International Journal of Robotics Research, vol 20, N 3, pp 188–208, March 2001

Planning Paths for Elastic Objects Under Manipulation Constraints

Florent Lamiraux* Lydia E. Kavraki
Department of Computer Science
Rice University
Houston, TX 77005
lamiraux|kavraki@cs.rice.edu

Abstract

This paper addresses the problem of planning paths for an elastic object from an initial to a final configuration in a static environment. It is assumed that the object is manipulated by two actuators and that it does not touch the obstacles in its environment at any time. The object may need to deform in order to achieve a collision-free path from the initial to the final configuration. Any required deformations are automatically computed by our planner according to the principles of elasticity theory from mechanics. The problem considered in this paper differs significantly from that of planning for a rigid or an articulated object. In the first part of the paper we point out these differences and highlight the reasons that make planning for elastic objects an extremely difficult task. We then present a randomized algorithm for computing collision-free paths for elastic objects under the above-mentioned restrictions of manipulation. The paper includes a number of experimental results. Our work is motivated by the need to consider the physical properties of objects while planning and has applications in industrial problems, in maintainability studies, in virtual reality environments, and in medical surgical settings.

1 Introduction and Motivation

The problem of planning a path for a robot consisting of one or more rigid objects has been studied extensively over the last decade [27, 36]. Today several planners exist that can efficiently produce paths for high dimensional robots (objects) moving in complex environments [25, 28, 36]. Despite the very large amount of work on the above problem, there exists almost no work on planning paths for objects that can deform. In this paper we discuss the reasons limiting the development of planners that take the physical properties of the manipulated objects into account. We then propose a solution for a restricted version of the problem.

We illustrate the problem examined in this paper with an example. Consider the scenario of Figure 1, where the ends of a flexible plate need to be placed in two hinges located on the side walls of a polyhedral box. The box is open from above and a top view is shown. The bottom part

*Author's present address: CNRS-LAAS, 7, avenue du Colonel Roche, 31077 Toulouse Cédex, France.

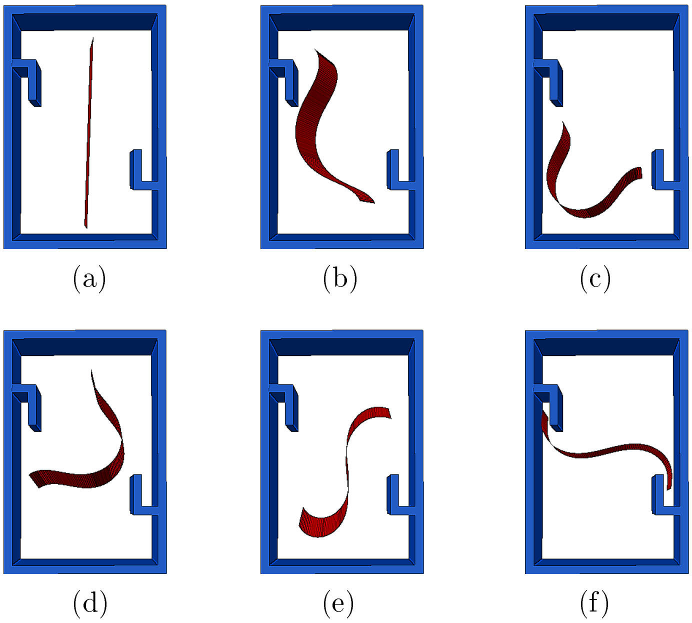

Figure 1: Snapshots along the path of an elastic plate computed by our planner. The box is shown from above and the bottom part of the box has been removed for display purposes. The plate is manipulated from above by two actuators that grasp its opposite short sides. The actuators are not shown in the figure.

of the box has been removed for visualization purposes. Two actuators (not shown in the figure) hold the two opposite short sides of the plate from above. We need the actuators to communicate energy to the plate, deform it, and manipulate it inside the box. We require that the plate does not touch the environment obstacles at any time. We also assume that the plate possesses linear elastic properties (is, for example, a sheet of metal). The main property of an elastic object is that after a deformation, it tends to recover its undeformed shape. According to elasticity theory from mechanics, when an elastic object is subjected to external constraints, it will end up with a shape that minimizes its elastic energy. This is a key point for the work presented in this paper. Our planner automatically computes the path shown in Figure 1. The path consists of configurations at which the elastic energy of the plate is minimized. Hence each of these configurations describes a shape of the plate that will be observed in practice. Elasticity limits are also respected so that the plate is not deformed permanently during the manipulation task. Our planner could be used to test the feasibility of performing the task shown in Figure 1 without permanently damaging the plate. Another example computed by our planner is shown in Figure 2, where a metallic belt is placed in a car assembly.

In this paper we do not plan the motion of the actuators. We only consider their effect on the manipulated object. Hence, our work is similar in spirit to the work done in assembly sequencing and assembly maintainability studies where removal paths for assembly parts are computed without taking into account the tools required to perform the removal [17, 63]. In assembly planning, as in our case, reasoning about the required tools complicates the problem to a degree that is very

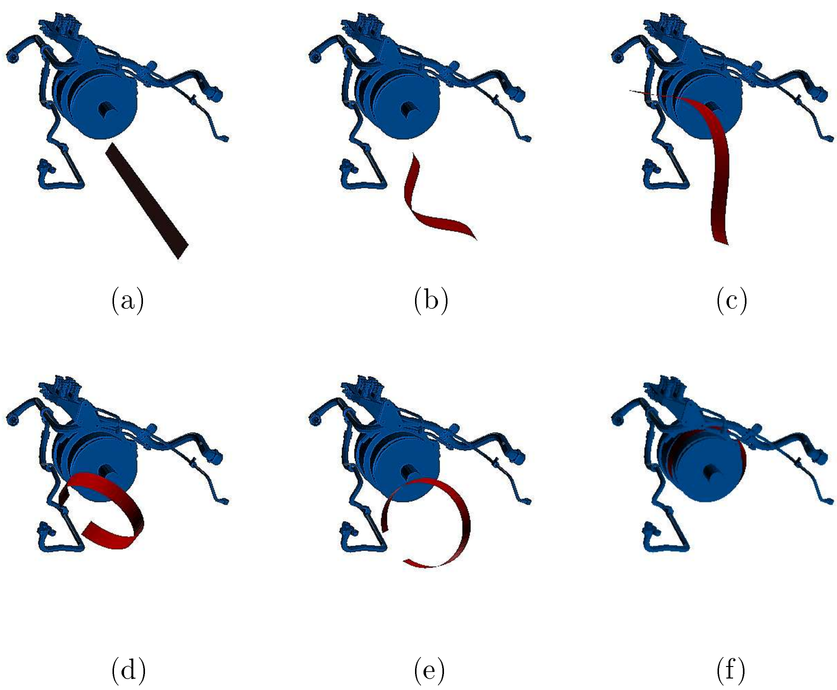

Figure 2: Snapshots along a path computed by our planner for positioning a metallic belt in a pipe assembly of a car. The actuators, not shown in the figure, grasp the two opposite short ends of the belt. At its final configuration the belt is around the main cylinder.

hard to address [64]. However, in both problem domains the information that is acquired without planning for the actuators is very useful as it can be used for feasibility studies and for testing the quality of product designs [17, 24]. As our understanding of the problem advances, it may be possible to consider the motion of the actuators. This topic is beyond the scope of the present paper.

The general problem of planning for deformable objects is a very important and extremely challenging problem. According to our current understanding, the general problem can hardly be addressed with the present state of the art in motion planning. The reasons are explained in Sections 2 and 4. Our work on a restricted version of the problem is a step in the direction of taking into account the physical properties of objects in planning applications. Several important applications could benefit from planners that account for the physical properties of the manipulated objects. For example, in industrial settings there is a need to handle sheets of metal [46], pipes that can bend [54], and cables [45]. In assembly maintainability studies done with virtual prototyping, planning is used to compute a removal path for a part from an assembly, given only the CAD model of the assembly [17]. The flexibility of the part needs to be considered as engineers use deformable parts to produce compact assemblies [16]. In medical and surgical procedures, flexible catheters are inserted in human vessels [5, 60]. Accurate planning studies may help in choosing the size and properties of the catheter used. In computer-assisted pharmaceutical drug design, path planning techniques are used to compute paths for drug molecules to their docking sites [53]. In that context, the rigorous treatment of the physical properties of the drug molecule, expressed by its internal energy, is crucial for obtaining sequences that are of low-energy and can thus be encountered in nature. Last but not least, several applications exist in domains like computer generated animation and virtual environments where the physical properties of objects need to be considered for the creation of realistic motions.

Contribution In the first part of this paper we give a precise definition of the planning problem we address in our work. To our knowledge such a definition has not appeared before. We carefully examine the different components of the planning problem before arriving to the statement of the problem. Our discussion reveals many of the difficulties of planning for deformable objects. The configuration of an elastic object is in general infinite-dimensional and cannot be represented by a vector as in the classical context of path planning. This raises approximation issues when representing the shape of the object. Although elasticity theory is fairly well understood in mechanics [12, 42, 43], its incorporation in a planning framework is a non-trivial task. We will show that it is computationally very expensive to express the elastic energy in terms of geometric representations from geometric modeling and that it is even more expensive to minimize the value of the energy function over the free parameters of shape.

The main part of this paper describes a randomized planning algorithm that can compute planning paths for elastic objects that are manipulated in a rather general way. Our algorithm is influenced by the probabilistic roadmap approach to motion planning [35]. Note that in the version of the problem that we address in this paper, the object does not have any contact with the environment obstacles. This enables us to decouple the deformation and the position of the object and arrive to a novel algorithm to compute legal paths. We have implemented our algorithm and present examples with two dimensional and three dimensional objects.

In our work we blend ideas from mechanics/physics (energy models for elasticity or other physical properties), geometric modeling (representations of curves and surfaces), and path planning for high-dimensional problems. Our developed planner can also be regarded as a test-bed for investigating several novel issues that arise in the context of planning with deformable objects. These include (a) acquiring accurate but computationally efficient energy models for the manipulated object, (b) understanding the implications of using different geometric representations for the object, (c) studying the attainable deformations of the object under manipulation constraints, (d) devising algorithms for planning low-energy paths between configurations with different deformations, (e) investigating approaches for efficient collision checking when the shape of the object changes, (f) developing methods for improving the overall quality of the computed path, and many others.

Parts of our work on planning for elastic objects have been presented in [4, 29, 33]. In this paper we unify and extend previous results and show how our planning method can be used without significant changes to plan for a variety of problems.

Organization Section 2 defines the problem we will consider in the rest of the paper. We carefully point out the differences of our problem compared to the traditional motion planning problem. Our discussion raises issues related to the geometric representation of elastic objects, the calculation of elastic energy given a geometric model, the allowable manipulation constraints and the estimation of stable equilibrium configurations, which are all examined in Section 2. We give related work in Section 3. We deemed this necessary as the discussion of the different components of the problem in Section 2 justifies why we looked into the robotics, mechanics, geometric modeling and graphics communities for related work. In Section 4 we present a randomized algorithm for our problem. In Section 5, we present three applications of our algorithm to different objects with different manipulation constraints and different geometric representations. We report experimental results for these applications. We discuss in Section 6 the computationally expensive parts of our framework and its limitations. We conclude by describing several open problems.

2 Definition of the Planning Problem for Elastic Objects Under Manipulation Constraints

In this section, we describe all the components required to define the problem of path planning for elastic objects under manipulation constraints considered in this paper. Because of the lack of definitions in Section 1 of this paper, the problem was illustrated with an example.

In the classical context of path planning, the robot consists of a set of rigid objects connected to each other by joints. The configuration space of such a system is finite-dimensional. For holonomic systems such as manipulator arms, a motor is associated to each degree of freedom, making any motion in a connected part of the free configuration space feasible [36]. In the case of nonholonomic systems such as mobile robots, the number of actuators is less than the dimension of the configuration space. However, in spite of the kinematic constraints, if the system is fully controllable, the existence of a feasible path is equivalent to the existence of any collision-free path [47].

When dealing with deformable objects, the configuration space of such an object can be infinite dimensional. In this case, the existence of a free path does not imply the existence of an energetically feasible path. The deformation of an elastic object in the context of our work is controlled by two actuators which constraint the position of a subset of the points of the object. The object ends up at a stable equilibrium configuration which minimizes its elastic energy according to the theory of elasticity from mechanics [42]. We assume that only the actuators are responsible for the deformations; the object is not allowed to come in contact with the environment obstacles and gravity by itself can not deform the object. First, we define the notion of configuration. Then, we define the elastic energy of a configuration using local deformation fields. We introduce manipulation constraints and we give a definition for stable equilibrium configurations. The computation of equilibrium configurations is a central operation when planning paths for elastic objects. We also introduce elasticity limits in the material that restrain the set of configurations that we can reach without permanently deforming (and thus damaging) the object. Finally, we state the problem of collision-free path planning for elastic objects considered in this paper.

2.1 Configuration

At its rest configuration q_0, an elastic object occupies a volume $V_0 \subset \mathbf{R}^3$. A *configuration* q of the object corresponds to a diffeomorphism[1] φ_q from V_0 to $V_q \subset \mathbf{R}^3$, mapping points in the rest configuration q_0 to their positions at configuration q. $V_q = \varphi_q(V_0)$ is the volume occupied by the object at configuration q. If $x \in V_0$ is a point in the rest configuration, we denote by $T_x\varphi_q$ the

[1] A diffeomorphism is a smooth one to one mapping with smooth inverse.

differential of φ_q at x. In the same way as φ_q maps points from configurations q_0 to q, $T_x\varphi_q$ maps vectors from q_0 to q. In general the configuration of an elastic object can be infinite dimensional and can not be represented by a vector. The differential enables us to define the local deformation of the object around x as described in the next paragraph.

2.2 Local Deformation Field

The deformation of an object is defined by a field of local deformations over the volume of the object. By definition, a rigid-body transformation keeps the inner product between any pair of vectors unchanged. Thus it seems natural that the local deformation about any point of the object is measured by the variation of the inner product about this point, as stated by the following definition.

Definition 1 *(Local Deformation)*
Let q be a configuration. Let $x \in V_0$ be a point in the rest configuration q_0 and $X = \varphi_q(x)$ be the same point in configuration q. For any vectors u and v at x, the images of these vectors in configuration q are the vectors $U = T_x\varphi_q(u)$ and $V = T_x\varphi_q(v)$ at X. The symmetric bilinear form $\mathbf{e}(x)$ defined on $\mathbf{R}^3 \times \mathbf{R}^3$ by

$$\mathbf{e}(x) : (u, v) \to \frac{1}{2}((U|V) - (u|v)),$$

where $(.|.)$ denotes the inner product, is the local deformation at x and it is also called the Green Lagrange strain tensor at x. We identify $\mathbf{e}(x)$ with its symmetric matrix in the local frame defined on the rest configuration q_0: $\mathbf{e}(x) = \frac{1}{2}(T_x\varphi_q^\top T_x\varphi_q - I_3)$, where I_3 is the identity matrix.

Notice that if φ_q is a rigid-body transformation, then for all $x \in V_0$, $T_x\varphi_q$ is a rotation, $T_x\varphi_q^\top T_x\varphi_q = I$ and then $\mathbf{e}(x) = 0$. From the above definition, it is straightforward that two configurations differing by a rigid-body transformation define the same Green Lagrange strain tensor field. The Green Lagrange strain tensor is used to define the elastic energy of a configuration.

2.3 Elasticity

Although the planner in this paper could be adapted to different types of mechanical behavior, we focus on the case of *elasticity* which is the most common class of mechanical models arising in real-world applications. Indeed, elasticity is closely related to reversibility of deformations. In a planning context, it is reasonable to expect that the shape of the object will not be affected at the end of the manipulation task.

Elasticity is a property of the material of an object. At each point $x \in V_0$ it defines a scalar function ψ called *density of elastic energy* [2]. The latter depends only on the local deformation $\mathbf{e}(x)$ at x. By integrating this local energy function over the domain of the object V_0, we obtain a functional over the configuration space. The value of this functional for any configuration is called *elastic energy*

$$E_{el}(q) = \int_{V_0} \psi(x, \mathbf{e}(x))dx. \tag{1}$$

[2] The density of elastic energy is an energy per unit of volume.

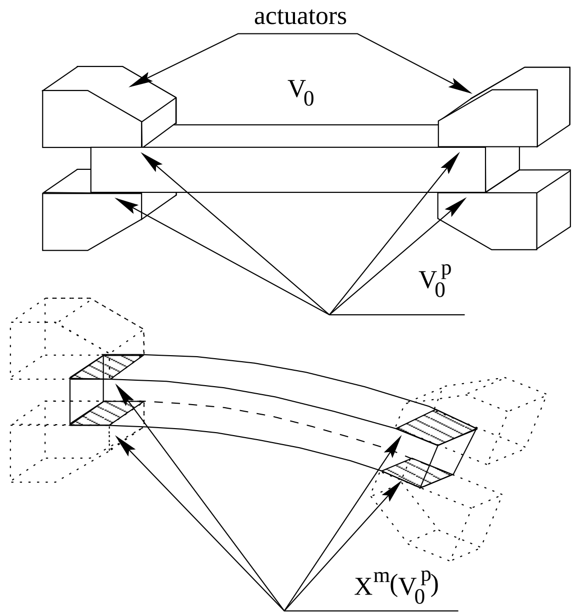

Figure 3: Manipulating an object consists in constraining the position of a given subset V_0^p of points of the object. This is done by grasping the object with actuators. For a given position m of the actuators, these points are moved to $X^m(V_0^p)$. The position of the other points of the object should be such that the elastic energy of the object is minimized.

The variable x of function ψ accounts for the fact that the material may not be homogeneous and that the relation between the local deformation and the density of elastic energy may vary within the material.

Homogeneous Isotropic Linear Elastic Material For the purposes of this paper we consider objects that are made from a homogeneous isotropic linear elastic material. This is a very commonly used model in mechanics since it perfectly describes materials like metals or composite materials. For these materials, the density of elastic energy is given by the following equation [12]:

$$\psi(\mathbf{e}) = \frac{E\nu}{2(1+\nu)(1-2\nu)}(\mathrm{tr}\ \mathbf{e})^2 + \frac{E}{2(1+\nu)}\mathrm{tr}\ \mathbf{e}^2, \tag{2}$$

where tr is the trace operator. E and ν are respectively called Young modulus and Poisson ratio and depend on the material of the object. These constants are known for a great variety of materials.

The main property of an elastic object is that after deformation, it tends to recover its undeformed shape. The undeformed shape has zero elastic energy. More generally, when subjected to external constraints, the elastic object will end up with a shape that minimizes its internal energy. In the following paragraphs, we define manipulation constraints and describe how to compute the effect of manipulation on the shape of the object.

2.4 Manipulation Constraints

In our planning framework, the object is typically grasped by two actuators and deformed by their action. Our definition of manipulation constraints is fairly general. The actuators constrain a subset $V_0^p \subset V_0$ of points of the object (see Figure 3). Let us denote by $\mathcal{M}$ the set of possible placements of the actuators relative to V_0. A placement $m \in \mathcal{M}$ of the actuators constrains the position of points in V_0^p defining a mapping X^m from V_0^p to $\mathbf{R}^3$. The manipulation constraint m defines a subset of configurations as follows.

Definition 2 *(Space of Configurations Fitting a Manipulation Constraint)*
Given a manipulation constraint m, we denote by $\mathcal{C}^m \subset \mathcal{C}$ the subset of configurations satisfying:

$$\forall x \in V_0^p, \varphi_q(x) = X^m(x).$$

$\mathcal{C}^m$ is called the subspace of configurations fitting m.

The subspace of configurations fitting a manipulation constraint is still infinite-dimensional. We do not consider the dynamic effects of the motion. We assume that the motion is slow enough to consider quasi-static paths. This means that along the motion, the object will stay in stable equilibrium configurations at all times.

2.5 Stable Equilibrium Configurations

Definition 3 *(Stable Equilibrium Configuration)*
A configuration fitting the manipulation constraint $m \in \mathcal{M}$ is a stable equilibrium configuration iff it is a local minimum of the elastic energy over the subspace $\mathcal{C}^m$ of configurations fitting m.

Besides constraining the position of a subset of points of the object, manipulation can also consist in applying forces to the object. In this case, the stable equilibrium configurations are those that minimize the sum of the elastic energy and the potential energy of the forces. In order not to affect the mechanical properties of the object, we need to restrict allowable deformations as described now.

2.6 Admissible Configurations

If we apply large deformations to an elastic material, the shape of the object in its rest configuration is different before and after the deformation. The relation between the local deformation field and the density of elastic energy is also different before and after the deformation. The energy of the rest configuration is not anymore zero because of internal constraints appearing in the material. This phenomenon has been studied extensively in mechanics and is generally called *elasto-plasticity*. When manipulating objects, we want to avoid deformations that affect the initial state of the material. For this reason, we stay within the elasticity limit of the material. The elasticity limit is characterized at each point $x \in V_0$ by a open subset of values of the Green Lagrange strain tensor $\mathbf{e}$ that contains $\mathbf{e} = 0$. This open subset depends on the material and can be determined by mechanical tests [12].

Definition 4 *(Admissible Configuration)*
We call admissible configuration any configuration q *for which* $\mathbf{e}(x)$ *is in the elasticity limit for any* $x \in V_0$.

2.7 The Planning Problem

Unlike in the classical context of path planning, any collision-free continuous curve between two configurations is not necessarily a solution to the path planning problem between these configurations. The problem has to be redefined in the context of elasticity.

Definition 5 *(Path Planning for Elastic Objects Under Manipulation Constraints)*
Let $\mathcal{C}_{free}$ *be the space of collision-free configurations of an elastic object. Let* q_1 *and* q_2 *be two free stable equilibrium configurations. A continuous curve* $\Gamma(s) \in \mathcal{C}_{free}, s \in [0,1]$ *connecting* q_1 *to* q_2 *is a solution of the path planning problem between* q_1 *and* q_2 *if and only if the following conditions are satisfied*

- *manipulability: each configuration along the path satisfies the imposed manipulation constraints,* $\forall s \in [0,1], \exists m \in \mathcal{M}, \Gamma(s) \in \mathcal{C}^m$,
- *quasi-staticity:* $\forall s \in [0,1], \Gamma(s)$ *is a stable equilibrium configuration (Definition 3), and*
- *elastic admissibility:* $\forall s \in [0,1], \Gamma(s)$ *is an admissible configuration (Definition 4).*

A collision-free path satisfying the above constraints is called an admissible quasi-static collision-free path between q_1 *and* q_2.

The computation of admissible quasi-static collision-free paths is the focus of the rest of this paper. In order to solve the problem in practice we propose to use a representation of the object from geometric modeling and approximate the possibly infinite dimensional configuration space of the problem by a finite dimensional one. We express the elastic energy in terms of the chosen geometric representation. Our manipulation constraints restrict the values of some parameters of the geometric representation while the values of the rest are found by minimizing the elastic energy of the object. These issues are further discussed in Section 4 where a randomized algorithm for the problem is presented.

3 Related Work

It is clear from Section 2 that our work combines topics from various disciplines. While planning has been studied in robotics issues relating to deformable objects have been studied mostly in the areas of mechanics, geometric modeling, and graphics. We briefly survey related work in each of the above areas.

Robotics In this paper we deal with high dimensional planning problems (more than 6 degrees of freedom). Hence we only survey methods that can plan for high dimensional systems. An extensive discussion of techniques that apply to low dimensional problems can be found in [36]. Due to the computational complexity of the planning problem [36], all planners that have been developed for high dimensional systems during the last decade have traded completeness for speed and simplicity (for a discussion see [28]).

The Randomized Path Planner (RPP) [7] is based on the use artificial potential fields coupled with randomization. RPP applies a potential defined across the workspace to several points on the robot, inducing a potential in the configuration space. The planner employs random walks to escape local minima and search for the goal configuration. Ariadne's clew algorithm [44] considers the initial configuration as a landmark. The algorithm incrementally builds a tree of feasible paths using genetic optimization to search for a collision-free path from one of the landmarks to a point as far as possible from any previous landmarks. A new landmark is then placed at that point. The process continues until the goal configuration can be connected to the tree.

Another approach, which is very relevant to our work in this paper, is the Probabilistic Roadmap approach (PRM) to path planning [34, 35, 49]. The idea behind PRM is to capture and represent the connectivity of the free configuration space by a random network (a roadmap), whose nodes and edges respectively correspond to randomly selected configurations, and collision-free path segments. Once the initial and the final configuration are connected to this network, a path can be found by graph search. Several variations of PRM exist (see for example [3, 30, 18, 10, 52, 37, 11, 62, 14]). Other interesting planners include [1, 2, 26, 32, 38, 31].

Although there exists efficient planners that take non-holonomic and kinodynamic constraints into account (for example [39, 31]), there are few cases where physical constraints and planning have been tightly coupled (one example is [20]). As far as deformable robots and parts are concerned, work has been done primarily in the context of manipulation. Robots with flexible links are now being built since they facilitate certain tasks (like hammering a peg into a hole) and their modeling and control is under development (for pointers to current work see [46]). Recent papers consider the dynamic analysis of robots with flexible payloads such as two robots manipulating a flexible metal sheet [46] or a vibrating object [54], or solve the task of inserting one end of a flexible wire into a hole while holding the other end [45]. Furthermore, research in snake-like robots has explored issues related to "geometric mechanics" that are relevant to our discussion [13, 48]. For example, the work in [48] describes the net motion of a snake robot as a function of variations in the mechanism's shape variables.

Mechanics Mechanics models physical properties such as elasticity [12]. Extensive tables exist detailing the elastic properties of several metals and composite materials. The work in [61] discusses the case of thin plates and develops an energy model for the deformation of a thin elastic plate that depends only on the planar deformation and the curvature of the plate. We use this model in our work. Let us note that the treatment of elasticity in mechanics is done independently of the geometric representation of the object. In this paper we need a geometric representation for the object in order to solve our planning problem in practice. Hence the models of elasticity can not be used as described in the mechanics field. We express them in terms of our chosen geometric representation.

Geometric modeling In geometric modeling several representations for curves and surfaces have been developed to enable accurate manipulation of shape while considering a relatively small number of parameters [8, 21]. This is very relevant to our work since we use such models to approximate the potentially infinite dimensional configuration space of a deformable object by a finite dimensional one. However, there is an important issue arising when using standard geometric representations. The focus in computer modeling has been in providing visually realistic models and little has been done to address issues like area or volume preservation. For example, geometric representations for curves, such as splines, do not preserve the length of the curve when the values of the parameters of the geometric representation change. In our work we enforce length preservation through the minimization of the elastic energy. A detailed discussion is given in Section 4. In this paper we consider Bézier representations, spline representations and spring models for our objects and discuss their tradeoffs.

Graphics In graphics physically based models have been proposed for deformable parts [56, 57]. A survey of deformable modeling in computer graphics can be found in [22]. The use of physical simulation and related optimization techniques as a means of geometric interaction has been applied to animation [58], drawing [59], free-form surface and volume modeling [15], mechanical design [65], and interactive molecular simulation [55]. For a discussion on the dynamic simulation of non-penetrating flexible bodies see [6]. Models and algorithms appropriate for the collision of deformable bodies are investigated in [19].

4 An Algorithm for Planning Paths for Elastic Objects Under Manipulation Constraints

To arrive to an algorithm for solving the problem defined in Section 2, we need to specify the geometric representation of the object and the way stable equilibrium configurations are computed. Each of these issues is described below.

4.1 Geometric Representation

The space $\mathcal{C}^m$ can be infinite-dimensional and finding a closed form for the diffeomorphism φ_q corresponding to a minimum configuration q is not always possible. For this reason, we need to approximate the configuration space by a finite-dimensional subspace. The goal of the geometric representation is to substitute the configuration space of the part with a finite-dimensional subspace in order to represent configurations by vectors. In general, the latter finite-dimensional subspace is an element of a family of subspaces approximating the configuration space with more and more accuracy, as stated in the following definition.

Definition 6 *(Geometric Representation of $\mathcal{C}$)*
Let $\mathcal{C}$ be the configuration space of the flexible object. A geometric representation of $\mathcal{C}$ is a family $(\mathcal{G}_n), n \in \mathbf{N}$ of finite-dimensional subspaces of $\mathcal{C}$ such that

$$\lim_{n\to\infty} \max_{q\in\mathcal{C}} d_{\mathcal{C}}(q, \mathcal{G}_n) = 0,$$

where $d_{\mathcal{C}}$ is a distance in $\mathcal{C}$.

Given a manipulation constraint $m \in \mathcal{M}$, we define $\mathcal{G}_n^m$ as the subspace $\mathcal{G}_n \cap \mathcal{C}^m$ of configurations of $\mathcal{G}_n$ fitting the manipulation constraint m. It is a good idea to choose a geometric representation in which it is easy to express the parameters of the manipulation constraint (see Section 5). Different geometric representations can be used to model the configuration space of a deformable object.

The most usual ones are polynomial and finite element representations [6, 21, 22]. In our work we have considered Bézier curves, splines and spring models.

4.2 Computation of Stable Equilibrium Configurations

Once a geometric representation has been chosen, the elastic energy of the object is obtained by integrating over the volume of the object the density of elastic energy ψ given by Equation (2). With certain geometric representations, the calculation of the elastic energy can be done analytically (see Section 5.2). In most cases however the integration is performed numerically. The integrand of Equation (1) is sampled on the volume of the object and summed using Simpson's formula.

A manipulation constraint restricts the position of several points on the object. The position of these points is expressed using the chosen geometric representation of the object. A stable equilibrium configuration corresponding to a manipulation constraint is computed by searching for a local minimum of the elastic energy E_{el} as defined in Equation (1) over the subspace $\mathcal{G}_n^m$, where n is chosen according to the desired accuracy. The constrained minimization can be done by a variety of methods [50, 51] depending on the degree of the function optimized and the availability of gradients. In our work we use a variation of a conjugate gradient method [50] to perform the minimization. Let us notice that several equilibrium configurations may fit the same manipulation constraint. As explained in Section 4.3, we carefully choose the initial configuration that is subjected to minimization to increase our chances of obtaining quickly a stable configuration. We also exploit the fact that the computed motion should be a continuous motion to deal with the potentially multiple local minima of the energy function. Ideally, one would like to have a geometric representation for the object that would facilitate the minimization of the elastic energy (for example, if gradients can be analytically computed this can speed up certain minimization procedures). Unfortunately none of the existing geometric representations have been developed with such a consideration in mind. Finding a geometric representation that can effectively support the calculation of the elastic energy of an object is a topic that deserves further research (see Section 6).

4.3 The Path Planning Algorithm

We present an algorithm that computes collision-free paths consisting of stable equilibrium configurations. The planner assumes that the manipulation constraint will not change during the motion (e.g., the object will be grasped along its two opposite edges throughout the motion) and that a parameterization of the manipulation constraint is available. In our framework, we specify the motion of the actuators by generating appropriate values for the parameters of the manipulation constraint. As discussed in the introduction, we do not explicitly compute the paths of the actuators. When the values of the parameters of the manipulation constraint change, the shape of the elastic object changes. We compute a stable equilibrium configuration for the object for each change in the values of the parameters. In the end, we return the path of the object.

Our planning algorithm builds on the PRM planner defined in [35]. Given an initial and a final configuration that specify a query, PRM builds a roadmap in the configuration space of the object. The roadmap initially contains only the initial and the final configurations. The planner iterates the following step. First, a number of free stable equilibrium configuration of the object are generated at random over the configuration space. These are the nodes of the roadmap. Then, the nodes are interconnected by a local planner that generates admissible quasi-static paths. Each time a path is found, an edge between the corresponding nodes is added to the roadmap. The above process is repeated until a solution to the planning query is found. This is achieved when the initial and final configurations are in the same connected component of the roadmap. A global path is returned by searching the roadmap and by concatenating local paths. The details of the planner are specified below.

A crucial element in our planner is that it decomposes the deformation and the position of the object. This is possible since we have assumed that the object is not allowed to touch the environment obstacles. The extension to the case where deformations can occur by contact with obstacles is clearly a challenging problem but it is outside the scope of this paper. The decomposition of the position and the deformation of the object serves a double purpose. Firstly, it facilitates the computation of paths where the flexible object may retain the same deformation (or a few deformations). Such an effect is desirable in practice. Secondly, it addresses a computational consideration: energy minimization is very time-consuming in our framework. By reusing minimized deformations as many times as possible we keep the running time of our approach within reasonable bounds.

The Algorithm The planner builds a graph $G = (V, E)$. Initially, $V = \{q_{init}, q_{goal}\}$ and $E = \emptyset$. The following steps are repeated until q_{init} and q_{goal} are in the same connected component of the roadmap.

1. *Node generation.* A random manipulation constraint m (*i.e.,* a random configuration of the actuators) is generated. This is done by selecting values for the parameters specifying the constraint uniformly at random from their allowed range. A deformation of the object is computed by minimizing its elastic energy. If the resulting deformation is not admissible (Section 2.6), another manipulation constraint is chosen and another minimization is performed until an admissible deformation has been computed. Then random rigid-body motions are generated and applied to the deformation, defining configurations with the same deformation. Each of the generated configurations is tested for collision with the environment obstacles and is added in V only if it is collision-free. This step generates N collision-free configurations with the same deformation.

2. *Node connection.* Each of the newly generated nodes from the previous step is tried for connection with its K closest neighbors in the roadmap. Distance in C should account both for rigid body transformation and for deformation; our particular choice is given below. Connections are performed by a deterministic local planner which generates quasi-static paths between pairs of configurations. We describe the local planner we use at the end of this section. If during the generation of the local path, a collision with the obstacles occurs, or the elasticity limits of the object's material are violated, the local planner simply fails. Successful executions of the local planner generate edges in E between the corresponding nodes. Note that the local path itself need not be retained as it can be recomputed on demand if it is part of the global path between q_{init} and q_{goal}.

3. *Enhancement.* At this step we identify configurations in V with few connections and generate more configurations close to them in an effort to increase the connectivity of R. It is assumed that configurations with few connections lie in difficult parts of $\mathcal{C}_{free}$. A configuration $q \in V$

is selected with probability $w(q) = \frac{\frac{1}{d_q+1}}{\sum_{i=1}^{N} \frac{1}{d_i+1}}$, where d_i is the degree of a node, that is the number of connections node i has with other nodes [34]. Then, we initiate a random walk in $\mathcal{C}_{free}$ from q. Keeping the deformation of the object the same, we pick a random direction in $\mathcal{C}_{free}$ and advance in this direction until an obstacle is found. Then a new direction (reflection) is chosen and the process is repeated until (a) maximum number of steps are taken, or (b) a maximum number of reflections are generated (see [9] for more details on reflections). The final configuration q_r of the random walk is added to V. The random walk itself is added to E and stored in the corresponding edge. q_r is tried for connection with its closest neighbors as in the connection step. A total number of M nodes are generated during the enhancement step.

At the end of the above loop, q_{init} and q_{goal} are in the same connected component of R. A graph search can yield a sequence of edges leading from q_{init} to q_{goal}. Concatenation of the corresponding local paths results in a global path between the two configurations. When we search R we look for a path that minimizes the number of distinct deformations of the nodes of V belonging to the path. This is done for practical purposes since we wish to reduce unnecessary deformations. The proposed planner suffers from all shortcomings of PRM-based planners. First of all, the approach is only probabilistically complete and a solution may not be returned even if one exists. Then the running may fluctuate: in some runs a critical deformation may be discovered quickly allowing the planner to find a path in a short amount of time. In other cases a long time may be spent before the critical deformation is found. Several crucial components of our algorithm are described below.

Energy Minimization Once a geometric representation has been chosen, a deformation is encoded by a vector. The elastic energy thus becomes a real function over a finite-dimensional vector space. With many geometric representations (e.g., Bézier curves and cubic splines), the elastic energy of a deformation can be expressed exactly with respect to the control points. Such an expression can be useful as it can speed up energy calculations. However it may be quite time-consuming to compute an analytical expression for the elastic energy (for example, while for cubic splines the energy and its gradient are obtained easily, for Bézier curves the expression becomes very complicated to compute as the number of the control point of the curve increases). Most often we approximate the elastic energy function and its gradient by sampling the density of elastic energy over the volume of the object and by computing numerically the integral of Equation (1) using Simpson's approximation [51]. We typically use the *conjugate gradient method* [51] to perform the minimization.

The Local Planner The local planner needs to be efficient when connecting configurations close to each other as it will be called a great number of times. It also needs to be deterministic to avoid storing the computed paths in the roadmap.

To make our local planner efficient, we exploit again the decoupling of deformation and position in the workspace. We attach a local frame to the object in such a way that if two configurations have the same deformation, they have the same expression in the local frame of the object. Any configuration can thus be seen as a pair $q = (d, r)$ were d is the deformation expressed in the local frame and $r \in \mathrm{SE}(3)$ is the position in space of the local frame. We denote by $\mathcal{D}$ the space of

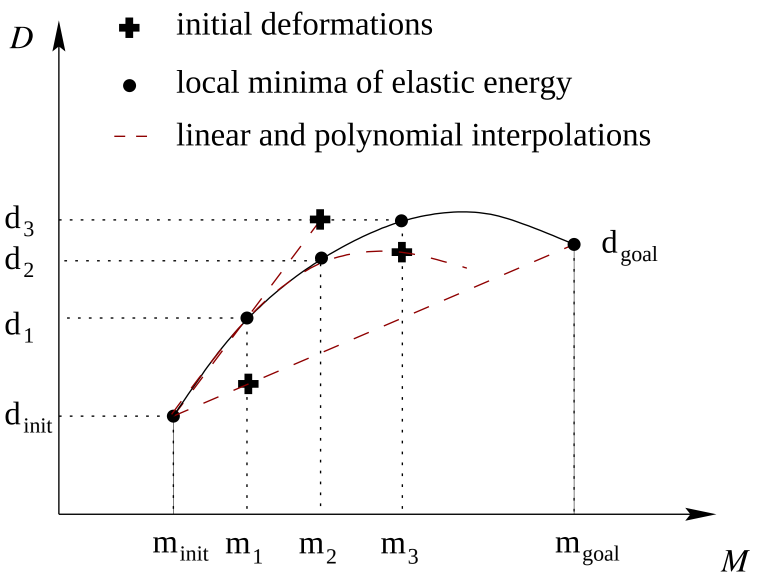

Figure 4: Manipulation constraints are sampled and energy minimization is performed for each sample point m_i. The initial deformation of each minimization is chosen as follows. d_1^{init} is the linear interpolation between d_{init} and d_{goal}. d_2^{init} is a linear extrapolation of $[d_{init}, d_1]$. d_i^{init} $(i > 2)$ is a quadratic extrapolation of $[d_{i-3}, d_{i-2}, d_{i-1}]$.

deformations expressed in the local frame of the object.

Let $q_{init} = (d_{init}, r_{init})$ and $q_{goal} = (d_{goal}, r_{goal})$ be two configurations and let m_{init} and m_{goal} be their respective manipulation constraints expressed in the local frame of each configuration. The path between q_{init} and q_{goal} is composed of a path between d_{init} and d_{goal} in $\mathcal{D}$ and a path between r_{init} and r_{goal} in SE(3).

The continuous deformation between d_{init} and d_{goal} is found in two steps:

- The control parameters of the deformation are the parameters of the manipulation constraint. We first define a path between m_{init} and m_{goal} by linear interpolation. This linear interpolation represents a continuous path for the actuators.

- We now discretize finely the linear path between m_{init} and m_{goal}. Each discretization point defines values for the parameters of the manipulation constraint. We compute a configuration of the object that corresponds to these values by minimizing the elastic energy of the object. The initial configuration of the object for the minimization procedure is extrapolated from the deformations already computed along the path (see Figure 4 for a detailed explanation). If the value of the elastic energy can not be reduced below the threshold value for maintaining elastic deformations, we assume that the local planner fails. We cache any valid paths in deformation space as they can be used for different configurations.

To compute a path in SE(3), the rigid body transformation $r_{goal} \circ r_{init}^{-1}$ transforming r_{init} into r_{goal} is encoded by a translation vector t and a rotation vector r. The path between r_{init} and r_{goal} is simply defined by linear interpolation in this parameter space $\mathbf{R}^6$.

To compute a path in $\mathcal{C}$, we first follow the path in $SE(3)$, keeping the deformation unchanged and then follow the path in $\mathcal{D}$. In both segments configurations are sampled finely along the path and each of them is checked for collision with the environment obstacles. If a collision if found, the local planner fails. We have observed that the rigid-body motion is much faster to compute since it does not involve any minimizations. With our approach, if a collision is found along the rigid-body motion, we avoid building the path in $\mathcal{D}$. Our experiments showed that the local planner described above is more efficient than a planner that simultaneously changes the deformation and the rigid body configuration of the object. This is the reason why we used the above local planner despite the fact that it is a conservative planner.

Distance Measure Our algorithm requires a distance measure between configurations. This distance is used to select the neighbors of a node and subsequently the local planner is used to connect neighbors. A good distance measure should account for the probability of failure of the local planner. Since the local path between two configurations consists firstly of the rigid body transformation and secondly of the change of deformation, as described in the previous paragraph, our distance measure is the sum of two distances

$$d(p, q) = d_d(p, q) + d_r(p, q),$$

where d_d is a *distance between deformations* and d_r is a distance between rigid-body motions. d_d is defined as follows. Points are sampled all over the surface of the object in its undeformed state. For two deformations expressed in the local frame of the object, we compute the Euclidean distance between each corresponding pair of points. d_d is defined to be the maximal distance computed. As far as d_r is concerned, we represent rigid body transformations by a rotation and translation vector and define d_r to be the Euclidean distance in $\mathbf{R}^6$. We have observed that in practice the above distance measure works well. Attempts to weight d_d and d_r have not yielded better results. However we noticed that using just d_d yields reasonable results.

Collision Checking Collision checking can be implemented using any standard collision checking library. We use the RAPID library [41]. This library takes as input collections of triangles describing the environment and the moving object. In our implementation, the object is approximated by a grid of points evenly sampled over the surface of the object. These points define triangles that are used by RAPID. The obstacles are also decomposed into triangle soups. Once an internal model of the object and a model of the obstacles have been created by RAPID, a configuration can be queried for collision by specifying a rigid transformation for both models. The creation of an internal model of the object is expensive compared to the actual collision checks. By keeping the deformation separate from the position in the workspace the internal model for any deformation can be built once and reused, speeding up collision checking. Better algorithms for collision checking for deformable objects are needed and this is a subject of current research [40].

5 Some Experimental Results

In this section, we apply our framework to three simple deformable objects. We use different types of manipulation constraints and different geometric representations in each case. Our goal is to demonstrate the feasibility of our approach and to also show that our framework is rather

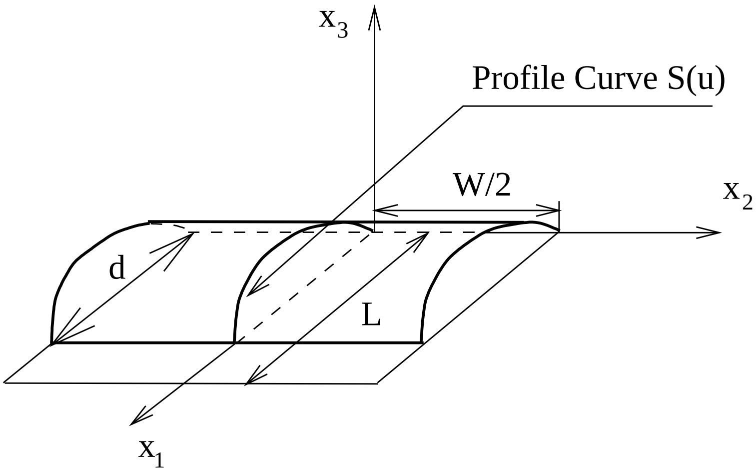

Figure 5: Manipulation of a plate by two opposite edges. In the local frame, one edge is fixed to the $(0, x_2)$ axis, while the opposite edge is in the plane (O, x_1, x_2), parallel to x_2 at a distance d. The deformation is one dimensional and can be represented by the profile curve $S(u)$.

general and can be applied to different examples without much tuning. We will use our experiments to raise interesting open questions for the problem of planning for deformable objects.

5.1 Bending an Elastic Plate

In this example, which is also described in [33], a rectangular thin plate is manipulated by grasping it at two opposite edges (see Figure 6).

Manipulation Constraint The size of the plate is L by W (see Figure 5). The grasping is done along the two opposite long edges and these edges are always kept parallel. The actuators in this case constrain the distance $d \leq L$ between the two opposite edges. Hence, the deformation is one-dimensional and the shape of the plate can be deduced from the profile curve $S(u)$ as indicated in Figure 5. The dimension of the planning problem is 7 (6 degrees for the placement of the plate and one for the deformation).

Mechanical Model We need to be able to compute the elastic energy of the plate with respect to a deformation. Suppose the plate is made of an homogeneous isotropic linear elastic material. In the general case of a volumetric object, the elastic energy, defined by Equation (1), is obtained by integration over the volume of the object of the density of elastic energy ψ. In the case of a thin plate manipulated as defined earlier, however, the local deformation is constant along x_2 and across the plate. The integral given in Equation (1) can be simplified to an integral along the profile curve and the density of elastic energy depends only on the stretching and curvature of the profile curve. For the detailed calculations leading to this approximation see [61]. Here we present only the results of these calculations. Let h the thickness of the plate. In the rest configuration, the profile curve is given by

$$x_1 = Lu, \quad x_3 = 0, \quad u \in [0, 1].$$

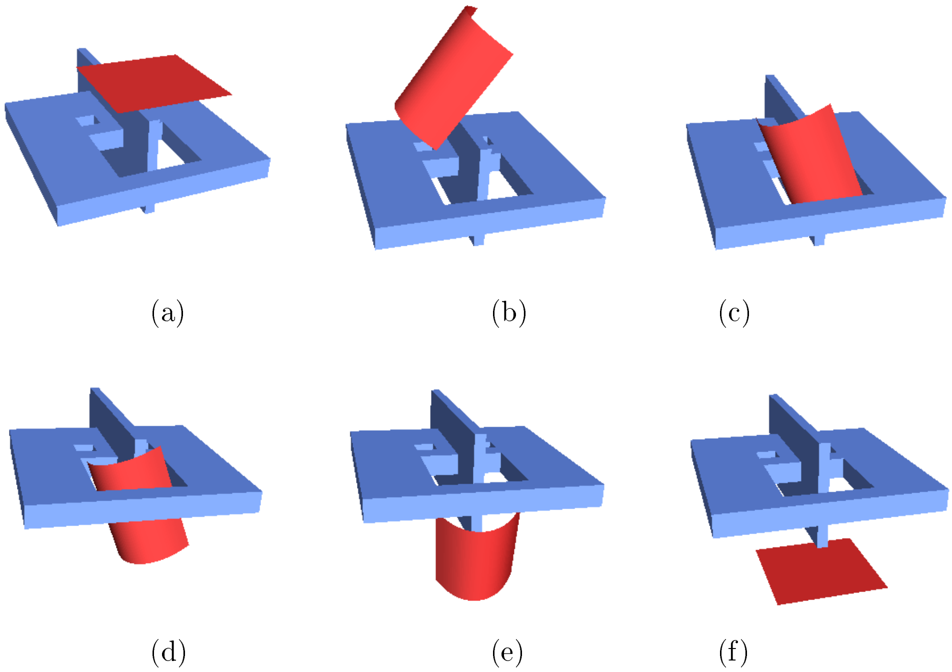

Figure 6: Motion of an elastic plate that can only bend.

For a given deformation d, u of the previous relations is mapped to $S(u)$, where $S(u), u \in [0,1]$ is the profile curve of deformation d. We define the following coefficients along the profile curve:

$$\begin{aligned} e(u) &= \frac{1}{2L^2}(\|S'(u)\|^2 - L^2), \\ \chi(u) &= \frac{\det(S'(u), S''(u))}{\|S'(u)\|^{3/2}}. \end{aligned}$$

These coefficients are respectively called the *stretching coefficient* and the *curvature coefficient*. In the above formula $S'(u)$ and $S''(u)$ are the first and second derivatives of the profile curve $S(u)$. Let us notice that $e(u)$ is the difference between the square norm of the tangent vector to the profile curve after and before a deformation and the difference represents the local stretching in the plate. In our case the strain tensor (see Section 2.2) is given by $\mathbf{e} = (e, \chi)$.

With the above notation and assumptions, Equation (1) becomes:

$$E_{el} = \frac{EWL}{2(1-\nu^2)} \int_0^1 he^2(u) + \frac{h^3}{12}\chi^2(u)du, \tag{3}$$

where E and ν are the Young Modulus and the Poisson ratio defined in Section 2.3. In order to be within the elasticity limit and avoid permanent deformations, we bound the deformation of the profile curve as follows:

$$|e| < e_{max} \text{ and } |\chi| < \chi_{max}.$$

To check the admissibility of a deformation with respect to the elasticity limit, we sample points along the profile curve and check if the local deformation at these points is admissible.

Geometric Representation We use Bézier curves to represent the profile curve. A Bézier curve is a polynomial curve expressed in the basis of Bernstein polynomials [21]

$$S(u) = \sum_{p=0}^{n} B_n^p(u) P_p,$$

where $P_0, ..., P_n$ are the control points and $B_n^p(u) = \begin{pmatrix} n \\ p \end{pmatrix} u^p(1-u)^{n-p}$ are the Bernstein polynomials. The manipulation constraint is easy to express in the above geometric model since the endpoints of a Bézier curve are the first and last control points. Thus, in the local frame of the plate, they are expressed as follows:

$$P_0 = (0,0), \qquad\qquad P_n = (d,0).$$

The elastic energy as defined by Equation (3) is computed numerically. The integrand is sampled along the profile curve and summed using Simpson's formula. An admissible deformation is found by minimizing the elastic energy over the free parameters (e.g., all control points expect P_0 and P_n).

Experimental Results Our planner is written in C++ and our experiments were performed on an SGI R10000. The problem shown in Figure 6 requires the thin plate to bend and go through a U-shaped hole. Note that the environment in Figure 6 is surrounded by walls that are not drawn in the figure, hence the plate has to go through the hole to attain its goal configuration. We used a 10 control point Bézier curve and we assumed we are dealing with a metallic plate. The parameters for the iterative step of our planner are $N = 200$, $M = 100$, and $K = 40$. During enhancement the random walk consists of a maximum of 10 reflections, each of which can be 100 steps long. We run our planner 10 different times changing the value of the random seed generator. The planner reliably solved the problem all 10 times with an average running time of 22.7 min. It generated on the average 12,500 nodes in the roadmap R. At the time when the planner succeeded, R had an average of 14 components. Several of these were small (contained less than 1% of the nodes in V).

5.2 More Complex Plate Bending

We still consider the case of a rectangular thin plate manipulated by two opposite edges but now we allow for more complex manipulation.

Manipulation Constraints The manipulation constraints specify both the position and tangent direction of two opposite edges of the plate as shown in Figure 7. To simplify notation, we assume that one end of the curve is fixed in the local frame of the plate, while the position of the other end is free. The planning problem we solve in this case is 9-dimensional. Of these, 3 degrees are needed for specifying the manipulation constraint and 6 are needed for the placement of the plate in its environment.

Mechanical Model We use the same elastic energy as in the former example (Equation 3).

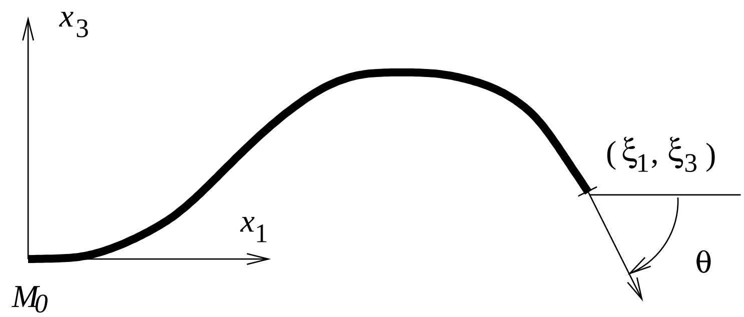

Figure 7: Manipulation of a plate by two opposite edges, specifying the position and tangent orientation. In the local frame of the plate, one edge is fixed to the (O, x_2) axis, while the opposite edge, is translated along x_1 and x_3 by respectively ξ_1 and ξ_3 and rotated about x_2 by θ. The deformation is again represented by the profile curve.

Geometric Representation To represent the profile curve, we use *pseudo cubic splines* [21]. Let us discretize the interval [0,1] into n segments of equal length, defining $u_n^i = i/n$, $i = 0, 1, ..., n$. Then a pseudo cubic spline is a C^2 curve over [0,1]. In fact it is a polynomial of degree 3 over each interval $[u_n^i, u_n^{i+1}]$. Given $n+1$ control points $P_0, ..., P_n$ and $n+1$ control vectors $V_0, ..., V_n$, there is exactly one pseudo cubic spline verifying $S(u_n^i) = P_i$ and $S'(u_n^i) = V_i$ for any i between 0 and n.

With this representation, the manipulation constraints can be written as follows

$$P_0 = (0, 0), P_n = (\xi_1, \xi_3), V_1 = (a, 0), V_n = (b \cos\theta, b \sin\theta),$$

where a and b are free parameters. In this case, the elastic energy (Equation (3)) and its gradient are computed exactly along each cubic segment and the values corresponding to each segment are summed. An admissible deformation is found by minimizing the elastic energy over the free parameters (control points, control vectors, a and b).

Experimental Results The problem shown in Figure 8 was drawn from a ship assembly. A plate is manipulated from above in a rather constrained space. Notice that the small part attached to the lower horizontal surface of the box does not allow the plate to move undeformed from its initial to its goal configuration. Again, the plate needs to flex to arrive to its final configuration. The plate is modeled with a 4 control point pseudo cubic spline. Our code was written in C++ and we obtained our results on an SGI R10000. The parameters of the planner were kept the same as in the previous example: $N = 200$, $M = 100$ $K = 40$. During enhancement the random walk consists of a maximum of 10 reflections, each of which can be 100 steps long. We run our planner 10 different times changing the value of the random seed generator. The average time to solve the problem was 4 hours 12 min. The significantly larger time is attributed to the following reasons. First of all, the space of deformations that needs to be explored is of higher dimension (3 against 1 in the previous problem). We need significantly more time to compute deformation paths because of the large number of minimizations involved. Secondly, the free space inside the box of Figure 8 is very constrained and the plate is almost as long as the box. Hence collisions with the obstacles are very likely. The average number of nodes in the roadmap R that solved the problem was 33,600 and the average number of connected components of R when the solution was found was 12. Again many of them contained less than %1 of the total nodes.

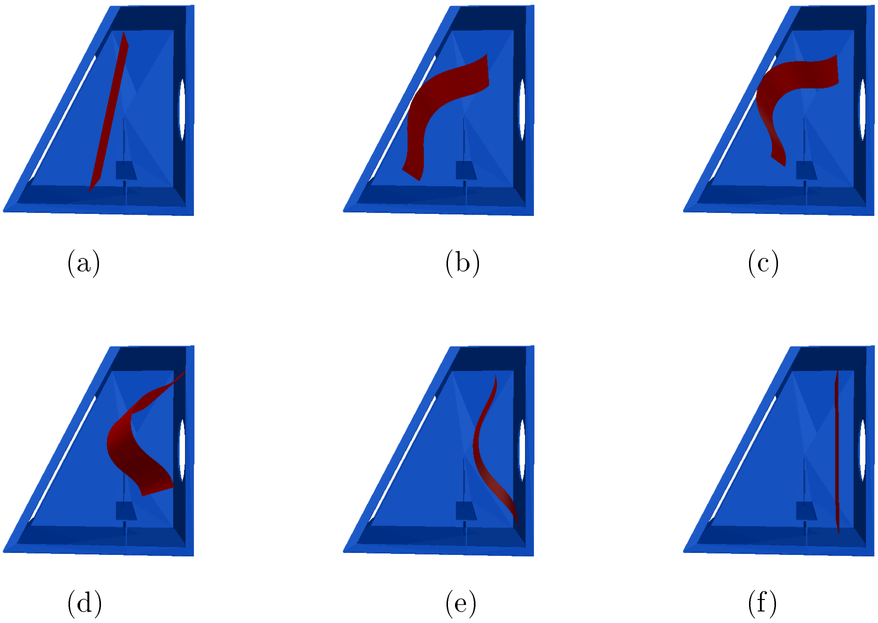

Figure 8: Path for a metallic plate in a ship assembly.

5.3 Manipulating an Elastic Pipe

In this section, we consider the case of an elastic pipe and we use a spring model to represent the pipe. Spring models have been studied extensively in the literature, especially in the context of dynamic simulations [19].

Manipulation Constraints The manipulation constraints specify both the position and tangent direction of the ends of the pipe. We do not allow twisting of the pipe. To simplify notation, we assume again that one end of the pipe is fixed in its local frame, while the other end is free. We specify manipulation constraints as indicated in Figure 9. Note that in this case 5 parameters are needed to specify the manipulation constraints.

Mechanical Model and Geometric Representation The idea behind spring models is that the mechanical behavior of the object is simulated by a lattice of mass-points connected to each other by linear and angular springs [19]. Except for boundary points, each point is connected to 6 neighbors by 6 linear springs and 3 angular springs (see Figure 10). A constant is associated with each spring and the elastic energy for the linear and angular springs respectively is of the following form:

$$E_{lin} = \frac{1}{2} k_{lin} (l - l_0)^2, \qquad E_{ang} = \frac{1}{2} k_{ang} \cos^2 \theta,$$

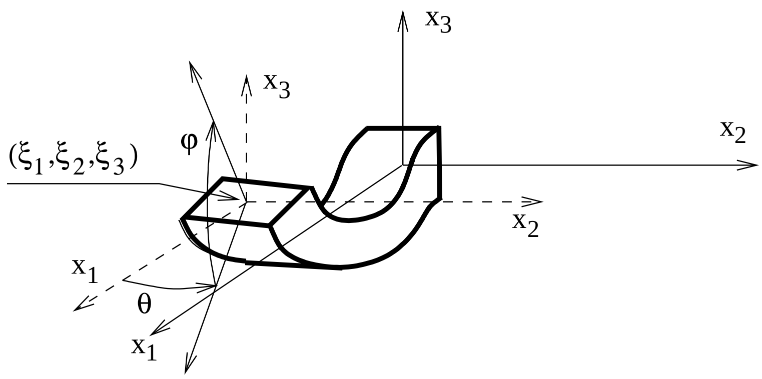

Figure 9: Manipulation of an elastic pipe. One end of the pipe is attached to the origin of the local frame while the position (ξ_1, ξ_2, ξ_3) and the orientation (θ, φ) of the other end are specified.

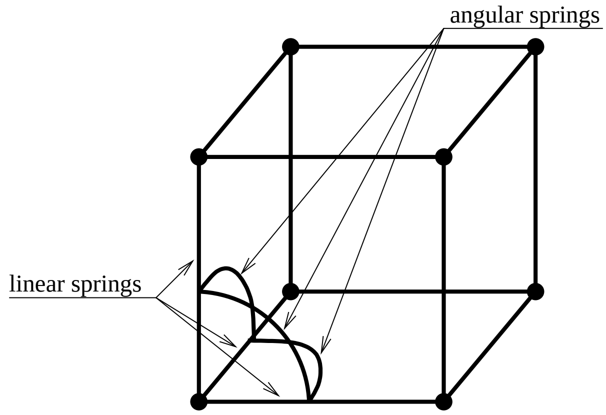

Figure 10: Spring model: The object is decomposed into elementary boxes. To each box is associated 3 linear and 3 angular springs. The elastic energy of these springs simulates the elastic energy of the associated box.

where k_{lin} and k_{ang} are constants that represent the stiffness of the springs. l_0 is the initial length of the linear spring and θ is the angle between two edges connected by an angular spring. The constants of these springs can be computed from the elasticity constants in a straightforward way. First, the homogeneity and isotropy of the pipe require that k_{lin} is the same for all linear springs and that k_{ang} is the same for all angular springs. We uniformly stretch the pipe in the x_1 direction from its undeformed shape and we equate the energy of the spring model with the elastic energy of the corresponding continuous deformation obtained from Equations (1) and (2). If we solve for k_{lin}, we obtain:

$$k_{lin} = \frac{(n_1 - 1)}{n_2 n_3} \frac{L_2 L_3}{L_1} \frac{E(1-\nu)}{(1+\nu)(1-2\nu)},$$

where L_1, L_2 and L_3 are the length, width and thickness of the pipe. n_1, n_2, and n_3 are the numbers of points of the lattice in the x, y, z directions. To determine k_{ang}, we shear the elastic pipe and equate in a similar way the elastic energy of the spring model with the elastic energy of the continuous mechanical model. We obtain when solving for k_{ang}:

$$k_{ang} = \frac{L_1 L_2 L_3}{2(n_1 - 1)(n_2 - 1)} \frac{E}{1 + \nu}.$$

For the detailed calculations we refer the reader to [4]. A configuration is a now represented by a vector of positions for each of the mass-points. Manipulation constraints restrict the position of the mass-points at the ends of the pipe. The elastic energy of a configuration is the sum of the energies of all the springs. An admissible deformation is found by minimizing the elastic energy over the free parameters (e.g., all coordinates of the free mass-points).

Experimental Results In the example of Figure 11, one end of the pipe is rigidly attached to a frame while the other is manipulated. In this experiment, the spring lattice is made of 32x3x3 points. Again our code was written in C++ and we obtained our results on an SGI R10000. Note that since one end of the pipe is fixed, each different deformation of the object represents a different configuration. So in this case, it does not make sense to generate a deformation and then create many placements of that deformation. We created 200 different deformations/configurations and we attempted to connect each of these with 40 neighbors ($K = 40$). In this example, we did not even need the enhancement step. We obtained a path with a roadmap of 200 nodes in all of the 10 runs of our planner. It took on the average 14.2 min and the produced roadmaps consisted on the average of 3 connected components. The relative high running time is due to the computationally expensive minimization. For the pipe of this example, we selected and minimized 100 random configurations. The mean time was 1.12 sec with a standard deviation of 0.98 sec.

5.4 Some Comments on the Geometric Models Used

In this section we were interested to demonstrate the versatility of our planning algorithm and we used a different geometric model for each of the three examples we examined. Each geometric model has its advantages and disadvantages. Bézier curves, for example, are very simple but the analytic computation of the elastic energy is expensive when the number of control points increases. Hence we need to resort to an approximate calculation of energy. Cubic splines are very useful and we can express the elastic energy of an object in terms of their parameters very easily. We observed however that our minimization procedure tends to converge slowly with this representation. Still we found cubic splines a good model and we recommend it for simple shapes. We also observed that not too many control points were needed when splines were used to represent an object (see Figure 12). Last but not least, mass spring models offer a very versatile model for three dimensional objects but again, the elastic energy has to be approximated and the energy minimization tends to be slow. Clearly, there is not a single geometric model that can be selected as best for the purposes of our work. In fact, available geometric models were not designed to support energetic calculations. It is an interesting open research topic to find models that facilitate energy calculations and use them in the context of planning. This point is further developed in the next section.

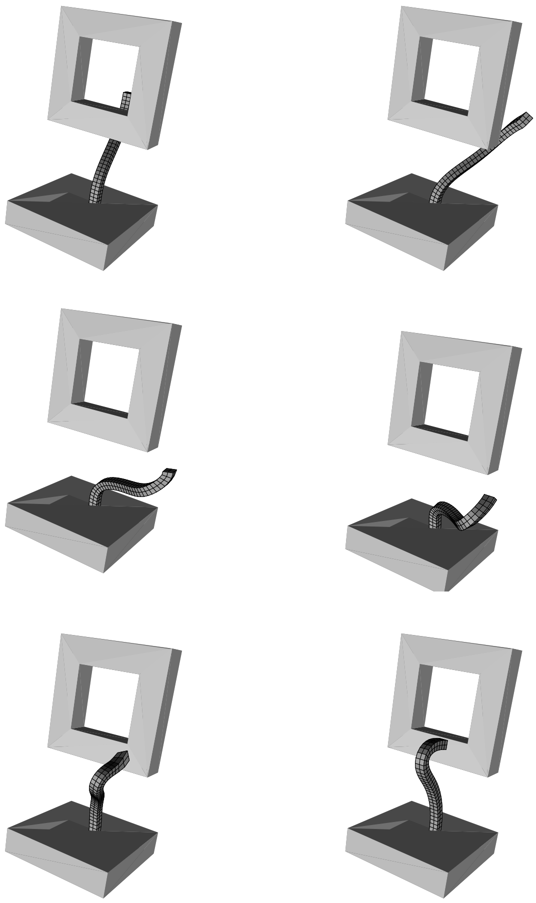

Figure 11: Snapshots along a path of a deformable cable whose one end is fixed to a base. The free end is manipulated by an actuator (not shown in the figure).

6 Concluding Remarks

In this paper we investigated the problem of planning paths for elastic objects under manipulation constraints. The problem differs significantly from the traditional path planning problem in robotics where only rigid or articulated bodies have been considered. Our work has applications in the manipulation of flexible plates, pipes, and cables in industrial settings, in virtual prototyping studies, in animation and virtual environments simulation, but also in medical studies and in computer-assisted pharmaceutical drug design.

In the first part of the paper we defined the different components of the problem considered in this work. In the second part of the paper we developed a planning framework to find admissible quasi-static paths for an elastic object that is manipulated by two actuators and that is not allowed to touch the obstacles in its environment. Our work is a first step in the direction of considering object flexibility during planning and raises many interesting directions for future research.

One important observation in our work is that available geometric models for representing shape are not well suited for expressing elastic energy. In most cases it is impossible to obtain a compact analytical expression for the elastic energy in terms of the parameters of the model. Even when this is achieved, the model does not preserve physical properties of the object such as surface area

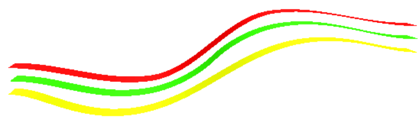

Figure 12: Representation of the same deformation using an 8 control point Bézier curve (bottom), a 4 control point cubic spline (top) and a 97 control point cubic spline (middle). Notice that a 4 control point cubic spline provides an accurate representation.

or volume: when the values of the parameters of the model change, physical quantities such as area and volume can fluctuate substantially. This is not surprising given the fact that most geometric models have been developed in the context of Computer-Aided Geometric Modeling, where the only requirement is visual realism. It would be very beneficial for our work if, for example, a geometric model could guarantee that the total area of the object would not change when the model deforms. It would also be very helpful if the model could facilitate the computation of elastic energy and its gradient. The topic is an interesting area for future research. Currently, we guarantee the preservation of physical quantities such as area and volume through the minimization of the elastic energy which is, however, very time consuming.

Another related issue concerns the geometric approximation done for representing the deformations of an object. A deeper investigation is needed to understand how the geometric approximation of the object interferes with the calculation and minimization of elastic energy and hence with our ability to express accurately the different deformations of the object. Such an investigation is a separate research topic in itself and it is not directly related to planning. For the context of our work, it would also be desirable to develop (a) energy models that are accurate but also efficient to compute and (b) minimization procedures that converge fast to local minima.

Let us conclude by reminding that we placed several restrictions on the problem we considered in this paper. An important one was that the object is not allowed to touch the obstacles in its environment. It is clear that deformations can also be created by contact with the obstacles. In that case, we can not decompose the deformation and the placement of the object as we did in this paper. Of course we can use the planner as is and generate a single configuration at each step. With our present understanding of the problem, the cost of energy minimization will be prohibitive for such a planner. Our current planner offers an excellent test-bed for studying the problem. It is clear that advances in many fronts will be required for the development of planners that can plan efficiently for deformable objects.

Acknowledgments

Work on this paper by Lydia Kavraki and Florent Lamiraux has been supported in part by NSF CAREER IRI-970228, NSF CISE SA1728-21122N, ATP 003604-0120-1999 and by a Sloan Fellowship to Lydia Kavraki. The authors would like to thank Rusty Holleman, Scott Owens and Elliot Anshelevich for their help in the implementation of the planner. The ship assembly was provided by Amrose and the car assembly by GE. The authors would also like to thank Joe Warren, Ron Goldman, Jean-Claude Latombe, and Leo Guibas for their comments.

References

[1] J. M. Ahuactzin and K. Gupta. A motion planning based approach for inverse kinematics of redundant robots: The kinematic roadmap. In *IEEE Int. Conf. Robot. & Autom.*, pages 3609–3614, 1997.

[2] J. M. Ahuactzin, K. Gupta, and E. Mazer. Manipulation planning for redundant robots: A practical approach. *Int. J. Robot. Res.*, 17(7):731–747, 1998.

[3] N. Amato, B. Bayazit, L. Dale, C. Jones, and D. Vallejo. Obprm: An obstacle-based prm for 3d workspaces. In P. Agarwal, L. Kavraki, and M. Mason, editors, *Robotics: The Algorithmic Perspective.* AK Peters, 1998.

[4] E. Anshelevich, S. Owens, F. Lamiraux, and L. Kavraki. Deformable volumes in path planning applications. In *IEEE Int. Conf. Robot. & Autom.*, pages 2290–2295, 2000.

[5] F. Arai, M. Tanimoto, T. Fukuda, K. Shimojima, and M. Negoro. Mutlimedia tele-surgery using high-speed optical fiber network. In *IEEE Int. Conf. Robot. & Autom.*, pages 878– 883, Minneapolis, MN, 1996.

[6] D. Baraff and A. Witkin. Dynamic simulation of non-penetrating rigid bodies. *Computer Graphics (SIGGRAPH'92)*, pages 303–308, 1992.

[7] J. Barraquand, B. Langlois, and J. C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Tr. Syst., Man, and Cybern.*, 22(2):224–241, 1992.

[8] R. Beach. *An introduction to the Curves and Surfaces of Computer-Aided Design.* Van Nostrand-Reinhold, NY, 1991.

[9] P. Bessiere, E. Mazer, and J.-M. Ahuactzin. Planning in continuous space with forbidden regions: The ariadne's clew algorithm. In K. G. et al, editor, *Algorithmic Foundations of Robotics*, pages 39–47. A.K. Peters, Wellsley MA, 1995.

[10] R. Bohlin and L. Kavraki. Path planning using lazy prm. In *IEEE Int. Conf. Robot. & Autom.*, 2000.

[11] V. Boor, M. Overmars, and F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In *IEEE Int. Conf. Robot. & Autom.*, pages 1018–1023, 1999.

[12] S. Borg. *Fundamentals of Engineering Elasticity.* World Scientific Publishing Company, Singapore, 1990.

[13] J. Burdick, J. Radford, and G. Chirikjian. A sidewinding locomotion gait for hyper-redundant robots. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, pages 101–106, Atlanta, GA, 1993.

[14] L. K. C. Holleman. A framework for using the workspace medial axis in PRM planners. In *IEEE Int. Conf. Robot. & Autom.*, pages 1408–1413, 2000.

[15] G. Celniker and D. Gossard. Deformable curve and surface finite-elements for free-form shape design. *Computer Graphics (SIGGRAPH'91)*, pages 257–266, 1991.

[16] H. Chang. Personal Communication.

[17] H. Chang and T. Y. Li. Assembly maintainability study with motion planning. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, pages 1012–1019, 1995.

[18] P. C. Chen and Y. K. Hwang. SANDROS:a dynamic graph search algorithm for motion planning. *IEEE Trans. Robot. & Autom.*, 14(3):390–403, 1998.

[19] A. Deguet, A. Joukhadar, and C. Laugier. Models and algorithms for the collision of rigid and deformable bodies. In P. Agarwal, L. Kavraki, and M. Mason, editors, *Robotics: the Algorithmic Perspective*, pages 327–338. A K Peters, 1998.

[20] B. Donald, P. Xavier, J. Canny, and J. Reif. Kinodynamic motion planning. *J. of the ACM*, 40:1048–1066, 1993.

[21] G. Farin. *Curves and Surfaces for Computer-Aided Geometric Design.* Academic Press, San Diego, 1988.

[22] S. Gibson and B. Mirtich. A survey of deformable modeling in computer graphics. Technical Report TR-97-19, MERL, 1997.

[23] K. Goldberg, D. Halperin, J. Latombe, and R. Wilson, editors. A.K. Peters, 1995.

[24] L. Guibas, D. Halperin, H. Hirukawa, J. Latombe, and R. Wilson. A simple and efficient procedure for polyhedral assembly partitioning under infinitesimal motions. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, pages 2553–2560, 1995.

[25] K. Gupta and A. P. del Pobil. *Practical Motion Planning in Robotics.* John Wiley, West Sussex, England, 1998.

[26] K. Gupta and Z. Guo. Motion planning with many degrees of freedom: sequential search with back-tracking. *IEEE Transactions on Robotics and Automation*, 6(11):897–906, 1995.

[27] D. Halperin, L. Kavraki, and J.-C. Latombe. Robotics. In J. Goodman and J. O'Rourke, editors, *Discrete and Computational Geometry*, pages 755–778. CRC Press, NY, 1997.

[28] D. Halperin, L. Kavraki, and J.-C. Latombe. Robot algorithms. In M. Atallah, editor, *CRC Algorithms and Theory of Computation Handbook.* CRC Press, NY, 1999.

[29] C. Holleman, L. Kavraki, and J. Warren. Planning paths for a flexible surface patch. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1998.

[30] D. Hsu, L. Kavraki, J. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In P. Agarwal, L. Kavraki, and M. Mason, editors, *Robotics: The Algorithmic Perspective*, pages 141–154. A K Peters, 1998.

[31] D. Hsu, R. Kindel, J. Latombe, and S. Rock. Randomized kinodynamic motion planning with moving obstacles. In *International Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2000.

[32] Y. Hwang and P. Chen. A heuristic and complete planner for the classical mover's problem. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1995.

[33] L. Kavraki, F. Lamiraux, and C. Holleman. Towards planning for elastic objects. In P. Agrawal, L. Kavraki, and M. Mason, editors, *Robotics: The Algorithmic Perspective*, pages 313–325. A.K. Peters, 1998.

[34] L. Kavraki and J.-C. Latombe. Randomized preprocessing of configuration space for path planning. In *IEEE Int. Conf. Robot. & Autom.*, pages 2138–2139, 1994.

[35] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. & Autom.*, 12(4):566–580, June 1996.

[36] J. Latombe. *Robot Motion Planning.* Kluwer, Boston, MA, 1991.

[37] J. Laumond and T. Siméon. Notes on visibility roadmaps and path planning. In *International Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2000.

[38] S. LaValle and J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *International Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2000.

[39] S. M. LaValle and J. J. K. Jr. Randomized kinodynamic planning. In *IEEE Int. Conf. Robot. & Autom.*, pages 473–479, 1999.

[40] M. Lin. Personal communication, 1999.

[41] M. Lin, D. Manocha, J. Cohen, and S. Gottschalk. Collision detection: Algorithms and applications. In Goldberg et al. [23], pages 129–141.

[42] A. E. H. Love. *A Treatise on the Mathematical Theory of Elasticity.* Dover, New York, 1988.

[43] J. Mark and B. Erman. *Rubberlike Elasticity : a Molecular Primer.* Wiley, New York, 1988.

[44] E. Mazer, J. Ahuactzin, and P. Bessière. The Ariadne's clew algorithm. *J. of Art. Intelligence Research*, 9:295–316, 1998.

[45] H. Nakagaki and K. Kitagaki. Study of deformation tasks of a flexible wire. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, Albuquerque, NM, 1997.

[46] W. Ngugen and J. Mills. Multi-robot control for flexible fixtureless assembly of flexible sheet metal auto body parts. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, pages 2340–2345, Minneapolis, MN, 1996.

[47] H. Nijmeijer and A. J. van der Schaft. *Nonlinear dynamical control systems.* Springer-Verlag, New York, 1990.

[48] J. Ostrowski and J. Burdick. Gait kinematics for a serpentine robot. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, pages 1294–1299, Minneapolis, MN, 1996.

[49] M. Overmars and P. Švestka. A probabilistic learning approach to motion planning. In Goldberg et al. [23], pages 19–37.

[50] D. A. Pierre. *Optimization Theory with Applications.* Dover, New York, NY, 1986.

[51] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, New York, NY, second edition, 1992.

[52] S. Sekhavat, P. Svestka, J.-P. Laumond, and M. H. Overmars. Multilevel path planning for nonholonomic robots using semiholonomic subsystems. *Int. J. Robot. Res.*, 17:840–857, 1998.

[53] A. Singh, J.-C. Latombe, and D. Brutlag. Path planning for molecular docking. Personal Communication.

[54] D. Sun, X. Shi, and Y. Liu. Modeling and cooperation of two-arm robotic system manipulating a deformable object. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, pages 2346–2351, Albuquerque, NM, 1996.

[55] M. Surles. An algorithm with linear complexity for interactive physically-based modeling of large proteins. *Computer Graphics*, 26:221–230, 1992.

[56] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics (SIGGRAPH'87)*, 21(4):205–214, 1987.

[57] D. Terzopoulos and A. Witkin. Physically based models with rigid and deformable components. *IEEE Computer Graphics and Applications*, pages 41–51, November 1988.

[58] D. Terzopoulos, A. Witkin, and M. Kass. Energy constraints on deformable models: recovering shape and non-rigid motion. In *Proc. of AAAI 87*, Seattle, 1987.

[59] J. Thingvold and E. Cohen. Physical modeling with b-spline surfaces for interactive design and animation. *Computer Graphics (SIGGRAPH'90)*, 24(2):129–139, 1990.

[60] T. van Walsum and M. Viergever. Deformable b-splines for catheter simulation. Technical report, Image Science Institute, Utrecht University, 1998.

[61] G. Wempner. *Mechanics of Solids with applications to thin bodies.* McGraw-Hill, NY, 1991.

[62] S. Wilmarth, N. Amato, and P. Stiller. MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space. In *IEEE Int. Conf. Robot. & Autom.*, pages 1024–1031, 1999.

[63] R. Wilson and J. Latombe. Geometric reasoning about mechanical assembly. *Artificial Intelligence*, 71:371–396, 1995.

[64] R. H. Wilson. A framework for geometric reasoning about tools in assembly. In *Proc. IEEE Int. Conf. on Rob. and Autom.*, pages 1837–1858, 1996.

[65] A. Witkin and W. Welch. Fast animation and control of non-rigid structures. *Computer Graphics (SIGGRAPH'90)*, pages 243–252, 1990.