Proceedings of the 2006 IEEE International Conference on Robotics and Automation
Orlando, Florida - May 2006

Discrete approximations to continuous curves

Sean B. Andersson
Department of Aerospace and Mechanical Engineering
Boston University, Boston, MA 02215
Email: sanderss@bu.edu

***Abstract*— We consider the problem of approximating a continuous curve by a piecewise linear one whose segments are assumed to be connected by universal joints. Rather than taking a least-squares approach, we require that the endpoints of the line segments lie on the continuous curve. We show that under these assumptions a single rotational degree of freedom remains. An algorithm is derived to determine the set of angles characterizing the relative orientation of each consecutive pair of line segments as a function of this rotational degree of freedom. Two examples are given to illustrate the procedure. The motivating application for this work is the control of a snake-like robot using a set of gaits designed from shape primitives.**

I. Introduction

In this paper we consider the problem of approximating a continuous parameterized 3-D curve with a piecewise linear one. The approximating curve is assumed to consist of N line segments of known lengths $L_1, \ldots, L_N$. We further assume that adjacent segments are connected at their end points using a universal joint. The problem is then one of finding the position and orientation of the first line segment with respect to a fixed world frame and the relative orientations of all subsequent segments such that the discrete curve approximates the continuous one in an appropriate sense.

There are many ways in which one could choose to approximate the original curve. For instance, one could seek to minimize some notion of distance between the continuous and the discrete curves. This technique often leads to a numerical solution which is computationally expensive and thus not well-suited for real-time implementation. In this work we take a different approach and assume that the end points of the line segments should lie on the curve to be approximated. Under this assumption we are able to solve the problem by splitting it into two stages. In the first stage, the locations of the end points along the original curve are determined. This problem is solved numerically using a bisection search method. In the second stage, the orientations of the line segments are determined such that the end points are placed at the desired locations. To solve this problem, the sequence of line segments is first modeled as a kinematic chain. We then take advantage of the product of exponentials formulation [1], [2] to derive expressions for the relative orientations of the segments as a function of a single degree of freedom, namely a rotation of the first segment around the axis defined by its end points. This rotation can be chosen arbitrarily or used to optimize some secondary task.

The motivation for this work lies in the control of snake-like (or hyper-redundant) mobile robots. These mechanisms provide a flexibility which is difficult or impossible to achieve with other locomotion modalities. They are well-suited for unstructured and highly cluttered environments, such as in the rubble of a collapsed building, and can, in principle at least, climb poles or stairs, navigate through narrow openings, and even manipulate objects. While the flexibility of a snake-like design makes it an interesting system to consider, it also makes it a difficult one to control. Determining how the system interacts and moves through the environment is a complicated task. One general approach which is useful in this setting is the notion of gait-based control [3]. This technique has been applied successfully to many novel locomotion systems, including hyper-redundant mechanisms [4], eel-like robots [5], [6], and a polychaete annelid robot [7].

As in the work of Chirikjian and Burdick [8], it is our view that it is useful to design gaits and shapes at the level of a continuum snake, ignoring at first the details of the discrete nature of the mechanism. Such an approach allows the control designer to take advantage of solutions found in nature (biomimetic design) as well as to use their own intuition. The work presented in this paper is aimed at translating the desired continuous curve into actuator positions such that the shape of the robot achieves the desired form. While the solution presented here is specific to one particular mechanism design, the fundamental idea is one of imposing suitable restrictions to reduce the complexity of the problem. This general approach, as well as the tools developed here based on the product-of-exponentials formulation, are thus useful for a variety of mechanism designs.

The remainder of this paper is organized as follows. In the next section we present a brief review of the product of exponentials representation of a serial kinematic chain and then derive the representation for the piecewise linear approximating curve. In Sec. III we discuss the general problem and our approach to solving it. The solution itself is described in Sec. IV together with examples to illustrate the procedure.

II. Robot kinematics

We give here a very brief description of a geometric approach to kinematic chains. For a more detailed description of the following see [1] or [2].

A serial kinematic chain is a series of rigid links with adjacent links connected by a joint with a single degree of freedom. Joints with additional degrees of freedom are modeled by allowing the distance between subsequent joints to be zero. To describe the configuration of the chain, a frame is attached to each joint and the position and orientation of each of these frames is specified with respect to a (fixed) world-reference frame. The configuration of a joint frame with respect to the world frame is given by an element of the special Euclidean group, $SE(3)$, the Lie group of rigid rotations and translations in 3-D. The configuration of the j^{th} frame with respect to the world frame is denoted $g_{w,j}$. In *homogeneous coordinates*, this configuration is given by

$$g_{w,j} = \begin{bmatrix} R_{w,j} & p_j \\ 0 & 1 \end{bmatrix} \tag{1}$$

where $R_{w,j} \in SO(3)$ is a rotation matrix and $p_j \in \mathbb{R}^3$ is the vector from the origin of the world frame to the origin of the jth frame. A point $q \in \mathbb{R}^3$ may also be expressed in homogeneous coordinates. This is denoted as $\bar{q}$ and is given by

$$\bar{q} = \begin{bmatrix} q \\ 1 \end{bmatrix}. \tag{2}$$

The configuration $g_{w,j}$ can then be viewed as a rigid transformation which acts on a point $\bar{q}$ expressed in the jth frame by $g_{w,j}\bar{q}$ to yield the homogeneous coordinates of the point q with respect to the world frame. Similarly the transformation $g_{i,j}$ denotes the mapping from the jth frame to the ith. These mappings can be concatenated, i.e. $g_{j-2,j-1}g_{j-1,j} = g_{j-2,j}$.

Because subsequent links are connected by a single degree of freedom, the configuration of the jth frame with respect to the previous one, $g_{j-1,j}$ is given by a one-parameter family of transformations. The parameter is often referred to as the *joint angle*, regardless of the type of joint. To represent this one-parameter family, we take advantage of the fact that any element of $SE(3)$ can be described as a *screw*, that is a rotation and translation along some axis. In turn, a screw can be described by the direction of rotation and translation, known as a *twist*, and by the amount of twist, θ. When the screw describes the configuration $g_{j-1.j}$ then the joint angle is precisely θ_{j-1}.

To express a twist and a screw, we need to define the hat map, a notation used here to denote two different mappings. First, it denotes a mapping from $\mathbb{R}^3$ into the space of 3×3 skew-symmetric matrices (the Lie algebra $so(3)$). This mapping is given by

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^{\wedge} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{3}$$

Note that for $\omega, p \in \mathbb{R}^3$ we have $\hat{\omega}p = \omega \times p$. Second, the hat map denotes a mapping from $\mathbb{R}^6$ into the Lie algebra $se(3)$, given by

$$\begin{bmatrix} v \\ \omega \end{bmatrix}^{\wedge} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix} \tag{4}$$

where $v = (v_1, v_2, v_3) \in \mathbb{R}^3$ and $\omega = (\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3$. The meaning of the notation should be clear from context.

The twist for joint j is denoted ξ_j. For a revolute joint, the twist is given by

$$\xi_j = \begin{bmatrix} -\omega_j \times q_j \\ \omega_j \end{bmatrix} \tag{5}$$

where ω_j is a unit vector along the axis of rotation and q_j is any point on that axis of rotation. For a prismatic joint, the twist is given by

$$\xi_j = \begin{bmatrix} v_j \\ 0 \end{bmatrix} \tag{6}$$

where v_j is a unit vector along the axis of translation. In either case the twist can be viewed as an element of the Lie algebra $se(3)$.

The screw corresponding to the jth joint is given by the exponential mapping of $se(3)$ into $SE(3)$. It can be shown that if ξ_j corresponds to a revolute joint then

$$e^{\hat{\xi}_j\theta_j} = \begin{bmatrix} e^{\hat{\omega}_j\theta_j} & \left(\mathbb{I} - e^{\hat{\omega}_j\theta_j}\right)(\omega_j \times v_j) \\ 0 & 1 \end{bmatrix} \tag{7}$$

where $\mathbb{I}$ is the identity matrix, $e^{\hat{\omega}_j\theta_j}$ is the matrix exponential of the skew symmetric matrix $\hat{\omega}_j\theta_j$ and $v_j = -\omega_j \times q_j$. If the joint is prismatic then

$$e^{\hat{\xi}_j\theta_j} = \begin{bmatrix} \mathbb{I} & v_j\theta_j \\ 0 & 1 \end{bmatrix}. \tag{8}$$

We can now express the rigid transformation of the first j joints with respect to the world frame by concatenation. Let θ denote the vector of all the joint angles and let $g_{w,j}(0)$ denote the configuration of the jth frame when all the joint angles are set to zero. Then

$$g_{w,j}(\theta) = e^{\hat{\xi}_1\theta_1}e^{\hat{\xi}_2\theta_2}\dots e^{\hat{\xi}_j\theta_j}g_{w,j}(0). \tag{9}$$

For the purposes of this work we assume that the discrete curve is comprised of $N-1$ line segments (links) connecting N blocks. Each block is connected at its center to the end point of a line segment by a single degree of freedom revolute joint; since each internal block is assumed to be connected to two segments along orthogonal axes, it represents a universal joint. To each joint we attach a frame with origin at the center of the corresponding block. The local $x-$axis is defined to be pointing towards the center of the subsequent block. The $y-$ and $z-$ axes are defined by the axes of rotation for the joints. We label the joints from 1 to $2N-2$ and define the joint angles to be such that the line is straight when all of the joints are set to zero. Without loss of generality we assume the joint on the first link rotates about the $z-$axis of the first frame, the second joint rotates about the $y-$axis of the second frame, the third about the $z-$axis of the third frame, and so on. Thus each odd-numbered joint is a rotation about the local $z-$axis and each even-numbered joint is a rotation about the local $y-$axis. A diagram of a discrete curve with 15 blocks is shown in Fig. 1.

To describe the internal *shape* of the discrete curve, we specify the configuration of all the frames with respect to the first frame, using (9) where the world frame is replaced with the first frame. To relate the configuration of the jth frame to

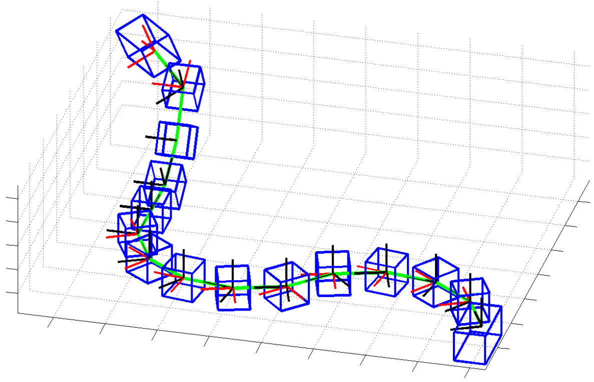

Fig. 1. An example of a kinematic chain with 15 blocks (14 links). The first and final links are connected to the chain with revolute joints while the remaining links are connected using universal joints, modeled as two revolute joints with orthogonal and intersecting axes. A frame is associated to each joint.

a given world frame, the transformation from the first frame to the world may be viewed as a (fixed) joint with associated transformation $g_{w,1}$. Thus

$$g_{w,j}(\theta) = g_{w,1} e^{\hat{\xi}_1 \theta_1} \cdots e^{\hat{\xi}_j \theta_j} g_{1,j}(0). \tag{10}$$

Since the reference frame is the first frame, we have

$$g_{1,j}(0) = \begin{bmatrix} 1 & 0 & 0 & \sum_{k=1}^{\lfloor \frac{j}{2} \rfloor} L_k \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

where $\lfloor \cdot \rfloor$ indicates the floor function. Each joint is a revolute joint with the axis of rotation given by

$$\omega_j = \begin{cases} (0,0,1)^T, & j = 1, 3, \ldots \\ (0,1,0)^T, & j = 2, 4, \ldots . \end{cases} \tag{12}$$

To express the twist of the jth joint as in (5), we must choose a vector q_j on the axis of rotation. We choose the origin of frame j (in the reference configuration), i.e.

$$q_j = \left(\sum_{k=1}^{\lfloor \frac{j}{2} \rfloor} L_k, 0, 0 \right)^T . \tag{13}$$

The twist for each joint is then given by

$$\xi_j = \begin{cases} \left(0, -\left(\frac{j-1}{2}\right) L, 0, 0, 0, 1\right)^T, & j = 1, 3, \ldots \\ \left(0, 0, \left(\frac{j}{2}\right) L, 0, 1, 0\right)^T, & j = 2, 4, \ldots . \end{cases} \tag{14}$$

Define $c\theta_j \triangleq \cos(\theta_j)$ and $s\theta_j \triangleq \sin(\theta_j)$. From (7) the screws for the joints are given by

$$e^{\hat{\xi}_j \theta_j} = \begin{bmatrix} c\theta_j & -s\theta_j & 0 & \left(\sum_{k=1}^{\frac{j-1}{2}} L_k\right)(1 - c\theta_j) \\ s\theta_j & c\theta_j & 0 & -\left(\sum_{k=1}^{\frac{j-1}{2}} L_k\right) s\theta_j \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{15}$$

for $j = 1, 3, \ldots$ and

$$e^{\hat{\xi}_j \theta_j} = \begin{bmatrix} c\theta_j & 0 & s\theta_j & \left(\sum_{k=1}^{\frac{j}{2}} L_k\right)(1 - c\theta_j) \\ 0 & 1 & 0 & 0 \\ -s\theta_j & 0 & c\theta_j & \left(\sum_{k=1}^{\frac{j}{2}} L_k\right) s\theta_j \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{16}$$

for $j = 2, 4, \ldots$.

III. Problem definition

We assume that we are given a desired parameterized 3-D curve $p_d(\cdot) : [0, L] \to \mathbb{R}^3$. We do not require that the parameterization is with respect to arclength. We assume further that the position of the first frame in the discrete curve is given by $p(0)$ and that the orientation of this frame can be specified freely. Because we want to approximate a shape rather than achieve a global position of a particular frame, this imposes no restrictions. The problem is to find the orientation of the first frame as well as that of the remaining joint angles such that the end points of each of the links lie on the original curve.

To solve this problem, we will split it into two subproblems. First, we seek the locations of the end points of each of the line segments.

Problem 1: Given a parameterized curve, $p_d(\cdot)$ and a sequence of line segments of lengths $L_1, \ldots, L_N$, find the values $0 = t_0 \leq t_1 \leq \cdots \leq t_N$ of the parameter such that $\|p_d(t_{j+1}) - p_d(t_j)\| = L_j$, $j = 1, \ldots, N$ where $\|\cdot\|$ denotes the standard Euclidean norm.

The second subproblem is to find the orientations of the line segments such that the frame origins lie at the locations given by the solution to Problem 1.

Problem 2: Given $t_0 \leq t_1 \leq \cdots \leq t_N$, find the orientation of the first frame and the relative orientations of the line segments such that the end points of the jth line segment are at $p(t_j)$ and $p(t_{j+1})$ for $j = 1, \ldots, N$.

IV. Problem solution

In this section we present a solution to Problems 1 and 2 and two examples to illustrate the approach.

A. Solving Problem 1

Problem 1 reduces to a sequence of 1-D searches along the desired curve. Because we assume only that the curve is continuous, the Euclidean distance between two points $p_d(t_j)$ and $p_d(t)$, $t > t_j$ is in general a nonlinear function of t; for each j, we seek the *first* t such that $\|p_d(t) - p_d(t_j)\| = L_j$. That is, we have

$$t_{j+1} = \arg\min_{t > t_j} (t - t_j) \quad \text{s. t.} \quad \|p_d(t) - p_d(t_j)\| = L_j \tag{17}$$

If t_{j+1} can be bracketed, then it can be found using a bisection search. To bracket the solution, we find t_+ such that $t_j \leq t_{j+1} \leq t_+$ by using the following algorithm.

Algorithm 4.1: Bracketing t_{j+1}

0. Initialize: Choose dt small enough such that there is at most one $t \in [t_j, t_j + dt]$ such that $\|p(t) - p(t_j)\| = L_j$ and set $t = t_j + dt$.
1. Check termination: If $\|p_d(t) - p_d(t_j)\| \geq L_j$ then set $t_+ = t$ and terminate.
2. Iterate: Set $t = t + dt$ and go to 1.

The appropriate value of dt depends on the curve. In practice it is typically enough to take dt much smaller then L_j. If the curve is given in the arclength parameterization then dt can be set equal to the segment length.

With t_{j+1} bracketed in this way, a numerical bisection search can be used to find the solution to within a desired accuracy.

Algorithm 4.2: Bisection

0. Initialize: Set $t_- = t_j$, $t_{\text{mid}} = \frac{t_+ - t_-}{2}$, and choose $\epsilon > 0$.
1. Check termination: If $(\|p_d(t_{j+1}) - p_d(t_j)\| - L_j)^2 \leq \epsilon$ then set $t_{j+1} = t_{\text{mid}}$ and terminate.
2. Shift end point: If $\|p_d(t_{\text{mid}}) - p_d(t_j)\| < L$ then set $t_- = t_{\text{mid}}$, else set $t_+ = t_{\text{mid}}$.
3. Iterate: Set $t_{\text{mid}} = \frac{t_+ - t_-}{2}$ and go to 1.

1) Bézier curve example : As a first example, consider the Bézier curve

$$p_d(t) = \sum_{k=0}^{6} p_{d_k} \frac{6!}{k!(6-k)!} t^k (1-t)^{N-k}, \quad 0 \leq t \leq 1 \quad (18)$$

with the seven control points p_{d_k} given by

$$\begin{bmatrix} 4 \\ 4 \\ 0 \end{bmatrix}, \begin{bmatrix} 4 \\ -4 \\ 0 \end{bmatrix}, \begin{bmatrix} -4 \\ -4 \\ 0 \end{bmatrix}, \begin{bmatrix} -4 \\ 4 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}, \begin{bmatrix} 4 \\ -4 \\ 6 \end{bmatrix}, \begin{bmatrix} 4 \\ 4 \\ 8 \end{bmatrix}.$$

To approximate this curve, we choose a piecewise linear curve with 15 segments. Each segment length is $L_j = 1.25$ units. The desired Bézier curve and the resulting points $p(t_j)$, $j = 0, \ldots, 15$ are shown in Fig. 2.

2) Helical curve example : As a second example, consider a circular helix in the arclength parameterization:

$$p_d(s) = \begin{bmatrix} R\sin(s\alpha) \\ -R\cos(s\alpha) \\ s\cos(\theta) \end{bmatrix} \quad (19)$$

where R is the radius, θ is the helical angle (the complement of the pitch angle), and $\alpha = \frac{\sin(\theta)}{R}$. We set $R = 4$ and $\theta = 1.4$ rad and choose to approximate the helix with a piecewise linear curve with 15 segments, each of length 1.25 units. The helical curve and the resulting points are shown in Fig. 3.

B. Solving Problem 2

We now seek the orientation of the first frame and the set of joint angles (the orientations of each of the line segments with respect to the previous segment) such that the end points of the line segments lie at the points found by the solution to Prob. 1, $p_d(t_0), p_d(t_1), \ldots, p_d(t_N)$.

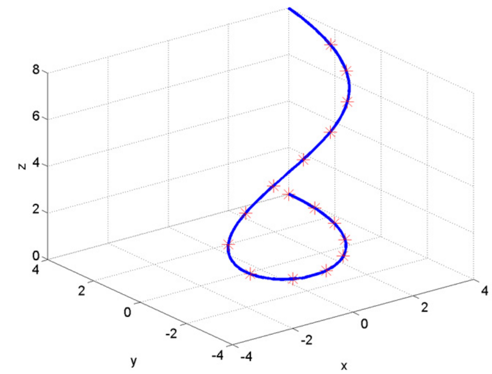

Fig. 2. An example of finding the set of discrete points along the continuous curve. The desired curve is a Bézier curve with 7 control points (given after (18)). The algorithm described in this section was used to find the 16 points (indicated on the curve by *) along the curve such that each adjacent pair is a distance of 1.25 units apart.

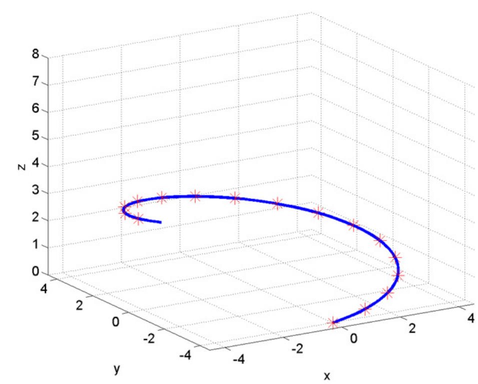

Fig. 3. In this example the desired curve is a circular helix of radius 1.25 and helical angle 1.4 rad. The curve was approximated with a 15 segment piecewise linear curve.

The transformation from the world frame to the first frame, $g_{w,1}$, can be viewed as a sequence of (fixed) joints: a translation of length $\|p_d(t_0)\|$ along the line from the origin to $p_d(t_0)$, a rotation θ_{z_0} about the $z-$axis of the first frame, a rotation θ_{y_0} about the $y-$axis of the first frame, and finally a rotation θ_{x_0} about the $x-$axis of the first frame. We assume that when each of these "joints" is at the zero position then the world frame and the first frame are aligned. Thus

$$g_{w,1} = e^{\hat{\xi}_{p_d(t_0)} \|p_d(t_0)\|} e^{\hat{\xi}_{z_0} \theta_{z_0}} e^{\hat{\xi}_{y_0} \theta_{y_0}} e^{\hat{\xi}_{x_0} \theta_{x_0}}. \quad (20)$$

As described in Sec. II, the point $p_d(t_1)$ lies L_1 units along the $x-$axis of the first frame. Using homogeneous coordinates, we have

$$\begin{bmatrix} p_d(t_1) \\ 1 \end{bmatrix} = g_{w,1} \begin{bmatrix} L_1 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (21)$$

Using (20) in (21), pre-multiplying both sides by $e^{-\hat{\xi}_{p_d(t_0)} \|p_d(t_0)\|}$ and recognizing that a rotation about the $x-$axis leaves $[L_1,0,0]^T$ invariant yields

$$e^{-\hat{\xi}_{p_d(t_0)}\|p_d(t_0)\|}\begin{bmatrix} p_d(t_1) \\ 1 \end{bmatrix} = e^{\hat{\xi}_{z_0}\theta_{z_0}} e^{\hat{\xi}_{y_0}\theta_{y_0}} \begin{bmatrix} L_1 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (22)$$

Using the form of a screw given by (7) and (8), (22) becomes

$$\begin{bmatrix} p_d(t_1) - p_d(t(0)) \\ 1 \end{bmatrix} = \begin{bmatrix} L_1 c\theta_{z_0} c\theta_{y_0} \\ L_1 s\theta_{z_0} c\theta_{y_0} \\ -L_1 s\theta_{y_0} \\ 1 \end{bmatrix} \quad (23)$$

Define $p_{i,j} \stackrel{\triangle}{=} p_d(t_i) - p_d(t_j)$. Then from (23) we have

$$\theta_{z_0} = \arctan 2\left([p_{1,0}]_2, [p_{1,0}]_1\right) \quad (24)$$

$$\theta_{y_0} = \arctan 2\left(-[p_{1,0}]_3, \left(([p_{1,0}]_1)^2 + ([p_{1,0}]_2)^2\right)^{\frac{1}{2}}\right) \quad (25)$$

where $[p]_i$ means the ith component of p. Thus the points $p_d(t_0)$ and $p_d(t_1)$ determine $g_{w,1}$ up to an arbitrary rotation θ_{x_0} about the $x-$axis of the first frame.

Under this construction, the $x-$axis of the first frame is aligned to point to $p_d(t_1)$. By the definition of the joint angles in Sec. II, the first joint angle is a rotation about the $z-$axis of the first frame. Therefore we have $\theta_1 = 0$. To determine the remaining joint angles as a function of θ_{x_0} we proceed as follows. The point $p_d(t_j)$ is located a distance L_j along the $x-$axis of frame $2j-1$. Thus

$$\begin{bmatrix} p_d(t_j) \\ 1 \end{bmatrix} = g_{w,2j-1} \begin{bmatrix} L_j \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (26)$$

The mapping $g_{w,2j-1}$ depends on the joint angles θ_1 through θ_{2j-1}. We will proceed in an iterative fashion and assume the first $2j-3$ joint angles are known. To isolate the values of the unknown joint angles, we pre-multiply (26) by $g^{-1}_{w,2j-3}$.

$$g^{-1}_{w,2j-3}\begin{bmatrix} p_d(t_j) \\ 1 \end{bmatrix} = g^{-1}_{w,2j-3} g_{w,2j-1} \begin{bmatrix} L_j \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (27)$$

Using (9) we calculate

$$\begin{aligned} & g^{-1}_{w,2j-3} g_{w,2j-1} \quad (28) \\ &= \left(g_{1,2j-3}(0)\right)^{-1} e^{\hat{\xi}_{2j-2}\theta_{2j-2}} e^{\hat{\xi}_{2j-1}\theta_{2j-1}} g_{1,2j-1}(0) \\ &= \begin{bmatrix} c\theta_{2j-2}c\theta_{2j-1} & -c\theta_{2j-2}s\theta_{2j-1} & s\theta_{2j-2} & L_{j-1} \\ s\theta_{2j-1} & c\theta_{2j-1} & 0 & 0 \\ -s\theta_{2j-2}c\theta_{2j-1} & s\theta_{2j-2}s\theta_{2j-1} & c\theta_{2j-2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} (29) \end{aligned}$$

where in the last step we have used (11). To calculate the left-hand side of (27), we first recall from (11) that the origin of frame $2j-3$ is at $p_d(t_{j-2})$. We then use (1) to write

$$g^{-1}_{w,2j-3} = \begin{bmatrix} R^{-1}_{w,2j-3} & -R^{-1}_{w,2j-3} p_d(t_{j-2}) \\ 0 & 1 \end{bmatrix}. \quad (30)$$

By definition, $R^{-1}_{w,2j-3}$ rotates the world frame into the frame $2j-3$. Using this, we can express the point $p_d(t_{j-2})$ in terms of the subsequent point $p_d(t_{j-1})$ and write (30) as

$$g^{-1}_{w,2j-3} = \begin{bmatrix} R^{-1}_{w,2j-3} & -R^{-1}_{w,2j-3} p_d(t_{j-1}) - \begin{bmatrix} L_{j-1} \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix}. \quad (31)$$

Inserting (29) and (31) in (27) we find

$$\begin{bmatrix} R^{-1}_{w,2j-3}\left(p_d(t_j) - p_d(t_{j-1})\right) \\ 1 \end{bmatrix} = \begin{bmatrix} L_j c\theta_{2j-2} c\theta_{2j-1} \\ L_j s\theta_{2j-1} \\ -L_j s\theta_{2j-2} c\theta_{2j-1} \\ 1 \end{bmatrix}. \quad (32)$$

If the joint angles θ_1 through θ_{2j-3} are known then the left hand side of (32) is known. Define $\Delta p_j \stackrel{\triangle}{=} R^{-1}_{w,2j-3}\left(p_d(t_j) - p_d(t_{j-1})\right)$. Then

$$\theta_{2j-2} = \arctan 2\left(-[\Delta p_j]_3, [\Delta p_j]_1\right), \quad (33)$$

$$\theta_{2j-1} = \arctan 2\left([\Delta p_j]_2, \left([\Delta p_j]_1^2 + [\Delta p_j]_3^2\right)^{\frac{1}{2}}\right). \quad (34)$$

Given a value for θ_{x_0}, the joint angles are determined by first finding the global orientation $\theta_{z_0}, \theta_{y_0}$ from (24) and (25). The joint angles are then solved iteratively using (33) and (34) for $j = 1, \ldots, N$ (with the exception that $\theta_1 = 0$ as discussed above). Note that the next rotation matrix $R^{-1}_{w,2j-1}$ can be found without doing a matrix inversion using

$$\begin{aligned} R^{-1}_{w,2j-1} &= R_{2j-1,w} \\ &= R_{2j-1,2j-2} R_{2j-2,2j-3} R_{2j-3,w} \\ &= e^{-\hat{\omega}_{2j-1}\theta_{2j-1}} e^{-\hat{\omega}_{2j-2}\theta_{2j-2}} R^{-1}_{w,2j-3} \quad (35) \end{aligned}$$

where ω_j is the axis of rotation for joint j.

The angle θ_{x_0} can be chosen arbitrarily or used to optimize some cost function. For, it may be desired to keep the joint angles as close to zero as possible. Then θ_{x_0} is given by the solution to

$$\min_{\theta_{x_0}} \sum_{i=1}^{2N-2} \theta_i^2 \quad \text{subj. to} \quad (24), (25), (33), \text{ and } (34). \quad (36)$$

1) Bézier curve example: Consider again the Bézier curve given in (18) and shown in Fig. 2. The end points for a 15 segment piecewise linear fitting curve were found in Sec. IV-A.1. Using the algorithm in this section and the minimization criterion in (36) to determine θ_{x_0}, the joint angles between the line segments were found. The resulting curve and the frames are shown in Fig. 4. The actual values of the angles are given in Table I.

To illustrate the dependence of the joint angles (and thus the orientations of the frames), we set $\theta_{x_0} = -\frac{\pi}{4}$ rad and ran the algorithm again. The curve and the frames are shown in Fig. 5.

2) Helical curve example: Consider the circular helix given in Sec. IV-A.2 and shown in Fig. 3. The angle θ_{x_0} was set to $\frac{-\pi}{4}$ and the remaining angles found using the algorithm in this section. The resulting frame orientations are shown in Fig. 6.

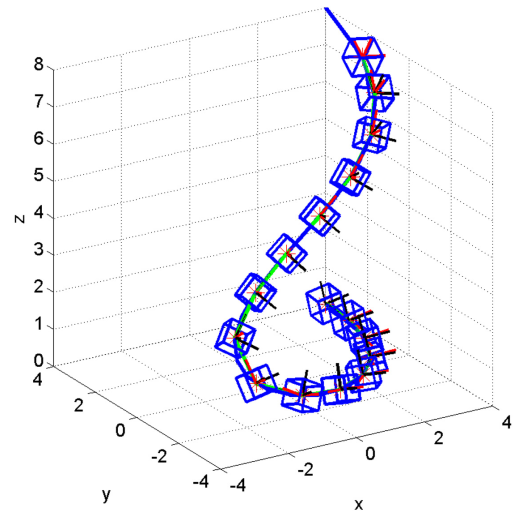

Fig. 4. A 15-segment piecewise linear curve in the shape of the Bézier curve (18). The arbitrary rotation around the $x-$axis of the first frame was chosen by minimizing the sum of the squares of all of the joint angles.

θ_{x_0}	0.218	θ_{y_0}	0	θ_{z_0}	-1.64
θ_1	0	θ_2	-0.035	θ_3	-0.156
θ_4	-0.048	θ_5	-0.196	θ_6	-0.070
θ_7	-0.251	θ_8	-0.113	θ_9	-0.317
θ_{10}	-0.198	θ_{11}	-0.367	θ_{12}	-0.394
θ_{13}	-0.347	θ_{14}	-0.934	θ_{15}	-0.074
θ_{16}	-0.834	θ_{17}	0.248	θ_{18}	-0.234
θ_{19}	0.094	θ_{20}	-0.056	θ_{21}	-0.003
θ_{22}	0.039	θ_{23}	-0.095	θ_{24}	0.156
θ_{25}	-0.232	θ_{26}	0.272	θ_{27}	-0.344
θ_{28}	0.236	θ_{29}	-0.263	θ_{30}	0.138

TABLE I

JOINT ANGLES TO APPROXIMATE THE BÉZIER CURVE IN (18) USING A 15 SEGMENT CURVE WITH θ_{x_0} FOUND FROM (36). ANGLES ARE IN RADIANS.

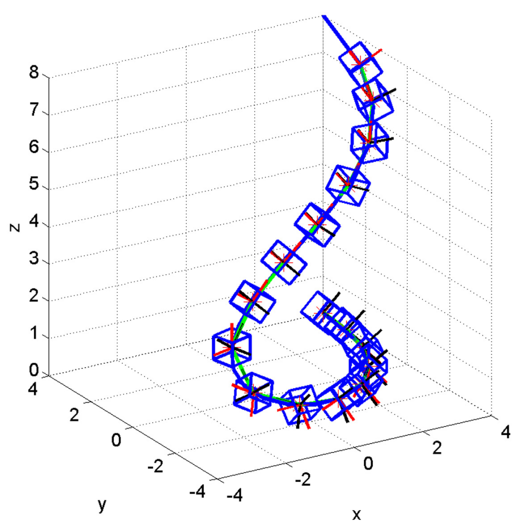

Fig. 5. A 15-segment piecewise linear curve in the shape of the Bézier curve (18). The angle θ_{x_0} rad was chosen to be $-\frac{\pi}{4}$.

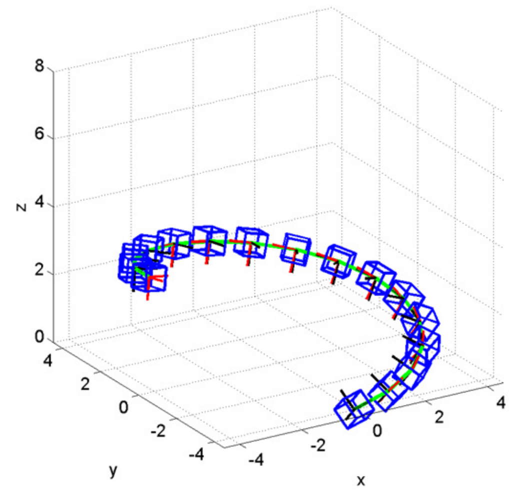

Fig. 6. A 15-segment piecewise linear curve in the shape of a circular helix with $R = 4$ and helical angle 1.4 rad. The angle θ_{x_0} was chosen to be $-\frac{\pi}{4}$.

V. CONCLUSIONS

In this paper we have presented an algorithm to approximate a parameterized 3-D curve by a piecewise linear one whose line segments are connected by universal joints. The algorithm uses a numerical search to find the locations of the end points of the line segments. Given these points, the joint angles are found analytically in terms of a single S^1 symmetry given by a rotation about the first line segment. This degree of freedom can be used to optimize a secondary task. The tool developed here can be used to generate joint angles for a snake-like robot to ensure the body of the robot takes on a desired shape.

ACKNOWLEDGMENT

This work was supported in part by DARPA under a Charles S. Draper Laboratory prime DL-H-551034. The author would like to thank R. Brockett of Harvard University for helpful discussions regarding this material. Thanks are also due to P. DeBitetto of Draper Labs for helpful suggestions.

REFERENCES

[1] A. Stokes and R. Brockett, "Dynamics of kinematic chains," *Int. J. Robot. Res.*, vol. 15, no. 4, pp. 393–405, 1996.

[2] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[3] R. Brockett, "Pattern generation and the control of nonlinear systems," *IEEE Trans. Automat. Contr.*, vol. 48, no. 10, pp. 1699–1711, 2003.

[4] G. Chirikjian and J. Burdick, "The kinematics of hyper-redundant robot locomotion," *IEEE Trans. Robot. Automat.*, vol. 11, no. 6, pp. 781–793, 1995.

[5] J. Cortés, S. Martinéz, J. Ostrowski, and K. McIsaac, "Optimal gaits for dynamic robot locomotion," *Int. J. Robot. Res.*, vol. 20, no. 9, pp. 707–728, 2001.

[6] K. McIsaac and J. Ostrowski, "A framework for steering dynamic robotic locomotion systems," *Int. J. Robot. Res.*, vol. 22, no. 2, pp. 83–97, 2003.

[7] D. Tsakiris, M. Sfakiotakis, A. Menciassi, G. LaSpina, and P. Dario, "Polychaete-like undulatory robotic locomotion," in *Proc. IEEE Conference on Robotics and Automation*, 2005, pp. 3029–3034.

[8] G. Chirikjian and J. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Trans. Robot. Automat.*, vol. 10, no. 3, pp. 343–354, 1994.