

# Path Planning for Minimal Energy Curves of Constant Length

Mark Moll

mmoll@cs.rice.edu

Lydia E. Kavraki

kavraki@cs.rice.edu

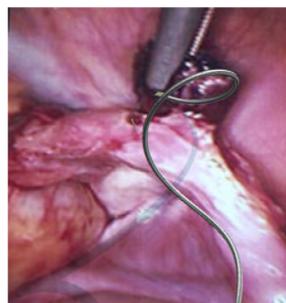
Department of Computer Science, Rice University, Houston, TX 77005

**Abstract**—In this paper we present a new path planning technique for a flexible wire. We first introduce a new parametrization designed to represent low-energy configurations. Based on this parametrization we can find curves that satisfy endpoint constraints. Next, we present three different techniques for minimizing energy within the self-motion manifold of the curve. We introduce a local planner to find smooth minimal energy deformations for these curves that can be used by a general path planning algorithm. Using a simplified model for obstacles, we can find minimal energy curves of fixed length that pass through specified tangents at given control points. Finally, we show that the parametrization introduced in this paper is a good approximation of true minimal energy curves. Our work has applications in surgical suturing and snake-like robots.

## I. INTRODUCTION

The path planning problem consists of finding a collision free path of configurations that connect a start and a goal configuration. This problem is known to be *PSPACE*-hard [1] and, hence, no efficient algorithm exists to solve it in general. However, there are several algorithms that have been shown to be very practical in solving path planning problems for robots with many degrees of freedom (DOFs) [2]–[5]. In recent years some progress has been made on path planning for *flexible* robots or objects [6]–[9], which is the topic of this paper. This problem is particularly challenging since there are potentially infinitely many DOFs. The complexity is further increased if we impose the geometric constraint that the length, surface area, or volume needs to be constant. Usually flexibility is considered a local deformation. But the length of a curve is a global property. It is non-trivial to satisfy this global constraint with local changes to the shape. Finally, we would like to impose energetic constraints. In particular, we restrict configurations to be at energy minima, so that it becomes easier to follow a trajectory. In this paper we describe techniques for finding minimal energy curves of constant length and paths consisting of such curves. An important aspect of the planning problem is the representation of the configuration space. The representation should be powerful enough to capture all feasible minimal-energy shapes and at the same time be small enough that we can actually solve planning problems.

There are many applications where we need to model flexibility, such as robots handling sheet metal or paper, pulling cables, and manipulating surgical sutures. Hyperredundant snake-like robots can (for path planning purposes) also be modeled as being flexible (see figure 1). In this paper we will



(a) A simulated surgical suture



(b) A snake robot

Fig. 1. Different application domains for minimal energy curves. Images courtesy of D. Pai and H. Choset, respectively.

focus on ‘flexible’ spatial curves of constant length. The curves represent the shape of, e.g., a suture or a snake robot.

The outline of the rest of the paper is as follows. Section III and IV describe a method for finding minimal energy curves with given endpoint constraints by first solving the geometric constraints and then the physical constraints. This method depends on a new compact curve representation introduced in the next section. Section V describes path planning for minimal curves insofar as it is different from the general path planning problem. In section VI we describe how we can extend the results from the previous sections to more than two control points. Section VII presents some results that indicate our parametrization is in fact a good approximation of true minimal energy curves (in the variational sense). Finally, we discuss our results and outline directions for future research.

## II. CURVE PARAMETRIZATION

When planning paths for, say, a suture or a snake robot we favor configurations with minimal strain. The main reason we focus on minimal strain curves is that plans consisting of only such configurations do not rely on dynamics and will be easier to execute. We assume that a straight line segment without torsion represents the shape with zero strain. The Darboux vector, defined in terms of the Frenet frame as  $D = \tau T + \kappa B$ , describes the rotational strain along the curve. Here  $T$  and  $B$  are the tangent and binormal, respectively, and  $\tau$  and  $\kappa$  denote the torsion and curvature. We assume there is no translational strain: the suture or robot 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. 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 rope 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 rather difficult [6] because we need many DOFs. Finding a smooth curve that satisfies endpoint constraints is difficult and finding minimum energy curves using a finite element method is even more challenging. We therefore need a novel parametrization. Below we will introduce our parametrization step by step, starting with a planar curve and building up towards a 10 DOF parametrization for spatial curves.

Without loss of generality we can assume that a minimal energy curve has length 1 and that one of the endpoints is at the origin with its tangent along the positive x-axis. If we parametrize a planar curve by turning angle the length constraint is automatically satisfied:

$$\mathbf{x}(\sigma) = \int_0^\sigma \begin{pmatrix} \cos \theta(s) \\ \sin \theta(s) \end{pmatrix} ds, \quad 0 \leq \sigma \leq 1,$$

where  $\mathbf{x}$  is the curve parametrized by arc length and  $\theta(\cdot)$  is some arbitrary smooth function. The energy of the curve is then given by

$$V = \int_0^1 \kappa^2(s) ds = \int_0^1 (\theta'(s))^2 ds.$$

(Note that for planar curves the torsion is always equal to 0.) We would like to find a parametrization for  $\theta(\cdot)$  such that it is easy to find a solution for given endpoint constraints. Also, the parametrization should result in curves that tend to be ‘‘close’’ to minimal energy curves. Below we will describe in more detail how one can minimize the energy of a curve (sec. IV) and how we can check the validity of a given parametrization (sec. VII). The last issue is a subtle but very important point. When we minimize energy of a curve, we do so with respect to a given parametrization. It is not guaranteed that such a curve is close to minimizing the energy in a variational sense.

Consider the following parametrization of  $\theta(\cdot)$  as a starting point:

$$\theta(s) = r(\sin(2\pi s - \delta) + \sin \delta).$$

The tangent at both endpoints ( $s = 0$  and  $s = 1$ ) is equal to  $(1, 0)^T$ , regardless of the values of the parameters  $r$  and  $\delta$ . The position of endpoint 0 is, by assumption, always at the origin. The position of endpoint 1 is given by [10]:

$$\mathbf{x}_1 = J_0(r) \begin{pmatrix} \cos(r \sin \delta) \\ \sin(r \sin \delta) \end{pmatrix},$$

where  $J_0(\cdot)$  is the zero-order Bessel function of the first kind [11]. Figure 2 shows some curves resulting from random values for  $r$  and  $\delta$ . This function can only be computed numerically. The ‘inverse kinematics’, i.e., the mapping from

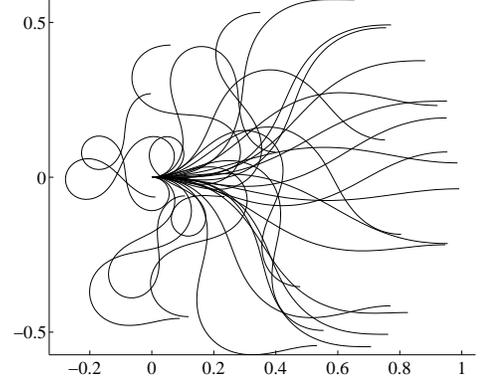


Fig. 2. Planar curves resulting from  $r$  and  $\delta$  drawn from a Gaussian distribution with zero mean and variance equal to 1.

$\mathbf{x}_1$  to  $(r, \delta)$ , is also solved for numerically. The energy is simply given by  $2\pi^2 r^2$ .

Despite the appealing simplicity of the parametrization and the smooth resulting curves, the above parametrization is flawed for our purposes: it does not give us any control over the tangent at endpoint 1. This can easily be fixed by adding an extra term to the previous parametrization:

$$\theta(s) = r(\sin(2\pi s - \delta) + \sin \delta) + \alpha s,$$

where  $\alpha$  is the desired angle of the tangent of endpoint 1. Although this seems like a very simple change, now no analytic solution exists for either endpoint 1 or the energy. Ignoring the effect of  $r$  and  $\delta$ , the  $\alpha$  parameter causes the tangent to rotate at a constant rate from the initial direction to the final direction. In general, this will result in low energy curves. There are, however, minimal energy curves where the curvature is more concentrated at the endpoints. Consider the case where the endpoints are almost at distance 1 apart. To meet the endpoints the curve may need to make a sharp turn at each end. To accommodate for this we introduce two new parameters,  $u$  and  $v$ , that can be used to control how much the curve turns at endpoint 0 and endpoint 1, respectively:

$$\theta(s) = r(\sin(2\pi s - \delta) + \sin \delta) + \frac{\alpha}{2}(1 - (1 - s)e^{-su^2} + se^{(s-1)v^2}).$$

The term containing  $u$  is a smooth concave function, where  $u$  controls how concave this function is. Analogously, the term containing  $v$  is a smooth convex function. Note that if  $u = v = 0$ , the behavior is the same as before. We can now describe a planar curve with a 5-dimensional vector  $\mathbf{q} = (r, \delta, u, v, \alpha)^T$ . The position and tangent endpoint constraints remove three DOFs. The remaining two DOFs can be used to minimize the energy of the curve.

For spatial curves we can parametrize the tangent in spherical coordinates using two angles:

$$\mathbf{x}(\sigma) = \int_0^\sigma \begin{pmatrix} \cos \theta(s) \\ \sin \theta(s) \cos \phi(s) \\ \sin \theta(s) \sin \phi(s) \end{pmatrix} ds, \quad 0 \leq \sigma \leq 1,$$

where

$$\begin{aligned}\theta(s) &= r_1 (\sin(2\pi s - \delta_1) + \sin(\delta_1)) \\ &\quad + \frac{\alpha}{2} (1 - (1-s)e^{-su_1^2} + se^{(s-1)v_1^2}) \\ \phi(s) &= r_2 (\sin(2\pi s - \delta_2) + \sin(\delta_2)) \\ &\quad + \frac{\beta}{2} (1 - (1-s)e^{-su_2^2} + se^{(s-1)v_2^2}).\end{aligned}$$

So spatial curves can be described using a 10-dimensional vector  $\mathbf{q} = (r_1, \delta_1, u_1, v_1, r_2, \delta_2, u_2, v_2, \alpha, \beta)^T$ . The endpoint constraint now removes 5 DOFs, leaving the remaining 5 DOFs for energy minimization.

Let  $\mathbf{t}(s)$  be the above integrand / tangent vector. The derivative of  $\mathbf{t}(s)$  with respect to  $s$  is given by

$$\mathbf{t}' = \theta' \begin{pmatrix} -\sin\theta \\ \cos\theta \cos\phi \\ \cos\theta \sin\phi \end{pmatrix} + \phi' \begin{pmatrix} 0 \\ -\sin\theta \sin\phi \\ \sin\theta \cos\phi \end{pmatrix}.$$

For convenience we dropped the argument  $s$ . The curvature function of  $\mathbf{x}$  is therefore given by

$$\kappa^2(s) = \mathbf{t}'(s) \cdot \mathbf{t}'(s) = (\theta'(s))^2 + \sin^2\theta(s) (\phi'(s))^2.$$

Similarly, it can be shown that the torsion along the curve is given by

$$\begin{aligned}\tau(s) &= \left( \phi'(s) \cos\theta(s) (\kappa^2(s) + (\theta'(s))^2) \right. \\ &\quad \left. + \sin\theta(s) (\theta'(s)\phi''(s) - \phi'(s)\theta''(s)) \right) / \kappa^2(s).\end{aligned}$$

The energy of a curve  $\mathbf{q}$ , denoted by  $V(\mathbf{q})$ , is defined as  $\int_0^1 \kappa(s)^2 + \tau(s)^2 ds$ . So we have an analytic expression for the integrand of the energy function. With a variable step size integration method the energy can be computed fairly efficiently, especially since the curves we are interested in tend to be very smooth.

### III. FINDING GEOMETRICALLY FEASIBLE CURVES

Before we turn to energy minimization, we first have to describe a method for finding a curve that satisfies given endpoint constraints. In robotics terms this means solving the inverse kinematics. The forward kinematics are described by a rather complicated integral for which no closed form expression exists. Nevertheless, the curves tend to be very smooth for a wide range of the parameters, and a variable step numerical integration will find the endpoint with only a small number of steps. We can immediately solve for the tangent at endpoint 1 by setting  $\alpha$  and  $\beta$  to the desired values. We need to find values for the remaining 8 parameters to satisfy the constraints on the endpoint position. Let the *forward kinematics* of a curve be defined as  $\mathbf{f}(\mathbf{q}) = \mathbf{x}(1; \mathbf{q})$ , that is,  $\mathbf{f}$  returns endpoint 1 of the curve with parameters given by  $\mathbf{q}$ . Suppose we happen to guess a configuration  $\mathbf{q}$  such that the endpoint position  $\mathbf{f}(\mathbf{q})$  is close to the desired position  $\mathbf{x}_d$ . We can then repeatedly use the pseudo-inverse of the Jacobian  $J = \partial \mathbf{f} / \partial \mathbf{q}$  to get closer to the desired position:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + K J^\dagger(\mathbf{q}_k) (\mathbf{x}_d - \mathbf{f}(\mathbf{q}_k)),$$

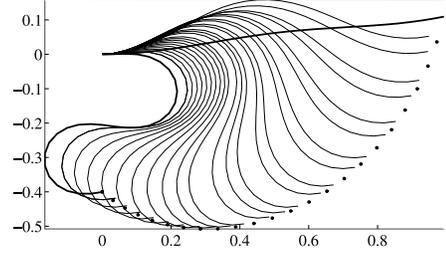


Fig. 3. Finding geometrically feasible curves. The almost straight curve on the right is the initial guess. The ‘Z’ shaped curve on the left is the final curve that matches the desired endpoint position.

for some constant  $K$ . This approach will not work in general. We can improve this method by introducing intermediate target positions. By repeatedly applying the above technique we can move the curve from the current position toward the next one until we reach the desired final position. The intermediate positions are chosen by linearly interpolating the start and goal in spherical coordinates. We chose to interpolate in spherical coordinates rather than Cartesian coordinates, because this method tends to avoid configurations where the curve is completely folded. Figure 3 shows an example. The target positions are marked by dots. Note that the intermediate configurations never quite reach these positions. The target positions are only there to guide the search to the goal position, but it is unnecessary (and undesirable, from a computational point of view) to match them *exactly*. The path formed by the curves is not yet a feasible path, because the curves are not necessarily minimal energy curves.

### IV. FINDING ENERGETICALLY FEASIBLE CURVES

In the previous section we presented a method for finding a curve of fixed length that matches certain endpoint constraints. We would like to take these curves as a starting point for energy minimization subject to the same endpoint constraints (otherwise the result of minimization would inevitably be a straight line). The set of configurations that satisfy given endpoint constraints form the so-called self-motion manifold. There are at least three different techniques for finding a minimal energy configuration in this set:

- Simulate the Lagrangian dynamics. The partial derivatives of the energy can be thought of as forces pulling the curve toward a minimal energy configuration.
- Sample random configurations in the null space of the Jacobian and use the pseudo-inverse technique from the previous section to satisfy endpoint constraints.
- Use a general purpose constrained optimization technique.

Below we will describe these techniques in more detail.

a) *Lagrangian Dynamics*: Let the Lagrangian for our system be defined as  $L(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{q}} - V(\mathbf{q})$ . The force acting on the curve is  $F(\mathbf{q}) = \frac{\partial V}{\partial \mathbf{q}}$ . The Lagrangian dynamics can then be written as [12]:

$$\frac{\partial L}{\partial \dot{\mathbf{q}}} - \frac{\partial L}{\partial \mathbf{q}} + J^T \lambda - F = \mathbf{0}, \quad \lambda = (J J^T)^{-1} (-J F + \dot{\mathbf{j}} \dot{\mathbf{q}}).$$

Here we have dropped the argument  $\mathbf{q}$  of  $J$  and  $F$ . This equation simplifies to

$$\ddot{\mathbf{q}} = -F - J^T \lambda.$$

By integrating this system of ordinary differential equations, we can find a minimal energy configuration. It will be convenient to add an extra damping term to the dynamics, so that the configuration does not oscillate too much:

$$\ddot{\mathbf{q}} = -F - J^T \lambda - k\dot{\mathbf{q}}.$$

This simple ODE is easy to solve numerically. Once both the acceleration and the velocity are below a certain threshold, we conclude that the curve is in a minimal energy configuration.

*b) Null Space Sampling:* If the energy landscape has many local minima, then methods relying on gradient information can easily get stuck. The null space of the Jacobian describes the tangent space of the self-motion manifold: instantaneously, configuration displacements that lie in the null space will not change the endpoint position. By sampling in the null space around the current configuration, we can obtain configurations whose endpoints are close to the desired endpoint. Each sample needs to be ‘pulled back’ to the desired endpoint position using the pseudo-inverse of the Jacobian.

*c) Constrained Optimization:* With the first solution technique the search direction is dictated by the dynamics, but it is possible that we can find a minimum faster if we can search in *any* direction. One way to solve a constrained optimization problem is to relax the problem to an unconstrained problem and use penalty methods to enforce the constraints. This approach is now considered rather inefficient. Instead, current optimization methods focus on solving the so-called Karush-Kuhn-Tucker (KKT) equations, which state necessary conditions for optimality for a constrained optimization problem [13]. Constrained quasi-Newton methods can guarantee super-linear convergence by accumulating second order information regarding the KKT equations using a quasi-Newton updating procedure. These methods are often referred to as Sequential Quadratic Programming methods, since at each iteration a quadratic programming problem is solved.

None of the above methods is clearly superior to the others. One difficulty in minimizing energy is that the energy is a numerically computed integral over a very nonlinear function. As a result the gradients of the energy function are often rather inaccurate. Also, different parametrizations may favor different methods. We have implemented all three methods. In our simulations we found the last method to give the best trade-off between efficiency and accuracy.

## V. MOTION PLANNING FOR MINIMAL ENERGY CURVES

The motion planning problem for minimal energy curves of constant length can be stated as follows: given two minimal energy curves, does there exist a smooth deformation from one curve to the other such that the intermediate curves are also minimal energy curves? In this paper we will ignore the possibility that there may be obstacles. (Collision detection is an independent problem and can be dealt with separately.) A

straight line path in the configuration space generally does not correspond to a set of minimal energy curves, so we need a different way to compute a path between two configurations. The following algorithm computes a path between two configurations  $\mathbf{q}_0$  and  $\mathbf{q}_1$ :

---

**Algorithm 1** compute\_path( $\mathbf{q}_0, \mathbf{q}_1$ )

---

**Input:** start configuration  $\mathbf{q}_0$ , goal configuration  $\mathbf{q}_1$

**Output:** a sequence of minimal energy curves connecting  $\mathbf{q}_0$  and  $\mathbf{q}_1$

**if**  $\|\mathbf{q}_0 - \mathbf{q}_1\| < \varepsilon$  **then**

    return  $[\mathbf{q}_0, \mathbf{q}_1]$

**else**

$\mathbf{q}_2 = (1 - \varepsilon)\mathbf{q}_0 + \varepsilon\mathbf{q}_1$

$\mathbf{q}_2^* = \text{minimize\_energy}(\mathbf{q}_2)$

**if**  $\|\mathbf{q}_2^* - \mathbf{q}_1\| > \|\mathbf{q}_0 - \mathbf{q}_1\|$  or  $\|\mathbf{q}_2^* - \mathbf{q}_0\| > 2\varepsilon$  **then**  
         return  $\emptyset$

**else**

        return  $[\mathbf{q}_0, \text{compute\_path}(\mathbf{q}_2^*, \mathbf{q}_1)]$

**end if**

**end if**

---

If the distance between  $\mathbf{q}_0$  and  $\mathbf{q}_1$  is smaller than some small constant  $\varepsilon$ , we assume such a path exists. Otherwise, let  $\mathbf{q}_2$  be a configuration on the straight-line interpolation between  $\mathbf{q}_0$  and  $\mathbf{q}_1$  at a distance  $\varepsilon$  away from  $\mathbf{q}_0$ . Let  $\mathbf{q}_2^*$  be the configuration resulting from minimizing the energy of  $\mathbf{q}_2$ . If the distance between  $\mathbf{q}_2^*$  and  $\mathbf{q}_1$  is greater than the distance between  $\mathbf{q}_0$  and  $\mathbf{q}_1$ , the connection attempt fails: with each step we would like to guarantee progress toward the goal. If  $\mathbf{q}_2^*$  is more than  $2\varepsilon$  away from  $\mathbf{q}_0$ , the connection attempt also fails. In this case, we can no longer guarantee that a minimal energy path exists between  $\mathbf{q}_0$  and  $\mathbf{q}_2^*$ . If the connection attempt did not fail, we let  $\mathbf{q}_0 = \mathbf{q}_2^*$  and recurse.

A higher level motion planner (such as roadmap based methods [2] or tree expansion algorithms [3], [4]) can treat this technique for connecting configurations as a basic primitive for doing local planning. The output of such planners generally consists of a sequence of configurations that connect a start and goal configuration. Usually these paths are not very optimal and some postprocessing is performed. Different problems require different optimizations. Often we want to minimize path length, but sometimes smoothness of the path is important as well. For a sequence of minimal energy curves we may wish to minimize the total energy of the path, that is, the integral of the curve energy along the path. Rather than directly minimizing this integral, we will minimize a weighted sum that approximates this integral. Let  $\mathbf{q}_1, \dots, \mathbf{q}_n$  be a path as returned by a planner. Let the cost of the edge connecting  $\mathbf{q}_i$  and  $\mathbf{q}_{i+1}$  be defined as :

$$c(e_{i,i+1}) = d_c(\mathbf{q}_i, \mathbf{q}_{i+1}) (V(\mathbf{q}_i) + V(\mathbf{q}_{i+1})) / 2,$$

where  $d_c$  is a configuration space distance metric. A roadmap-based method can use this function to find low-cost paths in the roadmap and avoid high-energy configurations. After we have

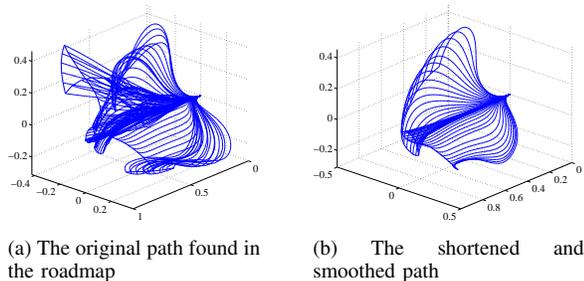


Fig. 4. The effects of shortening and smoothing a path

found a path, there are two ways we can optimize this path. First, we can *shorten* the path by removing a configuration  $q_i$  if we can connect configurations  $q_{i-1}$  and  $q_{i+1}$  directly. We repeat this step until no more configurations can be removed. Second, we can *smooth* the path by following the derivative of the path. For both procedures we need to check if applying a certain change results in a valid path and reduces the total cost of the path. An example of path optimization is shown in figure 4. We implemented a PRM planner for minimal energy curves. The path in the left plot was found using this planner.

## VI. 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 rope may collide with obstacles in the environment. We would like to model the constraints imposed by the obstacles as well. Solving for the contact points in general such that the rope is at an energy minimum is extremely difficult. 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 rope passing through a number of cylinders. To find a minimal energy configuration we solve for each curve segment between two control points separately while maintaining the global length constraint. Initially, we allocate to each segment a length of the curve proportional to the work space distance between the endpoints of the segment. The work space distance between control point  $i$  and  $i + 1$  is defined as

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

where  $\mathbf{p}_i$  and  $\mathbf{t}_i$  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.

After we have found initial guesses for the length needed to connect subsequent control points, we solve each minimal energy curve segment separately. First, we find the transform to bring the problem in ‘canonical form’: one endpoint is at the origin with tangent along the x-axis and the curve has length one. Next, we find the minimal energy curve for the problem in canonical form. Finally, we apply the inverse transform to obtain a minimal energy curve for the original problem.

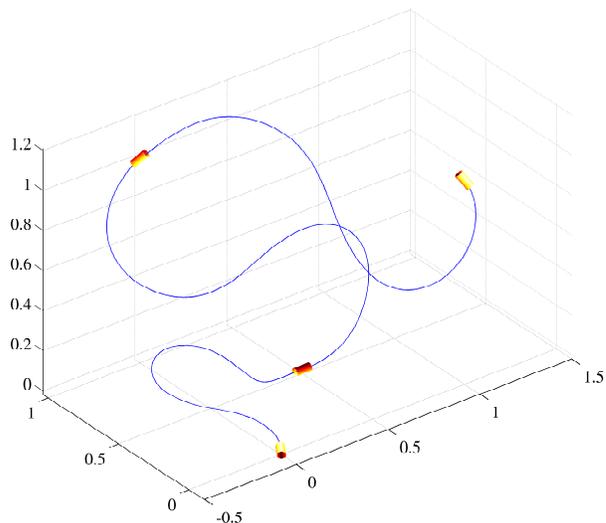


Fig. 5. A curve of length 7 that passes through the following four points:  $(0, 0, 0)$ ,  $(.5, .5, 0)$ ,  $(0, 1, 1)$ , and  $(1, 0, 1)$ . The tangents at these points were specified in spherical coordinates to be equal to  $(1, 0)$ ,  $(0, .5)$ ,  $(-1, -1)$ , and  $(2, 1)$ , respectively.

The energy of the whole curve is simply the sum of the energy of the curve segments. Suppose we have  $n$  curve segments and the lengths of the segments are given by  $l_1, \dots, l_n$ . Then we can further minimize the energy of the curve by varying the initial guesses for  $l_1, \dots, l_n$ . We have used a general constrained optimization technique. It will not necessarily find a global minimum, but in our simulations it produced good results. Figure 5 shows a minimal energy curve of fixed length connecting four control points. The control points are drawn as small cylinders to emphasize that the curve also needs to match the tangents at those points.

## VII. VALIDATION OF THE PARAMETRIZATION

So far we have ignored the difference between parametrized curves that have minimal energy with respect to the parametrization and curves that have minimal energy in a variational sense. To make sure our parametrization is valid, we need to check if a curve that is minimal with respect to the parametrization is very similar to a ‘true’ minimal energy curve. There is no easy way to compute a true minimal energy curve, but we can construct an approximation that can be made arbitrarily close to it. This approximation consists simply of  $n$  segments of piecewise constant curvature and torsion (pcct). Applying the energy minimization to such a curve results in a curve that gets closer to a true minimal curve as  $n$  increases.

Let  $\mathbf{q}_0$  be a minimal energy curve using our parametrization. We can numerically compute the curvature and torsion for  $n$  points along the curve. From this we can construct the pcct curve  $\mathbf{p}_0$ . This curve will have a slightly different endpoint position and tangent, so we find a configuration  $\mathbf{q}_0^*$  that matches the endpoint of  $\mathbf{p}_0$ . We then minimize the energy of both  $\mathbf{p}_0$  and  $\mathbf{q}_0^*$  and compare the resulting curves. Figure 6 shows the energy for 500 random minimal energy curves. The curves are sorted by the energy of the pcct curves, plotted as

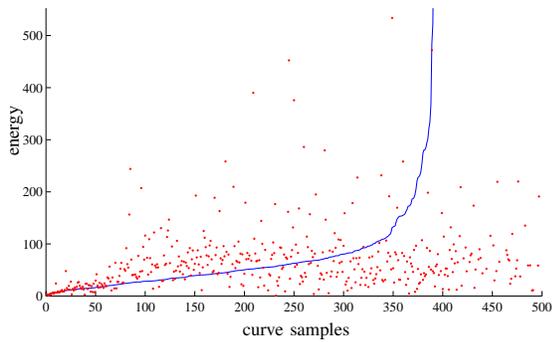


Fig. 6. Comparison of the energy of our parametrization (dots) and pcct curves (solid line).

a solid line. The dots show the energy of the corresponding curve with our parametrization. The number of segments in the pcct curves is equal to 10 so the curves have 20 DOFs. One problem with energy minimization with both types of curves is that there are many local minima. With our parametrization we were able to find slightly lower energy curves by using not just  $q_0$  as our initial estimate for  $q_0^*$  but also 20 completely random curves. The inverse kinematics solver would then sometimes converge to a slightly different curve with lower energy. For the pcct curves this is not practical; it would take a very long time for a random pcct curve to untangle and converge to *any* minimal energy curve, let alone one that satisfies the desired endpoint constraints. From the plot we learn that (1) for low-energy pcct curves, we can find a curve using our parametrization with comparable energy, (2) the energy minimization for pcct often diverges and returns a high-energy curve.

### VIII. DISCUSSION

In this paper we introduced a new curve parametrization designed for minimal energy curves of constant length. Based on this parametrization we have built a path planner for flexible wires that tries to minimize the bending and twisting of the wires. This has applications in simulated and automated suturing, and hyperredundant/continuum 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 VI 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 configuration space of minimal energy curves is still poorly understood. In our simulations we noticed that solving the inverse kinematics with random starting curves, followed by energy minimization would sometimes result in rather different curves. This raises the question whether the configuration space of minimal energy curves has several components. In other words, are there minimal energy curves that cannot be connected by a path of minimal energy curves?

We are also interested in different parametrizations for curves of constant length. We can improve the approximation of true minimal energy curves by adding more parameters, but this needs to be done in a systematic way. One approach we would like to explore is a variable resolution scheme for parametrizing the curve tangent or, alternatively, the curvature and torsion of the curve. Wavelets have been shown to provide a very compact representation of complex curves [14], [15]. Although we cannot apply wavelets directly to the curve itself (the curve length would change), we *can* use them for a curve's derivatives.

### ACKNOWLEDGMENT

Work on this project by Mark Moll is supported in part by NSF 0072743 and 9702288. Lydia Kavraki is supported in part by NSF 0205671, NSF 0308237 and a Sloan Fellowship.

### REFERENCES

- [1] J. Canny, *The Complexity of Robot Motion Planning*. Cambridge, MA: MIT Press, 1987.
- [2] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, Aug. 1996.
- [3] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, May 2001.
- [4] D. Hsu, J.-C. Latombe, and R. Motwani, "Path planning in expansive configuration spaces," *International Journal of Computational Geometry and Applications*, vol. 9, no. 4-5, pp. 495–512, 1999.
- [5] M. Akinc, K. E. Bekris, B. Y. Chen, A. M. Ladd, E. Plakue, and L. E. Kavraki, "Probabilistic roadmaps of trees for parallel computation of multiple query roadmaps," in *Proc. of Eleventh Intl. Symp. on Robotics Research*, Siena, Italy, Oct. 2003.
- [6] F. Lamiroux and L. E. Kavraki, "Planning paths for elastic objects under manipulation constraints," *International Journal of Robotics Research*, vol. 20, no. 3, pp. 188–208, 2001.
- [7] J. Phillips, A. Ladd, and L. E. Kavraki, "Simulated knot tying," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, DC, 2002, pp. 841–846.
- [8] D. K. Pai, "STRANDS: Interactive simulation of thin solids using Cosserat models," *Computer Graphics Forum (Eurographics 2002 Proceedings)*, vol. 21, no. 3, 2002.
- [9] A. M. Ladd and L. E. Kavraki, "Using motion planning for knot untying," *International Journal of Robotics Research*, 2004, to appear.
- [10] G. S. Chirikjian and J. W. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 343–354, 1994.
- [11] F. Bowman, *Introduction to Bessel Functions*. Dover Publications, 1958.
- [12] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, 1994.
- [13] M. Bazarara, D. Sherali, and C. Shetty, *Nonlinear Programming: Theory and Algorithms*. New York: John Wiley & Sons, 1992.
- [14] G. C.-H. Chuang and C.-C. J. Kuo, "Wavelet descriptor of planar curves: Theory and applications," *IEEE Transactions on Image Processing*, vol. 5, no. 1, pp. 56–70, Jan. 1996.
- [15] L.-M. Reissell, "Wavelet multiresolution representation of curves and surfaces," *Graphical Models and Image Processing*, vol. 58, no. 3, pp. 198–217, May 1996.