12-1-2005

A Methodology for Efficiently Sampling the Conformation Space of Molecular Structures

Audrey Lee
University of Massachusetts Amherst

Ileana Streinu
Smith College, istreinu@smith.edu

Oliver Brock
University of Massachusetts Amherst

A methodology for efficiently sampling the conformation space of molecular structures

2005 Phys. Biol. 2 S108

(http://iopscience.iop.org/1478-3975/2/4/S05)

INSTITUTE OF PHYSICS PUBLISHING PHYSICAL BIOLOGY

Phys. Biol. **2** (2005) S108–S115 doi:10.1088/1478-3975/2/4/S05

A methodology for efficiently sampling the conformation space of molecular structures

Audrey Lee[1], Ileana Streinu[2] and Oliver Brock[1]

[1] Department of Computer Science, University of Massachusetts Amherst, Amherst, MA, USA
[2] Department of Computer Science, Smith College, Northampton, MA, USA

E-mail: streinu@cs.smith.edu

Received 1 June 2005
Accepted for publication 29 July 2005
Published 9 November 2005
Online at stacks.iop.org/PhysBio/2/S108

Abstract

Motivated by recently developed computational techniques for studying protein flexibility, and their potential applications in docking, we propose an efficient method for sampling the conformational space of complex molecular structures. We focus on the *loop closure* problem, identified in the work of Thorpe and Lei (2004 *Phil. Mag.* **84** 1323–31) as a primary bottleneck in the fast simulation of molecular motions. By modeling a molecular structure as a *branching robot*, we use an intuitive method in which the robot holds onto itself for maintaining loop constraints. New conformations are generated by applying random *external forces*, while *internal, attractive forces* pull the loops closed. Our implementation, tested on several model molecules with low number of degrees of freedom but many interconnected loops, gives promising results that show an almost four times speed-up on the benchmark cube-molecule of Thorpe and Lei.

1. Introduction

Macromolecules undergo conformational changes to perform their biological function. Experimental techniques, such as x-ray crystallography and NMR, are used to extract information on individual conformations or ensembles, but lack the capability of capturing the *entire* conformation space. In many biological processes, for example protein docking, a protein molecule transitions from one low-energy conformation to another in which it is able to bind the ligand. To determine the function of a macromolecule, it is therefore necessary to build a representation of feasible conformational changes.

Many biologically relevant macromolecules contain flexible loops. A valid conformation for such a molecule must satisfy the loop closure constraints [15] for every one of its loops. The set of all conformations that satisfy these constraints represents a vanishingly small portion of the overall conformation space. As a result, most conformations generated by the perturbation of a set of bond angles will violate the loop constraints. This makes the exploration of the valid conformation space for a molecule with flexible loops computationally very difficult.

Approaches to sample the conformation space of molecules with loops employ a two-step approach. Initially, a new configuration is obtained by rotating some of the bonds of the molecule. The resulting conformation violates the loop closure constraints. In a second step, these constraints are re-attained using iterative procedures.

We propose a new, faster method for sampling the entire conformation space of molecular structures containing loops. The key contribution of our paper consists of our approach of re-attaining loop closure constraints for a randomly perturbed conformation. This method is based on standard techniques from robotics. We model the molecule as a *branching robot* that holds onto itself to maintain loop constraints. New conformations can efficiently be generated by applying random *external forces*, while *internal, attractive forces* pull the loops closed.

Our work builds upon the general methodology of Thorpe and Lei [32], which is incorporated into the ROCK software system [22, 26]. They decompose a protein structure into

regions with varying *degrees of flexibility*. Then they generate random motions of the atoms, while attempting to reinstate the constraints imposed by the bond lengths and angles of the structure. Substantial computational effort goes into satisfying the *loop closure* and *steric* constraints.

The proposed approach efficiently generates a representation of the valid conformation space of molecules with flexible loops. This valid conformation space is significantly smaller than the conformation space of all possible bond angles. Effectively, the proposed approaches exploit the kinematic constraints imposed by loops to determine a significantly reduced conformation space. Using this reduced space, it becomes computationally much more tractable to perform computations based on energetic considerations. In this paper, we focus on the computation of this smaller search space. Energetic considerations of search in this reduced space are not considered, but can be accomplished with a variety of methods found in the literature (see section 2).

The loop closure problem is also relevant in other areas of molecular biology. In homology modeling, for example, the structure of a protein is predicted by assembling fragments from a number of homologs. The matching of the termini of these fragments can be viewed as an instance of the loop closure problem. Canutescu and Dunbrack [7], Lotan *et al* [23] and Kolodny *et al* [18] have successfully applied robotics techniques to this domain.

2. Related work

Researchers have investigated a number of different approaches to sample the conformation space of molecules. Molecular dynamics [33] simulations numerically solve Newton's equations of motion for a detailed model (such as an atomic model) of a molecule. This procedure generates a trajectory through the conformation space of the molecule. Since the trajectory is computed in a discretized fashion, we can say that this method samples the conformation space of the molecule. Molecular dynamics simulations require large amounts of computation. Consequently, they can only compute a small fraction of the total accessible conformation space of a large molecule in a reasonable amount of time.

In molecular structures, atoms do not move independently but instead are constrained by covalent bonds and other interactions with atoms in the molecule. To ensure that a molecular dynamics simulation does not violate these constraints, researchers have proposed a number of augmentations to the basic molecular dynamics algorithm. The Shake method [27] implements an iterative procedure to satisfy spatial constraints between atoms. The Rattle algorithm [2] extends Shake by also considering velocity constraints. Settle [25] further improves on Rattle by providing more efficient computations. These approaches consider the motion of every atom independently. Due to this independence, the motion of an atom will violate the constraints imposed by the molecular structure. The Shake, Rattle and Settle methods employ an iterative approach to re-attain violated constraints. Because the various constraints are dependent, this is a computationally costly procedure. In contrast, the approach proposed in this paper exploits techniques from robotics to maintain all but one constraint per loop during the motion. Consequently, only a single constraint has to be re-attained per loop, irrespective of the number of atoms present in that loop. This significantly reduces the computational cost compared to methods that move each atom of a molecule independently. The computational cost of the proposed algorithm depends on the number of loops per molecule, rather than the number of atoms, making it efficient even for very large molecules.

Monte Carlo simulations are a different method of sampling the conformation space of a molecule. They perform a random walk in conformation space, biased by the Metropolis criterion [24]. This random walk is not constrained by Newton's law of motion, as was the case in molecular dynamics simulations, and hence Monte Carlo simulations are computationally more efficient than molecular dynamics simulations. One of the drawbacks of Monte Carlo simulations is that they spend considerable computational resources exploring local minima. Researchers have devised a large number of extensions to the Monte Carlo algorithm to address this problem. These extensions include: the replica Monte Carlo method [30], the multicanonical ensemble method [3], entropic sampling [21], methods based on weighted histograms [19], parallel tempering [17], jump walking [14], multicanonical jump walking [34], smart walking [36] and local energy flattening [35]. These methods attempt to render the exploration of conformation space more efficient based on energetic considerations. However, in the context of molecules with flexible loops, the most severe constraints on configuration space are imposed by loop closure constraints and not by energetics. An energetically favorable conformation will generally have very few other favorable conformations in its neighborhood, since adjacent conformations may violate the loop closure constraints. Without an explicit treatment of loop closure constraints, these extensions to the Monte Carlo method are not well suited for molecules with flexible loops.

The proposed approach can be viewed as complementary to Monte Carlo simulations in the context of molecules with flexible loops. The method described in this paper determines the part of the conformation space of a molecule with loops that satisfies the loop closure constraints. This subset of the conformation space is computed without considering energetics. Monte Carlo simulations that are restricted to this valid subset of the overall conformation space can then be used to consider energetic considerations much more efficiently, as they will only sample conformations that satisfy loop closure constraints. Effectively, the method described in this paper exploits the kinematics constraints of molecules with loops to reduce the search space available to Monte Carlo simulations, thus significantly increasing their computational efficiency.

During a Monte Carlo simulation, the random change of a single bond angle may result in a large displacement of large portions of the molecule. This is not necessarily biologically plausible, and therefore researchers have devised special Monte Carlo moves that only affect a local area of the molecule. A move consists of selecting a short linear

segment of the molecule and changing its conformation without affecting other parts of the molecule. Note that the short linear segment can be viewed as a flexible loop. Methods to perform such local moves include random walks in a lattice [29], numerical approaches to determine valid conformations for the short linear segment [13], 'wriggling' the chain and re-attaining constraints using linear algebra [6], and the local application of Shake [1]. All of these methods suffer from the same problems as Shake that were discussed above. Furthermore, they have only been applied to short segments, rather than loops of arbitrary length.

Another method of sampling the conformation space of molecules performs exhaustive search [4]. The search space for such an exhaustive search grows exponentially with the number of rotatable bonds. Consequently, these computations become intractable, even for relatively short molecules. Furthermore, this approach is not well suited for molecules with loops: depending on the discretization of the space, loop closure constraints cannot necessarily be satisfied by configurations chosen from a lattice in conformation space.

Thorpe and Lei [32] were the first to be explicitly concerned with molecules containing many interconnected, flexible loops. Given a conformation, Thorpe and Lei's approach attempts to sample the space around it by randomly perturbing the angles at a chosen set of bonds. The difficulty of the loop closure problem lies in finding a solution to a complicated algebraic system of equations, accomplished by minimizing a fictitious *loop closure potential*. However, because the perturbed bond angles are often interdependent, a solution may not exist, resulting in a high rejection rate.

To the best of our knowledge, only one other approach has been proposed for generating molecular conformations, with loops taken under consideration. In the recent work of [10], the general technique for loop closure of [11], called *random loop generator* (RLG), is applied to proteins. RLG has been developed in robotics and follows from the work of LaValle *et al* [20] and Han and Amato [16] in the domain of path planning. The algorithm of [20] breaks the loops, then uses iterative distance minimization to restore the loop constraints. Both [16] and [11] partition the loops into *active* and *passive* chains; forward kinematics is applied to the active chains, while inverse kinematics is applied to maintain the loop closure. More specifically, sampling is accomplished by selecting random values for the active chains, then finding values for the passive chains that close the loop; however, just as in the case of Thorpe and Lei's approach, it is not always possible to find a valid configuration for the passive chains, resulting in a rejection.

3. Methodology

We now describe our method for efficiently sampling the conformation space of a molecular structure *when steric clashes are ignored*. Indeed, as we already pointed out, a *major* challenge in generating new conformations is loop closure.

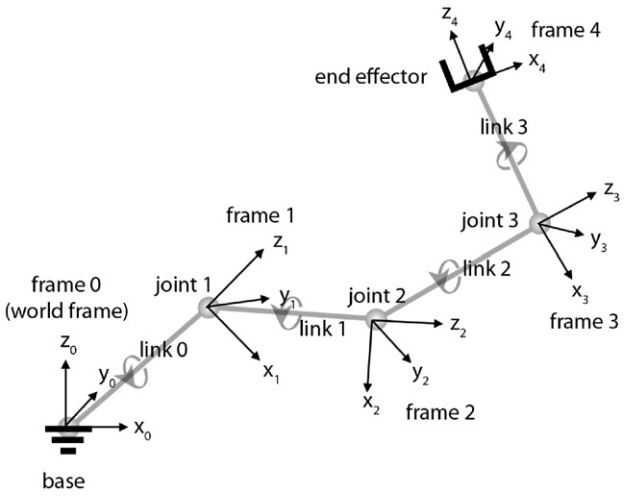

Figure 1. Example robot arm and attached frames.

3.1. Initial framework

To begin with, we describe the framework used to model the molecular structure: a *branching robot*, whose key building block is a *simple robot arm*.

3.1.1. Simple robot arm. The kinematics of robot arms is well understood and covered in most graduate-level robotics texts, such as [12]. We present here a brief overview, adapted to our specific context.

Frame assignment. A *robot arm* is a single chain consisting of *joints* connected by *links*. It is important to note that the first and last elements of the chain are special; they actually are not considered joints and are called the *base* and *end effectors*, respectively, of the robot arm. See figure 1 for an example. We consider only *revolute* joints, i.e. joints that allow rotational motion about a fixed axis; we call θ_i the variable representing this single degree of freedom. Joints are labeled $1, \ldots, n$; link i connects joint i to joint $i+1$. In our model, revolute joints are associated with atoms and constant-length links with bonds.

We attach rigidly to each link a *coordinate frame*; although any frame may be chosen, there is a standard convention in robotics for frame assignment. Refer to figure 1 for frame assignments on the example robot arm. We denote by frame i the frame attached to link i; by convention, its origin $\mathbf{p}_i$ lies on joint i and its $\mathbf{z}_i$ aligns with joint i's axis of rotation. The x-axis is on the common normal of $\mathbf{z}_i$ and $\mathbf{z}_{i+1}$, and the y-axis is chosen to obey a right-hand coordinate system. A special *world frame*, denoted frame 0, is chosen as the frame in which to express all others; for simplicity, its origin is chosen as the base's position. Once frames have been chosen, a 4×4 *homogeneous transformation matrix* T_i from the world frame to frame i is determined.

Force propagation. The *Jacobian matrix* J relates the rate of change of the position and orientation of the end effector to those of the joints. We now show how to compute it. The $6 \times n$ Jacobian consists of a column for each degree of freedom (or joint), with three rows corresponding to position and three

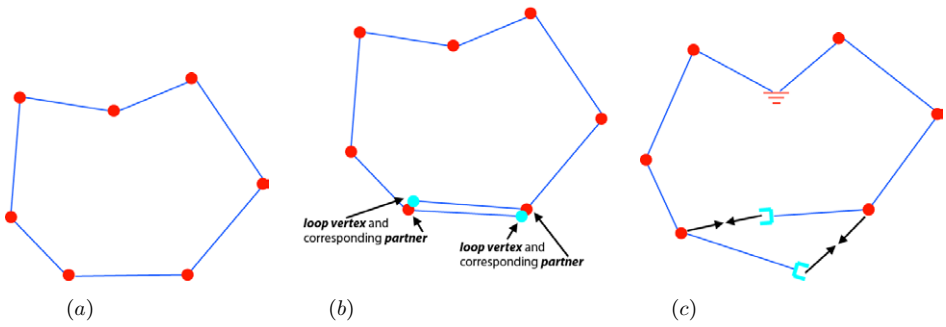

Figure 2. Initialization and loop closure on simple 2D example. (*a*) Initial structure. (*b*) Tree with new (lighter) loop vertices and corresponding partners. (*c*) Attractive loop closure forces on branching robot.

rows corresponding to orientation, and is expressed as follows:

$$\begin{bmatrix} \mathbf{z}_1 \times \mathbf{p}_1 & \mathbf{z}_2 \times \mathbf{p}_2 & \cdots & \mathbf{z}_i \times \mathbf{p}_i & \cdots & \mathbf{z}_n \times \mathbf{p}_n \\ \mathbf{z}_1 & \mathbf{z}_2 & \cdots & \mathbf{z}_i & \cdots & \mathbf{z}_n \end{bmatrix}.$$

For joint i, we use the axis of rotation $\mathbf{z}_i$ and origin $\mathbf{p}_i$ of the frame attached to link i, found in the third and fourth columns of T_i, respectively.

The instantaneous effect of a force F (with six components, corresponding to position and orientation) at the end effector can now be expressed by the equation

$$\tau = J^T F,$$

where τ is an n-vector for the torques at each joint resulting from the force. We approximate the next configuration of the robot arm by updating each θ_i to the value $\theta_i + k\tau_i$, where k is a constant that can be thought of as the time step; note that we consider all joints to have unit mass.

3.1.2. Branching robot. Molecular structures are often more complicated than a simple open chain. We introduce the concept of a *branching robot* to accommodate such structures. A branching robot is precisely its namesake: joints connected by links in a tree-like fashion. For such a robot, there is a special *base*, corresponding to the *root* of the tree. Every other joint is connected to the base by links along a single well-defined branch or path.

Branching robots [9] have the following theory established for force propagation. When a force is applied at a joint, we can consider the branch from the base to that joint to be a simple robot arm; the techniques from section 3.1.1 are applied. If multiple forces are applied simultaneously, the resulting torques are additive and independent; thus, we simply accumulate the torques from each force before approximating the next configuration of the robot.

Breaking loops. Because our input structure is expected to have many interconnected loops, we initially break these constraints to associate a branching robot. We begin by viewing the structure as a simple graph on edges and vertices; a root and spanning tree are arbitrarily chosen for this graph (using a standard linear-time spanning tree algorithm, e.g. breadth-first-search), to allow the structure to be interpreted as a branching robot. Then we introduce several new vertices, to be used in the loop closure method, as follows. Consider the edges not included in the spanning tree; they correspond to loops that have been broken. For each such edge, we duplicate the endpoints; we call the duplicated endpoints the *loop vertices* and the corresponding originals their *partners*. See figure 2 for a simple 2D example; figure 3 contains a 3D example. Each duplicated endpoint is then attached to the opposite original loop endpoint, maintaining the tree constraint. This completes the construction of the branching robot.

3.2. Sampling the conformation space

We now describe the sampling of the conformation space. While we experimented with a few other approaches, what we present here is the most effective and efficient. Generation of a new sample consists of two stages: the first stage breaks the loop constraints, while the second restores them.

3.2.1. Stage 1: external force. The first stage applies a random external force to the branching robot; the effect of such a force breaks the loop constraints. See figure 3(*b*) for an example of the effect of such a force. Note that the lighter loop vertices are no longer aligned with their partners.

3.2.2. Stage 2: internal, attractive forces. Once the loop constraints have been violated, we use internal, attractive forces to close the loops. Each loop vertex and partner that are not aligned results in an attractive force between them; see figure 2(*c*) for a simple two-dimensional example illustrating this concept. We apply all such internal forces and iterate until the loop vertices and their partners align; note that we allow a small error tolerance. Because rotation about the loop edge is allowed, this satisfies the loop constraint. A more complex three-dimensional example of the loop closure approach is shown in figure 3.

4. Model molecules and results

To validate our method and compare timings, we present three model molecules. They correspond to subdivisions of the simplest platonic polyhedra: the tetrahedron, cube and octahedron. The *cube model molecule* $C_8S_{20}H_8$ was designed in [32] to demonstrate the effectiveness of exploring conformation space of molecules with many interconnected loops. We use it to compare our method with [32]. We have

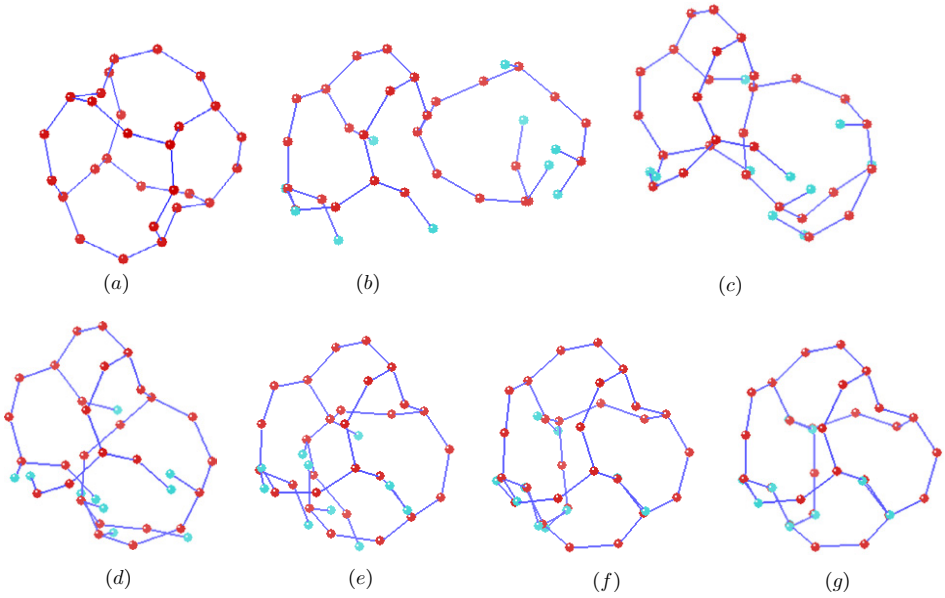

Figure 3. 3D example of loop closure approach. (*a*) Initial structure. (*b*) A random external force breaks loop constraints; note tree structure with (lighter) loop vertices. (*c*) Attractive, internal forces begin to close loops. (*d*) Loop closure. (*e*) Loop closure. (*f*) Loop closure. (*g*) Loops considered closed with a small error tolerance.

designed several other models starting from pure mathematical principles: the reader is advised that we did not intend them to have any chemical meaning. They have exactly two degrees of freedom, to allow for the effective visualization of the conformation space sampling. The broken loops correspond to faces of the corresponding polyhedron and allow for a reliable mathematical accounting of the stated degrees of freedom. They permit an easy replication of our computational experiments, which suggest that the method scales well with the number of broken loops. The geometry for our models was found using the freely licensed ArgusLab software [31], which also attached atomic interpretations to the nodes and added the hydrogen atoms connected to the corner nodes of our polyhedra.

4.1. Cube model molecule

Figure 4 depicts an example conformation of the molecule. It consists of eight carbon atoms (medium gray spheres) at the corners of a cube, with eight hydrogen atoms (small white spheres) attached to each; 20 sulfur atoms (large gold spheres) make up the rest of the molecule. The distances between carbon and sulfur, sulfur and sulfur, and carbon and hydrogen atoms are 1.805 Å, 2.019 Å and 1.120 Å, respectively. All bond angles about the carbon atoms are 109.5°; sulfur bond angles are 135°. As stated in [32], the molecule has two degrees of freedom, which can be represented as torsional angles θ_1 and θ_2 in figure 4.

4.1.1. Symmetries. The model molecule has two symmetries on θ_1 and θ_2.

Symmetry 1. $(\theta_1, \theta_2) \rightarrow (\theta_2, \theta_1)$

Symmetry 2. $(\theta_1, \theta_2) \rightarrow (-\theta_1, -\theta_2)$

We make use of them in our sampling method, as in [32].

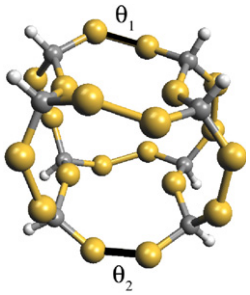

Figure 4. Model molecule $C_8S_{20}H_8$. There are eight carbon atoms (medium gray spheres) with eight hydrogen atoms (small white spheres) attached to each; 20 sulfur atoms (large gold spheres) make up the rest of the molecule. This molecule has two degrees of freedom, shown as torsional angles θ_1 and θ_2.

4.2. Tetrahedron and octahedron model molecules

The *tetrahedron model molecule* (depicted in figure 5(*a*)) has four carbon atoms (medium gray spheres) at the corners of a tetrahedron. Each carbon has a hydrogen atom (small white sphere) attached to it; 14 sulfur atoms (large gold spheres) lie along the edges of the tetrahedron. The *octahedron model molecule* is depicted in figure 5(*c*). An octahedron is defined by the six carbon atoms (medium gray spheres) with 32 sulfur atoms (large gold spheres) on the edges. It is interesting to note that the sampling results for the tetrahedron molecule show that not all the values of the sampled degrees of freedom are present in the real conformations of the molecule.

4.3. Results

Our system was implemented in Java and run on a Intel Pentium M 1600 MHz processor. Figure 6 depicts several

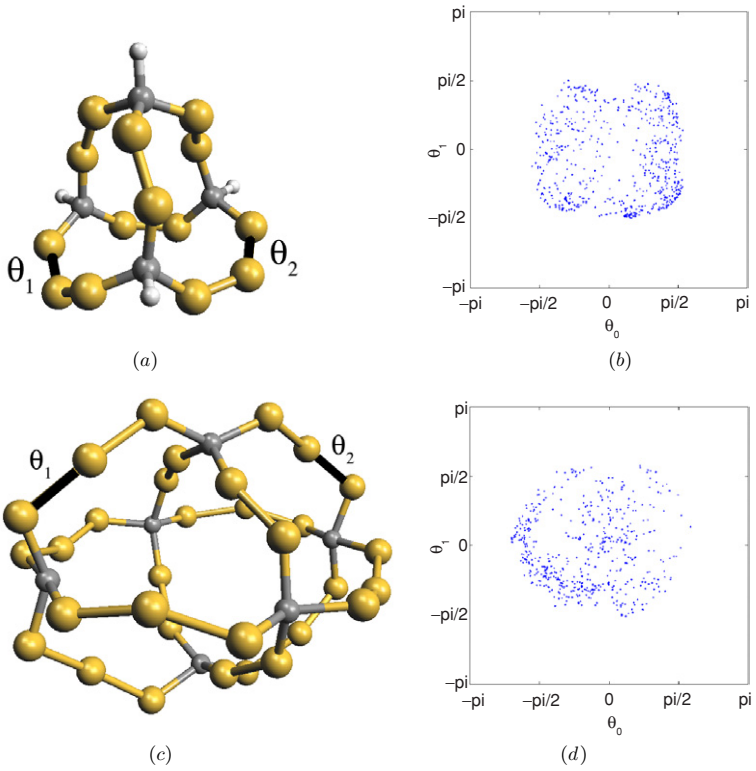

Figure 5. Tetrahedron and octahedron model molecules and results. (*a*) The tetrahedron model molecule has two degrees of freedom, shown as torsional angles θ_1 and θ_2. (*b*) Sampled conformation space of the tetrahedron model molecule. (*c*) The octahedron model molecule has two degrees of freedom, shown as torsional angles θ_1 and θ_2. (*d*) Sampled conformation space of the octahedron model molecule.

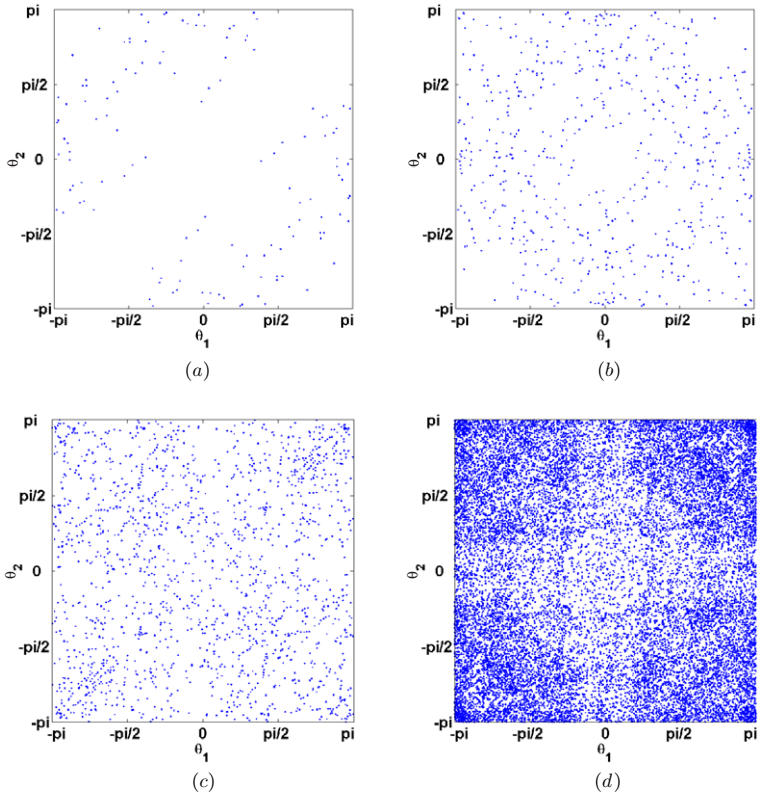

Figure 6. Sampled conformation space results; note that symmetries were applied to the generated conformations. (*a*) 32 conformations in 6 min. (*b*) 152 conformations in 22 min. (*c*) 415 conformations in 73 min. (*d*) 5022 conformations in 13 h, 27 min.

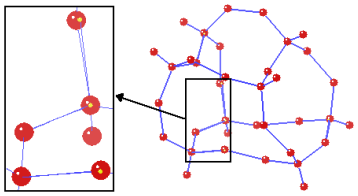

Figure 7. Generated conformation of model molecule. On the left is a zoom of several loop closures; the small vertices represent the loop vertices. Note the small amount of tolerated error.

Table 1. Comparison of three model molecules.

	Number of atoms	Number of bonds	Number of 'broken' loops	Time to generate 500 conformations (min)
Tetrahedron	22	24	3	60
Cube	36	40	5	100
Octahedron	38	44	7	130

results on the cube molecule, showing that our approach quickly begins sampling the space (see figure 6(*a*)); as more conformations are produced, the sampling becomes finer. Note that the previously discussed symmetries on the cube model molecule were applied. An example generated conformation along with a zoom of the loop closures is shown in figure 7.

Results for the tetrahedron and octahedron molecules are shown in figures 5(*b*) and 5(*d*), respectively. Table 1 compares results on all three molecules.

5. Conclusions and future work

We have presented a new approach based on robotics methodology for effectively sampling the conformation space of molecular structures, particularly those with interconnected loops. By modeling a molecule as a branching robot, we apply external forces to generate new conformations, and use attractive, internal forces to maintain loop closure constraints. While based on techniques from robotics, the background required in this field is minimal.

We are aware only of [32] and [10] as other algorithms for generating conformations of molecular structures with loop constraints taken into consideration. Both of them must deal with the problem that randomly perturbing portions of interconnected loops could result in an algebraic system with no solution; our approach is able to avoid the observed large rejection probability by using internal forces to pull loops closed.

The system, implemented in Java, was tested on several model molecules with many interconnected loops, but few degrees of freedom; even with no particular attention to code optimization, it performs almost four times as fast on the cube model molecule from [32]. Our results also show that the algorithm begins sampling the space very effectively within a short period of time (about 5 min), with the sampling becoming finer as more conformations are generated.

Future directions. While our results indicate significant speed-up in sampling conformational spaces, van der Waals overlap and energy calculations have not yet been considered in this paper. Their integration into our system, with a special attention paid to speed and accuracy and along with a systematic mathematical and bio-chemical validation, will be pursued next.

Acknowledgments

The authors would like to thank Dr Ming Lei for valuable discussions.

References

[1] Alves da Silva R, Degrève L and Caliri A 2004 LMProt: an efficient algorithm for Monte Carlo sampling of protein conformational space *Biophys. J.* **87** 1567–77
[2] Andersen H C 1983 Rattle: a 'velocity' version of the shake algorithm for molecular dynamics calculations *J. Comput. Phys.* **52** 24–34
[3] Berg B A and Neuhaus T 1992 Multicanonical ensemble: a new approach to simulate first-order phase transitions *Phys. Rev. Lett.* **68** 9–12
[4] Beusen D D, Shands E F B, Karasek S F, Marshall G R and Dammkoehler R A 1996 Systematic search in conformational analysis *J. Mol. Struct. (Theochem.)* **370** 157–71
[5] Blaney J M and Dixon J S 1994 Distance geometry in molecular modeling *Reviews in Computational Chemistry* vol 5 (Weinheim: VCH) pp 299–335
[6] Cahill M, Cahill S and Cahill K 2002 Proteins wriggle *Biophys. J.* **82** 2665–70
[7] Canutescu A A and Dunbrack R L Jr 2003 Cyclic coordinate descent: a robotics algorithm for protein loop closure *Protein Sci.* **12** 963–72
[8] Carlson H A and McCammon J A 2000 Accommodating protein flexibility in computational drug design *Mol. Pharmacol.* **57** 213–8
[9] Chang K-S 2000 Efficient algorithms for articulated branching mechanisms: dynamic modeling, control, and simulation *PhD Thesis* Department of Computer Science, Stanford University
[10] Cortés J, Siméon T, Ruiz de Angulo V, Guieysse D, Remaud-Siméon M and Tran V 2005 A path planning approach for computing large-amplitude motions of flexible molecules *Proc. 13th Annual Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*
[11] Cortés J, Siméon T and Laumond J P 2002 A random loop generator for planning the motions of closed kinematic chains using prm methods *Proc. IEEE Int. Conf. on Robotics and Automation*
[12] Craig J J 1989 *Introduction to Robotics: Mechanics and Control* (Boston, MA: Addison-Wesley Longman)
[13] Dodd L R, Boone T D and Theodorou D N 1993 A concerted rotation algorithm for atomistic Monte Carlo simulation of polymer melts and glasses *Mol. Phys.* **78** 961–96
[14] Frantz D D, Freeman D L and Doll J D 1990 Reducing quasi-ergodic behavior in Monte Carlo simulations by j-walking: applications to atomic clusters *J. Chem. Phys.* **93** 2769–84
[15] Gō N and Sheraga H A 1970 Ring closure and local conformational deformations of chain molecules *Macromolecules* **3** 178–87

[16] Han L and Amato N M 2000 A kinematics-based probabilistic roadmap method for closed chain systems *Workshop on the Algorithmic Foundations of Robotics (WAFR)* pp 233–46

[17] Hansmann U H E 1997 Parallel tempering algorithm for conformational studies of biological molecules *Chem. Phys. Lett.* **281** 140–50

[18] Kolodny R, Guibas L, Levitt M and Koehl P 2005 Inverse kinematics in biology: the protein loop closure problem *Int. J. Robot. Res.* **24** 151–62

[19] Kumar S, Rosenberg J M, Bouzida D, Swendsen R H and Kollman P A 1993 The weighted histogram analysis method for free-energy calculations on biomolecules: I. The method *J. Comput. Chem.* **13** 1011–21

[20] LaValle S M, Yakey J and Kavraki L 1999 A probabilistic roadmap approach for systems with closed kinematic chains *IEEE Int. Conf. on Robotics and Automation*

[21] Lee Y 1993 New Monte Carlo algorithm: entropic sampling *Phys. Rev. Lett.* **71** 211–4

[22] Lei M, Zavodszky M I, Kuhn L A and Thorpe M F 2004 Sampling protein conformations and pathways *J. Comput. Chem.* **25** 1133–48

[23] Lotan I, van den Bedem H, Deacon A M and Latombe J-C 2004 Computing protein structures from electron density maps: the missing loop problem *Workshop on the Algorithmic Foundations of Robotics (WAFR)* pp 153–68

[24] Metropolis N, Rosenbluth A W, Rosenbluth M N, Teller A N and Teller E 1954 Equation of state caluclations by fast computing machines *J. Chim. Phys.* **21** 1087–92

[25] Miyamota S and Kollman P A 1992 Settle: an analytical version of the Shake and Rattle algorithm for rigid water models *J. Comput. Chem.* **13** 952–62

[26] Rock: software for protein dynamics http://flexweb.la.asu.edu/rock_index.html

[27] Ryckaert J-P, Ciccotti G and Berendsen H J C 1977 Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of *n*-alkanes *J. Comput. Phys.* **23** 327–41

[28] Saunders M 1989 Stochastic search for the conformations of bicyclic hydrocarbons *J. Comput. Chem.* **10** 203–8

[29] Siepmann J I and Frenkel D 1992 Configurational bias Monte Carlo: a new sampling scheme for flexible chains *Mol. Phys.* **75** 59–70

[30] Swendsen R H and Wang J-S 1986 Replica Monte Carlo simulation of spin-glasses *Phys. Rev. Lett.* **57** 2607–9

[31] Thompson M A Arguslab computational chemistry software: a molecular modeling, graphics, and drug design program http://www.arguslab.com

[32] Thorpe M F and Lei Ming 2004 Macromolecular flexibility *Phil. Mag.* **84** 1323–31

[33] van Gunsteren W F and Berendsen H J C 1990 Computer simulation of molecular dynamics: methodology, applications and perspectives in chemistry *Angew. Chem. Int.* **29** 992–1023

[34] Xu H and Berne B J 1999 Multicanonical jump walking: a method for efficiently sampling rough energy landscapes *J. Chem. Phys.* **110** 10299–306

[35] Zhang Y, Kihara D and Skolnick J 2002 Local energy landscape flattening: parallel hyperbolic Monte Carlo sampling of protein folding *Proteins: Struct., Funct. Genet.* **48** 192–201

[36] Zhou R and Berne B J 1997 Smart walking: a new method for Boltzmann sampling of protein conformations *J. Chem. Phys.* **107** 9185–96