Appeared in ICRA 2005 Proceedings, pp. 2143–2147

Path Planning for Variable Resolution Minimal-Energy Curves of Constant Length*

Mark Moll and Lydia E. Kavraki
Department of Computer Science, Rice University, Houston, TX 77005
mmoll@cs.rice.edu, kavraki@cs.rice.edu

***Abstract*— We present a new approach to path planning for flexible wires. We introduce a method for computing stable configurations of a wire subject to manipulation constraints. These configurations correspond to minimal-energy curves. The 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. 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 motion planning for surgical suturing and snake-like robots.**

I. Introduction

We are interested in motion planning for flexible objects. Flexibility of an object often means that there is an infinite number of shapes that the object can take on. To plan motions for these objects efficiently we have to approximate their shapes with a finite number of parameters. We also need a model to describe the dynamics of the object given a certain parametrization. Applications of motion planning for flexible objects include suturing simulations, virtual reality simulations, routing of pipes and cables, graphics animation, and modeling the backbone of flexible macromolecules. So far there has been only limited success in developing planners for flexible objects. We are working towards this end. This paper will concentrate on representing and planning for curves of fixed length when given manipulation constraints (see figure 1). The constraints arise from robot grippers holding the endpoints of the wire, thereby fixing the positions and tangents at the endpoints.

*Work on this project by Mark Moll is supported in part by NSF 0205671 and 0203396. Lydia Kavraki is supported in part by NSF 0205671, NSF 0308237, a Sloan Fellowship, and a Whitaker Biomedical Engineering Research Grant.

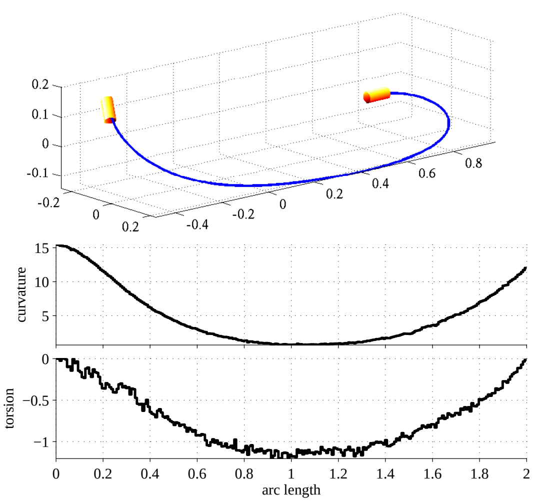

Fig. 1. A minimal-energy curve of length 2. The curve is held at the endpoints, constraining both the positions and the tangents. This is visualized using small cylinders. The bottom two plots show the curvature and torsion along the curve.

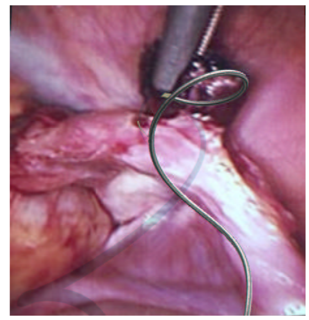

Fig. 2. A simulated surgical suture. *Image courtesy of D. Pai*

The main motivation for our research comes from surgical suturing (see figure 2). 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.

In previous work [1] we presented an approximate representation of minimal-energy curves using only 10 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.

The outline of the rest of the paper is as follows. The next section 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 alignment procedure is explained in section V. In section VI 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 VII. Section VIII describes our implementation and gives some performance results. Finally, section IX summarizes the contributions of this paper and outlines directions for future research.

II. Related Work

Motion planning for flexible objects is a very challenging problem. Lamiraux and Kavraki [2] introduced one of the first 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. [3] 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. Recently, Wakamatsu et al. [4] proposed a manipulation planner for knotting and raveling a rope. This planner has been implemented on a 6 DOF manipulator with a camera. Ladd and Kavraki [5] 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 and by carefully defining intermediate subgoals, they were able to untangle very complex knots. Sometimes hyper-redundant robots (or snake robots) are modeled as flexible curves [6]. In this context minimal-energy curves may provide good reference shapes for the robot that minimize joint movement.

Almost complementary to motion planning for flexible objects is the simulation of flexible objects. Phillips et al. [7] 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. [8] 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 [9] focuses on the internal 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. However, path planning requires inverting the dynamics equations, which is very difficult.

Minimal-energy curves appear in geometric design in the broader context of fair curve and surface design [10]–[14]. Here, 'fair' means minimizing some functional. In our case this is energy (which will be defined more precisely later on). Fair curve design focuses almost exclusively on planar design. The design of fair spatial curves appears to be still an open problem.

III. Minimal-Energy Curves

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 τ and κ 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 suture 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 [2] because we need many DOFs. 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 3D minimal-energy curves. For *planar* minimal-energy curves with endpoint constraints the following variational condition has to be satisfied along the curve: $\kappa''(s) + \frac{1}{2}\kappa^3(s) = c \cdot \kappa(s)$ for some constant c [12]. Such a constraint does not exist for *spatial* minimal-energy curves.

The following two observations will be important in the rest of this paper. First, 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. Second, for a minimal-energy curve, every segment of that curve is also a minimal-energy curve. This means 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. A Subdivision Scheme for Minimal-Energy Curves

Subdivision is an area of geometric modeling concerned with compact representations of curves and surfaces [15]. 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.

We have developed a subdivision scheme for representing minimal-energy curves. There are two factors that make this scheme more complicated than most subdivision schemes. First, 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. Second, 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. 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.

Given manipulator constraints like the endpoints and tangents where a suture is held, we can quickly find a minimal-energy curve that satisfies those constraints. 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. We also take advantage of the symmetries by solving only for minimal-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. Using upsampling and downsampling we can represent a curve at different levels of detail. Curves in this representation can also be compressed really well using, e.g., wavelets [16].

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). 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

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

where

$$\text{energy}(q) = \textstyle\sum_{i=1}^{n} (\kappa_i^2 + \tau_i^2) \cdot s_i, \tag{2}$$

K is a penalty constant, and the error is measured after alignment, as described in the next section. 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.

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. But satisfying the constraints requires at least *five* degrees of freedom: three for the endpoint position and two for the endpoint tangent. We therefore need to subdivide two segments at once, giving us eight degrees of freedom, 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

$$\left((\kappa_{i+1} - \kappa_i)^2 + (\tau_{i+1} - \tau_i)^2\right) \cdot \max(s_i, s_{i+1}). \tag{3}$$

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. 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.

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 (as defined below) is minimized.

First, we scale the curve in canonical form to have the desired length. Next, we translate and rotate the curve such that the Euclidean distance between the endpoints of the curve and the desired endpoint positions is minimized. Finally, we apply a rotation about the line passing through the endpoints that minimizes the angles between the endpoint tangents of the curve and the desired tangents. All these transforms can be computed analytically. We can combine the scaling, translation and rotations into one transform that aligns the curve with the constraints. The error in the alignment is simply a weighted sum of the error in the endpoint positions and the error in the tangents.

VI. Path Planning for Minimal-Energy Curves

The path planning problem for minimal-energy curves can be stated as: when given endpoint constraints for a start and goal curve can we find (a) minimal-energy curves that satisfy those constraints, and (b) 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 [18], but it is not tied to any roadmap construction algorithm. In fact, it could also be used with a tree-based planner [19], [20]. Various algorithms have been proposed for the construction of roadmaps and trees elsewhere, and will not be discussed in this paper. Instead, we will focus on the specifics of the local planner for minimal-energy curves.

To solve the path planning problem we propose the following approach. First, a roadmap of all minimal-energy curves is pre-computed 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 [21]. Whereas Leven and Hutchinson pre-compute 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 problem that the local planner needs to solve can be stated as: 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 [1]. 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. The distance between two curves is defined as

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

Because the curves have piecewise-constant curvature and torsion, the integral simplifies to a summation. The path planner recursively finds a path as follows. It first computes

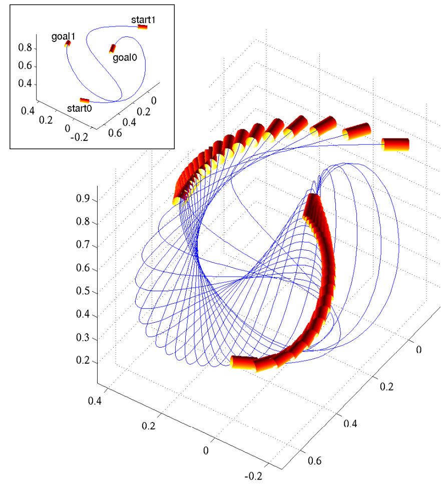

Fig. 3. A path of minimal-energy curves. The inset shows the start and goal curves. The start curve connects start0 and start1, the goal curve connects goal0 and goal1.

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 $\varepsilon/c, c > 1$, 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 *mid-point* between the endpoints. We use spherical interpolation to determine the position of the endpoints relative to the mid-point. The tangents are also spherically interpolated.

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. 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.

Figure 3 shows an example of a path as found by our path planner. Figure 4 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.

VII. Multiple Control Points

So far we have assumed that the only control points and tangents that a minimal-energy curve needs to pass through

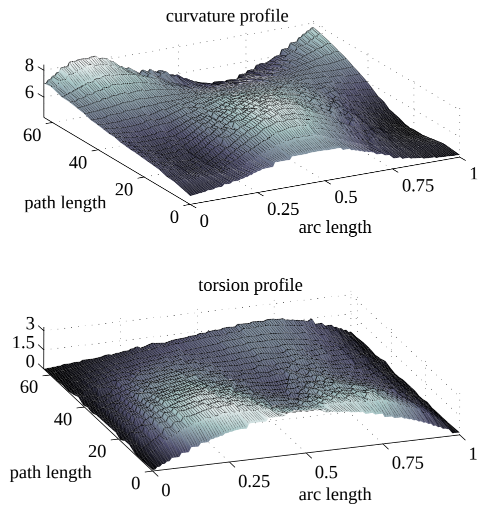

Fig. 4. Curvature and torsion along a path of minimal-energy curves.

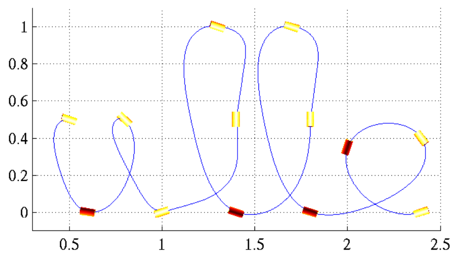

Fig. 5. A minimal-energy curve of length 9 passing through 13 control points and tangents to spell the word "cello."

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.

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 work space 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 start the energy minimization.

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 have used 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 i^{th} 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. Figure 5 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 3D; the curve in figure 5 is planar only because it is easier to visualize.

VIII. 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++ [22]. In particular, in the subdivision step we used the quasi-Newton method with numerically computed derivatives, and in the optimization of curve segment lengths we used a derivative-free parallel direct search.

We evaluated the performance of the subdivision scheme by randomly selecting constraints for the endpoints and timing how low 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, and the minimum segment length was set to 0.002. We generated 50,000 random curves and computed the following statistics:

	time (s)	**error**	**energy**	**#segments**
mean	0.209	5.01×10^{-3}	15.90	59.3
median	0.167	5.64×10^{-5}	14.64	48.0
std. dev.	0.154	0.0484	8.86	41.8

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.

IX. Discussion

This paper described a new approach to path planning for flexible curves. We introduce a subdivision scheme to construct representations of minimal-energy curves. 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, and hyper-redundant / 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 VII 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. Another issue that needs to be addressed is the situation where control points are not in general position. For instance, for planar minimal-energy curves changing the torsion will not help in minimizing the energy. So in our subdivision scheme the effective number of DOFs can be too small to minimize the energy and at the same time satisfy the endpoint constraints. To solve this problem, we need to recognize that we are at or near a singularity and increase the number of segments to be subdivided.

Acknowledgment

The authors would like to thank Prof. Joe Warren for useful discussions on minimal-energy curves.

References

[1] M. Moll and L. E. Kavraki, "Path planning for minimal energy curves of constant length," in *Proc. 2004 IEEE Intl. Conf. on Robotics and Automation*, 2004, pp. 2826–2831.

[2] F. Lamiraux and L. E. Kavraki, "Planning paths for elastic objects under manipulation constraints," *Intl. J. of Robotics Research*, vol. 20, no. 3, pp. 188–208, 2001.

[3] O. B. Bayazit, J.-M. Lien, and N. M. Amato, "Probabilistic roadmap motion planning for deformable objects," in *Proc. 2002 IEEE Intl. Conf. on Robotics and Automation*, vol. 2, 2002, pp. 2126–2133.

[4] H. Wakamatsu, A. Tsumaya, K. Shirase, E. Arai, and S. Hirai, "Knotting/raveling manipulation of linear objects," in *Proc. 2003 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2003, pp. 3156–3161.

[5] A. M. Ladd and L. E. Kavraki, "Using motion planning for knot untangling," *Intl. J. of Robotics Research*, vol. 23, no. 7–8, pp. 797–808, 2004.

[6] G. S. Chirikjian and J. W. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Trans. on Robotics and Automation*, vol. 10, pp. 343–354, 1994.

[7] J. Phillips, A. Ladd, and L. E. Kavraki, "Simulated knot tying," in *Proc. 2002 IEEE Intl. Conf. on Robotics and Automation*, Washington, DC, 2002, pp. 841–846.

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

[9] D. K. Pai, "STRANDS: Interactive simulation of thin solids using Cosserat models," *Computer Graphics Forum (Eurographics 2002 Proceedings)*, vol. 21, no. 3, 2002.

[10] H. P. Moreton and C. H. Séquin, "Function optimization for fair surface design," in *SIGGRAPH'92*, 1992, pp. 167–176.

[11] W. Wesselink and R. C. Veltkamp, "Interactive design of constrained variational curves," *Computer Aided Geometric Design*, vol. 12, pp. 533–546, 1995.

[12] E. Jou and W. Han, "Minimal-energy splines with various end constraints," in *Curve and Surface Design*. SIAM, 1992, pp. 23–40.

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

[14] B. K. P. Horn, "The curve of least energy," *ACM Transactions on Mathematical Software*, vol. 9, no. 4, pp. 441–460, Dec. 1983.

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

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

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

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

[19] D. Hsu, J.-C. Latombe, and R. Motwani, "Path planning in expansive configuration spaces," *Intl. J. of Computational Geometry and Applications*, vol. 9, no. 4-5, pp. 495–512, 1999.

[20] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Intl. J. of Robotics Research*, vol. 20, no. 5, pp. 378–400, May 2001.

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

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