Path Planning for Deformable Linear Objects

Mark Moll, *Member, IEEE*, and Lydia E. Kavraki, *Member, IEEE*

***Abstract*—We present a new approach to path planning for deformable linear (one-dimensional) objects such as flexible wires. We introduce a method for efficiently computing stable configurations of a wire subject to manipulation constraints. These configurations correspond to minimal-energy curves. By restricting the planner to minimal-energy curves, the execution of a path becomes easier. Our curve representation is adaptive in the sense that the number of parameters automatically varies with the complexity of the underlying curve. We introduce a planner that computes paths from one minimal-energy curve to another such that all intermediate curves are also minimal-energy curves. This planner can be used as a powerful local planner in a sampling-based roadmap method. This makes it possible to compute a roadmap of the entire "shape space," which is not possible with previous approaches. Using a simplified model for obstacles, we can find minimal-energy curves of fixed length that pass through specified tangents at given control points. Our work has applications in cable routing, and motion planning for surgical suturing and snake-like robots.**

***Index Terms*—Deformation, differential geometry, flexible manipulation, flexible object representation, minimal-energy curves, modeling, motion planning, path planning.**

I. Introduction

There are many examples of manipulation tasks where flexibility is important: routing cables in buildings or cars, robot-assisted surgery, virtual reality applications, and manipulating paper or sheet metal. To successfully perform manipulating tasks in these contexts, we need a model of deformation/flexibility. Once we have a model, we need a computationally efficient way to simulate this model or solve motion planning queries. As always, there is a tradeoff between modeling accuracy and the efficiency of a simulator or planner, and for flexible objects, this is an especially important problem. There is an infinite number of shapes that a flexible object can take on, so to plan motions efficiently we have to approximate these shapes with a finite number of parameters. We also need to model the physical properties of the object given a certain shape parametrization. The model and associated computational efficiency are influenced heavily by the material properties of the object under consideration and the manipulation task.

There has been great progress in simulating deformable objects, but so far there has been only limited success in developing planners for deformable linear (one-dimensional) objects. We are working towards this end. This paper concentrates on representing and planning for curves of fixed length when given manipulation constraints. The constraints arise from robot grippers holding the endpoints of a wire, thereby fixing the positions and tangents at the endpoints. We assume that the wire is free to rotate about the endpoint tangents. In other words, the manipulators have built-in compliance along the tangential direction. This can be thought of as holding a wire with slippery fingers. Our approach is almost independent of the model of the physical properties of the wire. The physical model used in this paper can easily be extended. We do not consider the motion planning for the grippers. This is similar to the work on assembly planning where the tools for assembly or removal of parts are not considered [1], [2]. The planning problem addressed by this paper is: Given manipulation constraints for start and goal configurations, 1) find stable configurations of the wire that satisfy those constraints, and 2) find a path between these configurations such that all configurations along the path are stable, as well. A stable configuration is defined as a configuration with minimal strain for given endpoint constraints. These configurations correspond to minimal-energy curves. The contributions of this paper are an efficient algorithm for computing minimal-energy curves and a powerful local planner that computes paths of minimal-energy curves. A key part of these contributions is a new, adaptive representation for deformable linear objects. This work represents a significant step towards a general purpose motion planner for deformable linear objects.

Deformable linear objects are usually represented using finite element models (FEMs) [3]. Since the complexity of path planning increases exponentially with the number of degrees of freedom (DOFs) [4], this means that an extensive exploration of the configuration space for deformable objects using FEMs is very hard. Moreover, finding configurations that satisfy endpoint constraints or energy constraints is complicated by the existence of many local minima. Our approach has been to use subdivision to make the path-planning problem for deformable objects more tractable. Subdivision is an area of geometric modeling concerned with the compact representation of curves and surfaces [5]. In our case, subdivision allows us to adaptively refine approximate solutions and reduce the tendency of getting stuck in local minima. In our parametrization, we can compactly represent shapes of varying complexity. Our planner can find paths between curve configurations with a different number of parameters.

The main motivation for our research comes from motion planning for deformable linear objects. One important application area is cable modeling and handling in the automotive industry. Another application is surgical suturing (see Fig. 1).

Manuscript received August 27, 2005; revised January 23, 2006. This paper was recommended for publication by Associate Editor F. Thomas and Editor H. Arai upon evaluation of the reviewers' comments. This work was performed when M. Moll was with the Department of Computer Science, Rice University. This work was supported in part by the National Science Foundation under Grants 0205671 and 0308237, in part by CCF-0523908, in part by a Sloan Fellowship, and in part by CNS-0454333.

M. Moll is with the Information Sciences Institute, University of Southern California, Marina del Rey, CA 90292 USA (e-mail: moll@isi.edu).

L. E. Kavraki is with the Department of Computer Science, Rice University, Houston, TX 77005 USA (e-mail: kavraki@cs.rice.edu).

Digital Object Identifier 10.1109/TRO.2006.878933

1552-3098/$20.00 © 2006 IEEE

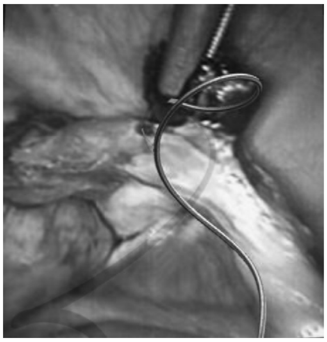

Fig. 1. Simulated surgical suture (image courtesy of D. Pai).

A suture is a flexible wire with negligible stretch that typically needs to go from a straight configuration to a knot. Limited visibility and limited tactile feedback can make this a challenging task for a surgeon. As part of a training simulator, a motion planner for sutures can be a very useful tool for training surgeons. A theoretical motivation for this work is to create a better understanding of the accessible configuration space of a flexible object. For example, little is known about the topology of the set of stable configurations of a given flexible object. Although this paper does not formally characterize the topology of deformable linear objects, it does provide techniques for approximating the accessible configuration space using probabilistic roadmap methods (PRMs) [6]. As the number of sampled configurations in the roadmap increases, the roadmap becomes a more accurate representation of the accessible configuration space.

The outline of the rest of the paper is as follows. Section II briefly describes some related work. Section III explains what minimal-energy curves are and why we are interested in them. In Section IV, we introduce a subdivision scheme for computing minimal-energy curves subject to endpoint constraints. As part of the subdivision scheme we need to align curves with these endpoint constraints. This exploits some of the symmetries that minimal-energy curves have. This alignment procedure is explained in Section V. In Section VI, some performance improvements of the basic algorithm are discussed. Here, we will also discuss the stability of the algorithm with respect to some algorithm parameters. In Section VII, we present a path-planning algorithm for minimal-energy curves. Our minimal-energy curve construction can be extended to multiple control points, which is described in Section VIII. Section IX describes our implementation and gives some performance results. Finally, Section X summarizes the contributions of this paper and outlines directions for future research.

II. Related Work

The related work on deformable linear objects can be divided into three overlapping categories: modeling, simulation, and planning. In the geometric design community, stable configurations of deformable linear objects are often called minimal-energy curves. These curves appear in the broader context of fair curve and surface design [7]–[11]. Here, "fair" means minimizing some functional (or energy function). In our case, this functional is defined as the integral of curvature squared plus torsion squared. We assume there is no stretching, i.e., the length of the curve is fixed. There is very little work on finding fair curves of fixed length subject to endpoint constraints. Our work provides an algorithm for doing just that. The definition of energy can be changed without affecting the rest of the algorithm.

Fair curve design focuses almost exclusively on planar curves. Usually, the length of the curve is either unconstrained or there is a stretching energy term in the energy functional. Horn [12] derives an analytic expression for a planar minimal-energy curve and uses arcs of a circle to approximate minimal-energy curves. Kallay [13] extends this result to planar minimal-energy curves of given length. Brunnett [11] derives several properties of so-called free elastic curves: planar minimal-energy curves of variable length without tangent constraints. Jou and Han [10] consider planar minimal-energy curves of given length *with* tangent constraints at the endpoints. They also present a simple algorithm for computing such curves. Simply put, the algorithm divides a curve in segments of constant curvature and solves the constrained energy minimization problem as function of the curvatures of these segments. In this paper, we extend this idea to three dimensions (3-D) and make the algorithm more scalable to a large number of segments.

Wesselink and Veltkamp [8], [9] describe several curve energy operators. The emphasis here is on interactive curve design. To make this practical, approximations for bending, twisting, and stretching energy are used. In contrast to other approaches, this work is not limited to planar curves. Kallay [14] presents a discrete approximation algorithm for finding 3-D minimal-energy curves of given length with endpoint constraints. Here, the energy is just the integral of curvature squared, whereas we include the integral of torsion squared. The algorithm constructs a polyline consisting of equal length segments. It iteratively rotates all the points between any pair of vertices of the polyline so as to minimize the discrete approximation of curvature.

Wakamatsu and Hirai [15] model the static deformation of a linear object with four functions: three to describe the change in orientation along the curve and one to describe the extension along the curve. These functions are approximated by a linear combination of a set of basis functions. Given an energy function in terms of these coefficients, one can then easily solve for the coefficients that result in a stable configuration. Wakamatsu and Hirai also validated this model experimentally. They found a good agreement between the theoretical predictions and experimental results. Compared to this work, our work offers a subdivision-based computational scheme to compute stable configurations that appears to be much faster. We also use this scheme to compute paths of stable configurations. In [16], Wakamatsu *et al.* extend their results to the two-dimensional dynamic case. The model can be used to control the deformation on a linear object. The control law in [16] was experimentally verified.

To simulate a deformable object, we need to compute *any* physically plausible configuration; not just stable configurations. The emphasis is on efficiency in computing the response of an object to internal and external forces. Phillips *et al.* [17] use a spline of linear springs. Adaptive subdivision is used to handle stretching and contraction of the rope. Friction is not modeled. Brown *et al.* [18] model a suture as a polyline (which during rendering is replaced with a smooth spline). Forces act on the vertices of the polyline. Using a few simple rules, the positions and velocities of all vertices can be updated in real time. Friction is not explicitly modeled, but the collision resolution scheme produces a friction-like effect. More so than the previous two papers, Pai [19] focuses on the dynamics of a suture. A suture is modeled as a so-called Cosserat rod: a curve with coordinate frames along the curve denoting the reference orientation. The differential equations describing the dynamics in this representation can be solved very efficiently. Remde and Henrich [20] give an overview of simulation of deformable linear objects and present a basic algorithm to perform "inverse simulation," i.e., solve for the object's shape for given endpoint constraints.

Hopcroft *et al.* [21] propose a programming language to program deformable object manipulation. They describe basic operations such as grasping a segment along a curve, moving it, and dropping it, but also more complex operations, such as crossing segments. A vision system is used to obtain the configuration of the physical rope. Several knot programs were experimentally verified. These knot programs were still written "by hand." Recently, Wakamatsu *et al.* [22] proposed a manipulation planner for knotting/unknotting that generates such programs automatically. This planner has been implemented on a 6-DOF manipulator with a camera.

Lamiraux and Kavraki [23] introduce one of the first open-loop motion planners that deals with flexibility explicitly. In their work, a flexible object is modeled using a finite element mesh. They find stable configurations subject to manipulation constraints using a global energy minimization. Bayazit *et al.* [24] propose a path planner that first produces a path where a deformable object is allowed to penetrate obstacles. It then proceeds to deform the object to resolve any collisions. The emphasis here is more on realistic looking motions rather than modeling the underlying physics. Gayle *et al.* [25] introduce a physically realistic planner with a new fast collision checking scheme for flexible objects. Their planner uses a roadmap based method [6] that samples in the workspace rather than in the configuration space. Paths are generated for a point-robot in this roadmap. A deformable object is then guided along these paths as long as physical constraints can be satisfied. This approach works well if the obstacle-free configuration space of the deformable object is not too dissimilar from the obstacle-free part of the workspace. For certain important applications, such as catheters moving through arteries, this is the case. Saha and Isto [26] are currently developing a roadmap based planner for deformable linear objects. They use a fast simulator as a subroutine to compute configurations near previously sampled configurations. Our work can be used to replace this subroutine with one that computes paths consisting only of stable configurations.

One of the difficulties with planning for deformable objects is contacts between a deformable object and the environment. Acker and Henrich [27] classify the different possibilities into different topological states and describe their stability. They also enumerate all possible state transitions. This kind of information can be used to guide a path planner.

Ladd and Kavraki [28] applied motion planning techniques to mathematical knots. Here, physical realism is irrelevant, but the configuration space tends to be more complex than in the aforementioned papers. Using an artificial potential function to guide the sampling, they were able to untangle very complex knots.

Sometimes, hyperredundant robots (or snake robots) are modeled as flexible curves [29], [30]. In this context, minimal-energy curves may provide good reference shapes for the robot that minimize joint movement. The work by Zanganeh and Angeles [30] is especially similar to the modeling of elastic wires. Similar to the approach taken in this paper, they pose the problem of finding the optimal shape as an optimization problem over spline parameters. Optimality of a shape is expressed in terms of curvature and torsion. A radically different approach to path planning for redundant manipulators was taken by Nakamura and Hanafusa [31]. They posed the problem as an optimal control problem. They reduce the problem to a boundary value problem and use Pontryagin's maximum principle [32] to find an optimal solution for different definitions of optimality.

In previous work [33], we presented an approximate representation of minimal-energy curves using only ten parameters. We described different methods to solve for these parameters for given endpoint constraints. Although this parametrization produced good results overall, there were cases where a good approximation of a minimal-energy curve could not be found. Moreover, it is computationally very expensive to verify if an approximation is close to a curve that has minimal energy in the variational sense. We, therefore, started investigating adaptive parametrizations that vary the number of parameters based on the complexity of a minimal-energy curve. We informally use the term "complexity of a curve" to describe some measure of the change in shape (i.e., curvature and torsion) along the curve. This paper is a revised and expanded version of [34].

III. Minimal-Energy Curves

Minimal-energy curves correspond to stable configurations of the wire they represent. If we assume quasistatic dynamics, the paths produced by our planner are as close as possible to what would happen if a robot executes the plan. With minimal-energy curves, we only have to consider dynamics to the extent that the kinetic energy should be smaller than the energy needed to leave a potential energy well. So, even if a robot cannot follow a computed trajectory exactly, the configurations along the trajectory act as attractors for configurations in a neighborhood.

Minimal-energy curves can be thought of as representing wires with minimal strain. We assume that a straight line segment without torsion represents the shape with zero strain. The Darboux vector [35, p. 205], defined in terms of the Frenet frame [36] as $D = \tau T + \kappa B$, describes the rotational strain along the curve. Here, T and B are the tangent and binormal, respectively, and τ and κ denote the torsion and curvature (see Section IV-A for an overview of the nomenclature used in this paper). We assume there is no translational strain: The wire does not stretch. We define the energy of a curve to be the integral of $\|D\|^2$ along the curve. In other words, the energy is the integral of the curvature squared plus the torsion squared over the entire length of the curve. This simple model captures the essential internal energy of a wire. Obviously, the bending and twisting energy can be weighted differently and other terms can be added, as well. For a more complete description of the dynamics of elastica, see, e.g., [37]. We will first consider only curves of constant length that satisfy constraints on the positions and tangents at the two endpoints. This corresponds to a wire being held by the endpoints. Finding such curves is nontrivial. Splines tend to produce very smooth low-energy curves that can match arbitrary endpoint constraints, but the length of the splines is variable. A finite-element method, where we would represent the curve by a large number of line segments would preserve the length, but makes planning difficult because we need many DOFs [23]. Finding a smooth curve of fixed length that satisfies endpoint constraints is difficult, and finding minimal-energy curves using a finite element method is even more challenging.

Very little is known about 3-D minimal-energy curves of given length. For *planar* minimal-energy curves with endpoint constraints the following variational condition on curvature and its second derivative has to be satisfied along a curve (parametrized by arc length s): $\kappa''(s) + (1/2)\kappa^3(s) = c \cdot \kappa(s)$ for some constant c [10] (without endpoint constraints $c = 0$). Unfortunately, there is no equivalent constraint for minimal-energy curves in 3-D.

The following two observations will be important in the rest of this paper.

- The space of all minimal-energy curves exhibits many symmetries: A minimal-energy curve is still a minimal-energy curve if we apply a translation, a rotation, a uniform scaling, or a reflection. We will take advantage of this property by only solving for minimal-energy curves in some canonical form, from which all symmetric curves can easily be derived.
- For a minimal-energy curve, every segment of that curve is also a minimal-energy curve. This suggests that we can locally improve an approximation of a minimal-energy curve. We, therefore, conjecture that the complexity of finding parameters for minimal-energy curves increases linearly with the number of parameters required to represent that curve instead of exponentially.

IV. Subdivision Scheme for Minimal-Energy Curves

Subdivision is an area of geometric modeling concerned with compact representations of curves and surfaces [5]. The representations consist of a coarse mesh or polyline and a set of refinement rules. The refinement rules define how elements of the mesh can be subdivided into smaller elements. The surface represented by the mesh and refinement rules is the limit surface obtained by iteratively applying the refinement rules to the mesh. Typically, the rules can be thought of as a weighted interpolation scheme.

A. Nomenclature

In this section, we will briefly introduce the notation used in this paper. Let a curve $\boldsymbol{x}$ be parametrized by arc length s. A point on the curve is denoted $\boldsymbol{x}(s)$. The tangent at that point is written as $\boldsymbol{x}'(s)$ or $T(s)$. The magnitude of T', the derivative of T, is called curvature and is written as $\kappa(s)$. The vector $N(s) = T'(s)/\kappa(s)$ is the normal of the curve at s and is orthogonal to $T(s)$. The cross product $B = T \times N$ is called the bi-normal. Together, T, N, and B form the so-called Frenet frame [36]. The magnitude of the B' is called torsion and is denoted $\tau(s)$. We will introduce a compact representation of curves with piecewise constant curvature and torsion. The parameters of such curves are described by configurations q_i.

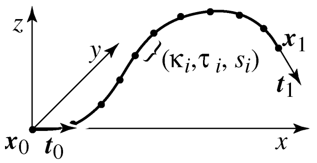

Fig. 2. Curve consisting of n helical segments. Each segment is parametrized by curvature κ_i, torsion τ_i, and segment length s_i. The positions and tangents of the endpoints are denoted by $\boldsymbol{x}_i$ and $\boldsymbol{t}_i$ ($i = 0,1$), respectively.

B. Representation

We have developed an algorithm for representing minimal-energy curves inspired by subdivision techniques. By using subdivision instead of a fine-grain representation, such as FEMs, we gain computational efficiency and reduce the tendency of getting stuck in local minima. There are three factors that make our scheme more complicated than most subdivision schemes. First, at each iteration we do not subdivide all segments simultaneously. Instead, we adaptively subdivide segments one at a time. Second, to minimize the energy and at the same time maintain the constraints on the endpoints, we need to solve a constrained minimization problem rather than simply apply an interpolation rule. Finally, we want to maintain the length of the curve. To accomplish this, we represent a curve as a sequence of n segments with constant curvature and torsion, i.e., parts of helices (see Fig. 2). When a segment is subdivided, the sum of the lengths of the new segments is equal to the length of the old segment. Each segment of a curve can be described by curvature, torsion, and length. So, for a curve consisting of n segments, we need $3n$ parameters. Fig. 3 shows a minimal-energy curve and corresponding curve parameters obtained using our algorithm described below.

We will describe a new algorithm that, given manipulator constraints like the endpoints and tangents where a wire is held, finds a minimal-energy curve that satisfies those constraints. We assume that the wire is free to rotate about the endpoint tangents to reach the shape with lowest energy. In other words, the manipulators have built-in compliance along the tangential direction. This can be thought of as holding a wire with slippery fingers. The idea is to start with a simple curve that just satisfies the endpoint constraints and keep refining it as long as we can lower the energy of a curve. The basic refinement step can informally be stated as follows: As long as the difference in curvature and torsion between a segment and one of its immediate neighbors is larger than some threshold, subdivide both and optimize the curve parameters of the subdivided segments so as to simultaneously minimize the energy and the error in the endpoint constraints. Here, we make use of the observation that we can locally change the shape to get closer to a minimal-energy curve.

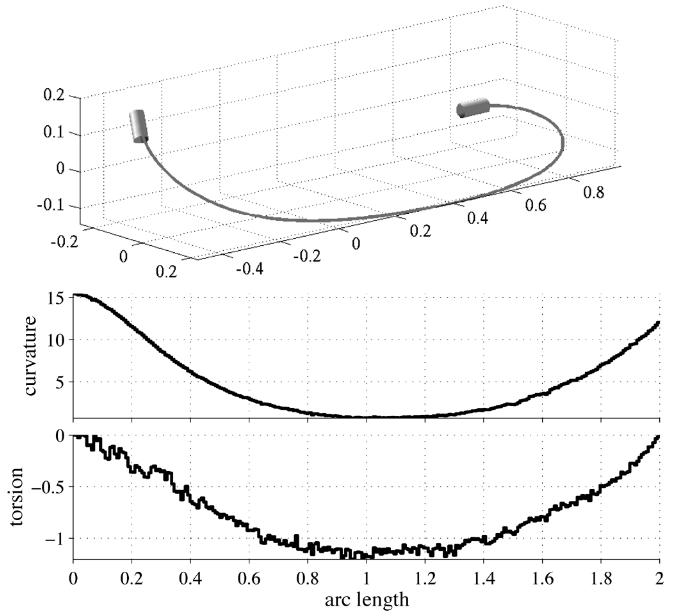

Fig. 3. Minimal-energy curve of length 2. The curve is held at the endpoints, constraining both the positions and the tangents. This is visualized using small cylinders. The bottom two plots show the curvature and torsion along the curve.

We also take advantage of the symmetries by solving only for minimal-energy curves in "canonical form" and aligning these curves through an affine transform and scaling with the desired endpoints and tangents. Typically, the error in the endpoint constraints is very close to zero after the first subdivision step. Subsequent steps minimize the energy while maintaining the constraints.

The parametrization supports the following operations in a straightforward manner: downsampling to a coarser resolution, upsampling to a finer resolution, computing the distance (or shape difference) between two curves, and finding points along a curve. All these operations take time linear in the number of segments. The distance between two curves q_0 and q_1 of length 1 is defined as

$$d(q_0, q_1) = \sqrt{\int_0^1 \left((\kappa_0(s) - \kappa_1(s))^2 + (\tau_0(s) - \tau_1(s))^2 \right) ds}. \tag{1}$$

Because the curves have piecewise-constant curvature and torsion, the integral simplifies to a summation. Using upsampling and downsampling we can represent a curve at different levels of detail. Curves in this representation can also be compressed very well using, e.g., wavelets [38]. These primitives are all necessary for path planning for minimal-energy curves in our representation. It allows us to connect configurations with a varying number of DOFs.

We can think of a curve in our representation as a kinematic chain where the affine transforms from the start of a segment to its end are controlled by the curvature and torsion parameters. Below, we will derive a closed-form expression for the forward kinematics of a piecewise-helical curve. This is useful to find a curve that satisfies endpoint constraints. Let a piecewise-helical curve consisting of n segments be described by a $n \times 3$ matrix q, where row i contains the parameters for segment i: (κ_i, τ_i, s_i). A helix with curvature κ and torsion τ can be described by a parametric unit-speed curve h

$$h(s) = \frac{1}{\kappa^2 + \tau^2} \begin{pmatrix} \kappa \cos(s\sqrt{\kappa^2 + \tau^2}) \\ \kappa \sin(s\sqrt{\kappa^2 + \tau^2}) \\ s\tau\sqrt{\kappa^2 + \tau^2} \end{pmatrix}.$$

Let $R(s)$ denote the Frenet frame [36] along the curve. The relative change in orientation between the frame at $s = 0$ and at $s = \ell$ is then given by $R(0)^T R(\ell)$. Similarly, the relative change in position is given by $R(0)^T(h(\ell) - h(0))$. After writing out these expressions, we can obtain an expression for the homogeneous transform from the start of segment $q_i = (\kappa_i, \tau_i, s_i)$ to the end

$$T(q_i) = \begin{pmatrix} \Delta R(q_i) & \Delta \boldsymbol{x}(q_i) \\ \mathbf{0} & 1 \end{pmatrix}$$

where

$$\begin{aligned} \Delta R(q_i) &= \frac{1}{\rho^2} \begin{pmatrix} \kappa^2 \cos r + \tau^2 & -\kappa\rho \sin r & \kappa\tau(1 - \cos r) \\ \kappa\rho \sin r & \rho^2 \cos r & -\tau\rho \sin r \\ \kappa\tau(1 - \cos r) & \tau\rho \sin r & \kappa^2 + \tau^2 \cos r \end{pmatrix} \\ \Delta \boldsymbol{x}(q_i) &= \frac{1}{\rho^3} \begin{pmatrix} \kappa^2 \sin r + \tau^2 r \\ \kappa\rho(1 - \cos r) \\ \kappa\tau(r - \sin r) \end{pmatrix}, \ r = \rho s \text{ and} \\ \rho &= \sqrt{\kappa^2 + \tau^2} \end{aligned}$$

(the subscript i has been dropped for convenience). The homogeneous transform from the start of the curve to the end is simply the product of the transforms for each segment $T(q) = \prod_{i=1}^{n} T(q_i)$.

C. Energy Minimization

If a curve segment q_i is subdivided into smaller segments, described by the matrix q_{new}, the curvature and torsion parameters of the smaller segments are optimized to minimize

$$\text{energy } (q_{\text{new}}) + K \cdot (\exp(\text{err}(q_{\text{new}})) - 1) \tag{2}$$

$$\text{where energy } (q) = \sum_{i=1}^{n} \left(\kappa_i^2 + \tau_i^2\right) \cdot s_i. \tag{3}$$

K is a penalty constant, and the error is measured after alignment, as described in the next section. A sufficiently large value for K, combined with the exponential scaling of the error, almost guarantees that the error is very close to zero after one or two subdivision steps. Note that we are *locally* optimizing the shape and at the same time trying to satisfy *global* endpoint constraints. Each subdivision can be performed fairly quickly, since we are minimizing over only a small number of parameters.

D. Subdivision Details

In our implementation, we have chosen to subdivide each segment into two smaller segments. Subdividing one segment would give us *four* parameters to optimize over: two curvature parameters and two torsion parameters. However, satisfying the constraints requires at least *five* DOFs: three for the endpoint position and two for the endpoint tangent. Therefore, we need to subdivide two segments at once, giving us eight DOFs, three of which can be used for energy minimization. Initially, we start off subdividing two helical segments of equal length with arbitrary curvature and torsion. To decide which segments to subdivide in subsequent steps, we consider the difference in curvature and torsion between consecutive segments. Let the difference between segment i and $i+1$ be defined as

$$\left((\kappa_{i+1}-\kappa_i)^2+(\tau_{i+1}-\tau_i)^2\right)\cdot\max(s_i,s_{i+1}).$$

Generally speaking, the minimization in a subdivision step will minimize the energy by smoothing out the difference in curvature and torsion. We maintain a priority queue of the differences between all consecutive segments. The largest difference is assigned the highest priority. Intuitively, subdividing the segments with the largest difference between them should likely result in the largest decrease in the energy and the error. We keep subdividing as long as the error in the endpoint constraints is larger than some threshold and as long as the difference between some consecutive segments is larger than some other threshold. If the difference in curvature and torsion between any pair of consecutive segments is small, then subdividing is not going to reduce the energy much.

One detail ignored in the explanation above is that the pair of segments being subdivided may not be of equal length. If they *are* of equal length, then both segments are subdivided. If one is longer than the other, only the longer segment is subdivided into segments with length equal to that of the shorter segment. This approach gives all segments equal weight during minimization. Observe that the number of curve parameters being optimized is at least 6, just large enough to maintain endpoint constraints and lower the energy. By keeping the branching factor of the subdivision as small as possible, we aim to arrive at the most compact representation.

V. Alignment of a Curve to Match Constraints

In our subdivision scheme, we maintain a curve in canonical form and use an alignment procedure to match up the curve with the endpoint constraints as best as possible. As we mentioned before, the curve representation would not change if we apply a translation, rotation, scaling, or reflection to the endpoint constraints. The alignment procedure returns the transform that brings the endpoint constraints in canonical form such that the error in satisfying the constraints (as defined below) is minimized. This way, we exploit the symmetries of minimal-energy curves.

Suppose we are interested in finding a curve of length L, having endpoints $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ and unit tangents $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$. Our subdivision scheme produces curves in canonical form: They are all of length 1 and with $\boldsymbol{x}_0^*=\boldsymbol{0}$ and $\boldsymbol{t}_0^*=(1,0,0)^T$. Let $\boldsymbol{x}_1^*$ and $\boldsymbol{t}_1^*$ be the position and tangent at endpoint 1 of a curve q in canonical form. They are simply the fourth and first column of $T(q)$. If q_{new} describes the curve parameters of the segments being subdivided, then $T(q)$ can be written as $T(q)=T_0\cdot T(q_{\mathrm{new}})\cdot T_1$, where T_0 and T_1 are the combined transforms for the segments that precede and follow q_{new}, respectively. So, as q_{new} changes during the energy minimization, it is very easy to update $T(q)$ and, therefore, also $\boldsymbol{x}_1^*$ and $\boldsymbol{t}_1^*$. The alignment procedure described below simply constructs another transform that aligns $T(q)$ with a transform derived from the endpoint constraints.

During energy minimization we use a penalty method to satisfy the endpoint constraints. The error in the endpoint constraints is measured after alignment. The alignment is done in two steps. First, we apply a transform that minimizes the translational error. In the second step, we apply a rotation that minimizes the error in the tangents, but does not move the endpoints. Let $\boldsymbol{x}=(\boldsymbol{x}_1-\boldsymbol{x}_0)/L$ and $\boldsymbol{x}^*=\boldsymbol{x}_1^*-\boldsymbol{x}_0^*$. The translational error is $\|\boldsymbol{x}-\boldsymbol{x}^*\|$. This error is minimized if we apply a rotation R to $\boldsymbol{x}^*$ such that the angle between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}=R\boldsymbol{x}^*$ is 0. The next step is to minimize the error in the tangents. Let $\boldsymbol{u}=\boldsymbol{x}/\|\boldsymbol{x}\|$. We want to find a rotation R_ϕ about this axis such that the following error measure is minimized

$$\mathrm{err}=\left(1-\boldsymbol{t}_0^T R_\phi \boldsymbol{t}_0^*\right)+\left(1-\boldsymbol{t}_1^T R_\phi \boldsymbol{t}_1^*\right)+c\cdot\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|^2$$

where c is some positive weighting constant. Note that the error is equal to 0 if and only if there is no translational error and there exists a rotation R_ϕ that simultaneously aligns $\boldsymbol{t}_0$ with $\boldsymbol{t}_0^*$ and $\boldsymbol{t}_1$ with $\boldsymbol{t}_1^*$. Using Rodrigues' formula [39], we can write R_ϕ as $I+\sin\phi\cdot\hat{\boldsymbol{u}}+(1-\cos\phi)\cdot\hat{\boldsymbol{u}}^2$, where $\hat{\boldsymbol{u}}$ is the matrix such that $\boldsymbol{u}\times\boldsymbol{v}=\hat{\boldsymbol{u}}\boldsymbol{v}$ for any vector $\boldsymbol{v}$. Hence, the derivative of the error function with respect to ϕ can be written as

$$\frac{\partial}{\partial\phi}\mathrm{err}=-\cos\phi\left(\overbrace{\boldsymbol{t}_0^T\hat{\boldsymbol{u}}\boldsymbol{t}_0^*+\boldsymbol{t}_1^T\hat{\boldsymbol{u}}\boldsymbol{t}_1^*}^{a}\right)-\sin\phi\left(\overbrace{\boldsymbol{t}_0^T\hat{\boldsymbol{u}}^2\boldsymbol{t}_0^*+\boldsymbol{t}_1^T\hat{\boldsymbol{u}}^2\boldsymbol{t}_1^*}^{b}\right).$$

So, the extrema of the error function are at $\phi=\arctan(-a/b)\mathrm{mod}\ \pi$. By inspection, we can determine which value for ϕ minimizes the error. The scaling, translation, and the two rotations R and R_ϕ can be combined in one transform that aligns a curve in canonical form to general constraints.

VI. Curve Refinement and Stability

In this section, we describe different ways to refine the basic subdivision algorithm for computing minimal-energy curves. Refinement is not strictly necessary, but is used to improve/test the stability and reproducibility. The goal of refinement is to lower the energy of a curve even further. Refinements come in two forms: local and global refinements. Local refinements change the basic subdivision step. Global refinements use the basic subdivision algorithm as a subroutine to find better approximations of minimal-energy curves. Related to refinement is the issue of stability. Ideally, the refinement procedures do not drastically change the shape. Small changes in the initial guess for torsion and curvature should produce curves that have similar curvature and torsion and, thus, similar energy and a similar embedding in $\mathbb{R}^3$. Although we cannot prove that our subdivision scheme has this property, we will show below that, at least in practice, that seems to be the case.

A. Local Refinement

There are several ways the subdivision scheme can be improved. The main flaw of the basic scheme is that it does not address singularities. By "singularities," we mean that the effective number of DOFs is smaller than the number of parameters we can change. If the "true" minimal-energy curve is planar, then making the torsion nonzero will increase the energy. The effective number of DOFs is, therefore, only four, if we are optimizing the curve parameters of four segments, rather than eight DOFs in the general case. Since we need five DOFs just to satisfy the endpoint constraints, we will not be able to minimize the energy. If we could recognize this situation, then we could increase the number of segments to be subdivided so that we have a sufficient number of DOFs. If the error in the endpoint constraints is small (which is usually the case after the first subdivision), then we can use the affine transform for the segments under consideration to check if the curve segment is close to being planar. A necessary condition for a curve segment to be planar is that the vector connecting the endpoints and the tangents are coplanar. Using the notation from the previous section, this condition can be expressed as

$$\boldsymbol{u} \cdot (\boldsymbol{t}_0^* \times \boldsymbol{t}_1^*) = 0.$$

The left-hand side measures the volume of the paralleliped defined by $\boldsymbol{u}$, $\boldsymbol{t}_0^*$, and $\boldsymbol{t}_1^*$. If this expression is 0 (or close to 0), we subdivide one more neighboring segment, namely the one that has the largest difference with the segments already selected. Sometimes, this refinement is still not enough. For instance, if after minimizing the energy has hardly decreased, then we may want to add a neighboring segment and redo the subdivision and energy minimization.

B. Global Refinement

In our implementation, the scheme starts off with two segments with the following parameters: $(\kappa_1, \tau_1, s_1) = (1, 2, (1/2))$ and $(\kappa_2, \tau_2, s_2) = (2, 1, (1/2))$. This choice is arbitrary; many other choices would work equally well, on average. However, for *specific* endpoint constraints, there may be *specific* initial conditions that result in a minimal-energy curve with lower energy. Note that, for given endpoint constraints, there may exist many different minimal-energy curves. To see this, consider the following simple example. Suppose we want a minimal-energy curve where both endpoints are at the origin and both tangents are along the x axis. Then, a circle of the appropriate radius is the global energy minimum, but a curve of constant curvature that makes a $2\pi k$ turn $(k = 2, 3, \ldots)$ is also a minimal-energy curve. If we make the curve parameters of the initial segments "small," then we are biasing the subdivision scheme towards the global energy minimum.

There are three ways to improve the chances of finding a curve at a global energy minimum. The first approach is to just repeatedly run the subdivision scheme starting from different initial curve parameters. Obviously, this is computationally rather expensive. The second approach is to run the subdivision scheme once with arbitrary initial curve parameters, and use the final curve to come up with an improved guess for the initial curve parameters. This improved guess can be derived by downsampling a minimal-energy curve to a coarse resolution and start the subdivision at that resolution. The second approach is more efficient than the first one, but is limited to finding minimal curves in a neighborhood of the initial curve. Finally, to further refine a curve we can perform a global energy minimization, where the energy and error (see expression 2) are minimized as a function of *all* curve parameters. Of course, all these techniques can be combined for even better results.

C. Refinement and Stability

We can now consider the stability of the subdivision algorithm. In our implementation, we added the local refinements mentioned above and tested the three global refinement techniques. Fig. 4 shows the curvature and torsion profiles for six different sets of endpoint constraints. The endpoint positions were picked uniformly random within a unit ball, and the tangents were picked uniformly random as points on a unit sphere. The curve length was set to be 2 (the diameter of a unit ball). The subdivision and energy minimization tolerances were set to very small values so that we can study what limit curves the subdivision algorithm converges to. For each set of endpoint constraints, we computed 30 different curves as follows. The thick solid curve shows the profile for the curve constructed using the default initial parameters. The dashed curve shows the profile after downsampling that curve to eight segments and rerunning the subdivision scheme. The remaining 28 curves were created with initial curve parameters drawn uniformly random from the $[-2, 2]$ interval. The curve with lowest energy is drawn with a dot-dash line. In the legend, the energies for these curves are shown. The color of the curves corresponds to the energy: A dark color means low energy, a light color means higher energy. This figure shows that for certain endpoint constraints the subdivision scheme is relatively insensitive to the initial conditions, whereas, for others, the scheme can converge to several distinct minimal-energy curves. Generally speaking, the different initial curve parameters produce a small number of clusters of minimal-energy curves. Within a cluster, curves tend to have similar energy. The default initial curve parameters tend to produce curves in a low-energy cluster. Notice in the last set of curves in Fig. 4, there can be a large difference between the curves, even though the curvature and torsion profiles do not seem to be that different. In this case, this is caused by the turn at the end of the curves in the low-energy cluster that is not present in the high-energy cluster.

We also tried the global optimization of curve parameters as described above. The minimization procedure used was a quasi-Newton method with a BFGS approximation to the Hessian, and with gradients computed using finite differences [40]. This produced curves that were very close to the starting curve for the optimization. This suggests that the subdivision scheme produces curves that are close to being minimal in the variational sense.

VII. Path Planning for Minimal-Energy Curves

The path-planning problem for minimal-energy curves can be stated as: Given endpoint constraints for a start and goal curve,

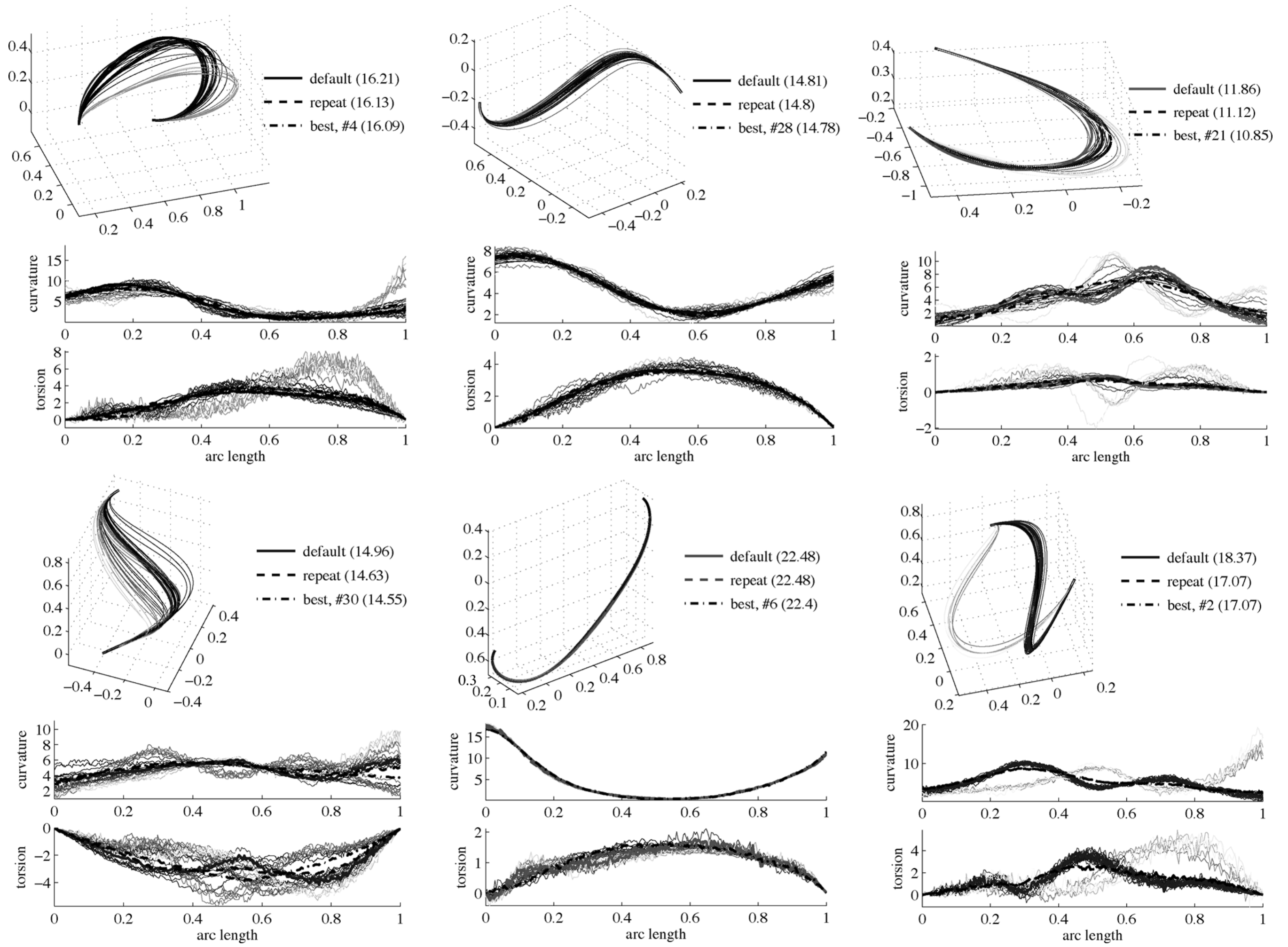

Fig. 4. Curve refinement. The curvature and torsion graphs show the curvature and torsion profile for different minimal-energy curves with the same endpoint constraints that are plotted in the 3-D graphs immediately above them.

can we find: 1) minimal-energy curves that satisfy those constraints; and 2) a deformation from the start curve to the goal curve such that all intermediate curves are also minimal-energy curves and are not colliding with any obstacles. The planner we present below is described in terms of a roadmap-based method [6], but it is not tied to any roadmap construction algorithm. In fact, it could also be used with a tree-based planner [41]–[43]. Various algorithms have been proposed for the construction of roadmaps and trees elsewhere [44], and will not be discussed in this paper. Instead, we will focus on the specifics of the local planner for minimal-energy curves. The general idea of roadmap and tree-based planners is to sample collision-free configurations and connect ones that are close together. This results in a graph representation that compactly represents the free space of a robot. Finding a path between configurations is reduced to finding a path in this graph. In our case, we do not handle collisions at this stage, but we impose the constraint that every configuration corresponds to a minimal-energy curve.

To solve the path-planning problem, we propose the following approach. First, a roadmap of all minimal-energy curves is precomputed in the absence of obstacles. Due to the symmetries that exist in the space of these curves, it suffices to build a roadmap for curves in canonical form. The local planner that connects minimal-energy curves is described below. The second step is to build another roadmap for the environment of interest that may include obstacles. The local planner for this roadmap uses the roadmap of the first stage as a lookup table. It will just need to check whether paths in the first roadmap after applying the alignment transform are collision free. This approach is reminiscent of the planner described in [45]. Whereas Leven and Hutchinson precompute a roadmap in configuration space and modify this roadmap as obstacles are added, we only do this for "shape space." By taking advantage of the symmetries in the configuration space, we can re-use the roadmap for shape space in other parts of the configuration space. The subdivision scheme presented in this paper is efficient enough that computing a roadmap of the shape space is possible.

The problem that the local planner needs to solve can be stated as follows. Given two minimal-energy curves, does there exist a deformation from one curve to another such that all intermediate curves are all minimal-energy curves? The solution we found is very similar to the approach we took in [33]. We find a sequence of minimal-energy curves connecting the start and goal curve such that consecutive curves are at most a distance ε apart [see (1)]. The path planner recursively finds a path as follows. It first computes minimal-energy curves for the start and goal. It then linearly interpolates the curvature and torsion between the two curves to obtain a curve that has distance ε/c,

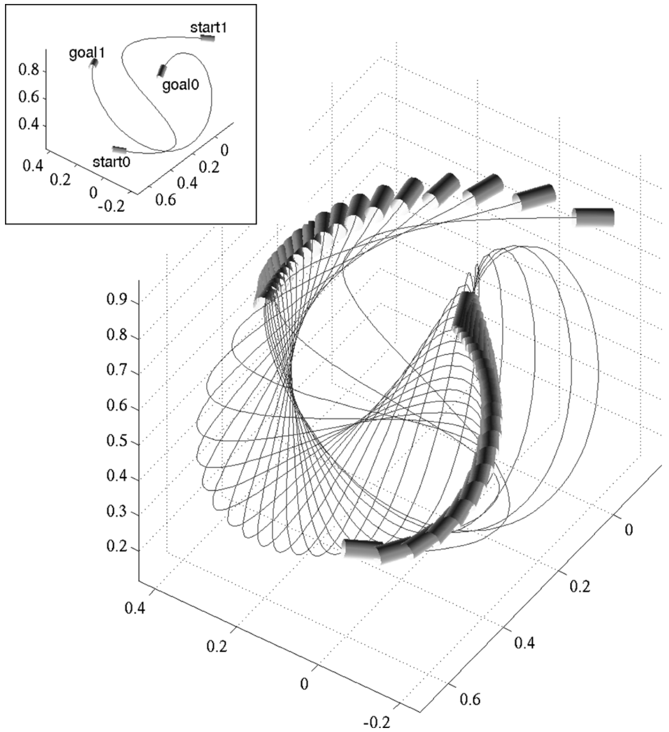

Fig. 5. Path of minimal-energy curves. The inset shows the start and goal curves. The start curve connects start0 and start1, the goal curve connects goal0 and goal1.

$c > 1$, to the start curve. This solution is downsampled to a very coarse resolution and is used as an initial guess for a minimal-energy curve that satisfies the interpolated endpoint constraints. The ability to quickly go from a complex representation to a very coarse one is critical in our path-planning algorithm.

The interpolation scheme for the endpoint constraints is slightly more complicated. A straight-line interpolation between endpoints would not work well, for instance, because this may cause the curve to "fold up" onto itself and cause large shape changes. Instead, we linearly interpolate the *midpoint* $(\boldsymbol{x}_0+\boldsymbol{x}_1)/2$ between the endpoints $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$. We use spherical interpolation to determine the position of the endpoints relative to the mid-point. The tangents are also spherically interpolated. This is done so that the relative change between the endpoints is small, which makes it more likely that the minimal-energy curves connecting successive pairs of interpolated endpoints are close together as well.

Given the interpolated endpoint constraints and the initial guess for the curve parameters, we apply our subdivision scheme to obtain a minimal-energy curve. If the distance between the resulting curve and the starting curve is larger than ε, the path planner fails. Otherwise, we make the new curve the starting curve and recurse. Alternatively, we can recurse by making the new curve the goal curve, and making the old goal curve the new start curve. This way a path is "grown" from both directions. The planner terminates if the distance between the start and goal is less than ε or if some maximum number of iterations is exceeded (in which case the planner fails). The path returned by the planner consists of all the minimal-energy curves generated.

Fig. 5 shows an example of a path as found by our path planner. Fig. 6 shows the curvature and torsion of the minimal-energy curves that constitute the path. From this figure it is clear that the planner is "well-behaved": The change in shape along each curve is smooth, as is the change in shape along the path.

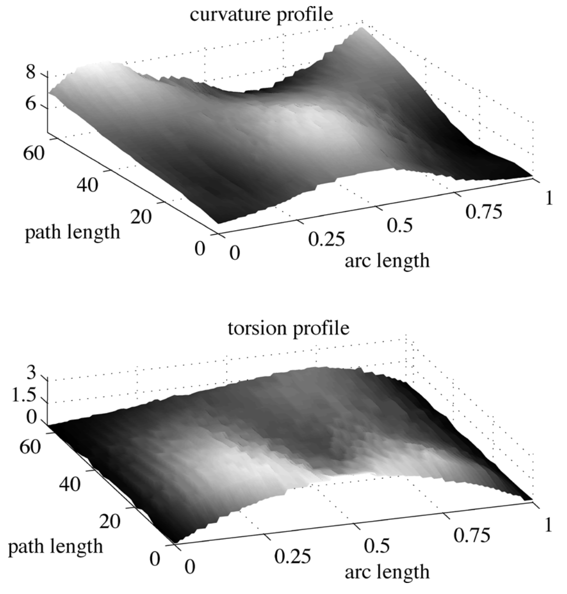

Fig. 6. Curvature and torsion along a path of minimal-energy curves.

VIII. Multiple Control Points

So far, we have assumed that the only control points and tangents that a minimal-energy curve needs to pass through are at the endpoints. In practice, a curve may collide with obstacles in the environment or with itself. We would like to model the constraints imposed by the obstacles as well. Solving for the contact points such that the curve is at an energy minimum is extremely difficult in general. To make the problem more tractable, we will assume that contact points are given as well as the tangents at those points. We can think of this as a curve passing through a number of cylinders. This is also a useful abstraction if we are trying to route cables through a number of rings.

To find a minimal-energy configuration, we solve for each curve segment between consecutive control points separately while maintaining the global length constraint. Initially, we allocate to each segment a length of the curve proportional to the workspace distance between the endpoints of the segment. The workspace distance between control point i and $i+1$ is defined as

$$d_w(i, i+1) = \|\boldsymbol{p}_i - \boldsymbol{p}_{i+1}\| + \arccos(\boldsymbol{t}_i \cdot \boldsymbol{t}_{i+1})$$

where $\boldsymbol{p}_i$ and $\boldsymbol{t}_i$ specify the position and tangent of control point i. If we think of tangents as points on a sphere, then the distance between tangents corresponds to the length of the shortest geodesic on the sphere connecting two tangents. So, the work space distance is simply the sum of the distance between the positions and the distance between the tangents. This distance is only used as a heuristic to pick initial guesses for the curve lengths between subsequent control points and start the energy minimization. As the linear distance increases, the difference in tangents matters less, since the curve will not have to make any sharp turns. This idea is captured by this heuristic.

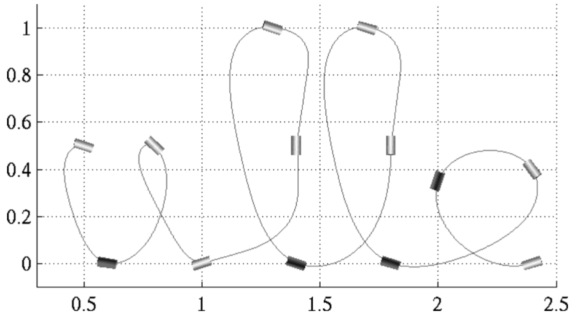

Fig. 7. Minimal-energy curve of length 9 passing through 13 control points and tangents to spell the word "cello."

After we have found initial guesses for the lengths needed to connect consecutive control points, we solve each minimal-energy curve segment separately. The energy of the whole curve is simply the sum of the energies of the curve segments. Suppose we have n curve segments and the lengths of the segments are given by $l_1, \ldots, l_n$. Then, we can further minimize the energy of the curve by varying the initial guesses for $l_1, \ldots, l_{n-1}$ (note that $l_n = L - \sum_{i=1}^{n-1} l_i$). We use a general constrained optimization technique for this. The constraints arise due to upper and lower bounds on the l_i's. A lower bound for l_i is the Cartesian distance between the corresponding control points, since a curve segment needs to be long enough to connect the control points. An upper bound for l_i is obtained by subtracting the lower bound for all l_j's $(j \neq i)$ from L. In other words, we cannot use a curve length for the ith segment that would make it impossible to satisfy the lower bounds on the other segments. The energy minimization will not necessarily find a global minimum, but in our simulations, it produced good results. Fig. 7 shows a minimal-energy curve of fixed length connecting 13 control points. The control points are drawn as small cylinders to emphasize that the curve also needs to match the tangents at those points. Our approach works in 3-D; the curve in Fig. 7 is planar only because it is easier to visualize.

IX. Notes on the Implementation

The subdivision scheme and the path planner described in this paper have been in implemented in C++. We also implemented Matlab bindings so that almost all functionality in the C++ classes can be accessed from Matlab. For energy minimization, we made use of a nonlinear optimization library called OPT++ [46]. In particular, in the subdivision step, we used a quasi-Newton method with a BFGS approximation to the Hessian, and with numerically computed derivatives [40]. In the optimization of curve segment lengths, we used a derivative-free parallel direct search [47].

We evaluated the performance of the subdivision scheme by randomly selecting constraints for the endpoints and timing how long it takes to compute the corresponding minimal-energy curve. The positions were picked uniformly at random within a unit ball, and the tangents were picked uniformly at random as points on a unit sphere. The curve length was set to be 2, the branching factor was 2, the subdivision tolerance was 0.001, the energy minimization tolerance was set to 10^{-8}, and the minimum segment length was set to 0.002. The implementation uses the local refinement rules described in Section VI, but not the global refinement techniques. We generated 50 000 random curves and computed the following statistics:

	time (s)	**error**	**energy**	**#segments**
mean	0.114	8.29×10^{-4}	19.9	36.8
median	0.103	6.39×10^{-5}	17.9	33.0
std. dev.	0.0636	0.183	20.7	17.7

The error denotes the error in the endpoint constraints after alignment as described in Section V. These results were obtained on a Linux workstation with an AMD Athlon XP 2600 processor. From the table above, we see that the computation of minimal-energy curves is reasonably fast. Note also that the number of segments needed to represent a minimal-energy curve varies significantly, which shows the benefit of a variable-resolution representation. It helps speed up path planning by using only as many parameters as necessary.

X. Discussion

This paper describes a new approach to path planning for deformable linear objects. Our approach makes it possible to explore the entire space of stable, collision-free configurations. The stable configurations can be represented with minimal-energy curves. We introduced an algorithm to construct such curves very efficiently. The size of the representation adapts automatically to the geometric complexity of the underlying curve. With this representation it is easy to find paths between minimal-energy curves such that all curves along the path are also minimal-energy curves. This work has applications in simulated and automated suturing, cable routing, and hyperredundant/snake robots.

In future work, we plan to explore the following problems. We would like to develop a more complete model for flexible objects in contact with obstacles. The results in Section VIII where we modeled contact points as being fixed in space are a starting point, but even finding the contact points such that a curve is at an energy minimum is very difficult. The location depends on the geometry of the obstacle and on the contact kinematics between the curve and the object.

The model we proposed can easily be extended to a more realistic model of energy. Clearly, giving bending energy and twisting energy different weights by weighting curvature and torsion differently in (3) would not affect the algorithm at all. Wire extension can be incorporated by relaxing the assumption that the segment lengths sum up to a constant length L. In the subdivision algorithm, segment lengths are, in that case, also optimized over to lower the energy. The energy function can then take the following form:

$$\text{energy }(q) = \sum_{i=1}^{n} \left(\kappa_i^2 + \tau_i^2 + (s_i - \bar{s}_i)^2\right)$$

where $\bar{s}_i$ is the natural, unextended length of segment i, and $s_i \geq \bar{s}_i$. Adding gravity is also possible, but this would remove some of the symmetries that we exploited in our canonical form. There is still translational symmetry and rotational symmetry about the z axis (assuming gravity acts along the z axis). To include gravity in the energy function, it is necessary to integrate the potential energy along the curve. Due to our simple parametrization, this integral can be simplified to a summation.

In conclusion, we have presented a general framework for path planning for deformable linear objects. It has few dependencies on the energy model used. This framework can be used to build up a roadmap of the entire shape space formed by all minimal-energy curves, which is something that was not possible with previously proposed methods.

ACKNOWLEDGMENT

The authors would like to thank Prof. J. Warren and Prof. L. Guibas for useful discussions on minimal-energy curves.

REFERENCES

[1] H. Chang and T. Y. Li, "Assembly maintainability study with motion planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1995, pp. 1012–1019.

[2] R. Wilson and J.-C. Latombe, "Geometric reasoning about mechanical assembly," *Artif. Intell.*, vol. 71, pp. 371–396, 1995.

[3] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu*, The Finite Element Method: Its Basis and Fundamentals*, 6th ed. New York: Butterworth-Heinemann, 2005.

[4] J. Canny*, The Complexity of Robot Motion Planning*. Cambridge, MA: MIT Press, 1987.

[5] J. Warren and H. Weimer*, Subdivision Methods for Geometric Design: A Constructive Approach*. San Mateo, CA: Morgan Kaufmann, 2001.

[6] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[7] H. P. Moreton and C. H. Séquin, "Function optimization for fair surface design," in *Proc. SIGGRAPH Conf.*, 1992, pp. 167–176.

[8] R. C. Veltkamp and W. Wesselink, "Modeling 3D curves of minimal energy," in *Proc. Computer Graphics Forum*, 1995, vol. 14, no. 3, pp. 97–110.

[9] W. Wesselink and R. C. Veltkamp, "Interactive design of constrained variational curves," *Comput. Aided Geom. Des.*, vol. 12, no. 5, pp. 533–546, 1995.

[10] E. Jou and W. Han, "Minimal-energy splines with various end constraints," in *Curve and Surface Design*, H. Hagen, Ed. Philadelphia, PA: SIAM, 1992, pp. 23–40.

[11] G. H. Brunnett, "Properties of minimal-energy splines," in *Curve and Surface Design*, H. Hagen, Ed. Philadelphia, PA: SIAM, 1992, pp. 3–22.

[12] B. K. P. Horn, "The curve of least energy," *ACM Trans. Math. Softw.*, vol. 9, no. 4, pp. 441–460, Dec. 1983.

[13] M. Kallay, "Plane curves of minimal energy," *ACM Trans. Math. Softw.*, vol. 12, no. 3, pp. 219–222, 1986.

[14] ——, "Method to approximate the space curve of least energy and prescribed length," *Comput.-Aided Des.*, vol. 19, no. 2, pp. 73–76, 1987.

[15] H. Wakamatsu and S. Hirai, "Static modeling of linear object deformation based on differential geometry," *Int. J. Robot. Res.*, vol. 23, no. 3, pp. 293–311, Mar. 2004.

[16] H. Wakamatsu, K. Takahashi, and S. Hirai, "Dynamic modeling of linear object deformation based on differential geometry coordinates," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 1040–1045.

[17] J. Phillips, A. Ladd, and L. E. Kavraki, "Simulated knot tying," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, DC, 2002, pp. 841–846.

[18] J. Brown, J.-C. Latombe, and K. Montgomery, "Real-time knot tying simulation," *Vis. Comput.*, vol. 20, no. 2–3, pp. 165–179, 2004.

[19] D. K. Pai, "STRANDS: Interactive simulation of thin solids using Cosserat models," presented at the Computer Graphics Forum, 2002.

[20] A. Remde and D. Henrich, "Direct and inverse simulation of deformable linear objects," in *Robot Manipulation of Deformable Objects*, ser. Advanced Manufacturing, D. Henrich and H. Wörn, Eds. New York: Springer-Verlag, 2000, pp. 43–70.

[21] J. E. Hopcroft and J. K. Kearney, "A case study of flexible object manipulation," *Int. J. Robot. Res.*, vol. 10, no. 1, pp. 41–50, 1991.

[22] H. Wakamatsu, A. Tsumaya, E. Arai, and S. Hirai, "Manipulation planning for knotting/unknotting and tightly tying of deformable linear objects," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 2516–2521.

[23] F. Lamiraux and L. E. Kavraki, "Planning paths for elastic objects under manipulation constraints," *Int. J. Robot. Res.*, vol. 20, no. 3, pp. 188–208, 2001.

[24] O. B. Bayazit, J.-M. Lien, and N. M. Amato, "Probabilistic roadmap motion planning for deformable objects," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, vol. 2, pp. 2126–2133.

[25] R. Gayle, W. Segars, M. Lin, and D. Manocha, "Path planning for deformable robots in complex environments," presented at the Robot.: Sci. Syst. Conf., 2005.

[26] M. Saha and P. Isto, "Motion planning for robotic manipulation of deformable linear objects," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, pp. 2478–2484.

[27] J. Acker and D. Henrich, "Manipulation of deformable linear objects: from geometric model towards program generation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 1553–1559.

[28] A. M. Ladd and L. E. Kavraki, "Using motion planning for knot untangling," *Int. J. Robot. Res.*, vol. 23, no. 7–8, pp. 797–808, 2004.

[29] G. S. Chirikjian and J. W. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Trans. Robot. Autom.*, vol. 10, no. 3, pp. 343–354, Jun. 1994.

[30] K. E. Zanganeh and J. Angeles, "The inverse kinematics of hyper-redundant manipulators using splines," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1995, pp. 2797–2802.

[31] Y. Nakamura and H. Hanafusa, "Optimal redundancy control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 1, pp. 32–42, 1987.

[32] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mischenko*, The Mathematical Theory of Optimal Processes*. New York: Wiley, 1962.

[33] M. Moll and L. E. Kavraki, "Path planning for minimal energy curves of constant length," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, pp. 2826–2831.

[34] ——, "Path planning for variable resolution minimal-energy curves of constant length," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 2142–2147.

[35] A. Gray*, Modern Differential Geometry of Curves and Surfaces with Mathematica*, 2nd ed. Boca Raton, FL: CRC, 1997.

[36] B. O'Neill*, Elementary Differential Geometry*, 2nd ed. San Diego, CA: Academic, 1997.

[37] A. E. H. Love*, A Treatise on the Mathematical Theory of Elasticity*, 4th ed. New York: Dover, 1987.

[38] G. Strang, "Wavelets and dilation equations: A brief introduction," *SIAM Rev.*, vol. 31, pp. 613–627, 1989.

[39] R. M. Murray, Z. Li, and S. S. Sastry*, A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC, 1994.

[40] J. Dennis and R. Schnabel*, Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[41] D. Hsu, J.-C. Latombe, and R. Motwani, "Path planning in expansive configuration spaces," *Int. J. Comput. Geom. Appl.*, vol. 9, no. 4–5, pp. 495–512, 1999.

[42] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, May 2001.

[43] A. Ladd and L. E. Kavraki, "Motion planning in the presenece of drift, underactuation, and discrete system changes," in *Robotics: Science and Systems I*. Boston, MA: MIT Press, 2005, pp. 233–241.

[44] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun*, Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA: MIT Press, 2005.

[45] P. Leven and S. Hutchinson, "Toward real-time path planning in changing environments," in *Algorithmic and Computational Robotics: New Directions*. New York: A. K. Peters, 2001.

[46] J. C. Meza, "OPT++: An object-oriented class library for nonlinear optimization," Sandia Nat. Lab., Tech. Rep. SAND94-8225, 1994 [Online]. Available: http://csmr.ca.sandia.gov/projects/opt++

[47] J. E. Dennis, Jr. and V. Torczon, "Direct search methods on parallel machines," *SIAM J. Optim.*, vol. 1, no. 4, pp. 448–474, Nov. 1991.

Mark Moll (M'99) received the M.S. degree in computer science from the University of Twente, Twente, The Netherlands, in 1995, and the Ph.D. degree in computer science from Carnegie Mellon University, Pittsburgh, PA, in 2002. His thesis work focused on shape reconstruction of unknown shapes using tactile data.

He is currently a Research Scientist with the Information Sciences Institute (ISI), Marina del Rey, CA. His research interests include self-reconfigurable robots, motion planning, robotic manipulation, computational geometry, shape reconstruction, tactile sensing, parts orienting, and computational biology.

Lydia E. Kavraki (M'99) received the B.A. degree in computer science from the University of Crete, Crete, Greece, and the Ph.D. degree in computer science from Stanford University, Stanford, CA.

She is the Noah Harding Professor of Computer Science and Bioengineering at Rice University, Houston, TX. Her research is in physical algorithms and their applications to robotics and bioinformatics. Her recent work focuses on the development of methods for robot planning in high dimensions and with physical constraints, planning with sensor nets, assembly planning, micromanipulation using microelectromechanical systems, and flexible object manipulation. She also applies robotics methods to computational structural biology.

Dr. Kavraki is a member of ACM and a Fellow of the American Institute for Medical and Biological Engineering. Her awards include the Grace Murray Hopper Award from the Association for Computing Machinery, a Sloan Fellowship, and the Early Academic Career Award from the IEEE Robotics and Automation Society.