HUMBOLDT-UNIVERSITÄT ZU BERLIN

Mathematisch-Naturwissenschaftliche Fakultät I
Institut für Biologie

Diplomarbeit
zum Erwerb des akademischen Grades
Diplom-Biophysiker

Using tree-based robot motion planning algorithms for protein loop closure

vorgelegt von

Florian Kamm
geboren am 14.05.1979 in Ulm

angefertigt im "Robotics and Biology Laboratory (RBO)"
am Institut für Technische Informatik und Mikroelektronik,
Technische Universität Berlin

Berlin, den
15. Dezember 2010

Betreuer:
Prof. Oliver Brock, Robotics and Biology Laboratory
Prof. Hanspeter Herzel, Molekulare und zelluläre Evolution

Using tree-based robot motion planning algorithms for protein loop closure

Florian Kamm

December 15, 2010

Abstract

Protein loop closure is a common problem in protein structure prediction. Recently published results are an evidence for the progress in the accuracy of the predictions and the efficiency of loop closure methods. Special attention is given to the conformational sampling for closed loop conformations. We propose a completely novel approach to this problem inspired from robotics using a mechanistic description of the loop chain and a motion planning technique.

The Jacobian of a kinematic chain is exploited for the mechanistic description of a protein loop. The transpose of the Jacobian relates end-effector forces to joint torques and finally to angle increments. Self-motions of the kinematic chain due to its redundancy in the number of degrees-of-freedom (DOF) are used to minimize an energy function. An iterated motion scheme is derived based on that mechanistic description.

The randomized motion planning technique that is applied to the problem of protein loop closure is based on Rapidly-exploring Random Trees (RRT). A motion planning algorithm combining a RRT with a local planner like the iterated motion scheme presented is developed.

Zusammenfassung

"Loop closure" von Proteinen ist ein allgegenwärtiges Problem bei der Proteinstrukturvorhersage. Kürzlich veröffentlichte Forschungsergebnisse belegen den Fortschritt in der Genauigkeit von Vorhersagen und der Effizienz von "loop closure"-Methoden. Besondere Aufmerksamkeit ist auf das "Sampling" von Konformationen geschlossener "loops" gerichtet. Wir schlagen einen völlig neuartigen Ansatz für dieses Problem vor, inspiriert von Methoden aus der Robotik, der auf einer mechanistischen Beschreibung der "loop"-Kette und einer Pfadplanungsmethode basiert.

Die Jacobi-Matrix einer kinematischen Kette wird für eine mechanistische Beschreibung von "protein loops" benutzt. Die Transponierte der Jacobi-Matrix bringt Kräfte, die auf den Endeffektor wirken, in Relation zu Drehmomenten und schließlich zu Winkelinkrementen an den Gelenken. "Self-motions" der kinematischen Kette, resultierend aus der Redundanz in der Anzahl an Freiheitsgraden, werden ausgenutzt, um eine Energiefunktion zu minimieren. Ein iteratives Bewegungsschema basierend auf dieser mechanistischen Beschreibung wird hergeleitet.

Die randomisierte Pfadplanungtechnik, die auf das "protein loop closure"-Problem angewendet wird, basiert auf "Rapidly-exploring Random Trees (RRT)". Ein Pfadplanungsalgorithmus, der einen RRT mit einem lokalen Planer wie dem präsentierten iterativen Bewegungsschema kombiniert, wird entwickelt.

Contents

Figures

Acknowledgments

At first, I would like to thank Prof. Hanspeter Herzel. Without his readiness and confidence to supervise an external work, this thesis would not have been possible at all. I hope that his engagement will be fruitful for me and future students.

I also thank Prof. Oliver Brock for making the work on this thesis possible and for helping me with sophisticated problems in situations when I was desperate of never finding a solution. I am also very grateful that it was possible to work in direct contact with my colleagues: Ines Putz, Nasir Mahmood, Michael Schneider, Georgios Fagogenis, Dov "Dubi" Katz and Clemens Eppner. Nobody gave me ever the feeling of being a student visiting the lab for a limited time. So many many thanks to the "Robotics and Biology Lab (RBO)" team.

George helped me in numerous situations where my knowledge in robotics techniques and rigid body physics was not sufficient enough. Conversations with Dubi were always very productive. Especially helpful were the discussions with the "bio people", Ines, Nasir and Michael. I profited wealthy from the thorough discussions and the cooperation between me and Ines at the beginning of my work at the lab and during the end. At this point many thanks to Nasir for the introduction in using the computer cluster and our interesting conversations. Finally, thank you Michael for all the discussions and the convenient atmosphere in our office room.

My final thank is due to my friend Jana Rother for sharing my apartment with me for nearly eight years now and for suffering with me during the last weeks of this writing.

1 Introduction

During the last decade, several approaches have been published that apply methods and techniques from robotics to problems arising in structural biology and especially in protein structure prediction. The methodology that is presented in this thesis is also inspired from robotics techniques that are applied to a widely encountered problem in protein structure prediction, specifically to the problem of predicting protein loop structures in a known environment. The so-called protein loop closure problem has drawn a lot of interest by researchers world wide during the last years. The approach presented in this thesis exploits methods from robotics to sample the conformation space for feasible loop structures and to generate directed motions of a perturbed protein loop chain for closing the gap between two fixed amino acid residues.

At first, the protein loop closure problem is accurately defined and the relevance of loop closure methods for protein structure prediction is discussed. Then, the functional role of protein loops and the importance to accurately predict their structures are exposed from a biological point of view. Finally, the motivation for the presented methodology is outlined in the context of existing methods from literature.

1.1 Biological relevance of Protein Loop Closure

The ability to predict the native three-dimensional structure of a protein from its amino acid sequence is and will remain the most challenging problem in structural biology. A vast amount of methods addressing this folding problem has been developed so far, utilizing different computational approaches and information.

homology modeling

Comparative or homology modeling techniques search the Protein Data Bank (PDB) (Berman *et al.*, 2000) of known protein structures for structures with an

amino acid sequence that is homologous to a target sequence by using sequence alignment techniques. These can be used as templates in protein structure prediction to determine the structure of the target sequence. However, homologous proteins differ in regions where the structure has not been conserved during evolution, corresponding to gaps or insertions in sequence. Theses sequences usually represent loop regions connecting secondary structure elements or refer to highly flexible regions of the protein. Often loop regions of protein structures determined by X-ray crystallography or Nuclear Magnetic Resonance (NMR) are of poor resolution due to the high conformational flexibility of the loop chains. In crystallography, it is therefore necessary to refine loop structures that could not be determined with sufficient accuracy. So, *ab initio* or *de novo* methods are used either if no template structure can be identified or to predict the structure of protein loops in the common situation that the rest of the protein structure is known. Thus, the problem of modeling protein loops is a quite common task in crystallography and homology modeling as well as in *ab initio* protein structure prediction.

ab initio methods

loop modeling

Despite the short length of loops relative to the sequence length of the whole protein, modeling protein loops is challenging not only because of the huge number of feasible conformations, but mainly because of the geometric constraints imposed on them by the fixed N- and C-terminal anchor residues of the protein backbone preceding and following the loop region in question. Thus, loop modeling requires to connect protein backbone segments that are regarded fixed in space with a reasonable loop segment that satisfies the so called loop closure constraints. The protein loop closure problem is then identified as closing the gap between two fixed backbone segments with a native-like loop structure that satisfies the closure constraints (Kolodny, 2005).

backbone anchors

protein loop closure problem

Many problems that arise in loop modeling are the same that protein structure prediction has to cope with, it is just a matter of scaling, i. e. in each case extensive conformational sampling for the most feasible structure and structure refinement on the basis of energy or scoring functions are necessary. So, loop modeling can be interpreted as a mini protein folding problem (Fiser *et al.*, 2000). Like in protein structure prediction, both *ab initio* and database-driven approaches are available that try to solve the protein loop closure problem. Several hybrid methods have also been described in literature that combine these two fundamental approaches. (Fiser *et al.*, 2000; Mönnigmann and Floudas, 2005; Soto *et al.*, 2008, for an overview of existing techniques)

Loops are not only of general interest for structure refinement in crystallography and for protein structure prediction methods. They play also an important role for

protein docking by contributing to active and binding sites of proteins. Protein loops are also relevant for surface recognition as loops are often exposed on the surface of the structure, for designing antibodies or modeling ion channels. Furthermore, studying the conformational changes of protein loops in atomic detail is important to understand its contribution to functional changes of the whole protein in general (Soto *et al.*, 2008; Fiser *et al.*, 2000).

1.2 Existing methods and motivation

Various methods for solving the protein loop closure problem originating from very different research fields and disciplines have been developed over the last decade (Cortés and Siméon, 2005; Zhu *et al.*, 2006; Fiser *et al.*, 2000; Canutescu and Dunbrack, 2003; Mandell *et al.*, 2009a; Lee *et al.*, 2010; Kolodny, 2005). Among them there are analytical and iterative closure approaches as well as database search techniques that are capable of generating loop conformations that fulfill the loop closure constraints imposed on initially perturbed loop structures.[1]

In order to find low-energy loop conformations close to the native state it is necessary to sample the conformation space of the loop regions for appropriate closed-chain structures. These structures are subsequently refined and filtered for the most feasible conformations by energy or scoring functions. Ideally, the computational sampling is performed until it is likely that structures close to the native state could be found. Thus, a computational loop modeling approach that is able to predict native-like conformations consists of efficient conformational sampling, a closure technique to solve for the loop closure constraints and of refinement methods (Monte Carlo Simulated Annealing (Kirkpatrick *et al.*, 1983), relaxation by energy minimization, ...).

The conformational sampling can be done either by exploiting the PDB database of known protein structures or by choosing the dihedral angles ϕ, ψ of the protein backbone according to a predefined probability distribution. To enforce the loop closure constraints the conformational sampling is commonly accompanied by a loop closure technique. The resulting conformations are further optimized and ranked on the basis of an energy or scoring function.

Cyclic Coordinate Descent

The Cyclic Coordinate Descent (CCD) algorithm (Canutescu and Dunbrack, 2003) as implemented in Rosetta (Rohl *et al.*, 2004) repeatedly inserts small structure fragments generated from the PDB to sample the conformation space and subse-

[1] so called loop decoy sets

quently closes the open loop by applying a robotics-inspired technique. This method as well as other database-driven approaches depend on the quality and accuracy of known protein and loop structures as the fragments are generated from them. Thus, the conformational sampling as performed by CCD in Rosetta is restricted by the accuracy of the assigned loop structures in the PDB.

Kinematic Closure

The Kinematic Closure (KIC) approach (Mandell *et al.*, 2009a) is claimed to be of sub-angstrom accuracy in the rmsd value between a crystallographic and a predicted loop structure. The prediction accuracy is achieved by a random perturbation of small segments of the loop chain according to a Ramachandran map and an analytical closure technique. The gain in accuracy when compared with CCD is claimed to be the result of an enhancement of the conformational sampling (Mandell *et al.*, 2009b).[1] However, the analytical closure method used by KIC is itself restricted to loop segments of a fixed maximum size.

We propose a new approach that is neither restricted on the length of the loop or of loop segments nor on the accuracy of extracted fragments from the PDB. Our approach tries to improve the conformational sampling of feasible loop structures by applying robot motion planning algorithms based on Rapidly-exploring Random Trees (RRT) (LaValle, 1998) as a search method. The problem of finding loop conformations that fulfill the loop closure constraints is interpreted as the problem of finding trajectories of the end-effector of a robot manipulator[2] moving step-wise from an initial position to a goal position. Consequently, the end-effector of the manipulator is represented by the *C*-terminal residue of the open loop chain, whereas the *N*-terminal residue is considered as the base of the manipulator fixed in space. The RRT is then build up incrementally by randomly sampling conformations towards which the RRT is expanded. New nodes are added until the goal position and orientation of the end-effector have been reached, i.e. a conformation has been found that satisfies the loop closure constraints within a given tolerance. The protein loop is then considered to be closed and further refinement of the structure may follow.

Due to the property of a RRT to expand towards unexplored regions[3] of the search space and the application of a transition test (Jaillet *et al.*, 2008) we suppose that our new approach has the capabilities of generating native-like loop conformations in a computationally efficient way. Hereby, the transition test restricts the exploration

[1] A direct comparison between these two methods is possible because both have been implemented in Rosetta.

[2] in this context the manipulator is considered as a robot arm with revolute joints and rigid links

[3] a Voronoi bias as discussed later

of the RRT to regions from where the goal can be reached with high probability in a reasonable number of iterations.

The outlined methodology differs from other methods in the way how the conformational sampling is biased. Both the conformational sampling of CCD and KIC are biased by exploiting structural knowledge obtained from known structures in the PDB. In contrast, the conformational sampling of our approach is biased towards unexplored regions of the space and by an acceptance test based on the evaluation of a knowledge-based energy function.

2 Methods & Techniques

First, the different representations of the protein structure are reviewed in order to derive the representation of the protein loop as a kinematic chain. Then, several important concepts from the field of robotics are introduced that are used later for the derivation of the loop closure procedure. After defining the problem of loop closure in detail, a short outline follows how our approach tries to solve it. The most important protein loop closure methods and concepts reviewed from literature are presented. Finally, the methodology on which our approach is based is elaborately described and the final loop closure procedure is derived.

2.1 Representations of protein structure

The structure of a protein can be described in different ways. The representation of all atom positions by Cartesian coordinates is convenient when a physical force field is applied to the structure or electrostatic interactions are studied. Internal coordinates, however, represent the protein structure in a chemical way in terms of bond length and bond angles. The representations can also differ in the way how the protein side chains are described: *full-atom* means that every atom of the side chains is explicitly modeled, whereas the *lollipop model* interprets the side chains as spheres of variable radii (Levitt, 1976). The latter is also known as the centroid representation. In addition, even more course-grained representations have been described in literature, from which the description by torsion angles is discussed in detail below.

2.1.1 Cartesian coordinates

In Cartesian space each atom position is described by (x, y, z)-coordinates. Therefore, the number of degrees-of-freedom (DOF) of a protein structure in Cartesian space is

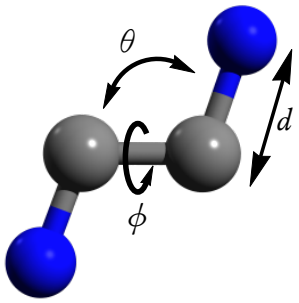

Fig. 2.1: The position of the rightmost atom is uniquely specified relative to the three previous atoms by the three internal coordinates: bond length, bond angle and torsion angle.

three times the number of atoms. When distance-dependent energy functions are applied in structure prediction methods the coordinates of the atom positions in Cartesian space are required. Also many file formats used in the field of molecular modeling like `.pdb` or `.mol2` are based on Cartesian coordinates. Thus, conversion into Cartesian coordinates from other structure representations is often mandatory, e. g. for energy minimization of distance-dependent potentials, or due to relaxation and folding methods that operate in Cartesian space like several "movers" as implemented in the well-known protein modeling framework Rosetta (Schueler-Furman *et al.*, 2005).

2.1.2 Internal coordinates

Internal coordinates describe the position of each atom relative to the other atoms in terms of a bond length, a bond angle and a torsion (dihedral) angle. So, the position of an atom is uniquely defined by placing it in a distance of a bond length away from the previous atom, rotating it around the bond angle formed by the atom itself and two of the previous atoms and finally rotating it around the torsion angle formed by the atom and three of the previous atoms, see Figure 2.1. The torsion angle is thereby defined as the angle between the planes formed by the first three and the last three atoms. Thus, each atom is represented by three internal coordinates, except the first three atoms in a chain. The first atom in a chain does not have any internal coordinates because it can freely placed in space, the second atom is placed in a distance of a bond length with respect to the first, and the position of the third atom is determined by a distance and a bond angle. From that, the total number of degrees-of-freedom of a protein described by internal coordinates is three times the

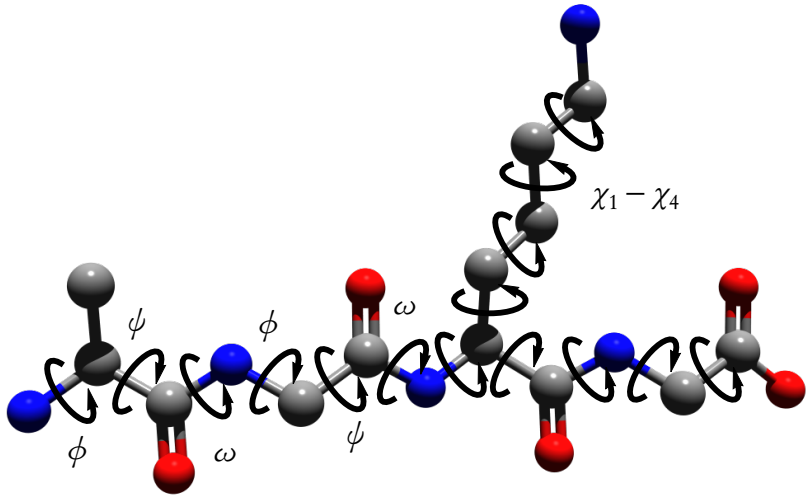

Fig. 2.2: Torsion angles representation. Bond lengths and bond angles are kept fixed, so the ϕ, ψ angles represent the DOF of the protein backbone.

total number of atoms reduced by six[1]. Internal coordinates are commonly written as a Z-matrix (Leach, 2001).

2.1.3 Conversion from internal to Cartesian coordinates

It is a quite common task for molecular modeling applications to convert between a torsion space representation given by the internal coordinates and Cartesian space. If, for instance, a Rosetta mover perturbs the protein structure by altering torsion angles, the Cartesian coordinates of atom positions are updated automatically so that the *in silico* representation of the protein is always consistent. This instantaneous conversion enables Rosetta to apply methods operating both in Cartesian space and torsion space.

Many procedures are available to achieve this: the General Rotation method, the Rodrigues-Gibbs-Formulation (RG), Quaternion Rotations, the Atomgroup Local

[1]three degrees-of-freedom each for the global position and orientation of the protein, that the internal coordinates representation does not take account for

Frames method and the Natural Extension Reference Frame (NeRF) method used by Rosetta (Choi, 2006; Zhang and Kavraki, 2002; Parsons *et al.*, 2005).

2.1.4 Torsion angles representation

A common way to represent proteins is by its torsion angles, where the bond lengths and bond angles are kept fixed at an ideal value. Changes in the conformation of a protein are then due to changes in the torsion angles only, i. e. by rotating around covalent bonds. Considering the bond lengths and bond angles invariable is a good approximation because changes in these parameters are usually very small (Engh and Huber, 1991). Also, this representation is computationally very efficient.

From a structural point of view the protein is built up by "building blocks" of peptide units (Brändén and Tooze, 1999; Cantor and Schimmel, 1980). A peptide unit is formed by all atoms in the chain segment from one C_α atom to the next C_α atom, without taking the side chains into account (see Figure 2.2). Thus, each C_α atom is part of two peptide units, except the first and the last. Due to the partial double-bond character of the peptide bond between the C' atom of one residue and the N atom of the next residue, the peptide unit is considered to be plane. Each peptide unit can either rotate around the $C_\alpha - C'$ covalent bond, or around the $N - C_\alpha$ covalent bond, where the first angle is denoted as ϕ and the latter as ψ (Figure 2.2). So, every amino acid residue is associated with two angles ϕ and ψ. For the sake of completeness, the third torsion angle by rotating around the $C' - N$ covalent bond (peptide bond) is commonly denoted as ω and fixed to 180°[1] due to the plane character of the peptide unit. The torsion angles of the side chains are termed χ_1-χ_4[2]. In this notation, the conformation of a protein backbone is completely described by the ϕ and ψ angles representing the degrees-of-freedom of the protein. Thus, the total number of degrees-of-freedom of a protein in torsion space representation is two times the number of amino acid residues.

ϕ, ψ angles

2.1.5 Kinematic chain of protein backbone

As we have seen in Subsection 2.1.4, the torsion angles representation of a protein structure explicitly underlines the chain-like properties of the protein backbone, either as a sequence of amino acid residues or as a concatenation of peptide units.

[1] $\omega = 0°$ corresponds to the cis conformation, $\omega = 180°$ to the trans conformation of the peptide bond, which is usually favored

[2] depending on the type of the amino acid there are one to four side chain torsion angles present

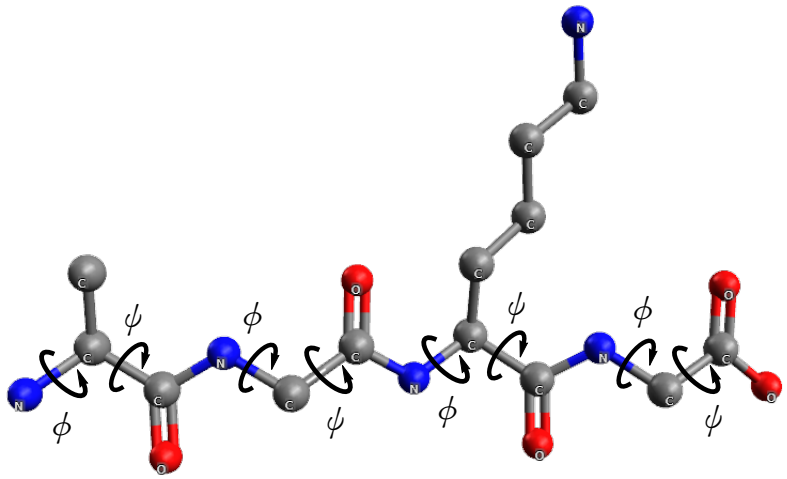

Fig. 2.3: Kinematic chain of a protein backbone. Rigid atom groups or protein links are separated by rotatable bonds representing revolute joints. The atoms are enumerated in order to easily identify the rigid atom groups as defined.

For simplification, the ω angles are kept fixed in trans-conformation corresponding to $\omega = 180°$. We will keep this assumption for the remainder of this thesis.

The covalent bonds $C_\alpha - C'$ and $N - C_\alpha$, around which the angles ϕ and ψ are defined, cut the protein backbone into two types of rigid groups (protein links): the *peptide group* includes the C'_{i-1}, O_{i-1} and N_i atoms between the ψ_{i-1} and ϕ_i bonds, whereas the *sidechain group* consists of the $C_{\alpha,i-1}$ and all attached side chain atoms except the hydrogen atoms between the ϕ_i and ψ_i bonds (Lotan, 2004). See Figure 2.3 for an illustration. The side chain atoms are included in the sidechain group though the χ angles are considered variable. However, variations to these angles do not contribute to a conformational change of the protein backbone.

peptide group

sidechain group

In terms of robotics notation the protein backbone can then be interpreted as a serial linkage of rigid links connected by revolute joints forming a kinematic chain. The rigid links correspond to either the peptide group or the sidechain group as introduced above and the revolute joints are represented by the rotatable bonds around the torsion angles ϕ and ψ. The mathematical aspects of a kinematic chain are described in more detail later.

In the following, we will make extensive use of the properties of the kinematic chain representation of a protein backbone. But first of all we will outline some

kinematic chain

fundamentals of robotics that are necessary for the further understanding.

2.2 Robotics fundamentals

A robot arm, called manipulator, is composed of rigid links and variable joints connecting the links. The robot hand is called end-effector and the mounting point of the robot manipulator the base. The composition of links and joints forms a kinematic chain as we have already seen in terms of the protein backbone. There are two types of joints: prismatic and revolute (rotatory). Prismatic joints allow a linear relative motion between two links, whereas revolute joints allow a relative rotation between two links. No prismatic joints are modeled for the kinematic chain of a protein backbone, so prismatic joints are not considered in the following. See Spong *et al.* (2006) or Craig (2005) for a good introduction into the field of robotics and for further information. In the next sections, we will mostly use the mathematical notation from Spong *et al.* (2006).

2.2.1 Robot modeling

configuration of manipulator

The configuration of a kinematic chain is specified by a set of values for the joint variables, hence the angles by which two connected links are rotated relative to each other around the joint axis. The term configuration commonly used in robotics is equivalent to the term conformation used in a biological context when specified as a set of torsion angles. For the remainder of the present thesis both terms are used interchangeably. The configuration space is then the set of all possible configurations, i.e. the permutation of all possible joint angle values. For conformations of proteins this space is consequently called conformation space.

configuration space

The vector of values of the joint variables is denoted by q. Then $q_i = \theta_i$ is the single joint angle value of the revolute joint i. Then, q has the dimension of the number of joints n of the manipulator. Thus, the dimension of the configuration space equals the number of degrees-of-freedom (DOF) of the kinematic chain and hence the number of joints n. In the three-dimensional space, three DOF are necessary for positioning and three DOF for the orientation of the manipulator. Thus, a minimum of six DOF is required for a manipulator to reach every point in a three-dimensional work space, the total volume that the end-effector is sweeping when the manipulator executes all possible motions. A manipulator with more than six DOF, i.e. composed of more than six links, is called redundant.

redundant manipulator

In the following, the axis of rotation of a revolute joint is denoted by z_i if the joint connects links i and $i+1$. The base of the manipulator is stationary fixed throughout this study.

2.2.2 Manipulator Jacobian

forward kinematics

Specifying the position and orientation of the end-effector given the joint variables, is known as the forward kinematics problem. The forward kinematics equations accomplish this task. To derive these equations a coordinate system $o_i(x_i, y_i, z_i)$ (local frame) is attached to each link i. The base frame $o_0(x_0, y_0, z_0)$ refers to the local frame of the base of the manipulator. Forward kinematics can then be expressed as the problem of finding the homogeneous transformation matrix

$$A_i(\theta) = \begin{bmatrix} R_i^{i-1} & o_i^{i-1} \\ 0 & 1 \end{bmatrix}$$

for each frame i. The matrices express the position and orientation of the frame i with respect to frame $i-1$ where

$$R_j^i = R_{i+1}^i \dots R_j^{j-1}$$

denotes to a 3×3 rotation matrix and

$$o_j^i = o_{j-1}^i + R_{j-1}^i o_j^{j-1}$$

homogeneous transformation

to a coordinate vector. Multiplying them together yields then the homogeneous transformation matrix

$$H = T_n^0 = \begin{bmatrix} R_n^0 & o_n^0 \\ 0 & 1 \end{bmatrix} = A_1(\theta) \dots A_n(\theta)$$

of the end-effector frame with respect to the base frame. The forward kinematics equations define a function that relates the positions and orientations in Cartesian space to the positions of the joints. As the manipulator executes a motion both the joint angles θ_i and the end-effector position o_n^0 and orientation R_n^0 are functions of time. Then, the Manipulator Jacobian[1] $J(\theta)$ of this function relates the linear and angular velocities of the end-effector to the joint velocities

manipulator Jacobian

$$\xi = J(\theta) \cdot \dot{\theta} \tag{1}$$

[1] partial time derivatives

with

$$\xi = \begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix} \quad \text{and} \quad J_n^0 = \begin{bmatrix} J_{v_n^0} \\ J_{\omega_n^0} \end{bmatrix}$$

where v_n^0 denotes the linear velocity and ω_n^0 the angular velocity vectors of the end-effector with respect to the base frame. $\theta = (\theta_1, \ldots, \theta_n)^T$ is the vector of joint angles and, consequently, $\dot{\theta} = \left(\dot{\theta}_1, \ldots, \dot{\theta}_n\right)^T$ the vector of joint velocities, $J_{v_n^0}$ and $J_{\omega_n^0}$ are $3 \times n$ matrices where n denotes the number of joints and hence the enumeration number of the end-effector. The superscript indicates that the components of the Jacobian and the velocities are expressed with respect to the base frame, whereas the subscript relates the velocity reference point to a physically real point on the end-effector (Orin and Schrader, 1984).

The Jacobian in the cross-product form (for revolute joints) is then derived as

$$J = \left[J_1 J_2 \ldots J_n\right] = \begin{bmatrix} J_{v_1} J_{v_2} \ldots J_{v_n} \\ J_{\omega_1} J_{\omega_2} \ldots J_{\omega_n} \end{bmatrix}$$

where the ith column J_i is determined by

$$J_i = \begin{bmatrix} J_{v_i} \\ J_{\omega_i} \end{bmatrix} = \begin{bmatrix} z_{i-1} \times (o_n - o_{i-1}) \\ z_{i-1} \end{bmatrix} \tag{2}$$

z_{i-1} is given by the third column of the rotation matrix of the forward kinematics equations for joint $i-1$ and is determined by

$$z_i = R_{i-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Herein, the superscript denoting that the components of J are expressed with respect to the base frame is omitted for simplicity. Note, that in this expression the velocity reference point is implicitly related to the origin of the end-effector frame, so there is no need for an additional subscript.

2.2.3 Motion Planning

In the context of protein loop closure we define the motion or path planning problem as the search for a path in task space to move the end-effector of the robot

manipulator to the goal position. A complete path describes then the motions of the manipulator from the initial configuration q_{init} to the goal configuration q_{goal} by discrete states. This path accounts for position and orientation of the end-effector but not for time, and hence not for velocities and accelerations. This motion is usually constrained because self-collisions have to be avoided and/or the motion itself is limited in some way, e.g. due to joint limits restricting a rotation to a valuable range of angles or other workspace constraints. The algorithms that are used in the following are not guaranteed to find a path, but easy to implement and they require only moderate computation time.

motion planning techniques

Various motion planning algorithms are available that all have pros and cons (LaValle, 2006). Among them there are methods using artificial potential fields (Spong *et al.*, 2006) and randomized methods like the Probabilistic Roadmap (PRM) method (Latombe *et al.*, 1996) or planners that use Rapidly-exploring Random Trees (RRT) (LaValle, 1998; LaValle and Kuffner, 2001).

RRT

2.3 Problem formulation

In *ab initio* protein structure prediction conformational sampling is regarded as a bottleneck, especially for the prediction of large and complex structures (Kim *et al.*, 2009). It has also been stated in another study about protein structure prediction using Rosetta that the energy function of Rosetta is probably good enough to sample native structures. Generally spoken, conformational sampling can be seen as one of the major drawbacks of most protein structure prediction methods.

Considering the protein loop closure problem as a mini protein folding problem, we assume that improving the conformational sampling will yield potentially more native-like protein loop structures. This assumption is also consistent with the conclusion of Mandell *et al.* (2009b) where the authors reduce the gain in accuracy of the calculated loop structures from improvements in conformational sampling though scoring errors cannot be completely ruled out. However, sampling protein loop conformations is even harder than sampling conformations of whole proteins because of the loop closure constraints that always have to be satisfied.

We propose a motion planning approach from the field of robotics using Rapidly-exploring Random Trees (RRT) for sampling conformations in the context of protein loop closure. A recently published algorithm called transition-based RRT (T-RRT) (Jaillet *et al.*, 2008, 2010) combines the classical RRT-approach with a cost function that restricts the exploration of the configuration space by rejecting unfeasible

T-RRT

configurations similar to the Metropolis test for Monte Carlo methods (Metropolis *et al.*, 1953).

The RRT is initialized with an initial loop conformation, e.g. a fully extended loop chain. The position and orientation of the goal is specified by the position and orientation of the fixed C-terminal anchor residue of the protein backbone. When the moving C-terminal residue of the open loop chain approaches this anchor residue up to a given tolerance, the loop is considered closed.

The motion of an open kinematic chain representing a protein loop conformation is due to a virtual task force[1] actuating on the end-effector pulling it into the desired direction. Exploiting the operational space formulation (Khatib, 1987) the applied virtual force results in torques on the joints and hence in changes to the angular accelerations of the links about the joint axes. Integrating the angular accelerations twice yields joint angle increments and finally a new conformation.

operational space formulation

However, the representation of a loop as a kinematic chain is redundant in its DOF as a loop chain usually has more than six DOF (a chain segment of three amino acid residues has exactly six DOF if the torsion angles ϕ, ψ are variable and ω is kept fixed). In case of redundancy there is an infinite number of joint motions that do not alter the position and orientation of the end-effector spanning so-called self-motion manifolds (Burdick, 1989). One of these self-motions can be selected in order to minimize a potential function (Khatib, 1990). We exploit the redundancy in DOF of the kinematic chain of a protein loop to minimize atom interactions with neighbor atoms based on a van-der-Waals potential function. Differently spoken, self-collision penalties of the protein loop chain can be minimized in this way.

self-motion manifold

An iterated motion due to applied task forces like described can then be used as a local planner for expanding the RRT towards randomly sampled conformations. The proposed conformation is added to the RRT as a new node if the transition test succeeds.

Closed loop conformations, i.e. those that satisfy the loop closure constraints, are added to a list. After the planner has stopped due to predefined stop conditions the predicted loop structures are further refined by optimization methods.

2.4 Related work

A lot of research has been done to solve the loop closure problem and recently published results indicate that it is going into the right direction (Shenkin *et al.*, 1987;

[1] also termed operational force

van Vlijmen and Karplus, 1997; Fiser *et al.*, 2000; Canutescu and Dunbrack, 2003; Cortés and Siméon, 2005; Zhu *et al.*, 2006; Mandell *et al.*, 2009a). For an overview, the most successful and promising methods from literature are shortly reviewed next. However, a thorough discussion of all methods proposed so far is beyond the scope of this thesis. See also Mönnigmann and Floudas (2005) for a good overview of loop prediction methods.

2.4.1 Randomized motion planning using Rapid Loop Generator (RLG)

Rapid Loop Generator (RLG)

The algorithm presented in (Cortés *et al.*, 2002; Cortés and Siméon, 2003, 2005) proposes an extension to randomized motion planning algorithms like PRM or RRT that is able to sample closed chain configurations, i.e. configurations that satisfy kinematic loop closure constraints. This is achieved by the Random Loop Generator (RLG) algorithm. A subchain of the loop chain is chosen so that it is non-redundant, called passive subchain. Consequently, the corresponding joint variables are called passive. The remaining joint variables, called active, form one or two active subchains. Then, the active joint variables span self-motion manifolds. RLG performs a random sampling of the active joint variables in a way so that the loop closure equations for the passive subchain could be solved. As the passive subchain is non-redundant, solutions for the closure constraints can be analytically calculated by exact inverse kinematics methods. The resultant configurations are then checked for collisions or filtered to satisfy further constraints.

This approach shares with our method the idea to exploit the redundancy of loop chains with more than six DOF. While this method samples self-motions for the active subchains only, our method uses the self-motions of the whole chain to minimize a potential function.

2.4.2 Tweak Methods

Random Tweak

The Random Tweak method (Shenkin *et al.*, 1987) starts with a random loop conformation and modifies all torsion angles of the open loop chain at once in each iteration. The iteration proceeds until the distance constraints between the atoms of the terminal residues of the loop chain have been satisfied, i.e., so that the chain can be rotated into place to close the loop. Lagrange multipliers are used to minimize the changes in the torsional angles while satisfying the distance constraints using the Jacobian of the distances. Due to the use of the Jacobian and its inverse or

pseudoinverse, the tweak methods tend to be numerically unstable.

The LOOPY algorithm (Xiang *et al.*, 2002) uses Random Tweak and Direct Tweak, a variant of Random Tweak that performs loop closure while additionally avoiding steric clashes. LOOPY uses Random Tweak to perform loop closure and Direct Tweak for minimization in torsion space. The LoopBuilder protocol (Soto *et al.*, 2008) combines LOOPY with a filter based on a scoring function and with an all-atom energy minimization method to predict native-like protein loop structures.

2.4.3 Modeller

modeller

Modeller uses an optimization-based approach of modeling protein loops in a given environment (Fiser *et al.*, 2000). This approach optimizes the positions of non-hydrogen atoms of a loop chain by minimizing a pseudo energy function. The optimization of the energy function is based on conjugate gradients combined with molecular dynamics (MD) and simulated annealing. The pseudo energy function exploits not only the physics of a loop chain in a fixed environment by using a molecular mechanics force field but also statistical preferences for dihedral angles and non-bonded atom interactions derived from known protein structures in the PDB.

2.4.4 Protein Local Optimization Program (PLOP)

PLOP

The Protein Local Optimization Program (PLOP) (Zhu *et al.*, 2006) builds up fragments from the fixed N-terminal and C-terminal anchor residues of the protein backbone that meet in the middle of the loop chain. The fragments are chosen from rotamer libraries for the torsion angles of the backbone, taking crystal contacts determined from crystallographic data into account. The original hierarchical sampling procedure is described in detail in Jacobson *et al.* (2004). When the fragments have been built up completely the fragment geometries have to be adjusted to close the loop chain. Those fragment pairs are identified that guarantee closure on basis of the distance between the C_α atoms of the meeting residues. The sampled loop structures are clustered to select the most feasible set that is being further optimized by using a molecular mechanics force field and a continuum solvation model.

Several modifications to PLOP have been proposed in literature, varying the force field, the solvation model or the sampling procedure. Zhu *et al.* (2006) added an additional hydrophobic term to the force field used by Jacobson *et al.* (2004). Sellers *et al.* (2008) augmented PLOP with a method to optimize the protein side chains surrounding the loop region that is being closed. Felts *et al.* (2008) replaced

the continuum solvation model used by PLOP by an alternative one including a novel non-polar hydration free energy model.

2.4.5 Kinematic closure (KIC)

Kinematic Closure (KIC)

The Kinematic Closure (KIC) method (Mandell *et al.*, 2009a) has been already introduced in Section 1.2. KIC randomly chooses three C_α atoms as pivots involving six torsional angles. Consequently, the remaining C_α atoms are called non-pivot atoms. The non-pivot torsional angles are randomly sampled from a Ramachandran map, which opens the chain. KIC then determines analytically all values for the six pivot torsions that close the loop (Go and Scheraga, 1970; Wedemeyer and Scheraga, 1999) while simultaneously sampling the non-pivot torsional angles using polynomial resultants (Coutsias *et al.*, 2004, 2006).

This method has been integrated into Rosetta and validated against different loop decoy sets, yielding loop structures of sub-angstrom accuracy when compared to crystallographic loop structures (Mandell *et al.*, 2009b). We will also compare our method with KIC, which is relatively easy because both have been implemented in Rosetta.

(Lee *et al.*, 2010) combined the analytical loop closure technique used by KIC with conformational sampling based on fragment assembly of generated structure fragments from the PDB. Furthermore, a torsional energy function is minimized simultaneously to the loop closure procedure.

2.4.6 Cyclic Coordinate Descent (CCD)

Cyclic Coordinate Descent (CCD)

The Cyclic Coordinate Descent (CCD) algorithm was originally developed in robotics to solve the inverse kinematics problem. In (Canutescu and Dunbrack, 2003) CCD has been adopted to the protein loop closure problem. There is also an implementation of CCD available in Rosetta which is used as the default loop closure method for *ab initio* protein structure prediction (Rohl *et al.*, 2004). The CCD algorithm for protein loop closure (Canutescu and Dunbrack, 2003) starts with an initial loop structure and iterates then over the torsion angles ϕ, ψ of the open loop chain from the *N*-terminal to the *C*-terminal end of the loop. One torsion angle is modified per iteration until the atoms of the moving *C*-terminal residue of the loop chain are superimposed on the atoms of the fixed *C*-terminal anchor residue of the backbone. Thereby, the angles are adjusted to minimize the sum of the squared distances between the atoms of the loop residue and the corresponding

anchor residue.

Usually, CCD is combined with a sampling and refinement procedure, e.g. a Monte Carlo method. The CCD algorithm has several advantages over competing methods: it is simple and hence easy to implement and relatively fast when compared to methods that yield similar accuracy of the predicted loop structures.

2.5 Iterative loop closure exploiting Jacobian transpose

CCD performs stepwise displacements of an open protein loop chain where one torsion angle is varied in each iteration as discussed in Subsection 2.4.6. We propose an alternative iterated motion scheme that describes a displacement of an open loop chain due to virtual task forces actuating on the end-effector of the corresponding kinematic chain that pull the chain into the desired direction (Jagodzinski and Brock, 2007). The motion scheme described exploits the operational space formulation introduced by Khatib (1987) to relate forces actuating on the end-effector to joint torques. The joint torques are finally mapped to joint angle increments. Thus, an elementary motion in the direction of the force corresponds to a variation of all torsion angles at once. Furthermore, the redundancy of a kinematic chain representing a protein loop can be used to perform a second task as discussed later.

2.5.1 Operational Space Formulation

Recall from Eq. 1 that the Jacobian relates the joint velocities to the operational or end-effector velocities. This equation can also be written in terms of infinitesimal small displacements of the end-effector δx and the joint angles $\delta\theta$

$$\delta x = J(\theta) \cdot \delta\theta$$

generalized force

A similar relationship can be derived (Khatib, 1987) between the generalized forces operating on the end-effector and the torques operating on the joints

$$\tau = J^T(\theta) \cdot F \tag{3}$$

where τ denote the joint torques and F the generalized end-effector or operational force. The end-effector force is also called task force if the force operating on the end-effector is due to the execution of a task.

task force

The generalized force takes the position and orientation of the end-effector into account so that it is not equivalent to a physical force. The generalized force can be

written in the form

$$F = \begin{bmatrix} n \\ f \end{bmatrix} \tag{4}$$

where f and n denote the force moment couple of a physical force operating on a rigid body as we will see in detail later.

2.5.2 Redundant manipulators

As we have seen the kinematic chain of a protein loop is usually redundant in its DOF. This means that there is an infinite number of joint angle displacements that can take place so that the configuration of the end-effector remains fixed. Differently spoken, the configuration of the end-effector can be determined with an infinite number of postures of the links composing the kinematic chain. This is equivalent to the description by self-motion manifolds (Burdick, 1989).

It can be shown that the displacements take place in the nullspace of the associated Jacobian. The matrix

$$\left[I - J(\theta)^T \bar{J(\theta)}^T \right]$$

defines the mapping to the nullspace associated with the transpose of the Jacobian $J(\theta)^T$ (Khatib, 1990). Here, $\bar{J(\theta)}^T$ denotes the transpose of the generalized inverse or pseudoinverse of the Jacobian. Thus, applying torques of the form

pseudoinverse

$$\left[I - J(\theta)^T \bar{J(\theta)}^T \right] \tau_0 \tag{5}$$

to the joints of a kinematic chain do not alter the position and orientation of the end-effector. There is an additional DOF associated with the nullspace of the Jacobian which can be exploited to minimize a potential function $V_0(\theta)$. This is done by selecting τ_0 as the gradient of this potential function

$$\tau_0 = -\nabla V_0(\theta) \tag{6}$$

2.5.3 Iterated motion in task space while minimizing energy

Combining Eq. 3 and Eq. 5 yields

$$\tau = J^T(\theta) \cdot F + \left[I - J(\theta)^T \bar{J(\theta)}^T \right] \tau_0$$

and with Eq. 6

$$\tau = J^T(\theta) \cdot F - \left[I - J(\theta)^T J(\bar{\theta})^T \right] \nabla V_0(\theta) \tag{7}$$

This relationship is exploited to derive an iterated motion scheme that will be applied in a protein loop closure procedure. In reach iteration angle increments for the torsion angles of a protein loop chain are calculated from the torques determined by Eq. 7. Adding the increments to the torsion angles of the chain determines the conformation of the protein loop after each iteration.

A protein loop is represented as a kinematic chain, where the base corresponds to its first link and hence to the first rigid group of the N-terminal residue of the loop. The choice of the end-effector link is arbitrary as long as a deviation to a goal position and orientation can be specified. Here, the end-effector is represented by the rigid atom group (protein link) formed of the C atom of the C-terminal residue of the loop chain and virtual N and C_α atoms extending the chain.

The generalized task force F actuates on the end-effector and generates joint torques τ according to the first term in Eq. 7. The main task is specified as moving the end-effector from an initial position and orientation to a goal position and orientation by applying an appropriate task force. Here, the position and orientation of the goal is determined by the position and orientation of the rigid atom group formed by the N, C and C_α atoms of the fixed C-terminal anchor residue of the backbone.

The generalized task force F that actuates on the rigid atom group representing the end-effector can be easily calculated. A position-dependent physical force f is defined that acts on the individual atoms of the rigid group. A force that acts on a rigid body can be replaced by a force and moment couple that acts on a arbitrarily chosen origin (Tipler and Mosca, 2007). If that origin is chosen as the origin of the global coordinate system, the generalized force F can be calculated by

force couples

$$F = \begin{bmatrix} n \\ f \end{bmatrix} = \begin{bmatrix} \sum_i n_i \\ \sum_i f_i \end{bmatrix} = \begin{bmatrix} \sum_i o_i \times f_i \\ \sum_i f_i \end{bmatrix} \tag{8}$$

where f_i denote the force acting on the ith atom of the rigid group and o_i the position of the atom Kazerounian *et al.* (2005).

The Jacobian can be easily calculated by using Eq. 2. If the forward kinematics equations are known, no further calculation will be needed. For computational

efficiency, the Jacobian is calculated column-wise from the base out to the end-effector.

The generalized task force F produces torques τ on the joints resulting in an angular acceleration around the joint axis due to

$$\tau = I \cdot \alpha$$

where α is the angular acceleration and I denotes to the matrix of moments of inertia which is considered to be identical to the identity matrix in course approximation. Double integration for elementary time steps yields then a vector of angular increments that is added to the previous torsional angles which corresponds to a motion of the end-effector in space.

This elementary move is accompanied by the minimization of an energy function in terms of a penalty for atoms that are too close to nearby non-bonded atoms. The second term in Eq. 7 corresponds to the minimization task which takes place in the associated nullspace of the Jacobian as we have seen previously.

The potential function $V_0(\theta)$ in Eq. 7 that is being minimized is represented by a van-der-Waals energy term accounting for non-bonded atom interactions. Calculating the negative gradients of the potential function yields joint torques τ_0 that, being projected onto the nullspace, do not alter the position and orientation of the end-effector.

The van-der-Waals energy function is evaluated for all rigid atom groups representing the links of the kinematic chain. Thus, the negative gradients of the energy function yield physical forces that act on the individual atoms of the protein links (Abe *et al.*, 1984; Wedemeyer and Baker, 2003). According to Eq. 8 these forces can be replaced by force and moment couples. Summing up these couples yield the generalized forces actuating on the protein links.

Joint torques in turn are calculated for each protein link due to Eq. 3 using the Jacobian of the corresponding joint and the generalized force as calculated before. Enumerating over all protein links, the torques actuating on each link are summed up to calculate the equivalent total torques τ_0 of the whole loop chain.

self-motion

To calculate the self-motion of the loop chain due to the minimization task the pseudoinverse or Moore-Penrose inverse $J(\bar{\theta})$ of the Jacobian is needed. The pseudoinverse can be efficiently calculated by applying a Singular Value Decomposition (SVD) on the Jacobian (Press, 2007). The second term in Eq. 8 projects the total joint torques onto the nullspace of the Jacobian which results in a self-motion of the protein loop chain minimizing the van-der-Waals energy function.

2.6 Conformational sampling by robot motion planning

We use a robot motion planning approach to search the conformation space for loop chains that satisfy the loop closure constraints. There are various randomized approaches available that are suited for motion planning problems in high-dimensional spaces. We favor an approach based on Rapidly-exploring Random Trees (RRT) over the potential field approach and the Probabilistic Roadmap Method (PRM).

The potential field method depends heavily on the definition of a potential function which can be extremely difficult to derive. PRM samples random configurations and tries to connect pairs of nearby configurations by the use of a local planner. However, connecting nearby configurations can be also a challenging problem. RRT shares several of the advantages of PRM and other randomized planning techniques but does not connect nearby configurations, so the RRT approach seems suitable for a search in conformation space in the context of protein loop closure (Lee *et al.*, 2005).

2.6.1 Rapidly-exploring Random Tree (RRT)

The Rapidly-exploring Random Tree (RRT) is a randomized data structure for motion planning that is capable of handling a broad range of path planning problems including problems with high DOF (LaValle, 1998). The general applicability of a RRT for path planning algorithms stems from its properties: (a) the expansion is biased by a randomly selected configuration; (b) the distribution of RRT nodes converges to the sampling distribution; (c) a RRT reaches uniform coverage of the configuration space; (d) a RRT is probabilistically complete under very general conditions and (e) a RRT is always connected. A further benefit of a RRT is that it can be easily and efficiently implemented.

In each iteration, a random configuration is sampled towards which the RRT is going to be expanded. The RRT is searched for the closest configuration with respect to the random configuration. A local planner generates a new configuration on basis of the closest configuration found in the tree towards the random configuration, e.g. linear interpolation or a stepwise procedure like described in Subsection 2.5.3 can be used. The new configuration is added as a new node to the tree.

Voronoi bias

By sampling random configurations and pulling the RRT towards them, the exploration of the RRT is biased toward unexplored regions of the search space (Voronoi bias).

Algorithm 1 RRT

```
input       root node q_init
output      tree T
begin
      T ← InitTree(q_init);
      while not StopCondition() do
            q_rand ← RandomConf();
            q_near ← NearestNeighbor(q_rand, T);
            q_new ← Extend(T, q_rand, q_near);
            if q_new != NULL
                  AddNode(T, q_new);
end
```

RRT construction algorithm

In general, a RRT is initialized with an initial configuration, the root node of the tree. The exploration of the search space proceeds until predefined stop conditions are satisfied. Algorithm 1 shows in pseudo-code how the RRT is constructed (LaValle, 1998).

A RRT can be used to create an efficient path planning algorithm. A huge amount of path planners using a RRT have been described in literature, differing in detail. We use only a small subset of planners that seem to be most suitable for our needs. In the next sections I will shortly describe the differences of the planners we are considering for the conformational sampling in the context of protein loop closure.

RRTGoalBias & RRTCon algorithms

In principle, the construction algorithm of a RRT as described in Algorithm 1 behaves like a planner. However, without any bias to a goal it will converge extremely slowly. A planner that is biased by the goal with a given probability (`RRTGoalBias`) will converge much faster (LaValle and Kuffner, 2001). The pseudo-code of `RRTGoalBias` differs from Algorithm 1 in the implementation of `RandomConf()` which is not shown here explicitly.

RRTGoalBias

In Algorithm 1 the `Extend(`T`,` q_{rand}`,` q_{near}`)` procedure can be replaced by the `connect` algorithm shown in pseudo-code in Algorithm 2 which iterates `Extend(`T`,`

Algorithm 2 `connect` algorithm

```
input       tree T, q_rand, q_near
output      q_new
begin
      while not StopConnect() do
            q_new ← Extend(T, q_rand, q_near);

      return q_new;
end
```

q_{rand}, q_{near}) until the random configuration has been reached or the tree cannot be expanded any more at all. Consequently, a planner that uses Algorithm 2 is called `RRTCon` (LaValle and Kuffner, 2001). `RRTGoalBias` combined with `RRTCon` results in a greedy planner that tries to aggressively connect a configuration to the goal with a given probability.

RRTCon

Transition-based RRT algorithm (T-RRT)

The Transition-based RRT (T-RRT) combines standard RRT algorithms with a transition test similar to the Metropolis test (Metropolis *et al.*, 1953) for Monte Carlo methods used by stochastic optimization methods (Jaillet *et al.*, 2008, 2010). The T-RRT algorithm is shown in pseudo-code in Algorithm 3.

T-RRT shares the principle concepts with Algorithm 1 how new nodes are sampled for the expansion of the tree. However, T-RRT does not add every sampled configuration into the tree, but tests potential new configurations for acceptance prior to inserting them as new nodes. Thus, unfeasible configurations are rejected due to predefined criteria.

The acceptance test is based on the Metropolis criterion widely used in molecular modeling. The probability of acceptance or the transition probability p_{ij} is defined as:

$$p_{ij} = \begin{cases} \exp\left(-\frac{(c_j - c_i)/d_{ij}}{K \cdot T}\right) & (c_j - c_i)/d_{ij} > 0 \\ 1 & \text{otherwise} \end{cases}$$

Algorithm 3 T-RRT

```
input       root node q_init
output      tree T
begin
      T ← InitTree(q_init);
      while not StopCondition() do
            q_rand ← RandomConf();
            q_near ← NearestNeighbor(q_rand, T);
            q_new ← Extend(T, q_rand, q_near);
            if q_new ≠ NULL
            and TransitionTest(c(q_near), c(q_rand), d_near−new)
            and MinExpand(T, q_near, q_rand) then
                  AddNode(T, q_new);
end
```

Algorithm 4 `TransitionTest(`c_i`,` c_j`,` d_{ij}`)` function

```
begin
      if c_j < c_i then return true;
      p = exp( −(c_j − c_i)/d_ij / K·T );
      if rand(0, 1) < p then
            T = T/α;
            nFail = 0;
            return true;
      else
            if nFail > nFail_max then
                  T = T · α;
                  nFail = 0;
            else
                  nFail = nFail + 1;

            return false;
end
```

Algorithm 5 `MinExpand(`T`,` q_{near}`,` q_{rand}`)` function

```
begin
    if distance(q_near, q_rand) > δ then
        nExplorations = nExplorations + 1;
        return true;
    else
        if (nRefinements+1)/(nExplorations+1) > ρ then
            return false;
        else
            nRefinements = nRefinements + 1;
            nExplorations = nExplorations + 1;
            return true;
end
```

where c_i is the cost of the new and c_j the cost of the randomly sampled configuration. d_{ij} denotes to the distance between the configurations. K is a normalizing factor defined as $K = (c_i + c_j)/2$, T is the temperature similarly defined as for Monte Carlo methods.

The strength of the acceptance filter is automatically controlled, see Algorithm 4 for the pseudo-code of the `TransitionTest(`c_i`,` c_j`,` d_{ij}`)` function.

The `MinExpand(`T`,` q_{near}`,` q_{rand}`)` function controls the minimal rate of expansion toward unexplored regions of the search space on the basis of the ratio between exploration and refinement steps, see Algorithm 5 for pseudo-code.

exploration vs. refinement

Expansion steps of the RRT are classified into exploration and refinement steps by the distance between the closest configuration in the tree and the randomly sampled configuration. If this distance is greater than a given threshold value δ, the new configuration is considered to participate in an exploration and the new configuration is added as a new node to the tree in the assumption that the transition test has been passed. Otherwise, the new configuration is considered to participate in the tree refinement. No new nodes are added to the tree if the minimal expansion test fails. This is the case if the ratio between the total numbers of refinement and exploration steps exceeds a given maximum value ρ.

2.6.2 Transition-based Task Space RRT

We use a RRT-based motion planning approach to sample the conformation space for feasible loop structures. Our approach combines features from T-RRT and `RRTGoalBias` or respectively `RRTCon` with the iterated motion generation scheme introduced in Subsection 2.5.3 as a local planner. It is called Transition-based Task Space RRT because parts of the exploration procedure takes place in task space rather than in configuration space, as we will see soon. Task space planners have been already proposed in literature, but seem to be rarely used (Bertram *et al.*, 2006; Vande Weghe *et al.*, 2007; Shkolnik and Tedrake, 2009).

The RRT is initialized with a random protein loop conformation, e.g. the fully extended loop chain, after the bond lengths and angles have been idealized and the torsion angles ϕ, ψ have been set to 150° and −150°, respectively. The initial conformation is used as a template for torsion angle sampling. The goal conformation of the motion planner cannot be specified so that the goal is represented by each conformation that satisfies the loop closure constraints. The closure constraints are formulated as the deviation in position and orientation between the end-effector link and the rigid atom group formed by the N, C and C_α atoms of the fixed C-terminal anchor residue of the backbone as defined in Subsection 2.5.3. We call the latter rigid group of atoms the anchor link.

anchor link

The randomly sampled loop conformations towards which the RRT is being expanded are generated by applying a random sampling procedure on the torsional angles of the template conformation. An uniform and a Gaussian sampler have been implemented in our approach (Kuffner, 2004). A conformation is searched in the tree closest to the random conformation by calculating the minimum of the Euclidean norm of the angle displacements between the random conformation and each conformation already added to the tree. Both the random sampling of conformations and the distance calculation between two conformations are performed in conformation space.

uniform sampling

Then, the local planner on basis of the method derived in Subsection 2.5.3 is used to pull the selected conformation towards the random conformation. This task is expressed in terms of a virtual task force derived from a distance constraint between the end-effector link of the selected loop conformation and the corresponding link of the random conformation. After a sufficient number of steps the local planner stops and returns the last iterated conformation which is afterwards checked for acceptance by a transition test as defined for the T-RRT algorithm. Accepted conformations are added to the tree.

The acceptance test as discussed in Subsection 2.6.1 is based on a distance metric and a cost function, denoted by d_{ij} and c_i respectively c_j in Algorithm 4. We use a task space metric to estimate the distance between two loop conformations defined as the deviation in position and orientation of two protein links. The cost of a conformation is evaluated by an energy function defined by a specific set of energy terms. A "chainbreak" energy term is included that accounts for the energetic penalty due to the opening of the loop chain.

"chainbreak"

With a predefined probability, the anchor link of the protein backbone is used to derive the virtual task force pulling the closest conformation in the tree towards closure. In this case, the closest conformation corresponds to the conformation with lowest distance between the end-effector link and the anchor link. Thus, the conformation in the tree with lowest deviation from closure is selected. The local planner pulls the selected conformation then towards closure.

The protein loop is considered closed when the distance between the end-effector and the anchor link is lower than a predefined convergence threshold. The exploration of the conformation space continues until a maximum number of closed loop structures has been found or the maximum number of exploration steps has been reached.

2.7 RRT-guided Iterative Loop Closure algorithm

The motion planning algorithm described in Subsection 2.6.2 can be used as a component for protein loop structure prediction. The key component of the presented methodology is the representation of a protein loop as a kinematic chain of rigid atom groups or protein links. So each protein structure prediction application that implements the forward kinematics relationships for the backbone representation of a protein could be easily enhanced with the algorithm derived in the previous sections. Rosetta uses an atom-tree representation for the protein structure implementing the forward kinematics equations (Parsons *et al.*, 2005). In the next chapter the implementation of the new "RRT-guided Iterative Loop Closure" algorithm in Rosetta is briefly discussed.

It is also possible to use the algorithm within a Monte Carlo procedure where the RRT is reinitialized with a different template conformation after a predefined number of Monte Carlo steps. This changes the bias of the random sampling procedure because of the shifted probability distribution by which the random conformations are estimated. Predicted loop structures will finally be evaluated by a

scoring function and a Metropolis test. The best-scored loop structure may then be further refined by repacking the amino acid side chains of the loop or of the whole protein backbone.

3 Application to protein structure prediction

The implementation of the RRT-guided Iterative Loop Closure algorithm will be done using the Rosetta protein modeling suite (Schueler-Furman *et al.*, 2005). So the algorithm can be directly compared with the already existing loop closure methods, the Cyclic Coordinate Descent (CCD) and the Kinematic Closure (KIC) algorithms (Canutescu and Dunbrack, 2003; Mandell *et al.*, 2009a). The implementation is also based on a linear algebra library called Eigen which is used for complex matrix computations and linear algebra operations (Guennebaud *et al.*, 2010).

Eigen

3.1 Rosetta protein-modeling suite

Rosetta is a software package for protein structure prediction, protein docking and protein design that performed excellently in the last CASP experiments. Many conformational sampling and optimization methods have been implemented. For evaluation and scoring Rosetta is based on a highly modular energy function that is primarily classified on the basis of how the protein side chains are modeled, in centroid or in all atom representation. The energy function itself is composed of both physical and statistically derived energy terms. Rosetta unions a huge amount of protocols where each is subject of a specific problem or task and each defines its own specific energy function composed of energy terms that are predominant for the modeled problem.

Various protocols have been implemented for specific problems in protein structure prediction, including protocols that address the protein loop closure problem (Kaufmann *et al.*, 2010). The loop closure protocols of Rosetta are associated with the Rosetta `LoopModel` application specifically designed for protein loop closure.

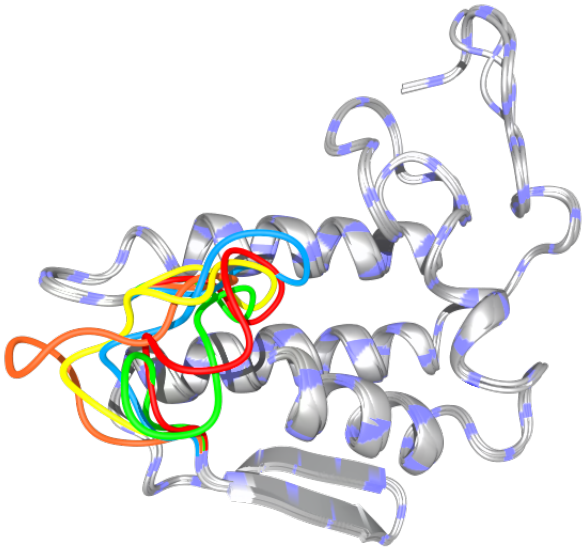

Fig. 3.1: Various protein loop structures within the backbone environment. The ensemble of loop structures has been determined by the Rosetta CCD protocol. The colored segments refer to the sampled loop structures. A huge amount of such loops is usually generated to select the best-scored.

Figure 3.1 shows closed protein loop structures generated by Rosetta CCD after superimposing the remaining protein backbones.

3.2 Implementation details

The program code of all parts of the algorithm has been implemented within the framework of Rosetta using existing methods and data structures whenever possible. Consequently, the `LoopModel` application was chosen as an interface to low-level functionality of the framework like I/O operations. A lot of care has been taken to separate independent parts for general applicability.

KinematicMap The key data structure specifically developed for the loop closure algorithm presented is called `KinematicMap`. It relates the DOF of a kinematic chain

representation to the internal coordinates used in the atom-tree representation of Rosetta. Thus, the two representations could be kept synchronized in an efficient way. It is also responsible for computing the Jacobian and for adding angle increments to the DOF.

Pseudoinverse (Moore-Penrose inverse) The Singular Value Decomposition (SVD) for the Jacobian is operated by Eigen (Guennebaud *et al.*, 2010). By using the SVD matrices the pseudoinverse can be calculated with the algorithm described in (Press, 2007).

Energy minimization by Steepest Descent and Line Search The energy minimization task by projecting gradients onto the nullspace is implemented as a steepest descent search combined with a simple line search algorithm. The line search algorithm widely used in optimization and molecular modeling calculates an optimal step size to minimize the energy score as far as possible (Leach, 2001).

RRT The RRT used in the presented algorithm has been written from scratch. However, the object-oriented design was inspired by the Motion Strategy Library (MSL) (LaValle and Others, 2003). Furthermore, the RRT has been implemented using Rosetta data structures and methods whenever appropriate. Though the RRT is relatively easy to understand, it is not a trivial task to implement them (Sucan and Kavraki, 2010).

3.3 Validation

For the validation of the predicted loop structures different loop model sets can be used, e.g. from Fiser *et al.* (2000). Many of them were composed for previous publications and several are freely available on the web or by request. Thus, available loop model sets can be used to compare the accuracy of the different computational loop closure approaches in predicting loop structures.

It is obvious to compare the "RRT-guided Iterative Loop Closure" algorithm to both CCD and KIC as all of them have been implemented in Rosetta. So, as long as all three methods make use of the same scoring function of Rosetta the accuracy of the predicted loop structures could be directly compared by the rmsd value between known crystallographic structures and the predicted ones.

3.4 Proof of concept

A thorough analysis of the efficiency and applicability of the presented method requires a lot of computer power and a huge amount of simulation trials. Furthermore, the experiments have to be evaluated and parameters eventually have to be adjusted. Weeks and lots of computer power are necessary to perform comparable experimental results. Due to the complexity of implementing the presented algorithm in an existing framework, no experimental results are available at the time of this writing. Therefore, an objective evaluation by comparing different closure methods is not possible.

However, the different components of the presented algorithm have been successfully tested with an example protein structure from the pdb. The algorithm was able to sample the conformation space for closed loop structures up to a given threshold. The correctness of the local planner has also been successfully evaluated.

So there is still no objection to assume that our algorithm has the potential to reach the accuracy of most of the published methods. We hope that we can perform a sufficient number of experiments with reasonable results in order to publish this algorithmic approach at a later point in time.

4 Conclusion

A lot of literature is available in the field of protein loop closure, but there is no recent review of existing loop closure methods. So the original publications of the most promising closure methods were elaborated beginning with the most recent. Thus, the overview presented in this thesis is just a small subset, but covers at least the most accurate methods.

4.1 Obstacles

When implementing, the main obstacle I had to cope with was the complexity of the Rosetta code base in a whole.[1] The existing loop closure protocols were hard to read and even harder to understand which made it really difficult to find a way how to start. Fortunately, Rosetta is designed in a sophisticated way so it is sufficient to read the most basic code parts at first and to concentrate then on the higher-level protocols. Irrelevant code must be rarely read due to the object-oriented design. However, the general obstacles for a developer who is not familiar with programming in Rosetta are extremely high.

4.2 Improvements & future work

Expansive Space Tree (EST)

The motion planning algorithm presented could be modified in many ways. For example, an Expansive Space Tree (EST) could be used instead of the RRT. It differs from RRT mainly in the way how it is expanded. A task space planner based on an EST has been recently reviewed in (Sucan and Kavraki, 2010).

The task space metrics used for calculating the distance of two kinematic chains of a protein loop can be defined in various ways, e.g. by using quaternions or by

[1] a lines-of-code (LOC) calculator indicated 1.2 million lines of C/C++ code

kd-tree search

the weighted sum of position and orientation displacements (Kuffner, 2004). The implementation of the nearest neighbor search during the expansion step of a RRT can be substantially improved by using `kd-tree` based search techniques (Atramentov and LaValle, 2002; Yershova and LaValle, 2007). Alternatives could also be considered for the Euclidean norm metric used to search the tree for the nearest neighbors or for the random sampling procedure (Kuffner, 2004). Defining a flexible convergence procedure for a motion that approaches a goal might be helpful to determine the stop condition of the closure procedure. Various parameters have to be tested to find their valuable range or replaced by others with a more suitable definition.

4.3 Final statement

Unfortunately, we do not have any experimental results yet, so any conclusion on a basis of how good the method works and how good the predicted structures are is impossible. Nevertheless, I am truly convinced that the presented methodology has lots of potential, though this has to be verified yet.

References

H. Abe, W. Braun, T. Noguti and N. Go. *Rapid calculation of first and second derivatives of conformational energy with respect to dihedral angles for proteins general recurrent equations*. *Computers & Chemistry*, 8(4):239–247, 1984.

http://linkinghub.elsevier.com/retrieve/pii/0097848584850159

A. Atramentov and S. LaValle. *Efficient nearest neighbor searching for motion planning*. In *2002 IEEE International Conference on Robotics and Automation*, volume pp, pages 632–637 (IEEE), 2002.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1013429

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov and P. E. Bourne. *The Protein Data Bank*. *Nucleic acids research*, 28(1):235–42, 2000.

http://www.ncbi.nlm.nih.gov/pubmed/10592235

D. Bertram, J. Kuffner, R. Dillmann and T. Asfour. *An integrated approach to inverse kinematics and path planning for redundant manipulators*. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, May, pages 1874–1879 (IEEE), 2006.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1641979

C. Brändén and J. Tooze. *Introduction to protein structure* (Garland Pub.), 2nd edition, 1999.

http://books.google.de/books?id=0Sx-MhISYKQC

J. Burdick. *On the inverse kinematics of redundant manipulators: characterization of the self-motion manifolds*. In *Proceedings, 1989 International Conference on Robotics and Automation*, pages 264–270 (IEEE Comput. Soc. Press), 1989.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=99999

C. Cantor and P. Schimmel. *Biophysical chemistry: The conformation of biological macromolecules* (W. H. Freeman), 1st edition, 1980.

http://books.google.de/books?id=Jb0fQH1Eto0C

A. A. Canutescu and R. L. Dunbrack. *Cyclic coordinate descent: A robotics algorithm for protein loop closure. Protein science : A publication of the Protein Society*, 12(5):963–72, 2003.

http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2323867&tool=pmcentrez&rendertype=abstract

V. Choi. *On Updating Torsion Angles of Molecular Conformations. Journal of Chemical Information and Modeling*, 46(1):438–444, 2006.

http://pubs.acs.org/doi/abs/10.1021/ci050253h

J. Cortés and T. Siméon. *Probabilistic motion planning for parallel mechanisms*. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, pages 4354–4359 (IEEE), 2003.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1242274

J. Cortés and T. Siméon. *Sampling-Based Motion Planning under Kinematic Loop-Closure Constraints*, volume 17 of *Springer Tracts in Advanced Robotics*, pages 75–90 (Springer-Verlag, Berlin/Heidelberg), 2005.

http://dx.doi.org/10.1007/10991541_7

J. Cortés, T. Siméon and J. Laumond. *A random loop generator for planning the motions of closed kinematic chains using PRM methods*. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, May, pages 2141–2146 (IEEE), 2002.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1014856

E. A. Coutsias, C. Seok, M. P. Jacobson and K. A. Dill. *A kinematic view of loop closure. Journal of computational chemistry*, 25(4):510–28, 2004.

http://www.ncbi.nlm.nih.gov/pubmed/14735570

E. A. Coutsias, C. Seok, M. J. Wester and K. A. Dill. *Resultants and loop closure. International Journal of Quantum Chemistry*, 106(1):176–189, 2006.

http://doi.wiley.com/10.1002/qua.20751

J. Craig. *Introduction to robotics: mechanics and control* (Pearson/Prentice Hall), 3rd edition, 2005.

http://books.google.de/books?id=MqMeAQAAIAAJ

R. A. Engh and R. Huber. *Accurate bond and angle parameters for X-ray protein structure refinement*. *Acta Crystallographica Section A Foundations of Crystallography*, 47(4):392–400, 1991.

http://scripts.iucr.org/cgi-bin/paper?S0108767391001071

A. K. Felts, E. Gallicchio, D. Chekmarev, K. A. Paris, R. A. Friesner and R. M. Levy. *Prediction of Protein Loop Conformations using the AGBNP Implicit Solvent Model and Torsion Angle Sampling*. *Journal of chemical theory and computation*, 4(5):855–868, 2008.

http://www.ncbi.nlm.nih.gov/pubmed/18787648

A. Fiser, R. K. Do and A. Sali. *Modeling of loops in protein structures*. *Protein Science: A Publication of the Protein Society*, 9(09):1753–73, 2000.

http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2144714&tool=pmcentrez&rendertype=abstract

N. Go and H. A. Scheraga. *Ring Closure and Local Conformational Deformations of Chain Molecules*. *Macromolecules*, 3(2):178–187, 1970.

http://pubs.acs.org/doi/abs/10.1021/ma60014a012

G. Guennebaud, B. Jacob and Others. *Eigen*, 2010.

http://eigen.tuxfamily.org

M. P. Jacobson, D. L. Pincus, C. S. Rapp, T. J. F. Day, B. Honig, D. E. Shaw and R. A. Friesner. *A hierarchical approach to all-atom protein loop prediction*. *Proteins*, 55(2):351–67, 2004.

http://www.ncbi.nlm.nih.gov/pubmed/15048827

F. Jagodzinski and O. Brock. *Towards a mechanistic view of protein motion*. *2007 46th IEEE Conference on Decision and Control*, pages 4557–4562, 2007.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4434801

L. Jaillet, J. Cortés and T. Siméon. *Transition-based RRT for path planning in continuous cost spaces*. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2145–2150, 2008.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4650993

L. Jaillet, J. Cortés and T. Siméon. *Sampling-Based Path Planning on Configuration-Space Costmaps*. *IEEE Transactions on Robotics*, 2010.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5477164

K. W. Kaufmann, G. H. Lemmon, S. L. Deluca, J. H. Sheehan and J. Meiler. *Practically Useful: What the Rosetta Protein Modeling Suite Can Do for You. Biochemistry*, (7), 2010.

http://www.ncbi.nlm.nih.gov/pubmed/20235548

K. Kazerounian, K. Latif and C. Alvarado. *Protofold: A Successive Kinetostatic Compliance Method for Protein Conformation Prediction*. *Journal of Mechanical Design*, 127(4):712, 2005.

http://link.aip.org/link/JMDEDB/v127/i4/p712/s1&Agg=doi

O. Khatib. *A unified approach for motion and force control of robot manipulators: The operational space formulation*. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1087068

O. Khatib. *Motion/force redundancy of manipulators*. In *Proceedings of Japan-USA Symposium on Flexible Automation*, volume 1, pages 337–342 (Kyoto, Japan), 1990.

http://www.cc.gatech.edu/~{}mstilman/class/RIP09/materials/KhatibMF90.pdf

D. E. Kim, B. Blum, P. Bradley and D. Baker. *Sampling bottlenecks in de novo protein structure prediction. Journal of molecular biology*, 393(1):249–60, 2009.

http://www.ncbi.nlm.nih.gov/pubmed/19646450

S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi. *Optimization by Simulated Annealing. Science (New York, N.Y.)*, 220(4598):671–680, 1983.

http://www.ncbi.nlm.nih.gov/pubmed/17813860

R. Kolodny. *Inverse Kinematics in Biology: The Protein Loop Closure Problem*. *The International Journal of Robotics Research*, 24(2-3):151–163, 2005.

http://ijr.sagepub.com/cgi/doi/10.1177/0278364905050352

J. Kuffner. *Effective sampling and distance metrics for 3D rigid body path planning*. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, Icra, pages 3993–3998 (IEEE), 2004.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1308895

J. Latombe, L. Kavraki, P. Svestka and M. Overmars. *Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

http://en.scientificcommons.org/42690062

S. LaValle. *Rapidly-exploring random trees: A new tool for path planning. Technical Report, Computer Science Department, Iowa State University*, TR 98-11, 1998.

http://msl.cs.uiuc.edu/~{}lavalle/papers/Lav98c.pdf

S. LaValle. *Planning algorithms* (Cambridge University Press), 1st edition, 2006.

http://books.google.de/books?id=Clg8SWNMSRAC

S. LaValle and J. Kuffner. *Rapidly-exploring random trees: Progress and prospects*, pages 293 – 308 (AK Peters, Ltd.), 2001.

http://msl.cs.uiuc.edu/~{}lavalle/papers/LavKuf01.pdf

S. M. LaValle and Others. *Motion Strategy Library (MSL)*, 2003.

http://msl.cs.uiuc.edu/msl/

A. R. Leach. *Molecular Modelling: Principles and Applications* (Prentice Hall, New York), 2nd edition, 2001.

http://books.google.de/books?id=kB7jsbV-uhkC

A. Lee, I. Streinu and O. Brock. *A methodology for efficiently sampling the conformation space of molecular structures. Physical biology*, 2(4):S108–15, 2005.

http://www.ncbi.nlm.nih.gov/pubmed/16280616

J. Lee, D. Lee, H. Park, E. A. Coutsias and C. Seok. *Protein loop modeling by using fragment assembly and analytical loop closure. Proteins: Structure, Function, and Bioinformatics*, pages n/a–n/a, 2010.

http://doi.wiley.com/10.1002/prot.22849

M. Levitt. *A simplified representation of protein conformations for rapid simulation of protein folding. Journal of molecular biology*, 104(1):59–107, 1976.

http://www.ncbi.nlm.nih.gov/pubmed/957439

I. Lotan. *Algorithms exploiting the chain structure of proteins.* Dissertation, Stanford University, 2004.

http://www-cs-students.stanford.edu/~{}itayl/mythesis.pdf

D. J. Mandell, E. A. Coutsias and T. Kortemme. *Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. Nature methods*, 6(8):551–2, 2009a.

http://www.ncbi.nlm.nih.gov/pubmed/19644455

D. J. Mandell, E. A. Coutsias and T. Kortemme. *Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling - Supplementary Text and Figures. Nature methods*, 6(8):551–2, 2009b.

http://www.ncbi.nlm.nih.gov/pubmed/19644455

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller. *Equation of State Calculations by Fast Computing Machines. The Journal of Chemical Physics*, 21(6):1087, 1953.

http://link.aip.org/link/JCPSA6/v21/i6/p1087/s1&Agg=doi

M. Mönnigmann and C. A. Floudas. *Protein loop structure prediction with flexible stem geometries. Proteins*, 61(4):748–62, 2005.

http://www.ncbi.nlm.nih.gov/pubmed/16222670

D. E. Orin and W. W. Schrader. *Efficient Computation of the Jacobian for Robot Manipulators. The International Journal of Robotics Research*, 3(4):66–75, 1984.

http://ijr.sagepub.com/cgi/doi/10.1177/027836498400300404

J. Parsons, J. B. Holmes, J. M. Rojas, J. Tsai and C. E. M. Strauss. *Practical conversion from torsion space to Cartesian space for in silico protein synthesis. Journal of computational chemistry*, 26(10):1063–8, 2005.

http://www.ncbi.nlm.nih.gov/pubmed/15898109

W. Press. *Numerical recipes: the art of scientific computing* (Cambridge University Press), 3rd edition, 2007.

http://books.google.de/books?id=w47cyQLVQowC

C. A. Rohl, C. E. M. Strauss, D. Chivian and D. Baker. *Modeling structurally variable regions in homologous proteins with rosetta. Proteins*, 55(3):656–77, 2004.

http://www.ncbi.nlm.nih.gov/pubmed/15103629

O. Schueler-Furman, C. Wang, P. Bradley, K. Misura and D. Baker. *Progress in modeling of protein structures and interactions. Science (New York, N.Y.)*, 310(5748):638–42, 2005.

http://www.ncbi.nlm.nih.gov/pubmed/16254179

B. D. Sellers, K. Zhu, S. Zhao, R. a. Friesner and M. P. Jacobson. *Toward better refinement of comparative models: predicting loops in inexact environments. Proteins*, 72(3):959–71, 2008.

http://www.ncbi.nlm.nih.gov/pubmed/18300241

P. S. Shenkin, D. L. Yarmush, R. M. Fine, H. J. Wang and C. Levinthal. *Predicting antibody hypervariable loop conformation. I. Ensembles of random conformations for ringlike structures. Biopolymers*, 26(12):2053–85, 1987.

http://www.ncbi.nlm.nih.gov/pubmed/3435744

A. Shkolnik and R. Tedrake. *Path planning in 1000+ dimensions using a task-space Voronoi bias*. In *2009 IEEE International Conference on Robotics and Automation*, pages 2061–2067 (Ieee), 2009.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5152638

C. S. Soto, M. Fasnacht, J. Zhu, L. Forrest and B. Honig. *Loop modeling: Sampling, filtering, and scoring. Proteins*, 70(3):834–43, 2008.

http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2553011&tool=pmcentrez&rendertype=abstract

M. Spong, S. Hutchinson and M. Vidyasagar. *Robot modeling and control* (John Wiley & Sons, Ltd), 1st edition, 2006.

http://books.google.de/books?id=wGapQAAACAAJ

I. A. Sucan and L. E. Kavraki. *On the implementation of single-query sampling-based motion planners*. In *2010 IEEE International Conference on Robotics and Automation*, pages 2005–2011 (IEEE), 2010.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5509172

P. A. Tipler and G. Mosca. *Physics for scientists and engineers* (W. H. Freeman), 6th edition, 2007.

http://books.google.de/books?id=_69JKgAACAAJ

H. W. van Vlijmen and M. Karplus. *PDB-based protein loop prediction: parameters for selection and methods for optimization. Journal of molecular biology*, 267(4):975–1001, 1997.

http://www.ncbi.nlm.nih.gov/pubmed/9135125

M. Vande Weghe, D. Ferguson and S. S. Srinivasa. *Randomized path planning for redundant manipulators without inverse kinematics*. In *2007 7th IEEE-RAS International Conference on Humanoid Robots*, pages 477–482 (IEEE), 2007.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4813913

W. J. Wedemeyer and D. Baker. *Efficient minimization of angle-dependent potentials for polypeptides in internal coordinates. Proteins*, 53(2):262–72, 2003.

http://www.ncbi.nlm.nih.gov/pubmed/14517977

W. J. Wedemeyer and H. A. Scheraga. *Exact analytical loop closure in proteins using polynomial equations. Journal of Computational Chemistry*, 20(8):819–844, 1999.

http://doi.wiley.com/10.1002/%28SICI%291096-987X%28199906%2920%3A8%3C819%3A%3AAID-JCC8%3E3.0.CO%3B2-Y

Z. Xiang, C. S. Soto and B. Honig. *Evaluating conformational free energies: the colony energy and its application to the problem of loop prediction. Proceedings of the National Academy of Sciences of the United States of America*, 99(11):7432–7, 2002.

http://www.ncbi.nlm.nih.gov/pubmed/12032300

A. Yershova and S. M. LaValle. *Improving Motion-Planning Algorithms by Efficient Nearest-Neighbor Searching. IEEE Transactions on Robotics*, 23(1):151–157, 2007.

http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4084578

M. Zhang and L. Kavraki. *A New Method for Fast and Accurate Derivation of Molecular Conformations. Journal of Chemical Information and Modeling*, 42(1):64–70, 2002.

http://pubs.acs.org/cgi-bin/doilookup/?10.1021/ci010327z

K. Zhu, D. L. Pincus, S. Zhao and R. A. Friesner. *Long loop prediction using the protein local optimization program. Proteins*, 65(2):438–52, 2006.

http://www.ncbi.nlm.nih.gov/pubmed/16927380

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Berlin, den 15. Dezember 2010