\cap F_{12})\wedge$$
$$E_{\vec{i}}/(F_6\cap F_8 a)+E_{\vec{i}}/(F_{10}\cap F_{12}\cap F_8 b)\wedge E_{\vec{j}}/N\wedge V_E/N$$

Fig. 4 Geometrical contact situation and resulting topological contact model (The face F_{12} introduced by the decomposition of non-convex environment objects are not visible)

In summary, the topological contact model describes all contacts in a topological manner, since it contains all contacts and the complete information about the necessary angles for any contact. On the other hand, all distances, sizes, the exact shape of the DLO and the faces that do not contact the DLO are dropped from the model.

IV. Discrete Contact State Model

For the automatic program generation, we need macros for each transition or class of transitions between contact states. The topological model from Section III is not fully discrete due to the exact DLO vectors and the normals for each face. But we need a pure *discrete contact state model* in order to appoint the contact state transitions automatically to macros. The key to such a model is the stability of a contact. For rigid objects, a small motion away from the obstacle always leads to a loss of the contact (Fig. 5). Due to the intrinsic compliance of DLOs, small motions do not always result in a loss of the contact for a DLO.

A. Contact Force

There is always a resulting force if the DLO is deformed due to the contact. Therefore, we model this intrinsic compliance and postulate a contact force $F_{\vec{C}}$ between the DLO and the environment primitive for every contact. This contact force ensures the stability of at least some contacts due to the deformation of the DLO.

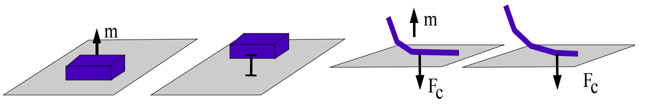


Fig. 5 A small motion away from an obstacle leads to a loss of the contact for a rigid object (left) and with an effective contact force for DLOs such a motion does not change the contact state (right).

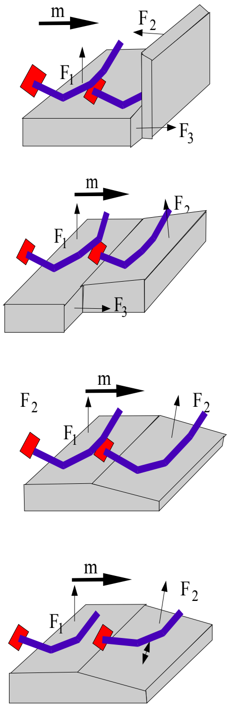
The optimal direction to ensure the stability of contacts of this contact force for any single or multiple contact is given by the *contact vector* $\vec{C}$ of such a contact (2).

$$\begin{array}{lcr}
\vec{C}(D/N) & := & \vec{0} \\
\vec{C}(D/G_{\vec{z}}) & := & \vec{z} \\
\vec{C}(D/F_i) & := & -\vec{N}(F_i) \\
\vec{C}(D/F_i\cap F_j\cap..\cap F_k) & := & \vec{C}(D/F_i)+..+\vec{C}(D/F_k) \\
\vec{C}(D/X_1+..+D/X_n) & := & \vec{C}(D/X_1)+..+\vec{C}(D/X_n) \\
 & & \forall D\in\{V,E\},\forall X_j\in EP
\end{array} \quad (2)$$

The contact vector $\vec{C}$ of unit length ensures that the contact force $F_{\vec{C}}$ is normal to the environment objects at the contact point. Thus, $\vec{0}$ is only permitted for $\vec{C}(V/N)$ and $\vec{C}(E/N)$. Let F_i and F_j belong to two different convex environment objects and $F_i \| F_j$. In this case, it is obvious from (2) that $\vec{C}(F_i+F_j)$ is $\vec{0}$. But due to assumption that the DLOs are ideal and the resulting assumption about distances (Section II), the DLO can never be in contact with two parallel faces. Therefore, $\vec{C}$ and the direction of the contact force are always well defined. Please also note, that the contact vector depends only on the environment objects but not on the DLO.

B. Stability and Transitions

In total, we can identify three different classes of stability. A contact is considered to be *stable* if any small motion in any direction does not result in a change of the contact state. The contacts which are not stable can further subdivided into semi-stable and instable contacts.



$$V_S/G_{\vec{z}}\wedge E_{\vec{c}}\wedge E_{\vec{a}}/F_1\wedge E_{\vec{b}}/N\wedge V_E/N$$
$$\rightarrow V_S/G_{\vec{z}}\wedge E_{\vec{c}}\wedge E_{\vec{a}}/(F_1\cap F_3) \qquad \text{(a)}$$
$$+E_{\vec{a}}/F_2\wedge E_{\vec{b}}/N\wedge V_E/N$$

$$V_S/G_{\vec{z}}\wedge E_{\vec{c}}\wedge E_{\vec{a}}/F_1\wedge E_{\vec{b}}/N\wedge V_E/N$$
$$\rightarrow V_E/G_{\vec{z}}\wedge E_{\vec{c}}/N\wedge E_{\vec{a}}/(F_1\cap F_3)$$
$$+E_{\vec{a}}/(F_2\cap F_4)\wedge E_{\vec{b}}/N\wedge V_E/N\rightarrow \qquad \text{(b)}$$
$$V_S/G_{\vec{z}}\wedge E_{\vec{c}}/N\wedge E_{\vec{c}}/F_2\wedge$$
$$E_{\vec{b}}/N\wedge V_E/N$$

$$V_S/G_{\vec{z}}\wedge E_{\vec{d}}\wedge E_{\vec{a}}/F_1\wedge E_{\vec{e}}\wedge V_E/N\rightarrow$$
$$V_S/G_{\vec{z}}\wedge E_{\vec{d}}/N\wedge E_{\vec{a}}/(F_1\cap F_2)\wedge \qquad \text{(c)}$$
$$E_{\vec{e}}/N\wedge V_E/N\rightarrow V_S/G_{\vec{z}}\wedge E_{\vec{e}}/N\wedge E_{\vec{c}}/$$
$$\wedge E_{\vec{f}}/N\wedge V_S/N$$

$$V_S/G_{\vec{z}}\wedge E_{\vec{d}}/N\wedge E_{\vec{a}}/F_1\wedge E_{\vec{d}}/N\wedge$$
$$V_S/N\rightarrow V_S/G_{\vec{z}}\wedge E/\vec{c}/N\wedge \qquad \text{(d)}$$
$$E_{\vec{a}}/(F_1\cap F_2)\wedge E_{\vec{d}}/N\wedge V_E/N\rightarrow$$
$$V_S/G_{\vec{z}}\wedge E_{\vec{e}}/N\wedge V_E/N$$

Fig. 6 Four similar topological contact situations in which a motion *m* parallel to the face is initiated from an *E/F* contact. Due to the angle between the normals of F_1 and F_2 a stable (a), transient (b) or spontaneous transition (c,d) occurs. Please note the faces F_3 and F_4 are not always visible in the geometrical view (Fig 2).

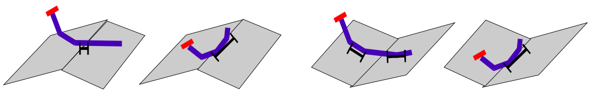
A contact C is *semi-stable* if and only if C is not stable and the loss of C is combined with further bending of the DLO. A contact C is *instable* if and only if C is not stable and the loss of C is possible without further bending of the DLO. Now that the different classes of stability have been defined, the effects of the semi- and instable contact states are to be investigated. We therefore examine the situations shown in Fig. 6.

The same initial situation is used in each of the four images in Fig. 6 and the DLO is dragged along the surface of a face towards an edge. In the first transition sequence (Fig. 6a), the DLO collides with another face and the result is a stable contact (concave edge). In the next sequence, the angle between the faces F_1 and F_2 is nearly 180° but the edge is still concave. In this case, the second multiple contact state is not stable, since a small motion, e.g. in the direction *m,* will cause the multiple contact to be lost. The result is another single contact and the DLO bends more due to the reduced distance between the obstacle and the gripped end of the DLO. Therefore, the intermediate contact state is semi-stable and the transition is called a *transient* transition. In general, the critical angle between stable and transient transitions depends not only on the degree of the angle but also on the ensuing friction. Usually, the critical angle is close to 180°. In the last two images Fig. 6c and Fig. 6d, the angle between the faces F_1 and F_2 is more than 180° (convex edge). Here, the intermediate contact is instable. The resulting stable contact state depends not only on the angle between both faces but also on the deformation of the DLO, making the control of such *spontaneous* transitions almost impossible. A more detailed investigation of transition conditions for spontaneous transitions can be found in [8].

C. Discrete Contact States

Stable contacts are most significant for the description of manipulation tasks, since semi-stable or instable contacts are almost impossible to maintain. On the other hand, semi-stable and instable contacts may appear as intermediate contact states between two stable contacts and indicate what type of transition (spontaneous or transient) occurred between the stable contact states, as shown above. Thus, semi- as well as instable contacts are still modelled at this abstraction level. The discrete contacts states are basically the combination of the two basic DLO primitives (*E, V*), the three basic environment primitives (*V, E, F*) and the stability classes *(S, T, I)*. Since face contacts are always stable, the combinations V/F^T, V/F^I, E/F^T and E/F^I do not exist. Therefore, only the vertex and edge contacts are annotated with the stability classes e.g. V/V^S.

The remaining contact states can be divided in two groups based on the intersection of the DLO and the environment. Since the intersection is either a point or a line, a *point* respectively a *line* contact implies that the intersection between the DLO and the environment objects at the contact point is a point respectively a line.



(a) (b) (c) (d)

Fig. 7 Different subtypes of *E/E*-type contacts on convex (a, b) and non-convex (c, d) edges.

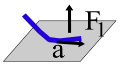
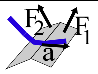
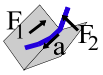
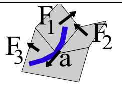
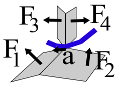
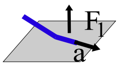
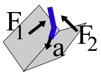
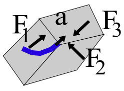
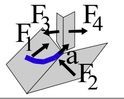
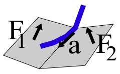
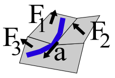
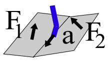
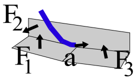
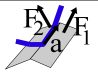
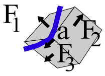
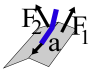
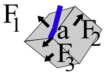
Stability	Discrete Contact States	*card*	Geometric Situation	Topological Contact State of Single or Multiple Contacts
stable	E/F	8		$E_{\vec{a}}/F_1$
	E/E^P	6		$E_{\vec{a}}/(F_1 \cap F_2)$
	E/E^{LS}	9		$E_{\vec{a}}/(F_1 \cap F_3) + E_{\vec{a}}/(F_2 \cap F_4)$
	E/V^S	7		$E_{\vec{a}}/(F_1 \cap F_3 \cap F_5)$ $+ E_{\vec{a}}/(F_2 \cap F_3 \cap F_5)$
				$E_{\vec{a}}/(F_1 \cap F_2 \cap F_5)$ $+ E_{\vec{a}}/(F_4 \cap F_3 \cap F_6)$
	V/F	7		$V_{\vec{a}}/F_1$
	V/E^{LS}	8		$V_{\vec{a}}/(F_1 \cap F_3)$ $+ V_{\vec{a}}/(F_2 \cap F_3)$
	V/V^S	9		$V_{\vec{a}}/(F_1 \cap F_6 \cap F_7) + V_{\vec{a}}/F_2$ $+ V_{\vec{a}}/(F_3 \cap F_8 \cap F_9)$
				$V_{\vec{a}}/(F_1 \cap F_5)$ $+ V_{\vec{a}}/(F_2 \cap F_6)$ $+ V_{\vec{a}}/(F_3 \cap F_4 \cap F_7)$
semi-stable	E/E^{LT}	9		$E_{\vec{a}}/(F_1 \cap F_3)$ $+ E_{\vec{a}}/(F_2 \cap F_4)$
	E/V^T	7		$E_{\vec{a}}/(F_1 \cap F_3 \cap F_4)$ $+ E_{\vec{a}}/(F_2 \cap F_3 \cap F_5)$
	V/E^{LT}	8		$V_{\vec{a}}/(F_1 \cap F_3)$ $+ V_{\vec{a}}/(F_2 \cap F_3)$
	V/V^T	9		$V_{\vec{a}}/(F_1 \cap F_6 \cap F_7) + V_{\vec{a}}/F_2$ $+ V_{\vec{a}}/(F_3 \cap F_8 \cap F_9)$
in-stable	E/E^{LI}	6		$E_{\vec{a}}/(F_1 \cap F_2)$
	E/V^I	4		$E_{\vec{a}}/(F_1 \cap F_2 \cap F_3)$
	V/E^{LI}	5		$V_{\vec{a}}/(F_1 \cap F_2)$
	V/V^I	3		$V_{\vec{a}}/(F_1 \cap F_2 \cap F_3)$

Fig. 8 All discrete contact states and some typical geometrical and topological situations for each stable, semi-stable and instable discrete contact states. The cardinality *card* is discussed in Section V.B. The geometric situations are reduced to the minimal set of faces for the sake of clarity (Fig. 2). Therefore, some topological descriptions contain more faces than the corresponding geometrical situation.

All contacts with a vertex as DLO or environment primitive like V/F or E/V^S are point contacts. The contact state E/F is always a line contact since the contact force presses the DLO against the obstacle, causing a deformation. Thus, the force flattens the DLO at the contact point resulting in a short line as intersection. But *E/E*-type contacts can belong to both groups (Fig. 7). Here, the intersection is either a line if the DLO-vector at the contact and the vector along the edge are parallel (Fig. 7b and 7d) or a point otherwise (Fig. 7a). Therefore, such *E/E*-type contacts are also annotated by *L* or *P* for line or point contact, e.g. E/E^{LS}. On the other hand, point contacts on non-convex edges are rather unlikely for elastic deformable objects. Usually, there is always a gap (*E/N*) like shown in Fig. 7d. Therefore, only the stable contact state on convex edges is further regarded (E/E^{P}).

The resulting discrete contact states are listed in Fig. 8. Generally, for most of the discrete contact states there exists more than one topological contact situation. Therefore, for some discrete contact states several typical topological contact situations are displayed. As a result, the listed topological contact situations are not complete but Fig. 6 a-d show the different basic types. The concave and convex borders of the faces are the key to mapping topological contact states to discrete contact states. Convex edges (or vertices) usually lead to instable configurations, whereas concavities lead to stable or semi-stable configurations.

Since the difference between semi-stable and stable contacts depends on the degree of the significant angles, there is a stable discrete contact state for each semi-stable discrete contact state in Fig. 8. The topological contact situations are the same down to the degree of the angles between the faces and the exact normals themselves.

V. Contact State Transition Classes

Sections II to IV mainly concerned static contact situations, but for program generation, the contact state transitions that have to be detected are of interest. Thus, we group the transitions between discrete contact states into basic classes which share some common transition characteristics. This is done in such a manner that the detection of all transitions in one class can be done with one algorithm. Therefore, the classification of the transitions enables the selection of a detection algorithm or skill for a given transition. The algorithms may be parameterized based on the more complete topological model, if necessary.

For rigid objects, translatory and rotatory motions are necessary to describe basic assembly tasks. Therefore, [2] introduces the basic skills “move-to-touch”, “rotate-to-level” and “rotate-to-insert”. For DLOs, translatory motions are sufficient to achieve all possible transitions between stable discrete contact states, we thus restrict further discussion to such motions.

A. Transitions Classes of Discrete Contact States

The first transition class contains all transitions where the DLO collides with an object in the environment, as displayed in Fig. 8. Therefore, this class can be named similarly to the “move-to-touch” skill for rigid objects, as establishing (*E*) and releasing (*R*) contacts. The common characteristic is that the newly created contact takes away one translatory degree of freedom.

The second class corresponds to the “rotate-to-level” skill, even if the transitions can be achieved with translatory motions (Fig. 9). For rigid objects, the “rotate-to-level” skill enables the change from line to point contacts. For DLOs, this class also contains all transitions from point to line contacts and vice versa. This class is called stable state transition class (*C*), since all transitions between two stable contact states as long as they are not part of the establishing and releasing contact class without intermediate contacts are transitions from point contacts to line contacts or vice versa.

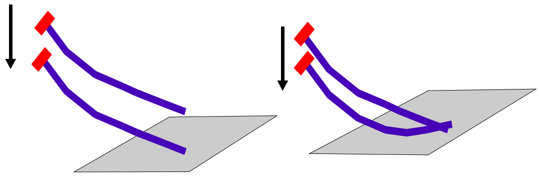


Fig. 9 A typical transition *V/N* → *V/F* which belongs to the contact establishing and releasing class (left) and a typical transition *V/F* → *E/F* of the stable state transition class (right).

The remaining classes (Fig. 6b-d) consist of sequences of two transitions between stable contact states with semi-stable or instable contact states in between. The first transition of such a sequence is also called initiated (*I*) and the second is either a transient or a spontaneous transition. Therefore, we group all such transitions occurring via semi-stable transitions into the class of transient contact state transitions (*T*). Finally, the class of spontaneous transitions contains all transitions with an intermediate instable contact state (*S*).

Table I maps each possible transition between discrete contact states without passing intermediate states to the previously defined transition classes by adopting the definitions of the transition classes. Contact state transitions like $E/F_1 \rightarrow E/F_2$ are impossible since they can not be achieved without passing intermediate states. Such impossible contact state transitions are marked with “-”.

TABLE I

Transition classes for each transition between discrete contact states .

		E								V							N
		F	E^P	E^{LS}	E^{LT}	E^{LI}	V^S	V^T	V^I	F	E^{LS}	E^T	E^I	V^S	V^T	V^I	
E	F	-	C	E	I	I	E,C	I	I	C	-	-	-	-	-	-	R
	E^P	C	-	-	-	-	E	I	I	-	-	-	I	-	-	-	R
	E^{LS}	R	-	-	-	-	C	-	-	-	C	-	-	-	-	-	R
	E^{LT}	T	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	E^{LI}	S	S	-	-	-	-	-	-	-	-	-	-	-	-	-	S
	V^S	R,C	R	C	C	C	-	-	-	-	-	-	-	-	-	I	R
	V^T	T	T	T	-	-	-	-	-	-	-	-	-	-	-	-	-
	V^I	S	S	-	-	S	-	-	-	-	-	-	-	-	-	S	S
V	F	C	-	-	-	-	-	-	-	-	E	I	I	E	I	I	R
	E^{LS}	C	-	C	-	-	-	-	-	R	-	-	-	E	I	I	R
	E^I	S	S	-	-	S	-	-	-	S	-	-	-	-	-	S	S
	E^{LT}	-	-	-	-	-	-	-	-	T	-	-	-	-	-	-	-
	V^S	C,R	-	-	-	-	-	-	-	R	C,R	-	-	-	-	-	R
	V^T	-	-	-	-	-	-	-	-	T	T	-	-	-	-	-	-
	V^I	-	-	-	-	-	-	-	-	S	-	-	S	-	-	-	S
N		E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	-

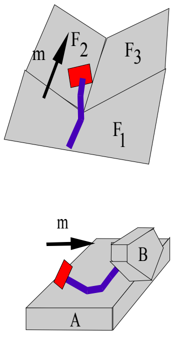
B. Transitions between Successional Discrete Contact States

Further, Table I includes only the transitions between single discrete contact states, i.e. successional contacts of length one. In general, a successional contact consists of more than one discrete contact state (Fig. 10). In order to determine the appropriate transition class for a transition between successional, we must find the most important contacts, the *principal contacts*. The *principal contacts* of a continuous contact are the contacts with the highest order *card*.

$$\begin{aligned} card(C) := \; & dim_{DLO}(C) \\ & + 2 \cdot DoF_{trans}(C) + 3 \cdot DoF_{blocked}(C) \end{aligned} \quad (3)$$

The function *card* defines a partial order on the discrete contact states and roughly estimates the stability of a discrete contact state. In particular, dim_{DLO} is zero for the vertices of the DLO and one for the edge. This coarsely reflects the lower stability of DLO-vertex contacts compared to DLO-edge contacts. The second addend DoF_{trans} is the number of translatory degrees of freedom possible without losing the contact. This means, *C* is stable against such motions. Therefore, the weight of DoF_{trans} is higher than the weight of dim_{DLO}. The last term $DoF_{blocked}$, represents the number of translatory degrees of freedom blocked by the contact. In combination with the contact force, the number of blocked translatory degrees of freedom imposes the strongest stability constraints. Thus, the weight of $DoF_{blocked}$ is the highest. As an example, $card(E/E^{LS})=1+2\cdot1+3\cdot2=9$. The values for all discrete contacts states are listed in Fig. 8.

Now, the transition class is usually determined by the changes of the principal contacts and the transition class can be directly derived from Table I. The exception is the discrete contact state transition *E/F* to *E/V^S*. This transition is ambiguous (Table I), since it either belongs to the transition class of stable state transitions (Fig. 10a) or to the establishing and releasing contact class (Fig. 10b). Here, the DLO is either dragged from a face F_1 towards a corner of a non-convex edge between F_2 and F_3 (Fig. 10a) or it is dragged along a face of an obstacle *A* towards another obstacle *B*, where the DLO collides with the surface of this obstacle (Fig. 10b). Further, both obstacles *A* and *B* intersect only in one point.



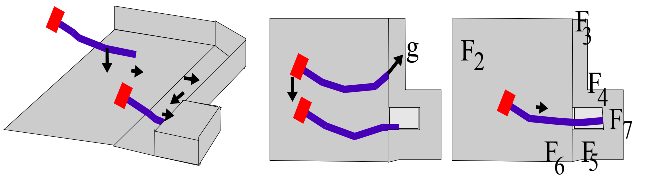
$$V_S/G \wedge E/N \wedge E/F_1 \wedge E/N \wedge V_E/N$$
$$\rightarrow V_S/G \wedge E/N \wedge E/V^S \wedge E/N \wedge V_E/N \quad \text{(a)}$$

$$V_S/G \wedge E/N \wedge E/F \wedge E/N \wedge V_E/N \rightarrow$$
$$V_S/G \wedge E/N \wedge E/E^{LI} \wedge E/V^S \wedge E/E^{LI} \quad \text{(b)}$$
$$\wedge E/N \wedge V_S/N$$

Fig. 10 Two situations of the discrete contact state transition *E/F* to *E/V^S*

In this case, the contacts with lower cardinality become important for the identification of the transition class since they resolve the ambiguity. In the example, the transition sequence of Fig. 10a does not contain any contacts with lower cardinality. Whereas, the edges of obstacle *A* (Fig. 10b) cause in-stable *E/E^LI* contacts. The complete successional contact is of course stable, since the principal contact is stable.

Recapitulatory, the transition classes can directly be derived from the sequence of discrete contact situations. Since the detection of all transitions in one transition class can be done by one algorithm, also the detection algorithm can be automatically derived from the sequence of discrete contact situations.



Step	Topological Contact States	Discrete Contact States	Tran-sition Class
1	$V_{\vec{a}}/G_{\vec{z}} \wedge E/N \wedge V/N$	$V/G \wedge E/N \wedge V/N$	-
2	$\rightarrow V_{\vec{a}}/G_{\vec{z}} \wedge E/N \wedge V_{\vec{b}}/F_1$	$V/G \wedge E/N \wedge V/F_1$	E
3	$\rightarrow V_{\vec{a}}/G_{\vec{z}} \wedge E/N \wedge E_{\vec{c}}/F_1 \wedge V_{\vec{d}}/N$	$V/G \wedge E/N \wedge E/F_1$	C
4	$\rightarrow V_{\vec{a}}/G_{\vec{z}} \wedge E/N \wedge E_{\vec{c}}/F_1 \wedge E_{\vec{c}}/(F_1 \cap F_1)$ $\wedge V_{\vec{e}}/N$	$V/G \wedge E/N \wedge E/F_1$ $\wedge E/E^p_{1,2} \wedge V/N$	-
5	$\rightarrow V_{\vec{a}}/G_{\vec{z}} \wedge E_N \wedge E_{\vec{f}}/F_1 \cap E_{\vec{f}}/F_1 \wedge V_{\vec{g}}/N$	$V/G \wedge E/N \wedge E/E^p_{1,2}$ $\wedge V/N$	C
6	$\rightarrow V_{\vec{a}}/G_{\vec{z}} \wedge E_N \wedge V_{\vec{h}}/F_2$	$V/G \wedge E/N \wedge V/F_2$	C
7	$\rightarrow V_{\vec{a}}/G_{\vec{z}} \wedge E/N \wedge V_{\vec{i}}/F_2 + V_{\vec{i}}/(F_3 \cap F_{10})$	$V/G \wedge E/N \wedge V/E^{LS}_{2,3}$	E
8	$\rightarrow V_{\vec{a}}/G_{\vec{z}} \wedge E_N \wedge V_{\vec{i}}/F_2$ $+V_{\vec{i}}/(F_3 \cap F_{10} \cap F_4)$	$V/G \wedge E/N$ $\wedge V/V^I_{2,3,4}$	I
9	$\rightarrow V_{\vec{a}}/G_{\vec{z}} \wedge E_N \wedge E_{\vec{j}}/F_2$ $+E_{\vec{j}}/(F_5 \cap F_6 \cap F_{12}) \wedge E_{\vec{k}}/F_2$ $+E_{\vec{k}}/(F_5 \cap F_{12}) \wedge$ $V_{\vec{l}}/F_2 + V_{\vec{l}}/(F_5 \cap F_{12})$	$V/G \wedge E/N \wedge$ $E/V^S_{2,5,6} \wedge$ $E/E^{LS}_{2,5} \wedge V/E^{LS}_{2,5}$	S,E
10	$\rightarrow V_{\vec{a}}/G_{\vec{z}} \wedge E_N \wedge E_{\vec{m}}/F_2$ $+E_{\vec{m}}/(F_5 \cap F_6 \cap F_{12}) \wedge E_{\vec{n}}/F_2$ $+E_{\vec{n}}/(F_5 \cap F_{12}) \wedge$ $V_{\vec{o}}/F_2 + V_{\vec{o}}/(F_5 \cap F_{12} \cap F_{13})$ $+V_{\vec{o}}/(F_7 \cap F_{14} \cap F_{15})$	$V/G \wedge E/N \wedge E/E^{LS}_{2,5}$ $\wedge E/V^S_{2,5,6} \wedge V/V^S_{2,5,7}$	E

Fig. 11 A “peg-in-hole” task and the resulting topological model as well as the discrete contact state model and the resulting transition classes. The left image in the first row shows the start and goal situation. The arrows indicate the different directions of motion of the gripper. The middle and right image give a more detailed view of the last steps. Both images show the top view, removing the upper face of the hole.

VI. Description of a Manipulation Task from Topological Model to Program Generation

In this section, we apply the presented procedure to a “Peg-in-Hole” task, as shown in Fig. 11. We assume, the task is demonstrated using the sequence of topological contact situations, as displayed in Fig. 11. The numbers indicate the order of the contact situations.

The environment is used as much as possible in order to guide the DLO to the goal situation. This is necessary because the DLO vectors are greatly affected by the intrinsic uncertainties of DLOs and are extracted from a difficult physical simulation. They are not suitable for a reliable manipulation because they must be known quite precisely.

Therefore, the first contact was made with the largest face. The transition is part of the establishing contact class since the DLO collides with one face. The next transition is a stable state transition, because a *V/F* contact changes into an *E/F* contact (Table I). The sequence $V/F \rightarrow E/F \rightarrow E/E^P \rightarrow E/F \rightarrow V/F$ for example, is more reliable than the sequence $V/F \rightarrow V/E^I \rightarrow V/F$. This is because the the first sequence contains only stable state transitions, whereas the outcome of the spontaneous transition following the instable V/E^I contact depends much on the DLO vector of the vertex [8].

Step 4 of this sequence has no contact transition class. Since the *E/F* contact is maintained, the principal contact of the continuous contact $E/F \wedge E/E^P$ does not differ from the principal contact of step 3. Therefore, no contact state transition class is extracted.

The actual insertion is first guided by a concave edge. Thus, in step 7 a V/E^{LS} is established. The insertion occurs based on a spontaneous transition in step 9. This spontaneous transition results in a new stable contact with *N* as the intermediate state. Therefore, the algorithms for detecting spontaneous contacts and for detecting new contacts should function (*S* and *E*).

This spontaneous transition is unavoidable except by using a rotatory motion like “rotate-to-insert” that is not regarded here. However, based on the geometric situation, the DLO will always hit the opposing edge F_5 (Fig. 11, top view) as long as the DLO vector *g* of the vertex points away from the hole, as displayed in Fig. 11. Therefore, the initial pose of the DLO at step 1 must be known well enough to ensure an appropriate DLO vector of the vertex just before the DLO insertion, otherwise the insertion task will fail.

Another way to cope with the often unreliable outcome of spontaneous transitions and with the uncertainties due to the shape of DLOs (DLO vectors) is to expand the sequence of contact state transitions to a network of contact state transitions. This network may include all possible consecutive contact situations and the likelihood of the transition. In this case, the program executing the task is not a simple sequence of macros realizing the detection of the contact state transitions. Instead, a probabilistic control mechanism is needed due to the stochastic behaviour of the contact state transitions. With such a graph, the shape of the DLO can be sensed step by step based on the established contacts, since each contact imposes some constraints on the DLO.

VII. Conclusions and Further Research

We presented a systematic process for automatically transferring a manipulation task that has been demonstrated in virtual reality into a simple sequence of detection macros. We first presented a model based on topological contact states to represent such manipulation tasks for DLOs and obstacles in a polyhedral environment. In order to enable the automatic selection of state transition detection algorithms, this topological contact state model is transferred into a more abstract model of discrete contact states. The selection of detection algorithms is achieved by the classification of the transitions between contact states. Finally, we applied this process to a “peg-in-hole” task as an example of use. The execution portion of the process is not covered by this work, but is discussed in [1] and especially [9].

In summation, the approach for manipulation of DLOs as presented in [3] and [8] was extended and applied to the contact state extraction and program generation portion of the systematic process for handling DLOs presented in [6].

Future work will first concentrate on the execution aspect of the system, i.e. the detection algorithms. Further, the simple sequences of contact state transitions must be extended to networks of contact state transition in order to better cope with intrinsic uncertainties such as shape changes of the DLOs (Section VI).

Additionally an extension of this approach from polyhedral environments to include environments with arbitrary curved surfaces is another topic. The presented approach is directly applicable to environments with round edges or slightly curved surfaces. Since a slightly curved surface means that the normal vector for all points of the surface is almost constant within the workspace of a robot, this situation scarcely differs from the polyhedral case, with constant normal vectors for each face. Round edges and vertices also differ little from the polyhedral case, so only environments with highly curved objects, e.g. hemispheres, require further investigation and an extended model.

References

[1] J. Acker and D. Henrich:"Manipulating deformable linear objects: Vision Based Detection of Contact State Transitions". In: *Proc. IEEE International Symposium on Assembly and Task Planning*, Besancon, France, 2003.

[2] Hasegawa T., Suehiro T., and Takase K.: "A model-based manipulation system with skill-based execution". In: *IEEE Transactions on Robotics and Automation*, vol. 8, No. 5, pp. 535-544, October 1992

[3] Henrich D., Ogasawara T., Wörn H.:"Manipulating deformable linear objects: Contact states and point contacts". In: *Proc. IEEE International Symposium on Assembly and Task Planning*, Porto, Portugal, 1999.

[4] K. Hirana, T Suzuki, S. Okuma, K. Itabashi, F. Fujiwara: "Realization of Skill Controller for Manipulation of Deformable Objects Based on Hybrid Automata". In: *Proc. 2001 Int. Conf. on Robotics and Automation*, vol. 3, pp. 2674-2679, Seoul, Korea, May 2001

[5] X. Ji and J. Xiao:"Automatic Generation of High-level Contact State Space". In: *Proc. IEEE Conference on Robotics and Automation*, Detroit, Michigan May 1999.

[6] B. Kahl, D. Henrich: "Virtual Robot Programming for Deformable Linear Objects: System Concept and Prototype Implementation". In: *12th International Symposium on Measurement and Control in Robotics* (ISMCR), Bourges, France, June 20 - 21, 2002.

[7] Nakagaki H.,et al.: "Study of deformation and insertion tasks of a flexible wire". In: *Proc. 1997 Int. Conf. on Robotics and Automation*, vol. 3, pp. 2397-2402, Albuquerque, USA, April 1997.

[8] Remde A., Henrich D., and Wörn H.: "Manipulating deformable linear objects: Contact state transitions and transition conditions". In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Kyongju, Korea, October 1999.

[9] Schlechter A. and Henrich D.: “Manipulating Deformable Linear Objects: Characteristics in Force Signals for Detecting Contact State Transitions” In: *Proc. of 10th Int. Conf. on Advanced Robotics* Budapest, 22-25. August 2001.

[10] H. Wakamatsu, A. Tsumaya, K. Shirase and E. Arai: "Representation of Deformable Thin Object Manipulation Based on Contact State". In: *Proc. Of the 5th Franco-Japanese Congress & 3rd European-Asian Congress* , Besancon, France, October 2001