the PA10-6CE utilizing the geometric and flexibility calibration method described here. We anticipate the combination of a detailed physical model, and its accompanying control design will provide an accurate measurement and control system suitable for demanding dynamic applications with the PA10-6CE robot arm.

REFERENCES

[1] T. D. Tuttle, "Understanding and modeling the behavior of a harmonic drive gear transmission,". (1992). *MIT Artif. Intell. Lab.,* Cambridge, MA, Tech. Rep. 1365, 1992, 1992.
[2] N. Kircanski and A. A. Goldenberg, "An experimental study of nonlinear stiffness, hysteresis, and friction effects in robot joints with harmonic drives and torque sensors," *Int. J. Robot. Res.*, vol. 16, no. 2, pp. 214–239, 1997.
[3] R. P. Judd and A. B. Knasinski, "A technique to calibrate industrial robots with experimental verification," *IEEE Trans. Robot. Autom.*, vol. 6, no. 1, pp. 20–30, Feb. 1990.
[4] H. D. Taghirad, P. R. Belanger, and A. Helmy, "An experimental study on harmonic drives,". (1996). *International Submarine Engineering Ltd., McGill Univ. Center for Intell. Mach, Tech. Rep., 1996.* 1996.
[5] C. W. Kennedy and J. P. Desai, "Modeling and Control of the Mitsubishi PA10 Robot Arm Harmonic Drive System," *IEEE/ASME Trans. Mechatronics*, vol. 10, no. 3, pp. 263–274, Jun. 2005.
[6] J. H. Jang, S. H. Kim, and Y. K. Kwak, "Calibration of geometric and non-geometric errors of an industrial robot," *Robotica*, vol. 19, pp. 311–321, 2001.
[7] M. A. Meggiolaro, S. Dubowsky, and C. Mavroidis, "Geometric and elastic error calibration of a high accuracy patient positioning system," *Mechanism Mach. Theory*, vol. 40, pp. 415–427, 2005.
[8] W. Khalil and S. Besnard, "Geometric calibration of robots with flexible joints and links," *J. Intell. Robotic Syst.*, vol. 34, pp. 357–379, 2002.
[9] R. T. Haftka and Z. Gurdal, *Elements of Structural Optimization*, 3rd ed. Dordrecht, The Netherlands: Kluwer Academic, 1992, pp. 387–414.
[10] C. D. Crane III and J. Duffy, *Kinematic Analysis of Robot Manipulators*. Cambridge, U.K.: Cambridge Univ. Press, 1998, ch. 2.
[11] S. A. Hayati, "Robot arm geometric link calibration," in *Proc. 22nd IEEE Conf. Decis. Control*, 1983, pp. 1477–1483.
[12] *Instruction Manual for Installation, Maintenance, and Safety*, Mitsubishi Heavy Industries, Tokyo, Japan, 2003.
[13] B. W. Mooring, Z. S. Roth, and M. R. Driels, *Fundamentals of Manipulator Calibration*. Hoboken, NJ: Wiley, 1991.
[14] Variable Resolution, *Monolithic Resolver-to-Digital Converters*. Norwood, MA: Analog Devices, 1998.
[15] C. Mavroidis, J. Flanz, S. Dubowsky, P. Drouet, and M. Goitein, "High performance medical robot requirements and accuracy analysis," *Robot. Comput. Integr. Manuf.*, vol. 14, no. 5–6, pp. 329–338, 1998.
[16] R. D. Howe and Y. Matsuoka, "Robotics for surgery," *Annu. Rev. Biomed. Eng.*, vol. 1, pp. 211–240, 1999.

Discretization of a Continuous Curve

Sean B. Andersson

Abstract—We consider the problem of approximating a finite-length continuous curve by a piecewise linear one whose segments are assumed to be connected by 2 DOF joints. We solve the problem under the assumption that the endpoints of the line segments lie on the continuous curve. Analytical expressions for the relative orientations of each pair of line segments as a function of a single rotational DOF are found. This angle can be chosen arbitrarily or used to optimize a secondary task. The motivating application for this paper is the control of a snake-like robot using gaits designed from shape primitives.

Index Terms—Kinematics, snake-like robot.

I. INTRODUCTION

In this paper, we consider the problem of approximating a continuous, finite-length 3-D curve with a piecewise linear one. The discrete curve consists of a sequence of blocks, each of which is connected to a line segment at its center by a revolute joint. Each internal block is connected to two line segments, and we assume that the joint axes are orthogonal, yielding a 2 DOF joint. We simplify the problem by assuming that the center of each block lies on the continuous curve. The location of these points along the curve to be approximated are found numerically using a bisection search method. Given these locations, we derive expressions for the joint angles as a function of a single DOF, namely a rotation of the first segment around the axis defined by its endpoints. This rotation can be either chosen arbitrarily or used to optimize some secondary task. The algorithm for the calculation of the joint angles is $O(n)$ in complexity where n is the number of line segments in the discrete curve.

The motivation for this research lies in the control of snake-like (or hyperredundant) mobile robots. This class of robot, pioneered by the active-cord mechanism of Hirose [1], provides a flexibility that is difficult or impossible to achieve with other locomotion modalities. The algorithm presented here can also be used for the control of a hyperredundant manipulator. These manipulators have the capability to perform unconventional tasks such as pipeline inspection, minimally invasive surgery [2], and whole-arm manipulation [3]. The very flexibility of such systems make them difficult to control. Many redundancy resolution schemes utilize some form of inverse of the Jacobian, often coupled with numerical optimization [4]–[6]. However, due to the very large number of DOFs for hyperredundant robots, the computational burden of these techniques is generally prohibitive.

An alternative approach is to perform path planning for the robot using a continuous-curve approximation. This is coupled with a fitting process for determining the joint angles such that the robot takes on the desired shape [7]–[11] or by determining the sequence of joint angles such that the robot extends outward along the path [12], [13]. In many cases, the fitting is achieved through numerical minimization of a distance function between the continuous curve and the robot. In [14], the authors avoid the need to directly solve the inverse kinematics

problem by deriving a control law for the joints such that the robot converges to the desired shape.

The complexity of control is exacerbated in the mobile robot setting since locomotion is achieved through interaction with the environment. One general approach useful in this setting is the notion of gait-based control [15], [16]. This technique has been applied successfully to many novel locomotion systems, including snake-like [8], [17]–[19], eel-like [20], and polychaete annelid-like robots [21]. One can view a gait as a sequence of shapes that the mechanism must take on. In many cases, this sequence is driven by the unknown and possibly changing environment, and therefore, cannot be known ahead of time. However, there are scenarios in which the sequence can be designed *a priori*. These include ascending or descending a stairway, climbing a pole, or maneuvering along a predetermined path using a pulse-like gait (see [7]).

The algorithm presented in this paper assumes that a desired shape is given, specified as a 3-D curve, and solves the inverse kinematics problem to achieve that shape. Because of this assumption, it is most applicable to the scenarios described before. In addition, due to its computational efficiency (as demonstrated in Fig. 3), it is particularly useful for a simulation environment. The results detailed here are for serial-type robots comprising links coupled by 2 DOF joints, and it is hoped that the general approach can be extended to other designs, such as serial robots with 3 DOF joints [22], serial mechanisms including prismatic joints [23], and continuum robots comprising sections of constant curvature [24].

II. PRODUCT OF EXPONENTIALS FORMULATION

We give here a brief description of the product of exponentials (POE) representation for kinematic chains [25]. The POE approach was chosen over the Denavit–Hartenberg representation because its compact form, geometric significance of the twists, and continuous nature make the kinematic equations relatively simple to manipulate. For a detailed discussion of the merits of the POE approach, see [26].

A serial kinematic chain is a series of rigid links connected by single DOF joints. Joints with more than one DOF are modeled by allowing the distance between subsequent joints to be zero. A frame is attached to each joint, and the position and orientation of each of these frames is specified with respect to a world frame. The configuration of the jth frame with respect to the world frame is denoted $g_{w,j}$ with

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

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. Similarly, $g_{i,j}$ denotes the configuration of the jth frame with respect to the ith frame. These mappings can be concatenated, for example, $g_{w,j-1}g_{j-1,j} = g_{w,j}$.

Because subsequent links are connected by a single DOF, 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 referred to as the *joint angle*. To represent this one-parameter family, recall 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}.

A twist and a screw can be expressed using the hat map, a notation used here in two different ways. First, it denotes a mapping from $\mathbb{R}^3$ into

the space of 3×3 skew-symmetric matrices [the Lie algebra so(3)]:

$$\widehat{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^{\widehat{}} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \qquad (2)$$

Second, the hat map denotes a mapping from $\mathbb{R}^6$ into the Lie algebra se(3). With $v, \omega \in \mathbb{R}^3$, we have

$$\begin{bmatrix} v \\ \omega \end{bmatrix}^{\widehat{}} = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix}. \qquad (3)$$

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

$$\xi_j = \begin{bmatrix} -\omega_j \times q_j \\ \omega_j \end{bmatrix} \qquad (4)$$

where ω_j is a unit vector along the axis of rotation and q_j is any point on that axis. For a prismatic joint, the twist is $\xi_j = [v_j^T \ 0]^T$ with v_j being a unit vector along the axis of translation.

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

$$e^{\widehat{\xi_j}\theta_j} = \begin{bmatrix} e^{\widehat{\omega_j}\theta_j} & (\mathbb{I} - e^{\widehat{\omega_j}\theta_j})(\omega_j \times v_j) \\ 0 & 1 \end{bmatrix} \qquad (5)$$

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

$$e^{\widehat{\xi_j}\theta_j} = \begin{bmatrix} \mathbb{I} & v_j\theta_j \\ 0 & 1 \end{bmatrix}. \qquad (6)$$

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

$$g_{w,j}(\theta) = e^{\widehat{\xi_1}\theta_1} e^{\widehat{\xi_2}\theta_2} \ldots e^{\widehat{\xi_j}\theta_j} g_{w,j}(0). \qquad (7)$$

III. KINEMATICS

In this paper, we assume that the discrete curve consists of a sequence of blocks connected by line segments. Each block is connected at its center to the endpoint of a line segment by a single DOF revolute joint. Internal blocks are connected to two segments along orthogonal axes. We attach to each joint a frame with origin at the center of the corresponding block. The local x-axis is defined to point toward the center of the subsequent block and the y- and z-axis are defined by the axes of rotation for the joints. This is illustrated in Fig. 1 in which two links and three joints are shown. Joints are labeled from 1 to N. The link lengths L_i, $i = 1, \ldots, N-1$, are defined to be the distances between subsequent joint centers with even numbered links having length zero. Each odd (even)-numbered joint rotates about the local z- (y-)axis.

To describe the internal *shape* of the discrete curve, we specify the configuration of all the frames with respect to the first frame, using (7) where the world frame is replaced with the first frame. The transformation from the first frame to the world may be viewed as a (fixed) joint with associated transformation $g_{w,1}$. Thus,

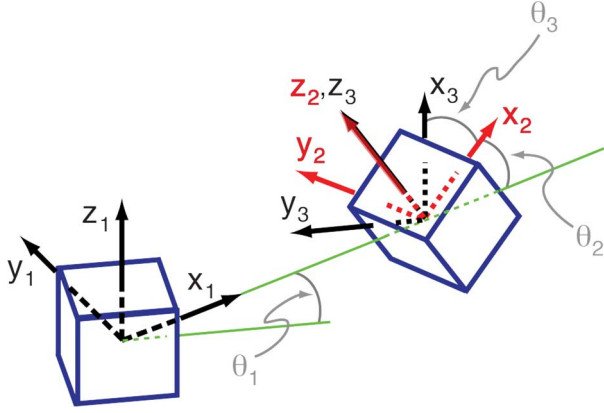$$g_{w,j}(\theta) = g_{w,1} e^{\hat{\xi_1}\theta_1} \ldots e^{\hat{\xi_j}\theta_j} g_{1,j}(0). \qquad (8)$$

Fig. 1. First two links and three joints of a kinematic chain. The first joint rotates by angle θ_1 around the z_1-axis, the second by θ_2 around the y_2-axis, the third by θ_3 around the z_3-axis, and so on.

Since subsequent links lie along the local x-axis, we have

$$
g_{1,j}(0) = \begin{bmatrix} 1 & 0 & 0 & \sum_{k=1}^{j-1} L_k \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}
$$

where it is understood that the sum is 0 when $j = 1$. Each joint is a revolute joint with the axis of rotation given by

$$
\omega_j = \begin{cases} (0,0,1)^T, & j = 1, 3, \dots \\ (0,1,0)^T, & j = 2, 4, \dots. \end{cases} \tag{10}
$$

To express the twist of the jth joint as in (4), we choose q_j to be the origin of frame j (in the reference configuration), i.e.,

$$
q_j = \left(\sum_{k=1}^{j-1} L_k, 0, 0 \right)^T. \tag{11}
$$

The twist of the jth joint is then

$$
\xi_j = \begin{cases} \left(0, -\sum_{k=1}^{j-1} L_k, 0, 0, 0, 1 \right)^T, & j = 1, 3, \dots \\ \left(0, 0, \sum_{k=1}^{j-1} L_k, 0, 1, 0 \right)^T, & j = 2, 4, \dots. \end{cases} \tag{12}
$$

From (5), the screws for the joints are given by

$$
e^{\hat{\xi}_j \theta_j} = \begin{bmatrix} c\theta_j & -s\theta_j & 0 & \left(\sum_{k=1}^{j-1} L_k\right)(1 - c\theta_j) \\ s\theta_j & c\theta_j & 0 & -\left(\sum_{k=1}^{j-1} L_k\right)s\theta_j \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}
$$

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

$$
e^{\hat{\xi}_j \theta_j} = \begin{bmatrix} c\theta_j & 0 & s\theta_j & \left(\sum_{k=1}^{j-1} L_k\right)(1 - c\theta_j) \\ 0 & 1 & 0 & 0 \\ -s\theta_j & 0 & c\theta_j & \left(\sum_{k=1}^{j-1} L_k\right)s\theta_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}
$$

for $j = 2, 4, \dots$. Note that, in (13) and (14), we have introduced the shorthand notation $c\theta_j \overset{\triangle}{=} \cos(\theta_j)$ and $s\theta_j \overset{\triangle}{=} \sin(\theta_j)$.

IV. APPROXIMATING A CURVE

Let $p_d(\cdot) : [0, L] \to \mathbb{R}^3$ be a continuous 3-D curve. We assume that the position of the first frame in the discrete curve is given by $p_d(0)$ and that the orientation of this frame can be specified freely. We define the shape approximation problem via the following two subproblems.

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

Subproblem 2: Given the t_j solving subproblem 1, find the orientation of the first frame and the relative orientations of the line segments such that the endpoints of the jth line segment are at $p_d(t_j)$ and $p_d(t_{j+1})$ for $j = 1, \dots, N$.

In what follows we present an analytical solution of the second subproblem that is $O(n)$ in complexity and a numerical solution of the first subproblem relying on a bisection search.

A. Solving Subproblem 1

Subproblem 1 reduces to a sequence of 1-D searches along the desired curve. 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 $\|p_d(t) - p_d(t_j)\| = L_j$. 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 $t_j \leq t_{j+1} \leq t_+$ as follows.

1) *Initialize:* Choose dt such that there is at most one $t \in [t_j, t_j + dt]$ with $\|p_d(t) - p_d(t_j)\| = L_j$. Set $t = t_j + dt$.
2) *Termination:* If $\|p_d(t) - p_d(t_j)\| \geq L_j$, then terminate.
3) *Iterate:* Set $t = t + dt$ and go to step 2).

The appropriate value of dt depends on the curve and parametrization. In practice, one can take dt much smaller than L_j. With t_{j+1} bracketed, a numerical bisection search can be used to find its value to within a desired accuracy.

B. Solving Subproblem 2

To solve for the joint angles, we first solve for the configuration of the initial joint with respect to the world frame, revealing in the process that a single DOF remains that can be chosen arbitrarily or used to optimize a secondary task (see Theorem 4.1). Given this initial configuration, we then show that the joint angles can be expressed analytically. The solution is a set of recursive inverse kinematic equations that depend on the arbitrary DOF (see Theorem 4.2).

1) Configuration of the First Frame: The transformation $g_{w,1}$ from the world frame to the first frame 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,

$$
g_{w,1} = e^{\hat{\xi}_{p_d(t_0)} \|p_d(t_0)\|} e^{\hat{\xi}_{z_0} \theta_{z_0}} e^{\hat{\xi}_{y_0} \theta_{y_0}} e^{\hat{\xi}_{x_0} \theta_{x_0}}. \tag{15}
$$

Theorem 4.1: Define $p_{i,j} \overset{\triangle}{=} p_d(t_i) - p_d(t_j)$. Then, the angle θ_{x_0} is arbitrary while the angles θ_{y_0} and θ_{z_0} are

$$
\theta_{z_0} = \operatorname{atan2}([p_{1,0}]_2, [p_{1,0}]_1) \tag{16a}
$$

$$
\theta_{y_0} = \operatorname{atan2}\left(-[p_{1,0}]_3, \left(([p_{1,0}]_1)^2 + ([p_{1,0}]_2)^2\right)^{1/2}\right) \tag{16b}
$$

where $[p]_i$ denotes the ith component of p.

Proof: As described in Section III, the point $p_d(t_1)$ lies L_1 units along the x-axis of the first frame. Thus,

$$\begin{bmatrix} p_d(t_1) \\ 1 \end{bmatrix} = g_{w,1} \begin{bmatrix} L_1 & 0 & 0 & 1 \end{bmatrix}^T.$$

Using (15) in this equation, premultiplying both sides by $e^{-\widehat{\xi}_{p_d(t_0)} \|p_d(t_0)\|}$ and recognizing that a rotation about the x-axis leaves $[L_1, 0, 0]^T$ invariant yields

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

Using the form of a screw given by (5) and (6), this becomes

$$\begin{bmatrix} p_d(t_1) - p_d(t_0) \\ 1 \end{bmatrix} = \begin{bmatrix} p_{1,0} \\ 1 \end{bmatrix} = \begin{bmatrix} L_1 c\theta_{z_0} c\theta_{y_0} \\ L_1 s\theta_{z_0} c\theta_{y_0} \\ -L_1 s\theta_{y_0} \\ 1 \end{bmatrix}$$

which yields the result. ∎

2) Determination of Joint Angles:

Theorem 4.2: Define $\Delta p_j \overset{\triangle}{=} R_{w,j-3}^{-1} (p_d(t_j) - p_d(t_{j-2}))$ and let $j \geq 4$ be even. Then, $\theta_1 = 0$ and the remaining joint angles are given by the recursive equations

$$\theta_{j-2} = \text{atan2}\left(-[\Delta p_j]_3, [\Delta p_j]_1\right), \tag{17a}$$

$$\theta_{j-1} = \text{atan2}\left([\Delta p_j]_2, \left([\Delta p_j]_1^2 + [\Delta p_j]_3^2\right)^{1/2}\right). \tag{17b}$$

Proof: By construction of the world orientation of the first frame, its x-axis is aligned to point $p_d(t_1)$. Therefore, $\theta_1 = 0$. Since j is even, the point $p_d(t_j)$ is L_{j-1} units along the x-axis of frame $j - 1$. Assume that the first $j - 3$ angles are known and write

$$g_{w,j-3}^{-1} \begin{bmatrix} p_d(t_j) \\ 1 \end{bmatrix} = g_{w,j-3}^{-1} g_{w,j-1} \begin{bmatrix} L_{j-1} & 0 & 0 & 1 \end{bmatrix}^T \tag{18}$$

where we have premultipled by $g_{w,j-3}^{-1}$ to isolate the unknown joint angles. From (7), we have

$$g_{w,j-3}^{-1} g_{w,j-1} = (g_{1,j-3}(0))^{-1} e^{\widehat{\xi}_{j-2} \theta_{j-2}} e^{\widehat{\xi}_{j-1} \theta_{j-1}} g_{1,j-1}(0).$$

To calculate the left-hand side of the equation, recall from (9) that the origin of frame $j - 3$ is at $p_d(t_{j-3})$. We then use (1) to write

$$g_{w,j-3}^{-1} = \begin{bmatrix} R_{w,j-3}^{-1} & -R_{w,j-3}^{-1} p_d(t_{j-3}) \\ 0 & 1 \end{bmatrix}.$$

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

$$g_{w,j-3}^{-1} = \begin{bmatrix} R_{w,j-3}^{-1} & -R_{w,j-3}^{-1} p_d(t_{j-2}) - \begin{bmatrix} L_{j-3} \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix}.$$

Using these expressions for $g_{w,j-3}^{-1} g_{w,j-1}$ and $g_{w,j-3}^{-1}$ in (18), we find

$$\begin{bmatrix} R_{w,j-3}^{-1} (p_d(t_j) - p_d(t_{j-2})) \\ 1 \end{bmatrix} = \begin{bmatrix} L_{j-1} c\theta_{j-2} c\theta_{j-1} \\ L_{j-1} s\theta_{j-1} \\ -L_{j-1} s\theta_{j-2} c\theta_{j-1} \\ 1 \end{bmatrix}.$$

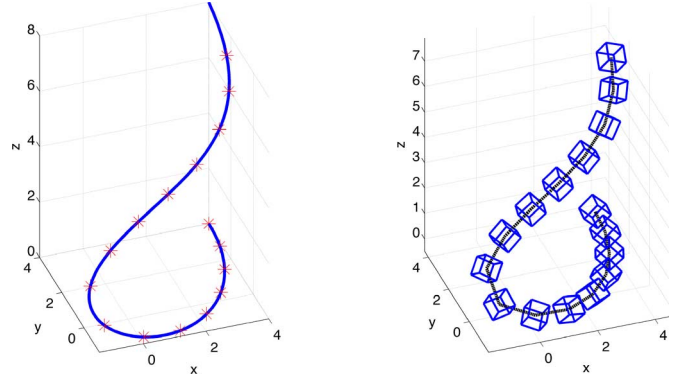Using the definition of Δp_j, this yields the stated result. ∎

Fig. 2. Thirty-joint piecewise linear curve in the shape of the Bézier curve (19). The joint frames have been omitted for clarity.

Note that the next rotation matrix $R_{w,j-1}^{-1}$ can be found without doing a matrix inversion using

$$R_{w,j-1}^{-1} = R_{j-1,j-2} R_{j-2,j-3} R_{j-3,w}$$

$$= e^{-\widehat{\omega}_{j-1} \theta_{j-1}} e^{-\widehat{\omega}_{j-2} \theta_{j-2}} R_{w,j-3}^{-1}$$

where ω_j is the axis of rotation for joint j. The rotation matrix $R_{w,j-3}^{-1}$ is known from the previous step.

C. Example

As an example, consider the Bézier curve

$$p_d(t) = \sum_{k=0}^{6} p_{d_k} \frac{6!}{k!(6-k)!} t^k (1-t)^{6-k}, \qquad 0 \leq t \leq 1 \tag{19}$$

with the seven control points p_{d_k}

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

We approximated this using a piecewise linear curve with 30 joints where each odd-numbered link had length 1.25 units and each even-numbered link had length 0.

The desired joint positions and the joint angles were determined using the algorithm described before. The angle θ_{x_0} was chosen arbitrarily to be $\pi/4$. The Bézier curve and the resulting points $p(t_j)$, $j = 0, \ldots, 29$, as well as the resulting shape of the discrete curve are shown in Fig. 2.

D. Performance

The algorithm consists of two stages: first, a fast bisection search to determine the positions of the joints, and second, a set of equations giving the joint angles. The computation required to find a pair of joint angles using (17) is independent of the total number of joints. Adding another pair of 1 DOF joints simply adds two more equations to be solved; therefore, the computational complexity of the analytical stage is $O(n)$.

To investigate the computational performance of the algorithm as a whole, the Bézier curve (19) was approximated using piecewise linear curves with 4–100 joints. All calculations were done in Matlab (v. 7.1) on an Apple Mac Pro running OS X (v. 10.4.11) with 3 GB of RAM. The results are shown in Fig. 3. The total run time with 100 joints was less than 35 ms with most of the computation time spent
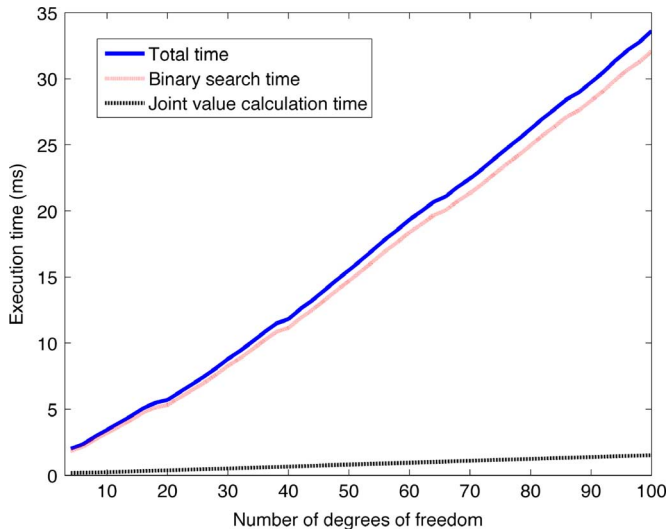
Fig. 3.   Execution time in Matlab of the algorithm for the Bézier curve. Even for a curve with 100 joints, the total execution time was less than 35 ms.

finding the desired locations of the joints. The joint value calculation time increases linearly with the number of joints, illustrating the $O(n)$ nature of the computation.

We note that the time to complete the numerical search could be reduced significantly if prior information about the approximate joint locations were known. For example, if the algorithms were used to generate joint angles as a function of time for a varying curve, the solution from the previous time step could be used to initialize the numerical search.

E. Comparison With Previous Methods

Existing fitting techniques for approximating a continuous curve with a discrete one rely on the numerical solution of a nonlinear problem. Initial efforts based on a least-squares optimization required the simultaneous solution of a system of nonlinear equation and was applicable only to planar robots with revolute joints and to a specific variable geometry truss [7]. This method was improved in [10] with a technique similar to the one presented here. Their algorithm also used a numerical search along the continuous curve for the locations of the joints followed by a set of recursive equations for the joint angles. However, the joint angles were implicit in the nonlinear equations, and thus, a numerical solution was required. By contrast, the algorithm presented here solves for the joint angles explicitly [cf., (17)] and makes explicit the single rotational DOF in the solution.

An entirely different approach to solving the curve fitting problem is developed in [14]. The authors consider a snake-like robot modeled by a second-order dynamic system and develop a nonlinear control law for the joint torques such that the joint angles converge to the desired values. This approach is well suited for shape control of hyperredundant mechanisms. However, to determine what the final joint angles will be requires numerical integration of the resulting equations. This will in general be significantly slower than the method described in this paper. In settings where one wishes to know the final configuration as quickly as possible, such as in simulations, user design of shapes, or preliminary design and verification of shape-based trajectories, the technique introduced here is to be preferred.

V. Conclusion

In this paper, we have considered the problem of approximating a continuous 3-D curve with a piecewise linear one. A bisection search together with a closed-form solution for the relative orientations between the segments of the discrete curve was derived. The method can be used to determine the joint angles for a snake-like robot so that it takes on a desired shape, and it is fast enough to be implemented in real time.

References

[1] S. Hirose, *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators*.   Oxford, U.K.: Oxford Science, 1993.
[2] A. B. Slatkin, J. W. Burdick, and W. Grundfest, "The development of a robotic endoscope," in *in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1995, pp. 162–171.
[3] J. K. Salisbury, "Whole arm manipulation," in *Proc. 4th Int. Symp. Robot. Res.*, 1987, pp. 183–189.
[4] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1985, pp. 722–728.
[5] H. Seraji, "Configuration control of redundant manipulators: Theory and implementation," *IEEE Trans. Robot. Autom.*, vol. 5, no. 4, pp. 472–490, Aug. 1989.
[6] S. Seereeram and J. T. Wen, "A global path planning approach for redundant manipulators," *IEEE Trans. Robot. Autom.*, vol. 11, no. 1, pp. 152–160, Feb. 1995.
[7] G. S. Chirikjian and J. W. Burdick, "A modal approach to hyperredundant manipulator kinematics," *IEEE Trans. Robot. Autom.*, vol. 10, no. 3, pp. 343–354, Jun. 1994.
[8] G. S. Chirikjian and J. W. Burdick, "The kinematics of hyper-redundant robot locomotion," *IEEE Trans. Robot. Autom.*, vol. 11, no. 6, pp. 781–793, Dec. 1995.
[9] H. Choset and W. Henning, "A follow-the-leader approach to serpentine robot motion planning," *ASCE J. Aerosp. Eng.*, vol. 12, no. 2, pp. 65–73, Apr. 1999.
[10] F. Fahimi, H. Ashrafiuon, and C. Nataraj, "An improved inverse kinematic and velocity solution for spatial hyper-redundant robots," *IEEE Trans. Robot. Autom.*, vol. 18, no. 1, pp. 103–107, Feb. 2002.
[11] S. Ma and M. Konno, "An obstacle avoidance scheme for hyperredundant manipulators-global motion planning in posture space," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1997, pp. 161–166.
[12] E. S. Conkur, "Path following algorithm for highly redundant manipulators," *Robot. Auton. Syst.*, vol. 45, no. 1, pp. 1–22, Oct. 2003.
[13] R. J. Schilling, R. Read, V. Lovass-Nagy, and G. Walker, "Path tracking with the links of a planar hyper-redundant manipulator," *J. Robot. Syst.*, vol. 12, no. 3, pp. 189–197, Mar. 1995.
[14] H. Mochiyama, E. Shimemura, and H. Kobayashi, "Shape control of manipulators with hyper degrees of freedom," *Int. J. Robot. Res.*, vol. 18, no. 6, pp. 584–600, Jun. 1999.
[15] J. P. Ostrowski and J. W. Burdick, "The geometric mechanics of undulatory robotic locomotion," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 683–701, Jul. 1998.
[16] R. W. Brockett, "Pattern generation and the control of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 48, no. 10, pp. 1699–1711, Oct. 2003.
[17] S. Ma, "Analysis of creeping locomotion of snake-like robot," *Adv. Robot.*, vol. 15, no. 2, pp. 205–224, Jun. 2001.
[18] M. Saito, M. Fukaya, and T. Iwasaki, "Serpentine locomotion with robotic snakes," *IEEE Control Syst. Mag.*, vol. 22, no. 1, pp. 64–81, Feb. 2002.
[19] G. Hicks and K. Ito, "A method for determination of optimal gaits with application to a snake-like serial-link structure," *IEEE Trans. Autom. Control*, vol. 50, no. 9, pp. 1291–1306, Sep. 2005.
[20] K. A. McIsaac and J. P. Ostrowski, "A framework for steering dynamic robot locomotion systems," *Int. J. Robot. Res.*, vol. 22, no. 2, pp. 83–97, Feb. 2003.
[21] D. P. Tsakiris, M. Sfakiotakis, A. Menciassi, G. L. Spina, and P. Dario, "Polychaete-like undulatory robotic locomotion," in *Proc. IEEE Int. Conf Robot. Autom.*, 2005, pp. 3029–3034.

[22] E. Shammas, A. Wolf, and H. Choset, "Three degrees-of-freedom joint for spatial hyper-redundant robots," *Mech. Mach. Theory*, vol. 41, no. 2, pp. 170–190, Feb. 2006.

[23] S. Hirose and A. Morishima, "Design and control of a mobile robot with an articulated body," *Int. J. Robot. Res.*, vol. 9, no. 2, pp. 99–114, Apr. 1990.

[24] B. A. Jones and I. D. Walker, "Kinematics for multisection continuum robots," *IEEE Trans. Robot.*, vol. 22, no. 1, pp. 43–55, Feb. 2006.

[25] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation.* Boca Raton, FL: CRC Press, 1994.

[26] F. C. Park, "Computational aspects of the product-of-exponentials forumla for robot kinematics," *IEEE Trans. Autom. Control*, vol. 39, no. 3, pp. 643–647, Mar. 1994.

Admittance Selection Conditions for Frictionless Force-Guided Assembly of Polyhedral Parts in Two Single-Point Principal Contacts

Shuguang Huang and Joseph M. Schimmels

Abstract—The admittance of a manipulator can be used to improve robotic assembly. If properly selected, the admittance will regulate a contact force and use it to guide the parts to proper positioning. In previous work, procedures for selecting the appropriate admittance for single principal contact (PC) cases were identified. This paper extends this research for some of the two PC cases—those for which each contact occurs at a single point. The conditions obtained ensure that the motion that results from frictionless contact always instantaneously reduces part misalignment. We show that, for bounded misalignments, if an admittance satisfies the misalignment-reducing conditions at a finite number of contact configurations, then the admittance will also satisfy the conditions at all intermediate configurations.

Index Terms—Assembly, compliance selection, multiple-point contact, spatial admittance.

I. INTRODUCTION

In robotic assembly, admittance control has been used to provide force regulation and force guidance. The admittance changes contact forces into changes in the velocity of the body held by the manipulator. If properly designed, the manipulator admittance will cause the held part to move toward the desired position, thus correcting misalignment. Here, procedures for selecting the appropriate manipulator admittance for polyhedral part assembly subtasks are identified.

A simple form of admittance, a linear admittance control law [1] is considered. For spatial applications, this admittance behavior has the form

$$\mathbf{v} = \mathbf{v}_0 + \mathbf{A}\mathbf{w} \qquad (1)$$

where $\mathbf{v}_0$ is the nominal twist (a six-vector), $\mathbf{w}$ is the contact wrench (force and torque) measured in the body frame (a six-vector), $\mathbf{A}$ is the admittance matrix (a 6×6 matrix), and $\mathbf{v}$ is the motion of the body.

When the held body is in contact with its mating part, the contact force will yield a motion by control law (1). An admittance design objective is that it achieves *force assembly*, i.e., for any given contact state, the admittance always leads to a motion that reduces the misalignment. Since the error-reducing motion is generated by the contact force, no sensors or actuators are needed. A desired admittance can be realized with either a passive compliant mechanism mounted on the end-effector of the manipulator [2] or through robot control [3].

Similar to previous work [4]–[6], in this paper, we consider a *measure* of error based on the Euclidean distance between an arbitrarily chosen single (fixed) point on the held body and its location when properly positioned. Since there is no "natural metric" for *finite* spatial error of a body, the error measure used is body-specific [7]. Using this measure of misalignment, the error-reduction condition can be expressed mathematically as

$$\mathbf{d}^T \mathbf{v} = \mathbf{d}^T (\mathbf{v}_0 + \mathbf{A}\mathbf{w}) < 0 \qquad (2)$$

where $\mathbf{d}$ (a six-vector for spatial motion) is the line vector from the selected point at its properly mated position to its current position and $\mathbf{w}$ is the contact wrench. Force assembly [1] requires that, at each possible misalignment, the contact force yields a motion that reduces the misalignment. As such, to achieve an error-reducing motion for a specific contact state, condition (2) must be satisfied for all possible misalignments in that contact state.

Since there are an infinite number of configurations within a given contact state, it is impossible to impose the error-reduction condition (2) on all configurations. Our objective is to identify a set of conditions that are imposed on the admittance at a finite number of configurations to ensure error-reducing motion for all configurations associated with a given contact state. Once these conditions are established, they can be used to guide the search for an appropriate admittance.

Others addressed the selection of an admittance for assembly [8]–[11]. The force-assembly approach used in this paper differs from the others' in that: 1) it applies to *any* two polyhedral parts and 2) ensures that error-reducing motion is achieved.

This paper is an extension of our previous work [6] in which sufficient conditions imposed on an admittance were used to ensure the force assembly of polyhedral parts for each of the six *single* principal contact (PC) cases. In this paper, conditions on an admittance for parts with two single-point PCs are identified. The overall admittance selection strategy is based on problem decomposition. Since for polyhedral parts, the set of assembly contacts can be decomposed into a set of contact states, the admittance for the assembly can be selected by selecting the appropriate admittance for each contact state. If sufficient conditions for each contact state in the assembly task are satisfied simultaneously, successful assembly can be ensured without having to determine in real time which contact state is being encountered.

In the two-PC cases, due to geometric constraints, the generalized coordinates used in [6] for each contact point are now coupled. This coupling is highly nonlinear and part geometry-specific. To address this in a rigorous and geometry-independent way, we consider the coupling that occurs between the two contact forces, but ignore the coupling of the two sets of coordinates. In doing this, a space of contact configurations larger than possible is considered. Due to the coupling of the two contact forces, the two sets of conditions developed independently for each of the two single-PC cases are not equivalent to the set of conditions developed for the corresponding two-PC case. Thus, in developing the admittance selection conditions for the two-PC cases, the contact forces associated with each contact are considered simultaneously.