

# Manipulation of deformable linear objects: Force-based simulation approach for haptic feedback

Björn Kahl and Dominik Henrich  
*Lehrstuhl für Angewandte Informatik III  
 (Robotik und Eingebettet Systeme)  
 Universität Bayreuth, D-95445 Bayreuth, Germany  
 [Bjoern.Kahl | Dominik.Henrich] @ uni-bayreuth.de*

**Abstract** - In this paper we present a new approach to simulate deformable linear objects (DLOs) for use in a haptic feedback system. Our goal is to optimize efficient computation and preciseness. The presented approach is inspired from the well known method of “finite elements”, but avoids a exact physical model.

**Index Terms** - Keywords: Modeling, deformable linear objects (DLOs), force-based simulation

## I. INTRODUCTION

Deformable Linear Objects (DLOs) such as ropes, wires or steel springs are used in a wide range of products, however inherent uncertainties such as bending and compliance make it difficult to handle these objects in a generalized and automated manner. One approach to overcome these difficulties is to use an abstract task model based on contact states [1] to describe a given assembly task. This allows one to avoid the use of exact coordinates and movement instructions in the assembly process as well as in the programming process. Our goal is to automatically generate a sequence of contact states by demonstrating the desired assembly task in virtual reality [11]. To achieve a simple and intuitive operation, we use a haptic feedback system to allow people to conduct a manipulation task using their sense of touch. To do so, we need a fast and reliable simulation of the static (and perhaps dynamic) behaviour of a DLO including contact forces.

Several researchers have published on the topic of DLO simulation. In the following overview, we give a short summary of prior work: A two-dimensional energy-based model is presented in [3], and an enhancement for a three-dimensional model can be found in [9]. Both approaches utilize the fact that the potential energy of the DLO is minimized in its stable configuration and [10] include consideration of dynamic energy. While all these approaches are not real-time enabled, [5] presents more efficient approximation methods. All of the above researchers use a linear combination of basic functions to approximate bending or the tangent of the DLO. Unfortunately, it is unclear how to extract forces from these energy-based simulation approaches, which are necessary for haptic feedback.

In [4] a mass-spring system (based on inner energy of the springs) is used to describe the DLO. An implicit Ordinary Differential Equation (ODE) solver is used to derive a linear system of equations, which in-turn is solved by an iterative solver. In contrast, our approach avoids both, calculating energy and runtime solving of differential equations.

Stable haptic feedback needs a real-time simulation

capable of providing approximately 1000 shape and force updates each second. We aim at running the whole system, that is the DLO simulation as well as the other parts of the programming system, on usual desktop-workstations. Fortunately, the simulation only requires a loose approximation of the DLOs geometry and not the exact shape.

Section II describes our approach in detail and introduces heuristics to determine values for the models parameters. Section III describes our experimental setup and Section IV is a discussion of our results. In Section V, we close with a brief description on how our model is used in our ViRoP (Virtual-Robot-Programming-) system.

## II. MODEL DESCRIPTION

We model our DLO as a fixed set  $S$  of  $P_i$  points (with  $i$  ranging from 0 to  $n$ ) in the 3D space, connected by cylinder segments. From the simulation functions point of view, these cylinders do **not** exist and the whole computation is

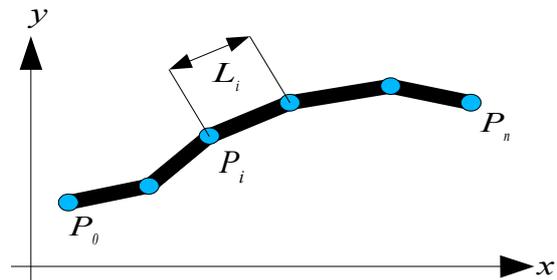


Fig. 1: A DLO is represented as a set of points and cylinder segments between these points. The cylinders can change their length, but can not be bended. For the simulation algorithm only the points  $P_i$  exist.

based on the position of these points  $P_i$  and their distance to the next ( $L_i$ ) and previous ( $L_{i-1}$ ) point (see Fig. 1).

To calculate the shape of our DLO, we compute the total force acting at each point  $P_i$  and update its position based on the computed force.

### A. Model

The shape of our DLO is determined by three force components:

- the inner force of bending
- the inner force of contraction
- environmental forces such as contact forces with obstacles or the gripper as well as gravitation force.

In the most basic form of our model, the sum of these three forces for each point  $P_i$  is translated (by applying an experimentally determined linear scaling factor  $c_F$ ) to a single displacement which is simply added to the current position of that point. Several variations to the model addressing stability and responsiveness issues of these scheme are discussed later.

### Notation used

All symbol names are set *cursive*, with an index  $i$  denoting the symbol depends on, or is somehow related to a distinct point of the DLO. The index  $j$  is used to enumerate some external forces (see later). All other indices used, do not represent an enumeration, but distinct different types of forces and constants. The symbol  $\langle \vec{x} | \vec{y} \rangle$  is used for the scalar product of two vectors  $\vec{x}$  and  $\vec{y}$ . Vectors in the text are set in **bold**, vectors in numbered equations carry an arrow on top.

### Inner forces: Contraction

The contraction force is computed by comparing the current distance  $L_i$  between point  $P_i$  and  $P_{i+1}$  with the expected distance  $L_s$  and projecting the resulting difference on the direction from point  $P_i$  to  $P_{i+1}$ .

$$\begin{aligned} \Delta L_i &= \frac{\|P_{i+1}^{\vec{}} - P_i^{\vec{}}\| - L_s}{L_s} \\ \vec{E}_i &= \frac{P_{i+1}^{\vec{}} - P_i^{\vec{}}}{\|P_{i+1}^{\vec{}} - P_i^{\vec{}}\|} \\ \vec{F}_i &= c_c \Delta L_i \vec{E}_i \end{aligned} \quad (1)$$

The parameter  $c_c$  controls the stiffness of the DLO with respect to changes in the length. See below for a heuristic to compute a value for  $c_c$ .

Of course the opposite force applies to the following point  $P_{i+1}$ . The resulting contraction force  $F_{c,i}$  is then

$$\begin{aligned} F_{c,0}^{\vec{}} &= \vec{F}_i \\ F_{c,i}^{\vec{}} &= \vec{F}_i - F_{i-1}^{\vec{}} \quad \forall i \in \{1, \dots, n-1\} \\ F_{c,n}^{\vec{}} &= -F_{n-1}^{\vec{}} \end{aligned} \quad (2)$$

### Inner forces: Bending

The bending force is derived from the angle  $\alpha$  between two connected segments of the DLO. The basic idea is to model each two connected segments as a triangle with a spring as the hypothesis pushing the end points to the full expanded (that is co-linear) position. Of course the length of the two connected segments should stay unchanged. To achieve this, only the force component orthogonal to the segments is used for the end points. The spring force  $F_{b0,i}$  for point  $P_i$  (see Fig. 2 for names and positions) is computed as follows:

$$\begin{aligned} d_i &= 1 + \frac{\langle P_{i-1}^{\vec{}} - P_i^{\vec{}} | P_{i+1}^{\vec{}} - P_i^{\vec{}} \rangle}{\|P_{i-1}^{\vec{}} - P_i^{\vec{}}\| \|P_{i+1}^{\vec{}} - P_i^{\vec{}}\|} \\ \vec{E}_{b,i} &= \frac{P_{i+1}^{\vec{}} - P_{i-1}^{\vec{}}}{\|P_{i+1}^{\vec{}} - P_{i-1}^{\vec{}}\|} \\ F_{b0,i}^{\vec{}} &= c_b d_i \vec{E}_{b,i} \end{aligned} \quad (3)$$

Here  $c_b$  is an experimentally determined scaling factor which controls the stiffness of the DLO with respect to bending. See below for a heuristic how to compute useful values for  $c_b$ .

The angle  $\alpha$  does not appear in Formula 3, instead of  $\alpha$  we use the scalar product between the two connected segments. From the well know relation  $\cos(\beta) = \langle \vec{v}_1 | \vec{v}_2 \rangle$  with  $\beta \nrightarrow \vec{v}_1, \vec{v}_2$  and the fact that  $\alpha = \pi - \beta$  and  $\cos(\pi - x) = -\cos(x)$ , it follows that  $-\cos(\alpha) = \langle P_{i-1}^{\vec{}} - P_i^{\vec{}} | P_{i+1}^{\vec{}} - P_i^{\vec{}} \rangle / (\|P_{i-1}^{\vec{}} - P_i^{\vec{}}\| \|P_{i+1}^{\vec{}} - P_i^{\vec{}}\|)$ .

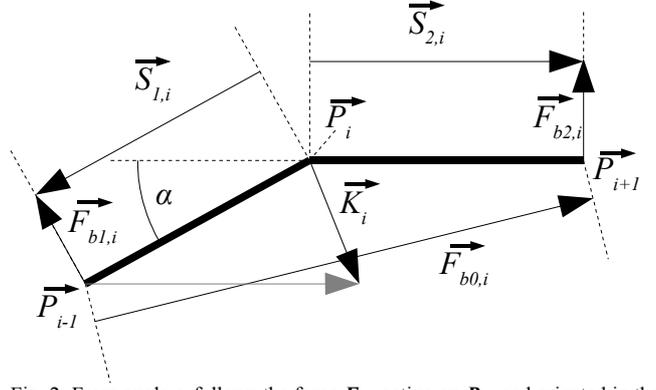


Fig. 2: From angle  $\alpha$  follows the force  $F_{b0,i}$  acting on  $P_{i+1}$  and oriented in the direction from  $P_{i-1}$  to  $P_{i+1}$ . Projecting  $F_{b0,i}$  on the two connected segments results in  $S_{1,i}$  and  $S_{2,i}$ . The sum of  $S_{1,i}$  and  $S_{2,i}$  is  $K_i$ , the movement of the point  $P_i$ .  $F_{b0,i}$ 's components orthogonal to the segments gives  $F_{b1,i}$  and  $F_{b2,i}$ , the movement of point  $P_{i-1}$  and  $P_{i+1}$ . - (Note: All vectors in this figure are unscaled.)

Further using the approximation  $1 - \cos(\alpha) \approx \alpha/10$  for  $|\alpha| \ll 1$  we can justify Formula 3. (Typical values for  $\alpha$  range from  $-0.15$  to  $0.15$ )

We compute the effective inner forces  $F_{b1,i}$ ,  $F_{b2,i}$  and  $K_i$  from  $F_{b0,i}$  following the observation, that in a real triangle (oriented as in Fig. 2) with a spring pushing its end points apart the middle point would move down-right and the endpoints would move up because of inertia (so the centre of mass keeps its position):

$$\begin{aligned} \vec{E}_{1,i} &= \frac{P_{i-1}^{\vec{}} - P_i^{\vec{}}}{\|P_{i-1}^{\vec{}} - P_i^{\vec{}}\|}, \quad \vec{S}_{1,i} = \langle F_{b0,i}^{\vec{}} | \vec{E}_{1,i} \rangle \vec{E}_{1,i} \\ \vec{E}_{2,i} &= \frac{P_{i+1}^{\vec{}} - P_i^{\vec{}}}{\|P_{i+1}^{\vec{}} - P_i^{\vec{}}\|}, \quad \vec{S}_{2,i} = \langle F_{b0,i}^{\vec{}} | \vec{E}_{2,i} \rangle \vec{E}_{2,i} \end{aligned} \quad (4)$$

$$\begin{aligned} \vec{K}_i &= \vec{S}_{1,i} + \vec{S}_{2,i} \\ F_{b1,i-1}^{\vec{}} &= F_{b0,i}^{\vec{}} + \vec{S}_{1,i} \\ F_{b2,i+1}^{\vec{}} &= F_{b0,i}^{\vec{}} - \vec{S}_{2,i} \end{aligned}$$

Formula 4 can be evaluated for all points  $P_i$  except points  $P_0$  and  $P_n$ . For these two points only  $F_{b1,0}$  or  $F_{b2,n}$  (computed from  $F_{b0,1}$  and  $F_{b0,n-1}$  respectively) apply.

### Environmental forces

External or environmental forces are added to each point they apply to. Our model supports three different kinds of external forces  $E_{F,j}$  to represent different types of constrains:

1. Global, position-independent forces that are not bound to a specific point of the DLO. These are used to model gravitation and have the same size and direction  $c_{e,j}$  for all points  $P_i$ .
2. Local, directed forces bound to a given point  $P_i$  of the DLO and depending in size on the distance of  $P_i$  to a plane given by a point  $c_{P,j}$  and a normal vector  $N$ . These are used to represent contact forces with obstacles.
3. Local, position-dependent forces bound to a given point  $P_i$  of the DLO and depending in size on the distance of  $P_i$  to another fixed point  $c_{P,j}$ . These are used to model the forces originating from a gripper.

Each point  $P_i$  can be subject to zero, one, or more external forces out of the index set  $S_E$  of all external forces  $E_{F,j}$ .

$$\vec{E}_{F,j}(\vec{P}_i) = \begin{cases} c_{e,j} & \text{constraint 1} \\ c_{e,j} \langle \vec{P}_i - c_{p,j} | \vec{N} \rangle \vec{N} & \text{constraint 2} \\ c_{e,j} (\vec{P}_i - c_{p,j}) & \text{constraint 3} \end{cases} \quad (5)$$

All these forces, internal and external ones, are added together to form the final force  $\vec{F}_{i,t}$  to be applied to that point:

$$\vec{F}_{i,t} = \vec{F}_{c,i} + \vec{F}_{b1,i} + \vec{F}_{b2,i} + \vec{K}_i + \sum_{j \in S_{E,i}} \vec{E}_{F,j}(\vec{P}_i) \quad (6)$$

Here  $S_{E,i}$  denotes the subset of external forces acting on point  $\vec{P}_i$  out of all defined external forces  $S_E$ . The new values for the points  $\vec{P}_i$  at time step “ $t+1$ ” are computed by adding a scaled version of  $\vec{F}_{i,t}$  to the last position of  $\vec{P}_i$  at time step “ $t$ ”:

$$\vec{P}_i|_{t+1} = \vec{P}_i|_t + c_F \vec{F}_{i,t} \quad (7)$$

The scaling factor  $c_F$  is experimentally determined and restricts the movement of a point to prevent oscillation.

Until here is no real difference to any (force-based) conventional FEM approach to simulate DLOs, at least when ignoring the fact, that our bending model is not physically correct. However it delivers reasonable good values for small bending angles.

In a conventional FEM system we would have to formulate the differential equations governing the motion of all points  $\vec{P}_i$  in our system:

$$\frac{d^2 \vec{P}_{i,k}}{dt^2} = \frac{1}{m} F_{i,k}(\vec{P}_i) \quad (8)$$

$$\forall i \in [0, 1, \dots, n], \quad k \in [x, y, z]$$

This is a heavily coupled system of  $3N$  (one for each point  $\vec{P}_i$  and space-dimension  $k$ ) 2<sup>nd</sup> order differential equations. As it is impossible to solve this ODE system analytical, we would have to solve it numerical in each discrete time step. This is computational expensive and time consuming. As the goal of our research work in the area of deformable object simulation is to come up with an approach fast enough to drive a haptic feedback device, runtime solving of the ODE system (and in consequence using conventional FEM systems), this is not an option.

Instead we collapse all the complexity in the one single scaling factor  $c_F$  in Formula 7. In the following Subsection B we describe how to compute  $c_F$  (and other parameters of our model).

All computations needed to determine the points  $\vec{P}_i$  at time  $t+1$  from the points  $\vec{P}_i$  and external forces  $\vec{E}_{F,j}$  at time  $t$  are summarised in the term “simulation step” from here on.

### B. Parameter optimization

Three constant parameters control the stability and responsiveness of the model:

- $c_c$  controls the stiffness for contraction
- $c_b$  controls the stiffness for bending
- $c_F$  controls the responsiveness and stability

Four additional constant parameters further influence the overall behaviour of to model:

- $c_{e,j}$  controls the stiffness of environmental object  $j$  (e.g. obstacles or the gripper)
- $l$  is the total length of the DLO
- $m$  is the total mass of the DLO

- $N$  is the number of points in the DLO

In the following we give some heuristics on how to determine values for these constants. We assume the simulated DLO is gripped somewhere near an end-point and the total length  $l$  as well as the total mass  $m$  are known from the application.

First, the number of points  $N$  has to be determined.  $N$  controls the computational load, the time (in terms of simulation steps) needed to propagate a change (e.g. a newly established collision with an obstacle) from one end of the DLO to the other, and the smoothness of the DLOs shape. Further it has influence on the stiffness controlled by  $c_c$  and  $c_b$ .

The number of simulation steps needed to propagate a change from one end of the DLO to the other is approximately 2 to 3 times the number of points in the DLO, depending on the current state of the DLO (average amount of inner forces on all points) and the amount of change.

In conjunction with  $c_b$ ,  $N$  influences the amount of force  $\vec{F}_{b1,i}$  needed to bend the DLO to a 90° quarter circle. Assuming the 90° are equally distributed over all points  $\vec{P}_i$ ,  $i=1, \dots, N-2$ , we find by combining Formula 3 for  $\vec{F}_{b0,i}$  with

$$-\cos(\alpha) = \langle \vec{P}_{i-1} - \vec{P}_i | \vec{P}_{i+1} - \vec{P}_i \rangle / (\| \vec{P}_{i-1} - \vec{P}_i \| \| \vec{P}_{i+1} - \vec{P}_i \|) : \quad \text{force} \propto \left( 1 - \cos\left(\frac{\pi/2}{N-1}\right) \right) \sqrt{\frac{1}{2} \left( 1 - \cos\left(\frac{\pi/2}{N-1}\right) \right)} \quad (9)$$

For values greater than four  $N$  (see Fig. 3) loses almost all influence on the force, thus  $N$  should be greater than four.

Another constrain is the diameter of the smallest possible 90° quarter circle for a given bending force  $\vec{F}_{b1,i}$ . For any given bending stiffness parameter  $c_b$  and bending force  $\vec{F}_{b1,i}$  follows a corresponding angle  $\alpha_i$  between to segments. (The relation between  $\vec{F}_{b1}$  and  $\alpha_i$  is discussed below.) From  $\alpha_i$  and the known total length of the DLO  $l$  and the number of points  $N$  follows the minimum diameter  $R$  and the number of segments  $n_b$  needed to build a 90° quarter circle:

$$R = \frac{1}{2} \frac{l}{N-1} \frac{1}{\sin(\alpha_i/2)} ; \quad n_b = \frac{\pi/2}{\alpha_i} \quad (10)$$

In summary,  $N$  should be at least greater than four and it should be large enough to allow all desired deformation without exceeding the force limit  $\vec{F}_{b1}$  per segment-angle  $\alpha_i$ .

Next step is to determine  $c_c$  and  $c_b$ . The value of  $c_c$

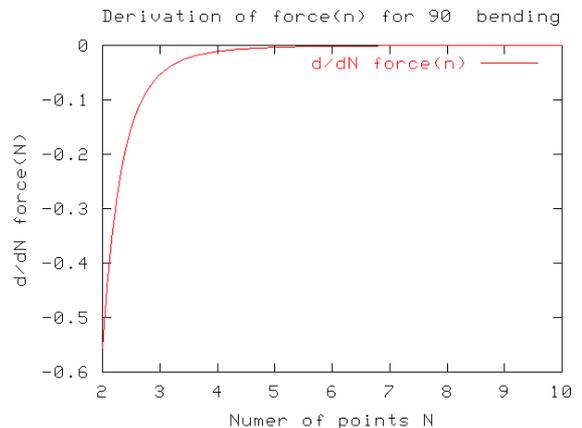


Fig. 3: Influence of number of points  $N$  on bending force.

basically controls the change in length of the DLO (in percent of total length) under a given force. Given a maximum change in length  $d_l$  by  $\Delta l$  percent (i.e.  $d_l = l \cdot \Delta l / 100$ ) and a corresponding force  $F_{\Delta l}$ , then

$$c_c = F_{\Delta l} \frac{100}{\Delta l} \quad (11)$$

A good candidate for  $F_{\Delta l}$  is the gravitational force, i.e.

$$c_c = m g \frac{100}{\Delta l} \quad (12)$$

Good values for  $\Delta l$  are 3 to 5.

For  $c_b$  we follow the same idea and relate the force  $F_{bl}$  to the bending  $\alpha_l$  created by the mass of the DLO at each DLO point. Starting from Formula 3 we find

$$c_b = \frac{F_{bl} \sin(\pi - \alpha_l)}{(1 - \cos \alpha_l) \sin(\alpha_l / 2)} \quad (13)$$

with  $F_{bl} = m g$  and  $\alpha_l$  approximately 3 to 5 degree.

The relation between  $N$ ,  $c_c$  and  $c_b$  with the elastic (or Young's) modulus  $E$  is discussed in Section III.

For the external forces representing the “glue” between gripper and gripped point, the same idea can be exploited. Given the maximum distance  $G_d$  between the gripper and its gripped point in case of a free hanging DLO (i.e. a vertical straight line) we find for  $c_{e,j}$  in constraint 3 of Formula 5:

$$c_{e,j} = \frac{m g}{G_d} \quad (14)$$

Good values for  $G_d$  are 1 to 2 percent of a DLOs segment length as in  $G_d = \frac{l}{N-1} \frac{p}{100}$  with  $p$  1% to 2% and  $N$  the number of points in the DLO. An example on how this is used to connect our model to a haptic feedback device is shown in Section V.

To prevent oscillation, no point of the DLO should ever move beyond its equality position where the force  $F_{ti}$  (as in Formula 6 and 7) acting on it becomes zero. To achieve this, the scaling factor  $c_F$  in Formula 7 should limit the resulting displacement to approximately 1/50 to 1/1000 of the length of one segment in the DLO. It is difficult to give a more precise value, as  $c_F$  obviously depends heavily on the values of the other parameters as well as the average speed with which the DLO is moved in the simulation.

In general,  $c_F$  should make sure a point  $P_i$  will not move more than 50% of the distance to that position where  $F_{ti}$  would be zero in the current simulation step. Unfortunately we were unable to analytical solve

$$0 \stackrel{!}{=} \vec{F}_{t,i}(\vec{P}_{i-1}, \vec{P}_i + \Delta t \vec{d}_i, \vec{P}_{i+1}) \quad (15)$$

with  $\vec{d}_i = \vec{F}_{t,i}(\vec{P}_{i-1}, \vec{P}_i, \vec{P}_{i+1})$

for  $t$  in order to find a better supported value for  $c_F$  than experimentally found by try-and-error.

As  $c_F$  establishes an upper limit (in the range of 1/50 to 1/1000 of the length of one segment) on the movement of any point of the DLO per simulation step, it effectively limits the speed of the DLO as a whole to this fraction of one segments length multiplied by the number of simulation steps per second.

For haptic applications at least 1000 force-feedback values per second are needed. Assuming that the gripper position updates at the same rate of 1000 Hz, and

considering a propagation speed for any change to travel through the DLO of 2 or 3 simulation steps per point of the DLO, a good guess at the number of required simulation steps per second is 20000 to 30000 (assuming  $N = 10$ ).

### C. Improving stability and responsiveness

The two main issues of the model as described so far are its high probability to start oscillating (if the displacements  $c_F \vec{F}_{t,i}$  become to big) and its poor responsiveness. With the term “responsiveness” we describe the ability of the simulated DLO to closely follow a gripper trajectory with both, high acceleration and high linear (unaccelerated) speed.

Careful analysis of the oscillation problem showed that the main source for instability is a point moving below the connecting line of its two neighbours (bending), or a point moving backwards between two points because of strong compression of one DLO segment (Fig. 4). Unfortunately, increasing stability decreases responsiveness as improved responsiveness basically means to move faster, and that in turn means to make wider steps in each iteration of the simulation.

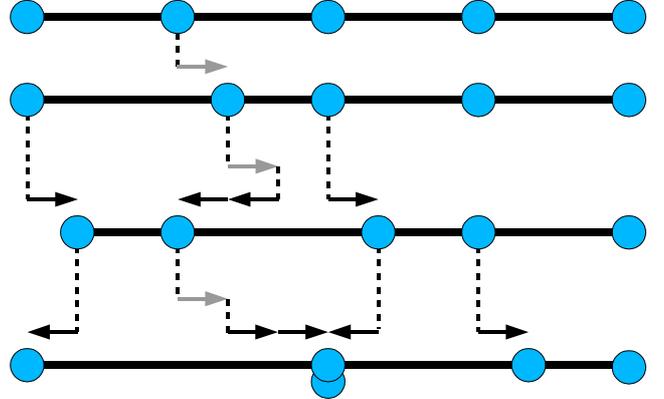


Fig. 4: Strong compression of one segment (second segment in the second line) can force a point to move backwards. The first (grey) vector on each second-left point denotes some external force driving the whole DLO to the right. All other forces are internal contraction forces driving the individual points. Note that in this example, the second and third point cross each other in the last line. (Vectors are unscaled.)

One approach to increase responsiveness without loosing stability is, to split the process of limiting the points movement in two parts. The scaling with  $c_F$  as described above is followed by a second stage where the resulting maximum movement  $d_m$  for all movements  $\Delta \vec{P}_i$ :

$$d_m = \max_{i=0}^{N-1} \|\Delta \vec{P}_i\| \quad \text{with} \quad \Delta \vec{P}_i = c_F \vec{F}_{t,i} \quad (16)$$

is compared to a maximum step width  $l_s$  and scaled down uniformly in case of exceeding this limiting value. By detecting oscillation and dynamically lowering  $l_s$  (or the resulting scaling factor  $c_m$ , Formula 17) by adding another scaling factor  $c_s$ , we can select  $l_s$  and  $c_F$  close to their values where oscillation starts for sure. This scaling factor  $c_s$  remains at value “1” until some oscillation is detected. For each point found to be oscillating in two consecutive simulation steps, the value of  $c_s$  increases and for each simulation step found to be free of any oscillating points  $c_s$  decreases until the level of “1” is reached again. Substituting  $c_s$  and Formula 16 in 7 we arrive at

$$c_m = \begin{cases} \frac{1}{c_s} \frac{l_s}{d_m} & : d_m > l_s \\ \frac{1}{c_s} & : \text{else} \end{cases} \quad (17)$$

$$\vec{P}_i|_t = \vec{P}_i|_{t-1} + c_m \Delta \vec{P}_i|_t$$

This allows a much higher value for  $c_F$ , somewhere in the range of limiting  $c_F \vec{F}_{t,i}$  to 1/10 to 1/100 of the length of the DLO segments, while steps small enough to prevent points moving backwards are possible.

The major disadvantage of such a dynamically computed down-scaling is the loss of the fixed relation between a value for  $F_{t,i}$  and resulting displacement per second. This disadvantage can be compensated by maintaining the sum of all  $c_m$  since the start of the simulation, and repeating the simulation step as required:

---

```
Pseudo code:
expected_sum_of_cm += steps_per_time
repeat
  do_simulation_step()
  sum_of_cm += get_last_cm()
until sum_of_cm ≥ expected_sum_of_cm
```

---

Text 1: Keeping relation between  $F_{t,i}$  and displacement per second.

To further improve responsiveness it is sufficient to limit the relative movement of a point with respect to its direct neighbours instead of limiting the overall movement. This can be exploited in two ways to improve responsiveness while retaining a good stability: Either on a per point basis (used here) or by subtracting the overall movement of the whole DLO prior to limiting the distance any point may move in one simulation step.

In both scenarios Formula 17 has to be extended to first compute relative point displacements  $\Delta_r \mathbf{P}_i$  (using a value for  $c_F$ , limiting  $\Delta \mathbf{P}_i$  (as in Formula 16) to something in the range of 1/10 to 1/100 of the length of the DLO segments), then compute the maximum relative displacement  $d_m$ . If the maximum displacement  $d_m$  is found to exceed the limiting value  $l_s$ , then scale all recorded displacements by this factor  $c_m$ . Finally apply the scaled displacements to the points of the DLO: (The oscillation damping by  $c_s$  is used as before.)

$$\begin{aligned} \Delta_r \vec{P}_i &= \Delta \vec{P}_i - \Delta \vec{P}_{i+1} \\ d_m &= \max_{i=0}^{N-2} \|\Delta_r \vec{P}_i\| \\ c_m &= \begin{cases} \frac{1}{c_s} \frac{l_s}{d_m} & : d_m > l_s \\ \frac{1}{c_s} & : \text{else} \end{cases} \quad (18) \\ \vec{P}_i|_t &= \vec{P}_i|_{t-1} + c_m \Delta \vec{P}_i|_t \end{aligned}$$

The effect of using Formula 18 instead of the original method in Formula 6 and 7 is discussed in Section III.

While Formula 18 allows different points to move at a wider range of different speed than in Formula 17, the maximum speed any point can move at is still very small and limited to  $c_F$  times  $F_{t,i}$  per simulation step. Especially no point can accelerate beyond this limit, effectively suppressing any sort of dynamic behaviour.

Because the stability of the simulation is only affected

by the relative movements of neighbouring points, there is no need to limit the absolute speed of any points. This allows us to extend Formula 18 by an additional term representing some sort of dynamic. For each point we record the movement made in the last simulation step, and add a fraction of it to the newly computed step:

$$\begin{aligned} \Delta \vec{P}_i|_t &= c_h \Delta \vec{P}_i|_{t-1} + c_F \vec{F}_{t,i} \\ \Delta_r \vec{P}_i &= \Delta \vec{P}_i - \Delta \vec{P}_{i+1} \end{aligned} \quad (19)$$

depending on  $c_h$ , this allows for point displacements  $\Delta_r \mathbf{P}_i$  several magnitudes greater than  $c_F \vec{F}_{t,i}$ . Substituting this new  $\Delta_r \mathbf{P}_i$  in Formula 18 we arrive at the final model:

$$\begin{aligned} d_m &= \max_{i=0}^{n-1} \|\Delta_r \vec{P}_i\| \\ c_m &= \begin{cases} \frac{1}{c_s} \frac{l_s}{d_m} & : d_m > l_s \\ \frac{1}{c_s} & : \text{else} \end{cases} \quad (20) \end{aligned}$$

$$\vec{P}_i|_t = \vec{P}_i|_{t-1} + c_m \Delta \vec{P}_i|_t$$

An alternative approach would be to maintain a current speed value  $v_i$  for each  $\mathbf{P}_i$  and to first update the  $v_i$  based on  $c_F$  and  $F_{t,i}$  (including an oscillation damping similar to the one discussed above) and then, in a second step, add a fraction of  $v_i$  to  $\mathbf{P}_i$  in each simulation step. This would be much closer to a real physical simulation (by including real dynamic and two-step “integration” of the ODE from Formula 8, at the expense of more computation needed and a harder to maintain stability.

### III. EXPERIMENTS

To determine real-world values for the model parameters and to verify the expected behaviour of some of the aspects of our simulation, we compared the final model with dynamic and relative limits (as per Formula 20) against the model without dynamics (as per Formula 18) using three test cases. The test cases are:

1. Rotation on one near-end point in the y-z-plane (i.e. Rotational axis is horizontal oriented)
2. Horizontal movement with horizontally oriented DLO
3. Horizontal movement with vertically oriented DLO

Additionally, for the first test case we compared the original model as per Formula 7 against the other two. All these experiments are conducted with the set of parameters displayed in Table 1.

The first series of experiments is to compare the original model (as per Formula 7) to the version with relative limits, but without dynamics. Fig. 5 shows the start position (hanging down), the situation after half of the movement and the end-position for the original model superimposed in one image. Fig. 6 shows same situations for the relative limits version.

Parameter	Value	Remarks
$l$	15 cm	Total Length
$m$	0.01 kg	Total Mass
$N$	10	Number of Points
$c_c$	1.56960	0.981 N at 5% segment length
$c_b$	3776.21457	0.981 N at 2.5 ° bending

Parameter	Value	Remarks
$c_e$ (gripper)	0.981 N	At 5% segment length
$c_e$ (grav)	-0.00892 N	Gravitation force at each point
$c_F$	0.00413	Scaling factor
$l_s$	0.00900	Max. allowed step size

Table 1: Basic geometric parameters used to compute other model parameters and the computed values of all internal scaling parameters used in our experiments.

In this test case there is no difference in using the relative limits variation. This is mostly because the overall movements are too small to exceed the set limits.

Adding dynamics changes the behaviour considerably. The DLO follows the gripper rotation much closer and reaches nearly its final position after half the time than without dynamics (see Fig. 5).

In compliance with this, the average forces as well as the oscillation damping needed by  $c_s$ , are much lower than in the situation without dynamics. Table 2 summarizes the number of simulation steps needed to complete this simulation task of 28 seconds simulated time as well as the number of times  $c_s$  was not "1" or  $d_m$  was greater than  $l_s$ .

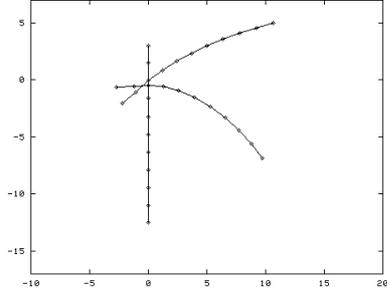


Fig. 5: Rotation around third point from left, unenhanced model.

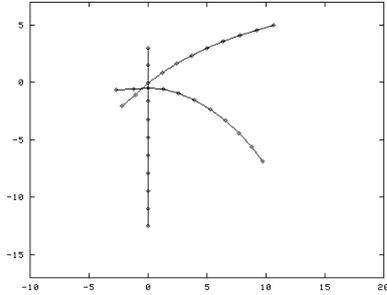


Fig. 6: Rotation around third point from left, with relative limits.

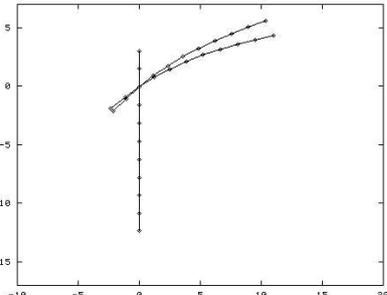


Fig. 7: Rotation around third point from left, with dynamics and relative limits.

Model	Steps needed	$c_s \neq 1$	$d_m > l_s$	Avg. force
static	116121	5259	963	2.01
dynamic	112998	0	1081	0.35
dynamic no rel. limit	113109	0	1266	0.36

Table 2: Comparison of some performance indicators of the DLO simulation. "Steps needed" is the number of times the model has been evaluated. The optimum value would be 112000, higher values are worse than lower ones. "Avg. force" is the average force applied to the gripper at point three. The last row repeats the second row, but with relative limits disabled.

The rotation of the gripper started 5 seconds after initialisation of the simulation and took 8 seconds. Each of the Figures 5, 6 and 7 shows the three time points 5, 13 and 27 seconds after initialisation.

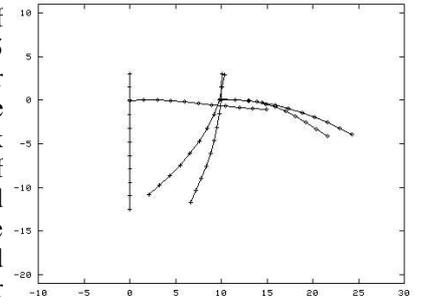


Fig. 8: Moving a DLO unaccelerated to the right, without dynamics.

The next series compares a linear horizontal movement with and without dynamics. The gripper starts moving with a fixed speed and stops instantaneous after 8 seconds.

The Figures 8 and 9 show the situation at the beginning of the grippers movement, at the end of the movement and 8 second after the movement stops. Table 4 summarizes some performance indicators for these tests.

The third series of experiments repeats the previous one, but this time with an accelerated movement.

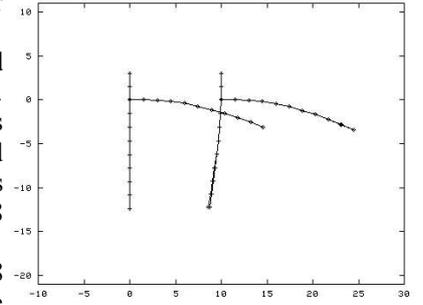


Fig. 9: Moving a DLO unaccelerated to the right, with dynamics enabled.

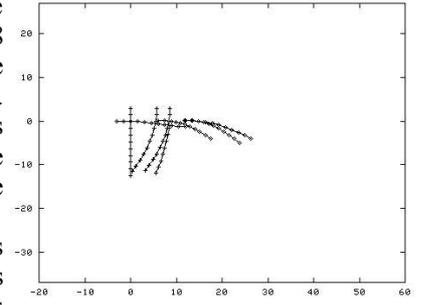


Fig. 10: Accelerated movement to the right, without dynamic.

mode	Steps needed	$d_m > l_s$	Avg. force
horizontal	112000	0	0.47
hor. + dynamic	112000	0	0.38
vertical	112878	963	0.82
vert. + dynamic	112998	1081	0.12

Table 4: Studing the effect of dynamic when moving a DLO horizontally, while the DLO itself is horizontally (first two rows) or vertically (last two rows) oriented. In all four cases  $c_s$  was always one. See Table 2 for meaning of the columns.

mode	Steps needed	$d_m > l_s$	Avg. force
horizontal	112891	974	0.65
hor. + dynamic	123364	25664	0.47
vertical	112878	963	0.67
vert. + dynamic	128770	34389	0.16

Table 3: Performance indicators for accelerated movement. The average forces are significant lower with dynamics enabled. In all four cases no oscillation has been occurred.

The Figures 10 and 11 show the situation at the beginning of the grippers movement, at the middle and the end of the movement and 8 second after the movement stopped.

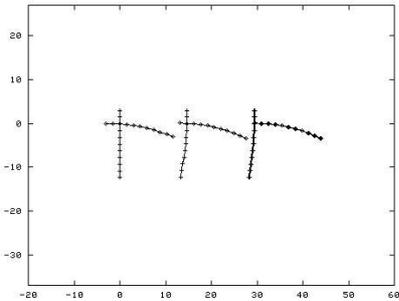


Table 3 displays some performance indicators for these tests. With dynamics disabled (Fig. 10), shortly after the movement starts, the points reach their displacement limit and all points (except the gripped one) effectively stop moving because of limit scaling. This results in large (and wrong) deformation of the DLO. Without such a limit (dynamics enabled, Fig. 11) the points can be accelerated beyond this displacement-limit and no deformation occurs.

Another series of experiments is used to evaluate the realism of our model compared to an energy-based simulation approach [8]. Within this experiment, we use a 50 cm long spring steel stick of 2 mm diameter and an elastic modulus  $E$  of 210000 N/mm<sup>2</sup>. To compare our force-based model with the energy-based model, we have to relate our model parameters  $c_F$ ,  $c_B$ ,  $c_C$  and  $N$  to the mentioned physical parameters. Using Formula 3 and

$F_{B, phys} = A E R^2 k^2 \frac{1}{4}$  with a maximum allowed bending  $k = 1/200 \text{ mm}^{-1}$  (derived from the allowed bending stress for the used spring steel [13]) we find:

$$c_b = A E R^2 k^2 \frac{1}{4} \frac{1}{(1 - \cos \alpha_l) \sin(\alpha_l/2)} \quad (21)$$

$$\text{with } \alpha_l = \frac{l k}{N - 1}$$

The difference in length between a partial circles arc and the corresponding chord has been neglected here. For  $c_c$  we find  $c_c = A E$  using Formula 11. The next table summarizes the values used:

Parameter	Value	Remarks
$E$	210000 N/mm <sup>2</sup>	Young's modulus
$\rho$	7.9 kg/dm <sup>3</sup>	Specific mass
$l$	50 cm	Total Length
$m$	0.013 kg	Total Mass of DLO
$k$	1/200 mm <sup>-1</sup>	Bending
$N$	16	Number of Points
$c_c$	659734	As per $A * E$
$c_b$	3574.96	As per Formula 21
$c_e$ (gripper)	0.981 N	At 5% segment length
$c_e$ (grav)	-0.00892 N	Gravitation force per point

Table 5: Model parameters derived from the physical material constants  $E$  (Young's modulus) and  $\rho$  (specific mass) for a 50 cm long, 2 mm diameter spring steel DLO. These values (except  $c_c$ , see next paragraph) are used to calculate the DLO shown in Fig. 12.

During several experiments using values from Table 5, we found that the physical "correct" value for  $c_c$  is to big by

two magnitudes to be used in our model. The model keeps oscillating and did not converge to a valid solution. Lowering the  $c_c$  value to 659.734 allows the model to converge.

The DLO shapes in Fig. 12 are calculated with a  $c_c$  of 659.734.

We have performed similar simulations with the energy-based approach found in [8] to evaluate the realism of the computed shapes.

As seen in Figures 12 and 13, the shapes produced by our force-based model using a given  $E$  modulus are typically closest to the shapes produced by the energy-based approach for a 1/3 lower  $E$  modulus.

We think this is due to the approximation  $1 - \cos(\alpha) \approx \alpha/10$  which is wrong for very small  $\alpha$ . Fig. 13 shows three DLO-shapes for the same physical parameters as used in the force-based approach, except that the value of  $E$  has been reduced to 1/3 of the value displayed in Table 5.

When comparing the two Figures 12 and 13, note that the energy-based approach cannot handle arbitrary grasp points and always treats the DLO as gripped at the rightmost point. This is an inherent limitation of the algorithm used. By contrast the current implementation of the force-based model can grasp the DLO anywhere except at the very first or last point of the DLO. This is not a limitation of the simulation algorithm itself, but of the current implementation of gripper DLO-interaction.

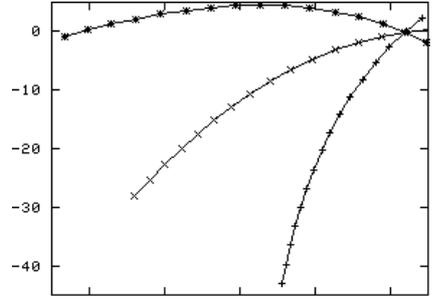


Fig. 12: DLO shapes calculated using the force-based model described in this paper, with the values from Table 5 (except  $c_c$  which is set to 659.734). The gripper orientation is  $-135^\circ$ ,  $-175^\circ$  and  $-215^\circ$ . The DLO is gripped at the 2<sup>nd</sup> point from right.

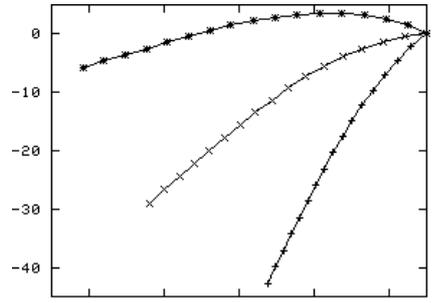


Fig. 13: DLO shapes calculated using the energy-based model described in [8], with the values from Table 5 (except  $E$  which was set to 63000). The gripper orientation is  $-135^\circ$ ,  $-175^\circ$  and  $-215^\circ$ . The DLO is gripped at the right most point

#### IV. CONCLUSIONS

We presented a fast and simple approach to compute 3D DLO simulations. While our simulation cannot produce physically exact shapes and forces, the simulated forces are realistic enough to drive a haptic feedback device like the "Phantom-Desktop" from Sensable Inc. All computations involved are basic 3D vector operations in Euclidean space. First experiments suggest, that the presented approach leaves enough spare time for other tasks and to fully integrate it in a 3D graphics, haptic feedback virtual programming system on current PC hardware.

The next section gives a brief overview on how we incorporate this new approach of a fast, force-based

simulation of DLOs for use in haptic feedback applications in our new, intuitive robot programming system “ViRoP” [3].

## V. APPLICATION EXAMPLE

The following is a brief overview on how to use the presented model for programming a manipulation task involving deformable linear objects (DLOs). See [3,12] for a more detailed discussion.

### A. System Overview

Our goal is to demonstrate an assembly task, involving a DLO in a rigid environment, in Virtual Reality using 3D optical and haptical feedback. The Virtual Reality system observes the demonstration and calculates all contact points and contact forces between the DLO and the environment. The gripper holding the DLO is just another part of the environment and contacts as well as contact forces to the gripper do not differ from all other contacts.

In a next step contact points, contact forces, the actual shape of the DLO and the grippers trajectory are used to compute a sequence of contact states and contact state changes including various parameters useful to re-identify such a contact state change in real sensor data, which is obtained by performing the same assembly task in real world and measuring torques and forces at the robots gripper. The idea is to have a robot controller that can use the sequence of contact state changes produced in Virtual Reality to conduct the assembly task in reality, based on his sensors, even in the presence of uncertainties in position and shape of the DLO and the environment. [6,7]

### B. Model Integration

The model is integrated in the system using the “Environmental Force” mechanism described in Section II.A. Contact forces with obstacles from the environment are modelled using constraint 2 from Formula 5 and gripper forces using constraint 3. While the gripper forces are updated at 1'000 Hz (frequency of the haptic feedback loop), the other contact forces are updated at approximately 25 Hz. The model itself runs at 10'000 – 30'000 Hz as stated above (discussion of  $c_F$  in Section II.B).

This is possible as only the parameters  $c_{P_j}$ ,  $c_{e_j}$  and  $N$  are updated at 1'000 Hz or 25 Hz, while the actual forces used in the model itself are calculated at the full 10'000 – 30'000 Hz rate based on these parameters and the current values for the points  $P_i$ . Adding an obstacles surface to the list of environmental forces, some time before the DLO hits the obstacle, prevents the DLO from ever moving beyond that obstacles surface.

The orientation of the DLO at the gripped point is maintained by gripping the DLO at two points instead of

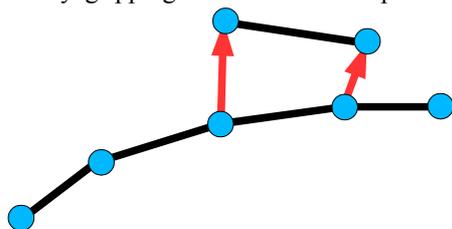


Fig. 14: The DLOs orientation is controlled by gripper forces acting on two different points of the DLO. The shape of the DLO will converge to the state where the forces from the gripper, from other contacts and the internal forces zero out.

one (Fig. 14). The forces acting back on the two points representing the gripper are used as force and torque values to drive a haptic feedback device.

## VI. REFERENCES

- [1] D. Henrich, H. Wörn (Eds.): "Robot manipulation of deformable objects". Springer-Verlag, London, 2000
- [2] S. Hirai, H. Wakamatsu and K. Iwata: "Modeling of Deformable Thin Pats for Their Manipulation". In: Proceedings of the 1994 International Conference on Robotics and Automation (ICRA'94), pp. 2955-2960, San Diego, USA, May 1994.
- [3] 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.
- [4] A. Loock und E. Schömer: "A Virtual Environment for Interactive Assembly Simulation: From Rigid Bodies to Deformable Cables". In: 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI'01), Vol. 3, P. 325-332
- [5] A. Remde, D. Henrich, S. Karl und H. Wörn: "Manipulating Deformable Linear Objects-Efficient Simulation of the Workpiece Behaviour". In: Int. Conf. on Robotics and Applications (IASTED RE'99), Santa Barbara, California, USA, Oct. 28-30, 1999.
- [6] A. Schlechter, D. Henrich: "Manipulating Deformable Linear Objects: Programming using Different Manipulation Skills". In: SPIE's International Technical Group Newsletter, Robotics and Machine Perception, vol.12 no. 1, March 2003.
- [7] A. Schlechter, D.Henrich et. al.: "Complete manipulation task of a leaf spring", MPEG-Video, <http://ai3.inf.uni-bayreuth.de/projects/rodeo/index.php>
- [8] S. Timm, B Kahl, D. Henrich: "Manipulation of deformable linear objects: Analysis of two-dimensional static approximation functions". Submitted to MMAR-05, Poland, June 2005.
- [9] H. Wakamatsu, S. Hirai und K. Iwata: "Modelling of linear objects considering bend, twist and extensional deformations". In: Proc. 1995 Int. Conf. on Robotics and Automation (ICRA'95), vol. 1, pp. 433438, Nagoya, Japan, May 1995.
- [10] H. Wakamatsu, T. Matsumura, E. Arai und S. Hirai: "Dynamic Analysis of Rodlike Object Deformation towards Their Dynamic Manipulation". In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Grenoble, France, Sept. 7-11, 19
- [11] S. Karl: "Physikalische Modellierung des mechanischen Verhaltens von deformierbaren linearen Objekten". Diplomarbeit, Fakultät für Physik, Universität Karlsruhe, 1999.
- [12] J. Bonnermann: "Virtuelle Roboterprogrammierung: Entwurf eines Prototypen mit haptischem Eingabegerät". Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 2002
- [13] Gieck: "Technische Formelsammlung", Gieck-Verlag 1989, ISBN 3-920379-17-9, p. Z-17