Kinematics for Multisection Continuum Robots

Bryan A. Jones, *Member, IEEE,* and Ian D. Walker, *Fellow, IEEE*

Abstract—We introduce a new method for synthesizing kinematic relationships for a general class of continuous backbone, or *continuum*, robots. The resulting kinematics enable real-time task and shape control by relating workspace (Cartesian) coordinates to actuator inputs, such as tendon lengths or pneumatic pressures, via robot shape coordinates. This novel approach, which carefully considers physical manipulator constraints, avoids artifacts of simplifying assumptions associated with previous approaches, such as the need to fit the resulting solutions to the physical robot. It is applicable to a wide class of existing continuum robots and models extension, as well as bending, of individual sections. In addition, this approach produces correct results for orientation, in contrast to some previously published approaches. Results of real-time implementations on two types of spatial multisection continuum manipulators are reported.

Index Terms—Biologically inspired robots, continuum robot, kinematics, tentacle, trunk.

I. INTRODUCTION

CONTINUUM robots, in contrast to traditional rigid-link robots, feature a continuous backbone with no joints [24], as illustrated in Fig. 1. In this sense, they are essentially invertebrate, as opposed to the vertebrate-like structure of conventional robots. The wide range of abilities demonstrated by invertebrate limbs, such as elephant trunks [9], [14], [28], octopus arms [19], [23], and squid tentacles [30], have motivated a recent surge of research activity in continuum robots. Potential applications include navigation and operation inside complex, congested environments, such as collapsed buildings in search and rescue operations [4], [29], [32] or the human body in medical applications [25], [26]. Continuum robots also enable novel forms of environmental interaction, via compliant and/or whole-arm manipulation [13].

Realization of continuum robots proves to be a very interesting and challenging problem. A large proportion of efforts in the area have focused on design and construction of continuum robot hardware. To date, a number of interesting hardware prototypes have been produced [1], [9], [14], [15], [19],

B. A. Jones is with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762 USA (e-mail: bjones@ece.msstate.edu).

I. D. Walker is with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634 USA (e-mail: ianw@ece.clemson.edu).
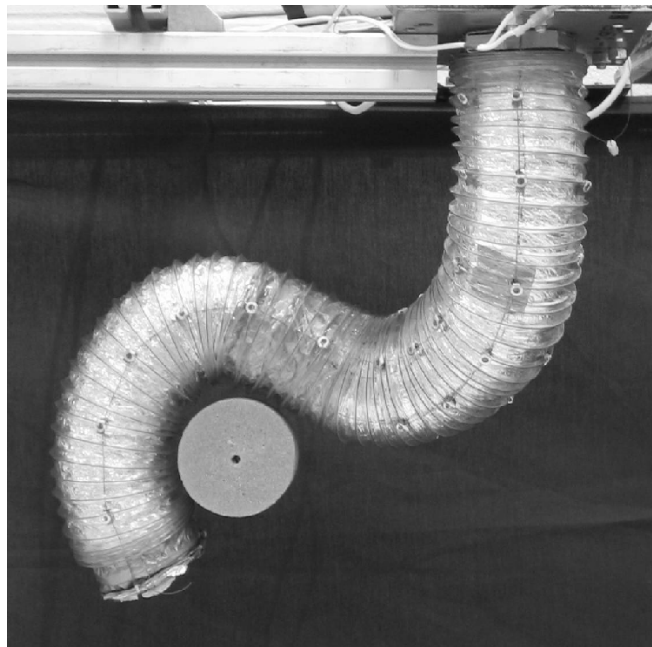
Fig. 1. Clemson's Air-OCTOR multisection continuum robot.

[22], [27], as well as several commercial products [3], [16], [28]. These manipulators demonstrate the strong potential of continuum robots. In comparison, efforts in modeling and real-time algorithm development to extract the full physical potential from these robots have lagged behind. For example, almost all previous approaches either do not address trunk orientation and approximate positional kinematics, or solve them only for limited kinematic models, which constrain the performance of continuum robot hardware to reduced classes of shapes and motions, restricting their potential applicability.

Hirose and colleagues designed the serpeniod curve, which closely matches the kinematics of a snake's body in 2-D as it crawls across the ground [15]. However, this paper does not address the 3-D kinematics of the many snake-like robots they designed, instead focusing on their planar dynamics. Chirikjian and Burdick took the opposite approach, in which they link the shape of a particular mathematical curve to a high-degree-of-freedom (DOF) robot [6]–[8]. Their modal decomposition then reduced the number of DOFs necessary to control this curve to which a robot was fitted, such as the 30-DOF robot discussed in their work. However, few continuum robots match the proposed curve, limiting its applicability. Although recent work [11] extends this approach by introducing a more convenient curve which eliminates the problem of mode switching, this method likewise relies on approximating the shape of an actual robot. An alternate approach is to exactly compute the kinematics of a high-DOF

manipulator composed of spherical joints and rigid links using standard Denavit–Hartenberg (D-H) methods [32]; however, this approach cannot be applied to continuum robots, which contain no prismatic or revolute joints. The authors of [14] returned to Hirose's concept by choosing a curve (an arc of a circle) which closely matches the kinematics of continuum robots, due to the equal distribution of forces inherent in the design of continuum robots [13], an assumption also made throughout this paper. However, the model in [14] assumes an inextensible backbone, does not correctly determine the orientation of the trunk tip due to a simplifying assumption in the model, and is limited to the particular physical design (a unique actuator arrangement) of hardware implemented by the authors of [14].

These approaches constrain the performance of continuum robot hardware to reduced classes of shapes and motions, which, in turn, has restricted their potential applicability. In this paper, we modify and extend some of the ideas in [14] as part of a new and quite general kinematic formulation for continuum robots. This approach, unlike previous work, produces complete and correct results for both task-space position and orientation. The method fully models for the first time, to the best of our knowledge, the exact kinematics of trunk sections which can extend in length, as well as bend in two dimensions using closed-form equations. The approach is also modular, allowing its application to a wide range of physical realizations, including, in particular, the common three evenly spaced tendons per section design. In addition, the resulting Jacobians, derived via a straightforward application of well-known D-H techniques and the chain rule, are relatively simple and computable in real time. This model also allows a straightforward analysis of singularities present in a continuum manipulator. In the following sections, we discuss the modeling approach, describe the various kinematic transformations and Jacobian synthesis in detail, analyze uniqueness, singularities, and accuracy, and report results from a real-time hardware implementation on two different multisection continuum robots.

II. MODULAR APPROACH

Deriving kinematics using the method introduced in this paper involves two steps. First, this work extends an earlier method [14] by formulating the trunk kinematics problem as a series of substitutions applied to a modified homogenous transformation matrix computed using a D-H-type approach. As in [14], this formulation also relies on the assumption that the trunk bends with constant curvature, due to the compliance of the trunk [13]. Second, velocity kinematics are developed by first determining the Jacobian using standard techniques based on the D-H table, then by chaining together the Jacobians produced by computing derivates of each succeeding substitution. The following paragraphs outline this approach, while the next section develops the resulting equations in detail.

In conventional manipulator forward kinematics, a standard homogenous transformation matrix $\mathbf{A}(\underline{\theta}, \underline{d})$ produced via a D-H table is used to transform local shape coordinates $(\underline{\theta}, \underline{d})$ into task coordinates $\underline{x}$, representing both end-effector position and orientation. This is based on a series of independent rotations θ_i
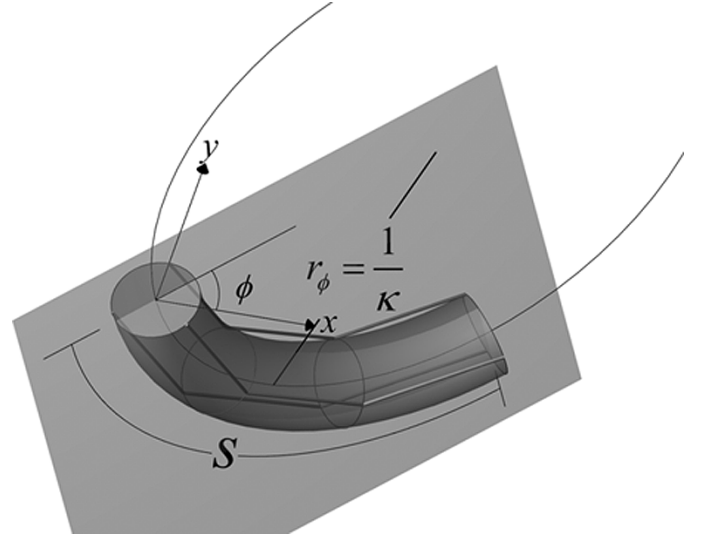

Fig. 2. Manipulator variables s, ϕ, and κ, where ϕ gives the rotation in the xy plane. Three cables of length $\underline{l}$, running in straight lines between equidistant cable guides, are also shown.

and translations d_i with one independent variable per joint at each of the i joints. However, continuum robots lack explicit joints, with the equivalent to rotations θ_i and translations d_i being an arc of constant curvature modeled by trunk parameters s, κ, and ϕ, where s gives trunk length, κ determines curvature, and ϕ determines the angle of curvature, as shown in Fig. 2. A critical question for continuum robot kinematics is how to modify conventional robot kinematics to fit this case.

A fundamental concept exploited in this paper, first introduced by [14], is to fit a conceptual conventional rigid-link manipulator to the continuous backbone of a continuum robot. Using simple geometric arguments, the work in [14] determined the relationship $[\underline{\theta} \; \underline{d}]^{\mathrm{T}} = \underline{f}_1(s, \kappa, \phi)$ between D-H table parameters θ_i and d_i and the trunk parameters s, κ, and ϕ. Implicit substitution enables computation of trunk tip position and orientation $\underline{x}$ as a function of s, κ, and ϕ as $\mathbf{A}(\underline{f}_1(s, \kappa, \phi)) = \mathbf{A}(s, \kappa, \phi)$. This allows the use of modified D-H procedures to relate end-effector task coordinates to robot shape [14].

Next, to map s, κ, and ϕ to specific continuum robot hardware, another transformation is necessary, which varies based on the construction of the robot, demonstrating the modularity of this approach. For the Clemson Elephant Trunk manipulator, the work in [14] determined a mapping $(s, k, \phi) = \underline{f}_{2a}(\underline{l})$ to compute s, κ, and ϕ, given the length $\underline{l}$ of the four cables which control each trunk section. That work, while innovative, was limited to the unique actuator arrangement in the robot in [14], and assumed constant trunk length, which restricted the generality of the results. In this paper, we introduce an analogous mapping initially reported in [18] and [19] of $(s, k, \phi) = \underline{f}_{2b}(\underline{l})$ from the lengths $\underline{l}$ of each of the three cables per section to trunk parameters s, κ, and ϕ illustrated in Fig. 2 for a class of designs typified by the Clemson Air-OCTOR robot shown in Fig. 1. This particular actuator arrangement is quite common across a wide variation in continuum robot-hardware realizations [3], [16], [23], [27], [28]. Alternatively, a third mapping $(s, \kappa, \phi) = \underline{f}_{2c}(\underline{p})$ between the air pressure in each of the three

McKibben actuators in a section of the OctArm manipulator [23] (shown in Fig. 20) may be used. Therefore, the work in this paper which builds on the above mapping from [18] and [19] is more generally applicable than the results in [14].

In summary, these forward kinematic transformations relate trunk tip position and orientation $\underline{x}$ to the cable length $\underline{l}$ or pressure $\underline{p}$ of the robot, according to the information flow

$$\underline{x} \xleftarrow{\text{D--H}} \underline{\theta}, \underline{d} \xleftarrow{f_1} s, \kappa, \phi \xleftarrow{f_2} \underline{l}/\underline{p} \qquad (1)$$

or, more formally

$$\underline{x} = \underline{f}_{\text{D--H}} \left(\underline{f}_1 \left(\underline{f}_2 (\underline{l}/\underline{p}) \right) \right). \qquad (2)$$

The corresponding transformations are fully developed and analyzed in the next section.

In forward velocity kinematics, the goal in the context of continuum robots is to relate trunk-tip linear and angular velocity $\dot{\underline{x}}$ to either cable velocities $\dot{\underline{l}}$ or actuator pressures $\dot{\underline{p}}$ through a Jacobian, so that $\dot{\underline{x}} = \mathbf{J}\dot{\underline{l}}$ or $\dot{\underline{x}} = \mathbf{J}\dot{\underline{p}}$. Following the approach of (1), this can be accomplished through a series of Jacobians according to the information flow

$$\dot{\underline{x}} \xleftarrow{\mathbf{J}_{D-H}} \dot{\underline{\theta}}, \dot{\underline{d}} \xleftarrow{\mathbf{J}_{f_1}} \dot{s}, \dot{\kappa}, \dot{\phi} \xleftarrow{\mathbf{J}_{f_2}} \dot{\underline{l}}/\dot{\underline{p}}. \qquad (3)$$

That is, a straightforward application of the chain rule [21] to (2) yields

$$\frac{d\underline{x}}{dt} = \frac{\partial \underline{x}}{\partial (\underline{\theta}, \underline{d})} \frac{\partial (\underline{\theta}, \underline{d})}{\partial (s, \kappa, \phi)} \frac{\partial (s, \kappa, \phi)}{\partial (\underline{l}/\underline{p})} \frac{d(\underline{l}/\underline{p})}{dt} \quad \text{or}$$

$$\dot{\underline{x}} = \mathbf{J}_{D-H} \mathbf{J}_{f_1} \mathbf{J}_{f_2} \left(\dot{\underline{l}}/\dot{\underline{p}} \right) \qquad (4)$$

where $\mathbf{J} = \mathbf{J}_{\text{D--H}} \mathbf{J}_{f_1} \mathbf{J}_{f_2}$. This formulation reveals an advantage of this approach: different Jacobian "modules," such as $\mathbf{J}_{f_1}$ for the Clemson Elephant Trunk, $\mathbf{J}_{f_{2b}}$ for Air-OCTOR, or $\mathbf{J}_{f_{2c}}$ for OctArm, allow different hardware realizations across a range of continuum hardware designs to be easily plugged into the derivation.

The use of standard techniques produces the Jacobian $\mathbf{J}_{\text{D--H}}$ from an appropriate D-H formulation, while $\mathbf{J}_{f_1}$ and $\mathbf{J}_{f_2}$ are directly computed as partial derivatives of $\underline{f}_1$ and $\underline{f}_2$. Section V presents each of these Jacobians in more detail. Initially, in Section III, we describe the first transformations, relating $\underline{x}$ to $(\underline{\theta}, \underline{d})$, and then relating $(\underline{\theta}, \underline{d})$ to (s, κ, ϕ). Section V then presents transformations between (s, κ, ϕ) and actuator input variables.

III. VERTEBRATE-TO-INVERTEBRATE (SHAPE) TRANSFORMATION: MODIFIED D-H KINEMATICS

The following paragraphs discuss the derivation of a new D-H formulation which fits a conceptual "virtual" (but convenient for calculation, as we shall see) rigid-link robot to the continuum backbone. Given this modified D-H formulation, we then develop geometrical solutions which relate the D-H model to the shape of a continuum trunk. The D-H table explicitly described in this paper is for a single section of a continuum arm. The model for the complete, multisection arm is formed by multiplying the single-section D-H transformation matrices in the conventional manner.
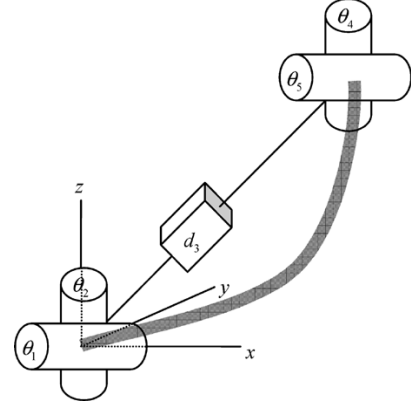

Fig. 3. Continuum robot section, shown as a thick, semitransparent line, is modeled by a coupled link/joint arrangement.

This new model, shown in Fig. 3, is based on a rigid-link arm composed of two revolute joints with intersecting axes, followed by a prismatic joint and completed with two more revolute joints, also with intersecting axes, whose joint variables are coupled to those of the first two revolute joints. This differs from the rigid-link system used in [14], where the approximation of a torsional term in the first link of the D-H table produces incorrect angular information in the resulting $\mathbf{A}$ matrix, as shown by Fig. 4. This torsional model, while a good approximation (another interesting approximate Jacobian is introduced in [20]), does not reflect the torsion-free construction of continuum robot backbones, yielding an incorrect orientation term. The kinematic arrangement introduced in this paper more precisely matches the capabilities of real continuum arms by modeling trunk bending as a bending about two axes, just as the actual robotic mechanism bends, therefore producing physically correct results for orientation, compared with the approach in [14], which models bending as a torsion followed by a single-axis bend.

The first two revolute joints which lie at the base of a trunk section point the local coordinate frame toward the section's tip. Next, the prismatic joint translates the local frame to the tip of the trunk. The final two revolute joints then orient the local frame to point along the tangent of the trunk section's tip, so that following trunk sections will be properly oriented. However, the first and last pairs of variables in this model are coupled, due to the constraints of the conceptual rigid-link robot fitting a constant-curvature continuous backbone. Therefore, the model contains only three independent variables, rather than five. This demonstrates the symmetry in the rigid-link model, reflecting that in the trunk section. Although the D-H approach above aligns trunk extension along the local x axis, geometrical transformations in (7) cause it to extend along the y axis. Therefore, the addition of a fixed rotation about the x axis at the beginning of the D-H table causes the trunk to extend along the z axis after transformations in (7), following the convention established in [14]. The fixed rotation at the end of the table then orients the tip coordinate frame to match the tip coordinate frame in [14].

This approach yields the D-H table given in Table I, illustrated in Fig. 5, which results in the homogenous transformation ma-
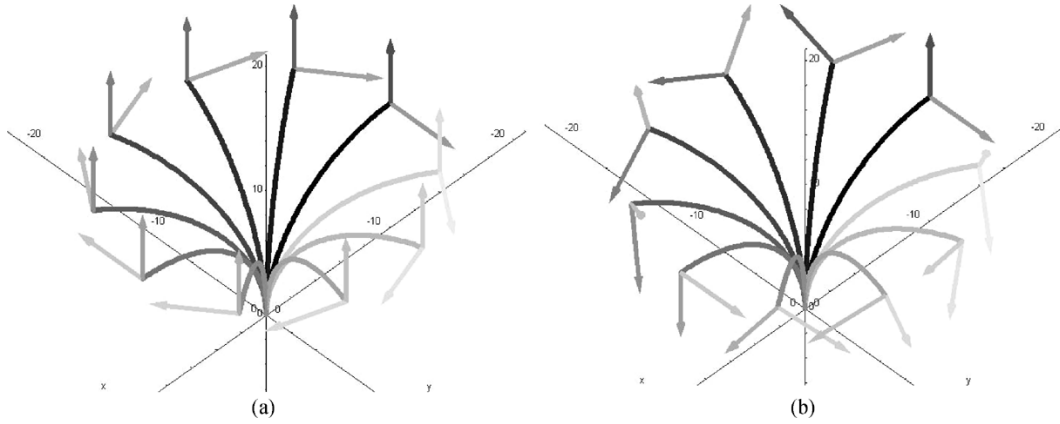
Fig. 4.　Comparison of orientational terms. Illustration (a) shows the results due to rotation of the trunk from $\phi = [0, 2\pi]$ using formulas taken from [14]. Local coordinates axes at the trunk tip indicate that the trunk torsionally rotates about the base frame z axis. However, the continuum robots discussed in this paper are mechanically constructed to be torsion-free, resulting in a local coordinate frame which rotates as shown in illustration (b), seeming to spin about the tip of the z axis. Videos at IEEE Xplore more clearly illustrate these effects.

TABLE I

NEW D-H TABLE, WHERE ASTERISKS REPRESENT JOINT VARIABLES. ADDITIONAL UNNUMBERED LINKS ORIENT TRUNK EXTENSION ALONG THE z AXIS

Link	a	α	d	θ
-	0	$\frac{\pi}{2}$	0	0
1	0	$\frac{\pi}{2}$	0	θ_1^*
2	0	$\frac{\pi}{2}$	0	$\theta_2^* + \frac{\pi}{2}$
3	0	$-\frac{\pi}{2}$	d_3^*	0
4	0	$-\frac{\pi}{2}$	0	$\theta_4^* - \frac{\pi}{2}$
5	0	0	0	θ_5^*
-	0	$-\frac{\pi}{2}$	0	0

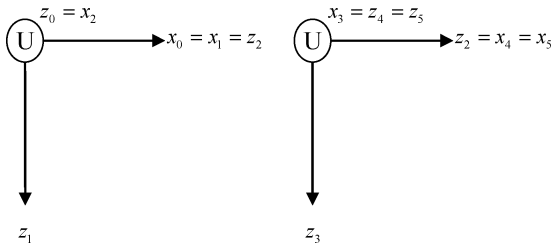Fig. 5.　Coordinate axes for D-H table given in Table I, where the circled U shows an axis extending upward from the page.
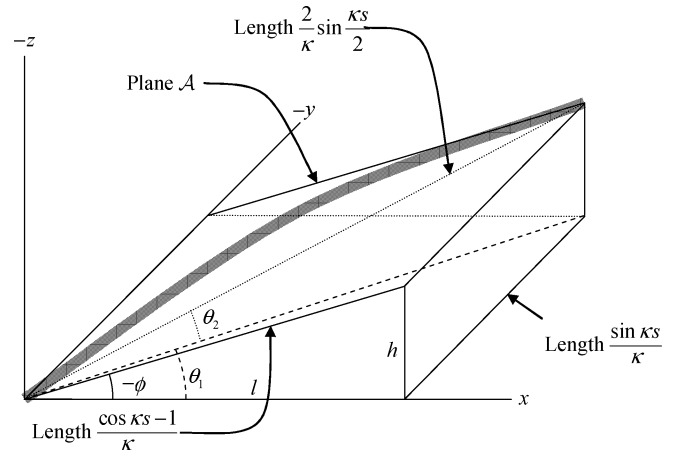

Fig. 6.　Determining a geometrical transformation between D-H parameters and trunk parameters. The semitransparent line shows the actual trunk section modeled by the geometry.

trix $\mathbf{A}$ of (5), shown at the bottom of the page, where $c_i \triangleq \cos\theta_i$ and $s_i \triangleq \sin\theta_i$. We note again here that the last two joint variables will be coupled to the first two, specifically that $\theta_4 = \theta_2$ and $\theta_5 = \theta_1 + \pi$.

Thus (5) represents the kinematics for a "virtual" rigid-link robot whose initial and final conditions align with the real trunk section. The next stage is to relate the joint variables of the virtual robot to the shape variables s, κ, and ϕ for the actual continuum section.

Thus, we now derive a transformation which relates the D-H parameters θ and d to trunk parameters s, κ, and ϕ. For this, consider Fig. 6. As shown in [14] by the translational terms in rows 1 and 2 of column 4 of the 2-D Jacobian given in [14, eq. (19)], the (x, y) coordinates of the trunk in the 2-D plane $\mathcal{A}$ are labeled in Fig. 6 as $((\cos\kappa s - 1)/\kappa, \sin\kappa s/\kappa)$. Also from [14], the distance from the base to the tip of the trunk is

$$d = \frac{2}{\kappa} \sin \frac{\kappa s}{2} \qquad (6)$$

$$\mathbf{A} = \begin{bmatrix} -c_1 s_2 s_4 c_5 + c_1 c_2 c_4 c_5 - s_1 s_5 & c_1 s_2 s_4 s_5 - c_1 c_2 c_4 s_5 - s_1 c_5 & -c_2 s_4 c_1 + s_2 c_4 & c_1 c_2 d_3 \\ -s_1 s_2 s_4 c_5 + s_1 c_2 c_4 c_5 + c_1 s_5 & s_1 s_2 s_4 s_5 - s_1 c_2 c_4 s_5 + c_1 c_5 & -s_1 c_2 s_4 + s_2 c_4 & s_1 c_2 d_3 \\ (c_2 s_4 + s_2 c_4) s_5 & -(c_2 s_4 + s_2 c_4) s_5 & c_2 c_4 - s_2 s_4 & s_2 d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (5)$$

as labeled in Fig. 6. Per the figure, note that the consideration of simple triangles yields the identities $\tan \theta_1 = (\sin \kappa s / \kappa)/l$ and $\sin \theta_2 = h/(2/\kappa) \sin(s\kappa/2)$. Lengths h and l can also be found from triangles in the diagram as $\sin(-\phi) = h/((\cos \kappa s - 1)/\kappa)$ and $\cos(-\phi) = l/((\cos \kappa s - 1)/\kappa)$, where the choice of $-\phi$ matches the convention established by [14] of defining the 2-D D-H table to bend along the $-y$ axis. Substituting and solving using the trigonometric identities $\sin(2\theta) = 2\sin\theta\cos\theta$ and $\sin A \sin B = -(1/2)(\cos(A+B) - \cos(A-B))$ gives $\theta_1 = \tan^{-1}(-1/\tan(\kappa s/2)\cos\phi)$ and $\theta_2 = \sin^{-1}(\sin(\kappa s/2)\sin\phi)$. Again from (6), the necessary translation d_3 is the distance from the trunk's base to its tip, so that $d_3 = (2/\kappa)\sin(\kappa s/2)$. The final two rotations, θ_4 and θ_5, simply repeat the rotations determined for θ_1 and θ_2, but in reverse order, so that $\theta_4 = \theta_2$ and $\theta_5 = \theta_1 + \pi$. The addition of π is necessary to correct for the choice made to bend the trunk along the $-y$ axis in [14]. Summarizing, the $\underline{f}_1$ found for the second arrow of (1) is

$$
\begin{bmatrix} \theta_1 & \theta_2 & d_3 & \theta_4 & \theta_5 \end{bmatrix}^T
$$

$$
= f_1(s, \kappa, \phi) = \begin{bmatrix} \tan^{-1}\left(\dfrac{\cos\left(\frac{\kappa s}{2}\right)}{-\sin\left(\frac{\kappa s}{2}\right)\cos\phi} \right) \\ \sin^{-1}\left(\sin\left(\frac{\kappa s}{2}\right)\sin\phi \right) \\ \frac{2}{\kappa}\sin\frac{\kappa s}{2} \\ \sin^{-1}\left(\sin\left(\frac{\kappa s}{2}\right)\sin\phi \right) \\ \tan^{-1}\left(\dfrac{\cos\left(\frac{\kappa s}{2}\right)}{-\sin\left(\frac{\kappa s}{2}\right)\cos\phi} \right) + \pi \end{bmatrix}. \quad (7)
$$

Substituting (7) into (5) produces (8), shown at the bottom of the page.

Thus, by fitting a particular "phantom" rigid-link manipulator to a constant-curvature continuum robot section, we are able to use a modified version of the classical D-H procedure to relate end-effector (task) coordinates to robot shape. For conventional rigid-link manipulators, this is directly achieved via the dependence of $\mathbf{A}$ on $(\underline{\theta}, \underline{d})$. However, since shape has a different meaning for continuum robots, the extra transformation $\underline{f}_1$ has been made, so that $\mathbf{A}$ is now an explicit function of (s, κ, ϕ). The next section discusses the derivation of $\underline{f}_{2b}$ and $\underline{f}_{2c}$, which allows the further transformation into $\mathbf{A}$ as a function of either cable lengths or actuator pressures.

IV. SHAPE-TO-INPUT TRANSFORMATIONS

Controlling the shape of the manipulator requires a kinematic model relating shape in terms of extension and bending to actuator inputs. This section outlines a new kinematic model, tailored to three-tendon, 120° configuration manipulator hardware, as illustrated by Fig. 2. The model is both new and novel, compared with previous models developed in [12] and [14], which were tailored to four-tendon sections, and therefore not
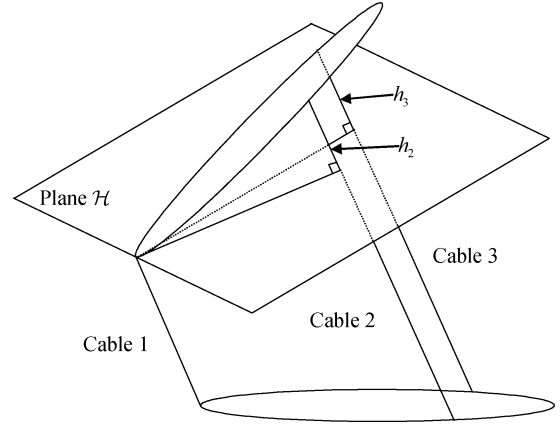

Fig. 7. Bent section of trunk.

applicable to the more general design in this paper. Likewise, this model extends the three-cable kinematics in both the trimole sensor [29], Hirose's Elastor cable kinematics [15], and various active endoscopes [5], [25], [26] by determining trunk length s based on an assumption of constant curvature.

Given three known cable lengths l_1, l_2, and l_3 in a section of the design, the constant distance d from the center of the trunk to the location of the cables, and n, the number of segments in this trunk section, the following analysis allows computation of the resulting trunk length s, curvature κ, and direction of curvature ϕ, as illustrated in Fig. 2. The three cables are assumed to lie equally spaced around the trunk, separated by angles of 120°. In addition, the curvature in each section of the trunk is assumed to be uniform, so that trunk sections can be modeled as arcs of varying-radii circles. This is typically achieved in hardware by distributing bending forces equally along the trunk, using pneumatic pressure or springs [13].

This section discusses computation of κ, ϕ, and s in three steps. First, equations are presented which determine per-cable curvatures κ_1, κ_2, and κ_3, related to the trunk's curvature. Next, from these distances, the curvature κ at angle ϕ is derived. Finally, additional equations compute the trunk length s from κ. Throughout this analysis, the problem of solving geometrical problems posed by the trunk is accomplished by analyzing a set of 2-D slices of the problem, which allow computation of the desired quantities.

A. Computation of Per-Cable Curvatures κ_i

Consider one bent segment in a section of the trunk, as illustrated by Fig. 7. This figure shows three cables, spaced at 120° intervals around the trunk, connecting two segments in a section of the trunk. The figure includes a plane $\mathcal{H}$, which passes through the point at which cable 1 connects to the trunk and is

$$
\mathbf{A} = \begin{bmatrix} \cos^2\phi(\cos\kappa s - 1) + 1 & \sin\phi\cos\phi(\cos\kappa s - 1) & -\cos\phi\sin\kappa s & \frac{\cos\phi(\cos\kappa s - 1)}{\kappa} \\ \sin\phi\cos\phi(\cos\kappa s - 1) & \cos^2\phi(1 - \cos\kappa s) + \cos\kappa s & -\sin\phi\sin\kappa s & \frac{\sin\phi(\cos\kappa s - 1)}{\kappa} \\ \cos\phi\sin\kappa s & \sin\phi\sin\kappa s & \cos\kappa s & \frac{\sin\kappa s}{\kappa} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)
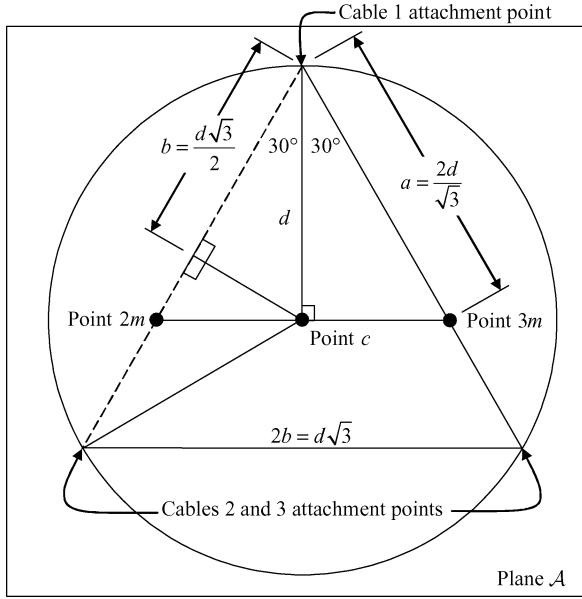$$

Fig. 8.   Top-down view of trunk.

Fig. 9.   Height of points 1 and 2 above plane.

Fig. 10.   Geometry of bent trunk section.

perpendicular to the other two cables. This plane exists due to the design of the trunk, which places the three cables at fixed intervals about the circumference of the trunk, insuring that all three cables parallel each other.

The curvature of the trunk can be calculated from h_c, the height of the point c in Fig. 8 above the plane $\mathcal{H}$. Finding this value requires knowledge of h_2 and h_3, the heights of the attachment points of cables 2 and 3 above the plane, as shown in Fig. 7. The lengths $2h_2$ and $2h_3$ by which cables 2 and 3 exceed the length l_1 of cable 1 are equally distributed above the plane shown in Fig. 7 and below a second plane, not shown in the figure, which contains the bottom attachment point of cable 1 and is perpendicular to cables 2 and 3. Therefore, the total length of cables 2 and 3 along the entire section of an n-segment trunk section can be expressed in terms of h_2 and h_3 as

$$h_2 = \frac{l_2 - l_1}{2n} \tag{9}$$

$$h_3 = \frac{l_3 - l_1}{2n}. \tag{10}$$

Note that because the plane passes through the trunk at the attachment point of cable 1, $h_1 = 0$ by definition.

Examining Fig. 8, which shows a 2-D slice of the trunk viewed looking down on its top segment, reveals $a = 2d/\sqrt{3}$ and $b = d\sqrt{3}/2$. As shown in Fig. 9, viewing a different 2-D slice perpendicular to plane $\mathcal{A}$ in Fig. 8 which contains the dotted line in Fig. 8 allows determination of $h_{2m} = (l_2 - l_1)/3n$. Likewise, $h_{3m} = (l_3 - l_1)/3n$. Height h_c is equidistant from these two points, so its height is their average, producing

$$h_c = \frac{l_3 + l_2 - 2l_1}{6n}. \tag{11}$$

Note that h_c may be negative at some orientations of the trunk, where $2l_1 > l_2 + l_3$. Though unexpected, the following geometrical uses of h_c are correct in this case.

Based on the assumptions, curvature of the trunk segment is evenly distributed along each of the n segments in the trunk section. Therefore, the curvature κ of each segment remains con-
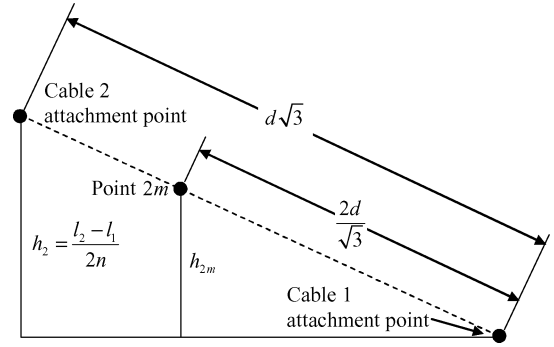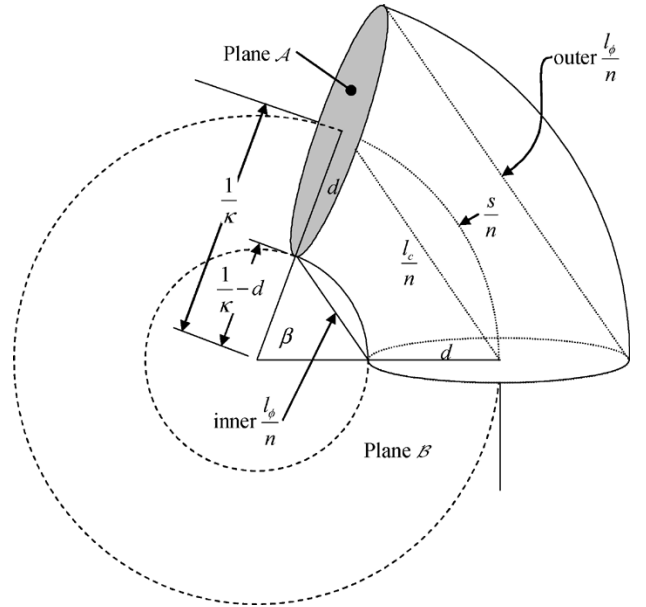
stant along the section. The curvature of a section is defined as $\kappa = 1/r_\phi$, where r_ϕ is the radius of a circle which fits the arc produced by the bending of a trunk section. Consider a virtual cable of length l_c/n as shown in the figure, attached to the center of a cross section of the trunk and traveling from top to bottom of a segment of the trunk. At some as yet unknown angle ϕ, the situation shown in Fig. 10 exists, where an isosceles triangle with sides of length $r_\phi = 1/\kappa$ and l_c/n subtends β degrees of an arc of the circle formed by the trunk. Considering plane $\mathcal{A}$ containing the circle which forms the top of the trunk shown in Fig. 8, note the leg of this triangle which lies in this plane. This leg can be viewed as the vector $\kappa\angle\phi$ in polar coordinates in plane $\mathcal{A}$, where the origin of the plane is at point c. The magnitude of corresponding vectors $\underline{\kappa}_1$, $\underline{\kappa}_2$, and $\underline{\kappa}_3$, notated κ_{1-3}, which pass through the attachment points for cables 1–3 allow determination of $\kappa\angle\phi$ via coordinate transformations detailed in the next section.

To compute κ_1, rotate a second plane $\mathcal{B}$, shown in Fig. 10, which is perpendicular to plane $\mathcal{A}$ containing l_ϕ/n and l_c/n, about l_c/n from angle ϕ to an absolute angle of $90°$, the location of cable 1, so that the plane now contains l_1 and l_c/n. Fig. 11 shows the contents of plane $\mathcal{B}$ after rotation, which contains an isosceles triangle with sides r_1 and l_c/n. Similar to the
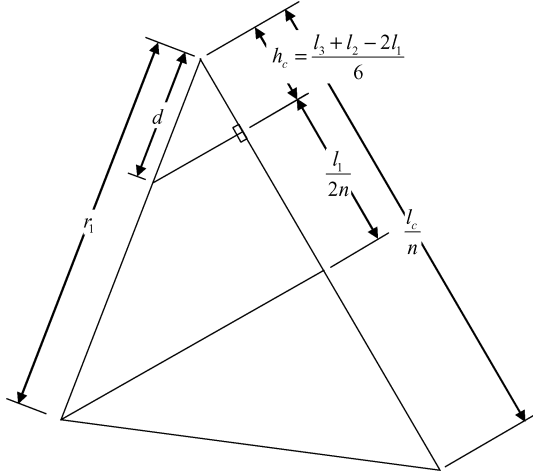
Fig. 11. Determination of r_1.

reasoning in (9) and (10), the length l_c/n of this virtual cable can be divided into l_1/n, the length of cable 1 in one trunk segment, and h_c, the height this virtual cable extends above the plane anchored by l_1, so that

$$\frac{l_c}{n} = \frac{l_1}{n} + 2h_c. \tag{12}$$

Relating this to the smaller similar triangle in the figure yields $h_c/d = (h_c + l_1/2n)/r_1$. Substituting in (11) gives $r_1 = d(l_1 + l_2 + l_3)/(l_3 + l_2 - 2l_1)$. Noting that $\kappa = 1/r$, $\kappa_1 = (l_3 + l_2 - 2l_1)/d(l_1 + l_2 + l_3)$. Repeating the process with the assumption that $h_2 = 0$ instead of $h_1 = 0$ as before

$$\kappa_2 = \frac{l_3 + l_1 - 2l_2}{d(l_1 + l_2 + l_3)}. \tag{13}$$

Similarly, by assuming $h_3 = 0$

$$\kappa_3 = \frac{l_2 + l_1 - 2l_3}{d(l_1 + l_2 + l_3)}. \tag{14}$$

B. Computation of κ and ϕ, Trunk Segment Curvature and Angle of Curvature

To convert the κ_i per-cable curvatures into ϕ, the angle at which the trunk bends, as illustrated in Fig. 2, and κ, the curvature at angle ϕ, consider determining ϕ and κ based on a coordinate transformation of values just derived for κ_1, κ_2, and κ_3. Note that the positions of cables 1–3 on the circumference of the trunk, with respect to the x axis, are $90°$, $210°$ and $-30°$, as shown in Fig. 12. Viewing these curvatures in plane $\mathcal{A}$ as vector quantities by noting the angle relative to the x axis along which each curvature extends yields $\underline{\kappa}_1 = \kappa_1 \angle 90°$, $\underline{\kappa}_2 = \kappa_2 \angle 210°$, and $\underline{\kappa}_3 = \kappa_3 \angle -30°$. Using the change of basis matrix

$$\mathbf{B} = \begin{bmatrix} \cos(210°) & \sin(210°) \\ \cos(-30°) & \sin(-30°) \end{bmatrix}$$

to convert to a standard orthogonal basis so that $\begin{bmatrix} \kappa_x & \kappa_y \end{bmatrix} = \mathbf{B}^{-1}\begin{bmatrix} \kappa_2 & \kappa_3 \end{bmatrix}$ yields $\kappa_x = (\kappa_3 - \kappa_2)/\sqrt{3}$ and $\kappa_y = -\kappa_2 - \kappa_3 = \kappa_1$. In polar form, $\phi = \tan^{-1}(\kappa_y/\kappa_x)$ and
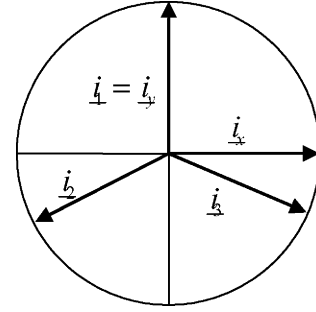
Fig. 12. Unit vectors $\underline{i}_{1-3,x,y}$ along which vectors $\underline{\kappa}_{1-3,x,y}$ lie.
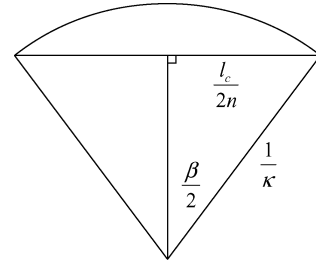
Fig. 13. Computing s.

$\kappa_\phi = \sqrt{\kappa_x^2 + \kappa_y^2}$. Substituting (13) and (14), then simplifying, yields

$$\kappa = 2\frac{\sqrt{l_1^2 + l_2^2 + l_3^2 - l_1 l_2 - l_2 l_3 - l_1 l_3}}{d(l_1 + l_2 + l_3)} \tag{15}$$

$$\phi = \tan^{-1}\left(\frac{\sqrt{3}}{3}\frac{l_3 + l_2 - 2l_1}{l_2 - l_3}\right). \tag{16}$$

C. Computation of s, the Length of a Trunk Section

Each section of the trunk has n segments. In each segment, the cable runs in a straight line from one cable guide to the next. Solving for the cable length in one section allows determination of the cable length in all sections. Therefore, consider just one section of the trunk, whose length is therefore s/n.

Bending the trunk produces the geometry shown in Fig. 10. From (12), $l_c = l_1 + 2nh_c$. Substituting from (11) yields

$$l_c = \frac{l_3 + l_2 + l_1}{3}. \tag{17}$$

Note that in a circle of radius r, the angle φ subtended by a portion of the circumference of length a obeys the relationship $2\pi r/a = 2\pi/\varphi$. To compute β for the outer circle shown in Fig. 10, substitute $r = 1/\kappa$, $a = s/n$, and $\varphi = \beta$ into the above equation, and solve for β to obtain

$$\beta = \frac{\kappa s}{n}. \tag{18}$$

Examination of the triangle drawn inside the outer circle of Fig. 10, shown in Fig. 13, reveals that $\sin(\beta/2) = (l_c/2n)/(1/\kappa)$. Substituting (18) and solving yields $s = (2n/\kappa)$
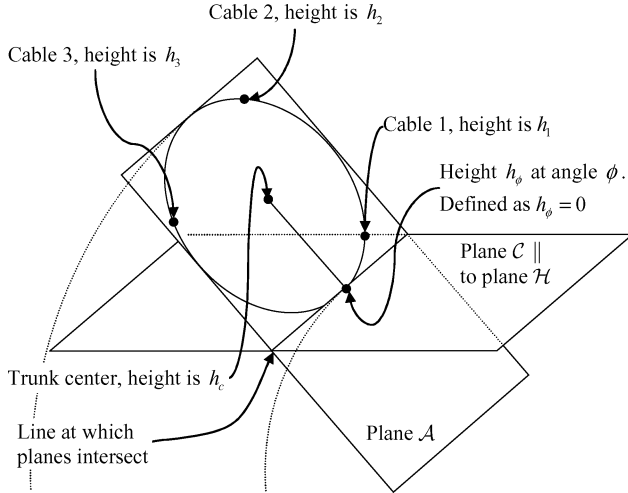
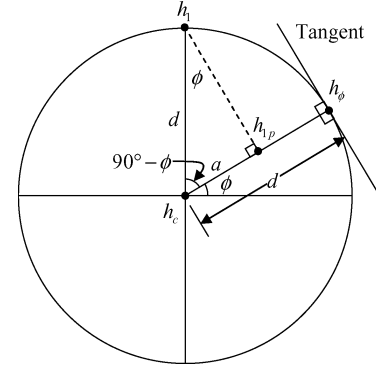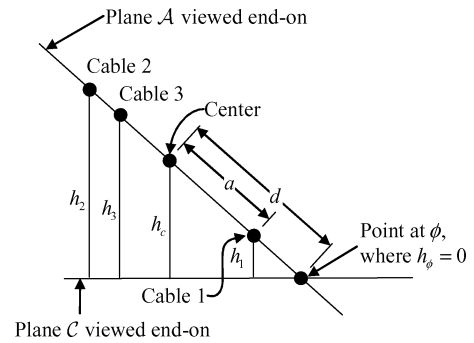Fig. 14.   Plane containing circle and reference plane.

Fig. 15.   Projecting h_1 along a line parallel to the tangent shown in Fig. 16.

Fig. 16.   End-on view of planar intersection shown in Fig. 13.

$\sin^{-1}(l_c\kappa/2n)$. Substituting from (15) and (17) yields s in terms of cable lengths as

$$s = \frac{nd\,(l_1 + l_2 + l_3)}{\sqrt{l_1^2 + l_2^2 + l_3^2 - l_1 l_2 - l_2 l_3 - l_1 l_3}}$$
$$\cdot \sin^{-1}\left(\frac{\sqrt{l_1^2 + l_2^2 + l_3^2 - l_1 l_2 - l_2 l_3 - l_1 l_3}}{3nd} \right). \quad (19)$$

Note that in the limiting case when the cable lengths are equal, the terms under the square root go to zero so that

$$\lim_{\sqrt{l_1^2 + \cdots} \to 0} s = (l_1 + l_2 + l_3)/3.$$

Together, (15), (16), and (19) comprise the function $(s, k, \phi) = \underline{f}_{2b}(\underline{l})$ for a cable-actuated robot.

Because pneumatic actuators bend continuously, the derivation for $\underline{f}_{2c}$ as defined in (1) can be found by increasing the number of cable guides to infinity. Examining the limiting case when $n \to \infty$ of $\underline{f}_{2b}$ yields identical expressions for (15) and (16), while (19) becomes

$$s = \frac{l_1 + l_2 + l_3}{3}. \quad (20)$$

Therefore, $(s, k, \phi) = \underline{f}_{2c}(\underline{l})$ is defined by (15), (16), and (20).

D. Inverse Kinematics (Manipulator Curvature to Cable Lengths)

Consider the manipulator section shown in Fig. 2. Given the length s of one section of the manipulator, the curvature κ of that section, the angle at which the curvature lies ϕ, the distance d from the center of the manipulator to the location of the cables, and the number of segments in this manipulator section n, this section discusses the computation of the lengths of the three cables l_1, l_2, and l_3 due to the configuration of the manipulator section. This is the inverse kinematics problem.

As shown in Fig. 14, the top of one segment in a section of the elephant trunk lies in plane $\mathcal{A}$. Heights can be measured from a second plane $\mathcal{C}$, which is parallel to plane $\mathcal{H}$ shown in Fig. 7. These two planes intersect along a single line tangent to height h_ϕ, as shown in the figure. Viewing these two planes

end-on as in Fig. 16, note that heights measured from plane $\mathcal{C}$ to a point on $\mathcal{A}$ do not change when following any line on $\mathcal{A}$ parallel to this tangent. For example, as illustrated in Fig. 15, projecting the height h_1 along a line parallel to the tangent at h_ϕ gives $h_1 = h_{1p}$, where h_{1p} is the height at a point along the line connecting h_c and h_ϕ. Knowledge of the slope of the line created by viewing plane $\mathcal{A}$ end-on and of a position on this line allows computation of the height of this position above plane $\mathcal{C}$, which appears as horizontal line in the figure. Therefore, the height of an arbitrary point on the circle such as $h_1 = h_{1p}$ can be determined by projecting the point along a line parallel to the tangent of h_ϕ onto the diagonal line in Fig. 16, then applying the slope-intercept form of that line to find the point's height.

In the forward kinematics derivation, the height $h_1 = 0$ was chosen to anchor the plane, as shown in Figs. 7–9. For the inverse kinematics derivation, the plane will be anchored by $h_\phi = 0$ instead. Examining the isosceles triangle with sides $1/\kappa$ and l_c/n shown in Fig. 10 creates Fig. 17, which reveals $\sin(\beta/2) = h_c/d$. Substituting (18) and solving produces $h_c = d\sin(\kappa s/2n)$. Projecting these points onto the diagonal line of Fig. 16 allows the line to be defined as $h_{1p} = h_1 = ((h_c - h_\phi)/d)(d - a) + h_\phi$. Also note from Fig. 15 that $\cos(\pi/2 - \phi) = a/d = \sin\phi$. Substituting and simplifying shows that $h_1 = d\sin(\kappa s/2n)(1 - \sin\phi)$.

Similar to the reasoning in (9) and (10), the length of cable 1 l_1 can be divided into l_ϕ, the length of a virtual cable attached at angle ϕ on the circumference of the manipulator and running the length of the manipulator, and h_1, the height cable 1 extends above the plane anchored by l_ϕ, so that $l_1 = l_\phi + 2nh_1$.
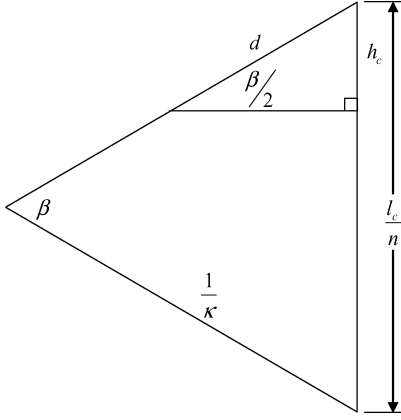
Fig. 17. Triangle from Fig. 10 used to relate h_c to β.
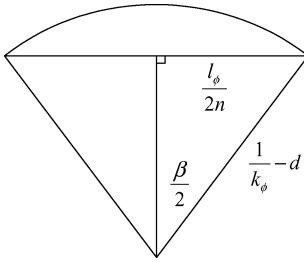
Fig. 18. Triangle from Fig. 10 used to determine length l_ϕ.

To find l_ϕ, note that the length of a cable l_ϕ attached to the circumference of the manipulator at the point of maximum curvature ϕ forms the triangle shown in the inner circle of Fig. 10, redrawn in Fig. 18, yielding $\sin(\beta/2) = (l_\phi/2n)/(1/\kappa - d)$. Therefore, substituting (18), the cable length can be computed as $l_\phi = 2n(1/\kappa - d)\sin(\kappa s/2n)$. Substituting and simplifying

$$l_1 = 2n\sin\left(\frac{\kappa s}{2n}\right)\left(\frac{1}{\kappa} - d\sin\phi\right). \tag{21}$$

Likewise, rotating to 210° and 330° to find h_2 and h_3, the heights of the attachment points of cables 2 and 3, then resolving produces

$$l_2 = 2n\sin\left(\frac{\kappa s}{2n}\right)\left(\frac{1}{\kappa} + d\sin\left(\frac{\pi}{3} + \phi\right)\right) \quad \text{and} \tag{22}$$

$$l_3 = 2n\sin\left(\frac{\kappa s}{2n}\right)\left(\frac{1}{\kappa} - d\cos\left(\frac{\pi}{6} + \phi\right)\right). \tag{23}$$

Note that because $\lim_{x\to 0}\sin x = x$, in the limiting case $\lim_{\kappa\to 0} l_1 = 2n(\kappa s/2n)(1/\kappa - d\sin\phi) = s$. In the same way

$$\lim_{\kappa\to 0} l_2 = \lim_{\kappa\to 0} l_3 = s.$$

This completes the derivation for a cable-actuator robot such as Air-OCTOR of $f_{2b}^{-1}(s,\kappa,\phi) = \begin{bmatrix} l_1 & l_2 & l_3 \end{bmatrix}^T$ as defined in (1), where l_1, l_2, and l_3 are given in (21)–(23) above.

Because pneumatic actuators bend continuously, the derivation for f_{2c}^{-1} as defined in (1) can be found by increasing the number of cable guides to infinity. Examining the limiting case when $n \to \infty$ of (21)–(23) yields

$$f_{2c}^{-1}(s,\kappa,\phi) = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} = \begin{bmatrix} s\left(1 - \kappa d\sin\phi\right) \\ s\left(1 + \kappa d\sin\left(\frac{\pi}{3} + \phi\right)\right) \\ s\left(1 - \kappa d\cos\left(\frac{\pi}{6} + \phi\right)\right) \end{bmatrix}. \tag{24}$$

Now that all the relationships in (1) describing the forward kinematics for a trunk have been determined, the next section will discuss the derivation of the forward velocity kinematics.

V. JACOBIAN DEVELOPMENT

As overviewed by (3) and (4), straightforward application of the chain rule with necessary substitutions yields the Jacobian

$$\dot{\underline{x}} = \mathbf{J}(\underline{l}/\underline{p})\dot{\underline{l}}/\dot{\underline{p}} \tag{25}$$

where $\dot{\underline{l}}$ and $\dot{\underline{p}}$ are the time derivates of length or pressure, and the notation $(\dot{\underline{l}}/\dot{\underline{p}})$ implies the selection of either $(\dot{\underline{l}})$ or $(\dot{\underline{p}})$. Each Jacobian in the chain of Jacobians must be computed. Computing the Jacobian which relates $\dot{\underline{x}}$ to $\dot{\underline{\theta}}$ and $\dot{\underline{d}}$ produced by classical D-H techniques based on the D-H table in Table I yields

$$\mathbf{J}_{D-H} = \begin{bmatrix} -s_1c_2d_3 & -c_1s_2d_4 & c_1c_2 & 0 & 0 \\ 0 & -c_2d_3 & -s_2 & 0 & 0 \\ c_1c_2d_3 & -s_1s_2d_3 & s_1c_2 & 0 & 0 \\ 0 & s_1 & 0 & s_1 & -c_1s_{24} \\ -1 & 0 & 0 & 0 & -c_{24} \\ 0 & -c_1 & 0 & -c_1 & -s_1s_{24} \end{bmatrix} \tag{26}$$

where $c_{ij} \triangleq \cos(\theta_i + \theta_j)$ and $s_{ij} \triangleq \sin(\theta_i + \theta_j)$. Computing $\mathbf{J}_{\underline{f}_1}$ from (7) yields

$$\mathbf{J}_{\underline{f}_1} = \begin{bmatrix} \frac{\sin\phi\sin\kappa s}{g} & -\frac{s\cos\phi}{g} & -\frac{\kappa\cos\phi}{g} \\ \frac{2\cos\phi\sin\frac{\kappa s}{2}}{h} & \frac{s\sin\phi\cos\frac{\kappa s}{2}}{h} & \frac{\kappa\sin\phi\cos\frac{\kappa s}{2}}{h} \\ 0 & \frac{\kappa s\cos\frac{\kappa s}{2} - 2\sin\frac{\kappa s}{2}}{\kappa^2} & \cos\frac{\kappa s}{2} \\ \frac{2\cos\phi\sin\frac{\kappa s}{2}}{h} & \frac{s\sin\phi\cos\frac{\kappa s}{2}}{h} & \frac{\kappa\sin\phi\cos\frac{\kappa s}{2}}{h} \\ \frac{\sin\phi\sin\kappa s}{g} & -\frac{s\cos\phi}{g} & -\frac{\kappa\cos\phi}{g} \end{bmatrix} \tag{27}$$

where $g = \cos^2\phi\cos\kappa s - \cos\kappa s - \cos^2\phi - 1$ and $h = \sqrt{2\cos^2\phi + 2 + 2\cos\kappa s - 2\cos\phi\cos\kappa s}$. Computing $\mathbf{J}_{\underline{f}_2}$ results in long terms, and is omitted for brevity.

Next, substituting (7) into (26), then multiplying by (27) produces

$$\mathbf{J}_{D-H,f_1}$$
$$= \begin{bmatrix} -\frac{\sin\phi(\cos\kappa s - 1)}{\kappa} & -\frac{\cos\phi(\kappa s\sin\kappa s + \cos\kappa s - 1)}{\kappa^2} & -\cos\phi\sin\kappa s \\ 0 & \frac{\kappa s\cos\kappa s - \sin\kappa s}{\kappa^2} & \cos\kappa s \\ \frac{\cos\phi(\cos\kappa s - 1)}{\kappa} & \frac{\sin\phi(\kappa s\sin\kappa s + \cos\kappa s - 1)}{\kappa^2} & \sin\phi\sin\kappa s \\ \cos\phi\sin\kappa s & s\sin\phi & \kappa\sin\phi \\ 1 - \cos\kappa s & 0 & 0 \\ -\sin\phi\sin\kappa s & s\cos\phi & \kappa\cos\phi \end{bmatrix}. \tag{28}$$

Finally, substituting $\underline{f}_2$ for either cable lengths or pressures into (28) and multiplying by $\mathbf{J}_{f_2}$ produces the final Jacobian, which satisfies (25). Due to its length, this result is omitted. However, the procedure is quite straightforward, and leads to computationally tractable solutions, as discussed in the following section. Notice that by including our more general formulation for $\underline{f}_2$, the resulting approach is applicable to a much wider class of continuum robots than the approach in [14].

VI. Discussion and Implementation

A. Uniqueness

There are a number of ways in which some of the results of the derivation developed in this paper can be obtained. This section reviews alternative representations, approaches, and derivations.

First, the parameters chosen to represent an arc in 3-D are not unique. Rather than describe the arc in terms of its length s, curvature κ, and angle of curvature ϕ, as shown in Fig. 2 (a polar representation), the arc could be described using a Cartesian coordinates in terms of length s, curvature along the x axis κ_x, and curvature along the y axis κ_y. Or, instead of arc length s, the number of radians subtended by the arc θ could be used. A number of other approaches are possible [2]; the choice of representation in this paper was motivated by both a desire to maintain a common scheme with previous work such as [14], and by the ease of modifying the angle of curvature ϕ to compensate for rotations relative to the home (typically user) coordinate frame, due to the way in which the robot is mounted, where actuators are placed in each section of the robot, and where the mobile platform base is placed relative to the home frame.

Likewise, the backbone presented in Section III as the D-H table given by Table I and transformations in Fig. 6 and (7) is not unique. An alternative approach, for example, is to append a counter-torsional term of $\theta_5 = -\phi$ to the four-link D-H table developed by [14]. A second approach is to abandon the backbone-fitting method, finding the position and orientation of the trunk tip through other methods. For example, rotating the trunk-tip position in plane $\mathcal{A}$ in Fig. 6 of $\left[(\cos \kappa s - 1)/\kappa \quad \sin \kappa s / \kappa \quad 0 \right]^{\mathrm{T}}$ about the y axis by $-\phi$ produces the same positional results as the backbone approach. Using a differential curve representation, such as the Serret–Frenet, can also produce identical positional results, as shown by [14]. Likewise, an integral representation, such as those developed by Chirikjian and Burdick [8] or in [13], can be used to obtain identical positional results. Following the approach of [8], this integral representation can then be factored into a modal decomposition, whose modal functions consist of arc length s, angle of curvature ϕ, and the product of arc length and curvature $\kappa s = \theta$, where θ gives the angle subtended by the arc representing the trunk. However, this model for the trunk is exact, instead of the approximate model introduced by [8], leading to greater ability to precisely specify trunk shape. Additionally, our directly physically motivated "backbone approach" produces enhanced insight into the mechanics of the trunk. For example, trunk sections in this paper are chosen not as arbitrary points, as in [8], but at the interface between two groups of McKibben actuators, again carefully modeling the physical construction of the trunk.

The orientation of the tip and the computation of its angular velocity requires additional work when the backbone approach is not followed. One approach, detailed in [8], is to include a rotated frame which allows specification of orientation, independent of the frame produced by the curve's mathematical representation. The angular velocity is then defined by $\mathbf{S}(\underline{\omega}) = \dot{\mathbf{R}}\mathbf{R}^{\mathrm{T}}$, where $\mathbf{R}$ gives the rotation matrix of the frame and $\mathbf{S}(\underline{\omega})$ describes factoring a skew-symmetric matrix $\mathbf{S}$ to a three-parameter vector $\underline{\omega}$. Factoring the resulting $\underline{\omega}$ into a Jacobian then yields results parallel to those obtained using the backbone approach. The backbone method was chosen in our approach because it provides a good method for computing the kinematics of a wide range of continuum manipulators, including those not considered in our work, simply by modifying the appropriate D-H entries and geometrical transformations to match a new design. In comparison, finding a correct orientation frame is nontrivial and involves additional computations to develop angular velocity.

B. Accuracy

Two concerns affect the accuracy of a continuum robot, defined as the distance between desired and achieved locations in the workspace. First, error due to the compliance of the manipulator, in contrast to the rigid construction of traditional robot, is significant; second, the nature of the modeling process must be examined for possible sources of error.

The first category of accuracy errors, as discussed in [33], arises from the construction of the robot. Traditional robots, due to their extremely rigid design, can achieve very high accuracy and repeatability. For example, a trunk-like high-DOF manipulator [32] can be expected to achieve very high accuracy. Continuum robots, due to their soft, compliant design, cannot. However, high accuracy is not the intended niche of continuum robots. They are generally poor as high-accuracy positioning tools when compared with traditional rigid-link robots. Instead, their compliance can be used to reduce the accuracy needed to perform manipulation tasks in unstructured environments by allowing the inherent flexibility of the manipulator to compensate for a lack of precise knowledge of objects to be grasped.

The second category of errors arises due to the modeling approach. Some approaches discussed in the introduction, such as [8], model the robot as an arbitrary curve, then perform an error-minimizing fitting process to match the actual robot to the curve. In contrast, this paper chooses a curve (an arc of a circle) which the manipulator naturally assumes due to its design, as forces due to pneumatic pressure, for example, are equally distributed into a constant curvature along each section of the robot [13]. Therefore, this approach does not suffer from significant errors due to a fitting process. However, this model assumes the absence of external forces which can bend the trunk in a noncircular fashion, leading to errors. For example, when the trunk grasps a noncircular object such as a box, it no longer bends in a circular arc. Likewise, when it supports a gravity load, either the mass of the trunk itself or of a payload the trunk is carrying, the trunk sags in a noncircular fashion [26]. Finally, this kinematic analysis ignores all dynamics effects due to inertial, centrifugal, and Coriolis forces. Unfortunately, dynamics in the 3-D case remain in integral-differential form, rendering them unsuitable for real-time control [13], [17], or can be computed only as an approximation in closed form [7]. However, as discussed above, these issues are less significant for continuum robots designed for operation in unstructured environments, than for traditional rigid-link manipulators which operate primarily in structured environments.

C. Singularities

Robots constructed of a combination of revolute joints typically exhibit singularities at their workspace boundaries, in addition to inherent singularities due to their mechanical design. Because the continuum robots analyzed in this paper can theoretically extend to any arbitrary arc length s, they do not possess workspace boundary singularities. Actuator length limits, however, impose workspace constraints, but they do not enter into the kinematics formulation. The only singular configuration occurs when the trunk is straight ($\kappa = 0$). In this case, velocities normal to the angle of trunk curvature ϕ cannot be produced. However, the angle of curvature is essentially meaningless when the trunk is not curved, so this configuration presents no practical problems. An alternate representation for the trunk, for example, as two orthogonal curvatures κ_x and κ_y discussed in Section VI-A, would eliminate this singularity.

D. Simulation Results

As a prelude to implementation on continuum robot hardware such as the Clemson Air-Octor robot [18] or the Penn State OctArm robot [23], the Jacobians derived in the previous section were symbolically computed using Maple 9.5. First, a D-H table for an n-section trunk is determined. Given that table, a second routine then computes $\mathbf{J}_{\mathrm{D-H}}$ by applying standard Jacobian techniques to the table. Next, a set of n-section geometrical transformations representing $\underline{f}_1$ are computed, and their Jacobian $\mathbf{J}_{\underline{f}_1}$ computed. The product $\mathbf{J}_{\mathrm{D-H}}\mathbf{J}_{\underline{f}_1}$ is then computed by substituting $\underline{f}_1$ into $\mathbf{J}_{\underline{f}_1}$, then multiplying. Finally, $\mathbf{J}_{\underline{f}_2}$ is formed for an n-section trunk, substitutions are made, then the product $\mathbf{J}_{\mathrm{D-H}}\mathbf{J}_{\underline{f}_1}\mathbf{J}_{\underline{f}_2}$ is formed. This symbolically computed matrix is then transformed into C code using Maple's CodeGeneration package. The resulting code, which has undergone common subexpression elimination via the optimize option to CodeGeneration, is then combined with a CLapack call to solve (25) as

$$\underline{\dot{l}}/\underline{\dot{p}} = \mathbf{J}^+ \left(\underline{l}/\underline{p}\right) \underline{\dot{x}} \tag{29}$$

for $\underline{\dot{l}}$ or $\underline{\dot{p}}$ using a QR rotation with column pivoting. This code is then placed in a Simulink block, which becomes part of a block diagram which reads $\underline{\dot{x}}$ from a joystick, computes the resulting $\underline{\dot{l}}/\underline{\dot{p}}$ based on the Jacobian, then updates a OpenInventor-based simulation of the trunk and also computes actuator torques or pneumatic pressures necessary to achieve the desired $\underline{\dot{l}}/\underline{\dot{p}}$. Fig. 19 illustrates the results of this simulation, as the trunk is moved along the z axis.

The performance of the algorithm is excellent. On a 2.8-GHz Pentium 4 processor for a three-section trunk, which has nine DOFs, the Jacobian is computed, then $\underline{\dot{l}}/\underline{\dot{p}}$ is computed based on a given 3-DOF linear velocity $\underline{\dot{x}}$ in 0.163 ms. Therefore, this algorithm can easily be included in a real-time control system, as detailed in the following paragraphs.

E. Implementation

Using the Mathwork's Real Time Workshop, the Simulink block diagram which contains code to solve inverse velocity kinematics was transformed into a C program, which outputs
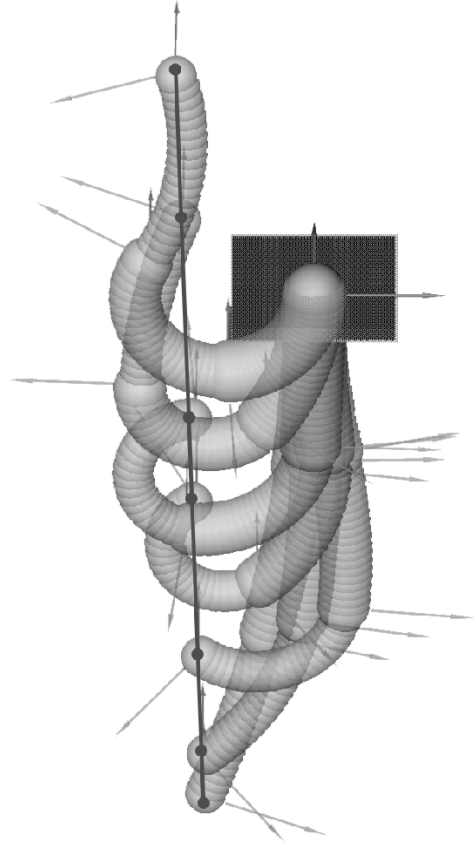

Fig. 19. OpenInventor-based trunk simulation showing the trunk, under Jacobian control, moving along a straight-line workspace trajectory. Arrows indicate the (x, y) axes local to each trunk section.

computed motor torques as voltages through a Servo-To-Go digital-to-analog board, and feeds motor encoder positions back to the control loop. The C code was then compiled as real-time executable under the QNX Momentics real-time operating system. The code runs efficiently in a real-time control loop operating at 500 Hz, providing responsive control to user input through a joystick. The following paragraphs detail the approach to implementation for both the cable-driven Air-Octor and the pneumatically actuated OctArm continuum robots.

The two-section Air-Octor robot [18] shown in Fig. 1 consists of a central hollow chamber, constructed of dryer hose, into which air pressure is blown, enabling variation of each section's stiffness. Helically wrapped metal coils within the hose allow the tube to extend along its length, but not to change its diameter. Surrounding this chamber, a second chamber of ferret tubing, again with helically wrapped metal coils, provides attachment points for a series of eight cable guides for each section. Six motors, three for each of the two sections, are arranged around the top of the robot at 60° intervals. Each motor also includes an encoder to measure the length of cable paid out of a spool attached to the motor shaft through a gearing mechanism. Three cables terminate at the first section, while the other three run through the first section, terminating at the second section, allowing shape control of both sections. A simple proportional-integral-derivative loop servos desired cable lengths to actual lengths. Implementation of inverse cable kinematics embodied in (21)–(23) map from desired trunk shape, specified by
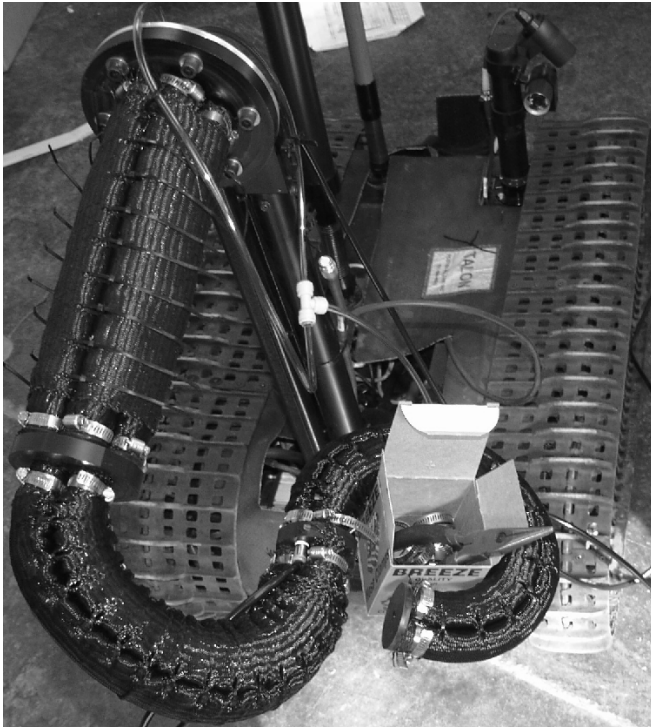
Fig. 20.   OctArm mounted on the Talon mobile platform. *Courtesy of Chris Rahn.*
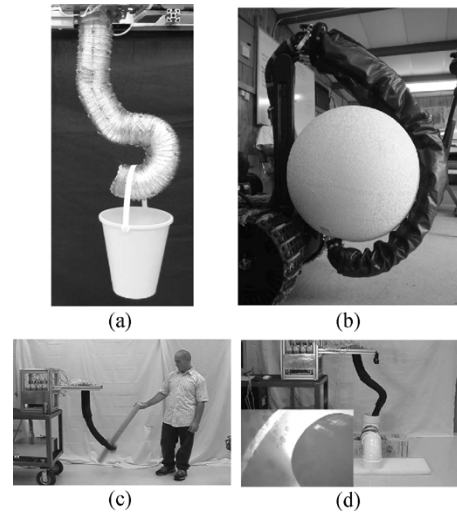

Fig. 21.   Abilities of the robot, such as (a) hook-style grasping, (b) large-object grasping and whole-arm manipulation, (c) prey capture, and (d) hole exploration using a tip-mounted camera view shown in the lower left corner of (d).

a joystick through several non-Jacobian control modes useful for whole-arm manipulation [10], to desired cable lengths. In addition, Jacobian (28) and (29) allow end-point control of the two-section robot based on joystick input.

The OctArm manipulator [23] shown in Fig. 20 consists of three or four sections, depending on the design. Each section is actuated by joining three McKibben actuators which extend only in length between two end plates, so that the trunk can extend or bend in any direction. Air pressure is regulated using a set of valves which maintain a desired pressure, input as a voltage. Shape feedback is still in development, so the arm runs open loop. The length of each section of the arm was measured by inflating all three actuators in a section to a given pressure, then measuring length. A third-order polynomial fit of this measured length-to-pressure relationship was then used to transform desired actuator lengths to pressures. Equation (24) allows control of trunk shape via the joystick for whole-arm grasping algorithms [10] or, using (29), end-point control using the Jacobian.

The arms prove quite capable in practice. Fig. 20 illustrates OctArm, mounted on the Foster–Miller Talon mobile platform, performing a simple grasping task. Extensive experiments pictured in Fig. 21 were carried out to examine the abilities of the manipulator. Use of the robot based on the algorithms discussed in this paper enabled biologically inspired behavior, such as a prey strike, whole-arm grasping over a wide range of object sizes, tunnel observation using a tip-mounted camera, and teleoperation using cameras mounted at the trunk base.

Prior to the analysis and algorithm development reported in this paper, the two hardware platforms were operated in a section-by-section fashion, with the operator moving only one section of the robot at any time. This proved to be a slow, labo-

rious, and often unsuccessful approach. Implementation of the approach described in this paper transformed the operation of the hardware, enabling successful operation over a wide range of missions, as shown in Figs. 20 and 21. Additionally, the ability to directly match the underlying models to the hardware constraints, particularly the constant-curvature sections, via relatively simple models has allowed us to adapt easily to other effects of strong practical importance, such as actuator travel limits. This will be reported on in future papers.

VII. Conclusion

In this paper, we have presented a new and complete approach to kinematic development for continuum robots. The approach extends previous work by modeling extension, as well as bending, of sections of the arm, and by presenting a complete, correct solution for orientation, as well as position. The approach allows for simple adaptation to a wide class of existing designs, through the inclusion of a "shape-to-actuator" module. Novel examples of such modules for pneumatically actuated and tendon-based sections are described. The resulting Jacobians are relatively simple and amenable to computation in real time, enabling real-time kinematic control at the velocity level. Implementations on two spatial continuum robots are summarized.

References

[1] T. Aoki, A. Ochiai, and S. Hirose, "Study on slime robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, New Orleans, LA, Apr. 2004, pp. 2808–2813.

[2] R. Bishop, "There is more than one way to frame a curve," *Amer. Math. Monthly*, vol. 82, no. 3, pp. 246–251, Mar. 1975.

[3] R. Buckingham, "Snake arm robots," *Ind. Robot: An Int. J.*, vol. 29, no. 3, pp. 242–245, 2002.

[4] J. Casper and R. R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center," *IEEE Trans. Syst., Man, Cybern.*, vol. 33, no. 3, pp. 367–385, Jun. 2003.

[5] G. Chen, M. T. Pham, T. Redarce, C. Prelle, and F. Lamarquem, "Design and modeling of a micro-robotic manipulator for colonoscopy," in *Proc. 5th Int. Workshop Res. Educ. Mechatron.*, Annecy, France, Jun. 2005, pp. 109–114.

[6] G. S. Chirikjian, "Design and analysis of some nonanthropomorphic, biologically inspired robots: An overview," *J. Robot. Syst.*, vol. 18, no. 12, pp. 701–713, 2001.

[7] ——, "Hyper-redundant manipulator dynamics: A continuum approximation," *Adv. Robot.*, vol. 9, no. 3, pp. 217–243, 1995.

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

[9] R. Cieslak and A. Morecki, "Elephant trunk type elastic manipulator – A tool for bulk and liquid type materials transportation," *Robotica*, vol. 17, pp. 11–16, 1999.

[10] M. Cscenstis, B. A. Jones, W. McMahan, V. Iyengar, and I. D. Walker, "User interfaces for continuum robot arms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, AB, Canada, Aug. 2005, pp. 3011–3018.

[11] 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.

[12] I. A. Gravagne and I. D. Walker, "On the kinematics of remotely-actuated continuum robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, 2000, pp. 2544–2550.

[13] ——, "Large deflection dynamics and control for planar continuum robots," *IEEE/ASME Trans. Mechatron.*, vol. 8, no. 2, pp. 299–307, Jun. 2003.

[14] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *J. Robot. Syst.*, vol. 20, no. 2, pp. 45–63, Feb. 2003.

[15] S. Hirose, *Biologically Inspired Robots*. Oxford, U.K.: Oxford Univ. Press, 1993, pp. 147–155.

[16] G. Immega and K. Antonelli, "The KSI tentacle manipulator," in *Proc. IEEE Conf. Robot. Autom.*, vol. 3, Nagoya, Japan, May 1995, pp. 3149–3154.

[17] M. Ivanescu, "A variable length tentacle manipulator control system," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 2005, pp. 3285–3290.

[18] B. A. Jones, W. McMahan, and I. D. Walker, "Design and analysis of a novel pneumatic manipulator," in *Proc. 3rd IFAC Symp. Mechatron. Syst.*, Sydney, Australia, Sep. 2004, pp. 745–750.

[19] W. McMahan, B. A. Jones, I. Walker, V. Chitrakaran, A. Seshadri, and D. Dawson, "Robotic manipulators inspired by cephalopod limbs," in *Proc. CDEN Des. Conf.*, Montreal, QC, Canada, 2004, pp. 1–10.

[20] H. Mochiyama and H. Kobayashi, "The shape Jacobian of a manipulator with hyper degrees of freedom," in *Proc. IEEE Int. Conf. Robot. Autom.*, Detroit, MI, 1999, pp. 2837–2842.

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

[22] H. Ohno and S. Hirose, "Design of slim slime robot and its gait of locomotion," in *Proc. IEEE/RSJ Int. Conf. Intell. Syst.*, vol. 2, Maui, HI, Oct. 2001, pp. 707–715.

[23] M. B. Pritts and C. D. Rahn, "Design of an artificial muscle continuum robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 5, New Orleans, LA, Apr. 2004, pp. 4742–4746.

[24] G. Robinson and J. B. C. Davies, "Continuum robots—A state of the art," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 4, Detroit, MI, May 1999, pp. 2849–2854.

[25] N. Simaan, "Snake-like units using flexible backbones and actuation redundancy for enhanced miniaturization," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 2005, pp. 3023–3028.

[26] N. Simaan, R. Taylor, and P. Flint, "A dexterous system for laryngeal surgery," in *Proc. IEEE Int. Conf. Robot. Autom.*, New Orleans, LA, Apr. 2004, pp. 351–357.

[27] K. Suzumori, S. Iikura, and H. Tanaka, "Development of flexible microactuator and its application to robotic mechanisms," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, Sacramento, CA, Apr. 1991, pp. 1622–1627.

[28] Temple Allen Industries.. [Online]. Available: http://www.templeallen.com/

[29] H. Tsukagoshi, A. Kitagawa, and M. Segawa, "Active hose: An artificial elephant's nose with maneuverability for rescue operation," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, Seoul, Korea, May 2001, pp. 2454–2459.

[30] J. F. Wilson, D. Li, Z. Chen, and R. T. George, Jr., "Flexible robot manipulators and grippers: Relatives of elephant trunks and squid tentacles," in *Proc. Robots Biol. Syst.: Toward a New Bionics?*, 1993, pp. 474–479.

[31] J. F. Wilson and J. M. Snyder, "The elastica with end-load flip-over," *J. Appl. Mech.*, vol. 55, no. 4, pp. 845–848, Dec. 1988.

[32] A. Wolf, H. B. Brown, R. Casciola, A. Costa, M. Schwerin, E. Shamas, and H. Choset, "A mobile hyper redundant mechanism for search and rescue tasks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 3, Taipei, Taiwan, R.O.C., May 2003, pp. 2889–2895.

[33] C. Wu, "A kinematic CAD tool for the design and control of a robot manipulator," *Int. J. Robot. Res.*, vol. 3, no. 1, pp. 58–67, 1984.

Bryan A. Jones (S'00–M'05) received the B.S.E.E. and M.S. degrees in electrical engineering from Rice University, Houston, TX, in 1995 and 2002, respectively, and the Ph.D. degree in electrical engineering from Clemson University, Clemson, SC, in 2005.

From 1996 to 2000, he was a Hardware Design Engineer for Compaq, specializing in board layout for high-availability RAID controllers. He is currently an Assistant Professor at Mississippi State University, Mississippi State, MS. His research interests include robotics, real-time control-system implementation, rapid prototyping for real-time systems, and modeling and analysis of mechatronic systems.

Ian D. Walker (S'84–M'85–SM'02–F'06) received the B.Sc. degree in mathematics from the University of Hull, Hull, U.K., and the M.S. and Ph.D. degrees in electrical engineering from the University of Texas at Austin, in 1983, 1985, and 1989, respectively.

He is a Professor of Electrical and Computer Engineering at Clemson University, Clemson, SC. His research interests are in robotics, particularly kinematically redundant robots, robot reliability and fault detection, and biologically inspired robots.