

Controlling Humanoid Robots in Topology Coordinates

Edmond S.L. Ho, Taku Komura, Subramanian Ramamoorthy, Sethu Vijayakumar

{s.l.ho,tkomura,s.ramamoorthy,sethu.vijayakumar}@ed.ac.uk

Abstract—This paper presents an approach to the control of humanoid robot motion, e.g., holding another robot or tangled interactions involving multiple limbs, in a space defined by 'topology coordinates'. The constraints of tangling can be linearized at every frame of motion synthesis, and can be used together with constraints such as defined by the Zero Moment Point, Center of Mass, inverse kinematics and angular momentum for computing the postures by a linear programming procedure. We demonstrate the utility of this approach using the simulator for the Nao humanoid robot. We show that this approach enables us to synthesize complex motion, such as tangling, very efficiently.

I. INTRODUCTION

One of the major drivers of research in the area of humanoid robotics is the desire to achieve motions involving close contact between robots and the environment or people, such as while carrying an injured person, handling flexible objects such as the straps of a knapsack or clothes. Currently, these applications seem beyond the ability of existing motion synthesis techniques due to the underlying computational complexity in an open-ended environment.

Traditional methods for motion synthesis suffer from two major bottlenecks. Firstly, a significant amount of computation is required for collision detection and obstacle avoidance in the presence of numerous close contacts between manipulator segments and objects. Secondly, any particular computed solution can easily become invalid as the environment changes. For instance, if the robot were handling an object such as a knapsack, even small deformations of this flexible object and minor changes in object dimensions (e.g., between an empty bag and a stuffed bag) requires complete re-planning in the current way of solving the problem.

The fundamental problem lies in the representation of the state of the world and the robot. Typically, motion is synthesized in a complete configuration or state space represented at the level of generalized coordinates enumerating all joint angles and their 3D location/orientation with respect to some world reference frame. This implies the need for large amounts of collision checking calculations and randomized exploration in a very large search space. Moreover, it is very hard to encode higher level, semantic, specifications at this level of description as the individual values of the generalized coordinates do not tell us anything about this aspect of the task unless further calculations are carried out to ensure satisfaction of further constraints. This is particularly inconvenient when searching for a motion in a large database

such as might be obtained from human demonstration or other related means.

The focus of this research is to alleviate these problems by developing methods that exploit the underlying topological structure in these problems, e.g., in the space of postures. This allows us to define a new search space where the coordinates are based on spatial relationships, such as between link segments. We refer to this space in terms of 'topology coordinates' [7]. In this paper, we show how this notion can be applied to the control of humanoid robots. Using the Nao [5] humanoid robot, we show that motions such as holding and carrying can easily be synthesized in real-time without recourse to computationally expensive global search methods.

Background

Motion synthesis and generalization to novel situations is an actively researched topic in robotics, character animation and graphics. Here, we review some state-of-the-art methods and identify some of the key issues that form the focus of our own work.

A standard and commonly used method for motion synthesis involves a global path planning problem to compute collision free paths between start and goal states in a configuration space. Given that deterministic versions of such problems suffer from severe computational complexity restrictions, randomized approximations are preferred. Randomized methods such as Rapidly-exploring Random Trees (RRT)[10] have been developed to solve such problems. Motions such as moving objects [20] and grabbing [1] have been synthesized by such methods. However, although the randomized algorithms are more efficient than the deterministic counterpart the fact remains that the computational cost scales exponentially with the number of degrees of freedom that define the problem. In problems involving close contact between complex articulated mechanisms and objects, these numbers can quickly grow.

A promising approach to overcoming this problem is to define the task in terms of low-dimensional representations, separating the complexity of enumerating the environment from the essence of the task. Our interest is in low-dimensional representations that describe the topology of the task (abstracting away from many of the metric details). Once an approximate valid path is found in this way, optimization-based motion synthesis approaches may be used to synthesize the detailed trajectory that gets executed on the high-dimensional system. Spacetime optimization [19] is one such approach that synthesize motion by minimizing

The authors are with the Institute of Perception, Action and Behaviour in the School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, UK

a specific objective function while satisfying constraints. It has been used to solve for positional constraints [4], generate physically-based animation [12], [3], and stabilize the gait of biped humanoid robots [13]. However, optimization-based techniques also suffer from the same complexity limitations when handling collisions or motion with many close contacts. We argue that the situation is significantly improved by optimizing the motion directly in topology space, enabling us to efficiently produce motion sequences such as for holding, carrying, tangling and grabbing.

Although prior work on topology-based motion synthesis is limited, there have indeed been some interesting successes. A recent example in the area of manipulation is that of knotting [17], [16], where the object state is represented in terms of projections to a 2-dim plane. While this is an interesting direction, one needs a more general representation to deal with extended objects such as cloth and also when the task is defined in a cluttered environment. Topological representations are also beginning to be used successfully in the area of distributed robotics, e.g., to ensure coverage in sensor networks [2]. Our pilot studies [6], [7] have realized techniques for computing topological relationship of objects directly from the 3-dim coordinates of the robot body by using the mathematical notion of Gauss Linking Integrals (GLI) [11] (Fig. Figure 1).

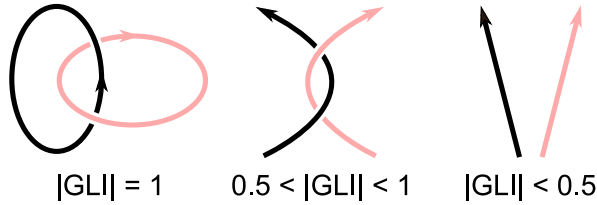


Fig. 1. The GLI outputs of two tangled strands

This has significant advantages over previous knot theory based approaches [17], [16] which all require the projection to a 2-dim plane. While this assumption is sufficient for closed knots or links, in which the projection plane does not affect invariants such as the minimum crossing numbers (Figure 2, left), the projection plane does affect the number of crossings in open curves or tangles (Figure 2, right). Defining a projection plane is difficult in case of humanoid robots as the bodies are always moving around and their orientations are continually changing. On the other hand, the Gauss Linking Integral can be directly computed from the 3-dim coordinates of the tangled links .

II. METHODOLOGY

We first briefly review the notion of topology coordinates [7], which is the basic representation used for the control of tangling motions. Next, we explain how the topology coordinates can be exploited for the control of humanoid robots.

A. Topology Coordinates

Topology coordinates enable characters to tangle their bodies with other characters without recourse to expensive

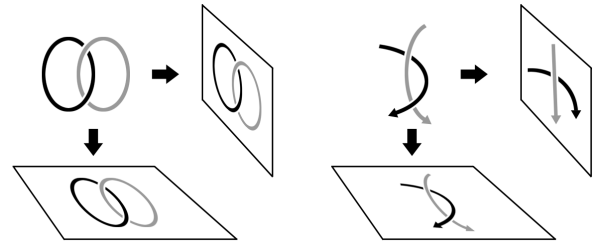


Fig. 2. The projection plane does not affect the minimum crossing numbers for links (left) but does for tangles (right)

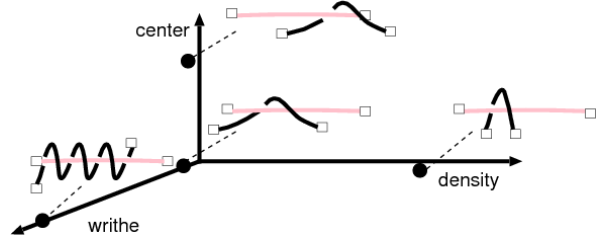


Fig. 3. The three axes in 'topology space' : writhe, center and density. The center, which specifies the central location of the twist relative to each strand, is actually composed of two scalar parameters, although it is represented by a single axis in this figure. The density tells which strand is playing the major role to compose the twist.

global path planning. The topology coordinates are composed of three attributes: writhe, center and density. The first attribute **writhe** counts how much the two curves are twisting around each other. Writhe can be calculated by using Gauss Linking Integral (GLI) [15] by integrating along the two curves γ_1 and γ_2 as:

$$GLI(\gamma_1, \gamma_2) = \frac{1}{4\pi} \int_{\gamma_1} \int_{\gamma_2} \frac{d\gamma_1 \times d\gamma_2 \cdot (\gamma_1 - \gamma_2)}{\|\gamma_1 - \gamma_2\|^3} \quad (1)$$

where \times and \cdot are cross product and dot product operators, respectively. The GLI computes the average number of crossings when viewing the tangle from all directions.

Curves can twist around each other in various ways. In order to further specify the status of the two chains, we introduce the other two attributes, **center** and **density**. Examples of changing these attributes for a pair of strands are shown in Figure 3. The center, which is composed of two scalar parameters, explains the center location of the twisted area, relative to each strand. The density, which is a single scalar parameter, explains how much the twisted area is concentrated at one location along the strands. When the density is zero, the twist is spread out all over the two strands. When the density value is either very large or very small, we can say one strand is playing a major role to compose the twist, as it is twisting around the other strand which is kept relatively straight (Figure 3). When the density turns from negative to positive, or vice versa, the strand playing the major role switches.

B. Controlling the Humanoid Robots in Topology Coordinates

We first represent the skeletal structure of a humanoid robot by a set of line segments. Therefore, we first define the topology coordinates of serial chains. Let us assume we have two chains S_1 and S_2 , each composed of n_1 and n_2 line segments, connected by revolute, universal or gimbal joints (Figure 4). In this case, we can compute the total writhe by summing the writhes by each pair of segments:

$$w = GLI(S_1, S_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} T_{i,j} \quad (2)$$

where w represents the writhe, $T_{i,j}$ is the writhe between segment i on S_1 and j on S_2 . Let us define a $n_1 \times n_2$ matrix \mathbf{T} whose (i, j) -th element is $T_{i,j}$, and call this the **writhe matrix**. The writhe matrix explains how much each pair of segments from S_1 and S_2 contributes to the total writhe value. Various twists of two serial chains and the corresponding writhe matrices are shown in Figure 5.

The topology coordinates can be updated by changing the distribution of the elements in the writhe matrix using basic operations such as rotation, translation and scaling. Rotating the elements results in changing the density. Translating the elements results in changing the center. Scaling the whole matrix results in changing the writhe. Let us define these operations by $R(M, d)$, $Tr(M, c)$ and $S(M, w)$, respectively, where M is the input matrix and (d, c, w) are topology coordinates, each of which representing the density, center and writhe, respectively.

Rather than directly manipulating the writhe matrix of the characters, we first compute an ideal, desired writhe matrix and try to minimize the difference of the character's writhe matrix and the desired writhe matrix. The desired writhe matrix T_d that corresponds to topology coordinates (d, c, w) is computed by sequentially applying $R()$, $Tr()$ and $S()$ to a matrix \mathbf{I} , which is a $n_1 \times n_2$ matrix who has values evenly distributed at the $(n_2 + 1)/2$ -th column if n_2 is odd, or at both the $n_2/2$ and $n_2/2 + 1$ -th column if it is even:

$$\mathbf{T} = S(Tr(R(\mathbf{I}, d - \frac{\pi}{4}), \mathbf{c}), w). \quad (3)$$

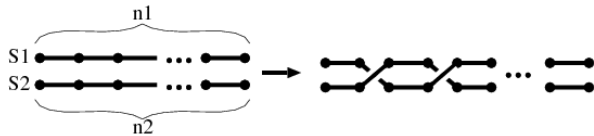


Fig. 4. Twisting a chain of line segments around each other

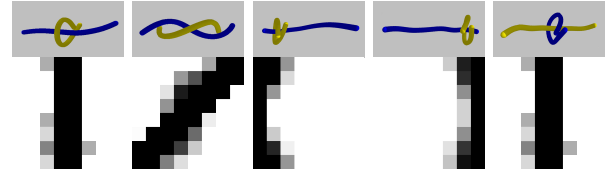


Fig. 5. (upper) Tangles with different density and center, and (lower) the distribution of elements with large absolute values in the corresponding writhe matrices. The level of brightness represents the amplitude of the absolute value.

where

$$\mathbf{I} = \begin{cases} \begin{pmatrix} 0 \cdots, \frac{1}{n_1}, \cdots, 0 \\ \vdots \\ 0 \cdots, \frac{1}{n_1}, \cdots, 0 \end{pmatrix} & (n_2 \text{ is odd}) \\ \begin{pmatrix} 0 \cdots, \frac{1}{2n_1}, \frac{1}{2n_1}, \cdots, 0 \\ \vdots \\ 0 \cdots, \frac{1}{2n_1}, \frac{1}{2n_1}, \cdots, 0 \end{pmatrix} & (n_2 \text{ is even}) \end{cases} \quad (4)$$

and $\frac{\pi}{4}$ is an offset to adjust the density d due to its definition [7].

Once the desired writhe matrix T_d is computed, the character is guided to the desired posture by updating the generalized coordinates so that the writhe matrix of the character T becomes similar to the desired writhe matrix T_d . This problem is solved by quadratic programming:

$$\min_{\Delta \mathbf{q}_1, \Delta \mathbf{q}_2, \delta} \|\Delta \mathbf{q}_1\|^2 + \|\Delta \mathbf{q}_2\|^2 + \|\delta\|^2 \text{ s.t.} \quad (5)$$

$$\Delta \mathbf{T} = \frac{\partial \mathbf{T}}{\partial \mathbf{q}_1} \Delta \mathbf{q}_1 + \frac{\partial \mathbf{T}}{\partial \mathbf{q}_2} \Delta \mathbf{q}_2 \quad (6)$$

$$|T_{i,j} + \Delta T_{i,j}| \leq \sigma (1 \leq i \leq n_1, 1 \leq j \leq n_2) \quad (7)$$

$$\mathbf{T} + \Delta \mathbf{T} - \mathbf{T}^d + \delta = \mathbf{0} \quad (8)$$

$$\mathbf{r} = \mathbf{J}_1 \Delta \mathbf{q}_1 + \mathbf{J}_2 \Delta \mathbf{q}_2 \quad (9)$$

where $(\mathbf{q}_1, \mathbf{q}_2)$ are the generalized coordinates of the two chains, $(\Delta \mathbf{q}_1, \Delta \mathbf{q}_2)$ are their updates to be made at this iteration, $\Delta \mathbf{T}_d$ is the update of the writhe matrix, σ is a threshold, that is set to 0.2 in our experiments to avoid the segments to approach too close to each other, δ is a matrix of slack parameters introduced to minimize the difference of the desired writhe matrix and that of the controlled characters, Equation (9) represents the other kinematical constraints which can be linearized with respect to $(\Delta \mathbf{q}_1, \Delta \mathbf{q}_2)$ when the movement is small, such as the positional constraints, center of mass constraints, angular momentum constraints and ZMP constraints [9], [8]. $\mathbf{J}_1, \mathbf{J}_2$ are the Jacobians of this constraint, and \mathbf{r} is the linearized output of this constraint. The updated generalized coordinates $(\mathbf{q}_1 + \Delta \mathbf{q}_1, \mathbf{q}_2 + \Delta \mathbf{q}_2)$ correspond to the target topology coordinate at the next time step, $(w + \Delta w, d + \Delta d, \mathbf{c} + \Delta \mathbf{c})$. By solving Equation (5) at every time step, we simulate the interactions of two virtual wrestlers.

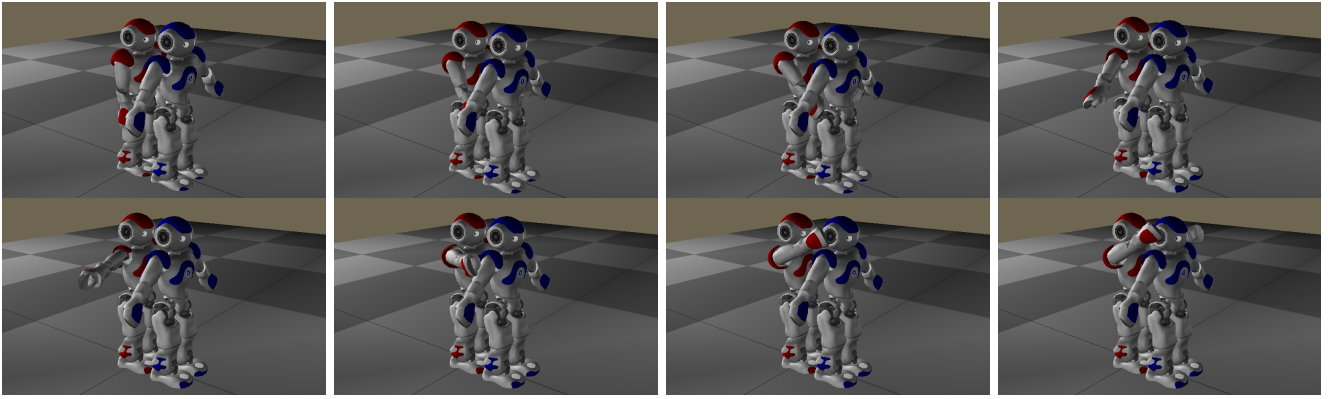


Fig. 7. The snapshots of the holding motion from the upright postures (top), and the movements to switch to the posture of holding the neck (neck).

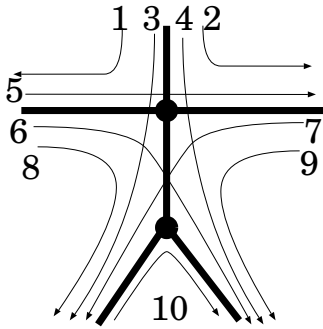


Fig. 6. The tree structure of the graph that is used to represent the body structure. There are 10 paths that connect the end effectors.

C. Tangles of tree structures

As we need to handle human characters, we have to compute the tangles made between tree structures. Trees are composed of edges and nodes. Therefore, tangles made between them will be more complex than those between parts of strings. The tangles between trees can be examined by checking all the tangles made between the paths connecting the end effectors of the trees. The graph structure that is used to represent the human body in this research is shown in Figure 6. There are ten paths connecting the end effectors of this graph.

III. EXPERIMENTAL RESULTS

Our experiments involve the Nao V3 model [5] in the Webots simulator [18], to synthesize tangling motions of two robots holding each other - starting from postures of one robot standing in front of another. We have also synthesized motions to switch from this posture to another posture of holding the neck by linearly interpolating the topology coordinates. Note that if these motion sequences were to be computed directly in joint space, one would need to solve a global planning problem in a space with multiple local minima, etc.

Key snapshots of the movement are shown in Figure 7. The computation of the motion can be accomplished in 31ms per frame, on a computer with Intel Core2 Duo Mobile Processor

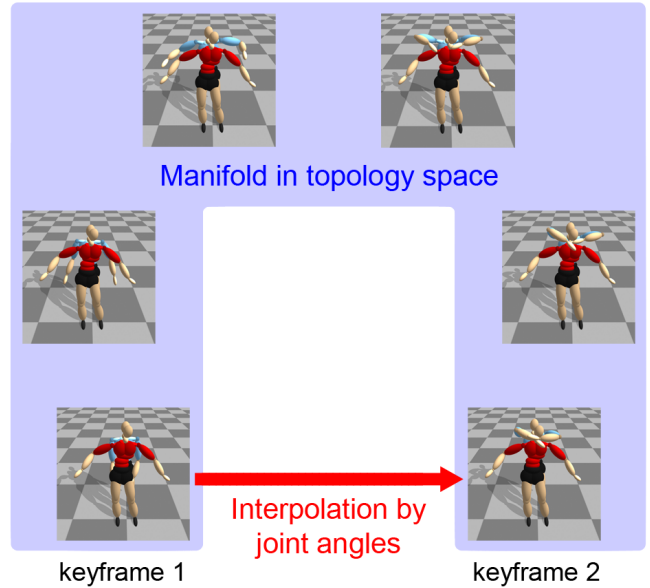


Fig. 8. The motion to interpolate two keyframes along the manifold of motion space (top) and in the joint angle space (bottom).

P7450 (2.13 GHz) and with 4GB RAM. All the calculation was done on a single core. The system is implemented on Windows by Visual C++, UMFPACK is used for the linear solver.

IV. DISCUSSIONS AND CONCLUSION

The problem with planning motion at the level of joint angles is that the topological structure of the state space is different from the more intuitive variables that we might otherwise invoke. Continuous motions in the joint angle space can be discontinuous in the humanoid motion space, due to limitations of collision and penetration of the body parts.

For example, let us think of interpolating two keyframe postures shown in Figure 8. These two postures are not very far away from each other in the joint angle space. However, if they are linearly interpolated by joint angles, the arms of the humanoids will penetrate through each other, which means

that the motion goes out from the feasible portions of the humanoid motion space. Our ‘topology coordinates’ define a subspace which has the same topological structure (in the sense of homotopy equivalence) as that of the motion space. A continuous motion in the topology coordinates results in a continuous motion by the humanoid robot and vice versa. As a result, complex motions of tangling movements can be easily synthesized by interpolating postures by topology coordinates.

Abstraction of movements is important for humanoid control due to the high number of degrees of freedom and the associated exponential computational cost for motion planning. Topology-based abstraction provides an intuitive solution for tasks such as those involving tangled postures. For tangled postures, people are not really interested in the individual joint angles or joint positions but the way the limbs and body parts are tangled with each other. By using the topology coordinates as constraints, the humanoid can easily adjust their postures in real-time to maintain the tangled state even when the other humanoid is moving around in an arbitrary way. The topology coordinates can be consistent under various conditions, such as when the humanoid robots are replaced with those of totally different morphology. Such a feature can greatly help to recycle motions.

As topology coordinates represent spatial relationships, they can also help to plan/evaluate in larger scale scenes. The parameters of the topology coordinates have semantic meanings which are easy to qualitatively interpret and explain the scene [14]. This can greatly help to synthesize movements from scripts or conversations, which is a key long term goal for intelligent robotics.

Combining our method with stochastic approaches that takes into account the effect of noise is another interesting direction for further research. We are also currently working on representations based on the relationship of surfaces, which can consider the volume and area of the humanoid robots at the stage of motion planning.

REFERENCES

- [1] Dmitry Berenso, Rosen Diankovage, Koichi Nishiwaki, Satoshi Kagami, and James Kuffner. Grasp planning in complex scenes. *IEEE/RAS Humanoids*, 2007.
- [2] V. de Silva and R. Ghrist. Coordinate-free coverage in sensor networks with controlled boundaries via homology. *Intl. J. Robotics Research*, 25(12):1205–1222, 2006.
- [3] Anthony C. Fang and Nancy S. Pollard. Efficient synthesis of physically valid human motion. *ACM Transactions on Graphics*, 22(3):417–426, 2003.
- [4] Michael Gleicher. Retargetting motion to new characters. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 33–42, New York, NY, USA, 1998. ACM Press.
- [5] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jérôme Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre, and Bruno Maisonnier. Mechatronic design of nao humanoid. In *ICRA'09: Proceedings of the 2009 IEEE international conference on Robotics and Automation*, pages 2124–2129, Piscataway, NJ, USA, 2009. IEEE Press.
- [6] Edmond S. L. Ho and Taku Komura. Wrestle alone: Creating tangled motions of multiple avatars from individually captured motions. In *Proceedings of Pacific Graphics 2007*, pages 427–430, 2007.
- [7] Edmond S. L. Ho and Taku Komura. Character motion synthesis by topology coordinates. *Computer Graphics Forum (Special issue of Eurographics 2009)*, 28(2), 2009.
- [8] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Kazuhito Yokoi, and Hiroshi Hirukawa. Resolved momentum control: humanoid motion planning based on the linear and angular momentum. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1644–1650, 2003.
- [9] Shunsuke Kudoh and Taku Komura. C^2 continuous gait-pattern generation for biped robots. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1644–1650, 2003.
- [10] S.M. LaValle and J.J. Kuffner. Rapidly-exploring random trees: Progress and prospects. *Robotics: The Algorithmic Perspective. 4th Int'l Workshop on the Algorithmic Foundations of Robotics*, 2001.
- [11] Michael Levitt. Protein folding by restrained energy minimization and molecular dynamics. *Journal of Molecular Biology*, 170:723–764, 1983.
- [12] C. Karen Liu and Zoran Popović'. Synthesis of complex dynamic character motion from simple animations. *ACM Transactions on Graphics*, 21(3):408–416, 2002.
- [13] Koichi Nishiwaki, Tomomichi Sugihara, Satoshi Kagami, Masayuki Inaba, and Hirochika Inoue. Online mixture and connection of basic motions for humanoid walking control by footprint specification. *2001 IEEE International Conference on Robotics and Automation (ICRA'01)*, pages 4110–4115, 2001.
- [14] Hanna M. Pasula, Luke S. Zettermoyer, and Leslie Pack Kaelbling. Learning symbolic models of stochastic domain. *Journal of Artificial Intelligence Research*, 29:309–352, 2007.
- [15] W.F. Pohl. The self-linking number of a closed space curve. *Journal of Mathematics and Mechanics*, 17:975–985, 1968.
- [16] Mitul Saha and Pekka Isto. Manipulation planning for deformable linear objects. *IEEE Transactions on Robotics*, 23(6):1141–1150, 2007.
- [17] Jun Takamatsu, Takuma Morita, Koichi Ogawara, Hiroshi Kimura, and Katsushi Ikeuchi. Representation for knot-tying tasks. *IEEE Transactions on Robotics*, 22(1):65–78, 2006.
- [18] Webots. <http://www.cyberbotics.com>. Commercial Mobile Robot Simulation Software.
- [19] Andrew Witkin and Michael Kass. Spacetime constraints. *Computer Graphics (Proceedings of SIGGRAPH 88)*, 22:159–168, 1988.
- [20] K. Yamane, J. Kuffner, and J.K. Hodgins. Synthesizing animations of human manipulation tasks. *ACM Transactions on Graphics*, 23(3):532–539, 2004.