

# ROBOTIC PATH PLANNING AND PROTEIN MODELING\*

Lydia E. Kavraki

This work is produced by The Connexions Project and licensed under the Creative Commons Attribution License †

## Abstract

This module explains how to model a protein as a robotic manipulator and introduces the robotic path planning problem and algorithms for solving it, including the probabilistic roadmap method. Variations of the Probabilistic Roadmap Method have been employed for problems related to protein motion.

## Topics in this Module

- Proteins as Robotic Manipulators (Section 1: Proteins as Robotic Manipulators)
- Robotic Path Planning (Section 2: Robotic Path Planning)
- Sampling-Based Planners for Proteins (Section 2.3: Sampling-Based Path Planning)

## 1 Proteins as Robotic Manipulators

In the modules on protein kinematics<sup>1</sup> and inverse kinematics<sup>2</sup>, it was shown that, structurally and kinematically, proteins are very similar to a class of **robotic manipulators** consisting of arms connected by revolute joints. Because of this analogy, in the late 1990s, some robotics researchers began to speculate that methods developed for use with robots might also be applicable to the study of proteins. For the remainder of this module, the analogy between robots and proteins will be explored, and then a class of robotics algorithms, called path planners, that will be adapted in a later module to use with proteins, will be introduced.

Recall that a protein is a chain of amino acid residues. Each residue contributes two rotatable dihedral angles, designated  $\phi$  and  $\psi$ , to the main chain of the protein, and may additionally have a side chain with up to five rotatable dihedral angles. Under the **rigid geometry** simplification, these rotatable dihedral angles are the only degrees of freedom available to the protein.

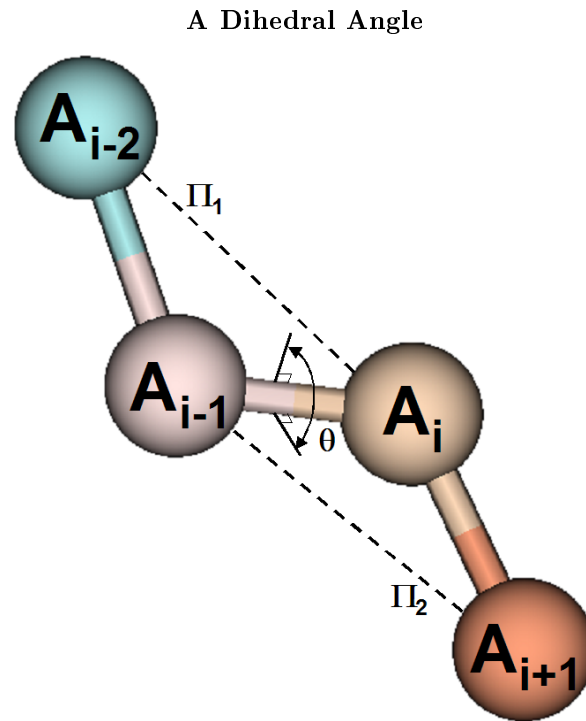
---

\*Version 1.22: Mar 10, 2007 7:45 pm US/Central

†<http://creativecommons.org/licenses/by/1.0>

<sup>1</sup>"Representing Proteins in Silico and Protein Forward Kinematics" <<http://cnx.org/content/m11621/latest/>>

<sup>2</sup>"Protein Inverse Kinematics and the Loop Closure Problem" <<http://cnx.org/content/m11613/latest/>>



**Figure 1:**  $\pi_1$  is the plane uniquely defined by the first three atoms  $A_{i-2}$ ,  $A_{i-1}$ , and  $A_i$ . Similarly,  $\pi_2$  is the plane uniquely defined by the last three atoms  $A_{i-1}$ ,  $A_i$ , and  $A_{i+1}$ . The dihedral angle,  $\theta$ , is defined as the smallest angle between these two planes<sup>3</sup>. You can read more about the angle between two intersecting planes<sup>3</sup>.

If we replace each bond by a rigid bar and each rotatable dihedral by a revolute joint, we can build a robotic linkage kinematically equivalent to a protein under rigid geometry.

## 2 Robotic Path Planning

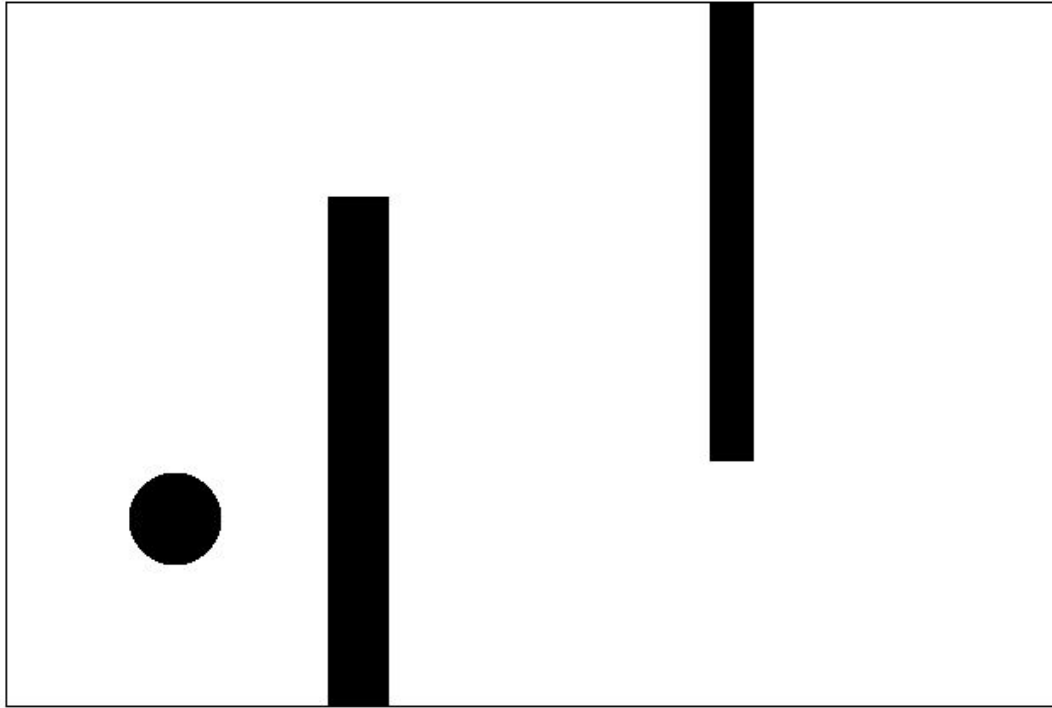
### 2.1 Background

#### Terms from robotic path planning

- **Work space:** The work space is the geometric space in which a robot operates. It consists of obstacles and empty space that may be occupied by the robot.
- **Configuration:** A configuration of the robot is a full description of the robot's state, including its position, orientation, and the states of any internal degrees of freedom (such as revolute joint angles).
- **Collision:** A configuration is said to be in collision if any part of the robot overlaps with either another part of the robot or with a work space obstacle.
- **Free:** A configuration is said to be free if it is not in collision.
- **Configuration space (C-space):** The space of all configurations of the robot, annotated by whether the robot is in collision or free at each configuration.
- **Free space:** The space of all free configurations.

<sup>3</sup><http://mathworld.wolfram.com/Plane.html>

### Work Space, Two-Dimensional Disc Robot



**Figure 2:** The work space for a simple robot. The robot is a two dimensional disc. Configurations are therefore  $(x,y)$  pairs. If the robot were not circular, an orientational degree of freedom would also need to be specified.

### Configuration Space, Two-Dimensional Disc Robot



**Figure 3:** The configuration space corresponding to the above work space. Each point now corresponds to a configuration of the robot, rather than a real point in space. To see how it was constructed, choose any point on the robot as a reference. Now imagine the curves swept out by that point as the robot slides along the surface of the obstacles and along the border of the work space. Every configuration closer to the obstacles than this curve is in collision, and every configuration farther than the curve is free.

In the above figures, the configuration space is two dimensional because the robot has two degrees of freedom. If the heading of the robot mattered (i.e., if the robot were not circular), then a configuration would consist of a position and an orientation. The configuration space would therefore be three dimensional. If the robot had a rotatable joint, this would add another degree of freedom and another dimension to the C-space.

#### 2.2 The Path Planning Problem

The robotic path planning problem is, given a robot, a work space, and starting and goal configurations for the robot in the work space, find a collision-free path for the robot from the starting configuration to the goal, if one exists. Otherwise determine that no such path exists. An extensive introduction to the path planning problem and existing solutions may be found in [1].

Early approaches to path planning included:

- Construction of visibility graphs between the vertices of C-space obstacles.
- Decomposition of the C-space, effectively into subproblems.

- Potential field methods, in which the goal exerts an attractive force on the robot, and the obstacles exert repulsive forces.

The first two methods scale poorly with the dimensionality of the C-space, since the complexity of the C-space affects their run time. Potential fields are subject to local minima. A robot moving down the potential gradient might get stuck in a potential well before it reaches the global potential minimum at the goal.

### 2.3 Sampling-Based Path Planning

One solution to the scalability problem was to find methods whose run time does not depend on the dimensionality of the C-space, but on some other factor. This led to **sampling-based** path planning. Rather than making some explicit analysis of the whole C-space, sampling based planners built their representation of C-space by sampling random configurations and using a fast collision checker to determine whether they are in collision.

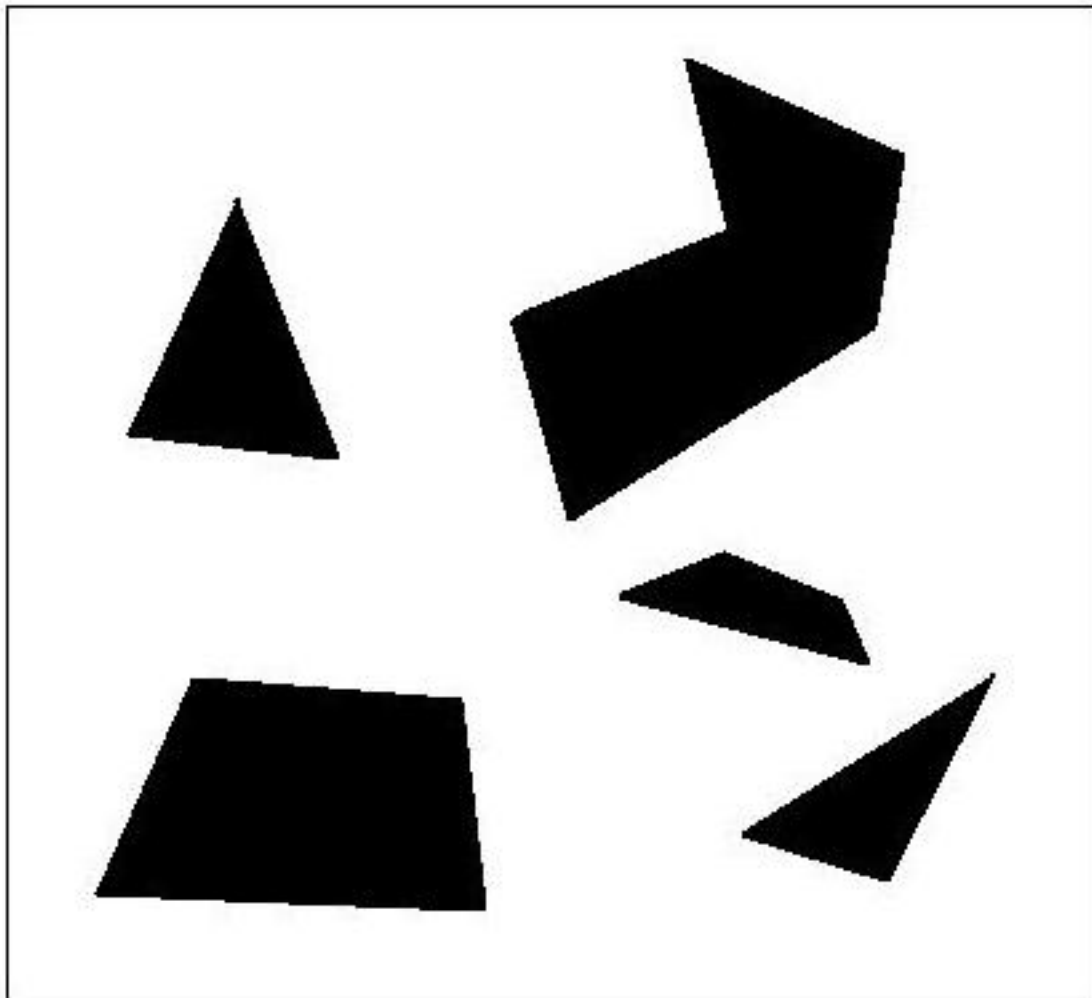
The basis of many modern sampling-based planners is the **Probabilistic Roadmap Method (PRM)**[2]. Although the implementation details can become complicated, the basic algorithmic framework is quite straightforward and easy to understand.

#### The PRM algorithmic framework:

- Randomly sample a large number of points in C-space, keeping any that are not in collision. This creates a point set in C-space.
- Using a **local planner**, attempt to connect pairs of samples that are relatively close to each other by thoroughly sampling and collision checking configurations between them. This creates a graph data structure called a roadmap.
- To query the roadmap, first attempt to connect the start and goal configurations to the existing graph. If that is successful, search the graph for a path from start to goal using any standard graph search method (often A\*).

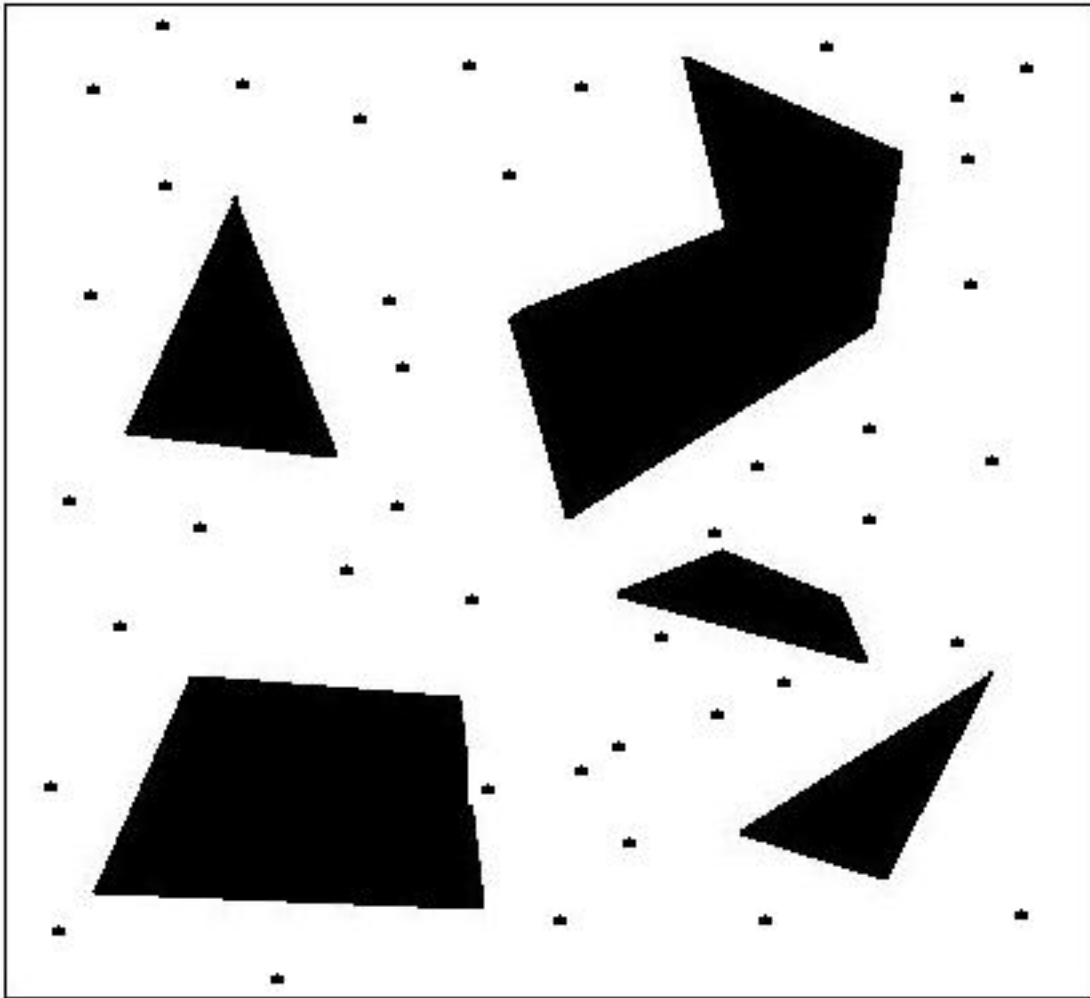
PRM implementations vary in terms of how the points are sampled—remember that random does not mean uniformly at random—as well as in how the local planner attempts to connect nearby configurations.

### Configuration Space for Path Planning



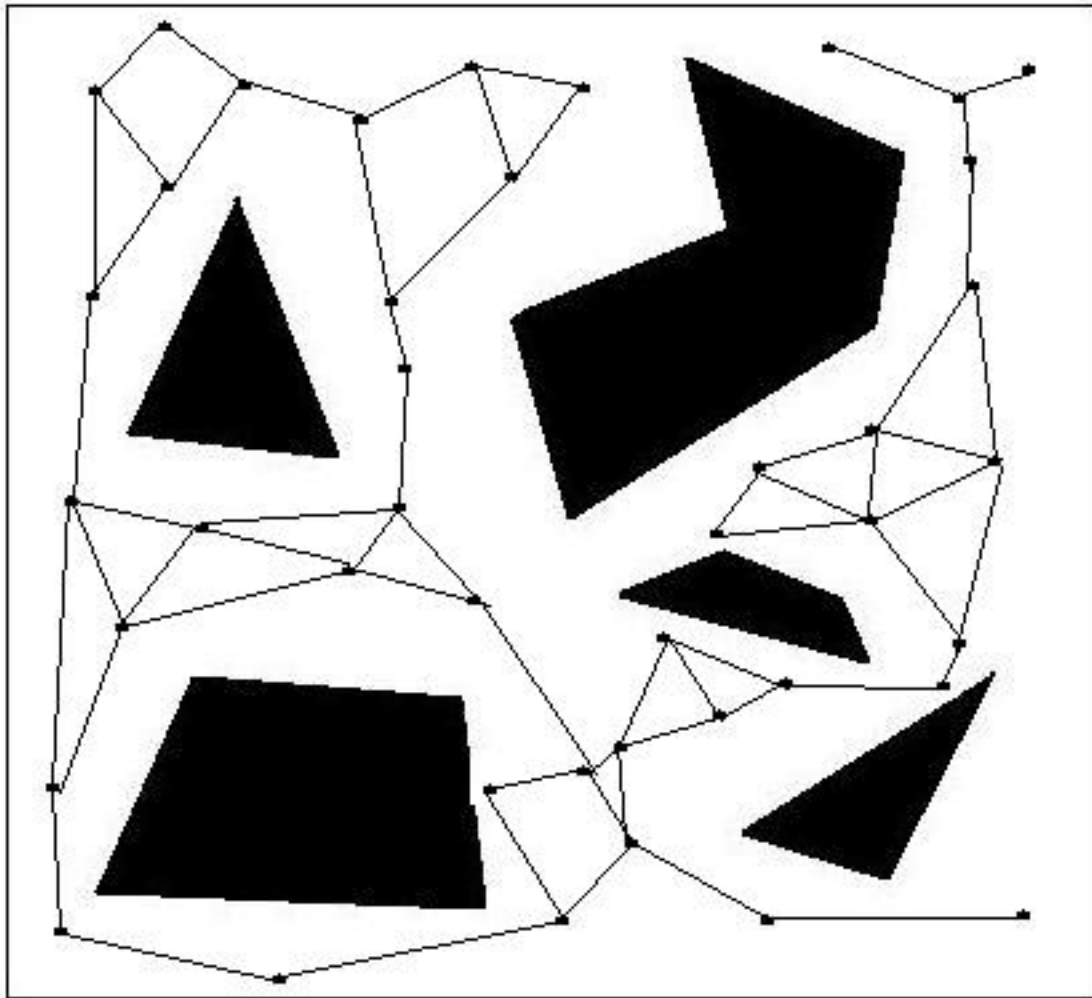
**Figure 4:** A configuration space in which we wish to build a Probabilistic Roadmap.

## Sampling



**Figure 5:** Random points in the configuration space are tested for collision with obstacles. Collision-free points are kept.

## Connection

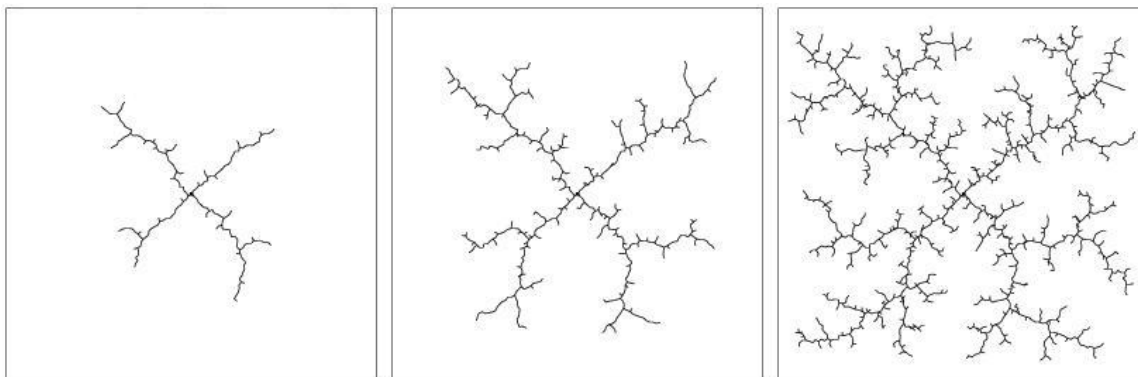


**Figure 6:** The roadmap is constructed by connecting nearby samples. Many collision checks are made along each edge to ensure that the connection is legitimate.



planning query, and build a tree data structure of samples growing away from it, with a bias toward the goal configuration.

### Operation of a Tree-Based Planner



**Figure 8:** Tree-based planners start at a point and grow outwards, covering the configuration space more and more densely as time goes on.

Single-query planners come in three basic types, each of which has been subject to numerous variations and enhancements since its original development:

#### Types of Single-Query Planners

- **Expansive Spaces Trees (ESTs):** At each step of building an EST, a node of the tree is selected for expansion. Nodes in more sparsely sampled parts of the configuration space are more likely to be chosen, and some bias may also be present for nodes closer to the goal. A number of samples are made in the vicinity of this node, and those that can be connected to it are added to the tree.
- **Rapidly-exploring Random Trees (RRTs):** At each step of RRT-building, a random point is selected in the configuration space, with some bias toward the goal configuration. The nearest node in the tree is found, and a path is created from it toward the random point, either for a set distance or until an obstacle is encountered, whichever comes first. The final collision-free configuration on this path is added to the tree.
- **Path-Directed Subdivision Trees (PDSTs):** PDSTs store edges and branch points as their primitive data rather than nodes. It also maintains a cell decomposition of the configuration space and assigns paths to cells. At each step of building the tree, an edge is deterministically selected based on an estimate of how well-sampled its surroundings are (using the cell decomposition), and a random point on the edge is selected for branching. The old edge is divided in two, a new edge is added at the branch point, and the cell decomposition is updated. Thus, the tree expands outward from its starting point, steered toward less well-sampled regions by the cell decomposition. PDSTs are well-suited for robotics problems with realistic and complex physics.

### 3 Sampling Based Planners for Proteins

With adjustments, one can apply sampling-based robotic path planning to study protein motion. First and foremost, the configuration space, which sorts conveniently into occupied and free configurations, must be replaced by the more complicated molecular **conformation space**. Each conformation of a molecule has a **free energy**, which depends on chemical and physical interactions between its component atoms and

those of any other molecules (such as solvent molecules) that may be nearby. This free energy is related to the probability of a protein being in a conformation. Thus, instead of dealing with the binary problem of colliding and free conformations, the conformation space is one of continuously varying probabilities. This problem may be simplified by the use of energy cutoffs, but it is difficult to decide, for any given problem, what energy is so high that a conformation is effectively in collision. Free energy and how it is incorporated into the study of molecular motion is discussed in more detail in *Motion Planning for Proteins: Biophysics and Applications*<sup>4</sup>.

### Recommended Reading

Choset, Howie, Kevin M. Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia E. Kavraki and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA: MIT Press, 2005. Chapter 1 for introduction to robotics and analogy between robots and biomolecules. Chapter 7 for a summary of sampling-based planners.

## References

- [1] Kevin M. Lynch Seth Hutchinson George Kantor Wolfram Burgard Lydia E. Kavraki Choset, Howie and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Cambridge, MA, 2005.
- [2] Svestka P. Latombe J.-C. Kavraki, L.E. and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12:566–580, 1996.

---

<sup>4</sup>"Motion Planning for Proteins: Biophysics and Applications" <<http://cnx.org/content/m11449/latest/>>