Cable Route Planning in Complex Environments Using Constrained Sampling

Abstract

We present a route planning algorithm for cable and wire layouts in complex environments. Our algorithm precomputes a global roadmap of the environment by using a variant of the probabilistic roadmap method (PRM) and performs constrained sampling near the contact space. Given the initial and the final configurations, we compute an approximate path using the initial roadmap generated on the contact space. We refine the approximate path by performing constrained sampling and use adaptive forward dynamics to compute a penetration-free path. Our algorithm takes into account geometric constraints like non-penetration and physical constraints like multi-body dynamics and joint limits. We highlight the performance of our planner on different scenarios of varying complexity.

Keywords: motion planning, adaptive dynamics, articulated bodies, cable route planning

1 Introduction

Designing cable and wire layouts is often a complex and tedious process in building construction. Incorrect cable and wire layouts can be costly due to poor planning and may require significant modifications and design reviews. Current practices include scaled prototypes onto which the cable (or wire) layout is constructed by using route planning algorithms for motionless rigid cable segments. CAD systems are often used to assist the process during the early design stage. In these systems, the paths of the cables or wires are first calculated, then rigid segments are put through these paths to simulate cable or wire layout. However, current CAD systems do not take into account the cable's or wire's dynamics properties and their interaction with the environment.

To the best of our knowledge, there is very little or no work done on cable (or wire) routing that includes both realistic physical simulation and motion planning. Cable and wire routing can be posed as a robot motion planning problem to compute the path for a highly articulated and deformable robot, i.e. the cable (or the wire). However, the dimensionality of the configuration space of such a deformable robot is very high. In addition, the simulation of highly articulated and deformable bodies can be expensive and time consuming.

1.1 Main Results

In this paper, we present a novel approach for cable routing in complex environments. We assume that the cable can be modeled as a highly articulated robot with multiple links and many degrees of freedom (dofs).

Our route planning algorithm is based on a variant of PRM that samples near the surfaces of the obstacles in the workspace. It initially generates random samples at the corners and edges of the environment. If a path cannot be found, it samples on the obstacles by using ray shooting approach. The resulting global roadmap computed this way lies near the C-obstacles in the configuration space, also known as the *contact space* [Latombe 1991]. This roadmap is then used as the guiding path to plan the motion of the entire cable. During the simulation, whenever a collision occurs between the first link of the cable and the environment, the initial path is adjusted based on "constrained sampling", which recomputes a new node near the contact space. Our method expands the initial roadmap from a node belonging to the contact space (computed from a contact point) and finds a new node around the neighborhood of the contacting node using the projection method [Wilhelmsen 1976]. In order to simulate cable dynamics, we use an adaptive forward dynamics algorithm [Redon et al. 2005] that selects the most important joints to perform bounded error dynamic simulation in a hierarchical manner. Moreover, we develop efficient collision handling techniques to resolve the contacts for the remaining links during the simulation.

Our algorithm achieves realistic simulation results and performs each simulation step at interactive rates. We have implemented and tested our system on Alienware AMD Athlon^{TM} 64 X2 Dual-Core with 2.41 GHz Processor 4800++ and 2GB of RAM. Our algorithm can simulate a path for cables consisting of 200-300 links (or dofs) at interactive rates (averaging 60 fps) in modestly complex environments with tens of thousands of polygons. Our adaptive dynamics algorithm provides significant performance gain compared to the prior linear-time forward dynamics algorithms. In addition, our constrained sampling technique generates effective and useful samples near the obstacles, such as the walls, doorways, and around other support structures.

1.2 Overview

The rest of the paper is organized as follows. In Section 2, we briefly survey the prior work on motion planning for deformable robots, cable simulation and cable routing. We give an overview of our approach in Section 3. We present our cable simulation algorithm based on adaptive forward dynamics in Section 4. We describe our planning algorithm in Section 5 and highlight its performance in Section 6.

2 Previous Work

In this section, we give a brief overview of prior work in cable simulation, cable routing and motion planning for deformable objects.

2.1 Cable Simulation

Cable simulation has been studied in the recent years. Hergenrother and Dahne present an algorithm for the real-time simulation of virtual cables [Hergenrother and Dahne 2000]. Their simulation is based on inverse kinematics.They model the cable by using consecutive cylinder segments that are connected by ball joints. Given the start and goal positions of the cable, their algorithm calculates the shape of the cable by considering energy minimization. In practice, their algorithm is fast and applicable to interactive applications. However, their approach does not simulate the dynamics of the cable and does not perform collision handling, which is essential for realistic applications.

Loock and Schomer describe an application of rigid body simulation to assembly tasks in virtual environments and extend their system to real-time simulation of deformable cables [Loock and Schomer 2001]. They model the cable as a chain of rigid segments. The cable simulation is achieved by using mass-spring model with generalized springs. They use stiff linear springs and torsion springs to preserve the length of the cable.

Gregoire and Schomer present a numerically stable and physically accurate simulation tool for one dimensional components in [Gregoire and Schomer 2006]. They model the bending and torsion using the Cosserat model and they use a generalized spring-mass system with a mixed coordinate system for the simulation. However, their simulation is slow for modeling long cables. For example, the simulation time for cable that has 300 points is around 128 ms.

2.2 Cable Routing

The design of cable harness is a complex and costly process. Many approaches have been proposed to automate the design process. Conru [Conru 1994] describe a system to route cable harness using genetic algorithms. The genetic algorithms are used to search routes which are close to the global minimum. The cable harness routing problem is decomposed into generating a harness configuration and computing a route for the harness in the environment. Both of these problems are solved using genetic algorithms.

Holt et al.[Holt et al. 2004] present a virtual reality system to aid cable harness designers. Their approach focuses on usage of human engineer's knowledge in the design process. Their goal is to provide an engineer with interactive tools to design a human-in-the-loop system.

Both of these approaches do not consider the physical properties of the cable. They assume that the cable is composed of rigid segments and the number of segments can be adjusted by the user.

2.3 Motion Planning for Articulated and Deformable Robots

Motion planning is a well studied problem in robotics. Most of the work has been on rigid or articulated robots with a few degrees of freedom. Our variant of PRM and constrained sampling is similar to the OBPRM algorithm [Amato et al. 1998] and local planning in the contact space [Redon and Lin 2005], although our realization is quite different. In addition, we incorporate physical and mechanical constraints using adaptive forward dynamics and contact handling, whereas prior algorithm mostly deal with geometric constraints such as non-penetration.

There is relatively less work on motion planning for deformable objects. Some of the earlier work on deformable robots included specialized algorithms for bending pipes [Sun et al. 1996], cables [Nakagaki and Kitagaki 1997] and metal sheets [Ngugen and Mills 1996]. Holleman et al. [Holleman et al. 1998] present a probabilistic planner capable of finding paths for a flexible surface patch, modeled as a low degree Bèzier patch, using an approximate energy function to model deformation of the part. Guibas et al. [Guibas et al. 1999] describe a probabilistic algorithm for a surface patch, modeled as the medial axis of the workspace. Anshelevich et al. [Anshelevich et al. 2000] present a path planning algorithm for simple volumes such as pipes and cables by using a mass-spring representation. Lamiraux et al. [Lamiraux and Kavraki 2001] propose a probabilistic planner capable of finding paths for a flexible object under manipulation constraints. The deformation of object is computed by using the principles of elastic energy from mechanics which makes the motion planning difficult for handling the end constraints and finding minimum energy curves. In [Moll and Kavraki 2004], a different curve parametrization technique is used for handling low-energy configurations. In addition, contact points with simple obstacles are considered in finding a minimal energy curve configuration. However, finding the exact contact points that makes the curve at minimum energy is still a difficult task. Bayazit et al. [Bayazit et al. 2002] describe a two-stage approach that initially computes an approximate path and then refines the path by applying geometric-based free-form deformation to the robot. Gayle et al. [Gayle et al. 2005] present an algorithm for path planning for a flexible robot in complex environments. The algorithm computes collision free paths based on physical and geometric constraints. The collision detection between deformable object and the environment is achieved by using graphics processors. Saha et al.[Saha and Isto 2006] present a motion planning technique for the manipulation of deformable linear objects. The application of their method in self-knotting and knotting around simple static objects by using coordinating dual robot arms is illustrated in the paper. The motion planning algorithm depends on the geometrical model of the deformable object and the robot arms. It does not consider any physical properties of the deformable linear object. Many of these algorithms exploit geometric properties and often do not consider the physical constraints of the robot, such as collision detection and contact handling. Some recent approaches deal with general deformable robots and environments [Rodriguez et al. 2006], but do not model friction and motion constraints (e.g. joint limits) which could be necessary to realistically model the interaction between the cable and the rest of the environment.

3 Overview

In this section, we give an overview of our planning algorithm. We introduce the notation used in the remainder of the paper and present our framework to solve motion planning as a constrained dynamical system and model the robot as an articulated chain with a high number of dofs.

3.1 Notation

We assume that each cable can be modeled as a sequence of m rigid bodies connected by $m-1$ 2-dof revolute joints. The joints are implemented as two 1-dof revolute joints, one of which is rotated 90 degrees about the central axis of the cable. The configuration, $C(t)$, of the cable at time t can be described as a vector of joint angles along with the position and orientation of the base of the robot. The position of the head (first link) of the cable is represented by $C_{head}(t)$, which is the position and orientation of this link.

We assume that the set of obstacles are rigid and they are represented as $O = \{o_1, o_2, ...\}$ in the workspace, W. A roadmap, $G = \{V, E\}$, in the free workspace, the space external to the obstacles, consists of a milestones $V = \{v_1, v_2, ...\}$ and links $E = \{e_1, e_2, ...\}$. A path in this roadmap is an acyclic sequence that connects two milestones.

Problem Formulation: The problem can be stated as follows: Find a sequential set of cable configurations $C(t_1)$,...,$C(t_f)$ such that no

$C(t_i)$ intersects with any obstacle in O, and $C(t_i)$ is near the obstacles, where $C(t_1)$ and $C(t_f)$ are the initial and final configurations of the robot (cable) respectively.

3.2 Cable Simulation

When simulating a cable, there are a number of considerations which must be accounted first. First, to model a realistic cable, we assume that its length will not change during the simulation. This holds since little or no stretching will occur in most wires and cables when forces are applied to the ends.

In addition, our simulation must preserve geometric and mechanical properties of the cable itself. This primarily accounts for penetrations with the environment and also with itself as well as preventing a cable from bending too much. To efficiently simulate the cable, we take advantage of adaptive forward dynamics for articulated bodies [Redon et al. 2005] and develop a new collision handling method.

Our planner applies several constraints to the cable in order to ensure that it can successfully reach its goal. Path constraints are used to move it along a specified path, and collision constraints ensure that it is penetration free and also enforces joint angles.

3.3 Motion Planning

Our planning algorithm builds upon constraint-based motion planning (CBMP) proposed by [Garber and Lin 2002].

This approach involves two main stages; a roadmap generation stage and an execution stage. The first stage is largely done as a precomputation step and is respondsible for finding a *guiding path* through the environment. This guiding path does not need to be completely collision free with respect to the robot. The second stage uses constrained dynamical simulation to move the robot along this guiding path toward the goal. This allows the robot to locally adapt the guiding path to its own structure.

There are several advantages to the CBMP approach. Guiding path generation is both efficient and simple, while the path itself also automatically ensures that both geometric and physical constraints are preserved. For our problem, this allows the cable to move along a simple guiding path without violating its constraints.

4 Cable Simulation

The geometric and physical characteristics of the cables should be simulated realistically and efficiently for real-time applications. In this section, we first state our assumptions and the basic representation for modeling cables, then we describe our approach for simulating cable dynamics.

4.1 Assumptions and Representations

Cables have various material properties depending on its type and usage. In this work, we assume that a cable conserves its length during the simulation. A cable consists of rigid segments that cannot be stretched or sheared. Its length remains constant when dragged or pulled. Its cross section is undeformable and the mass of each segment is same along the entire cable. To model each cable, the length of the cable can be specified by the number of rigid segments or the length of each rigid segment. By changing one of these parameters, the user can model cables of varying lengths. Based on these assumptions, we can model each cable as a chain of articulated linkages, i.e. as a highly articulated robot.

The optimal algorithm for computing the dynamics of a kinematic chain takes linear time [Redon et al. 2005]. To approximate a cable well using a highly articulated chain would require many linkages. This computation can become rather costly for long cables. Therefore, we use the *adaptive forward dynamics* algorithm [Redon et al. 2005] for cable simulation. The main advantage of this approach is that it lazily recomputes the forward dynamics of a cable by only simulating the joints that best approximates the overall motion of the entire chain with bounded errors. The adaptive dynamics automatically selects active joints based on motion error metrics to compute an error-bounded approximation of the articulated-body dynamics. The user can change the motion error metric for controlling the number of joints that will be active or rigidified during the simulation.

The contact handling is also an important issue in cable simulation. The cable should deform realistically due to contacts and the penetration should be prevented when there is a collision between the cable and the nearby obstacles. We develop a penalty-based contact response capable of handling joint limits and other external forces along with a fast collision detection method to achieve sub-linear-time collision handling for multi-body systems.

Next we will describe the basic components of the adaptive multi-body dynamics and contact handling for our cable simulation.

4.2 Articulated-body dynamics

The adaptive dynamics algorithm is built upon Featherstone's *divide-and-conquer algorithm* (DCA) [Featherstone 1999a], [Featherstone 1999b]. Featherstone's algorithm is a linear time algorithm to compute the forward dynamics of an articulated body based on the forces applied to it. The algorithm relies on the following *articulated-body equation:*

$$\begin{bmatrix} \hat{\mathbf{a}}_1 \\ \hat{\mathbf{a}}_2 \\ \vdots \\ \hat{\mathbf{a}}_m \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_1 & \mathbf{\Phi}_{12} & \cdots & \mathbf{\Phi}_{1m} \\ \mathbf{\Phi}_{21} & \mathbf{\Phi}_2 & \cdots & \mathbf{\Phi}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Phi}_{m1} & \mathbf{\Phi}_{m2} & \cdots & \mathbf{\Phi}_m \end{bmatrix} \begin{bmatrix} \hat{\mathbf{f}}_1 \\ \hat{\mathbf{f}}_2 \\ \vdots \\ \hat{\mathbf{f}}_m \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{b}}_1 \\ \hat{\mathbf{b}}_2 \\ \vdots \\ \hat{\mathbf{b}}_m \end{bmatrix}, \tag{1}$$

where $\hat{\mathbf{a}}_i$ is the 6×1 spatial acceleration of link i, $\hat{\mathbf{f}}_i$ is the 6×1 spatial force applied to link i, $\hat{\mathbf{b}}_i$ is the 6×1 bias acceleration of link i (the acceleration link i would have if all link forces were zero), $\mathbf{\Phi}_i$ is the 6×6 inverse articulated-body inertia of link i, and $\mathbf{\Phi}_{ij}$ is the 6×6 cross-coupling inverse inertia between links i and j.

The DCA employs a recursive definition of an articulated body: an articulated body is a pair of articulated bodies connected by a joint. The sequence of assembly operations is described in an *assembly tree*: each leaf node of the assembly tree represents a rigid body, while each internal node describes an assembly operation, *i.e.* a subassembly of the articulated body. The root node of the assembly tree represents the complete articulated body.

The forward dynamics of the articulated body are computed in essentially two steps. First, the *main pass* recursively computes the inverse inertias, the inverse cross-coupling inertias, and the bias accelerations of each node in the assembly tree, from the bottom up. Then the *back-substitution pass* computes the acceleration and the kinematic constraint forces relative to the principal joint of each internal node in the assembly tree, in a top-down way. When the

DCA completes, all joint accelerations and all kinematic constraint forces are known.

4.3 Adaptive articulated-body dynamics

Featherstone's DCA is linear in the number of joints in the articulated body: all nodes have to be processed in each pass of the algorithm, and each joint acceleration has to be computed. Determining a path or resolving contacts for a highly articulated body could be prohibitively slow using a typical forward dynamics algorithm. In order to improve the performance of the planner, we incorporate the adaptive dynamics algorithm by Redon *et al.* [Redon et al. 2005] to lazily simulate the articulated body motion that best represents the overall motion of the robot with an error-bounded approximation. Essentially, this enhanced algorithm allows us to systematically choose the appropriate number of joints that are simulated in the articulated body, by automatically determining *which* joints should be simulated, in order to provide a high-quality approximation of the articulated-body motion.

Essentially, the adaptive algorithm relies on the proof that it is possible to compute an *acceleration metric value*

$$\mathscr{A}(C) = \sum_{i \in C} \ddot{\mathbf{q}}_i^T \mathbf{A}_i \ddot{\mathbf{q}}_i \tag{2}$$

i.e. a weighted sum of the joint accelerations in the articulated body, *before computing the joint accelerations themselves*. Specifically, they show that the acceleration metric value $\mathscr{A}(C)$ of an articulated body can be computed from the forces applied to it:

$$\mathscr{A}(C) = \begin{bmatrix} \hat{\mathbf{f}}_1 \\ \hat{\mathbf{f}}_2 \\ \vdots \\ \hat{\mathbf{f}}_m \end{bmatrix}^T \begin{bmatrix} \mathbf{\Psi}_1 & \mathbf{\Psi}_{12} & \cdots & \mathbf{\Psi}_{1m} \\ \mathbf{\Psi}_{21} & \mathbf{\Psi}_2 & \cdots & \mathbf{\Psi}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Psi}_{m1} & \mathbf{\Psi}_{m2} & \cdots & \mathbf{\Psi}_m \end{bmatrix} \begin{bmatrix} \hat{\mathbf{f}}_1 \\ \hat{\mathbf{f}}_2 \\ \vdots \\ \hat{\mathbf{f}}_m \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{f}}_1 \\ \hat{\mathbf{f}}_2 \\ \vdots \\ \hat{\mathbf{f}}_m \end{bmatrix}^T \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_m \end{bmatrix} + \eta, \tag{3}$$

where $\mathbf{\Psi}_i$ and $\mathbf{\Psi}_{ij}$ are 6×6 matrices, $\mathbf{p}_i$ is a 6×1 vector, and η is in $\mathbb{R}$. The coefficients $\mathbf{\Psi}_i$, $\mathbf{\Psi}_{ij}$, $\mathbf{p}_i$ and η are called the *acceleration metric coefficients* of the articulated body.

The acceleration metric is used to predict which nodes have the largest overall acceleration during the top-down back substitution pass. Computation of joint accelerations are restricted to these nodes while implicitly assuming that the other joint accelerations are zero. In effect, this allows us to determine an *error-bounded approximation* of the articulated-body acceleration, and evolve the set of active joints accordingly.

In summary, the adaptive algorithm is able to automatically determine which joints move the most, according to the acceleration and a similar velocity motion metric[1], based on the forces applied to the articulated body.

4.4 Collision Handling

The collision detection between the cable and the environment is performed using a hybrid bounding volume hierarchy. We perform two culling steps, based on *axis-aligned bounding boxes* (AABBs) and *oriented bounding boxes* (OBBs), to help localize potential collisions, before performing intersection tests at the triangle level. We pre-compute and store one hierarchy of oriented bounding boxes for each rigid cable link and each rigid environment obstacle. We also precompute one axis-aligned bounding box for each obstacle in the

[1] In our implementation, we use identity weight matrices.

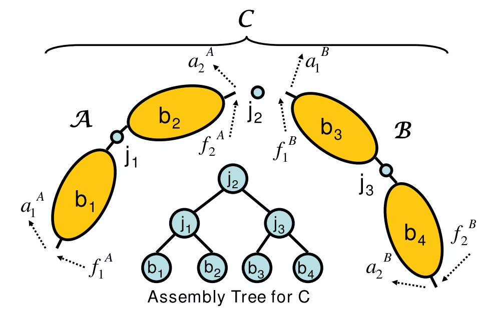

Figure 1: Construction of an articulated body. An articulated body $\mathbf{A}$ is connected to body $\mathbf{B}$ at the principal joint, j_2, to form body $\mathbf{C}$. The assembly tree for $\mathbf{C}$ is shown beneath the body. Forces and accelerations which govern $\mathbf{C}$'s motion are shown.

environment. The OBB hierarchies and AABBs of environment obstacles do not have to be updated during planning.

At runtime, we determine the intersections between the cable and the environment obstacles using the following collision detection algorithm:

- **AABB hierarchies update:** For each mobile rigid link, we determine a bounding AABB using the root OBB of the OBB-Trees [Gottschalk et al. 1996]. We then compute an AABB for each node of the assembly tree (*i.e.* for each subassembly of the articulated body) using a bottom-up pass. We thus obtain one AABB hierarchy per articulated body, whose structure is *identical to the assembly tree of the articulated body*.
- **AABB culling:** Using the AABB hierarchies, we detect potentially colliding rigid links and objects.
- **OBB culling:** When two rigid objects are found to potentially intersect after the AABB culling step, we recursively and simultaneously traverse their OBB hierarchies to help localize potential collisions between pairs of triangles [Gottschalk et al. 1996].
- **Triangle/triangle intersection tests:** Whenever two leaf-OBBs are found to overlap, a triangle/triangle intersection test is performed to determine whether the triangles contained in the leaf-OBBs intersect. When they do, we report the the corresponding intersection segment.

After collisions between the cable and the obstacles have been determined, the collision response is computed using a penalty-based approach as described in Sec. 4.5.3.

updated at each frame. Initially, the AABB of each link in the cable is updated using the root OBB of the OBB-tree.

4.5 Constraint Forces

In our simulation, in addition to the internal forces for kinematic constraints, we also apply external constraint forces for making the cable move along the computed path and for handling interaction between the cable and the obstacles. Next, we will describe the formulations of these constraint forces.

4.5.1 Attraction Force to the Goal

To make the rest of the cable follow the computed path, we apply attraction forces to the first link of the cable along the direction of the path. Initially, we discretize the path into a sequence of very close milestones $\mathbf{q}_i$ and then we apply attraction force to the first link of the cable to constrain it along the path. This force is basically a spring force between the first link of the cable and the computed path, and its magnitude is proportional to the distance between the first link of the cable and the intermediate goal $\mathbf{q}_i$ on the path.

$$\mathbf{f}_i^{Attraction} = \mathbf{k}(\mathbf{L}_i - \mathbf{L}_{rest}) \quad (4)$$

In this formulation, k is the spring constant, $\mathbf{L}_i$ is the distance between the milestone $\mathbf{q}_i$ and the first link of the cable, and $\mathbf{L}_{rest}$ is the intended distance between the first link of the cable and the milestone $\mathbf{q}_i$. When this force becomes less than a user defined threshold, we conclude that the cable reached the milestone $\mathbf{q}_i$ and we reset the intermediate goal to $\mathbf{q}_{i+1}$.

4.5.2 Path Constraint Forces

In order to constrain the motion of the entire cable along the computed path, we apply path constraint forces to a subset of the links of the cable, as applying force to each link is computationally expensive and inefficient. We select the links to be constrained based on their position with respect to the curved portions (such as around the corners) of the path followed by the first link of the cable due to the attraction forces to the goal configuration.

At each iteration, we find the nearest links to the curved portion and we apply constraint forces to these links (see Fig. 2):

$$\mathbf{f}_i^{Constraint} = \mathbf{w}^{curvature} \frac{(\mathbf{x}_i^{proj} - \mathbf{x}_i)}{\|\mathbf{x}_i^{proj} - \mathbf{x}_i\|} \quad (5)$$

where $\mathbf{w}^{curvature}$ is a weight depends on the curvature of the path segment that node i is constrained, x_i is the position of node i and $\mathbf{x}_i^{proj}$ is the projection of x_i on path segment.

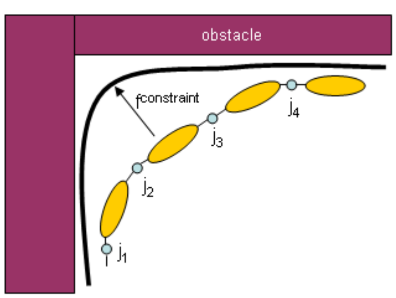

Figure 2: Path constraint force

4.5.3 Contact Handling Forces

When a collision is detected between the cable and the obstacle during the "Exact Contact Determination" step of the collision detection algorithm, we apply a force in the direction of contact normal to keep the cable apart from the surface of the obstacle. Let x_i be the position of contact link i, x_i^{proj} is the projection of x_i on the obstacle and N_i is the normal vector at position x_i^{proj}. Then the collision response force will be proportional to the penetration distance d, where

$$\mathbf{d} = (\mathbf{x}_i^{proj} - \mathbf{x}_i) \cdot \mathbf{N}_i \quad (6)$$

and the contact handling force will be

$$\mathbf{f}_i^{Contact} = \mathbf{k}\mathbf{N}_i\mathbf{d} \quad (7)$$

where k is the collision coefficient.

Given the basic steps of our cable simulation module, we will next describe our motion planning algorithm and show how our cable simulation is used in our cable route planning.

5 Motion Planning for Cable Layouts

Our cable route planning algorithm is composed of three key phases:

1. Global Roadmap Generation
2. Guiding Path Estimation
3. Path Adjustment and Constrained Dynamics Simulation
   (a) If there exists a collision between the first link and the environment, compute the next milestone using constrained sampling and modify the path accordingly;
   (b) For the remaining links, simulate the cable motion using adaptive forward dynamics with efficient contact handling and positional constraints.

Next, we will describe each step in more detail.

5.1 Global Roadmap construction

The probabilistic roadmap (PRM) algorithm is commonly used for robot motion planning. We develop a variant of PRM for computing the roadmap of the environment. For simplicity and efficiency, we first treat the cable as a point robot and then make path adjustments using constrained sampling and constrained dynamics simulation. This strategy makes the roadmap construction fast and effective.

5.1.1 Contact-Space Sampling

One of the important aspects of any randomized roadmap planner is the method used to generate samples in the configuration space. Since typically cables and wires should be placed in such a way that they attract as little attention as possible, one of our sampling goals is to find samples in the free space which are close to the obstacles (e.g. on or near the wall) and near the corners of the buildings.

We achieve this goal by systematically finding "corners" of a large structure. This is done by first computing the intersections of three or more constraint planes (walls), then the intersections of two constraint planes. Random samples are first taken at these locations. These points are shifted away from the walls by an amount δ to ensure that the local planning can construct the roadmap of the environment. After generating the samples near the corners, we construct our roadmap and we check whether a valid path exists. If there is no viable path from the initial to the goal configuration, then extra milestones are generated by shooting rays on these constraint planes (walls), which typically are the xy-, yz-, or xz planes for a building.

5.1.2 Local Planning

Another important step in PRM is local planning which connects collision-free nodes. We use a distance-based approach for finding the edges in the roadmap from a milestone v_i. All milestones that are within some search radius of v_i are placed into a neighbor set, N. For each milestone v_j in N, we determine if there is a straight line segment between v_i and v_j that does not intersect with any obstacle in O. For connecting the initial and final configuration to the roadmap, we also place them in the milestone set and perform local planning for them as well.

5.2 Computing a Guiding Path

Once we have generated a roadmap that provides the desired amount of coverage, we use it to generate paths between some given start and goal configurations. We first link the start and goal configurations to the roadmap. In particular, we add the start and goal configurations to our graph as query nodes and create edges between these nodes and all other nodes reachable from them. After these nodes have been added we can execute any graph search to find a path from the start to the goal. We use Dijkstra's shortest path algorithm with distance values as weights for computing the initial path.

5.3 Simulation and Runtime Path Modification

Once an estimated path is computed, we begin our simulation loop with the robot in its initial configuration. To make the cable follow the path during the simulation, we apply our attraction force to the first link of the cable in the direction of the path. This makes the first link of the cable move along the direction of the path, however the rest of the cable does not always follow the approximate path due to the dynamics of the cable. In order to constrain the motion of the entire cable, we apply additional path constraint forces to a subset of the links of the cable which are selected based on their position with respect to the curved regions of the path segments followed by the first link of the cable. This approach is similar to fixing portions of the cable against a wall to ensure it remains there.

As the cable traverses the path, we check the collisions between the cable and the obstacles. If there exists a collision between the cable and the obstacles, we apply collision handling algorithm to avoid the penetrations. In addition, if there is a collision between the first segment of the cable and the obstacles, we update the current path segment by computing a new, constrained milestone as described in Sec. 5.4.

For each simulation or path-query step, we apply the following algorithm:

1. Apply attraction force (Sec. 4.5.1) to the first link of the cable and positional constraint forces (Sec. 4.5.2) to the selected links in the cable;
2. Sum all forces and deform the cable using adaptive forward dynamics (Sec. 4.3);
3. Perform collision detection (Sec. 4.4) between cable and obstacles;
4. Apply contact response (Sec. 4.5.3) to resolve the collisions;
5. If there exists a collision between the first segment of the cable and the obstacle, update the current path segment between v_i and v_{i+1}.
   (a) Compute new milestone v_{new} using constrained sampling (the approach is presented in section 5.4);
   (b) Update the position of milestone v_{i+1} to v_{new};
   (c) Check collision between the new path segment v_i-v_{i+1} and the environment.

We repeat steps (1) - (4) until a path is found between the initial and final configurations, v_i and v_f. If no path is found, it is reported and the user can either add more milestones or conclude that there is no safe path. Otherwise, we end the simulation when it is reported that the object is sufficiently close to the final configuration.

5.4 Constrained sampling

Given the contact information between the first segment of the cable and the obstacle, we can determine the local tangent space of the obstacles for a given pair of sampled nodes near the obstacles. We use this tangent space to constrain the search of a new node when we found an intersection between the first link of the cable. and the obstacles.

Assume C_i is a configuration point in the roadmap and an external force toward this configuration is applied to the robot. Let $P = (P_x, P_y, P_z)^T$ denote the Cartesian coordinates of the contact point in the world frame. The contact point P corresponds to a point on the first link of the robot, and is a 3-vector $P(C_i)$ depending on the configuration C_i of the cable and the environment. Let $n = (n_x, n_y, n_z)^T$ denote the Cartesian coordinates of the contact normal in the world frame, and assume that this normal is directed towards the exterior of the obstacle. The local non-penetration constraint is:

$$dP(C_i) \cdot n \geq 0, \tag{8}$$

where $dP(C_i)$ is the small variation in the position of P corresponding to a small variation $d(C_i)$ around the configuration C_i of the robot. This non-penetration constraint defines a polyhedral set of valid variations. Provided the robot in the configuration C_i is in a consistent state with no interpenetration, the valid variations set is non-empty.

Because we use linear constraints, the set of valid variations is only a local piecewise-linear characterization of the valid space around the configuration C_i. This set of valid variations can be efficiently used for local planning. At runtime, whenever there exist a contact between the first link of the robot and the obstacle, we use the set of valid variations to help determine a new node C_{new}, by projecting the intentional variation on the set of valid variations. More specifically, whenever a new tentative node C_t is randomly chosen in the neighborhood of the contact-space node, we perform the following operations before checking its validity with a discrete collision checker:

1. Compute the corresponding tentative variation $dC_t = C_t - C_i$.
2. Project the tentative variation dC_t on the set of valid variations to obtain the new variation dC.
3. Set the new milestone $C_{new} = C_i + dC$

The projection of the intentional variation dC_t onto the polyhedral set of constraints is performed by using the Wilhelmsen projection algorithm [Wilhelmsen 1976].

Finally, we note that the combination of constrained sampling in the contact space and sampling by ray shooting allows us to accumulate obstacle constraints: whenever a new node C_{new} is determined by extending a node C_i, we check whether the near-obstacle constraints of C_i, are still valid in the new configuration C_{new}. This occurs for example when the robot comes in contact with an obstacle.

6 Results

In this section, we present the results of our approach on three different benchmarks of varying complexity. Below are our benchmark scenarios:

- **Bridge Model-** This model consists of over 5,000 polygons and the cable is represented as a 280-link robot with 5,000 polygons.
- **House Model-**This model consists of over 14,500 polygons and the cable is represented as a 280-link robot with 5,000 polygons. The house has four rooms and there is only one entry to the house.
- **Building Model-**This model consists of over 18,200 polygons and the cable is represented as a 280-link robot with 5,000 polygons. The building has seven floors and each floor has different shapes and dimensions. In order to illustrate the cable simulation and planning visually, we place the cable around the outside of the building.
- **Car Model-**This model consists of over 19,668 polygons and the cable is represented as a 280-link robot with 5,000 polygons. This model is obtained from the MPK model database.

Our algorithm runs at interactive rates, averaging about 60 fps. Fig. 3 illustrates the timings of our planning and sampling technique in detail.

In our simulation, in order to make the cable follow the exact path, we applied positional constraints to the selected joints based on their position instead of applying constraint to each link. Fig. 4 illustrates the comparison of the simulation time for positional constraints.

We also compared our algorithm by changing the sampling technique in our simulation. Instead of constrained sampling, we use random sampling in 3D. In constrained based sampling, we generated samples near the corners, edges and walls, and we applied constraint sampling during the simulation. However, in random sampling the samples are randomly generated in 3D. Fig. 5 shows the timing comparison of two sampling techniques.

To demonstrate the path quality generated by our algorithm with constrained sampling in the contact space vs. using random sampling in our approach, we show a comparison image highlighting the difference in path quality for the Bridge in Fig. 6. Notice that constrained sampling results in better cable placement than random sampling for cable route planning.

We also include additional sequences taken from our simulation runs (see Fig. 7–10). Our cable route planning algorithm is able to automatically compute realistic paths for these modestly complex models. Our algorithm using constrained sampling systematically finds milestones which are either on or near to the obstacles (walls, corners, structural supports, etc.) and the cable moves along the path naturally using our adaptive multi-body dynamics at interactive rates.

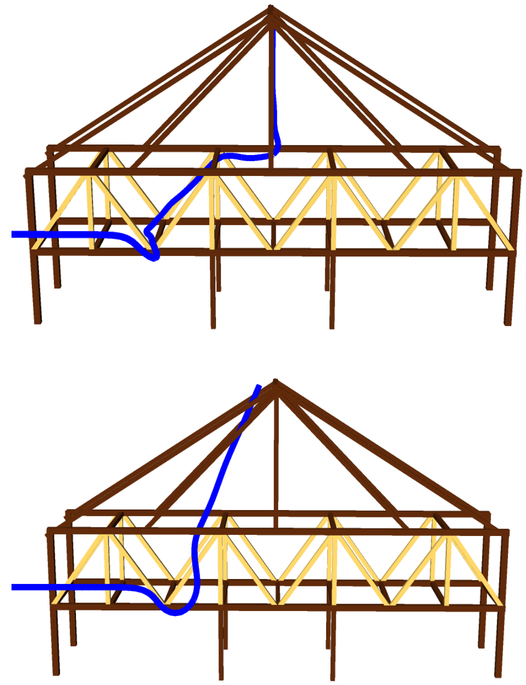

Figure 6: Path quality comparison on the Bridge Model using our algorithm with constrained sampling in the contact space (Top) and with random sampling (Bottom). Notice that the cable is better placed near the support structures on the bridge with constrained sampling. Using random sampling, the cable is simply routed from its initial configuration to the final configuration through uncluttered space and the placement of the cable appears rather awkward.

7 Conclusion

We present a novel algorithm for planning of cable layouts in complex environments. We propose a variant of PRM using a novel constrained sampling coupled with a fast adaptive forward dynamics algorithm and efficient collision handling. Our approach is applicable to many applications, such as wire routing. We have demonstrated our algorithm on four interesting scenarios and the initial results are rather promising.

Our current approach has some limitations. We first perform constrained sampling for the head of the cable and then the rest of the cable follows by adaptively computing its forward dynamics with contact handling for the entire cable. This algorithm does not guarantee a physically-correct motion and deformation at all time, but an error-bounded approximation. Nevertheless, the resulting path is always valid and collision free.

There are several directions for future investigation. It is possible that we can tightly integrate the adaptive forward dynamics framework with constrained sampling. This may result in a more robust system, but at higher computational costs. We can apply continuous collision detection to the cable simulation during the path following.

We would also like to explore cable route planning for cables with multiple branches, which can be useful in some real-world applications. In addition, we plan to apply our algorithm to more complex environments, such as a power plant. We would like to further improve the roadmap construction and collision detection for massive models consisting of many millions of geometric primitives.

Scene	# of Tri. in Cable	# of Tri. in Environment	# of Positional Constraints	Total joints	Motion Error Metric	Avg. Time Dynamics(s)	Avg. Time Simulation(s)	Total Sim. Time (s)
Bridge	5,600	5,000	6	280	0.01%	0.0026	0.0045	27.03
					0.1%	0.0019	0.0040	23.67
					0.5%	0.0013	0.0034	20.07
House	5,600	14,500	9	280	0.01%	0.0030	0.0051	49.98
					0.1%	0.0022	0.0048	47.04
					0.5%	0.0017	0.0041	40.01
Building	5,600	18,200	6	280	0.01%	0.0031	0.0051	41.31
					0.1%	0.0025	0.0047	38.07
					0.5%	0.0019	0.0040	32.40
Car	5,600	19,668	7	280	0.01%	0.0034	0.0055	32.49
					0.1%	0.0020	0.0040	23.66
					0.5%	0.0016	0.0037	22.23

Figure 3: Performance for different benchmarks

Scene	# of Positional Constraints	Motion Error Metric	Positional Constraint to Selected Links		Positional Constraint To Every Link	
			Avg. Time Dynamics(s)	Avg. Sim. Time (s)	Avg. Time Dynamics(s)	Avg. Sim. Time (s)
Bridge	6	0.01%	0.0026	0.0045	0.0063	0.013
		0.1%	0.0019	0.0040	0.0041	0.011
		0.5%	0.0013	0.0034	0.0032	0.009
House	9	0.01%	0.0030	0.0051	0.0062	0.018
		0.1%	0.0022	0.0048	0.0052	0.017
		0.5%	0.0017	0.0041	0.0043	0.016
Building	6	0.01%	0.0031	0.0051	0.0062	0.013
		0.1%	0.0025	0.0047	0.0055	0.013
		0.5%	0.0019	0.0040	0.0044	0.011
Car	7	0.01%	0.0034	0.0055	0.0074	0.014
		0.1%	0.0020	0.0040	0.0049	0.012
		0.5%	0.0016	0.0037	0.0040	0.011

Figure 4: Comparison of applying positional constraints to each link in the cable vs. selected links

Scene	Total joints	Motion Error Metric	Constrained Based Sampling		Random Sampling	
			Roadmap + Query Time (s)	Total Simulation Time (s)	Roadmap + Query Time (s)	Total Simulation Time (s)
Bridge	280	0.5%	12.31	20.07	8.82	26.82
House	280	0.5%	36.42	40.01	53. 26	49.86
Building	280	0.5%	41.64	32.40	39.74	42.93
Car	280	0.5%	46.21	22.23	43.25	26.23

Figure 5: Comparison between Constrained Based Sampling (CBS) and Random Sampling

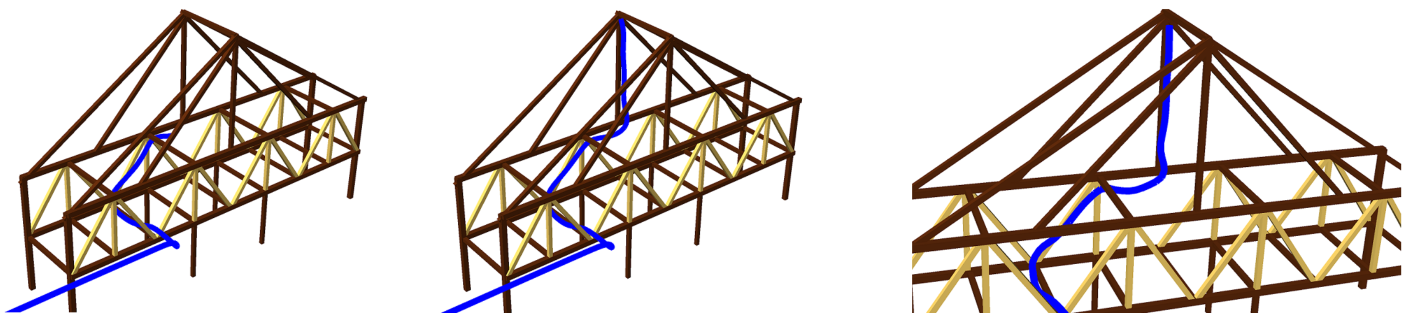

Figure 7: Cable route planning on the Bridge Model

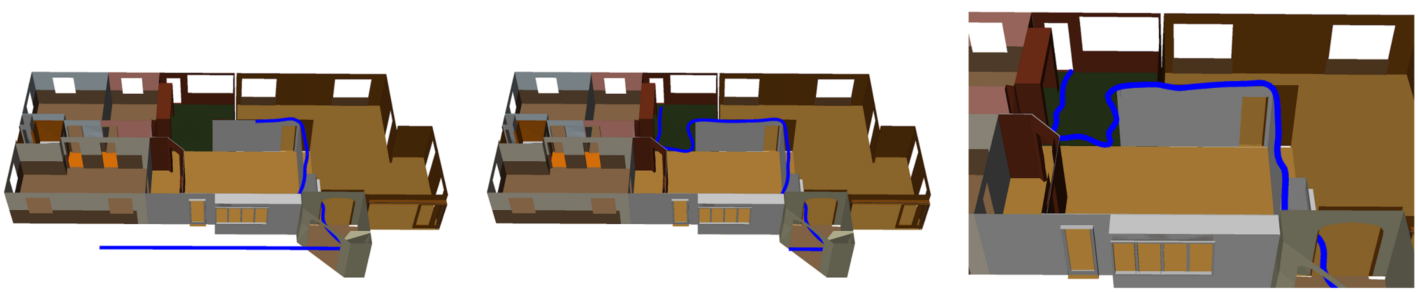

Figure 8: Cable route planning on the House Model

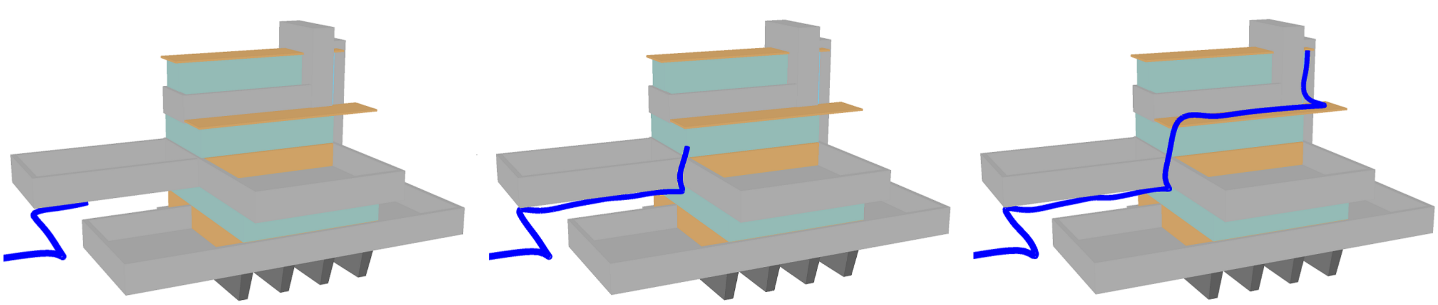

Figure 9: Cable route planning on the Building Model

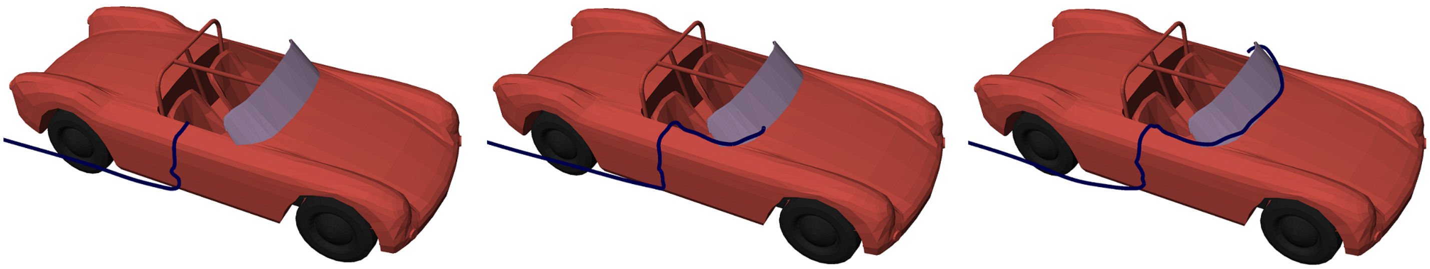

Figure 10: Cable route planning on the Car Model

References

AMATO, N., BAYAZIT, O., DALE, L., JONES, C., AND VALLEJO, D. 1998. Obprm: An obstacle-based prm for 3d workspaces. *Proceedings of WAFR98*, 197–204.

ANSHELEVICH, E., OWENS, S., LAMIRAUX, F., AND KAVRAKI, L. 2000. Deformable volumes in path planning applications. *ICRA*.

BAYAZIT, O. B., LIEN, J., AND AMATO, N. M. 2002. Probabilistic roadmap motion planning for deformable objects. *ICRA*, 2126–2133.

CONRU, A. B. 1994. A genetic approach to the cable harness routing problem. In *Proceedings of the IEEE World Congress on Computation Intelligence*.

FEATHERSTONE, R. 1999. A divide-and-conquer articulated body algorithm for parallel o(log(n)) calculation of rigid body dynamics. part 1: Basic algorithm. *International Journal of Robotics Research*, 18(9):867–875.

FEATHERSTONE, R. 1999. A divide-and-conquer articulated body algorithm for parallel o(log(n)) calculation of rigid body dynamics. part 2: Trees, loops, and accuracy. *International Journal of Robotics Research*, 18(9):876–892.

GARBER, M., AND LIN, M. 2002. Constraint-based motion planning for virtual prototyping. *UNC Technical Report*.

GOTTSCHALK, S., LIN, M., AND MANOCHA, D. OBB-Tree: A hierarchical structure for rapid interference detection. *Proc. of ACM Siggraph'96*, pages 171–180, 1996.

GAYLE, R., W.SEGARS, LIN, M., AND MANOCHA, D. Path planning for deformable robots in complex environments. In *Robotics: Systems and Science*. 2005.

GAYLE, R., LIN, M., AND MANOCHA, D. 2006. Adaptive dynamics with efficient contact handling for articulated robots. In *Proc. of Robotics: Systems and Science*.

GREGOIRE, M., AND SCHOMER, E. 2006. Interactive simulation of one-dimensional flexible parts. In *ACM Symposium on Solid and Physical Modeling (SPM'06)*, 95–103.

GUIBAS, L., HOLLEMAN, C., AND KAVRAKI, L. 1999. A probabilistic roadmap planner for flexible objects with a workspace medial-axis-based sampling approach. In *Proc. of IROS*.

HERGENROTHER, E., AND DAHNE, P. 2000. Real-time virtual cables based on kinematic simulation. In *Proceedings of the WSCG*.

HOLLEMAN, C., KAVRAKI, L., AND WARREN, J. 1998. Planning paths for a flexible surface patch. *IEEE Int. Conf. Robot. Autom. (ICRA)*.

HOLT, P., RITCHIE, J., DAY, P., SIMMONS, J., ROBINSON, G., RUSSELL, G., AND NG, F. 2004. Immersive virtual reality in cable and pipe routing: Design metaphors and cognitive ergonomics. In *Journal of Computing and Information Science in Engineering*, vol. 4, 161–277.

LAMIRAUX, F., AND KAVRAKI, L. 2001. Path planning for elastic objects under manipulation constraints. *International Journal of Robotics Research*, 20(3):188–208.

LATOMBE, J. 1991. *Robot Motion Planning*. Kluwer Academic Publishers.

LOOCK, A., AND SCHOMER, E. 2001. Virtual environment for interactive assembly simulation: From rigid bodies to deformable cables. In *Multiconference on Systemics, Cybernetics and Informatics (SCI'01)*, Virtual Engineering and Emergent Computing.

MOLL, M., AND KAVRAKI, L. 2004. Path planning for minimal energy curves of constant length. In *ICRA*.

NAKAGAKI, H., AND KITAGAKI, K. 1997. Study of deformation tasks of a flexible wire. *Proc. of IEEE Int. Conf. on Robotics and Automation*.

NGUGEN, W., AND MILLS, J. 1996. Multi-robot control for flexible fixtureless assembly of flexible sheet metal auto parts. *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2340–2345.

REDON, S., AND LIN, M. 2005. Practical local planning in the contact space. *ICRA*.

REDON, S., GALOPPO, N., AND LIN, M. 2005. Adaptive dynamics of articulated bodies. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH) 24*, 3.

RODRIGUEZ, S., LIEN, J., AND AMATO, N. 2006. Planning motion in completely deformable environments. *Proc. of IEEE Int. Conf. Robot. Autom. (ICRA)*.

SAHA, M., AND ISTO, P. 2006. Motion planning for robotic manipulation of deformable linear objects. In *ICRA*.

SUN, D., SHI, X., AND LIU, Y. 1996. Modeling and cooperation of two-arm robotic system manipulating a deformable object. *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2346–2351.

WILHELMSEN, D. R. 1976. A nearest point algorithm for convex polyhedral cones and applications to positive linear approximations. *Mathematics of computation*, 48–57.