Kinematics and the Implementation of an Elephant's Trunk Manipulator and Other Continuum Style Robots

Michael W. Hannan and Ian D. Walker
Department of Electrical and Computer Engineering
Clemson University
Clemson, South Carolina 29634
e-mail: mhannan@ces.clemson.edu; ianw@ces.clemson.edu

Received 7 April 2002; accepted 23 September 2002

Traditionally, robot manipulators have been a simple arrangement of a small number of serially connected links and actuated joints. Though these manipulators prove to be very effective for many tasks, they are not without their limitations, due mainly to their lack of maneuverability or total degrees of freedom. Continuum style (i.e., continuous "backbone") robots, on the other hand, exhibit a wide range of maneuverability, and can have a large number of degrees of freedom. The motion of continuum style robots is generated through the bending of the robot over a given section; unlike traditional robots where the motion occurs in discrete locations, i.e., joints. The motion of continuum manipulators is often compared to that of biological manipulators such as trunks and tentacles. These continuum style robots can achieve motions that could only be obtainable by a conventionally designed robot with many more degrees of freedom. In this paper we present a detailed formulation and explanation of a novel kinematic model for continuum style robots. The design, construction, and implementation of our continuum style robot called the elephant trunk manipulator is presented. Experimental results are then provided to verify the legitimacy of our model when applied to our physical manipulator. We also provide a set of obstacle avoidance experiments that help to exhibit the practical implementation of both our manipulator and our kinematic model. © 2003 Wiley Periodicals, Inc.

1. INTRODUCTION

Over the last several decades the research area of robotic manipulators has focused mainly on designs that resemble the human arm. These conventional robots can be best described as discrete manipulators,[1] where their design is based on a small number of actuatable joints that are serially connected by discrete rigid links (see Figure 1). Discrete manipulators havebeen proven to be very useful and effective for many different tasks, but they are not without their limitations. These robots most often have five to seven de-

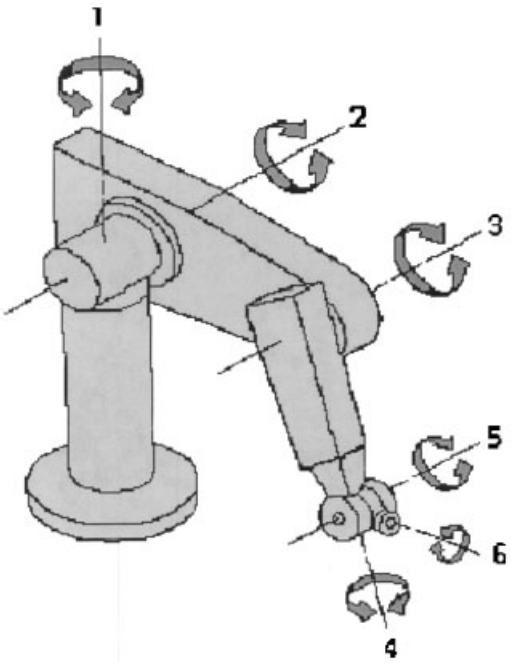

Figure 1. Discrete manipulator.

grees of freedom (DOF), thus in a spatial environ ment they will most often need all of their DOF just to position the end-effector. This design is very efficient for open environments, but as constraints are added to the environment it is possible that the manipulator can not reach its desired end-effector position. This failure is due to the lack of DOF in the robot to meet both the environmental constraint conditions and the desired end-effector position requirements. Another drawback of discrete manipulators is that they require some specialized device for manipulation of an object. Most often this manipulation is done by attaching a gripper/hand at the end of the robot. Once again this design works well in many cases, but it is possible that a strategy where the sections of a manipulator do the grasping, i.e., whole arm grasping, might be a better solution.

The addition of more DOF for these conventional manipulators can further enhance their maneuverability and flexibility. We can relate this situation to the diversity of biological manipulators. As noted earlier, discrete manipulators resemble the human arm in structure, but if we look at other biological manipulators we see that this is not the only design. Animals such as snakes, elephants, and octopuses can produce motions from their appendages or body that allow the effective manipulation of objects, even though they are very different in structure compared to the human arm. Even among these appendages, i.e., trunks, tentacles, etc., their physical structure can vary, but they all share the trait of having a relatively large number of DOF.

Research into manipulators with a large number of DOF or a high degree of maneuverability has spawned many different designs. The most popular design uses some type of "backbone" structure that is actuated by sets of cables.[2–6] Another design uses fluid filled tubes for actuation.[7,8] There are also other designs that are based on more conventional robot construction techniques.[9] These types of manipulators are commonly referred to as a hyper-redundant manipulators due to their number of *actuatable* DOF being much larger than the DOF of their intended workspace. Due to the vast design possibilities Robinson and Davies[1] developed three classifications for the different possible designs. As introduced earlier, the first class which describes conventional manipulators is termed *discrete* robots. As the redundancy/maneuverability of the manipulator increases over that of discrete manipulators by increasing the number of discrete joints, it moves into the second classification known as *serpentine* robots. This classification would include hyper-redundant manipulators. The third classification of robots, known as *continuum* robots, does not contain discrete joints and rigid links as in the previous two classifications. Instead the manipulator bends continuously along its length similar to that of biological trunks and tentacles.

The increased DOF that allows hyper-redundant and continuum robots to become more maneuverable has the adverse effect of complicating the kinematics for these manipulators. Chirikjian and Burdick[10–13] proposed a great deal of theory that has laid a foundation for the kinematics of hyper-redundant robots. Mochiyama *et al.*[14–16] presented research in the area of kinematics and the shape correspondence between a hyper-redundant robot and a desired spatial curve. More recently, Gravagne[17–19] has presented work in the kinematics for continuum robots.

This paper focuses on continuum style robots; this includes both continuum robots and the spectrum of serpentine robots that behave very similarly to continuum robots. We have developed a continuum style robot which resembles an elephant's trunk, see Figure 2, which is somewhat similar in structure to those developed in refs. 2–5. More importantly, we present here a kinematic model that not only applies to our robot, but also to most other types of continuum style robots. Our kinematic approach is unlike the other models presented by researchers such as Chirikjian, Mochiyama, and Gravagne, in that our model utilizes the concept of constant curvature

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

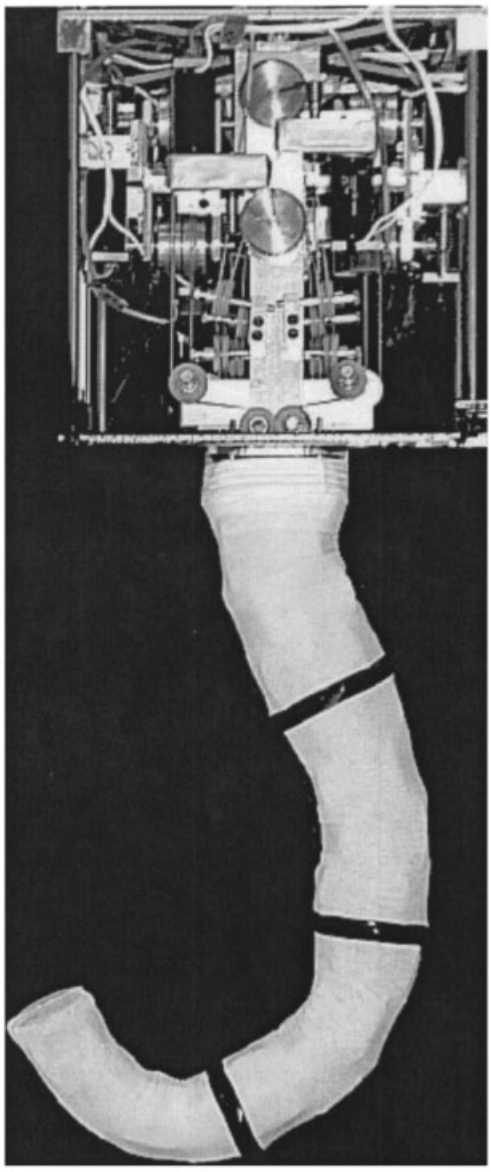

Figure 2. Elephant's trunk manipulator.

sections, and incorporates them through the use of differential geometry into a modified Denavit-Hartenberg procedure to determine the kinematics. The Denavit-Hartenberg procedure is the most commonly used approach for determining conventional robot kinematics. This provides a solid base for understanding and implementing our approach. The resulting form of our kinematic model allows the use of conventionally designed motion planning and redundancy resolution techniques. Our kinematic model can also be easily implemented on both planar and spatial continuum robots, and we present here experimental results with our continuum style robot that verify the viability of our kinematic model and its real time implementation. In essence, our approach allows the complex kinematics of continuum style robots to be expressed in such a way that can be easily understood and implemented by anyone familiar with conventional robotic manipulators.

2. CONTINUUM STYLE ROBOTS

First we would like to point out why we choose to distinguish between continuum style and hyper-redundant designs. Hyper-redundant robots are usually defined as having many more kinematically *actuatable* DOF than the number of DOF of the robot's workspace. However, continuum style robots do not have to fit this definition. Continuum style manipulators can exhibit a large number of kinematic DOF, but not all of these DOF are directly actuated. Therefore it is possible to have a continuum robot that is not technically a hyper-redundant robot. The design difference between conventional manipulators and continuum style manipulators is that the conventional manipulator is a serial connection of actuated joints. That is, every joint on the robot is actuated, and it is then rigidly connected to the next joint. On the other hand, continuum style manipulators are not designed this way. Consider a three degree of freedom section. The traditional approach would be to actuate all three joints, thus obtaining three actuated DOF. The continuum style robot would provide a torque at the tip of the robot, and then some type of coupling, i.e., springs, would be used to transmit the torque to the three joints. Thus, the available DOF are coupled in such a way that there is only one actuated DOF. Though this seems at first to be counter-productive, the kinematics generated by this configuration prove to be very beneficial in some cases.

A simple example is if the robot needs to reach around an obstacle (see Figure 3). The discrete manipulator must provide actuation to all three of the joints to reach around the obstacle, but the continuum robot can achieve the same motion with only one actuator instead of three. Expanding this concept to robots with hyper-DOF, then one can see the benefit of continuum robots. The complexity of the mechanical design, and the control for all of the actuators for a conventionally designed robot would be extremely difficult to implement. Using a continuum robot design we can achieve many of the same configurations, but the design can be greatly simplified.

In refs. 20 and 21 we outlined the most common

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

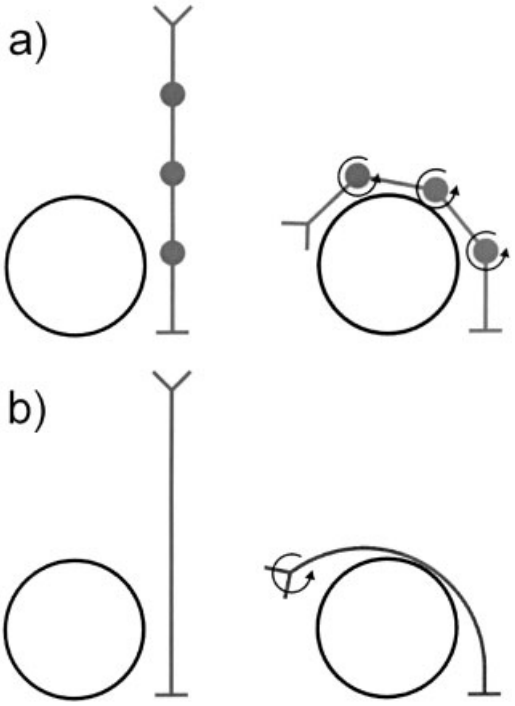

Figure 3. Obstacle example: (a) discrete manipulator and (b) continuum manipulator.

actuation and structural design strategies for continuum robots. The prevalent designs for actuation are cable/tendon systems and pressurized fluid systems. Both of these styles are remote actuation strategies where the bulk of the mechanisms used for actuation are not located directly on the robot. This is because the use of actuators located directly on the robot, as in conventional designs, proves to be extremely complicated and inefficient to implement. In the area of structural design there are also two main designs. In both cases the robot is based upon some type of "backbone" structure, where the backbone serves to determine both the manipulator's maneuverability and overall shape. The two backbone designs can be described as either being segmented or continuous in structure, where in either case the structure must contain a sufficient amount of "bending stiffness" for proper support and maneuverability.[21]

3. ELEPHANT'S TRUNK ROBOT

Though the classifications by Robinson and Davies[1] help to better describe the different styles of robotic manipulators, there are some possible manipulator configurations that do not readily fit into one of those classes. One such design is that of the elephant's trunk manipulator (see Figure 4). This design is more of a hybrid of all three classes. As described in detail in refs. 21 and 22, the fundamental structure of the robot is composed of four sections, where each section has two actuatable DOF which are actuated by a cable servo system. This yields a manipulator with eight actuatable DOF. Though this manipulator is redundant, it would more appropriately fall in to the discrete classification of manipulators. However, it is the construction of these sections that allows the robot to fit into more than one classification. The structure of each section is based on four very small links serial connected by four, two-DOF joints (see Figure

Figure 4. Elephant's trunk manipulator.

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

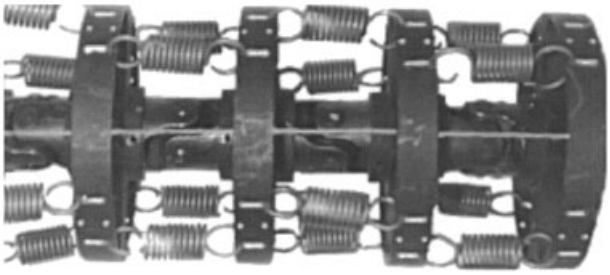

Figure 5. Tip section of the elephant's trunk manipulator.

5). There are elastic connections between each joint, i.e., springs, thus coupling all of the joints. This yields a total of 32 kinematic DOF for the manipulator, and it would seem to fall in to the serpentine classification. However, due to this combination of 32 coupled joints and eight cable actuatable DOF, the manipulator takes on the appearance of a continuum robot in the sense that each section of the robot will take on a curved appearance when actuated. When all the sections are viewed together the robot takes on a smooth continuous shape as a continuum robot would. This can be seen easily from Figure 4, and is thus the reason that we model the elephant's trunk robot as a continuum style robot.

4. KINEMATICS

In this section, we introduce a new technique for modeling the kinematics of continuum style robots. First we need to understand how the motions of continuum designs can be quantified. Unlike conventional manipulators the use of joint angles and link lengths does not provide an easy and physically realizable description of the manipulator's shape due to the continuous nature of the design. Instead a kinematic model that uses curvatures to describe the shape of the manipulator can provide a more physical and intuitive description of the manipulator. Before we apply this concept to the manipulator, we must first introduce some preliminary mathematical steps.

4.1. Fundamental Mathematics

Classical differential geometry[23] parametrizes a spatial curve C by its arc length s. The position vector $\mathbf{x}$ is defined in the spatial case to be $\mathbf{x}=[x,y,z]^T$. The unit tangent vector to C at $\mathbf{x}$ is $\mathbf{t}=d\mathbf{x}/ds$, the principal normal, $\mathbf{n}$, is placed such that $\mathbf{t}\cdot\mathbf{n}=\mathbf{0}$, and the binormal, $\mathbf{b}$, is defined as $\mathbf{b}=\mathbf{t}\times\mathbf{n}$. The collection of these three vectors at a point can be viewed as a new frame of reference. The formulas of Serret-Frenet describe

Figure 6. Planar continuum robot.

the motion of this frame as it changes along C. The three vector formulas can be written as

$$\frac{d\mathbf{t}}{ds}=\kappa\mathbf{n},\quad \frac{d\mathbf{n}}{ds}=-\kappa\mathbf{t}+\tau\mathbf{b},\ \text{and}\ \frac{d\mathbf{b}}{ds}=-\tau\mathbf{n}. \quad (1)$$

The scalar parameters κ and τ are called respectively the curvature and torsion which may be either positive or negative. An important note is that Eq. (1) defines τ uniquely, but only κ^2 not κ can be defined uniquely.

In the above, we have assumed that the curvatures and torsions have been a general function of arc length within a curve. If each section of a continuum robot is optimally constructed,[24] then each section will bend such that the curvature and torsion within that section is approximately constant. Given in Figure 6 is a picture of a planar continuum robot with an overlay of an arc with constant curvature. From inspection it can been seen that the curvature of the robot is approximately constant. Our work focuses on the description of robots that contain constant curvature and zero torsion sections. We proceed in this fashion due to the fact that the constant-curvature-only model best represents the experimental robots we are developing, and those designs which have been successfully implemented previously.[2–5]

4.2. Planar Curve Kinematics

In the following two sections we describe how the kinematics of a planar curve can be generated, in a form convenient for modeling continuum manipulators.

4.2.1. Differential Geometry

Under the assumption of constant curvature and that the torsion is zero for $\mathbf{x}\in\Re^2$, Eq. (1) can be written in the following linear differential equation,

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

$$\mathbf{t}'' + \kappa^2 \mathbf{t} = 0, \tag{2}$$

where the prime indicates differentiation with respect to s, i.e., $\mathbf{t}' = d\mathbf{t}/ds$. Upon solving Eq. (2) with the initial conditions: $\mathbf{t}_0 = \mathbf{t}(0)$ and $\mathbf{n}_0 = \mathbf{n}(0)$, the equation for the tangent vector as a function of s is

$$\mathbf{t}(s) = \mathbf{t}_0 \cos(\kappa\ s) + \mathbf{n}_0 \sin(\kappa\ s). \tag{3}$$

Integrating Eq. (3) from 0 to s yields the position vector

$$\mathbf{x}(s) = \frac{\mathbf{t}_0}{\kappa} \sin(\kappa\ s) + \frac{\mathbf{n}_0}{\kappa}\{1 - \cos(\kappa s)\}. \tag{4}$$

Using Eq. (1) and that the property that the Serret-Frenet frame is a right-hand frame, $\mathbf{n}$ can be formed by a rotation of $\pi/2$ about $\mathbf{b}$. If $\mathbf{t}_0$ is defined as $\mathbf{t}_0 = [t_{0x}, t_{0y}]^T$, then $\mathbf{n}_0$ can be written as $\mathbf{n}_0 = [-t_{0y}, t_{0x}]^T$. Substituting the previous two definitions into Eq. (4) we have

$$[x,y]^T = \frac{1}{\kappa}[t_{0x}, t_{0y}]^T \sin(\kappa\ s) + \frac{1}{\kappa}[-t_{0y}, t_{0x}]^T\{1 - \cos(\kappa\ s)\}. \tag{5}$$

The magnitude of $\mathbf{x}$ is given by

$$\|\mathbf{x}\| = \sqrt{x^2 + y^2}, \tag{6}$$

where the undefined terms are

$$\begin{aligned} x^2 &= \frac{1}{\kappa^2} t_{0x}^2 \sin^2(\kappa\ s) \\ &\quad - \frac{2}{\kappa^2} t_{0x} t_{0y} \sin(\kappa\ s)\{1 - \cos(\kappa\ s)\} \\ &\quad + \frac{1}{\kappa^2} t_{0y}^2 \{1 - 2\cos(\kappa\ s) + \cos^2(\kappa\ s)\}, \\ y^2 &= \frac{1}{\kappa^2} t_{0y}^2 \sin^2(\kappa\ s) \\ &\quad + \frac{2}{\kappa^2} t_{0x} t_{0y} \sin(\kappa\ s)\{1 - \cos(\kappa\ s)\} \\ &\quad + \frac{1}{\kappa^2} t_{0x}^2 \{1 - 2\cos(\kappa\ s) + \cos^2(\kappa\ s)\}. \end{aligned} \tag{7}$$

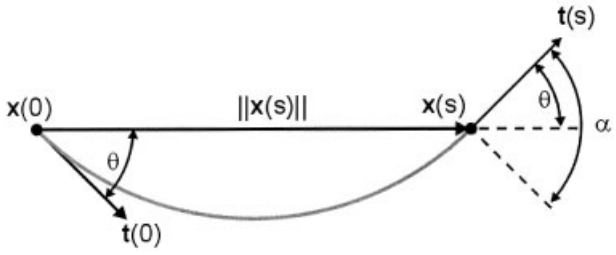

Figure 7. Magnitude and angle.

Substituting Eq. (7) into Eq. (6), and using the fact that $\|\mathbf{t}_0\| = 1$ we obtain

$$\|\mathbf{x}\| = \frac{\sqrt{2}}{\kappa} \sqrt{1 - \cos(\kappa\ s)}. \tag{8}$$

If θ is defined to be the angle between $\mathbf{x}$ and $\mathbf{t}_0$, as is shown in Figure 7, then

$$\sin(\theta) = \frac{\|\mathbf{x} \times \mathbf{t}_0\|}{\|\mathbf{x}\|\ \|\mathbf{t}_0\|} = \frac{\sqrt{2}}{2} \sqrt{1 - \cos(\kappa\ s)}, \tag{9}$$

$$\cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{t}_0}{\|\mathbf{x}\|\ \|\mathbf{t}_0\|} = \frac{\sqrt{2}\sin(\kappa\ s)}{2\sqrt{1 - \cos(\kappa\ s)}}. \tag{10}$$

Using Eq. (9), Eq. (10), and the trigonometric identity of $\sin(2\theta) = 2\sin(\theta)\cos(\theta)$ we obtain the relation

$$\theta = \frac{\kappa\ s}{2}. \tag{11}$$

Substituting Eq. (11) into Eq. (8) yields

$$\|\mathbf{x}\| = \frac{s}{\theta} \sin(\theta). \tag{12}$$

The position vector $\mathbf{x}$ can now be described by a magnitude, Eq. (12), and its angle, Eq. (11).[25] Note that Eq. (12) is a function of the angle θ which provides a coupling between the magnitude and angle of $\mathbf{x}$. Intuitively, as the sections bend, the net effect is that of bending a link of variable length.

4.2.2. Simple Geometry

Though the use of differential geometry gives use a very good understanding of the mechanics behind the curves motion, it is not the only way to arrive at

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

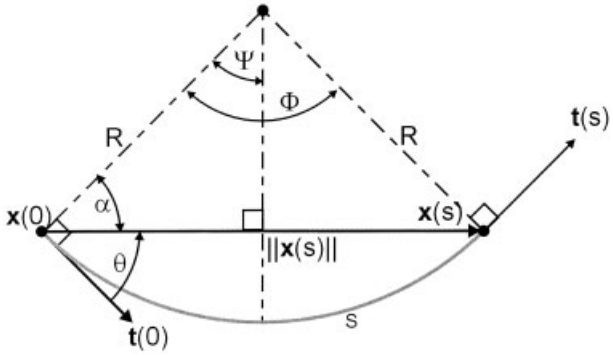

Figure 8. Standard geometry.

Eqs. (11) and (12). We can use some simple geometry to obtain the same results. Using the angles labeled in Figure 8 we have

$$\Phi = \frac{s}{R} = \kappa s, \tag{13}$$

$$\Psi = \frac{\Phi}{2} = \frac{\kappa s}{2}, \tag{14}$$

$$\alpha = \pi - \Psi, \tag{15}$$

$$\theta = \pi - \alpha = \Psi = \frac{\kappa s}{2}. \tag{16}$$

Thus we have determined Eq. (11). Again referring to Figure 8, we then have

$$\frac{\|\mathbf{x}\|}{2} = \frac{1}{\kappa}\sin(\Psi), \tag{17}$$

$$\|\mathbf{x}\| = \frac{2}{\kappa}\sin(\theta) = \frac{s}{\theta}\sin(\theta), \tag{18}$$

thus giving Eq. (12).

4.3. Planar Robot Kinematics

As discussed in refs. 25 and 20, using the above analysis the movement of a planar curve with constant curvature can be described by three coupled movements:

1. rotation by an angle θ,
2. translation by an amount of $\|\mathbf{x}\|$, and
3. rotation by the angle θ again,

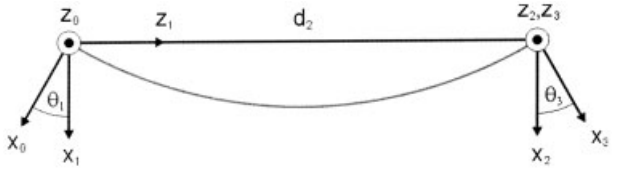

Figure 9. D-H frames for a planar section.

where **x** is defined to be the position vector of the endpoint of the curve. Now we have a description of the motion of the curve in terms of discrete movements. We can now apply a modified Denavit-Hartenberg (D-H) procedure for the forward kinematic analysis as would be done for conventional manipulators.[26] Thus, our approach allows one to model continuum robots using a procedure that is almost universally accepted and used. We term the D-H procedure as modified because the basis of the D-H procedure is that all of the joints are independent, but in our case the three motions are coupled through the curvature. In effect, we are modeling a complex one DOF motion by three simple, coupled motions. If the D-H frames for one section are set up as in Figure 9, then the D-H table is

link	θ	d	a	α
1	*	0	0	$-90°$
2	0	*	0	$90°$
3	*	0	0	$0°$

Using the above D-H table the homogeneous transformation matrix for the curve can be written in terms of the curvature κ and the total arc length l as

$$A_0^3 = \begin{bmatrix} \cos(\kappa l) & -\sin(\kappa l) & 0 & \frac{1}{\kappa}\{\cos(\kappa l) - 1\} \\ \sin(\kappa l) & \cos(\kappa l) & 0 & \frac{1}{\kappa}\sin(\kappa l) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{19}$$

4.4. Spatial Robot Kinematics

As shown in Ref. 27, the forward kinematics for a spatial robot can be easily generated from the planar case. After analyzing a spatial curve with only constant curvature and no torsion, the curve can be viewed as a planar curve rotated out of the plane. Using this concept, we can apply the planar kinematic technique with only the addition of an angle of rotation about the initial tangent. One possible way to

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

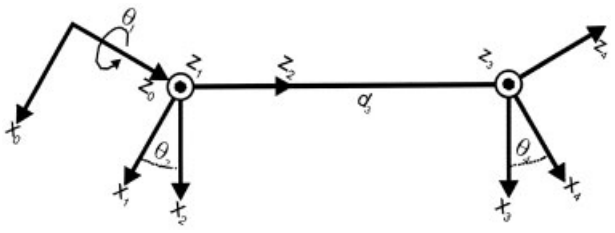

Figure 10. D-H frames for a spatial curve.

set up the frames for the spatial curve is given in Figure 10. By necessity, the frames were chosen so that frames 0 and 4 will have a direct relationship to the orientation of the tangents. This is done so the out of plane rotation will be about the tangent vector. The D-H table corresponding to Figure 10 is

link	θ	d	a	α
1	*	0	0	90°
2	*	0	0	−90°
3	0	*	0	90°
4	*	0	0	−90°

Using the table, where $\theta_1 = \phi$, $\theta_4 = \theta_2$, Eq. (11) for θ_2, and Eq. (12) for d_3 we obtain the transformation matrix in terms of the curvature κ, the total arc length l of the curve, and the rotation angle ϕ about the tangent as

$$A_0^4 = \begin{bmatrix} \cos(\phi)\cos(\kappa l) & -\sin(\phi) & -\cos(\phi)\sin(\kappa l) & -\frac{1}{\kappa}\cos(\phi) + \frac{1}{\kappa}\cos(\phi)\cos(\kappa l) \\ \sin(\phi)\sin(\kappa l) & \cos(\phi) & -\sin(\phi)\sin(\kappa l) & -\frac{1}{\kappa}\sin(\phi) + \frac{1}{\kappa}\sin(\phi)\cos(\kappa l) \\ \sin(\kappa l) & 0 & \cos(\kappa l) & \frac{1}{\kappa}\sin(\kappa l) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{20}$$

This homogeneous transformation matrix gives the forward kinematics for one section of a spatial continuum robot. Note that since κ can take on both positive and negative values, ϕ needs to only take on the values in a range of π to uniquely describe the curve in space.

The forward kinematics for an n section manipulator can then be generated by the product of n matrices of the form given in Eq. (20). The forward kinematics for our elephant trunk robot with its four sections can be calculated as

$$T_1^5 = T_1^2 \cdot T_2^3 \cdot T_3^4 \cdot T_4^5, \tag{21}$$

where $T_i^{i+1} = A_0^4$ for section i of the manipulator. Note that the total arc length for section $i = \{1,...,n-1\}$ must be used so that section n is properly oriented, but any arc length can be used for the final section depending on where the point of interest lies in the section. Using the total arc length for the final section gives the kinematics in terms of the end point of the section.

4.5. Prismatic Joint

Due to the fact that our kinematic model is based on the Denavit-Hartenberg procedure modeling, the addition of extra DOF is straightforward. To expand the usefulness and "flexibility" of the elephant's trunk robot we added a prismatic joint at its base.[32] To incorporate this joint into the kinematics only requires the multiplication of one extra matrix which has the following form,

$$T_0^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{22}$$

where d is the prismatic joint's amount of extension/retraction. Therefore, using Eq. (21), the forward kinematics for the robot are

$$T_0^5 = T_0^1 \cdot T_1^2 \cdot T_2^3 \cdot T_3^4 \cdot T_4^5. \tag{23}$$

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

4.6. Velocity Kinematics

Analogous to conventional kinematic analysis,[26] the velocity kinematics can be written as

$$\dot{\mathbf{x}} = J\dot{\mathbf{q}}, \tag{24}$$

where $\mathbf{x} \in \Re^{\mathbf{mx1}}$ is the task space vector, i.e., position and/or orientation, and the dot implies differentiation with respect to time. For our four section spatial robot

$$\mathbf{q} = [\mathrm{d}_0,\ \phi_1,\ \kappa_1,\ \phi_2,\ \kappa_2,\ \phi_3,\ \kappa_3,\ \phi_4,\ \kappa_4]^T. \tag{25}$$

The matrix J is the Jacobian, and is a function of the "joint" variables $\mathbf{q}$.

Conceptually, Eq. (23) is found explicitly in terms of sines, cosines, $\mathbf{q}$, etc., and then J is determined explicitly by taking the time derivative of the needed elements. For conventional robots the dimension of $\mathbf{q}$ is small, and the overall complexity of the forward kinematics is not overwhelming. However, for continuum robots the elements in the homogeneous transformation matrix can be extremely long. Though relatively simple and repetitive in nature the sheer number of terms in each element makes explicit determination of the elements for both the forward and inverse kinematics very inefficient. However, the numeric computation of these matrix elements is relatively simple. The forward kinematics can be computed in a straightforward manner by the matrix multiplication of separate matrices determined by numerically evaluating the matrix given by Eqs. (20) and (22).[27] The generation of the Jacobian matrix can then be found numerically through the use of the chain rule in differentiation. Differentiating Eq. (23) with respect to time yields

$$\dot{T}_0^5 = (\dot{T}_0^1 \cdot T_1^5) + (T_0^1 \cdot \dot{T}_1^2 \cdot T_2^5) + (T_0^2 \cdot \dot{T}_2^3 \cdot T_3^5) + (T_0^3 \cdot \dot{T}_3^4 \cdot T_4^5) + (T_0^4 \cdot \dot{T}_4^5). \tag{26}$$

The time derivative of the prismatic joint matrix given by Eq. (22) is obviously

$$\dot{T}_0^1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dot{d}_0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{27}$$

For each section $i = \{1,2,3,4\}$, the time derivative of Eq. (20) is

$$\dot{T}_i^{i-1} = \begin{bmatrix} -\sin(\phi_i)\cos(\kappa_i l_i)\dot{\phi}_i - \cos(\phi_i)\sin(\kappa_i l_i)\dot{\kappa}_i l_i & -\cos(\phi_i)\dot{\phi}_i & a_{13} & a_{14} \\ \cos(\phi_i)\cos(\kappa_i l_i)\dot{\phi}_i - \sin(\phi_i)\sin(\kappa_i l_i)\dot{\kappa}_i l_i & -\sin(\phi_i)\dot{\phi}_i & a_{23} & a_{24} \\ \cos(\kappa_i l_i)\dot{\kappa}_i l_i & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{28}$$

where the undefined elements are

$$a_{13} = \sin(\phi_i)\sin(\kappa_i l_i)\dot{\phi}_i - \cos(\phi_i)\cos(\kappa_i l_i)\dot{\kappa}_i l_i,$$

$$a_{23} = -\cos(\phi_i)\sin(\kappa_i l_i)\dot{\phi}_i - \sin(\phi_i)\cos(\kappa_i l_i)\dot{\kappa}_i l_i,$$

$$a_{33} = -\sin(\kappa_i l_i)\dot{\kappa}_i l_i,$$

$$a_{14} = \frac{\dot{\kappa}_i}{\kappa_i^2}\{\cos(\phi_i) - \cos(\phi_i)\cos(\kappa_i l_i)\} + \frac{\dot{\phi}_i}{\kappa_i}\{\sin(\phi_i) - \sin(\phi_i)\cos(\kappa_i l_i)\} - \frac{\dot{\kappa}_i l_i}{\kappa_i}\cos(\phi_i)\sin(\kappa_i l_i),$$

$$a_{24} = \frac{\dot{\kappa}_i}{\kappa_i^2}\{\sin(\phi_i) - \sin(\phi_i)\cos(\kappa_i l_i)\} + \frac{\dot{\phi}_i}{\kappa_i}\{\cos(\phi_i)\cos(\kappa_i l_i) - \cos(\phi_i)\} - \frac{\dot{\kappa}_i l_i}{\kappa_i}\sin(\phi_i)\sin(\kappa_i l_i),$$

$$a_{34} = -\frac{\dot{\kappa}_i}{\kappa_i^2}\sin(\kappa_i l_i) + \frac{\dot{\kappa}_i l_i}{\kappa_i}\cos(\kappa_i l_i).$$

Now, using Eqs. (22), (27), (20), and (28), the matrix in Eq. (26) can be computed numerically. Once Eq. (26)

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

is computed, the Jacobian matrix can be generated by factoring the appropriate elements from the matrix of time derivatives. If

$$\dot{T}_0^5 = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & \alpha_{44} \end{bmatrix} \tag{29}$$

and $\mathbf{x}=[x,y,z]^T$, then $J\dot{\mathbf{q}}$ can be constructed from Eq. (29) as

$$J\dot{\mathbf{q}} = [\alpha_{14}, \quad \alpha_{24}, \quad \alpha_{34}]^T. \tag{30}$$

The Jacobian matrix J can then be factored out of Eq. (30).

If it is desired to have $\mathbf{x}=[x,y,z,\theta_x,\theta_y]^T$, where $\boldsymbol{\theta}=[\theta_x,\theta_y]^T$ are the orientation angles of the tangent vector $\mathbf{t}=[\mathbf{t_x},\mathbf{t_y},\mathbf{t_z}]$ with respect to the x and y axes, then it is possible to determine J as follows. If the orientation angles are defined as

$$\boldsymbol{\theta} = \begin{bmatrix} \arctan\left(\dfrac{t_y}{t_z}\right) \\ \arctan\left(\dfrac{t_x}{t_z}\right) \end{bmatrix}, \tag{31}$$

where the tangent vector $\mathbf{t}$ can be determined from the third column of the forward kinematic's matrix, and the time derivative of Eq. (31) can be written as

$$\dot{\boldsymbol{\theta}} = \begin{bmatrix} \dfrac{\dot{t}_y t_z - t_y \dot{t}_z}{t_z^2 + t_y^2} \\ \dfrac{\dot{t}_x t_z - t_x \dot{t}_z}{t_z^2 + t_x^2} \end{bmatrix}, \tag{32}$$

the vector $\dot{\mathbf{t}}$ is formed as

$$\dot{\mathbf{t}} = J_t \dot{\mathbf{q}} = \begin{bmatrix} \alpha_{13} \\ \alpha_{23} \\ \alpha_{33} \end{bmatrix}, \tag{33}$$

where $J_t=[J_{t1},J_{t2},J_{t3}]^T$ is the associated Jacobian matrix for the tangent vector. Using Eqs. (32) and (33), the Jacobian matrix for the orientation angles can be constructed as

$$J_\theta = \begin{bmatrix} J_{\theta 1} \\ J_{\theta 2} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{t_z^2 + t_y^2}(t_z J_{t2} - t_y J_{t3}) \\ \dfrac{1}{t_z^2 + t_x^2}(t_z J_{t1} - t_x J_{t3}) \end{bmatrix}. \tag{34}$$

Thus, given $\mathbf{x}=[x,y,z,\theta_x,\theta_y]^T$, the velocity kinematics are

$$J\dot{\mathbf{q}} = \begin{bmatrix} \alpha_{14} \\ \alpha_{24} \\ \alpha_{34} \\ J_{\theta 1}\dot{q} \\ J_{\theta 2}\dot{q} \end{bmatrix}. \tag{35}$$

As before, the Jacobian matrix J can be factored out of Eq. (35).

4.7. Motion Planning

Now that the forward and velocity kinematics have been established we can use them for the motion planning of the robot. If we are given the vector of desired task space velocities, then the general solution[29] to Eq. (24) is

$$\dot{\mathbf{q}} = J(\mathbf{q})^+ \dot{\mathbf{x}} + \{I - J(\mathbf{q})^+ J(\mathbf{q})\}\varepsilon. \tag{36}$$

The term $J(\mathbf{q})^+$ in Eq. (36) is the pseudo inverse of $J(\mathbf{q})$ and is given in its most commonly used form as

$$J^+ = W^{-1}J^T(JW^{-1}J^T)^{-1}, \tag{37}$$

where W is a positive definite weighting matrix. If W is equal to the identity matrix, then the term $J^+\dot{\mathbf{x}}$ is the least squares solution of Eq. (24).[29] The term $(I - J^+J)\varepsilon$ is the null space projection of the solution of Eq. (24). This solution only generates motion in the "joint" space of the manipulator, and not the task space of the robot.[29] This joint space motion is also known as the self motion of the robot. The manipulator can now be controlled using any of the standard redundancy techniques, i.e., refs. 29–31, to generate the desired robot motion.

5. EXPERIMENTS

Now that a theoretical kinematic model has been presented for continuum style robots the next logical step is to provide experimental verification of the model, and also the viability and "flexibility" of both

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

the elephant's trunk manipulator and our kinematic model. A video of a real time implementation of similar experiments can be viewed in ref. 32.

5.1. Implementation

Before various experiments are presented, some important information is presented to help understand how the kinematic model was implemented. We present how the curvature values were determined, how the curvatures obtained from the robot relate to the kinematic variables, and the control used to servo the robot.

5.1.1. Real Time

Real time implementation of the elephant trunk robotic manipulator was conducted using the QMotor 3.0 program by Quality Real-Time Systems, and is run under the QNX 4.0 real time operating system. The basis of the control program was implemented using a C++ program, and real time data acquisition for the main sections of the robot was performed by a Quanser MultiQ III I/O board, and a ServoToGo I/O board was used for the prismatic joint. To increase the "flexibility" of the system the desired velocity trajectories could be given by a pair of three degree of freedom joysticks. The servo output from the I/O boards was connected to four dual channel Techron linear amplifiers which powered the elephant's trunk dc motors. All of the calculations and I/O operations were conducted at a frequency of 1500 Hz on a AMD 1300 MHz CPU.

5.1.2. Determination of Curvature

As with most continuum style robots, due to the elephant's trunk construction, in particular the large number of joints and the inability to mount measurement devices for the joint angles, the determination of the manipulators shape is a problem. There are several different technologies that could help solve this problem, but they are very difficult and costly to implement on a three dimensional robot. The solution adapted for the elephant's trunk incorporates the geometry of the manipulator, the cable lengths, and the basic assumption that each of the joints in a section rotate by the same amount. This assumption is a direct result of the assumption that the sections should bend with constant curvature. From this information an approximate curvature can be fitted to each section of the manipulator as follows.

Figure 11 shows a simplified diagram of one side

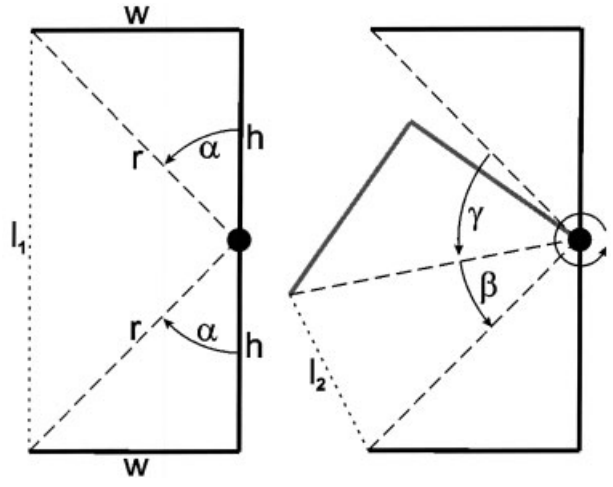

Figure 11. Diagram for the determination of curvature.

of one of the four segments in each section, where w is the radius of the segment, h is the distance from the center of the segment's joint to the end of the segment, $r=\sqrt{w^2+h^2}$, l_1 is the length of cable over the segment before it has been actuated, and l_2 is the cable length after actuation. The important value that needs to be determined is the angle γ that each joint in a section moves through given a change in cable length. We define Δl as

$$\Delta l = l_1 - l_2 . \tag{38}$$

From Figure 11 we have the relation

$$\pi = \gamma + 2\alpha + \beta, \tag{39}$$

where the angle α is

$$\alpha = \arctan\left(\frac{w}{h}\right) \tag{40}$$

and the angle β is

$$\beta = \arctan\left(\frac{\sqrt{4r^4-(2r^2-l_2^2)^2}}{2r^2-l_2^2}\right). \tag{41}$$

Combining Eqs. (39), (40), and (41) the total angle of rotation of one joint in a section after actuation is

$$\gamma = \pi - 2\arctan\left(\frac{w}{h}\right) - \arctan\left(\frac{\sqrt{4r^4-(2r^2-l_2^2)^2}}{2r^2-l_2^2}\right). \tag{42}$$

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

The cable length l_2 can be determined by solving Eq. (38), where Δl is the amount of cable retracted for a given section. Using Eq. (11) and noting that each section contains n segments we have

$$\theta = n\gamma = \frac{\kappa s}{2}, \tag{43}$$

and solving for the curvature yields

$$\kappa = \frac{2n\gamma}{s}. \tag{44}$$

For the elephant's trunk manipulator we have $n=4$ and $s=8$ in., and thus for section $i=\{1,2,3,4\}$ we have

$$\kappa_i = \tfrac{1}{2}\gamma_i, \tag{45}$$

where the units of the curvature are 1/in. The calculated curvature is an approximation to the exact curvature for each section due to the fact that s does not remain exactly a constant value, but through experimentation proves to be very close to constant.

We would like to make the note that the design of the cable drive system has the cable pairs of each plane running parallel to each other off the center axis of the manipulator by some distance. This has the effect of coupling the cable actuation of a section with all of the previous sections. When one of the previous sections moves, then, since the cables are not on the center line of the robot, the following sections' curvatures will also change. Therefore by Eqs. (42) and (45) all the cables that run through that section must be altered by the amount of cable length in the actuated section to maintain the proper curvature of the sections.

5.1.3. Curvature to Rotation Angle Relationship

Up to this point the kinematics of a continuum style robot's sections have been described by a curvature and a rotation angle. However, the elephant's trunk manipulator physically does not move in this fashion. The information that is retrieved from each of the robot's sections is two orthogonal curvature measurements via Eqs. (42) and (45), instead of one curvature and one rotation angle. However, there is a convenient transformation between these two sets of variables.

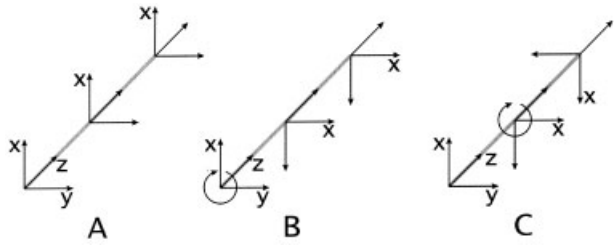

Figure 12. Effects of rotation on kinematic frames.

To transform from the two orthogonal curvatures to one curvature and one rotation angle only requires the following transformation:

$$\kappa = \sqrt{\kappa_x^2 + \kappa_y^2}, \qquad \phi = \arctan\left(\frac{\kappa_y}{\kappa_x}\right). \tag{46}$$

If $-\pi/2 \leqslant \phi \leqslant \pi/2$, then κ is positive, else κ is negative and ϕ is altered by π radians so that $-\pi/2 \leqslant \phi \leqslant \pi/2$.

The inverse transformation from the kinematic variables to the robot variables involves a slightly different approach. The problem here lies in that the rotation angle of one section is a function of the previous section's rotation angle. If one can imagine looking down the "backbone" of a continuum robot that is in a straight configuration [see Figure 12(A)], one can see that as each section in the kinematic model rotates all the sections after that section rotate around the backbone by the same amount [see Figures 12(B) and (C)]. The calculation of the two orthogonal curvatures in a section with respect to that section's corresponding rotation angle is

$$\kappa_x = \kappa\cos(\phi), \qquad \kappa_y = \kappa\sin(\phi). \tag{47}$$

These values now must be transformed to their proper values by accounting for the rotation angles of all the previous sections. The transformation is simply

$$\begin{bmatrix} \kappa^*_{i,x} \\ \kappa^*_{i,y} \end{bmatrix} = \begin{bmatrix} \cos(\phi_{i-1}+\phi_{i-2}+\cdots) & -\sin(\phi_{i-1}+\phi_{i-2}+\cdots) \\ \sin(\phi_{i-1}+\phi_{i-2}+\cdots) & \cos(\phi_{i-1}+\phi_{i-2}+\cdots) \end{bmatrix} \begin{bmatrix} \kappa_{i,x} \\ \kappa_{i,y} \end{bmatrix}, \tag{48}$$

where $i=\{1,2,3,4\}$.

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

5.1.4. Shape Control

The real time low level controller for the elephant's trunk is composed of two main steps. During the first step the controller converts the kinematic variables to the robot's curvature variables via Eq. (48). Then for the second step a PD controller servos between the desired curvature κ_d and the present curvature κ found through Eq. (45) for each axis in a section. The control algorithm for one axis in a section is as follows:

$$\tau = G_p(\kappa_d - \kappa) - G_d\dot{\kappa}, \tag{49}$$

where τ is the voltage to the servo amplifiers, $G_p = 2000$, $G_d = 1.0$, and $\dot{\kappa}$ is the velocity of the curvature. Since $\dot{\kappa}$ is not directly measurable, it was obtained through a two step approach. First κ' was calculated using the following equation,

$$\kappa' = \frac{\Delta\kappa}{\Delta t}, \tag{50}$$

where $\Delta\kappa$ is the change in the curvature over the sample time of the control. For a control frequency of 1500 Hz, $\Delta t = 667$ μs. A simple digital filter was then used to smooth κ'. The equation for the digital filter was

$$\dot{\kappa} = \left\{\frac{1}{2+\omega_c}\right\}\{\omega_c\kappa' + \omega_c\kappa_0' - (\omega_c - 2)\dot{\kappa}_0\}, \tag{51}$$

where $\omega_c = 9.42 \times 10^3$ for a control frequency of 1500 Hz, κ_0' is the value of κ' during the previous control cycle, and $\dot{\kappa}_0$ is the value of $\dot{\kappa}$ during the previous control cycle.

The input to the controller for the robot is obtained in two ways. The first approach is through the use of a trajectory simulator that uses an iterative scheme based on Eq. (36) to compute the trajectory. The complete trajectory of kinematic variables is then fed into the robot's low level controller. The second approach follows the real time acquisition of desired endpoint velocities, $\dot{\mathbf{x}}_\mathbf{d}$, that are provided by the joysticks, i.e., tele-operation. This approach uses the following form of Eq. (36),

$$\dot{\mathbf{q}}_\mathbf{d} = J(\mathbf{q}_\mathbf{d})^+\dot{\mathbf{x}}_\mathbf{d}, \tag{52}$$

where the kinematic variables are calculated as

$$\mathbf{q}_d = \int \dot{\mathbf{q}}_d \, dt. \tag{53}$$

These kinematic variables are then used in the low level joint controller in the same way as in the first approach.

5.2. Configuration Tracking

Given a predefined set of kinematic variables the configuration tracking approach compares the differences in the configuration of a simulated robot based on the theoretical kinematics to that of a physical continuum style robot. The idea is that any inaccuracies in the kinematic model will show up as deviations in the configuration of the physical robot from that of the configuration for the simulated robot. If the kinematics are valid, then there should be no discernible difference between the configurations.

In both of the following experiments the initial configuration was $\mathbf{q} = [0,0,0,0,0,0,0,0,0]^T$, yielding an initial end-point location of $\mathbf{x} = [0,0,32]$ in., and the desired end-point location was $\mathbf{x_d} = [10,0,10]$ in. In each of the experiments a figure shows a comparison between the simulated configurations at several points in the linear trajectory and the corresponding photographic snapshots of the physical robot at the same points. Snapshots of the manipulator are shown due to the lack of ability to precisely measure the robot's curvatures and end-point location, therefore the errors $\mathbf{x_d} - \mathbf{x}$ and $\mathbf{q_d} - \mathbf{q}$ cannot be verified precisely. It is important to note that manipulators of this design are not well suited to high precision applications due to their actuation and physical construction, but are instead best suited for applications where their flexibility and compliance can be exploited.

We would like to point out that a great deal of the robot's inaccuracy is inherited from the method used for the calculation of curvature and the coupling between the cables of the sections. It is clear that a more accurate determination of a section's curvature, i.e., direct measurement, and a better cable actuation design would yield better results. Even with these limitations in measurement and actuation, the robot's configurations are very close to that of the simulation. We would also like to point out that even though our kinematic model has no problem dealing with a 3D workspace, the following experiments were conducted in 2D for ease of presentation.

5.2.1. Prismatic Joint Excluded (Figure 13)

In this experiment the weighting matrix in Eq. (37) is set such that $W^{-1} = [0,1,1,1,1,1,1,1,1]$. This has the ef-

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

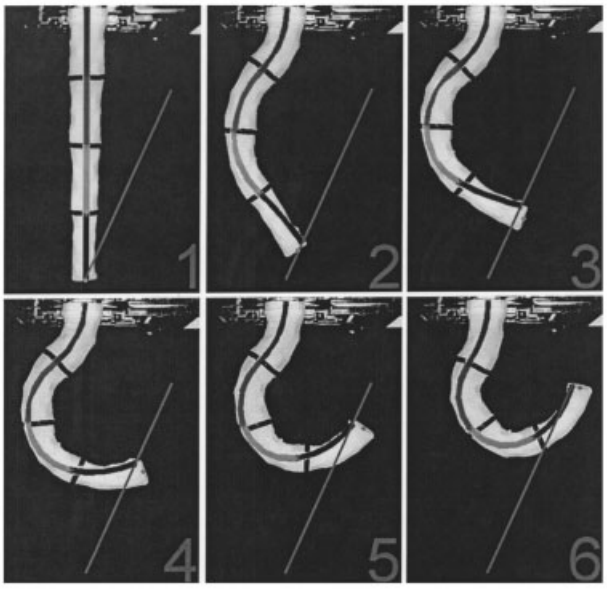

Figure 13. Configuration tracking without prismatic joint.

fect of nullifying any motions of the prismatic joint, thus allowing only the four sections of the manipulator to move in order to obtain the proper motion of the end point of the manipulator. The results of the experiment are shown in Figure 13. The manipulator does not exactly follow the desired trajectory, but the amount of error is very small given the physical limitations of our manipulator.

5.2.2. Prismatic Joint Included (Figure 14)

In this experiment the weight matrix is set to W^{-1} =[1000,1,1,1,1,1,1,1]. The reason that the weighting factor for the prismatic joint is set to a much greater value than the joint variables is to help equalize its contribution to the kinematics. The result of the experiment is shown in Figure 14. Even with the addition of the prismatic joint, our kinematic model remains a very close approximation of the manipulator's kinematics.

5.3. End-point Tracking

The end-point velocity tracking approach compares the actual end-point trajectory of the robot to that of the desired trajectory. The desired trajectory is obtained by specifying velocities for the end-point. Thus, the desired kinematic variables can be calculated with Eqs. (52) and (53), where $\dot{\mathbf{x}}_\mathbf{d}$ was the desired

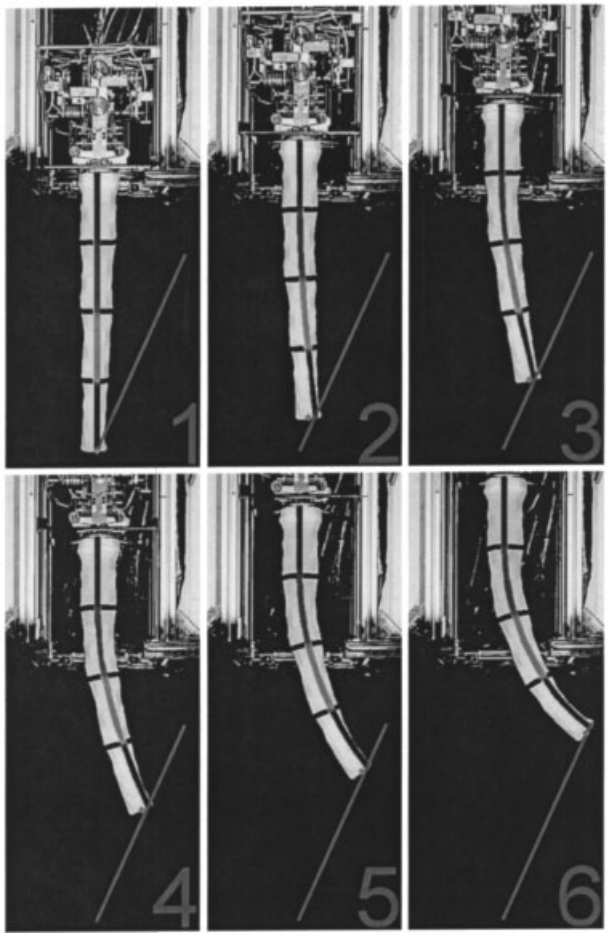

Figure 14. Configuration tracking with the prismatic joint.

velocity provided via one of the elephant's trunk setup's joystick. In this situation there is no feedback control on the end-point's position. The idea is that any error in the kinematic model will show up as deviations in the robot's end-point trajectory from that of the desired end-point trajectory.

5.3.1. Prismatic Joint Excluded (Figure 15)

Shown in Figure 15 are six sequential photographs that show the robot's movement when following the desired end-point velocity. In this experiment the prismatic joint was disabled by again using W^{-1} =[0,1,1,1,1,1,1,1]. From the figure it can be seen that our kinematic model still provides a very good approximation to the robot's actual kinematics.

5.3.2. Prismatic Joint Included (Figure 16)

In this experiment the weight matrix is set to W^{-1} =[1000,1,1,1,1,1,1,1]. The results of the experiment

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

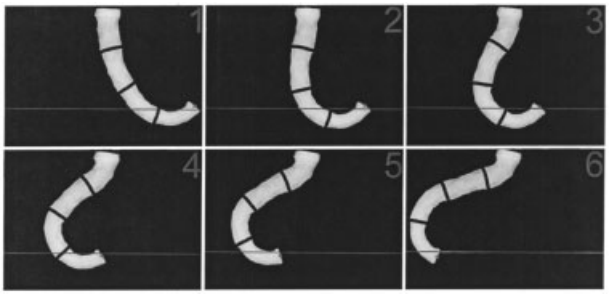

Figure 15. End-point tracking without prismatic joint.

are shown in Figure 16, and again it can be seen that our kinematic model very closely approximates the physical setup of the manipulator.

5.4. Obstacle Avoidance

As was demonstrated in Figure 3, continuum style manipulators are well suited for obstacle avoidance. Avoiding obstacles inherently requires a manipulator to exhibit a large number of degrees of freedom, and though continuum style robots may not have a large number of actuated degrees of freedom, in general their physical structure does exhibit a large number of degrees of freedom. This helps them to achieve complex motions over large ranges with minimum actuation, and thus makes them very efficient for obstacle avoidance. Conventional style manipulators, on the other hand, would require many more actuators to achieve the same motions. Many different obstacle avoidance strategies have been developed for conventional manipulators, but none of them have been applied to continuum style robots. In the following section two different strategies are presented to help demonstrate the viability of continuum robots in obstacle avoidance.

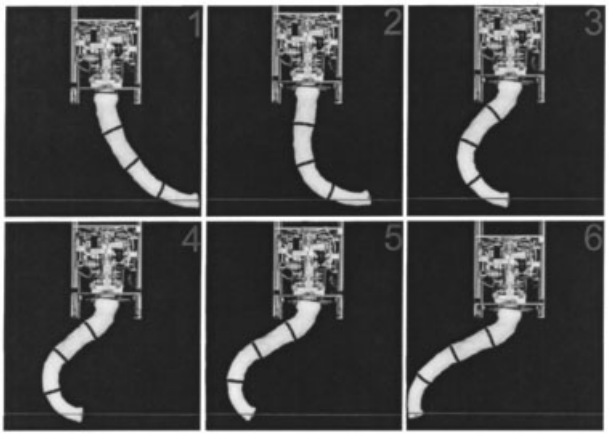

Figure 16. End-point tracking with the prismatic joint.

5.4.1. Reference Configuration

As presented in ref. 31, a simple approach to obstacle avoidance is through the concept of subtasks. In many cases obstacle avoidance is a secondary objective. The manipulator's main task is that of following a given end-point trajectory, and whatever ability for motion is left over after satisfying the end-point task can be used to avoid the obstacle. The idea is to define a static reference configuration for the robot that sufficiently avoids the obstacle, and then use the null space projection, which does not affect the end-point trajectory, to cause the manipulator to adopt postures which resemble this configuration. Thus, the robot will try and maintain a configuration that will exactly satisfy the end-point constraint as well as trying to maintain the reference configuration that avoids the obstacle. This approach was implemented by defining $\boldsymbol{\varepsilon}$ in Eq. (36) to be

$$\boldsymbol{\varepsilon} = k(\mathbf{q_r} - \mathbf{q}), \tag{54}$$

where $k = 0.001$ is positive control gain and $\mathbf{q_r} = [0, 0, \pi/2, 0.35, 0, -0.25, 0, -0.15]^T$ is the desired reference configuration that avoids the obstacle. Note, this experiment was performed before the prismatic joint was added, and therefore $\mathbf{q_r}$ has only eight variables. The experimental results are shown in Figure 17. The first picture shows the spatial relationship between the robot and the obstacle. The second picture shows the configuration of the robot before Eq. (54) is implemented. The position was obtained after moving from the initial position, first picture, to that position with $\boldsymbol{\varepsilon} = \mathbf{0}$. This configuration of the robot is not conducive for avoiding the obstacle. If only the least squares solution was implemented from this configuration, then the robot would collide with the obstacle. In the third picture, as the robot begins to approach the obstacle, Eqs. (36) and (54) are used to calculate $\boldsymbol{\varepsilon}$. In this and the following pictures one can see the configurations of the robot evolving in such a way that they try to obtain the reference position, and thus sufficiently avoid the obstacle.

5.4.2. Reference Curve

The previous experiment is useful in showing how the robot can avoid an obstacle given that we already know the exact manipulator configuration which

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

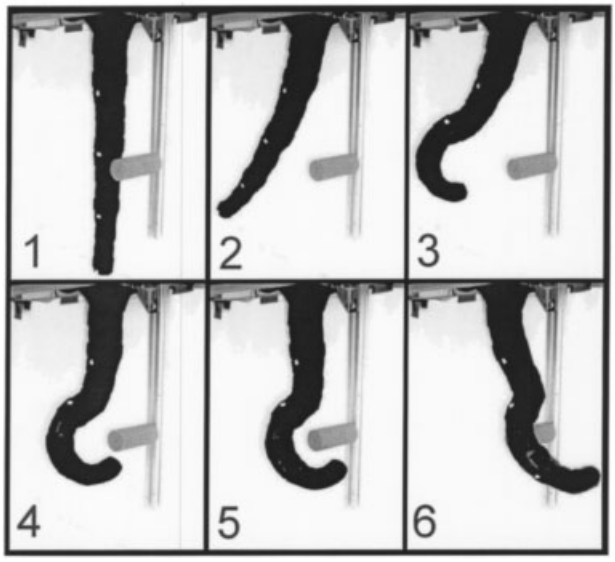

Figure 17. Obstacle avoidance using a reference configuration.

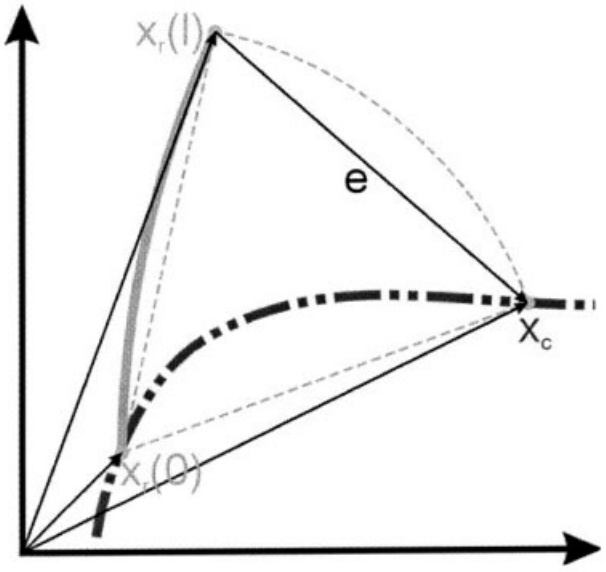

Figure 18. The desired shape verses the manipulator's shape.

avoids the obstacle. This approach can be used in many different situations, but it does have its limitations. In this section a new alternative strategy for planar obstacle avoidance is presented. Based on research presented in refs. 14–16, another possible way to design an obstacle avoidance strategy would be to define a curve that would sufficiently clear the obstacle. The robot would then be "servoed" in such a way that it would take on the basic shape of the curve. The idea is that the defined curve is simple, and, with the exception of a few constraints, requires little knowledge of the robots kinematics. With the aid of some basic information about the environment, the operator defines a curve that avoids the obstacle, and once the curve is generated, the kinematics of the robot are used to servo the robot to the curve.

As shown in Figure 18, to implement such a strategy the predetermined obstacle avoidance curve is defined as

$$\mathbf{x_c} = \mathbf{f}(\mathbf{s_c}), \tag{55}$$

and from the forward kinematics the position of one section of the robot can be generally defined as

$$\mathbf{x_r} = \mathbf{g}(\mathbf{s_r}), \tag{56}$$

where s is the arc length in each case. The basis of this approach is that the ends of each of the robot's sections do not initially line up with the curve. The approach is to servo the robot in such a way that the end of each section will lie on the curve. The beginning of each section is the end of the previous section so there is no need to servo this point for each section. Also, the constraint that the first section's origin must be on the curve is imposed. This condition is easily satisfied by impressing the constraint that the curve must be designed such that it and the robot originate from the same location. The problem in servoing the robot to the curve is that given $\mathbf{x_c}$, Eq. (11), and Eq. (12) cannot be solved directly for κ. Therefore, some type of iterative scheme can be used.

If it is given that the origin of the manipulator's section, $\mathbf{x_r}(\mathbf{0})$, is on the curve, then some way is needed to determine the proper κ that places the end of the section on $\mathbf{x_c}$. The error between the curve and the manipulator is defined as

$$\mathbf{e} = \mathbf{x_c} - \mathbf{x_r}. \tag{57}$$

The problem is: given $\mathbf{x_r}(\mathbf{l})$, ($\mathbf{l}$ is total length of the section), how can $\mathbf{x_c}$ be determined? This is done by defining $\mathbf{x_c}$ in terms of its magnitude and angle such that it is solvable for s_c, then equation $|\mathbf{x_c}| = |\mathbf{x_r}|$ is solved for s_c. Once s_c has been obtained, $\mathbf{x_c}$ and thus $\mathbf{e}$ can be calculated. Now that the error vector has been determined, it can be used to determine κ by iterating the following equation,

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

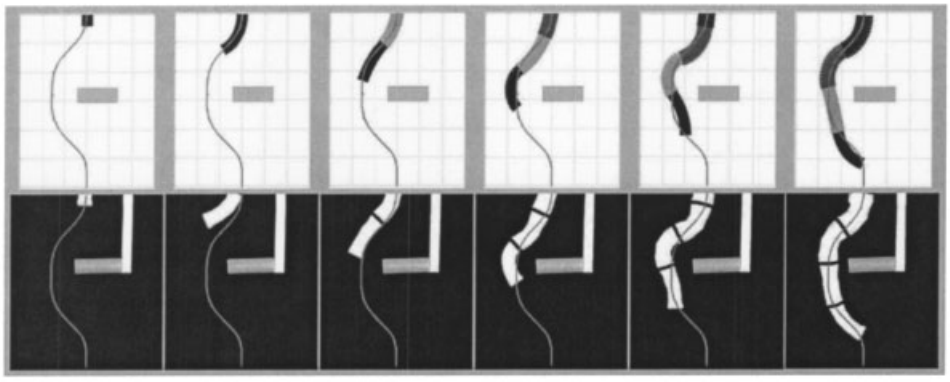

Figure 19. Obstacle avoidance using a static curve (top row is simulation, bottom column is experiment).

$$\frac{d\boldsymbol{\kappa}}{dt} = -k_p \mathbf{e}, \tag{58}$$

where k_p is a positive constant. What is important to note about this strategy is that the desired result is that the end of each section lies on the curve. This does not guarantee that the manipulator will always have the exact shape of the desired curve, but it will have the same general shape.

To show the viability of this strategy, a simple experiment that incorporates not only the different sections of the robot, but also the prismatic joint is presented. As shown in the right hand column of Figure 19, a curve was generated by using several key points that avoid the obstacle. A cubic spline was then fitted through those points to determine $\mathbf{x_c}$. The robot is then moved along the curve with the prismatic joint. It can be seen from the different configurations in Figure 19 that even though the end of each section is on the curve, the robot does not have the exact shape of the curve. However, the robot does maintain the same basic shape, and thus this strategy still provides a very useful and easily implementable strategy for complicated obstacle avoidance.

5.5. Grasping

Not only are continuum style robots well suited for obstacle avoidance, but their design is also very conducive for grasping. Unlike conventional manipulators which require some type of end-effector or grasping mechanism, continuum style manipulators can perform what is commonly known as whole arm grasping. In whole arm grasping the manipulator uses itself to grasp an obstacle without the aid of any additional mechanisms. The manipulator uses its "whole arm" to grasp the object, very much like an elephant can do with its trunk. This allows the manipulator to grasp a large range of objects independent of the objects size or shape.

To demonstrate the ability of continuum robots to perform whole arm grasping we presented a simple strategy in ref. 21. The basic approach is to use one joystick to provide the information for Eq. (36), with $\boldsymbol{\varepsilon} = \mathbf{0}$ and $W^{-1} = [0,1,1,1,1,1,1,1,1]$, to maneuver the manipulator around. As a second control input, another joystick was used to control the last sections shape by providing the desired "joint" velocities. The last section was manipulated in such a way that it could grasp the desired object. The weighting matrix was then set to $W^{-1} = [0,1,1,1,1,1,1,1,0]$ so the last section would not change its grasping position. Setting the weighting matrix in this fashion has the effect of "locking" the last section of the robot in such a way that it will not change its shape, but the inverse kinematics will still function properly. Figure 20 shows one of the images from ref. 21 that demonstrates the physical robot grasping a small cylinder. Figure 21 shows a different image of the physical robot grasping a large diameter ball.

6. CONCLUSIONS

In this paper the elephant's trunk robotic manipulator is presented. The elephant's trunk is a new and novel design for a continuum style robot. The robot's basic structure is very similar to that of a backbone. It is composed of a serial connection of 16 two degree of freedom joints, and is actuated by a cable servo system. To facilitate intelligent motion for continuum style manipulators, such as the elephant's trunk, a new kinematic model was developed. Unlike

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

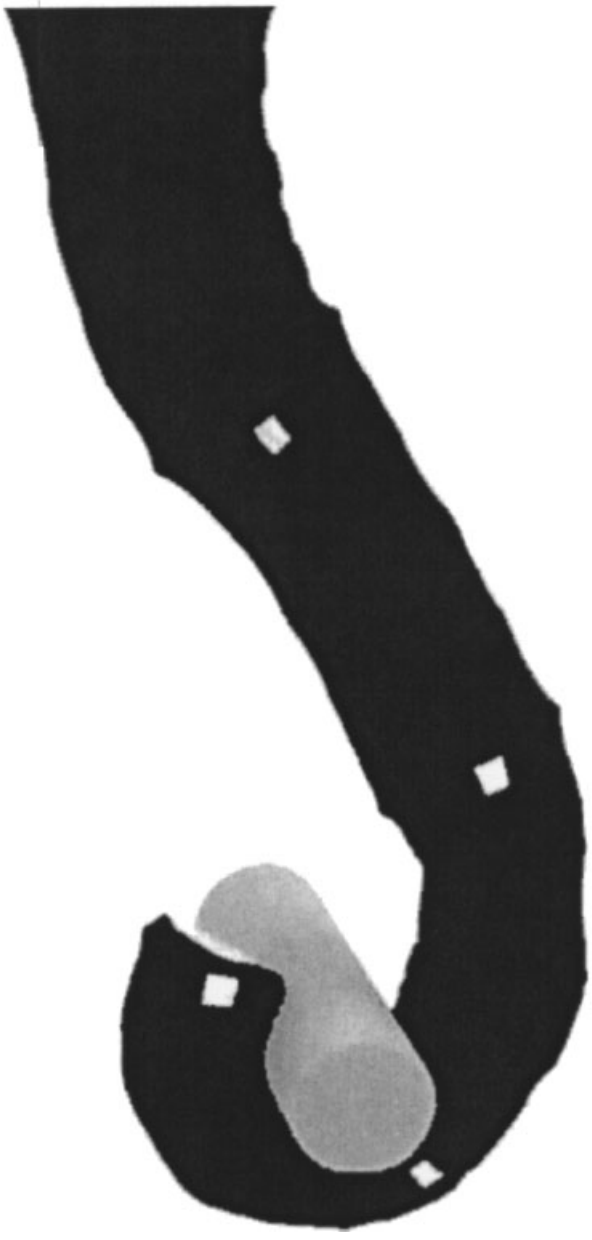

Figure 20. Whole arm grasping.

Figure 21. Grasping of a ball.

conventional manipulators which use link lengths and joints as a basis for their kinematic model, the model presented uses the concept of arc length and curvature to describe the kinematics. The kinematic model uses the idea that the bending motions demonstrated by continuum style robots can be best described by using the concept of constant curvature. These motions exhibited by curves that can only bend with sections of constant curvature is examined, and the resulting analysis demonstrates that the complex motion exhibited by these curves can be dissected into a series of simple, but coupled motions. This idea is then incorporated into the widely used Denavit-Hartenberg procedure for kinematic analysis. This ability to rewrite the kinematics of continuum robots in a form that can be incorporated into the Denavit-Hartenberg procedure exposes continuum robots to much of the kinematic analysis and motion planning research that is used for conventional robots. The model has the ability to account for any number and arrangement of robot sections. Also the model can easily account for not only continuum style robot actuators, but also the incorporation any type of conventional joints, i.e., rotational and prismatic joints.

Experimental results are presented to validate both the accuracy and viability of the designed kinematic model using the elephant's trunk continuum style robot. With the kinematic model verified, several different strategies were implemented on the elephant's trunk to demonstrate the usefulness of the synthesized kinematic model. The first set of strategies dealt with obstacle avoidance. First a simple avoidance strategy is used to demonstrate the viability of continuum robots for obstacle avoidance. The adopted strategy uses the idea of subtasks and a predefined reference configuration for the robot to avoid an obstacle. This strategy uses the idea of subtasks and a predefined reference configuration for the robot to avoid an obstacle. A second new obstacle avoidance strategy is also provided. This strategy uses the idea of a predefined curve that avoids the obstacle. A simple obstacle avoidance experiment based on

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

inserting the elephant's trunk robot into a cluttered environment is performed, and thus demonstrates some of the more powerful uses for continuum style manipulators. Also, work is presented that helps to demonstrates the usefulness of continuum style robots for whole arm grasping.

The research presented in this paper provides a solid foundation for both the mechanical design and the kinematics of continuum style robotic manipulators. Both the elephant's trunk and the kinematic model prove to work very well.

REFERENCES

1. G. Robinson and J. B. C. Davies, Continuum Robots-A State of the Art, IEEE Conf. on Robotics and Automation, 1999, pp. 2849–2854.
2. V. C. Anderson and R. C. Horn, "Tensor Arm Manipulator Design" ASME paper 67-DE-57.
3. R. Cieslak and A. Morecki, Elephant Trunk Type Elastic Manipulator-A Tool for Bulk and Liquid Materials Transportation, Robotica 17 (1999), 11–16.
4. S. Hirose, Biologically inspired robots, Oxford U.P., Oxford, 1993.
5. G. Immega and K. Antonelli, The KSI Tentacle Manipulator, IEEE Conf. on Robotics and Automation, 1995, pp. 3149–3154.
6. H. Ohno and S. Hirose, Study on Slime Robot (Proposal of Slime Robot and Design of Slim Slime Robot), IEEE Conf. on Intelligent Robots and Systems, 2000, pp. 2218–2223.
7. K. Suzumori, S. Iikura, and H. Tanaka, Development of Flexible Microactuator and its Applications to Robotic Mechanisms, IEEE Conf. on Robotics and Automation, 1991, pp. 1622–1627.
8. J. F. Wilson, D. Li, Z. Chen, and R. T. George, Flexible Robot Manipulators and Grippers: Relatives of Elephant Trunks and Squid Tentacles, *Robots and Biological Systems: Towards a New Bionics?*, 1993, pp. 474–494.
9. E. Paljug, T. Ohm, and S. Hayati, The JPL Serpentine Robot: a 12 DOF System for Inspection, IEEE Conf. on Robotics and Automation, 1995, pp. 3143–3148.
10. G. S. Chirikjian, Theory and Applications of Hyper-Redundant Robotic Mechanisms, Ph.D. Thesis, Dept. of Applied Mechanics, California Institute of Technology, 1992.
11. G. S. Chirikjian, A General Numerical Method for Hyper-Redundant Manipulator Inverse Kinematics, IEEE Conf. on Robotics and Automation, 1993, pp. 107–112.
12. G. S. Chirikjian and J. W. Burdick, A Model Approach to Hyper-Redundant Manipulator Kinematics, IEEE Conf. on Robotics and Automation, 1994, pp. 343–354.
13. G. S. Chirikjian and J. W. Burdick, Kinematically Optimal Hyper-Redundant Manipulator Configurations, IEEE Conf. on Robotics and Automation, 1995, pp. 794–780.
14. H. Mochiyama, E. Shimemura, and H. Kobayashi. Direct Kinematics of Manipulators with Hyper Degrees of Freedom and Serret-Frenet Formulas, IEEE Conf. on Robotics and Automation, 1998, pp. 1653–1658.
15. H. Mochiyama, E. Shimemura, and H. Kobayashi. Shape Correspondance Between a Spatial Curve and a Manipulator with Hyper Degrees of Freedom, IEEE Conf. on Intelligent Robots and Systems, 1998, pp. 161–166.
16. H. Mochiyama and H. Kobayashi, The Shape Jacobian of a Manipulator with Hyper Degrees of Freedom, IEEE Conf. on Robotics and Automation, 1999, pp. 2837–2842.
17. I. Gravagne and I. D. Walker, On the Kinematics of Remotely-Actuated Continuum Robots, IEEE Conf. on Robotics and Automation, 2000, pp. 2544–2550.
18. I. Gravagne and I. D. Walker, Kinematic Transformations for Remotely-Actuated Planar Continuum, IEEE Conf. on Robotics and Automation, 2000, pp. 19–26.
19. I. A. Gravagne and I. D. Walker, Kinematics for Constrained Continuum Robots Using Wavelet Decomposition, in Robotics 2000, Proceedings of the 4th Int. Conf. and Expo./Demo. on Robotics for Challenging Situations and Environments, 2000, pp. 292–298.
20. M. W. Hannan and I. D. Walker, Analysis and Initial Experiments for a Novel Elephant's Trunk Robot, IEEE Conf. on Intelligent Robots and Systems, 2000, pp. 330–337.
21. M. W. Hannan and I. D. Walker, Analysis and Experiments with an Elephant's Trunk Robot, to appear in the International Journal of the Robotics Society of Japan, 2001.
22. I. D. Walker and M. W. Hannan, A Novel Elephant's Trunk Robot, IEEE/ASME Conf. on Advanced Intelligent Mechatronics, 1999, pp. 410–415.
23. D. J. Struik, Lectures on classical differential geometry, Addison-Wesley, Reading, MA, 1961.
24. C. Li and C. D. Rahn, Nonlinear Kinematics for a Continuous Backbone, Cable-Driven Robot, 20th Southeastern Conf. on Theoretical and Applied Mechanics, 2000.
25. M. W. Hannan and I. D. Walker, Novel Kinematics for Continuum Robots, 7th International Symposium on Advances in Robot Kinematics, 2000, pp. 227–238.
26. M. W. Spong and M. Vidyasagar, Robot dynamics and control, Wiley, New York, 1989.
27. M. W. Hannan and I. D. Walker, The 'Elephant Trunk' Manipulator, Design and Implementation, IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics 2001, pp. 14–19.
28. S. Hirose and S. Ma, Moray Drive for Multijoint Manipulators, Fifth International Conference on Advanced Robotics, 1991, pp. 521–526.
29. D. N. Nenchev, Redundancy Resolution through Local Optimization: A Review, J Robot Syst 6 (1989), 769–798.
30. B. Siciliano, Kinematic Control of Redundant Robot Manipulators: A Tutorial, J Itel Robot Syst 3 (1990), 201–212.
31. T. Yoshikawa, Analysis and Control of Robot Manipulators with Redundancy, Robotics research—The first international symposium, MIT, Cambridge, 1984, pp. 735–747.
32. M. W. Hannan and I. D. Walker, 'Elephant Trunk' Robot Arm, Video Proceedings of the IEEE International Conference on Robotics and Automation, May 2001.

10974563, 2003, 2, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 by Huazhong University Of Sci & Tech, Wiley Online Library on [26/06/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License