

Practical Kinematics for Real-Time Implementation of Continuum Robots

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

Abstract—This paper introduces three algorithms which are essential for the practical, real-time implementation of continuum robots. Continuum robots lack the joints and links which compose traditional and high-degree-of-freedom robots, instead relying on finite actuation mechanisms to shape the robot into a smooth curve. Actuator length limits shape the configuration or joint space of continuum manipulators, introducing couplings analyzed in this paper which must be understood to make effective use of continuum robot hardware. Based on the new understanding of the configuration space uncovered, this paper then derives the workspace of continuum robots when constrained by actuator length limits. Finally, a tangle/untangle algorithm correctly computes the shape of the distal segments of multisection tendon-actuated continuum robots. These contributions are essential for effective use of a wide range of continuum robots, and have been implemented and tested on two different types of continuum robots. Results and insight gained from this implementation are presented.

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

I. INTRODUCTION

INSPIRED by elephant trunks, mammalian tongues, and octopus arms, appendages which possess remarkable manipulation abilities without a typical skeletal structure, continuum robots seek to mimic the astonishing abilities of these organisms through designs which lack the rigid links which compose traditional robots. Just as an octopus explores under rocks and in dark holes by feeling with its arms, so also continuum trunks could be used in urban search and rescue operations in the unstructured, confined environments of collapsed buildings. By encircling objects of widely varying sizes and *a priori* unknown shapes, continuum robots may use whole-arm grasping to manipulate objects too large, too fragile, or too unstructured for traditional parallel-jaw grippers in a manner similar to the abilities evidenced by elephant trunks, as shown in Fig. 1.

However, in spite of the large number of experimental and commercial continuum robots constructed [1], development



Fig. 1. In (a), the OctArm continuum manipulator grasps and lifts a traffic cone. As shown here, it replaces the end-effector of a Talon robot. In (b), the two-section Air-Octor robot is shown.

of kinematics to model and operate these robots repeatedly and reliably under real-time control remains an open research area. Continuum robots are typically kinematically redundant; the lack of discrete joints additionally renders their kinematic analysis more complex than rigid-link robots. Hirose and colleagues developed kinematics in 2-D for a number of snake-like robots constructed in their lab by introducing the serpenoid curve which closely matches the shape of a snake [2]. In contrast, the modal approach followed by [3] involves linking the shape of a general mathematical curve to a high-degree-of-freedom (HDOF) robot. This modal decomposition then reduces the number of DOFs necessary to control a robot whose macroscopic structure is captured by this curve, such as the 30-DOF robot discussed in their work. However, few continuum robots well match the proposed curve, limiting the applicability of the approach. By examining the underlying mechanics of a continuum robot, which shows that these types of structures bend with constant curvature when unloaded [4], newer approaches [5]–[7] followed Hirose's method of finding a curve, an arc of constant curvature, which more directly models the robot's shape. Our extensions to the three-actuator case [8]–[10] broadened the applicability of this theory, representing a viable foundation for modeling and real-time control of continuum robots.

However, development of a theoretical foundation for continuum kinematics must be accompanied by solutions amenable to real-time implementation of the resulting kinematics. In practice, actuator length limits, though not present or accounted for in the previously published kinematics, shape the achievable configuration space of the robot in unexpected ways, due to the coupling present in continuum robots. To achieve full use of these robots, these couplings must be understood and exploited. This paper introduces a new and complete practical solution for

Manuscript received November 19, 2005; revised June 21, 2006. This paper was recommended for publication by Associate Editor S. Ma and Editor K. Lynch upon evaluation of the reviewers' comments. This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) under Contract N66001-C-8043. Color versions of Figs. 1, 4–11, and 13–15 are available at <http://ieeexplore.org>.

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).

Digital Object Identifier 10.1109/TRO.2006.886268

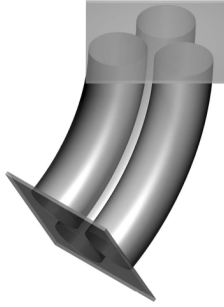


Fig. 2. Intrinsic actuation scheme. Three actuators are connected in parallel between two plates. Differences in length cause bending of the structure.

the inclusion of actuator length limits for continuum robot kinematics.

The workspace shape for continuum robots is likewise made significantly more complex, due to configuration space couplings resulting from actuator length limits. This hinders use of the robot because any object to be manipulated must lie within the robot's workspace. This paper presents new results and understanding in continuum robot workspace calculation. Finally, the coupling inherent in the many tendon-driven designs requires a "tangle/untangle" step to correctly compute the position of a distal (farthest from the trunk base) trunk section after cables which control its shape have passed through a number of proximal sections, altering the distal cable's length due to the shape of each of the proximal sections.

This set of essential implementation barriers must be breached for proper and effective operation of a continuum robot. Indeed, almost all continuum designs to date are affected by these problems. These designs can be divided into the three categories of continuum robots proposed in [1] of intrinsic, extrinsic, and hybrid actuation strategies. Intrinsically actuated continuum robots, typically composed of three actuators affixed to a plate at each end which extend longitudinally, produce bending due to differences in the actuator lengths, as shown in Fig. 2. Therefore, they are affected by actuator length limits due to minimum and maximum extension of the three component actuators, which requires additional analysis to determine the resulting configuration and workspace of the robots. Robots in this category, which are partially drawn from [1], include the flexible micro-actuator [11], the AMADEUS hand [12], pizeohydraulic systems [13], the active hose [14], the EDORA colonoscope [15], the slim slime robot [16], shaped-memory alloy tentacles [17], and McKibben-based trunks [18]. Novel actuation schemes, such as electrorheological fluid-based manipulators suggested by [19] and electrostrictive polymer artificial muscles suggested by [20], are likewise affected by maximum and minimum length limits.

Both extrinsic and hybrid actuation strategies typically use three tendons, such as steel cables or flexible plastic rods, to shape the robot as diagrammed in Fig. 3. Robots in this category include OCRobotics trunks [21], the Air-Octor trunk [22], the KSI tentacle [23], and the Elastor and ACM IV manipulators [2]. Because the tendons which define the shape of the distal trunk sections pass through proximal sections, their lengths must be suitably corrected based on proximal section shape.

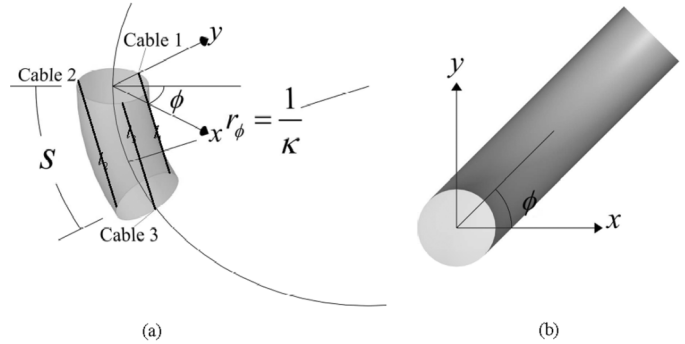


Fig. 3. Extrinsic and hybrid actuation schemes. Three tendons, labeled cable 1 to 3, of lengths l_{1-3} , shape the robot as shown from the side in (a). In the center of the trunk lies a passive backbone in the extrinsic case or a flexible actuator such as pneumatically pressurized tube in the hybrid case. Trunk shape is defined in terms of a trunk length s , amount of curvature κ , and direction of curvature ϕ . A top view in (b) more clearly illustrates the definition of ϕ .

The cable tangling/untangling algorithms introduced in this paper provide the first complete solution for performing this function. Although the algorithm is solved using three-tendon kinematics, it can be easily modified for four-tendon kinematics such as those used in the tensor arm manipulator [24], the SPINE manipulator [25], an elephant-trunk-type elastic manipulator [26], the Clemson elephant trunk [5], and the Clemson tentacle [4]. Previous work in this area began with [5], which briefly describes a method for untangling a simpler case of a four-tendon robot with all tendons placed at 90° intervals. However, that method forces the awkward placement of actuators vertically above one another and handles only the four-tendon case, neither of which can be used with newer designs.

Solution of these three problems (of properly handling a coupled configuration space, workspace derivation which incorporates actuator length limits, and cable tangling/untangling) is, therefore, essential for practical and effective use of continuum robots. The following sections of this paper address these problems in order, present detailed solutions to them, then discuss the practical results and improvements obtained in the operation of two different continuum robots using these solutions. The analysis begins by examining the effects actuator length limits have on the configuration or joint space of the robot.

II. EFFECTS OF ACTUATOR LENGTH LIMITS

A strongly desired feature of a trunk is its ability to manipulate objects within its workspace by grasping them. Although an end-effector can be mounted on the tip of the robot, the trunk itself forms a versatile and powerful manipulator by wrapping itself around a desired item, much in the way a boa constrictor grasps its prey. This form of manipulation, termed whole-arm manipulation [27], relies on a trunk with sufficient flexibility to curl around on itself to form an enveloping grasp. A single-section grasp can be achieved by bending the trunk through a wrap angle of 360° ; a two-section grasp, by bending each section through 180° ; and, in general, an n -section grasp by bending through $360^\circ/n$ degrees. Failing to achieve the curvature necessary to bend through the required number of degrees prevents the trunk from enveloping, then grasping, objects in this fashion.

A. Geometrical Interpretation of Actuator Length Limits

Like many continuum manipulators, the three pneumatic actuators powering the OctArm trunk [18]¹ shown in Fig. 1 are placed in parallel between two end plates as diagrammed in Fig. 2, forming a parallel manipulator whose geometry is determined by actuator lengths l_{1-3} . Given a minimum and maximum length $[l_{\min}, l_{\max}]$ for each of the three actuators due to their physical design, the configuration space accessible under these actuator length limits is, therefore, a cube from l_{\min} to l_{\max} along the l_1, l_2 , and l_3 axes. However, this configuration space does not obviously determine the whole-arm manipulation capabilities of the trunk in terms of the maximum achievable wrap angle θ or minimum achievable bend radius $r = 1/\kappa$.

Equations derived in [8] provide a mapping from pneumatic actuator lengths l_{1-3} to a given trunk length s , curvature κ , and direction of curvature ϕ , as defined in Fig. 3, as

$$\begin{bmatrix} s \\ \kappa \\ \phi \end{bmatrix} = \begin{bmatrix} \frac{l_1 + l_2 + l_3}{3} \\ 2 \frac{\sqrt{g}}{d(l_1 + l_2 + l_3)} \\ \tan^{-1} \left(\frac{\sqrt{3} l_3 + l_2 - 2l_1}{l_2 - l_3} \right) \end{bmatrix} \quad (1)$$

where $g = l_1^2 + l_2^2 + l_3^2 - l_1 l_2 - l_2 l_3 - l_1 l_3$ and d is the radius of the trunk.

These equations effectively convert the parallel mechanism of the trunk into a serial manipulator, which can then be analyzed through traditional processes involving use of the Denavit–Hartenburg procedure [8]. This serial parameterization of the trunk by (s, κ, ϕ) provides valuable information on the capabilities of the trunk by directly computing the inverse of bend radius κ , and providing the angle of wrap as $\theta = s\kappa = 2\sqrt{g}/3d$.

As proven below, this serial parameterization represents a straightforward geometrical transformation of the configuration space in terms of actuator lengths, relating the actuator length limits to the resulting limits on the direction of curvature κ and wrap angle θ . We will find geometric interpretations of s, κ , and ϕ in terms of the box representing joint (actuator) lengths and their physical limits, as illustrated in Fig. 4. Specifically, projecting the point $\underline{l} = [l_1 \ l_2 \ l_3]^T$ onto the line $l_1 = l_2 = l_3$ by computing

$$\frac{\underline{l} \cdot [1 \ 1 \ 1]^T}{\|[1 \ 1 \ 1]^T\|} = \frac{\sqrt{3}(l_1 + l_2 + l_3)}{3} = s\sqrt{3}$$

where \cdot is the dot product, yields a trunk length s , which is proportional to the length of that projection along the line. The distance from that projection to the line $l_1 = l_2 = l_3$ to \underline{l} is

$$\left\| \underline{l} - \frac{\underline{l} \cdot [1 \ 1 \ 1]^T}{\|[1 \ 1 \ 1]^T\|^2} [1 \ 1 \ 1]^T \right\| = \frac{\sqrt{6g}}{3} = \frac{d\sqrt{6}}{2} \theta \quad (2)$$

which is proportional to the wrap angle θ . The direction of curvature ϕ is defined as the angle along which the distance θ lies. Referencing the definition of ϕ in (1), this angle is measured from a y axis defined by the vector $[-2\sqrt{3} \ \sqrt{3} \ \sqrt{3}]^T$ and an x axis of $[0 \ 3 \ -3]^T$.

¹We adopt the case of the continuum robots introduced in [8] and [18] as examples in this paper. However, the issues apply across the spectrum of existing continuum robot designs, as noted in the introduction.

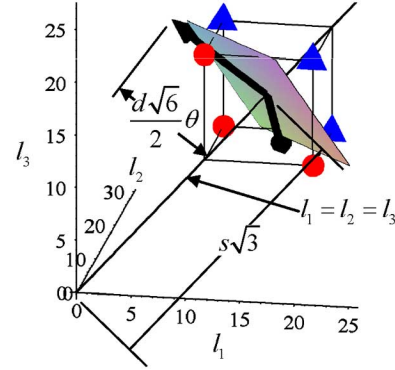


Fig. 4. Geometrical interpretation of ϕ, θ , and s in configuration space. Any point in configuration space can be expressed in terms of a length $s\sqrt{3}$ along the line $l_1 = l_2 = l_3$, the distance $\theta d\sqrt{6}/2$ from this line in the plane perpendicular to the line, and an angle ϕ (not shown, but measured about the axes defined by the two arrows) in that plane.

This geometrical interpretation provides immediate insight into the effects of actuator length limits on trunk capabilities, as plotted in Fig. 4. A large wrap angle θ is produced by escaping as far as possible from the line $l_1 = l_2 = l_3$, where the wrap angle is zero. While two cube corners $(l_{\max}, l_{\max}, l_{\max})$ and $(l_{\min}, l_{\min}, l_{\min})$ lie at the zero wrap angle, the remaining six represent the maximum possible distance from the line $l_1 = l_2 = l_3$, and therefore, the maximum achievable wrap angle, while still respecting actuator length limits by remaining inside the cube. However, these wrap angles can only be achieved at specific values of ϕ which point toward the cube corners. Because curvature $\kappa = \theta/s$, the three cube corners $(l_{\min}, l_{\min}, l_{\max})$ with a small s value provide a smaller radius of curvature $1/\kappa$ and a tighter grasp than the three cube corners $(l_{\min}, l_{\max}, l_{\max})$ with a larger s . Specifically, evaluating κ given in (1) at $(l_{\min}, l_{\min}, l_{\max})$ gives

$$\kappa_{\max} = 2 \frac{l_{\max} - l_{\min}}{d(l_{\max} + 2l_{\min})}. \quad (3)$$

A second local maximum curvature is likewise obtained by choosing one actuator to be of length l_{\min} , and the remaining two actuators to be of length l_{\max} , producing a curvature of

$$\kappa_{2\max} = 2 \frac{l_{\max} - l_{\min}}{d(2l_{\max} + l_{\min})}. \quad (4)$$

Examining the definition of trunk curvature κ given in (1) provides insight into proving that the maximum possible curvature κ given in (3) is achieved at $(l_{\min}, l_{\min}, l_{\max})$. As expected, κ goes to zero when the length of all three actuators is equal, revealing the line $l_1 = l_2 = l_3$. Assuming these lengths are real and nonnegative, minimization detailed in the Appendix proves that this line is indeed a global minima for κ , and that no other extrema exist.

Next, consider the isosurface of this function which includes all points for which the value of κ does not change. First, as proven in the Appendix, changing the direction of curvature ϕ leaves κ unchanged. In terms of actuator lengths, a rotation of the point $\underline{l} = [l_1 \ l_2 \ l_3]^T$ about the line $l_1 = l_2 = l_3$ by ϕ , using the standard axis/angle rotation matrix such as that given in [28]

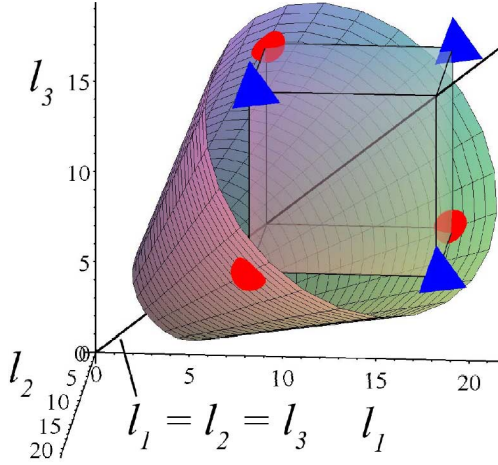


Fig. 5. Isosurface for which the value of κ is invariant. The volume of the cube contains the accessible configuration space defined by $l_{\min} = 25$ cm and $l_{\max} = 35$ cm. The thick line shown is $l_1 = l_2 = l_3$, at which both curvature κ and angle of wrap θ are zero. The three spheres indicate the three corners of the cube which provide maximum curvature κ_{\max} , as defined in (3), while the triangles illustrate $\kappa_{2\max}$ from (4). Maximum curvature at a given direction of curvature $\kappa_{\max}(\phi)$ in (11) is obtained by choosing a point on the lines which connect these six points, chosen based on the given ϕ .

to compute $l'_i = \mathbf{R}_{\phi, (1,1,1)} l_i$, produces the equation shown at the bottom of the page, where l'_i gives a new value for the i th actuator without changing κ , providing a method to compute one dimension of the isosurface. Second, recalling that $\kappa = \theta/s$, increasing both θ and s leaves κ unchanged. Computations given in the Appendix show that the second dimension of the isosurface is determined by $l' = cl$, where c is a positive constant.

Therefore, the resulting isosurface is a funnel, as shown in Fig. 5, produced by a rotation of ϕ about the line $l_1 = l_2 = l_3$ and a scaling of c of all three actuator values. Points inside the funnel result from lesser curvature due to moving closer to the zero curvature line $l_1 = l_2 = l_3$, while points outside the funnel have greater curvature. Placing the funnel so that it touches only the three bottom corners of the cube demonstrates that all other points inside the funnel, which includes the remainder of the cube, consists of lower curvature points. No points on the cube fall outside the funnel, so no higher curvature points exist within the given actuator length limits. This proves that these three corners, located at $(l_{\min}, l_{\min}, l_{\max})$, are the maximum possible curvature achievable by the trunk under actuator length limits.

B. Practical Application Under Actuator Length Limits

As shown in the previous section, this geometrical understanding of the cube in the actuator or configuration space (l_1, l_2, l_3) which encompasses all valid configurations of a trunk section constrained by actuator length limits can be alternately parameterized by the more physically meaningful coordinate system of trunk length s , curvature κ , and direction

of curvature ϕ , in which $s\sqrt{3}$ is the distance along the line $l_1 = l_2 = l_3$, $\theta d\sqrt{6}/2$ is the perpendicular distance from the line, and ϕ is the angle about that line. The following section applies this understanding to determine the effects of these actuator length limits on s , κ , and ϕ .

To find the maximum achievable curvature and angle of wrap at some desired direction of curvature ϕ , the geometry of configuration space dictates a walk along the six cube edges which connect the six cube corners of maximum wrap angle. Armed with this insight, consider the equations developed in [9] which relate s , κ , and ϕ to actuator lengths l_{1-3}

$$\begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} = \begin{bmatrix} s(1 - \kappa d \sin \phi) \\ s(1 + \kappa d \sin(\pi/3 + \phi)) \\ s(1 - \kappa d \cos(\pi/6 + \phi)) \end{bmatrix}.$$

Therefore, actuator length can be rewritten as

$$l_i = s(1 + \kappa df_i(\phi)) = s + \theta df_i(\phi) \quad (5)$$

using the identity $\theta = \kappa s$, where $f(\phi) = [-\sin \phi \sin(\pi/3 + \phi) - \cos(\pi/6 + \phi)]^T$. Guaranteeing that one actuator length remains in the range of valid lengths $[l_{\min}, l_{\max}]$, while fixing the other two actuator lengths at l_{\max} and l_{\min} to perform the cube edge walk, requires a careful mapping of these lengths to particular actuators. Specifically, choosing the actuator for which $f_{\max}(\phi) = \max_{i=1,2,3} f_i(\phi)$ to have length l_{\max} , and choosing the actuator for which $f_{\min}(\phi) = \min_{i=1,2,3} f_i(\phi)$ to have length l_{\min} , demonstrates that (5) can be bounded as $l_{\min} \leq l_{\text{mid}} = s(1 + \kappa df_{\text{mid}}) \leq l_{\max}$, guaranteeing that the remaining actuator's length l_{mid} is valid.

The conditions necessary to achieve the maximum curvature $\kappa_{\max}(\phi)$ in desired direction ϕ can then be determined by solving (5) either for curvature κ to produce

$$\kappa = \frac{l_i - s}{s df_i} \quad (6)$$

or for angle of wrap θ , yielding

$$\theta = \frac{l_i - s}{df_i}. \quad (7)$$

Evaluating (6) and (7) at the points (l_{\max}, f_{\max}) and (l_{\min}, f_{\min}) yields

$$\kappa_{\max}(\phi) = \frac{l_{\max} - s}{s df_{\max}} = \frac{l_{\min} - s}{s df_{\min}} \quad (8)$$

$$\theta_{\max}(\phi) = \frac{l_{\max} - s}{df_{\max}} = \frac{l_{\min} - s}{df_{\min}} \quad (9)$$

$$\begin{bmatrix} l'_1 \\ l'_2 \\ l'_3 \end{bmatrix} = 3 \begin{bmatrix} (2l_1 - l_2 - l_3) \cos \phi - \sqrt{3}(l_2 - l_3) \sin \phi + (l_1 + l_2 + l_3) \\ (2l_2 - l_1 - l_3) \cos \phi - \sqrt{3}(l_1 - l_3) \sin \phi + (l_1 + l_2 + l_3) \\ (2l_3 - l_1 - l_2) \cos \phi - \sqrt{3}(l_1 - l_2) \sin \phi + (l_1 + l_2 + l_3) \end{bmatrix}$$

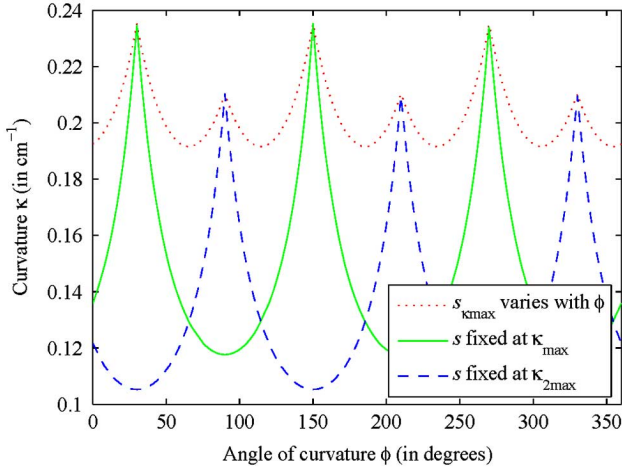


Fig. 6. Trunk maximum curvature $\kappa_{\max}(\phi)$ determined by evaluating (11), shown in a dotted line as the direction of curvature ϕ varies. The dashed line and solid line illustrate the detrimental effects of fixing trunk length s , determined by evaluating (13). Trunk minimum actuator length was $l_{\min} = 25$ cm and maximum actuator length was $l_{\max} = 35$ cm for a trunk of radius $d = 1$ cm.

detailing how this maximum curvature at ϕ can only be achieved at a specific trunk length $s_{\kappa_{\max}}(\phi)$ obtained by solving either equation for s as

$$s_{\kappa_{\max}}(\phi) = \frac{f_{\max}l_{\min} - f_{\min}l_{\max}}{f_{\max} - f_{\min}}. \quad (10)$$

Substituting this result into (8) and (9) reveals the maximum curvature and wrap achievable in direction ϕ , given that the trunk length specified by (10) is

$$\kappa_{\max}(\phi) = \frac{l_{\max} - l_{\min}}{d(f_{\max}l_{\min} - f_{\min}l_{\max})} \quad (11)$$

$$\theta_{\max}(\phi) = \frac{l_{\max} - l_{\min}}{d(f_{\max} - f_{\min})}. \quad (12)$$

Verifying this derivation by substituting (10) and (11) or (12) into (5) proves that one actuator is fixed at l_{\max} , a second actuator at l_{\min} , and the third actuator's length varies between the two while performing this cube edge walk, along which maxima of the function occur as discussed in the previous section. Therefore, the absolute maximum curvature determined in (3) occurs at $\phi = \{30^\circ, 150^\circ, 270^\circ\}$ with two actuators at l_{\min} and one at l_{\max} , while the second local maximum given in (4) with two actuators at l_{\max} and one at l_{\min} occurs at $\phi = \{90^\circ, 210^\circ, 330^\circ\}$. This approach achieves the maximum possible curvature κ in the desired direction of curvature ϕ , as illustrated by Fig. 6 and the maximum possible angle of wrap θ at the desired ϕ . These figures also illustrate how essential correct trunk length s is to producing maximum curvature.

This delicate interaction in which trunk length s affects achievable curvature κ at a given direction of curvature ϕ is more completely understood when examining the configuration space of a trunk. The maximal configuration space, expressed as the maximum achievable curvature κ given a desired direction of curvature ϕ and fixed trunk length s as plotted in Fig. 7, can be determined by examining (6), noting that only one free variable, l_i , remains. As before, choosing $l_i = l_{\max}$ when $f_i = f_{\max}$ or $l_i = l_{\min}$ when $f_i = f_{\min}$ produces

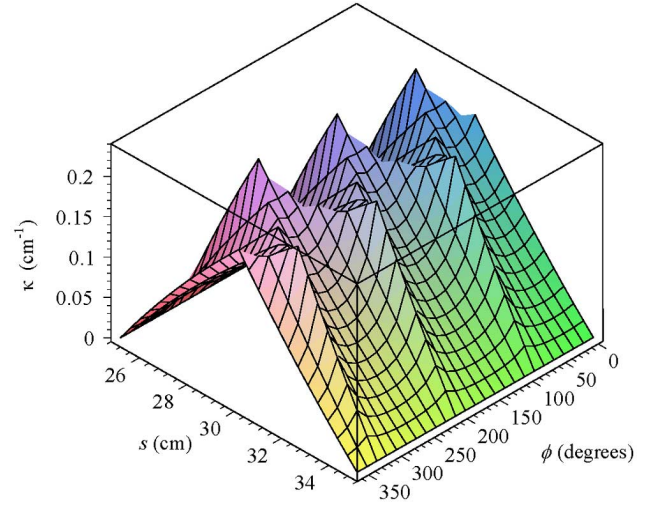


Fig. 7. Maximum achievable trunk curvature $\kappa_{\max}(s, \phi)$ in a given direction of curvature ϕ and trunk length s computed by applying (13). Results were obtained for a trunk where maximum and minimum actuator lengths are $l_{\max} = 35$ cm, $l_{\min} = 25$ cm, and the trunk radius is $d = 1$ cm.

the maximum cable length difference, given the other two cable lengths are fixed by the given trunk length s and direction of curvature ϕ . The two possible solutions are therefore $\{(l_{\max} - s)/(sdf_{\max}), (l_{\min} - s)/(sdf_{\min})\}$.

However, only the solution which respects the relationship between trunk length and actuator length limits can be used; for example, choosing $s = l_{\max}$ requires choosing $l_i = l_{\max}$. The valid region for each of the two solutions can be determined by observing that an achievable curvature is limited by the maximum possible curvature determined in (11), so that

$$\kappa_{\max}(\phi) \geq \begin{cases} \frac{l_{\max} - s}{sdf_{\max}} \\ \frac{l_{\min} - s}{sdf_{\min}} \end{cases}.$$

Solving for s by substituting in (11) produces two inequalities similar to (10), $s \leq (f_{\max}l_{\min} - f_{\min}l_{\max})/(f_{\max} - f_{\min})$ when choosing l_{\max} and f_{\max} or $s \geq (f_{\max}l_{\min} - f_{\min}l_{\max})/(f_{\max} - f_{\min})$ when choosing l_{\min} and f_{\min} . Summarizing, the maximum achievable curvature $\kappa_{\max}(\phi, s)$ at a specified trunk length s and direction of curvature ϕ is

$$\kappa_{\max}(s, \phi) = \begin{cases} \frac{l_{\max} - s}{sdf_{\max}} & \text{when } s \geq \frac{f_{\max}l_{\min} - f_{\min}l_{\max}}{f_{\max} - f_{\min}} \\ \frac{l_{\min} - s}{sdf_{\min}} & \text{when } s \leq \frac{f_{\max}l_{\min} - f_{\min}l_{\max}}{f_{\max} - f_{\min}} \end{cases}. \quad (13)$$

Similar results for $\theta_{\max}(s, \phi)$ can be obtained by repeating the process with (7) for the limits given in (12).

Plotting this function produces the graph shown in Fig. 7. As expected, the plot contains six peaks, three maxima at $\phi = \{30^\circ, 150^\circ, 270^\circ\}$, and three secondary local maxima at $\phi = \{90^\circ, 210^\circ, 330^\circ\}$. Viewed from the side, so that ϕ is visible along the bottom axis and κ extends vertically upwards, produces a plot identical to the $s_{\kappa_{\max}}$ line in Fig. 6. The peak of the curve shows the maximum possible curvature in a given direction ϕ , also plotted as the dotted line of Fig. 6, as trunk length is varied by evaluating (10). The sharpness of the peak indicates that maximum possible curvature κ in a particular direction ϕ can only be obtained by a precise choice of s ; varying s only a little causes a dramatic drop in the achievable curvature κ .

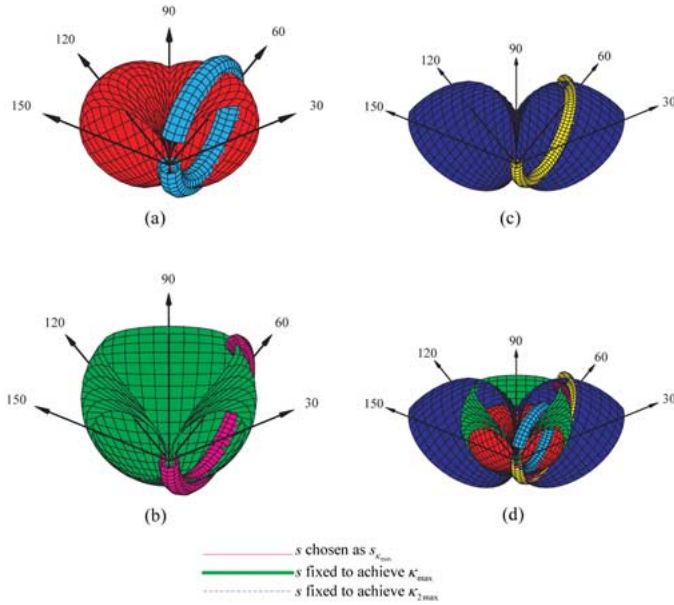


Fig. 8. Plot of the surface formed by rotating the trunk, drawn at $\phi = 60^\circ$ from $\phi = [30^\circ, 150^\circ]$. Plot (a) was formed by choosing $s = s_{\kappa_{\max}}(\phi)$ from (10) and $\kappa = \kappa_{\max}(\phi)$ from (11). Plot (b) was formed by fixing s to achieve κ_{\max} as defined in (3), and plot (c) by fixing s to achieve $\kappa_{2\max}$ as defined by (4). The trunk bends into a circle of varying radius based on actuator length limits of $l_{\min} = 25$ cm and $l_{\max} = 35$ cm. Trunk diameter is $d = 1$ cm.

Fig. 8 illustrates the results shown in Fig. 7 from another perspective, showing the effect that trunk length s has on curvature κ . The plots were formed by first bending a single-section trunk in the direction $\phi = 60^\circ$ as much as possible, given the trunk length chosen as indicated in the legend for a single-section trunk, then by sweeping the trunk from $\phi = [30^\circ, 150^\circ]$ to produce a surface. The trunk bends downward from the origin of the coordinate system, located at the base of the arrows that label ϕ . Plot (a) shows that, by choosing s and κ from (10) and (11), the trunk achieves almost a full 360° of wrap θ , forming a circle, regardless of the direction ϕ chosen. The radius of the circle changes as trunk length changes, due to actuator length limits. Therefore, these equations enable whole-arm grasping in any direction by appropriately varying trunk length to form an encircling grasp. In contrast, simpler approaches form an encircling grasp only at specific points. Plot (b) illustrates the effects of fixing trunk length to achieve the maximum curvature defined by (3). In this case, the trunk forms an encircling grasp only at $\phi = \{30^\circ, 150^\circ, 270^\circ\}$, as discussed earlier. The first two points of $\phi = \{30^\circ, 150^\circ\}$ are shown in the figure. Likewise, plot (c) illustrates the effects of fixing trunk curvature to achieve the second local maximum given in (4); an encircling grasp can only be formed at $\phi = \{90^\circ, 210^\circ, 330^\circ\}$, of which the point $\phi = 90^\circ$ is shown in the plot. Plot (d) combines all three, showing that the tightest curvature is achieved in (a), second tightest in (b), and least tight in (c).

Lab experiments show that manually varying trunk length s to achieve the maximum curvature κ necessary to perform a whole-arm grasp of an object by encircling it with a trunk section is extremely difficult, necessitating the use of automatic control of trunk length. However, a strategy of simply choosing trunk length s by applying (10) results in problematic operation near the origin, when $\kappa \cong 0$, and restricts full usage of

the trunk's capabilities. Moving the trunk a very small amount leaves $\kappa \cong 0$, but produces large changes in ϕ , resulting in large jumps in trunk length. For example, when $\kappa = 0.001 \text{ cm}^{-1}$, moving from $\phi = 30^\circ$ to $\phi = 90^\circ$ results in s changing from $s = 28.3$ cm to $s = 31.7$ cm, a difference of 3.4 cm, which manifests itself as jerky, step-like behavior near the origin. In addition, this approach removes all control of trunk length from the system, thereby reducing the DOFs available. It chooses a single trunk length which can achieve the maximum curvature at ϕ , when a range of valid trunk lengths to achieve the current, rather than the maximum, curvature exist. For example, Fig. 7 shows that when half the maximum possible curvature of $\kappa = 0.1 \text{ cm}^{-1}$ at $\phi = 30^\circ$ is desired, trunk length can range between $s = 26.3$ cm and $s = 31.8$ cm. However, choosing s using (10) fixes $s = 28.3$ cm to lie only on the peak of the figure, an unnecessary restriction.

Therefore, these problems necessitate an expanded approach which determines the range of allowable trunk lengths, given a desired curvature κ and direction of curvature ϕ . The earlier approach of determining maximum curvature at the two points (l_{\max}, f_{\max}) and (l_{\min}, f_{\min}) again suggests a solution. Solving (5) for s produces $s = l_i / (1 + \kappa df_i)$. Evaluating at the maximum and minimum points while noting that s cannot exceed the actuator limits yields

$$s_{\min}(\phi) = \frac{l_{\min}}{1 + \kappa df_{\min}} \geq s \geq \frac{l_{\max}}{1 + \kappa df_{\max}} = s_{\max}(\phi). \quad (14)$$

These expressions respect the physical actuator length limits. Because f_{\min} is always negative, $1 + \kappa df_{\min}$ will be less than 1, producing an $s_{\min} \geq l_{\min}$. Likewise, f_{\max} is always positive, so that $1 + \kappa df_{\max}$ produces an $s_{\max} \leq l_{\max}$. The point $s_{\max} = s_{\min}$ occurs at $s_{\kappa_{\max}}(\phi)$ defined in (10); values for which $s_{\max} < s_{\min}$ indicate unachievable curvatures for any trunk length s . Choosing instead $s = l_i - \theta df_i$ from (5) then repeating the process yields the solution in terms of θ

$$s_{\min}(\phi) = l_{\min} - \theta df_{\min} \geq s \geq l_{\max} - \theta df_{\max} = s_{\max}(\phi).$$

Fig. 9 illustrates these results graphically. As shown in the figure, when $\kappa \cong 0$, almost the full range of trunk lengths is allowable, producing only minor variation in trunk length as the direction of curvature ϕ varies. For example, choosing $\kappa = 0.001 \text{ cm}^{-1}$ for $l_{\min} = 25$ cm and $l_{\max} = 35$ cm results in $s_{\min} = 25.02$ cm and $s_{\max} = 34.97$ cm, providing a wide range of user choices for a trunk length s which achieves the desired curvature of $\kappa = 0.001 \text{ cm}^{-1}$. In addition, the area between the two surfaces in the figure indicates space in which the trunk length s can be modified at will given the desired curvature, providing an additional DOF in specifying the robot's shape, in contrast to fixing s to achieve the maximum curvature even when maximum curvature is not required.

These algorithms and equations find extensive use in continuum robot implementation. When mapping joystick position to curvature, maximum joystick deflection must be mapped to maximum achievable curvature, which can be chosen as the maximum possible curvature given in (3) or the maximum curvature in the current direction of curvature given by (11). To

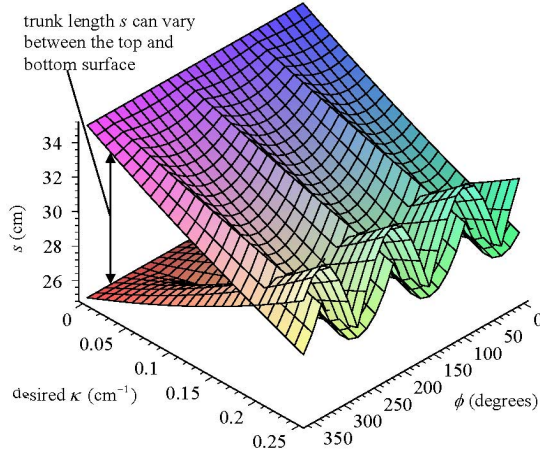


Fig. 9. Plot of s_{\min} and s_{\max} determined by evaluating (14) over ϕ and κ . The two begin to overlap near κ_{\max} , indicating that maximum curvature can only be obtained at a specific trunk length $s_{\kappa_{\max}}(\phi)$. Results were obtained for a trunk where maximum and minimum actuator lengths are $l_{\max} = 35$ cm, $l_{\min} = 25$ cm, and the trunk radius is $d = 1$ cm.

form the tightest possible whole-arm grasp, the direction of curvature should be chosen as $\phi = \{30^\circ, 150^\circ, 270^\circ\}$ as noted in the discussion following (11), while trunk length should be chosen by evaluating (10). General movement of the trunk in which the user chooses the desired amount of curvature κ , direction of curvature ϕ , and trunk length s can be performed by limiting the commanded s to lie between the limits computed by (14). This allows exploration of the entire workspace derived by (13) and plotted in Fig. 7, automatically “tuning” s only when necessary to achieve the desired κ and ϕ .

C. Workspace Geometry Due to Actuator Length Limits

Essential to characterizing and understanding the capability of any robotic manipulator is the derivation of its workspace. Objects cannot be grasped or manipulated if they lie outside the robot’s workspace, causing user confusion and frustration when the robot cannot perform a seemingly simple task which, unknown to the user, lies outside the workspace. Singularities within the workspace limit the robot’s ability to move in all directions, while workspace boundaries can cause infinite computed actuator torque when evaluating the Jacobian near these points. The unusual nature of continuum robots, such as the interaction between achievable curvature and trunk length discussed in the previous section, results in a workspace, shown in Fig. 10, unlike those of traditional robots. Therefore, this workspace must be derived and investigated to make practical use of the robot. The workspace derivation introduced below is, to the best of the authors’ knowledge, unique within the field.

An observation, based on the analysis already detailed in this paper makes workspace derivation of a single-section trunk simpler. Note that the configuration-space or joint-space boundaries have already been derived in the previous section for a single-section robot. Specifically, (13) gives the maximum achievable trunk curvature κ over all directions of curvature ϕ and valid lengths s . Simply computing the homogenous transformation

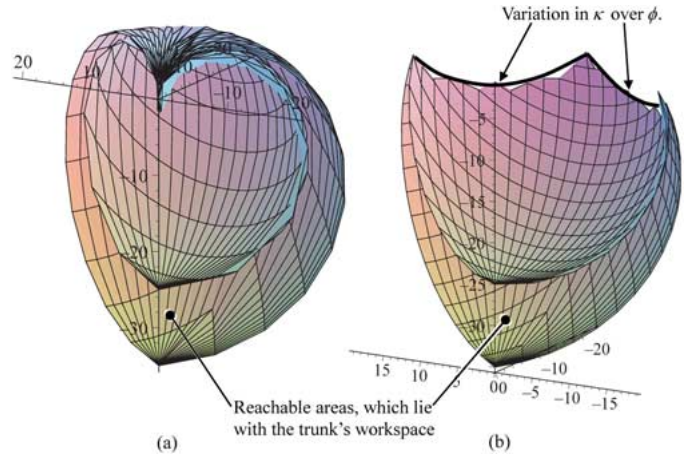


Fig. 10. Workspace of a single-section robot drawn from $\phi = [30^\circ, 150^\circ]$ with trunk radius $d = 1$ cm in (a) and $d = 2.25$ cm in (b). Actuator length limits are 25 and 35 cm in both cases.

matrix \mathbf{A} parameterized by the parameters ϕ , κ , and s as given in [10] allows mapping joint space to the workspace. Therefore, evaluating the translational portion of \mathbf{A}

$$\mathbf{A}_{1...3,4}(\phi, \kappa, s) = \begin{bmatrix} \kappa^{-1} \cos \phi (\cos \kappa s - 1) \\ \kappa^{-1} \sin \phi (\cos \kappa s - 1) \\ \kappa^{-1} \sin \kappa s \end{bmatrix} \quad (15)$$

over the surface illustrated in Fig. 7 and defined by (13) gives the x , y , and z coordinates of the trunk tip as it travels along the workspace boundary.

Fig. 10 illustrates the workspace generated using this method for two cases, one in which a trunk radius of $d = 1$ cm allows a wrap angle θ of approximately 360° , while a larger radius of $d = 2.25$ cm in the second case limits the wrap angle θ to less than 180° .

The reachable workspace lies between the upper and lower surfaces of each plot. To illustrate the workspace, only $1/3$ of the full workspace, from $\phi = [30^\circ, 150^\circ]$, was plotted. As shown in earlier plots, the three-actuator construction of the trunk produces identical results for the other $2/3$ of the robot’s workspace. As expected, a trunk can reach any point on the z axis between its maximum and minimum length. The more the trunk bends, the less the trunk length s can vary until it achieves maximum curvature at a single s value, analytically determined by (10) and (11). The right-hand trunk clearly displays the variation in achievable curvature illustrated by the dotted red line in Fig. 6 at the top of the reachable workspace. The same variation exists in the left-hand trunk, but is not visible, since it is mapped very close to the origin due to the left-hand trunk’s ability to bend through $\sim 360^\circ$.

Analytical derivation of the workspace for a multisection trunk is more difficult, being essentially the 3-D convolution of the individual workspaces. Therefore, a systematic, numerical method of plotting the workspace by exhaustive iteration through all possible joint space poses provides one method to visualize multisection workspaces. This and other methods provide avenues for obtaining multisection results.

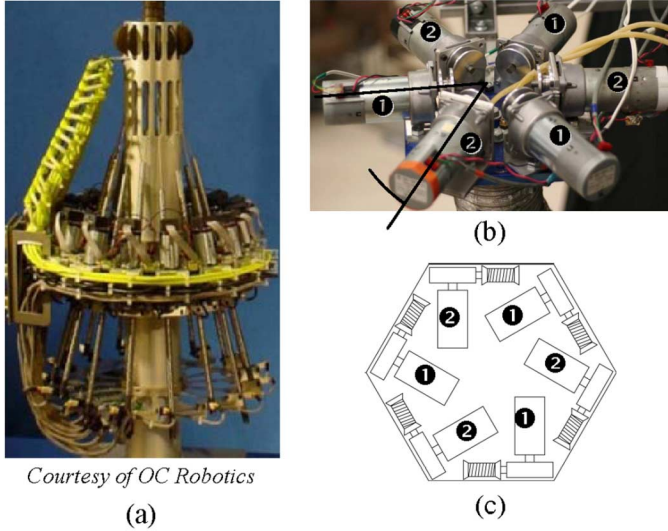


Fig. 11. Various radial actuator layouts. Item (a) shows a multisecton OC Robotics trunk rendering with motors and their linear actuators surrounding the trunk base [29], (b) the two-section Air-Octor motor arrangement, and (c) the layout of the two-section KSI trunk's actuators seen from above [30]. Note that actuators in (b) and (c) which control each section of these trunks, labeled 1 and 2, are spaced 60° apart.

III. TANGLE/UNTANGLE

Many trunk robots are remotely actuated, so that the trunk can be light and soft while the actuators, which are bulky and rigid, remain outside the trunk. To simplify the routing of the tendons in these extrinsic or hybrid actuation trunks, tendons are not colocated, but instead typically arranged with a fixed section-to-section offset such as the 60° offset shown in Fig. 11(b) and (c). Trunks following this arrangement of a fixed section-to-section offset include [2] and [21]–[23], as detailed in the Introduction. As illustrated by the designs shown in the figure, this fixed offset simplifies the mechanical layout of the actuators by placing them radially about the trunk.

One immediate problem this fixed section-to-section offset produces is the necessary coupling between the cable lengths of proximal (closer to the trunk base) sections and distal (farther from the trunk base) trunk segments, as illustrated in Fig. 12. In the two-section Air-Octor, for example, extending the length of the top, proximal section means that cables which run to the bottom, distal section must also be extended by the same amount. When the proximal section is bent, the amount by which each distal cable should be extended in order to maintain distal section shape is unclear. A simplistic approach illustrated in Fig. 13 and detailed in the experimental results given at the end of this section produces incorrect results, because in many continuum robots such as [2] and [21]–[23], proximal and distal cables are not colocated, but rather offset from each other. The following tangle and untangle algorithms derive the coupling relationships necessary to correctly operate a multisecton cable-driven robot by preventing changes in the shape of proximal sections from affecting the shape of distal sections. They represent a new, and to the best of the authors' knowledge, unique contribution to the field of continuum robots. In addition, this algorithm also covers the simpler case

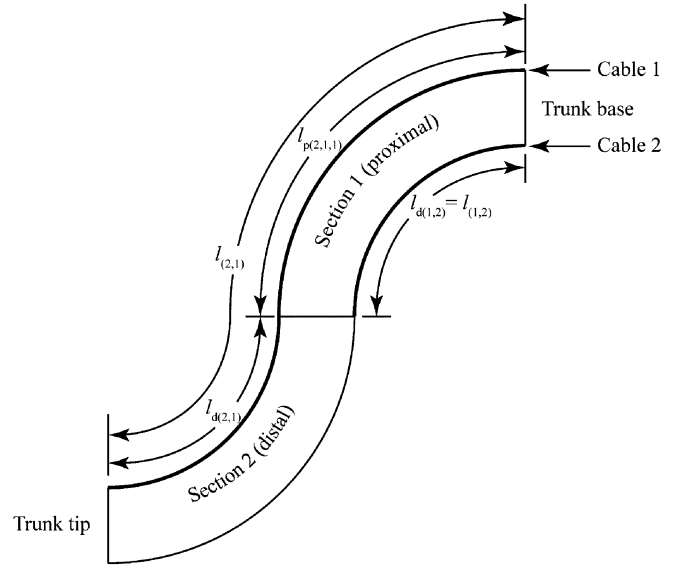


Fig. 12. Simplified diagram illustrating cable tangling/untangling terminology. Only two cables are shown, arranged at 180° from each other for illustration purposes. Cable lengths are given in terms of overall cable lengths l , proximal length l_p , and distal length l_d . Subscripts for these lengths are ([terminating] section, cable number, [section passing through]). Cable 1 controls shape of the distal trunk section, while cable 2 shapes the proximal section. The length of cable 1 can be divided into its length passing through Section 1 and eventually terminating in Section 2, $l_{p(2,1,1)}$, distal length $l_{d(2,1)}$, and their sum, the overall length $l_{2,1}$. Cable 2's length has no proximal component, so that its distal length $l_{d(1,2)}$ is equal to the overall length $l_{1,2}$.

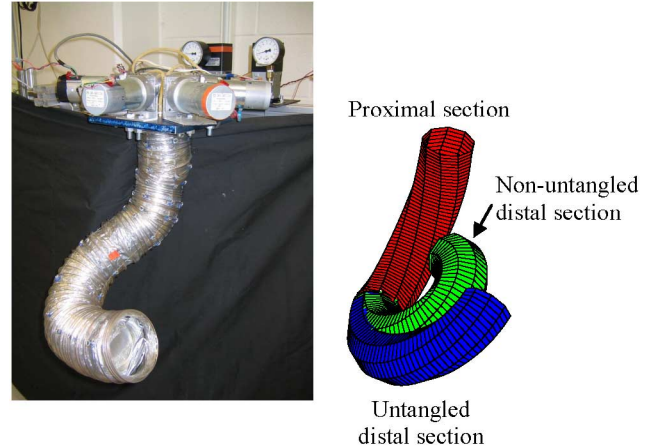


Fig. 13. Results of applying the untangling algorithm to a two-section Air-Octor, comparing measured trunk position shown on the left with and without untangling with the robot from which the measurements were taken.

of a zero section-to-section offset by choosing this offset to be 0° , yielding solutions identical to those sketched in [5] which apply to robots such as [4], [5], and [24]–[26].

As derived in [8], the forward kinematic equations map between trunk shape and cable length for an extrinsically actuated robot are similar (1), except that

$$s = \frac{nd(l_1 + l_2 + l_3)}{\sqrt{g}} \sin^{-1} \left(\frac{\sqrt{g}}{3nd} \right) \quad (16)$$

where $g = l_1^2 + l_2^2 + l_3^2 - l_1 l_2 - l_2 l_3 - l_1 l_3$, n is the number of cable guides in a trunk section, and d is the radius of the trunk. Likewise, the inverse equations

$$\begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} = \begin{bmatrix} 2n \sin\left(\frac{\kappa s}{2n}\right) \left(\frac{1}{\kappa} - d \sin \phi\right) \\ 2n \sin\left(\frac{\kappa s}{2n}\right) \left(\frac{1}{\kappa} + d \sin(\pi/3 + \phi)\right) \\ 2n \sin\left(\frac{\kappa s}{2n}\right) \left(\frac{1}{\kappa} - d \cos(\pi/6 + \phi)\right) \end{bmatrix} \quad (17)$$

from [9] perform the opposite mapping. However, these expressions are insufficient for controlling a multisection cable-actuated robot whose cables are not colocated, such as Air-Octor and trunks discussed in [2], [4], [5], and [21]–[26], due to the effect of the shape of proximal sections on the length of cables passing through these proximal sections to distal sections as diagrammed in Fig. 12. In general, the length $l_{n,i}$ of the i th cable controlling the n th section can be found by summing the length $l_{p(n,i,j)}$ required for the i th cable, which terminates in section n , to pass through the j th proximal section over the $n-1$ proximal sections, then adding the length $l_{d(n,i)}$ of the n th distal section being controlled by cable i , so that

$$l_{n,i} = \sum_{j=1}^{n-1} l_{p(n,i,j)} + l_{d(n,i)}. \quad (18)$$

Algorithms which correctly compensate for these difficulties, termed “tangling” and “untangling,” are presented in the Appendix. Experimental results verify the correctness of these algorithms. Consider the two-section Air-Octor manipulator in which the distance from the trunk center to the cables is $d = 4.5$ cm, number of trunk segments per section $n = 8$, and the length of a nonbending intersection connection is 3 cm. Tangled cable lengths, measured in centimeters, for the proximal section are $(l_{1,1}, l_{1,2}, l_{1,3}) = 35.8, 30.9, 40.9$ cm, and for the distal section are $(l_{2,1}, l_{2,2}, l_{2,3}) = 91.0, 69.2, 63.5$ cm. Untangled lengths for the proximal section are identical by definition, because the proximal section has no additional proximal sections through which its cables must pass. Computing untangled lengths for the distal section gives $(l_{d(2,1)}, l_{d(2,2)}, l_{d(2,3)}) = 53.2, 41.3, 30.6$ cm. This results in the robot configuration shown in Fig. 13, also illustrated by a 3-D model in the same figure. The 3-D models illustrates two results, one using the untangling algorithm derived here and the other without untangling, instead simply subtracting proximal section cables lengths from distal section cable lengths in order to obtain a nonuntangled result of $(l_{d(2,1)}, l_{d(2,2)}, l_{d(2,3)}) = 55.2, 38.3, 22.6$ cm for comparison.

Results obtained for trunk parameters for the proximal section are $\phi = 179^\circ$, $\kappa = 0.0358 \text{ cm}^{-1}$, and $s = 35.9$ cm for both untangling methods. Therefore, as shown in the figure, the proximal section for both methods is placed identically. The distal section shows untangled section results of $\phi = -122^\circ$, $\kappa = 0.0696 \text{ cm}^{-1}$, and $s = 41.9$ cm while the nonuntangled results for the distal section are $\phi = -121^\circ$, $\kappa = 0.108 \text{ cm}^{-1}$, and $s = 39.1$ cm. As illustrated in the figure, the untangled trunk reproduces the curvature exhibited by the robot, while the nonuntangled result does not. Therefore, when using the untangling algorithm, changes made to the shape of the proximal section do not affect distal section shape. In contrast, without an untangling algorithm, changes to the proximal section affect the shape

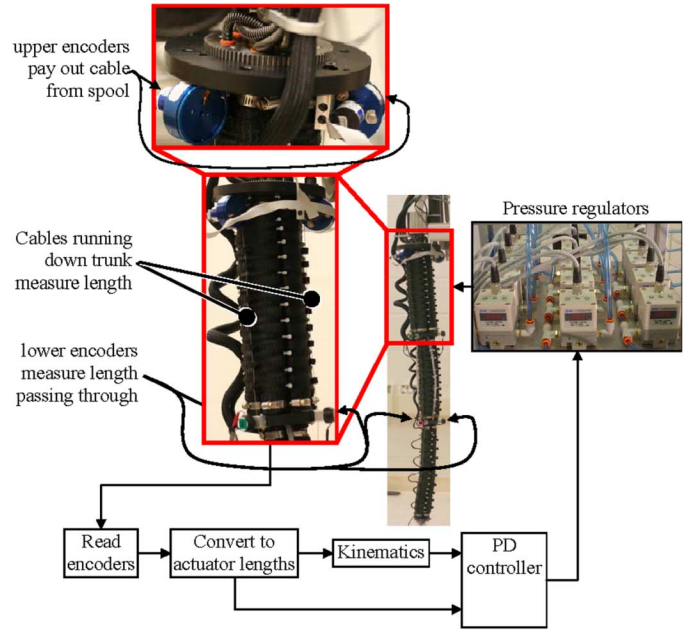


Fig. 14. Details of the OctArm trunk control system.

of both sections. Note also that the untangled and actual trunks differ from the physical trunk primarily in the curvature of the first section, due to measurement errors and unmodeled effects such as the force of gravity, which causes the robot to droop due to the weight of the second section. Because the algorithms are not tied to the specific model given in (16) and (17), the same algorithms could be used with alternative kinematics in the future that, for example, compensate for gravity load for higher accuracy results.

IV. IMPLEMENTATION

The algorithms introduced in this paper were successfully implemented on two continuum robots. The operational flow used for both robots is similar. First, user input specifying a desired trajectory or trunk pose is read from a joystick. Next, a kinematics block which implements both the kinematics given in [8]–[10], and the implementation-specific algorithms developed in this paper compute desired actuator lengths based on the given user input. Finally, a proportional-derivative (PD) loop moves the position of the trunk as measured by encoders to the desired trunk position computed by the kinematics.

The first continuum robot, OctArm, is an intrinsically actuated trunk consisting of three sections, shaped by pneumatic pressure applied to three McKibben actuators which compose each trunk section [18]. As diagrammed in Fig. 14, three encoders mounted at the trunk base (proximal portion, or top in the figures) measure the cable length running from the top to the bottom of the trunk, while two sets of three encoders mounted at the second and third sections measure the length of cable passing through them. A conversion block shown in the figure subtracts these lengths to compute a per-section measured length. Note that untangling is not necessary in this case because the same three cables pass through all sections of the trunk, a colocated design, unlike the per-section rotation illustrated in Fig. 12 and diagrammed in Fig. 11(b), which

requires tangling and untangling. Next, a PD block examines the difference between measured length and desired length computed by the kinematics and produces a desired pressure. Commercially-available pressure regulators then inflate each of the nine actuators in the trunk with the desired pressure, causing the trunk to achieve the desired shape.

Kinematic computations rely on the understanding developed in Section II to calculate desired trunk position. As described in [31], three control modes enable the user to position the trunk. In position mode, the trunk's position reflects joystick position by mapping the joystick's position in (x, y) coordinates to polar coordinates (r, ϕ) , which determine trunk posture by choosing the trunk's direction of curvature as ϕ and curvature as $\kappa = cr$, where c is a scaling constant. Assuming the joystick's range is normalized so that the maximum r produced is 1, the scaling constant c can be determined either by choosing $c = \kappa_{\max}(\phi)$ from (11) so that $\kappa = \kappa_{\max}(\phi)$ when $r = 1$, yielding the maximum possible curvature at any direction of curvature, or by choosing $c = \kappa_{\max}$ given in (3), yielding uniform maximum curvature regardless of the direction of curvature. Both the independent and dependent velocity modes apply a velocity to the current trunk position, computing updated positions as $(\kappa_{\text{new}}, \phi_{\text{new}}) = (\kappa_{\text{current}} + t\dot{\kappa}, \phi_{\text{current}} + t\dot{\phi})$. Commanded curvature κ_{new} must not exceed the maximum achievable curvature as defined by (11). All control modes allow the user to choose a trunk length s which lies within range of trunk lengths s which enable the trunk to achieve the desired curvature κ in the commanded direction of curvature ϕ by computing the range $[s_{\min}(\phi), s_{\max}(\phi)]$ by evaluating (14). All three modes also provide a grasping mode, in which the direction of curvature is chosen as $\phi = \{30^\circ, 150^\circ, 270^\circ\}$ as derived in the discussion following (11), and trunk length is chosen as $s_{\max}(\phi)$ from (10) in order to produce the tightest possible whole-arm grasp.

Fig. 1(a) illustrates the results of this approach. Using a joystick, the operator begins with a straightened trunk, then commands the tip sections to bend in a circular arc, forming an encircling grasp about the cone. Trunk length s is automatically adjusted using algorithms in this paper to provide the maximum possible curvature to create the tightest possible grasp in that direction. Next, snapping the grasp direction to one of the optimal angles at $\phi = \{30^\circ, 150^\circ, 270^\circ\}$ allows the best possible grasp. Due to gravity sag, the trunk conveniently forms a helix, rather than a circle, about the cone. The remaining trunk sections and rigid-link joints on the Talon mobile platform then provide the ability to lift the cone and move it as desired, for example, by setting it down on another cone.

This figure also provides an example of the unique capabilities of continuum robots. Experiments in both real-world [32] and laboratory [33] environments demonstrate that the same enveloping grasp allows the trunk to securely grasp and manipulate a wide variety of useful objects. For example, the trunk can grasp small and large objects, fragile objects made of glass or aluminum foil, unusual shapes such as collections of tree branches or a pillow, grasps robust against severe vibrations such as dragging a pallet at high speeds, hook-style grasps by inserting the trunk tip through an opening in the object, and biologically inspired prey strikes to quickly seize an object. The use of a simple operator interface, in which joystick positions

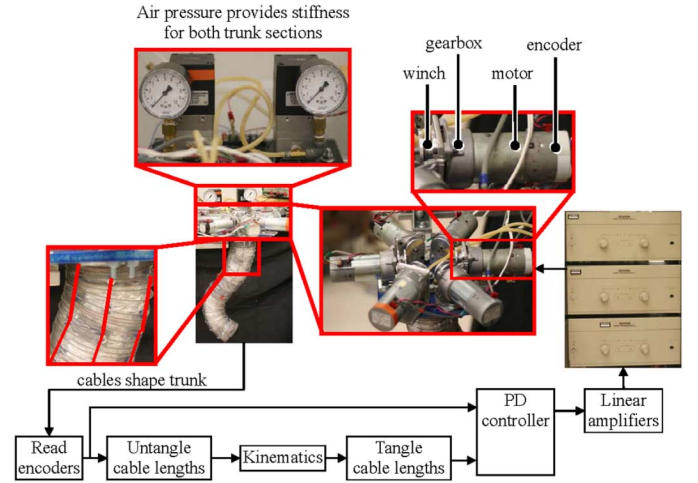


Fig. 15. Details of the Air-Octor trunk control system.

map to trunk velocity trajectories, allows the operator to naturally move the trunk in a desired direction.

The second continuum robot, Air-Octor, is a hybrid actuated trunk. Air pressure blown into the center of each of the two sections provides stiffness, while electric motors winch in three cables per section to shape the trunk. As illustrated in Fig. 15, encoders mounted on the motors provide position feedback by measuring overall (tangled) cable length, which an untangle block converts to per-section (untangled) actual lengths. The kinematics block, given these per-section actual lengths and input from a joystick, produces desired per-section lengths, which a tangle block then converts to overall (tangled) cable length. A PD loop regulates trunk position by servoing actual overall cable lengths to the desired overall cable lengths. Voltages sent to a linear amplifier are then applied to the motors by the PD block. The tangle/untangle process is therefore essential to the control loop, because it both provides accurate shape information to the kinematics and converts output from the kinematics into a usable form for PD control of actual, measured (tangled) cable length.

V. RESULTS AND CONCLUSIONS

The algorithms and analysis introduced in this paper, somewhat analogous to the handling of joint limits in conventional manipulators, overcome most of the (previously unsolved) key implementation problems unique to continuum robots. They also make a wide variety of continuum robots more usable to operators in several essential ways. First, deriving the constraints which physical actuator length limits place on an intrinsically actuated trunk reveals the complex nature of the relationship between trunk length, curvature, and direction of curvature. The analysis then provides a method for computing the range of trunk lengths that enable the trunk to achieve the commanded curvature at the desired direction of curvature. These methods enable joystick-based command of trunk shape, and are therefore essential components of the of the implemented OctArm control system. In addition, the derivation of the workspace of a trunk under actuator length limits both defines the capabilities of the trunk and provides a method to

choose actuators to enable the reaching of all points within a given workspace.

Second, the tangle/untangle algorithm provides correct shape information essential both when using cables to sense the trunk's shape, or when using cables attached to actuators to define the shape of the trunk. Cable length sensed by encoders can therefore be untangled to per-section length, providing input for kinematics algorithms. Their output, again as a set of per-section lengths, can then be tangled and used in a control loop to affect trunk shape. These algorithms therefore eliminate a significant source of error, enabling much more accurate control of trunk shape.

Finally, an implementation of all the algorithms discussed in this paper validates their correctness and verifies their usefulness. Two different robots, representative of the two classes of continuum trunk design, provide unique, specific examples of both the special implementation problems presented by continuum robots and their solutions. They motivate the development of grasping strategies which require the maximum possible curvature to successfully encircle, then grasp, an object, and provide a vehicle for a future series of experiments to define and extend the capabilities of continuum manipulators. The progress reported herein provides a set of essential algorithms to enable the implementation of continuum robotics throughout the research community.

Field and laboratory testing performed based on the techniques developed in this paper also indicates the need for further development of modeling techniques for continuum robots. In particular, the field lacks approaches the dynamic modeling of a trunk suited to real-time control. Existing methods [4], [17], [34], [35] do not provide closed-form solutions, requiring computationally expensive numerical integration or iteration to find a solution. These challenges are additionally complicated by the dynamics of the pneumatic nature of the McKibben actuators [18], providing rich avenues for future work.

APPENDIX

A. Derivation of Extrema Stated in Section II

As discussed in Section II, the global minimum of the function κ given in (1) occurs along the line $l_1 = l_2 = l_3$. To verify this, proceed using standard minimization techniques by first computing $\partial\kappa/\partial l_1 = 0$, then solving to yield $l_1 = (l_2^2 + l_3^2)/(l_2 + l_3)$. The same approach for l_2 and l_3 likewise gives $l_2 = (l_1^2 + l_3^2)/(l_1 + l_3)$ and $l_3 = (l_1^2 + l_2^2)/(l_1 + l_2)$. Simultaneous solution of these equations yields a complex conjugate pair and the line $l_1 = l_2 = l_3$. Because actuator lengths are assumed to be real, the complex-valued solution is rejected. To determine the type of this extremum, the second derivate can be evaluated at this point. Factoring the second derivative produces

$$\frac{8g^2 + 4gr^2 - 4gtr - r^2t^2}{2g^{3/2}r^3d}$$

where $t = 2l_1 - l_2 - l_3$, $r = l_1 + l_2 + l_3$, d is the radius of the trunk, and $g = l_1^2 + l_2^2 + l_3^2 - l_1l_2 - l_2l_3 - l_1l_3$. Careful evaluation of this expression at its limiting point when all three

cable lengths are equal gives a result of ∞ assuming $d > 0$ and $l_{1-3} > 0$, proving this point a minimum.

To determine the isosurface of κ , recall that $\kappa = \theta/s$ so that changing both θ and s by the same amount must leave κ unchanged. In terms of actuator lengths, we seek a new set of lengths \underline{l}' which satisfies these constraints given an existing set of actuator lengths \underline{l} . For a three-variable system composed of three unknowns \underline{l}'_{1-3} , constraints on θ and s in terms of cable lengths can be determined by evaluating (1) at $\theta' = c\theta$ and $s' = cs$, where c is some positive constant, in order to keep the resulting trunk length s' positive. A third constraint is required to solve the system of equations. While a constraint of $\phi = 0$ provides this constraint, the introduction of arctan terms in (1) prevent solution of the resulting equations. As a substitute, the geometrical equivalent of $\phi = 0$ can be found by determining the vector along which movement about ϕ occurs. First, recall that any vector on the line $l_1 = l_2 = l_3$ points in the direction of s . Referring to (2), a vector from a point \underline{l} to the line $l_1 = l_2 = l_3$ is proportional to θ . The cross-product of these two vectors therefore represents a vector along the only remaining variable ϕ . Computing this cross-product gives

$$\left(\underline{l} - \frac{\underline{l} \cdot [1 \ 1 \ 1]^T}{\|[1 \ 1 \ 1]^T\|^2} [1 \ 1 \ 1]^T \right) \times [1 \ 1 \ 1]^T = \begin{bmatrix} l_2 - l_3 \\ l_3 - l_1 \\ l_1 - l_2 \end{bmatrix} = \phi_v$$

where ϕ_v gives the vector which points along ϕ . Therefore, choosing $\phi = 0$ is equivalent to insuring that the resulting vector \underline{l}' has no components along ϕ_v by choosing a dot product of zero, $\underline{l}' \cdot \phi_v = 0$, as the third constraint. Solving yields two solutions, one of which flips the trunk by 180° and another which gives the promised $\underline{l}' = c\underline{l}$. Rejecting the first solution because it modifies ϕ by causing ϕ_v to point in the opposite direction yields the desired result of $\underline{l}' = c\underline{l}$ for one dimension of the isosurface.

B. Tangle and Untangle Algorithms

Consider the two-section Air-Octor robot when the proximal (top) section is straight, implying its curvature $\kappa = 0$. In this case, the cable lengths of the proximal section $l_{1,1...3}$ measured from the actuators to their anchor points at the end of the proximal section are all equal to its arc length s_p , so that $l_{1,1...3} = s_p$. Cable lengths of the distal (bottom) section, measured from the actuators to their anchoring points at the end of the distal section, must pass through the proximal section, so that evaluating (18) for $n = 2$ yields $l_{2,1...3} = l_{p(2,1...3,1)} + l_{d(2,1...3)}$. However, because the proximal section's curvature is zero, any cable, such as the portion of the distal section's cables traveling to through the proximal section to the distal section, must have the length s_p . Therefore, the overall cable length for the distal section cable can be easily solved as $l_{2,1...3} = s_p + l_{d(2,1...3)}$.

However, determining l_p when $\kappa \neq 0$ requires additional computation. As shown in Fig. 11(b), the three cables controlling the proximal section are offset by 60° from the three cables controlling the distal section. Therefore, the length $l_{p(1,1...3)}$ of the portion of the distal-section cables which pass through the

proximal section is equal not to the length of the proximal-section cables $l_{1,1...3}$, but instead to the length of the proximal-section cables when direction ϕ_1 of the proximal section's curvature κ_1 is rotated by the section-to-section offset of 60° . This suggests an algorithm: first convert the proximal (top)-section cable lengths to trunk parameters ϕ_p, κ_p , and s_p , then compute $l_{p(1,1...3)}$ from the rotated trunk parameters $\phi_p + 60^\circ, \kappa_p$, and s_p .

This algorithm, termed cable untangling, can be generalized to an n -section robot. This assumes the actuators are radially arranged such the section-to-section offset is $120^\circ/n$, a common design shared by many robots such as [2] and [21]–[23]. Given “tangled” cable lengths $l_{1...n,1...3}$, measured from each actuator to the cable's termination, this algorithm computes the “untangled” length $l_{d(1...n,1...3)}$ of cable which passes only through the distalmost portion of each section of the robot by descending the trunk one section at a time, subtracting off proximal length $l_{p(1...n,1...3,1...n-1)}$ from all distal cables passing through the current section.

Assign $l_{d(1...n,1...3)} \leftarrow l_{1...n,1...3}$.

For $j = 1$ to n

 Compute ϕ, κ , and s from $l_{d(j,1...3)}$ using (16) because the j th section is now “untangled.”

 For $k = j + 1$ to n

 Compute $l_{p(k,1...3,j)}$ from $\phi + 120^\circ(k-1)/n, \kappa$, and s using (17).

 Subtract proximal length: $l_{d(k,1...3)} \leftarrow l_{d(k,1...3)} - l_{p(j,1...3,k)}$.

 End for

End for

The order in which these computations occur is important, because the formulas derived in the earlier section used to convert between trunk parameters ϕ, κ , and s and cable lengths l_d apply only to “untangled” or distal cable lengths. This prohibits the straightforward solution of (18) as $l_{d(n,i)} = l_{n,i} - \sum_{j=1}^{n-1} l_{p(n,i,j)}$ because $l_{p(n,1...3,j)}$ for the j th section must be computed from ϕ, κ , and s , which can only be found by first knowing the distal cable length $l_{d(j,1...3)}$ for that section. This distal cable length depends on proximal lengths from sections 1 to $j-1$, which then depend on distal lengths from sections 1 to $j-1$. To resolve this chain of dependencies, note that the first (most proximal) section has no proximal dependencies, so that distal cable length $l_{d(1,1...3)} = l_{1,1...3}$, the overall cable length. The algorithm can then walk down the trunk, “untangling” the current section by computing its distal cable lengths, transforming these to trunk parameters ϕ, κ , and s , then transforming them back to proximal cable lengths of trunk sections passing through the current section.

Similarly, the following cable tangle algorithm computes “tangled” cable lengths $l_{1...n,1...3}$ from “untangled” lengths $l_{d(1...n,1...3)}$ by descending the trunk once section at a time,

adding in proximal length $l_{p(1...n,1...3,1...n-1)}$ to all distal cables passing through the current section. Once again, this structure is mandated by the dependencies inherent in the defining equations.

Assign $l_{1...n,1...3} \leftarrow 0$.

For $j = 1$ to n

 Compute ϕ, κ , and s from $l_{j,1...3}$ using (16).

 Add distal length $l_{j,1...3} \leftarrow l_{j,1...3} + l_{d(j,1...3)}$.

 For $k = j + 1$ to n

 Compute $l_{p(k,1...3,j)}$ from $\phi + 120^\circ(k-1)/n, \kappa$, and s using (17).

 Add proximal length: $l_{k,1...3} \leftarrow l_{k,1...3} + l_{p(k,1...3,j)}$.

 End for

End for

These algorithms complement and complete the kinematics derived earlier. Cable lengths sensed by the trunk's hardware, such as encoders on the motors of Air-Octor, are first untangled then fed to the kinematics. A control system specifies a desired trajectory, such as that produced by evaluating the velocity kinematics in [10] based on joystick input. These new desired positions are then tangled and fed to a control loop which serves the actual (measured) tangled lengths to the desired tangled lengths, moving the robot to the commanded location.

Because the heart of these algorithms is a transformation from cable lengths to trunk parameters, they can be easily modified to accommodate various design choices by appropriately modifying the trunk parameters. For example, trunks such as [5], whose distance d from the trunk center to the location of the cable varies by section, can be easily tangled or untangled by modifying the d used to convert between cable lengths and trunk parameters.

REFERENCES

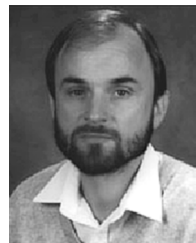
- [1] G. Robinson and J. B. C. Davies, “Continuum robots—A state of the art,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Detroit, MI, 1999, pp. 2849–2854.
- [2] S. Hirose, *Biologically Inspired Robots*. Oxford, U.K.: Oxford Univ. Press, 1993.
- [3] G. S. Chirikjian and J. W. Burdick, “A modal approach to hyper-redundant manipulator kinematics,” *IEEE Trans. Robot. Autom.*, vol. 10, no. 3, pp. 343–354, Jun. 1994.
- [4] I. A. Gravagne, C. D. Rahn, and I. D. Walker, “Large deflection dynamics and control for planar continuum robots,” *IEEE/ASME Trans. Mechatron.*, vol. 8, no. 4, pp. 299–307, Jul. 2003.
- [5] 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, pp. 45–63, Feb. 2003.
- [6] S. Hirose and K. Yokoshima, “2 DOF Moray drive for hyper redundant manipulator,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1992, pp. 1735–1740B.
- [7] S. Ma, S. Hirose, and H. Yoshinada, “Development of a hyper-redundant multijoint manipulator for maintenance of nuclear reactors,” *Int. J. Adv. Robot.*, vol. 9, pp. 281–300, Jun. 1995.
- [8] 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, 2004, pp. 745–750.

- [9] W. McMahan, B. A. Jones, I. D. 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.
- [10] B. A. Jones and I. D. Walker, "A new approach to Jacobian formulation for a class of multi-section continuum robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 3279–3284.
- [11] K. Suzumori, S. Iikura, and H. Tanaka, "Development of flexible microactuator and its applications to robotic mechanisms," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sacramento, CA, 1991, pp. 1622–1627.
- [12] D. M. Lane, J. B. C. Davies, G. Robinson, D. J. O'Brien, J. Sneddon, E. Seaton, and A. Elfstrom, "The AMADEUS dextrous subsea hand: Design, modeling, and sensor processing," *IEEE J. Ocean. Eng.*, vol. 24, no. 1, pp. 96–111, Jan. 1999.
- [13] P. Kallio, M. Lind, Z. Quan, and H. N. Koivo, "A 3-DOF piezohydraulic parallel micromanipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Leuven, Belgium, 1998, pp. 1823–1828.
- [14] 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.*, Seoul, Korea, 2001, pp. 2454–2459.
- [15] G. Chen, P. M. Tu, T. R. Herve, and C. Prella, "Design and modeling of a micro-robotic manipulator for colonoscopy," in *Proc. 5th Int. Workshop Res. Educ. Mechatron.*, Annecy, France, 2005, pp. 109–114.
- [16] H. Ohno and S. Hirose, "Design of slim slime robot and its gait of locomotion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Maui, HI, 2001, pp. 707–715.
- [17] M. Ivanescu, N. Bizdoaca, and D. Pana, "Dynamic control for a tentacle manipulator with SMA actuators," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, 2003, pp. 2079–2084.
- [18] M. B. Pritts and C. D. Rahn, "Design of an artificial muscle continuum robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, New Orleans, LA, 2004, pp. 4742–4746.
- [19] M. Ivanescu and V. Stoian, "A variable structure controller for a tentacle manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Nagoya, Japan, 1995, pp. 3155–3160.
- [20] R. Kornbluh, R. Pelrine, J. Eckerle, and J. Joseph, "Electrostrictive polymer artificial muscle actuators," in *Proc. IEEE Int. Conf. Robot. Autom.*, Leuven, Belgium, 1998, pp. 2147–2154.
- [21] R. Buckingham, "Snake arm robots," *Ind. Robot: An Int. J.*, vol. 29, pp. 242–245, 2002.
- [22] W. McMahan, B. A. Jones, and I. D. Walker, "Design and implementation of a multi-section continuum robot: Air-Octor," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, AB, Canada, 2005, pp. 3345–3352.
- [23] G. Immega and K. Antonelli, "The KSI tentacle manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Nagoya, Japan, 1995, pp. 3149–3154.
- [24] V. C. Anderson and R. C. Horn, "Tensor arm manipulator design," *Trans. ASME*, vol. 67-DE-57, pp. 1–12, 1967.
- [25] A. Morecki and J. Knapczyk, "Basics of robotics: Theory and components of manipulators and robots," in *International Center for Mechanical Sciences (CISM) Courses and Lectures no. 402*. New York: Springer-Verlag, 1999, pp. 500–501.
- [26] 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.
- [27] K. Salisbury, W. Townsend, B. Ebrman, and D. DiPietro, "Preliminary design of a whole-arm manipulation system (WAMS)," in *Proc. IEEE Int. Conf. Robot. Autom.*, Philadelphia, PA, 1988, pp. 254–260.
- [28] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. New York: Wiley, 2006.
- [29] OCRobotics, <http://www.ocrobotics.com/>.
- [30] G. Immega, "Tentacle-Like Manipulators With Adjustable Tension Lines," U.S. Patent 5 317 952, Jun. 7, 1994.
- [31] M. Csencsits, B. A. Jones, and W. McMahan, "User interfaces for continuum robot arms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, AB, Canada, 2005, pp. 3011–3018.
- [32] W. McMahan, B. A. Jones, V. Chitrakaran, M. Csencsits, M. Grissom, M. Pritts, C. D. Rahn, and I. D. Walker, "Field trials and testing of the OctArm continuum manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, 2006, pp. 2336–2341.
- [33] B. A. Jones, M. Csencsits, W. McMahan, V. Chitrakaran, M. Grissom, M. Pritts, C. D. Rahn, and I. D. Walker, "Grasping, manipulation, and exploration tasks with the OctArm continuum manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, 2006, video.
- [34] G. S. Chirikjian, "Hyper-redundant manipulator dynamics: A continuum approximation," *Adv. Robot.*, vol. 9, pp. 217–243, 1995.
- [35] H. Mochiyama and T. Suzuki, "Kinematics and dynamics of a cable-like hyper-flexible manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, 2003, pp. 3672–3677.



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.