2008 IEEE/RSJ International Conference on Intelligent Robots and Systems
Acropolis Convention Center
Nice, France, Sept, 22-26, 2008

A Geometrical Approach to Inverse Kinematics for Continuum Manipulators

Srinivas Neppalli, Matthew A. Csencsits, *Student member*, *IEEE*, Bryan A. Jones, *Member, IEEE*, and Ian Walker, *Fellow, IEEE*

***Abstract*—We present a new geometrical approach to solving inverse kinematics for continuous backbone (continuum) robot manipulators. First, this paper presents a solution to the inverse kinematics problem for a single-section trunk. Assuming end-points for all sections of a multi-section trunk are known, this paper then details applying single-section inverse kinematics to each section of the multi-section trunk by compensating for resulting changes in orientation. Finally, an approach which computes per-section endpoints given only a final-section endpoint provides a complete solution to the multi-section inverse kinematics problem. The results of implementing these algorithms in simulation and on a physical continuum robot are presented and possible applications are discussed.**

I. Introduction

Kinematic redundancy, where more degrees of freedom exist in the system than are strictly required for task execution, offers the benefit of improved performance in the form of singularity avoidance, obstacle avoidance as illustrated in Fig. 1, fault tolerance, joint torque optimization, and impact minimization via effective use of the self-motion inherent in the resulting systems. Kinematic redundancy in manipulators has been extensively studied, and surveys of many of the fundamental results for conventional (rigid-link) redundant manipulators are presented in [1, 2].

However, for the recently emerging class of continuum manipulators [3], progress in developing practical kinematics has been slower. Continuum robots, resembling biological trunks and tentacles, feature continuous backbones, for which conventional kinematics algorithms do not apply. While numerous hardware realizations of continuum manipulators have appeared [3], only recently have accurate and practical kinematic models for continuum manipulators emerged [4, 5].

Many existing continuum robot designs are kinematically redundant. Indeed, the inclusion of many extra degrees of freedom (hyper-redundancy) has been a key motivation for continuum robots, enabling them to maneuver in congested environments [6] and allowing them to form whole arm grasps [7] of a wide range of objects. While there have been attempts to adapt the conventional (rigid link) approaches to redundancy resolution by appropriately selecting the shape of the robot subject to task constraints [8], their practical effectiveness have been hampered by the complexity of the analysis, particularly in the resulting Jacobians.

This paper presents a geometric approach to determining the inverse kinematics for single and multi-section continuum robots. The algorithm given in section II determines a closed-form solution to the inverse kinematics problem for a single continuum section trunk.

Section III discusses extending the results from section II to an *n*-section continuum manipulator, assuming knowledge

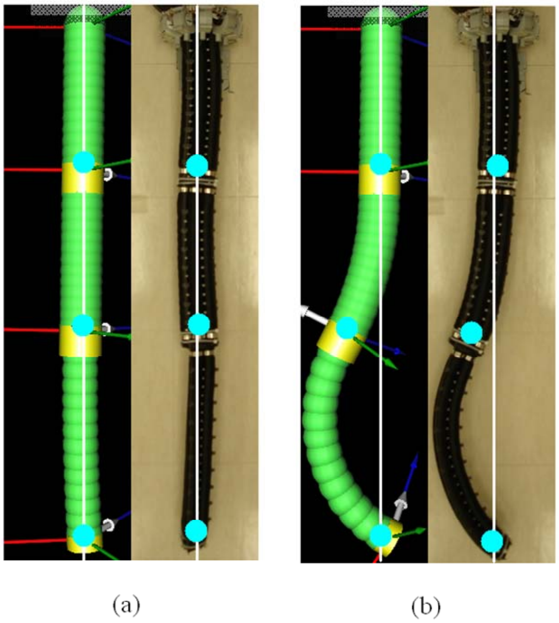

Fig. 1 – The inverse kinematics algorithms described in section II move both a simulated and an actual trunk from a vertical starting posture in (a) to a bent posture in (b) while maintaining tip position, moving only the section 2 end-point. This maneuver could be used to avoid obstacles in the trunk path while maintaining a desired tip position.

Manuscript received February 22, 2008. This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) Defense Sciences Office through the Space and Naval Warfare Systems Center, San Diego, Contract Number N66001-03-C-8043.

Srinivas Neppalli is with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762 USA (phone : 662-312-3424; e-mail: sn140@ece.msstate.edu).

Matt A. Csencsits is with the Electrical & Computer Engineering Department, Clemson University, Clemson, SC 29634 USA (phone: 864-656-7956; e-mail: csencsm@clemson.edu).

Bryan A. Jones is with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762 USA (phone: 662-325-3149; fax: 662-325-9438; e-mail: bjones @ ece.msstate.edu).

Ian D. Walker is with the Electrical & Computer Engineering Department, Clemson University, Clemson, SC 29634 USA (e-mail: ianw@ces.clemson.edu).

978-1-4244-2058-2/08/$25.00 ©2008 IEEE.

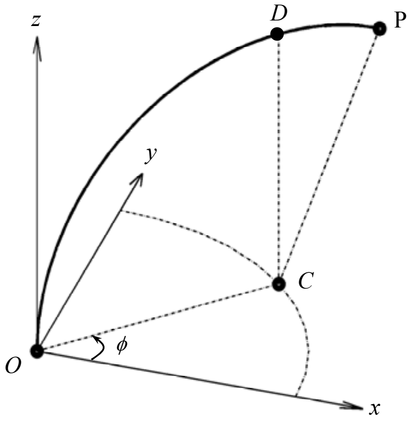

Fig. 2 – A single section of a continuum trunk modeled as an arc of a circle in 3D space with its center in *xy* plane. One of the end-points *O* is at the origin and other end-point *P* is located anywhere in 3D space.

of the end-point locations for each section of the trunk. Section IV presents a procedure to compute these per-section end-points given a single end-point for the entire trunk. Next, section V presents results obtained by implementing these inverse kinematics, in simulation and on a physical device (OctArm VI), as shown in Fig. 1. Section VI concludes with a discussion of the advantages and disadvantages of this approach and potential applications.

II. Single-Section Kinematics

For our analysis we model a single section of a continuum manipulator as an arc of a circle with one end-point *O* fixed to the origin of a right-handed Euclidean space, the other end-point *P* located anywhere in the space, and the center of the arc *C* in the *xy* plane (see Fig. 2). We parameterize a section of a continuum manipulator by its arc length s, its curvature κ, and its orientation ϕ as shown in Fig. 3. From these parameters the tip location of a single continuum section is calculated [9]. These assumptions reflect the physical structure of many continuum manipulators when subjected to a constant moment applied to the end of the section as derived in [10] and applied in [4, 5, 8, 11] including Air-Octor [12] and the OctArm [13] series of manipulators. In particular, the ability of these trunks to not only move to a given curvature κ and direction of curvature ϕ but also to extend to a trunk length s enables them to attain the desired tip position based on the ϕ, κ, and s determined by the inverse kinematics.

A. Inverse Kinematics

The trunk parameters s, κ, and ϕ for a single continuum section can be determined given the end-point location *P* in a closed-form expression. The direction of bending ϕ can be trivially determined by dividing the x and y coordinates, giving

$$\phi = \tan^{-1}\left(\frac{y}{x}\right). \tag{1}$$

The curvature can be determined by finding the distance from the origin to the center of the arc formed by the continuum section. Rotating *P* about the z axis by $-\phi$ produces a point P' such that $x' = \sqrt{x^2+y^2}$, $y' = 0$, and $z' = z$ (see Fig. 4), yielding an arc of the same curvature which lies entirely in the *xz* plane. Our model assumes the center of the arc to be in the *xy* plane; after rotation, this center must lie along the x axis. Therefore, the radius r of the center of this arc *C* lies at $(r, 0)$ in the *xz* plane. Noting that the end-point and the origin of the arc must be equidistant from *C* and recalling that the origin of the arc coincides with the origin of the coordinate system gives $(x'-r)^2 + z'^2 = r^2$. Solving for r and noting that $\kappa = r^{-1}$, $\kappa = 2x' / (z'^2 + x'^2) z'$. Substituting for x' and z',

$$\kappa = \frac{2\sqrt{x^2+y^2}}{x^2+y^2+z^2}. \tag{2}$$

The angle θ as shown in Fig. 3 can be calculated from the curvature and the Cartesian coordinates of *P*. Looking at the planar case of P', examining the $\triangle C'P'D'$ in Fig. 4 gives $\theta = \cos^{-1}\left(\left(\kappa^{-1} - x'\right) / \kappa^{-1}\right)$ when $z' > 0$ and $\theta = 2\pi - \cos^{-1}\left(\left(\kappa^{-1} - x'\right) / \kappa^{-1}\right)$ when $z' \le 0$. Noting that the rotation of *P* does not affect the arc-length, $x' = \sqrt{x^2+y^2}$ as before. Simplifying gives

$$\theta = \begin{cases} \cos^{-1}\left(1 - \kappa\sqrt{x^2+y^2}\right), & z > 0 \\ 2\pi - \cos^{-1}\left(1 - \kappa\sqrt{x^2+y^2}\right), & z \le 0. \end{cases} \tag{3}$$

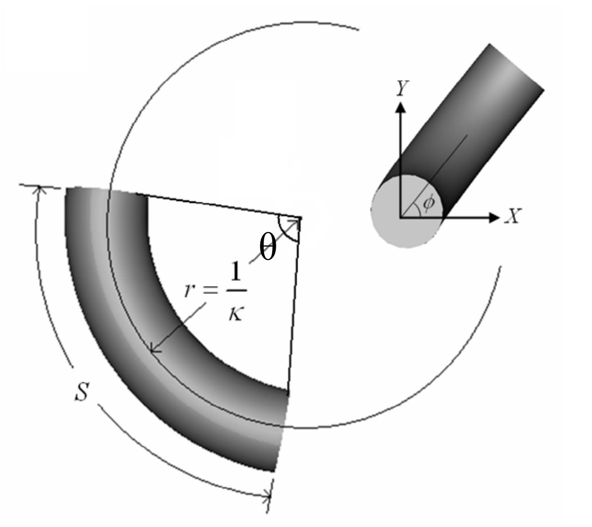

Fig. 3 – Manipulator variables s, κ, and ϕ, where ϕ gives the direction of bending measured in the *xy* plane, κ defines the curvature as the inverse of the trunk radius and s gives the length of the trunk.

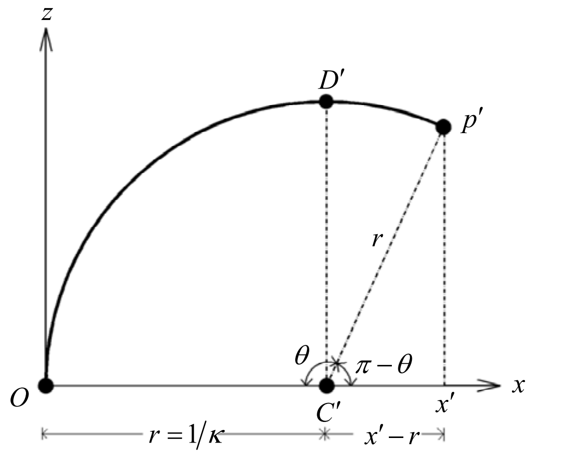

Fig. 4 – A single section of continuum trunk that lies entirely in *xz* plane obtained by rotating end-point *P* about the *z* axis by $-\phi$ (see Fig. 2). Observing $\triangle C'P'x'$ and applying law of cosines, $\cos(\pi-\theta)=(x'-r)/r$. Therefore, $\cos\theta=(\kappa^{-1}-x')/\kappa^{-1}$.

Knowing that length of arc is the product of the angle subtended by the arc and the radius of the arc, the length of the trunk section $s = r\theta$, where $r = 1/\kappa$ (see Fig. 3).

B. Special cases (singularities)

Endpoint coordinates along the z axis present singularities in the inverse kinematics calculations and can be grouped into three different cases: $z > 0$, $z = 0$, and $z < 0$. Coordinates along the z axis with $z > 0$ produce (correct) curvature values of zero; this creates a divide-by-zero condition in the arc-length calculation. When $x = 0$ and $y = 0$ the orientation calculation also produces the divide-by-zero condition. This case is easily handled by assigning ϕ to any arbitrary value and determining the arc length as $s = z$.

In the second case, when $P = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^{\mathrm{T}}$, multiple solutions exist as an arc forming a complete circle with any radius at any orientation satisfies this condition. In this case, choose $\theta = 2\pi$ and choose any value for ϕ and κ.

The last case occurs when P lies along the z axis where $z < 0$. This case poses an impossibility given the physical constraints of a continuum manipulator section, requiring a solution of $\kappa = 0$ and $s = z$ where ϕ is arbitrary.

III. Multi-section Kinematics

The inverse kinematics derived in the previous section can be iteratively applied to multiple, serially-linked continuum sections to model an n-section continuum manipulator.

A. Inverse Kinematics Algorithm

Given a list of end-points (one for each section), the values of s, κ, and ϕ can be computed for each section by determining the values of s, κ, and ϕ for the base section, subtracting the translation due to the base section from the remaining endpoints, applying the opposite rotation due to the base section to the remaining endpoints, and then repeating this process with the remaining sections. Recalling from [9] the rotation due to a single trunk section occurs about the axis $\underline{\omega} = \begin{bmatrix}-\sin\phi & \cos\phi & 0\end{bmatrix}^{\mathrm{T}}$ by the angle θ, the adjusted end-point coordinates can be expressed as $\underline{p}_{next} = \mathbf{R}_{\underline{\omega},-\theta}\left(\underline{p}_{next} - \underline{p}_{current}\right)$ where $\underline{p}_{current}$ is the end-point of the section whose s, κ, and ϕ values are currently being computed and $\underline{p}_{next}$ is the end-point of a remaining, distal section.

B. Incorporating Dead-Length Sections

Many actual continuum manipulator devices contain lengths of space between each section that do not bend. There are three ways to represent these 'dead' lengths as part of each section. The non-bending length of each section can be included at either end of the section or split between the two. Taking the approach of including the non-bending length at the end of each section, incorporating these 'dead' lengths can be easily handled by adding an appropriate translation at the beginning of each loop in the inverse algorithm. Following this method, simply subtract the vector $\begin{bmatrix}0 & 0 & l\end{bmatrix}^{\mathrm{T}}$ where l gives the dead length for the current section from $\underline{p}_{new}$ computed for the following sections, $\underline{p}_{next} = \mathbf{R}_{\underline{\omega},-\theta}\left(\underline{p}_{next} - \underline{p}_{current}\right) - \begin{bmatrix}0 & 0 & l_{current}\end{bmatrix}^{\mathrm{T}}$.

IV. End-point locations of each section for a multi-section continuum robot

An essential ingredient to applying the inverse kinematics in the previous section is the x, y, and z coordinate of the end-points of each section of the trunk in addition to the end-point of the trunk itself. This section presents an algorithm to assist in choosing these intermediate coordinates while also exposing structure of the solution space of the inverse kinematics problem, providing the possibility of using this solution space for choosing configurations of the trunk which avoid obstacles, minimize trunk curvature, or maximize some other desirable trunk characteristic.

The well-known difficulty of deriving the inverse kinematics for an arbitrary rigid-link robot stems from the complex nature of the non-linear equations involved. These complex non-linear equations can be resolved into simple

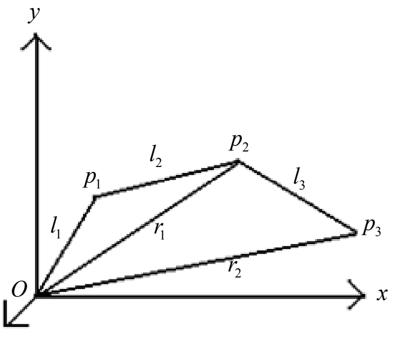

Fig. 5 – Figure showing the rigid-link configuration of a robot with link lengths l_1, l_2, and l_3. The tip of the robot lies at $\underline{p}_3$.

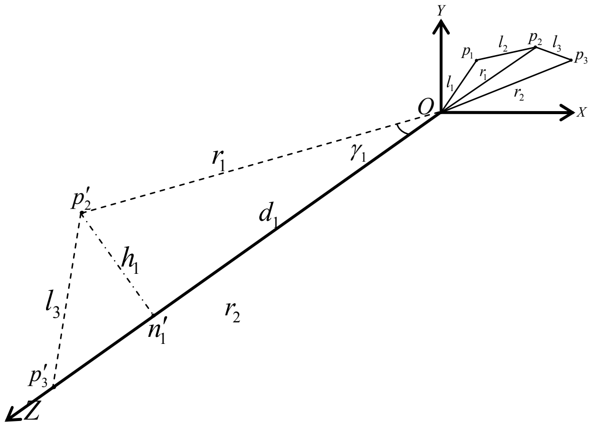

Fig. 6 – The rigid link robot after a transformation of coordinate frames from $OXYZ$ to $OX'Y'Z'$ to align $\overrightarrow{Op_3}$ along the +z axis. Links in the transformed coordinate frame are indicated by dashed lines. Point $\underline{p}_2'$ is placed in the yz plane and $\underline{p}_3'$ lies on the z axis; therefore, $\triangle Op_2'p_3'$ is present in the yz plane

inequalities for any rigid-link robot composed of spherical joints by following a geometric approach as detailed in [14, 15]. Observing that each section of a continuum robot consists of the equivalent of a spherical joint, this paper applies the solution procedure in [14, 15] to a three-section continuum robot by modeling it as a three-link rigid-link robot composed of spherical joints. The endpoints of each of the rigid links produced by this algorithm then provide the necessary endpoints for the multi-section inverse kinematics algorithm described in the previous section which fits a trunk to these endpoints.

A. Overview

To formally stating the problem solved in this section: given the endpoint $\underline{p}_3$ and the link lengths l_{1-3} of a three-link rigid-link robot composed of spherical joints, find the endpoints $\underline{p}_1$ and $\underline{p}_2$ of the first and second links of the rigid-link robot as shown in Fig. 5. The procedure begins by forming two triangles from $\triangle O\underline{p}_1\underline{p}_2$ and $\triangle O\underline{p}_2\underline{p}_3$ based on this information, where r_1 represents unknown length. Inequalities on r_1 given in (4) define one dimension of the resulting solution space. Choosing any value which satisfies these constraints completes the first step. Next, knowing the lengths r_1, r_2, and l_3 which define one triangle and the coordinate of two of its endpoints (O and $\underline{p}_3$), the second step gives the second dimension of the solution space as an arbitrary rotation of $\underline{p}_2$ about $\overrightarrow{Op_3}$ and computes a specific $\underline{p}_2$ given that rotation angle. In the final step, $\underline{p}_1$ is determined as a rotation of the other triangle about $\overrightarrow{Op_2}$, completing the solution.

B. Derivation

Given a desired end-point $\underline{p}_3$ and lengths l_1, l_2, and l_3 shown in Fig. 5 which specify fixed lengths of the straight lines joining the start-point and endpoint of sections one, two, and three respectively, this algorithm computes per-section endpoints $\underline{p}_1$ and $\underline{p}_2$. Referring to Fig. 5, length $r_2 = \left\|\underline{p}_3\right\|$ while triangle inequality theorems for $\triangle Op_2p_3$ and $\triangle Op_1p_2r_1$ bound length r_1 as

$$\begin{gathered} r_2 + l_3 \geq r_1 \geq \left|r_2 - l_3\right| \\ l_1 + l_2 \geq r_1 \geq \left|l_1 - l_2\right| \end{gathered} \tag{4}$$

Step 1: Choose any r_1 which satisfies the inequalities above. A complete solution space that includes all possible configurations of the robot can be built by repeating the rest of the derivation using all valid values of r_1. The equality sign observed in the inequalities (4) implies a "flat" triangle consisting of a single line and corresponds to a singular configuration of robot, as discussed in section V and illustrated in Fig. 8 and Fig. 9.

Step 2: With $\underline{p}_3$, r_1, r_2, and l_3 fixed, triangle $\triangle Op_2p_3$ constrains $\underline{p}_2$ to lie on a circle formed by rotating $\underline{p}_2$ about $\overleftrightarrow{Op_3}$. Choose any dihedral angle θ_1 which gives the rotation of $\triangle Op_2p_3$ about Op_3 and therefore determines the location of $\underline{p}_2$.

To calculate $\underline{p}_2$ from θ_1, first rotate the coordinate frame $OXYZ$ to $OX'Y'Z'$ in such a way that $\overrightarrow{Op_3'}$ aligns with the positive z axis. In this configuration, apply θ_1 as a rotation about the $+z$ axis. Finally, perform the inverse rotations to return to OXYZ with $\underline{p}_2$ now determined.

This initial transformation to $OX'Y'Z'$ can be achieved by performing two consecutive rotations first about the y axis then about the z axis by angles $-\beta_1$ and $-\alpha_1$ respectively, which are calculated in (7)-(10).

Due to this rotation, $\underline{p}_3'$ now lies on the $+z$ axis, at a distance of r_2 from O as shown in Fig. 6, making its location $\begin{bmatrix}0 & 0 & r_2\end{bmatrix}^{\mathrm{T}}$. Considering $\triangle Op_2'p_3'$, point $\underline{p}_2'$ can be any point on the circle around the z axis centered at n_1' with radius equal to height h_1. Since $\underline{p}_2'$ can be any point around the z

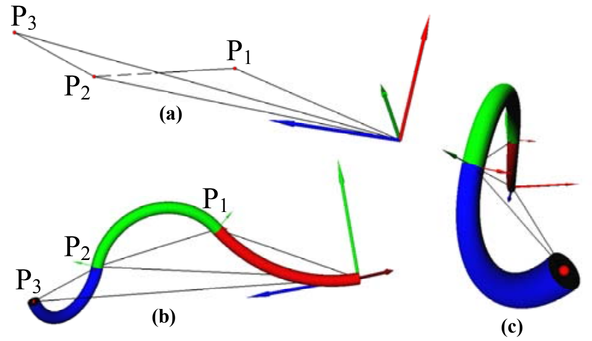

Fig. 7 – Results of the algorithms in section III.A and IV.B. Graphic (a) illustrates the result of IV.B exhibiting the orientation of triangles $\triangle Op_2p_3$ and $\triangle Op_1p_2$ in 3-D space. $\triangle Op_2p_3$ and $\triangle Op_1p_2$ lie in two different planes in space inclined at an angle to each other. The angles of orientation of the triangular planes are termed dihedral angles. Items (b) and (c) illustrate a continuum trunk fit to the skeleton in (a) from differing perspectives.

axis, begin by placing it in the *yz* plane. Applying the law of cosines, $\gamma_1 = \cos^{-1}\left(\left(r_2^2 + r_1^2 - l_3^2\right)/2r_1r_2\right)$. The location of $\underline{p}_2'$ is therefore $\begin{bmatrix}0 & h_1 & d_1\end{bmatrix}^{\mathrm{T}}$ where $d_1 = r_1 \cos\gamma_1$ and $h_1 = r_1 \sin\gamma_1$. After rotating $\underline{p}_2'$ about the *z* axis by θ_1, the following equation rotates the coordinate frame back to *OXYZ* to obtain the coordinates of $\underline{p}_2$:

$$\underline{p}_2 = \mathbf{R}_{\underline{z},\alpha_1}\mathbf{R}_{\underline{y},\beta_1}\mathbf{R}_{\underline{z},-\theta_1}\underline{p}_2', \tag{5}$$

where $\mathbf{R}_{\underline{\omega},\theta}$ represents a rotation about axis $\underline{\omega}$ by angle θ. Substituting the individual transformation matrices in equation (5) yields

$$\underline{p}_2 = \begin{bmatrix} c_{\theta_1}c_{\beta_1}c_{\alpha_1} + s_{\theta_1}s_{\alpha_1} & s_{\theta_1}c_{\beta_1}c_{\alpha_1} - c_{\theta_1}s_{\alpha_1} & s_{\beta_1}c_{\alpha_1} \\ c_{\theta_1}c_{\beta_1}s_{\alpha 1} - s_{\theta_1}c_{\alpha_1} & s_{\theta_1}c_{\beta_1}s_{\alpha_1} + c_{\theta_1}c_{\alpha_1} & s_{\beta_1}s_{\alpha_1} \\ -s_{\beta_1}c_{\theta_1} & -s_{\theta_1}s_{\beta_1} & c_{\beta_1} \end{bmatrix}\underline{p}_2{}', \tag{6}$$

where $c_\theta = \cos\theta$ and $s_\theta = \sin\theta$. The rotation angles, taken from a standard axis/angle rotation, are

$$\sin\alpha_1 = k_{1y}\Big/\sqrt{k_{1x}^2 + k_{1y}^2}, \tag{7}$$

$$\cos\alpha_1 = k_{1x}\Big/\sqrt{k_{1x}^2 + k_{1y}^2}, \tag{8}$$

$$\sin\beta_1 = \sqrt{k_{1x}^2 + k_{1y}^2}, \text{ and} \tag{9}$$

$$\cos\beta_1 = k_{1z}, \tag{10}$$

where $\underline{k}_1 = \begin{bmatrix}k_{1x} & k_{1y} & k_{1z}\end{bmatrix}^{\mathrm{T}}$ is a unit vector along $\overrightarrow{Op_3}$.

Step 3: Choose dihedral angle θ_2 which orients $\triangle Op_1p_2$ in 3-D space by following a similar process. After rotation such that $\overrightarrow{Op_2}$ is aligned with +*z* axis, applying the law of cosines produces $\gamma_2 = \cos^{-1}\left((r_1^2 + l_2^2 - l_1^2)/2r_1l_2\right)$. The location of $\underline{p}_1'$ is therefore $\begin{bmatrix}0 & h_2 & d_2\end{bmatrix}^T$ where $d_2 = l_1 \cos\gamma_2$ and $h_2 = l_1 \sin\gamma_2$. The position $\underline{p}_1 = \mathbf{R}\underline{p}_1'$ where $\mathbf{R}$ is given in (6) and $\underline{k}_1$ in (7)-(10) is replaced by $\underline{k}_2$, a unit vector along $\overrightarrow{Op_2}$.

V. Results

The OctArm continuum trunk [13] consists of a pneumatically-actuated, three-section, intrinsically-actuated trunk. Pressure regulation values control length and bending of each section while string encoders measure the resulting curvature for feedback to a PC-104-based control system. A remote PC accepts user input via joystick and relays desired trunk postures to the OctArm system; the remote PC also displays a real-time, 3D model of the expected trunk shape.

One difficulty faced when evaluating the trunk in the field [13] was the inability to command the trunk to avoid obstacles while maintaining tip position for insertion or inspection tasks. Although traditional Jacobian null-space techniques could be used, these lack a user-centric method of specifying how the trunk should be shaped to avoid these obstacles. To remedy this, the single-section kinematics described in this section were implemented by adding an additional control mode to the user interface routines described in [16]. In this mode, the operator can select trunk sections to move using the inverse kinematic algorithms given in section II and then control resulting trunk movement via the joystick.

A maneuver designed to illustrate potential obstacle avoidance techniques produced using these algorithms is pictured in Fig. 1. Beginning with a straight trunk shown in Fig. 1(a), the user then selected the second to last section of the trunk and moved it to the left via the joystick, while the algorithm kept all other points stationary. Thus shaped, the tip of the trunk could now be moved around an obstacle located at the bend in the second section.

To assess the suitability of the multi-section inverse kinematics algorithms presented for operation in real time, timing results for the multi-section algorithm obtained on a 3.0 GHz Pentium 4 show that the algorithm requires 0.3 ms to execute for a 3 section continuum robot, making it eminently suitable for real-time application. However, this algorithm has not yet been evaluated on the actual robot.

In addition, the algorithms and derivations in section III.A and IV.B were implemented in Matlab and visualized in a 3D graphics library. Fig. **7** shows a three section continuum manipulator starting at origin and reaching to $\begin{bmatrix}1 & 1 & 11\end{bmatrix}^{\mathrm{T}}$ with rigid-link lengths of $l_1 = 5$, $l_2 = 4$, and $l_3 = 3$ and dihedral angles $\theta_1 = 2\pi/3$ and $\theta_2 = 0$. The output of the procedure given in section IV.B is shown in Fig. **7**(a) where $\underline{p}_1$, $\underline{p}_2$, and $\underline{p}_3$ are indicated as small red spheres. This shows triangles $\triangle Op_2p_3$ and $\triangle Op_1p_2$ in 3D space with orientations $2\pi/3$ and 0 respectively. Fig. **7**(b) shows the final output from algorithm III.A where a continuum manipulator can be

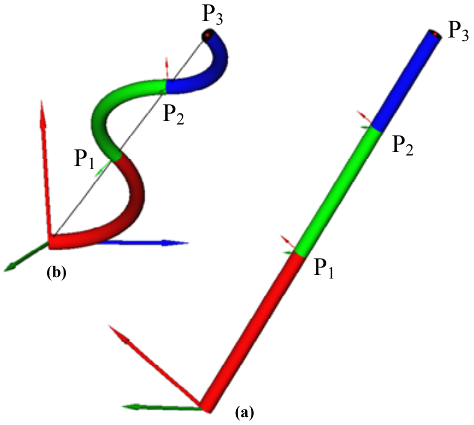

Fig. 8 – Simulation results illustrating the possible singular configurations. In (a), algorithms in both section III.A and in IV.B result in singular configuration where the rigid-link robot as well as the continuum trunk are stretched completely in order to reach the farthest tip location. In (b), only the rigid-link robot assumes a singular configuration with all the links extended in a straight line.

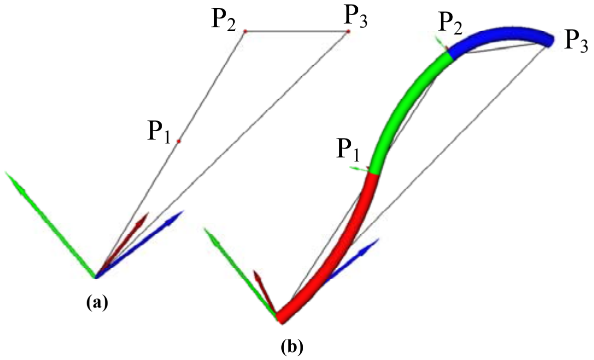

Fig. 9 – Item (a) shows a singular configuration of the rigid-link robot produced by a "flat" triangle configuration where $r_1 = l_1 + l_2$. Image (b) shows the continuum trunk developed from the knowledge of end-points given in (a). Red, green, and blue coordinate axis represent *x*, *y*, and *z* axis respectively. Red, green, and blue sections of trunk represent sections 1, 2, and 3 respectively.

seen along the skeleton obtained from applying derivation IV.B. Singular configurations explored in Fig. 9 illustrate a case with a "flat" triangle (when $r_1 = l_1 + l_2$); as a result $\triangle Op_1 p_2$ is no longer present. Two different singular configurations are illustrated in Fig. 8. Fig. 8(a) is a case where both III.A and IV.B achieve a singular configuration when robot is reaching the farthest point it can go to by stretching completely. Fig. 8(b) is a case where only IV.B produces a singular configuration with all rigid links in one line.

VI. Conclusion/Potential Applications

The unique nature of a hyper-redundant continuum trunk presents both daunting challenges and fascinating opportunities for grasping and manipulation of a wide range of objects. Given a desired tip position, algorithms presented in this paper provide a simple, closed-form solution to move a single trunk section (which possesses three degrees of freedom) to the given endpoint. The ability to choose the endpoint for each section of a multi-section trunk allows fine control of trunk shape for obstacle avoidance, grasping, and related tasks as illustrated in Fig. 1. Additional algorithms allow specification of a single end-point for the entire trunk and provide insight into the solution space of the system. These inverse kinematics tools provide a foundation for additional exploration into methods to make use of the marvelous dexterity present in continuum manipulators.

References

[1] D. Nenchev, "Redundancy resolution through local optimization: a review," *Journal of Robotic Systems,* vol. 6, pp. 769-798, 1989.

[2] B. Siciliano, "Kinematic control of redundant robot manipulators: a tutorial," *Journal of Intelligent and Robotic Systems,* vol. 3, pp. 201-212, Sept. 1990.

[3] G. Robinson and J. B. C. Davies, "Continuum robots - a state of the art," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, Michigan, 1999, pp. 2849-2854.

[4] B. A. Jones and I. D. Walker, "Kinematics for Multisection Continuum Robots," *IEEE Transactions on Robotics,* vol. 22, pp. 43-55, Feb. 2006.

[5] R. J. Webster, A. M. Okamura, and N. J. Cowan, "Toward Active Cannulas: Miniature Snake-Like Surgical Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 2857-2863.

[6] R. Buckingham, "Snake arm robots," *Industrial Robot: An International Journal,* vol. 29, pp. 242-245, 2002.

[7] H. Mochiyama, "Whole-arm impedence of a serial-chain manipulator," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, 2001, pp. 2223-2228.

[8] M. W. Hannan and I. D. Walker, "Kinematics and the Implementation of an elephant's trunk manipulator and other continuum style robots," *Journal of Robotic Systems,* vol. 20, pp. 45-63, Feb. 2003.

[9] S. Neppalli and B. A. Jones, "Design, Construction, and Analysis of a Continuum Robot," in *Proceedings of the International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 2007, pp. 1503-1507.

[10] I. A. Gravagne, C. D. Rahn, and I. D. Walker, "Large deflection dynamics and control for planar continuum robots," *IEEE/ASME Transactions on Mechatronics,* vol. 8, pp. 299-307, June 2003.

[11] P. Sears and P. Dupont, "A Steerable Needle Technology Using Curved Concentric Tubes," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 2850-2856.

[12] W. McMahan, B. A. Jones, and I. D. Walker, "Design and implementation of a multi-section continuum robot: Air-Octor," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, 2005, pp. 3345-3352.

[13] W. McMahan, B. A. Jones, V. Chitrakaran, M. Csencsits, M. Grissom, M. Pritts, C. D. Rahn, and I. D. Walker, "Field trials and testing of the OctArm continuum manipulator," in *Proceedings of the International Conference on Robotics and Automation*, Orlando, FL, USA, 2006, pp. 2336-2341.

[14] L. Han and L. Rudolph, "The inverse kinematics of a serial chain with joints under distance constraints," in *Proceedings of Robotics: Science and Systems (RSS)*, Philadelphia, Pennsylvania, USA, 2006.

[15] L. Han and L. Rudolph, "A unified geometric approach for inverse kinematics of a spatial chain with spherical joints," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, 2007, pp. 4420-4427.

[16] M. Csencsits, B. A. Jones, and W. McMahan, "User interfaces for continuum robot arms," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, 2005, pp. 3011-3018.