2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

Forward Kinematic Model for Continuum Robotic Surfaces

Jessica Merino[†] Anthony L. Threatt, Ian D. Walker, and Keith E. Green

Abstract— **In this paper, we consider the modeling of robotic continuous “continuum” two-dimensional surfaces. We discuss the fundamental differences between such robot surfaces and traditional rigid link and continuum robots. We then introduce new kinematic models for continuum robotic surfaces. We compare the kinematic models to physical continuum surfaces and validate their performance.**

I. Introduction

The traditional approach to providing movement in both robotics and architecture focuses on the movement of rigid structures (e.g. manipulator links, wheels, doors, and windows) along or about surfaces of 1-dimension (1-D) (i.e. lines and axes). Consequently, functionally, traditional robot manipulators trace “jagged lines in space,” and their kinematics identify a series of straight lines (or, for mobile robots, a geometry of line-based movements), which, via movement along them and at their intersections, define the end effector (or vehicle center) location, orientation, and movement of the robot. Physically, their rigid links typically form a jagged, vertebrate-like backbone, with only a small portion of their bodies having meaningful interactions with their environment. While design in these traditional areas of robotics has focused on extracting optimal performance from specific parts of the robot (e.g. end effector and vehicle center), in general, the local (internal body) movement capability of the robot is still restricted to 1-D. Even the realization that concentrating two or more movements at the same location (e.g. in a spherical wrist) significantly increases performance does not overshadow the acknowledgment that although this approach has the advantage of being relatively simple to engineer and analyze, it is restrictive in the local movements afforded and, thus, the spatial configurations the system can attain. Therefore, we have begun to more deeply explore the class of robotics known as continuum robotics with the intent of utilizing their local movements. Furthermore, unlike our predecessors who have explored continuum lines in space, we have extended our research to continuum robotic surfaces. Although we are not the only researchers currently exploring such surfaces [1], to our knowledge, we are the first to explore the forward kinematics of such devices. The concept is, however, in part inspired by the creations of Oosterhuis [2] in exploration of adaptive spaces.

There are a large number of unusual potential applications for two-dimensional (2-D) robotic surfaces that would utilize and require the local movement capabilities of a true continuum 2-D robotic surface. Reconfigurable antenna arrays could utilize the surface to create a configuration needed for optimal performance [3]. Robot surfaces may be developed to optimize the aerodynamics of race cars or boats; continuum surface shells could reconfigure themselves to yield different acoustic sounds in concert halls; continuum surface lamps may be developed to aid in the adaptive distribution or placement of light. Another increasingly promising application for a continuum robot surface lies in the healthcare realm. Robot surfaces are good candidates for adapting a flexible surface that lies in the hard shell of the bio-mask [4]. Also, the ability of a robot surface to continuously adapt its shape could transform the nature of over-the-bed tables found in both healthcare facilities and at home, significantly improving the lives of people with impaired mobility [5]. Programmable robotic surfaces and slings could revolutionize rehabilitation in numerous conditions, such as stroke therapy. Therefore, the development of such robotic surfaces is of significant interest.

We are not claiming that conventional robotic (and other) systems do not involve the movement of surfaces. Indeed, probably the best-known and still the most successful manipulation system - the parallel jaw end-effector - is based on the movement of surfaces (i.e. the jaws of the gripper). For mobile robots, the “rubber meets the road,” literally, via surface contact between tire and ground. Also, the effect of surface-fingertip contact has been discussed extensively in the multi-fingered manipulation community. However, in none of these cases can the shape of the robot surface be actively controlled and reconfigured. Particularly for human-robot interaction, we believe that the current ”missing dimension” in robot movements (2-D) will be essential for complex applications.

Consequently, we make the argument for an alternative design approach based on the actuation of flexible, continuous 2-D surfaces, which promises the aforementioned new and novel applications for both robotics and architecture. Just as the concept of robotic surfaces is not new, the concept of continuum robots is not new [6]. In recent years, significant theoretical understanding [7],[8],[9],[10],[11],[12],[13],[14], [15], [16], [17], [18], [19] and practical development [20], [21] of “trunk and tentacle” robots has been achieved. While numerous physical designs have been proposed [22], [23], [24], [25] and demonstrated [26], [27], [28], [29], [30], [31], the common overall design goal has been to create the effect of a continuous “invertebrate-like” backbone, as

[†] To whom all correspondence should be addressed.

J. Merino and I. D. Walker are with the Department of Electrical & Computer Engineering, Clemson University, Clemson, SC 29634-0915 ((jmerino, iwalker)@clemson.edu). A.L. Threatt and K.E. Green are with the School of Architecture, Clemson Univsersity, Clemson, SC 29634-0915 ((anthont, kegreen)@clemson.edu). This research was supported by the U.S. National Science Foundation under awards IIS-SHB-1116075 and IIS-0904116.

978-1-4673-1735-1/12/S31.00 ©2012 IEEE

opposed to the "vertebrate-like" backbone of conventional manipulators. This led to a core innovation of allowing movement along curved lines as opposed to the rigid, straight lines of traditional robots, which, in turn, allows continuum robots to better match their configuration (i.e. shape) to their environments [32], [33]. However, despite the novel capabilities offered by this emerging class of continuum robots, they also remain, in essence, "lines in space." Extending upon this, we herein explore a new class of 2-D continuum robots, "programmable surfaces in space."

Aside from their lack of localized movement, one may wonder why the more conventional, traditional design approaches could not be adapted for the aforementioned applications. Conventional robot structures prove to be too rigid and inflexible (away from their joints) to generate the smooth, compliant and safe movements desired from the proposed applications. The key difference between the previously discussed examples of surface contact and the continuum-robot surfaces we propose is that the 2-D shape of the continuum-robot surfaces need to be actively controlled, as distinguished from the fixed plane of the parallel jaw gripper or the passive compliance of tires and typical soft fingertips. In short, this active-shaping of the surface in 2-D distinguishes our approach from previously engineered surfaces (e.g. rudders, sails, and fins for marine robots, and various "robotic walls") [34], [35], [36], [37], [38]. This is important because, for instance, in stroke rehabilitation, it is believed that the 2-D movement capabilities throughout the robotic continuum surface will allow the patient to feel more comfortable and supported while exercising the injured limb. However, the fundamental problem of how to build robot surfaces that will allow such reconfigurations to be realized has received little attention in the robotic literature. Further, to effectively deploy such surfaces, kinematic models will be necessary to plan and control desired configurations. The forward kinematic models for continuum surfaces introduced herein are an initial step in achieving this goal; we introduce a new kinematic model for continuum robotic surfaces and validate it via hardware realizations.

II. Base Kinematic Model

In this paper, we specifically consider and model 2-D continuum robotic surfaces actuated by tendons or muscles embedded within the surface. Our physical prototypes (detailed in Section V) are actuated by McKibben (air-) muscles, which are pneumatic artificial muscles. However, the modeling approach applies to any physical actuation scheme which results in locally constant curvature.

A. Discrete/Continuous Base Models

The phrases "muscle location" and "muscle arrangement" will be used throughout this paper to describe the continuum surface. The term "muscle location" describes the location at which the muscle is placed within the continuum surface. The muscle location, not the coordinates of points on the surface, will be characterized by three values y,z,$\phi \in \mathbb{R}$, represented as (y, z, ϕ); y and z represent the y-coordinate and z-coordinate values of the tube end of the muscle (the end in which the air tube enters), and ϕ represents the angle that the muscle has been rotated counter-clockwise from the z-axis, around the x-axis, where the fixed point of rotation is (y, z). The term "muscle arrangement" is used to describe a set of multiple muscle locations. We consider how muscles located at different points on the continuum surface combine to produce different effects on the movement of the surface.

The first step in developing the forward kinematics for continuum surfaces involves defining the base continuum kinematic model from which the new kinematics will build. This model is restricted to the kinematics for a line in space but will herein be extended to define the kinematics for a continuum surface. This model is similar to [39]; however, it differs in the manner in which the angle of rotation is referenced. Also, note that, like [39], we assume constant curvature and constant length of the muscles.

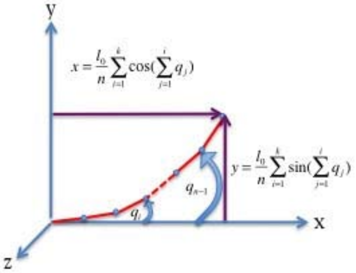

y
$x=\frac{l_0}{n}\sum_{i=1}^{k}\cos(\sum_{j=1}^{i}q_j)$
$y=\frac{l_0}{n}\sum_{i=1}^{k}\sin(\sum_{j=1}^{i}q_j)$
q_{n-1}
q_i
x
z

Fig. 1. Representation of the Discrete Kinematics

For the conventional, planar, discrete (rigid link) case, illustrated in Figure 1, the (x, y) coordinates of a joint on the backbone can be expressed as

$$x = \frac{l_0}{n}\sum_{i=1}^{k}\cos\sum_{j=1}^{i}q_i; \qquad y = \frac{l_0}{n}\sum_{i=1}^{k}\sin\sum_{j=1}^{i}q_i, \quad (1)$$

where $l_0 \in \mathbb{R}^+$ represents the length of the manipulator, $n \in \mathbb{R}^+$ represents the number of links in the manipulator, $\sum_{j=1}^{i} q_i$ represents the orientation of the i^{th}, where $i \in [1, n]$, link measured counter-clockwise from the base frame x-axis, and (x, y) represents the position of the k^{th}, where $k \in [1, n]$, joint of the manipulator measured in the base frame. Note, all link lengths are equal.

For the continuous case, the number of links approaches infinity. Correspondingly, as seen in [39], the equations will no longer be discrete.

B. Modified Kinematic Equations

For the development of our continuum surface models in the following sections, the direction from which $\theta(s)$ is referenced will be altered from its definition in [39] to reference the angles as counter-clockwise from the y-axis as opposed to clockwise. This modification is made to be representative of our physical continuum surfaces - the continuum surface was hung perpendicular to the floor - that were tested in our lab. This change, illustrated in Figure 2, results in the following kinematics:

$$x(s) = \int_0^s sin(\int_0^\sigma k(v)\,dv)\,d\sigma \qquad (2)$$

$$y(s) = \int_0^s -cos(\int_0^\sigma k(v)\,dv)\,d\sigma, \qquad (3)$$

where $x(s), y(s) \in \mathbb{R}$ jointly represent the position vector to the point associated with arclength $s \in [0, l_0]$ from the origin. The expression

$$\theta(s) = \int_0^s k(\sigma)d\sigma, \qquad (4)$$

where $\theta(s) \in \mathbb{R}$, represents the orientation of the tangent to the curve at each value of s, and it is measured as counter-clockwise from the y-axis. In equations (2), (3), and (4), $k(\bullet) \in \mathbb{R}$ represents the curvature function.

The only difference between equations (2) and (3) and the continuous equations that result from [39] is that the $y(s)$ terms are of opposite sign to one another; our $y(s)$ is negative whereas the $y(s)$ from [39] is positive.

These kinematic formulas are used for muscles that are located at $(0, z, 90)$, in which $z \in (0, z_{max})$, where z_{max} corresponds to the maximum z-coordinate position at which an edge of the surface lies (which corresponds to the length of the surface). Similarly, y_{min} will correspond to the most negative y-coordinate position at which an edge of the surface lies (which corresponds to the width of the surface).

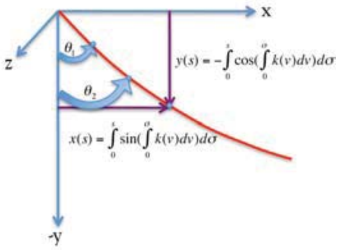

x
z
θ_1
θ_2
$y(s) = -\int_0^s \cos(\int_0^\sigma k(v)dv)d\sigma$
$x(s) = \int_0^s \sin(\int_0^\sigma k(v)dv)d\sigma$
-y

Fig. 2. Representation of the Continuum Kinematics

III. Extension to a Continuum Surface

A pneumatically driven continuum surface can be viewed as a collection of curves in space, which are induced by and are parallel to pneumatic muscles embedded within and actuating the surface. We present the idea that the curvatures of the lines induced by the muscle(s) will be calculated via an interpolation function; this concept is portrayed in Figure 3. Depending on the position and/or arrangement of the muscle(s), either a cubic or an exponential interpolation function will be used, subsequently, to calculate the curvatures of the lines induced by the pneumatic muscle(s); muscle position and/or arrangement affect the shapes that are created along the edges of the surface when the muscles are actuated. The use of one function over the other depends on the position, and in one case the arrangement, of the actuating muscles within the surface. The calculation of the curvatures is necessary to compute the coordinates of any point on the continuum surface after it is actuated.

A. Interpolation

The interpolation procedure proposed for continuum surfaces is of the general form:

$$k(z) = g(\frac{z}{z_{max}})k_1 + g(1 - \frac{z}{z_{max}})k_2. \qquad (5)$$

In equation (5), $k(z) \in \mathbb{R}$ is the curvature of the curve induced at coordinate $z \in [0, z_{max}]$, and $k_1 \in \mathbb{R}^+$ and $k_2 \in \mathbb{R}^+$ are the curvature values of muscles located on the surface. Note: the values of k_1 and k_2 are dependent upon the characteristics of the pneumatic muscles (for example, the length) as well as the pressure inside of the muscles. Further research is necessary to develop functions to calculate these values; our values were determined via experimentation. For two muscles that lie along the same axis, at different ends of the axis, the combined effect of equation (5) acts as a weighted average of the effects of two muscles. These effects are conveyed by the monotonic $g(\bullet) \in [0, 1]$. For cubic interpolation, the term $g(\frac{z}{z_{zmax}})$ corresponds to the effect given by an inducing muscle (when unpressurized) that lies along either the negative y-axis at $z = 0$ or along the z-axis at $y = 0$; and, the term $g(1 - \frac{z}{z_{max}})$ reflects the effect given by an inducing muscle that lies along the negative y-axis at $z = z_{max}$ or along the z-axis at $y = y_{min}$. The effect is reversed for exponential interpolation.

We adopt the convention that $k_1 \in \mathbb{R}$ will always represent the greater curvature value; so, in (5), the curvature of a given muscle will either multiply the $g(\frac{z}{z_{zmax}})$ term or the $g(1 - \frac{z}{z_{max}})$ term, depending on the location of the muscle with the greatest curvature. Note that this is only for cases where two muscles are parallel. Additionally, the term parallel muscles refers to two parallel muscles in the sense that the muscles lie parallel to each other on the plane of an unactuated surface.

B. Phantom Muscle Concept

A phantom muscle is defined as an imaginary muscle with a curvature value that is a fraction of the curvature value of an actual muscle (1/4 the value of the muscle's curvature in our subsequent examples) and lies at the side of the surface opposite from the actual muscle. When only one muscle is present, it was observed in practice that the actuated muscle induces a slight curvature at the side of the surface opposite from the muscle where the curvature value is assumed to be zero by simple interpolation. Thus,

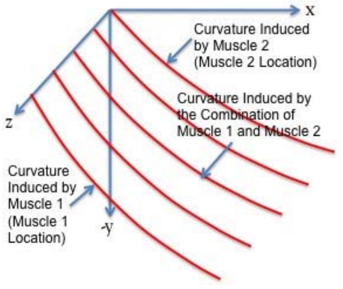

x
Curvature Induced by Muscle 2 (Muscle 2 Location)
Curvature Induced by the Combination of Muscle 1 and Muscle 2
z
Curvature Induced by Muscle 1 (Muscle 1 Location)
-y

Fig. 3. Representation of the Curves Induced Across a Surface

to analytically represent this, a 'phantom muscle' is used. Utilizing the adopted convention expressed in Section III A, k_1 will influence the function that corresponds to the location of the real muscle, and k_2 will influence the function that corresponds to the location of the phantom muscle.

1) Cubic Interpolation: For two muscle locations and the parallel muscle arrangement, it was empirically determined that cubic interpolation is an effective means by which the induced curvature values for the surface could be calculated. We use the cubic function[1] $g(z) = 2|z|^3 - 3z^2 + 1$.

The surface interpolation function $k_c(z)$ becomes:

$$k_c(z) = (2|\frac{z}{z_{max}}|^3 - 3\frac{z}{z_{max}}^{2} + 1)k_1$$
$$+(2|\frac{z_{max} - z}{z_{max}}|^3 - 3(\frac{z_{max} - z}{z_{max}})^2 + 1)k_2\ . \tag{6}$$

For the parallel muscle arrangement shown in Figure 4, k_1 and k_2 directly represent the curvatures of the muscles. However, a more general function is developed for muscles that lie along one of the axes and somewhere between the boundaries of the surface. For muscle locations where the muscle lies in the center of the surface, a piecewise function is used to represent the overall surface. When a muscle does not lie on an edge of the surface, the muscle splits the surface into two parts. One portion will utilize an interpolation function where k_1 is the gain for the $g(\frac{z}{z_{max}})$ term, and for the other, k_1 will be the gain for the $g(1-\frac{z}{z_{max}})$ term; therefore, two phantom muscles will be required. The interpolation function for these muscle locations is:

$$k_{c_{mid}} = \begin{cases} g(\frac{z}{z_{loc}})k_2 + g(\frac{z_{loc}-z}{z_{loc}})k_1 \\ \qquad\qquad z \in [0, z_{loc}] \\ g(\frac{z-z_{loc}}{z_{max}-z_{loc}})k_1 + g(\frac{z_{max}-z}{z_{max}-z_{loc}})k_2 \\ \qquad\qquad z \in [z_{loc}, z_{max}], \end{cases} \tag{7}$$

where $z_{loc} \in [0, z_{max}]$ is the z-coordinate of the muscle.

2) Exponential Interpolation: While testing some physical prototypes, it was observed that there are cases where muscle locations induce curvatures within the surface whose values take a shape that could be better represented by an exponential function. In these cases, only one real muscle (this muscle must lie along the edge of the surface) will be inducing curvatures along the surface that are parallel to this real muscle. Therefore, only one curvature value, that of the inducing muscle, will be known. For these muscle locations, the other curvature value will be that of the phantom muscle. The exponential interpolation function is:

$$k_e(z) = \frac{z}{z_{max}}\frac{e^{\frac{z}{z_{max}}}}{e^1}k_1 + \frac{z_{max} - z}{z_{max}}\frac{e^{\frac{z_{max}-z}{z_{max}}}}{e^1}k_2, \tag{8}$$

where $k_e(z) \in \mathbb{R}$.

Depending on the muscle location, k_1 and k_2 will be associated with different portions of the interpolation function. If k_1 is the gain for $g(\frac{z}{z_{max}})$, the exponential function will be denoted as $k_{e_{max}} \in \mathbb{R}^+$. If k_1 is the gain for $g(1-\frac{z}{z_{max}})$, the exponential function will be denoted as $k_{e_0} \in \mathbb{R}^+$.

Our models thus far have been developed for when there is only one muscle on the surface or for the case of parallel muscles. When multiple muscles are added to the surface to form non-parallel muscle arrangements, the superposition principle is employed. (For each point along the surface, the effects incurred from one muscle are added to the effects incurred from the other muscle at that same point.)

IV. Examples

In this section, we examine and assess the effectiveness of the core kinematic models by comparing a variety of muscle arrangements that actuate a physical square continuum surface with the associated three-dimensional models. We detail the kinematics for each muscle arrangement.

The physical prototypes developed for this testing were created by sewing off-the-shelf McKibben muscles, which were built in our lab, onto mattress foam. Different flexible surfaces were tested, and it was determined that the mattress foam yielded the best performance.

A. Parallel Muscle Arrangement

The kinematics for this muscle arrangement are (Fig. 4):

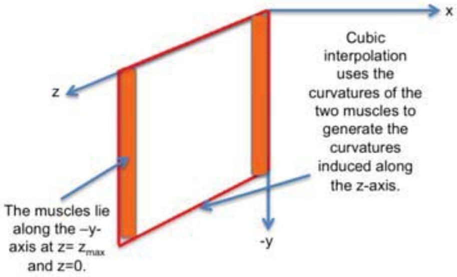

x
z
-y
Cubic interpolation uses the curvatures of the two muscles to generate the curvatures induced along the z-axis.
The muscles lie along the –y-axis at z= z_max and z=0.

Fig. 4. Parallel Muscle Arrangement

$$x_{parallel}(z, s) = \int_0^s sin(\int_0^\sigma k_c(z)\,dz)\,d\sigma \tag{9}$$

$$y_{parallel}(z, s) = \int_0^s -cos(\int_0^\sigma k_c(z)\,dz)\,d\sigma, \tag{10}$$

where $x_{parallel} \in \mathbb{R}$ and $y_{parallel} \in \mathbb{R}$.

B. Perpendicular Muscle Arrangement

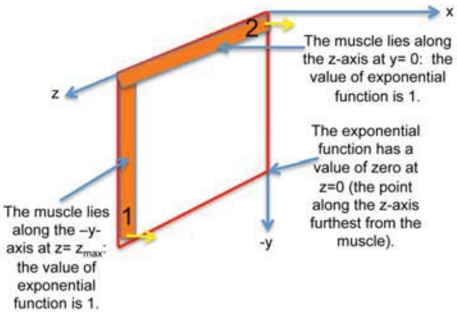

x
z
-y
2
1
The muscle lies along the z-axis at y= 0: the value of exponential function is 1.
The exponential function has a value of zero at z=0 (the point along the z-axis furthest from the muscle).
The muscle lies along the –y-axis at z= z_max: the value of exponential function is 1.

Fig. 5. Perpendicular Muscle Arrangement

[1] Credit for this interpolation function is given to David W. Jacobs, at the University of Maryland.

This muscle arrangement superimposes the x-coordinate values for muscles 1 and 2, which are at locations (0,z_{max}, 90) and (0,z_{max},180), respectively. Notice that muscle 2 in this arrangement produces curves that are slightly different to the curves described by the modified kinematic equations in Section II. Therefore, it is necessary to develop the kinematic formulas for scenarios where a muscle is located at $(y, z_{max}, 180)$, where $y \in (0, y_{min})$. The key differences between curves generated by a muscle in this location and one in the location seen in the base kinematic model is that the muscle lies along the $z-axis$, and it is shifted upward by z_{max}. So, the kinematics for muscles in this location are:

$$x(s) = \int_0^{(z_{max}-s)} sin(\int_0^{\sigma} k(v)\,dv)\,d\sigma \tag{11}$$

$$z(s) = z_{max} - \int_0^{(z_{max}-s)} cos(\int_0^{\sigma} k(v)\,dv)\,d\sigma. \tag{12}$$

The kinematic equations for muscle 1 are (Figure 5):

$$x_{(0,z_{max},90)}(s_y, s_z) = \int_0^{s_y} sin(\int_0^{\sigma} k_{e_{max}}(s_z)\,dy)\,d\sigma \tag{13}$$

$$y_{(0,z_{max},90)}(s_y, s_z) = \int_0^{s_y} -cos(\int_0^{\sigma} k_{e_{max}}(s_z)\,dy)\,d\sigma \tag{14}$$

$$z_{(0,z_{max},90)}(s_y, s_z) = s_z. \tag{15}$$

The kinematics for muscle 2 are:

$$x_{(0,z_{max},180)}(s_y, s_z) = \int_0^{(z_{max}-s_z)} sin(\int_0^{\sigma} k_{e_0}(s_y)\,dz)\,d\sigma \tag{16}$$

$$z_{(0,z_{max},180)}(s_y, s_z) = \left\{ \begin{array}{l} z_{max} \\ -\int_0^{(z_{max}-s_z)} cos(\int_0^{\sigma} k_{e_0}(s_y)\,dz)\,d\sigma \end{array} \right. \tag{17}$$

$$y_{(0,z_{max},180)}(s_y, s_z) = s_y. \tag{18}$$

In this muscle arrangement, only the x-direction is influenced by both muscles; therefore, the x-components are superimposed, and the kinematics for this muscle arrangement are:

$$x_{perp}(s_y, s_z) = x_{(0,z_{max},90)} + x_{(0,z_{max},180)} \tag{19}$$

$$y_{perp}(s_y, s_z) = y_{(0,z_{max},90)} \tag{20}$$

$$z_{perp}(s_y, s_z) = z_{(0,z_{max},180)}. \tag{21}$$

C. Plus-Shaped Muscle Arrangement

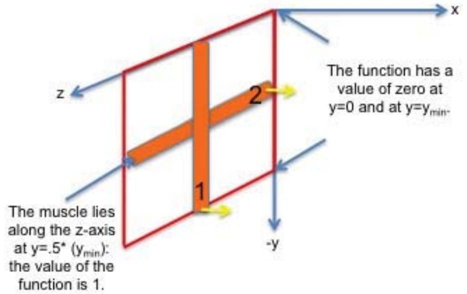

x
z
2
The function has a value of zero at y=0 and at y=y$_{min}$.
1
The muscle lies along the z-axis at y=.5* (y$_{min}$): the value of the function is 1.
-y

Fig. 6. Plus-Shape Muscles

This muscle arrangement also superimposes the effects on the x-component from muscles 1 and 2, which are at locations (0,0.5z_{max}, 90) and (0.5y_{min}, z_{max},180), respectively. The kinematic equations for muscle 1 are (Figure 6):

$$x_{(0,0.5z_{max},90)}(s_y, s_z) = \int_0^{s_y} sin(\int_0^{\sigma} k_{c_{mid}}(s_z)\,dy)\,d\sigma \tag{22}$$

$$y_{(0,0.5z_{max},90)}(s_y, s_z) = \int_0^{s_y} -cos(\int_0^{\sigma} k_{c_{mid}}(s_z)\,dy)\,d\sigma \tag{23}$$

$$z_{(0,0.5z_{max},90)}(s_y, s_z) = s_z. \tag{24}$$

The kinematics for muscle 2 are:

$$x_{(0.5y_{min},z_{max},180)}(s_y, s_z) = \int_0^{(z_{max}-s_z)} sin(\int_0^{\sigma} k_{c_{mid}}(s_y)\,dz)\,d\sigma \tag{25}$$

$$z_{(0.5y_{min},z_{max},180)}(s_y, s_z) = z_{max} - \int_0^{(z_{max}-s_z)} cos(\int_0^{\sigma} k_{c_{mid}}(s_y)\,dz)\,d\sigma \tag{26}$$

$$y_{(0.5y_{min},z_{max},180)}(s_y, s_z) = s_y. \tag{27}$$

In this muscle arrangement, the x-direction is the direction influenced by both muscles; therefore, the x-components are superimposed, and the kinematics for this arrangement are:

$$x_{plus}(s_y, s_z) = x_{(0,0.5z_{max},90)} + x_{(0.5y_{min},z_{max},180)} \tag{28}$$

$$y_{plus}(s_y, s_z) = y_{(0,0.5z_{max},90)} \tag{29}$$

$$z_{plus}(s_y, s_z) = z_{(0.5y_{min},z_{max},180)}. \tag{30}$$

D. ϕ-Degree Angle Muscle Arrangement

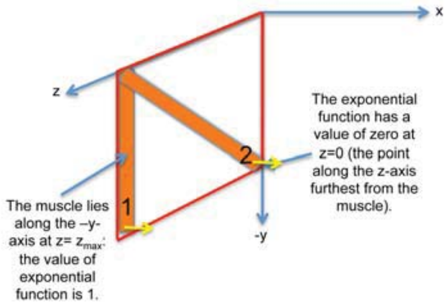
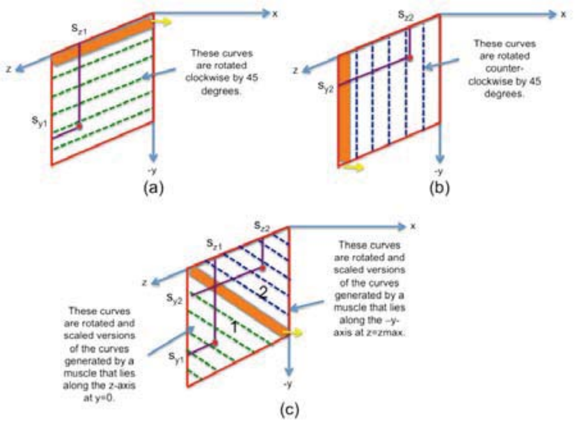

x
z
The exponential function has a value of zero at z=0 (the point along the z-axis furthest from the muscle).
2
1
The muscle lies along the –y-axis at z= z$_{max}$: the value of exponential function is 1.
-y

Fig. 7. 135-Degree Angle Muscle Arrangement

The kinematic model for muscles located at nonparallel or nonorthogonal angles is more complex. Consider a muscle that lies at ϕ degrees - ϕ was defined in Section II A. In this case, the curves induced by the muscle are not parallel to the major axes of the plane in which the surface lies. So, it was envisioned that the induced curves under the actuation of the ϕ-degree muscle can be thought of as being rotated versions of two different sets of induced curves, and we assume that each set occupies one half of the surface (Figure 8).



S_{z1}
x
z
These curves are rotated clockwise by 45 degrees.
S_{y1}
-y
(a)
S_{z2}
x
z
S_{y2}
These curves are rotated counter-clockwise by 45 degrees.
-y
(b)
S_{z2}
S_{z1}
x
z
S_{y2}
2
1
S_{y1}
These curves are rotated and scaled versions of the curves generated by a muscle that lies along the z-axis at y=0.
These curves are rotated and scaled versions of the curves generated by a muscle that lies along the –y-axis at z=zmax.
-y
(c)

Fig. 8. Breakdown of the 135-Degree Angle Kinematics

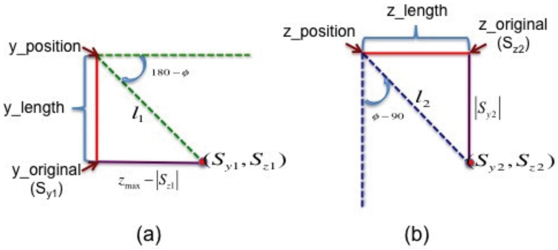


Fig. 9. How to Calculate the Axis Intercepts for Interpolation Calculation

Figure 8, which shows a muscle with $\phi = 135$, illustrates that the curves induced by muscle 2 in Figure 7 can be viewed as rotated versions of the curves induced by muscles located at (0,z_{max},180) and (0,z_{max},90). Each set of induced curves occupies a portion of the surface, which is divided by the line $y = -\frac{z}{tan((\phi-90)^\circ)} + \frac{z_{max}}{tan((\phi-90)^\circ)}$. When given the point on the surface[2] (s_y, s_z) - where $s_y \in [0, y_{min}]$ and $s_z \in [0, z_{max}]$ are the arclengths along the muscle in the y and z directions, respectively - for which we want to calculate the curvature, we are required to do one of two steps, dependent upon in which region the point lies: 1) calculate the y-coordinate, which we will denote as $y_{position}$, at which the rotated line (on which the point lies) intersects the line $z = z_{max}$ if the point lies in region 1 or 2) calculate the z-coordinate, which we will denote as $z_{position}$, at which the rotated line (on which the point lies) intersects the line $y = 0$ if the point lies in region 2. These concepts are visually presented in parts (a) and (b) of Figure 9 for a muscle with $\phi = 135$, respectively. The intersection point, which is denoted by either $y_{position}$ or $z_{position}$ based on in which region the point lies, is derived as either:

$$y_{length} = (z_{max} - |s_{z1}|) * tan((180-\phi)^\circ) \tag{31}$$

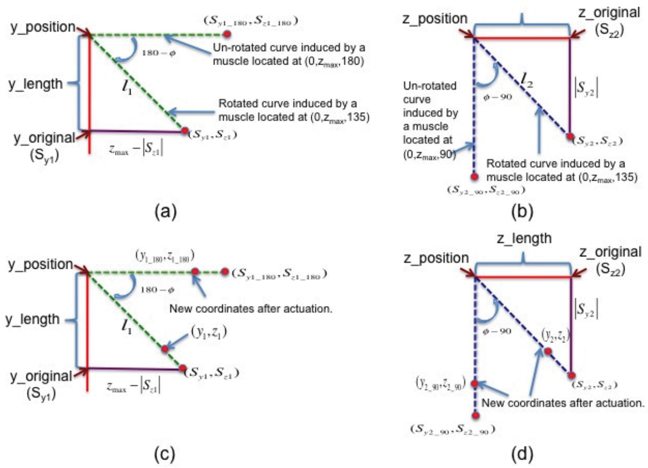
$$y_{position} = s_{y1} - y_{length} \tag{32}$$

or

$$z_{length} = |s_{y2}| * tan((\phi-90)^\circ) \tag{33}$$

$$z_{position} = s_{z2} + z_{length}, \tag{34}$$

where $y_{length} \in \mathbb{R}$, $y_{position} \in \mathbb{R}$, $z_{length} \in \mathbb{R}$, and $z_{position} \in \mathbb{R}$.



y_position
(S_{y1_180}, S_{z1_180})
$180-\phi$
Un-rotated curve induced by a muscle located at (0,z_{max},180)
y_length
l_1
Rotated curve induced by a muscle located at (0,z_{max},135)
y_original (S_{y1})
$z_{max}-|S_{z1}|$
(S_{y1}, S_{z1})
(a)
z_position
z_original (S_{z2})
l_2
$\phi-90$
$|S_{y2}|$
Un-rotated curve induced by a muscle located at (0,z_{max},90)
(S_{y2}, S_{z2})
Rotated curve induced by a muscle located at (0,z_{max},135)
(S_{y2_90}, S_{z2_90})
(b)
y_position
(y_{1_180}, z_{1_180})
(S_{y1_180}, S_{z1_180})
$180-\phi$
New coordinates after actuation.
y_length
l_1
(y_1, z_1)
y_original (S_{y1})
$z_{max}-|S_{z1}|$
(S_{y1}, S_{z1})
(c)
z_length
z_position
z_original (S_{z2})
$|S_{y2}|$
$\phi-90$
(y_2, z_2)
(y_{2_90}, z_{2_90})
(S_{y2}, S_{z2})
New coordinates after actuation.
(S_{y2_90}, S_{z2_90})
(d)

Fig. 10. Calculating the Arclengths of the Generated Curves Calculation

Thus, the interpolation function is based on the following:

$$\begin{aligned}&\text{If } s_y > -\frac{s_z}{tan((\phi-90)^\circ)} + \frac{z_{max}}{tan((\phi-90)^\circ)},\\ &\quad\text{then } \ k_{e_\phi}(s_y, s_z) = k_{e_0}(y_{position});\\ &\text{if } s_y \le -\frac{s_z}{tan((\phi-90)^\circ)} + \frac{z_{max}}{tan((\phi-90)^\circ)},\\ &\quad\text{then } \ k_{e_\phi}(s_y, s_z) = k_{e_{max}}(z_{position}),\end{aligned} \tag{35}$$

where $k_{e_\phi}(s_y, s_z) \in \mathbb{R}$.

Typically, once we have the interpolation function, we can use that function in conjunction with the kinematic formulas generated earlier. However, the basic kinematic models do not take into account that each curve generated by the ϕ-degree muscle is not the same length; this is why simply rotating the kinematics that correspond to a muscle located at $(0, z_{max}, 90)$ by ϕ degrees around the x-axis will not suffice. Therefore, we must perform extra steps to calculate the length, which will be denoted as either l_1 or l_2, of the curve on which the point on the surface in question lies; that calculated length will then be used to calculate the corresponding arc lengths of the unrotated induced curve, which will be denoted as (s_{y1_180}, s_{z1_180}) if the point is in region 1 or (s_{y2_90}, s_{z2_90}) if the point is in region 2. From parts (a) and (b) of Figure 10, one can see that the lengths of the induced curves for regions 1 and 2, respectively, are calculated as:

$$l_1 = \frac{z_{max} - |s_{z1}|}{cos((180-\phi)^\circ)} \tag{36}$$

$$l_2 = \frac{|s_{y2}|}{cos((\phi-90)^\circ)}, \tag{37}$$

where $l_1 \in \mathbb{R}^+$ and $l_2 \in \mathbb{R}^+$. From this, s_{y1_180} and s_{z2_90} become $y_{position}$ and $z_{position}$, respectively, and s_{z1_180} and s_{y2_90} become $z_{max} - l_1$ and l_2, respectively.

[2] Note, we will represent a point that lies in region 1 as (s_{y1}, s_{z1}) and a point that lies in region 2 as (s_{y2}, s_{z2}).

Once the corresponding arc lengths on the unrotated induced curves have been calculated, we can use the kinematics for a muscle located at $(0, z_{max}, 180)$ (region 1) or the kinematics for a muscle located at $(0, z_{max}, 90)$ (region 2) to calculate the y- or z-coordinate post-actuation for the unrotated induced curves; if the point lies in region 1, the coordinates will be denoted as (y_{1_180}, z_{1_180}), and if the point lies in region 1, the coordinates will be denoted as (y_{2_90}, z_{2_90}). Lastly, we must rotate the coordinates for the unrotated induced curves to their new coordinates on the ϕ-degree rotated curve. From parts (c) and (d) of Figure 10 we can see how the coordinates are rotated:

$$\begin{aligned} y_1 = (z_{(0,z_{max},180)}(s_{y1\_180}, s_{z1\_180}) - z_{max}) sin((180-\phi)^\circ) \\ -(y_{position}) \end{aligned} \tag{38}$$

$$\begin{aligned} z_1 = z_{max} + (z_{(0,z_{max},180)}(s_{y1\_180}, s_{z1\_180}) \\ -z_{max}) cos((180-\phi)^\circ), \end{aligned} \tag{39}$$

and

$$y_2 = y_{(0,z_{max},90)}(s_{y2\_90}, s_{z2\_90}) cos((\phi-90)^\circ) \tag{40}$$

$$z_2 = z_{position} + y_{(0,z_{max},90)}(s_{y2\_90}, s_{z2\_90}) sin((\phi-90)^\circ). \tag{41}$$

The actuation of muscle 2 induces movement in the x-, negative y-, and z-directions. For this muscle, the x-, negative y-, and z- displacements are dependent upon whether the induced curve lies in the lower left or upper right half of the surface. For muscle 1, displacement only occurs in the negative y- and x-directions as previously discussed. Therefore, the kinematics for the overall surface will take the displacements for each muscle in the x- and negative y-directions and superimpose them, and the z-coordinate will be the calculated z-coordinate for muscle 2. So, the kinematics for this arrangement are:

$$\begin{aligned} &\text{If } s_y > -\tfrac{s_z}{tan((\phi-90)^\circ)} + \tfrac{z_{max}}{tan((\phi-90)^\circ)}, \\ &\text{then } \; x_\phi(s_y, s_z) = \textstyle\int_0^{l_1} sin(\int_0^\sigma k_{e_\phi}(y_{position})\, dz)\, d\sigma \\ &\qquad + x_{(0,z_{max},90)}(s_{y1}, s_{z1}); \\ &\text{if } s_y \le -\tfrac{s_z}{tan((\phi-90)^\circ)} + \tfrac{z_{max}}{tan((\phi-90)^\circ)}, \\ &\text{then } \; x_\phi(s_y, s_z) = \textstyle\int_0^{l_2} sin(\int_0^\sigma k_{e_\phi}(z_{position})\, dy)\, d\sigma \\ &\qquad + x_{(0,z_{max},90)}(s_{y2}, s_{z2}), \end{aligned} \tag{42}$$

where $x_\phi(s_y, s_z) \in \mathbb{R}$.

$$\begin{aligned} &\text{If } s_y > -\tfrac{s_z}{tan((\phi-90)^\circ)} + \tfrac{z_{max}}{tan((\phi-90)^\circ)}, \\ &\text{then } \; y_\phi(s_y, s_z) = y_1 + (s_y + y_{(0,z_{max},90)}(s_{y1}, s_{z1})); \\ &\text{if } s_y \le -\tfrac{s_z}{tan((\phi-90)^\circ)} + \tfrac{z_{max}}{tan((\phi-90)^\circ)}, \\ &\text{then } \; y_\phi(s_y, s_z) = y_2 + (s_y + y_{(0,z_{max},90)}(s_{y2}, s_{z2})), \end{aligned} \tag{43}$$

where $y_\phi(s_y, s_z) \in \mathbb{R}$.

$$\begin{aligned} &\text{If } s_y > -\tfrac{s_z}{tan((\phi-90)^\circ)} + \tfrac{z_{max}}{tan((\phi-90)^\circ)}, \\ &\text{then } \; z_\phi(s_y, s_z) = z_1, \\ &\text{if } s_y \le -\tfrac{s_z}{tan((\phi-90)^\circ)} + \tfrac{z_{max}}{tan((\phi-90)^\circ)}, \\ &\text{then } \; z_\phi(s_y, s_z) = z_2, \end{aligned} \tag{44}$$

where $z_\phi(s_y, s_z) \in \mathbb{R}$.

V. Quantitative Analysis

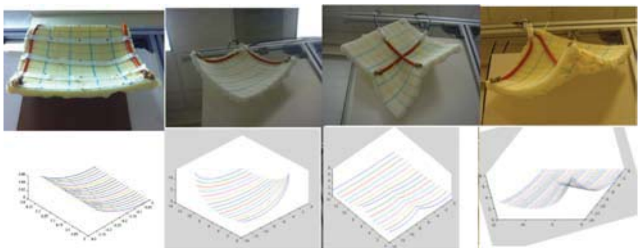


Fig. 11. Configurations of Continuum Surface Hardware (top row), with Results of Corresponding Kinematic Model Predictions (bottom row), for Key Configurations Reported in Section IV

In order to assess the precision of our continuum surface kinematics, we performed quantitative analysis using the Microsoft Kinect sensor. For this, 25 points were selected on the surface; each point was either 8 or 10 cm from the previous point on a 36 cm-sided square surface. For each muscle arrangement, two sets of data were collected. The first set measured the depth (the distance of the point on the surface from the sensor) of the unactuated surface. The second set, similarly, measured the depth of the points on the actuated surface by using a computer program that allowed the user to choose feature points in two different image frames. The clicks in the first image frame captured the depth for the unactuated surface at each point, and those in the second image frame captured the depth for the actuated surface at each point. The sensor was positioned directly in front of, at the same height as, and parallel to the surface.

To calculate the distance that each point on the surface had moved in the x-direction (the height the surface moved), the actuated data was subtracted from the unactuated data. Once the x-distance data had been calculated for each point, the x-distances for the same points were calculated using the appropriate kinematic model (muscle arrangement dependent); then, the mean square error (MSE) between them was calculated. Additionally, the MSE between the same kinematic data and the physical data for an unactuated surface, which would have x-distances of zero, was calculated for comparison; this MSE would represent the worst-case scenario and should, therefore, be larger than the MSE for the actuated surface. Table I shows an analysis of the data.

TABLE I

Mean Squared Error Between the Kinematic Height Data and the Physical Height Data When the Surface is Unactuated and Actuated (MSE) [m^2]

Muscle Arrangement	Perpendicular	Parallel	$\phi = 30^\circ$	Plus
Unactuated Surface	0.2896	0.2135	0.3480	0.4005
Actuated Surface	0.0958	0.0459	0.2204	0.2895

From Table I, the model/hardware MSEs are small, relative to the reference MSE for the flat, unactuated surface. This shows that the kinematics for the different muscle arrangements approximate the height data (x-coordinate) well. The y- and z-coordinate data is consistent with the kinematics developed by [39] and was, therefore, not taken into account.

However, the kinematics for the ϕ° and Plus arrangements did not approximate as well as those for the parallel and perpendicular arrangements. For the ϕ° and Plus arrangements, the increased discrepancy is likely due not only to the complexity of the shape of the surfaces after actuation but also to the assumptions made for the interpolation equations. Experimentation and fine tuning of the interpolation functions are likely to improve these results. More generally, the curvature value of the mathematical model is an approximation of the curvature value of the physical surface; a slight mismatch between the actual and approximated curvature values affects the degree to which the physical and kinematic data match. Also, as with any sensor, the precision can be an issue. Lastly, and more importantly, two main assumptions were made about the surface kinematics, which simplify the mathematics, and, thereby, introduce error into the results: constant curvature and constant length.

VI. Conclusion

In this paper, we have introduced new (forward) kinematic models for continuum robot surfaces. Although the models do not fit the physical surfaces exactly, the results illustrate that the developed models approximate the shape of the physical continuum surface models very well. In the near future, we plan to develop the kinematics for arbitrary muscle arrangements, to increase kinematic precision, and to begin inverse kinematics development.

References

[1] T. Zheng, D. T. Branson, E. Guglielmino, and D. G. Caldwell, "Sheetbot: Two-dimensional sheet-like robot as a tool for constructing universal decentralized control systems," in *ICRA*, 2012, pp. 3733–3738.

[2] K. Oosterhuis, *Hyperbodies: Towards an E-motive Architecture*. Basel, Switzerland: Birkauser, 2003.

[3] P. Shuch, "Multiband reconfigurable synthetic aperture radar antenna," in *Earth Science Technology Conference*, 2004, pp. 1–5.

[4] H. Booth, "UT Arlington engineer developing biomask to aid soldiers recovering from facial burns," January 2012. [Online]. Available: http://www.uta.edu/news/releases/2012/01/moss-biomask-research

[5] J. O. Brooks, I. D. Walker, K. E. Green, J. Manganelli, J. Merino, L. Smolentzov, T. Threatt, and P. M. Yanik, "Robot bedside environments for healthcare," in *ISPRA*, 2012, pp. 32–37.

[6] G. Robinson and J. Davies, "Continuum robots - a state of the art," in *ICRA*, Detroit, MI, 1999, pp. 2849–2854.

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

[8] I. Gravagne and I. Walker, "Kinematic transformations for remotely-actuated planar continuum robots," in *ICRA*, 2000, pp. 19–26.

[9] M. Ivanescu and V. Stoian, "A variable structure controller for a tentacle manipulator," in *ICRA*, Nagoya, Japan, 1995, pp. 3155–3160.

[10] B. Jones and I. Walker, "Kinematics of multisection continuum robots," *IEEE Trans. Robot.*, vol. 22, no. 1, pp. 43–57, Feb. 2006.

[11] G. Gallot and O. I. W. Khalil, "Dynamic modeling and simulation of a 3-d eel-like robot," in *ICRA*, Rome, Italy, 2007, pp. 1486–1491.

[12] N. Li, T. Zhao, and Y. Zhao, *The Dynamic Modeling of Snake-Like Robot Using Nominal Mechanism Method/ Intel. Robot. and Appl.* Berlin-Heidelberg, Germany: Springer-Verlag, 2008.

[13] H. Mochiyama and T. Suzuki, "Kinematics and dynamics of a cable-like hyper-flexible manipulator," in *ICRA*, 2003, pp. 3672–3677.

[14] D. Trivedi, A. Lofti, and C. Rahn, "Geometricaly exact dynamic models for soft robotic manipulators," in *IROS*.

[15] I. S. Godage, D. Branson, E. Guglielmino, G. Medrano-Cerda, and D. Caldwell, "Shape function-based kinematics and dynamics for variable length continuum robotic arms," in *ICRA*, 2011, pp. 452–457.

[16] ——, "Novel modal approach for kinematics and multisection continuum arms," in *IROS*, San Francisco, USA, 2011, pp. 1093–1098.

[17] T. Kano, Y. Watanabe, and A. Ishiguro, "A 3d dynamic model for continuum robots inspired by an octopus arm," in *ICRA*, Shanghai, China, 2011, pp. 3652–3657.

[18] R. Kang, A. Kazakidi, E. Guglielmino, D. T. Branson, D. Tsakiris, J. A. Ekaterinaris, and D. G. Caldwell, "Dynamic model of a hyper-redundant, octopus-like manipulator for underwater applications," in *IROS*, San Francisco, USA, 2011, pp. 4054–4059.

[19] M. Calisti, M. Giorelli, G. Levy, B. Mazzolai, B. Hochner, C. Laschi, and P. Dario, "An octopus-bioinspired solution to movement and manipulation for soft robots," *Bioinsp. & Biomi.*, vol. 6, no. 3, Sep. 2011.

[20] D. Braganza, D. Dawson, I. Walker, and N. Nath, "A neural network controller for continuum robots," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1270–1277, Dec. 2006.

[21] B. Jones and I. Walker, "Practical kinematics for real-time implementation of continuum robots," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1087–1099, Dec. 2006.

[22] S. Hirose, *Biologically Inspired Robots*. Oxford, UK: Oxford University Press, 1993.

[23] W. McMahan, B. Jones, and I. Walker, "Robotic manipulators inspired by cephalopod limbs," *Jour. Eng. Des. Inno.*, vol. 1, no. P, p. 01P2, Jun. 2005.

[24] K. Suzimori, S. Iikura, and H. Tanaka, "Development of a flexible microactuator and its applications to robotic mechanisms," in *ICRA*, Sacramento, CA, 1991, pp. 1622–1627.

[25] H. Tsukagoshi, A. Kitagawa, and M. Segawa, "Active hose: An artificial elephant's nose with maneuverability for rescue operations," in *ICRA*, Seoul, South Korea, 2001, pp. 2454–2459.

[26] M. Grissom, V. Chitrakaran, D. Dienno, M. Csencsits, M. Pritts, B. Jones, W. McMahan, D. Dawson, C. Rahn, and I. Walker, "Design and experimental testing of the octarm soft robot manipulator," in *Proc. SPIE Conf. Unmanned Sys. Tech.*, Kissimee, FL, 2006, pp. 109–114.

[27] M. Hannan and I. Walker, "Kinematics and the implementation of an elephant trunk manipulator and other continuum style robots," *J. Robot. Sys.*, vol. 20, no. 2, p. 4563, Feb. 2003.

[28] G. Immega and K. Antonelli, "The ksi tentacle manipulator," in *ICRA*, Nagoya, Japan, 1995, pp. 3149–3154.

[29] D. Lane, B. Davies, G. Robinson, D. O'Brien, J. Sneddon, E. Seaton, and A. Elfstrom, "The amadeus dextrous subsea hand: Design, modeling, and sensor processing," *IEEE Jour. Ocean. Eng.*, vol. 24, no. 1, pp. 96–111, 1999.

[30] H. Ohno and S. Hirose, "Design of a slim slime robot and its gait of locomotion," in *IROS*, Maui, HI, 2001, pp. 707–715.

[31] R. Buckingham and A. Graham, "Snaking around a nuclear jungle," *Ind. Robot: An Int. Journal*, vol. 32, no. 2, pp. 120–127, Feb. 2005.

[32] G. Chen, T. Pu, T. Herve, and C. Prelle, "Design and modeling of a micro-robotic manipulator for colonoscopy," in *Proc. SICE Ann. Conf.*, Annecy, France, 2005, pp. 109–114.

[33] G. Chen, M. Pham, and T. Redarce, "Development and kinematic analysis of a silicon rubber bending tip for colonoscopy," in *IROS*, Beijing, China, 2006, pp. 168–173.

[34] "Flockwall," 2012. [Online]. Available: http://www.robotecture.com/showcase/flock-wall/

[35] K. Green, I. Walker, L. Gugerty, J. Witte, H. Houayek, M. Kwoka, J. Johnson, K. Teja, and N. Kuntzi, "A robotic wall and reconfigurable desk supporting working life in a digital society," in *Video Proc., IEEE/RSJ Int. Conf. on Intel. Robot. Sys.*, October 2009.

[36] M. Hosale and C. Kievid, "Digitally driven architecture," *Footprint*, pp. 55–67, Spring 2010.

[37] H. Houayek, K. Green, and I. Walker, *The Animated Work Environment: An Architectural-Robotic System for a Digital Society*. VDM-Verlag, 2009.

[38] "Supercity urban expansion joint," September 2010. [Online]. Available: http://vimeo.com/14696537

[39] G. Chirikjian and J. Burdick, "Selected applications and intrinsic kinematics of hyper-redundant robots," in *4th Amer. Nuc. Soc. Topical Meeting on Robotics and Remote Systems*, 1991, pp. 61–70.