2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

Determining "Grasping" Configurations for a Spatial Continuum Manipulator

Jinglin Li and Jing Xiao

***Abstract*—Unlike a conventional articulated manipulator, where only the gripper manipulates objects, a continuum manipulator, such as a multi-section trunk/tentacle robot, is promising for deft manipulation of a wide range of objects of different shapes and sizes. Given an object, a continuum manipulator tries to grasp it by wrapping around and squeezing it. A main open problem is how to determine if the object can be grasped and if so, the whole-arm wrapping around configurations of the robot to grasp it, which we call *grasping configurations*.**

In this paper, we provide a general and complete analysis of grasping configurations of a spatial continuum manipulator consisting of three constant-curvature sections, for any given 3-D object. We formulate conditions for existence of solutions and describe how to determine valid grasping configurations. Our method can extend to general continuum manipulators of n constant-curvature sections (where $n \geq 3$).

I. Introduction

Continuum manipulators are usually defined to be those featuring continuous back bone structures, inspired by invertebrate structures found in nature, such as octopus arms [9] and elephant trunks [2]. The OctArm manipulator is such a continuum manipulator (Fig. 1). While a conventional manipulator has an articulated arm with a gripper, there is no divide between the "arm" and "hand/gripper" for a continuum manipulator. A "grasping" configuration for a continuum manipulator affects the whole arm as it wraps around an object. So far there is no systematic study on the existence and forms of grasping configurations for a continuum manipulator in the literature: coiling for grasping was studied in [4], some empirical [10] and heuristic [18] methods were proposed, and there was also a study on human controlled operations [13]. In related work, shape optimization and control for hyperredundant robots were studied [1], [12]. Caging was studied for grasping with conventional manipulators [3], [8].

Virtually all continuum robots feature constant-curvature sections (modulo external loading due to gravity or payload) [17] because of actuating the (theoretically infinite) degrees of freedom of the continuously bendable backbone with finite actuators. This means that, whether with pneumatics like the OctArms or via tendons, forces are applied at a finite number of locations, and between those locations, the internal backbone forces "even out", making the backbone shape tend to constant curvature, i.e., circular, sections. The OctArm, with constant-curvature sections, is representative of the general class of continuum robots developed by researchers [16], [17]. Thus the work presented in this paper can easily apply or extend to other continuum robots.

J. Li is a PhD student in Computer Science, University of North Carolina at Charlotte, USA. jli41@uncc.edu

J. Xiao is with the faculty of Computer Science, University of North Carolina at Charlotte, USA. xiao@uncc.edu

Fig. 1. An OctArm manipulator (by the courtesy of Ian Walker)

In this paper, we address the open problem of how to find configurations of a spatial OctArm manipulator to grasp (i.e., wrap tightly around) a 3-D object. We first describe object models for such kind of grasping, then present the conditions of grasping configurations, and next introduce a systematic and efficient method to find all types of valid grasping configurations.

II. Manipulator Model

The OctArm consists of three *constant-curvature* sections (see Fig. 1). Each section is like a cylinder bended into a circular shape (when not in contact), with its central axis bended into a circular curve. All section cylinders have approximately the same width (i.e., the cylinder diameter) w. We can represent each section $i, i = 1, 2, 3$, in terms of its central circular axis with two end points: a *base* point p_{i-1} and a *tip* point p_i. We call the circle that section i's central axis curves along the *section i's circle*. When we refer section i, we mean the curve of its central axis except for situations where the section width has to be considered (to be explained in Section III below).

The base frame of the robot is set at p_0 with z_0 axis tangent to section 1's circle. The section i's frame is formed at p_{i-1} with the z_i axis tangent to the section circle at p_{i-1}. The base of section i is the tip of section i-1. Two adjacent sections i-1 and i are connected *tangentially* at the connection point p_{i-1}, i.e., the two sections share the same tangent at p_{i-1}. For clarity, we consistently use black, red, and green colors to draw section 1, section 2, and section 3 of the OctArm in this paper.

978-1-61284-456-5/11/$26.00 ©2011 IEEE

Fig. 2 illustrates the three sections of the OctArm and their respective frames. Because of the mechanical structure, section i of the OctArm can bend either along the $+z_i$ axis or the $-z_i$ axis but *not* both.

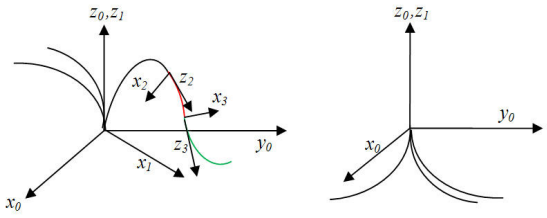


Fig. 2. OctArm manipulator frames. Note that section 1 can bend along either (a) $+z_1$ axis (with different orientations) or (b) $-z_1$ axis (with different orientations), but *not* both directions of z_1.

Note that the circle center of section i, c_i, always lies on the x_i axis, with $\mp 1/\kappa_i$ being the x coordinate in the i-th frame, where κ_i is the curvature. Note also that c_i lies on the positive x_i axis if $\kappa_i < 0$ (see Fig. 3) and on the negative x_i axis if $\kappa_i > 0$. When $\kappa_i = 0$, section i is a straight-line segment starting from the origin p_{i-1} and along the z_i axis.

Although each section can bend passively anywhere, it has a finite number of degrees of freedom that can be directly changed by the OctArm actuators [6], which are controllable variables: curvature κ_i, length s_i, and orientation angle ϕ_i from the plane of section i-1 to that of section i about z_i axis, i.e., the angle from y_{i-1} axis to y_i axis about z_i axis. Fig. 3 showed one example section i, its frame, and controllable variables.

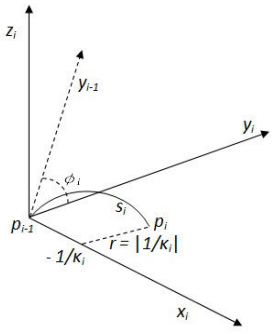


Fig. 3. Section i, its frame, and its variables κ_i, length s_i, and ϕ_i

A *configuration* of the OctArm can be expressed by the controllable variables as $[\kappa_1, s_1, \phi_1, \kappa_2, s_2, \phi_2, \kappa_3, s_3, \phi_3]^T$. Thus, we can treat this (κ, s, ϕ) space the *configuration space* of the OctArm robot. Given the position of the base point p_{i-1}, and κ_i, s_i, and ϕ_i values, the position of the tip point p_i of the section can be computed [6].

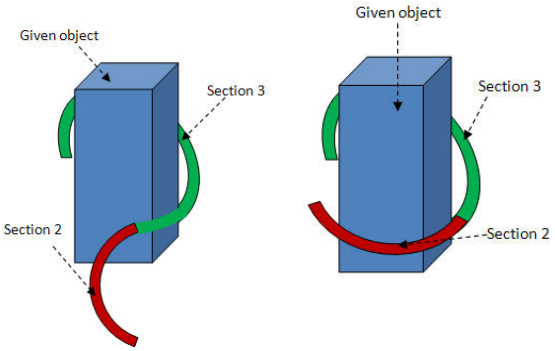
The values of κ_i, s_i and ϕ_i are within finite ranges: $[\kappa_{i,-max}, \kappa_{i,max}]$, including κ_i=0, $[s_{i,min}, s_{i,max}]$, and $[\phi_{i,min}, \phi_{i,max}]$ respectively. Specifically, for the OctArm, the range of ϕ_i is $[-\pi, \pi]$ for all sections, and the ranges of κ_i and s_i are:

- $\kappa_1 \in [-0.0189, 0.0228]$ (1/cm), $s_1 \in [28, 42]$ (cm),
- $\kappa_2 \in [0.0327, 0.0379]$ (1/cm), $s_2 \in [26.5, 44]$ (cm),
- $\kappa_3 \in [-0.045, 0.0808]$ (1/cm), $s_3 \in [32.5, 53.5]$ (cm).

III. Object Grasping Models

For a given 3D object, we define three *grasping models* for the object:

- Grasping Model 1: section 3 of the robot wraps around the object, and other sections of the robot may wrap around the object but not along section 3's circle. See Fig. 4.
- Grasping Model 2: both section 2 and section 3 of the robot wrap around the object along the same circle. See Fig. 5.
- Grasping Model 3: all sections of the robot wrap around the object along the same circle.



(a) Only section 3 wraps around the object. (b) Section 2 also wraps around the object along a different circle.

Fig. 4. Two examples of Grasping Model 1

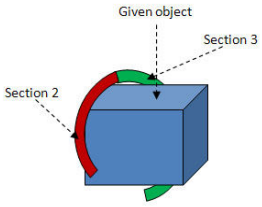


Fig. 5. Grasping Model 2 where section 2 (red) and section 3 (green) are on the same circle

To realize these grasping models, we need to first obtain bounding circles of the object, and then choose feasible bounding circles such that either the section 3 of the robot, or the combined sections 2 and 3 of the robot, or all sections of the robot can wrap around the object, i.e., make sure that the value ranges of the section curvature and length of the robot allow one of the grasping models to happen. Once a suitable object bounding circle for a grasping model is obtained, we can

further determine (in subsequent sections of the paper) suitable configurations of the entire robot that realize the corresponding grasping model, which we call, *grasping configurations*.

Different bounding circles of an object can be obtained from different cross sections of the object. A bounding circle cir_o can be obtained automatically with the following steps:

- Select a plane through the center of mass of the object[1].
- Intersect the plane with the object to create a cross-section, which can be expressed as a polygon.
- Find the minimum bounding circle cir_o of the cross section polygon [7].

Next, we need to check if the bounding circle cir_o is of suitable size for the robot to grasp. Let r_o be the radius of cir_o. We grow cir_o by $w/2 - \delta$ to take into acount the width w of the actual robot (see Section II). Note that δ is a small value to ensure tight wrap, taking advantage of the inherent compliance and small deformation of the OctArm. For convenience, we call such a grown circle cir_o the *object circle*. If one of the following conditions is satisfied, cir_o has a suitable size:

$$s_{3,min} \leq a\pi r_o \leq s_{3,max} \tag{1}$$

$$s_{3,min} + s_{2,min} \leq a\pi r_o \leq s_{3,max} + s_{2,max} \tag{2}$$

$$s_{3,min} + s_{2,min} + s_{1,min} \leq a\pi r_o \leq s_{3,max} + s_{2,max} + s_{1,max} \tag{3}$$

where $a \in (0, 2]$ is a coefficient determining how much the object bounding circle has to be wrapped. Its value depends on the shape, size, and material characteristics of the target object, as well as on the task of manipulation. For example, the task of pulling an object could require a smaller a than that of picking up the object. Conditions (1), (2),or (3) determine suitable cir_o for the Grasping Model 1, 2, or 3 respectively, where the respective (central axes of) robot section(s) can curve along.

Note that a configuration of an OctArm is in terms of only controllable variables of its sections (as defined in Section II). However, the OctArm can bend anywhere passively (i.e., with infinite passive degrees of freedom). The smooth and compliant nature of such a continuum structure allows it to gently interact with the object by adapting its shape to the object it wraps. Therefore, for any of the grasping models, once a corresponding configuration of the OctArm is found, the inherent compliance of the arm will allow a tight wrap with as much continuum contact as possible (hence as much friction as possible) to make the grasp robust [11].

We now consider how to find grasping configurations for each grasping model. Grasping Model 3 presents the trivial case: by setting the circle center and radius of each section of the robot as the same as the center and radius of the given object circle respectively, a unique grasping configuration can be found – the length of each section can be set to its maximum. We focus on the cases for Grasping Model 1 and Grasping Model 2 respectively in the rest of the paper.

[1]Or of the preferred grasping part, such as a handle, depending on the object.

IV. Finding Configurations For Grasping Model 1

For Grasping Model 1, the object circle (described in Section III) is the desired section 3's circle for grasping. Now, given this section 3's circle, i.e., its center position $\mathbf{c}_3$, radius r_3 (or curvature $|\kappa_3|$), and the circle plane normal, which can be expressed as $\mathbf{y}_3$ – the unit vector of the y axis of the section 3's frame, we need to solve for values of the other OctArm configuration variables: κ_1 (or r_1), ϕ_1, s_1, κ_2 (or r_2), ϕ_2, and s_2.

There are two situations of configurations for Grasping Model 1, where the section 3's circle is given and not shared by section 2:

- *general situation*: section 1 and section 2 do not share the same circle;
- *special situation*: section 1 and section 2 share the same circle.

We focus on finding grasping configurations of the general situation in this section. We will first specify constraints relating sections of the OctArm for the Grasping Model 1 and then use those constraints to solve for the unknowns to obtain grasping configurations.

The special situation can be handled in a way similar to that for determining grasping configurations for Grasping Model 2, which is described in Section V.

A. Inter-section Constraints

For Grasping Model 1, in the general situation where no two sections of the OctArm share the same circle, the section circles have to satisfy the following:

- the section 1's circle is tangent to the section 2's circle at point p_1, which is also the end point of section 1, and
- the section 2's circle is tangent to the section 3's circle at point p_2, which is also the end point of section 2.

p_1 and p_2 are both on section 2's circle. Let l_{12} and l_{23} be the tangent lines going through p_1 and p_2 respectively, as shown in Fig. 6. They must satisfy the following two constraints:

- **Constraint 1**: the tangent lines l_{12} and l_{23} must be coplanar.
- **Constraint 2**: the tangent lines l_{12} and l_{23} must be on the same circle.

Define two vectors along l_{12} and l_{23} respectively as the following:

$$\mathbf{l}_{12} = \mathbf{y}_1 \times (\mathbf{p}_1 - \mathbf{c}_1) \tag{4}$$

and

$$\mathbf{l}_{23} = \mathbf{y}_3 \times (\mathbf{p}_2 - \mathbf{c}_3) \tag{5}$$

where $\mathbf{y}_1$ and $\mathbf{y}_3$ are the unit vectors of the y axes of section 1 and section 3 respectively, which also represent plane normals of the two sections respectively; $\mathbf{c}_1$ and $\mathbf{c}_3$ are the position vectors of the centers of section 1's and section 3's circles respectively; $\mathbf{p}_1$ and $\mathbf{p}_2$ are the position vectors of p_1 and p_2 respectively, all in the robot's base frame. $\mathbf{y}_1$ can be further expressed in terms of ϕ_1 as:

$$\mathbf{y}_1 = R(z_1, \phi_1)\mathbf{y}_0 \tag{6}$$

where $R(z_1, \phi_1)$ is the rotation matrix of the rotation about the z_1 axis with angle ϕ_1, and $\mathbf{y}_0$ is the (fixed) y axis of the base frame of the robot.

$\mathbf{c}_1$ can be expressed in terms of ϕ_1 and κ_1 as:

$$\mathbf{c}_1 = R(z_1, \phi_1)[-\frac{1}{\kappa_1}, 0, 0]^T \tag{7}$$

where $[-\frac{1}{\kappa_1}, 0, 0]^T$ is the center position of section 1 expressed in section 1's frame.

Now, for the case that l_{12} and l_{23} are *not parallel*, **Constraint 1** can be expressed as:

$$(\mathbf{p}_2 - \mathbf{p}_1) \cdot (\mathbf{l}_{12} \times \mathbf{l}_{23}) = 0 \tag{8}$$

Let α_1 be the angle between $\mathbf{p}_2 - \mathbf{p}_1$ and $\mathbf{l}_{12}$ and α_2 be the angle between $\mathbf{p}_2 - \mathbf{p}_1$ and $\mathbf{l}_{23}$, as shown in Fig. 7, such that[2]

$$sin(\alpha_1) = \frac{\|\mathbf{l}_{12} \times (\mathbf{p}_2 - \mathbf{p}_1)\|}{\|\mathbf{l}_{12}\|\|\mathbf{p}_2 - \mathbf{p}_1\|} \tag{9}$$

and

$$sin(\alpha_2) = \frac{\|\mathbf{l}_{23} \times (\mathbf{p}_2 - \mathbf{p}_1)\|}{\|\mathbf{l}_{23}\|\|\mathbf{p}_2 - \mathbf{p}_1\|} \tag{10}$$

Then, to satisfy **Constraint 2**, α_1 and α_2 must be either equal or complementary (see Fig. 7), depending on the directions of $\mathbf{l}_{12}$ and $\mathbf{l}_{23}$; that is, the following equation must be satisfied:

$$sin(\alpha_1) = sin(\alpha_2) \tag{11}$$

For the case that l_{12} and l_{23} are *parallel*, then **Constraint 1** is satisfied since

$$\mathbf{l}_{12} \times \mathbf{l}_{23} = 0 \tag{12}$$

and **Constraint 2** can be expressed as:

$$(\mathbf{p}_2 - \mathbf{p}_1) \cdot \mathbf{l}_{12} = 0. \tag{13}$$

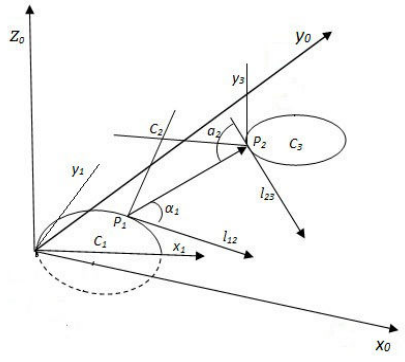


Fig. 6. Two tangent lines l_{12} and l_{23} pass p_1 and p_2 respectively.

Since p_1 and p_2 are on section 1's circle and section 3's circle respectively, each can be expressed in terms of a scalar angle as derived below. Define a local coordinate system for section i's circle, as illustrated in Fig. 8, such that its origin is at the circle center c_i, and two unit vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ form the orthogonal axes on the circle plane. $\mathbf{u}_i$ and $\mathbf{v}_i$ are functions of $\mathbf{y}_i$; Denote $r_i = \frac{1}{|\kappa_i|}$ as the radius of the circle for section i.

[2]by the definition of cross product.

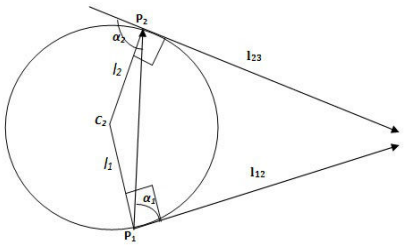


Fig. 7. The tangent lines l_{12} and l_{23}, normal lines l_1 and l_2, and section 2's circle are on the same plane; α_1 and α_2 are complementary.

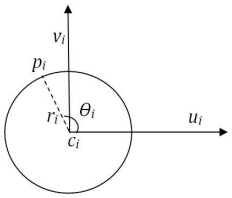


Fig. 8. The local coordinate system of section i's circle

Thus, since p_1 is on the section 1's circle, its position vector (in the robot base frame) must satisfy:

$$\mathbf{p}_1 = \mathbf{c}_1 + r_1 cos(\theta_1)\mathbf{u}_1 + r_1 sin(\theta_1)\mathbf{v}_1 \tag{14}$$

where θ_1 is the angle from the vector $\mathbf{u}_1$ to $\mathbf{p}_1 - \mathbf{c}_1$.

Similarly, since p_2 is on the section 3's circle, its position vector must satisfy:

$$\mathbf{p}_2 = \mathbf{c}_3 + r_3 cos(\theta_3)\mathbf{u}_3 + r_3 sin(\theta_3)\mathbf{v}_3 \tag{15}$$

where θ_3 is the angle from the vector $\mathbf{u}_3$ to $\mathbf{p}_2 - \mathbf{c}_3$.

B. Finding solutions

From the above, the inter-section constraint equations for the case where the two tangent lines l_{12} and l_{23} are not parallel are (8) and (11). The constraint equations for the case where l_{12} and l_{23} are parallel are equations (12) and (13).

With equations (4) to (11), we can re-write the inter-section constraint equations in terms of four variables: κ_1, ϕ_1, θ_1, and θ_3. With two equations and four variables in either case, we can solve for two variables with the other two variables assuming any values within their value ranges. If we assign values to κ_1 and ϕ_1, i.e., specify the section 1's circle, we can solve for θ_1 and θ_3. Once θ_1 and θ_3 are solved, we can further solve for section 2's circle and then the corresponding grasping configuration.

We describe the process in turn below.

*1) **Solving for** θ_1 **and** θ_3:* Now we describe how to solve for θ_1 and θ_3 for a given pair of κ_1 and ϕ_1 values within their respective ranges. Since the constraint equations (in either the non-parallel or the parallel case) written in terms of θ_1 and θ_3 are high-order, non-linear trigonometric equations of those two variables, we can only obtain numerical solutions (by Newton's method), which require good initial guesses.

In order to be efficient, we narrow down the range of search for initial guesses to be those that satisfy the length ranges of the continuum manipulator. We first discretize both θ_1 and θ_3, ranging from 0 to 2π, into small intervals. For each pair of θ_1 and θ_3 values, we then use equations (14) and (15) to compute the corresponding $\mathbf{p}_1$ and $\mathbf{p}_2$.

If $\mathbf{p}_1$ and $\mathbf{p}_2$ do not satisfy the following length constraints,

$$s_{2,max} \geq ||\mathbf{p}_2 - \mathbf{p}_1|| \geq s_{2,min} \tag{16}$$

and

$$s_{1,max} \geq ||\mathbf{p}_1|| \geq s_{1,min} \tag{17}$$

we discard the corresponding θ_1 and θ_3.

From those pairs of θ_1 and θ_3 that satisfy inequalities (16) and (17), we search for pairs that approximately satisfy the constraint equations as initial guesses of θ_1 and θ_3.

With the initial guesses, for the case where l_{12} and l_{23} are not parallel and the case where l_{12} and l_{23} are parallel, the corresponding inter-section constraints in terms of θ_1 and θ_3 are then solved numerically.

*2) **Determining grasping configurations**:* Once θ_1 and θ_3 are solved, the corresponding $\mathbf{p}_1$ and $\mathbf{p}_2$ can be solved from equations (14) and (15), and $\mathbf{l}_{12}$ and $\mathbf{l}_{23}$ can be obtained from equations (4) and (5) respectively. Next we need to solve for the plane, center, and radius of section 2's circle.

In the case where two tangent lines l_{12} and l_{23} are parallel, the center c_2 of section 2's circle is on the same line as p_1 and p_2, and thus, its position satisfies:

$$\mathbf{c}_2 = \frac{(\mathbf{p}_1 + \mathbf{p}_2)}{2} \tag{18}$$

In the case where l_{12} and l_{23} are not parallel, l_{12} and l_{23} determines the plane of the section 2. Line l_1 through p_1, perpendicular to l_{12}, and on the plane of section 2 can be determined, and similarly line l_2 through p_2, perpendicular to l_{23}, and on the plane of section 2 can be determined as shown in Fig. 7. Then the intersection point of l_1 and l_2 is the center c_2 of section 2's circle.

In both cases, The radius of section 2's circle is:

$$r_2 = \|\mathbf{p}_1 - \mathbf{c}_2\| \tag{19}$$

From section 1's circle (for the specified values of κ_1 and ϕ_1) and section 2's circle, as well as p_1 and p_2, which are end-points of section 1 and section 2 respectively, the unknown configuration parameters of section 1 and section 2: s_1, κ_2, s_2, and ϕ_2 can be found easily [14]. They are valid if their values are within their respective ranges. The length s_3 of section 3 can be set to its maximum value to maximize wrapping along the section 3's circle (which is given). Now a grasping configuration of the entire OctArm is found for Grasping Model 1.

V. Finding Configurations for Grasping Model 2

For Grasping Model 2, the given object circle (described in Section III) is shared by both section 2 and section 3, i.e., the section 3's circle and section 2's circle are the same, with the following known parameters: the circle center position $\mathbf{c}_3 = \mathbf{c}_2$, the radius $r_3 = r_2$, the unit normal vector of the circle plane $\mathbf{y}_2$, and $\phi_3 = 0$ or π. We only need to find the configuration parameters of section 1 that satisfy the Grasping Model 2.

Section 1's circle and section 2's circle must be tangent at point p_1, sharing the same tangent line l_{12} through p_1. Now, axis z_1 is also tangent to the section 1's circle and is the same as the z_0 axis of the robot base frame, with unit vector $\mathbf{z}_1$. Thus, l_{12} must satisfy the following two constraints:

- **Constraint A**: l_{12} and axis z_1 must be coplanar.
- **Constraint B**: l_{12} and z_1 must be on the same circle.

Define a vector along l_{12} as:

$$\mathbf{l}_{12} = \mathbf{y}_2 \times (\mathbf{p}_1 - \mathbf{c}_2) \tag{20}$$

For the case of non-parallel l_{12} and z_1, **Constraint A** can be expressed as:

$$\mathbf{p}_1 \cdot (\mathbf{l}_{12} \times \mathbf{z}_1) = 0 \tag{21}$$

The expression of **Constraint B** can be obtained similarly to **Constraint 2** in Section IV.A as:

$$\frac{\|\mathbf{l}_{12} \times \mathbf{p}_1\|}{\|\mathbf{l}_{12}\|} = \frac{\|\mathbf{z}_1 \times \mathbf{p}_1\|}{\|\mathbf{z}_1\|} \tag{22}$$

For the case of parallel l_{12} and z_1, the above two constraints become

$$(\mathbf{l}_{12} \times \mathbf{z}_1) = 0 \tag{23}$$

and

$$\mathbf{p}_1 \cdot \mathbf{z}_1 = 0 \tag{24}$$

Since $\mathbf{p}_1$ is on section 2's circle, it satisfies:

$$\mathbf{p}_1 = \mathbf{c}_2 + r_2 cos(\theta_2)\mathbf{u}_2 + r_2 sin(\theta_2)\mathbf{v}_2 \tag{25}$$

where $\mathbf{u}_2$ and $\mathbf{v}_2$ are functions of the given $\mathbf{y}_2$. By substituting equation (20) for $\mathbf{l}_{12}$ and then equation (25) for $\mathbf{p}_1$ in the constraint equations above, we can re-write the two constraint equations in each case in terms of a single variable θ_2. Solving the equations numerically, in a fashinon similar to that described in Seciton IV.B.1), yields at most two solutions of θ_2 (in either the non-parallel or parallel case).

Next $\mathbf{p}_1$ and $\mathbf{l}_{12}$ can be solved via equations (25) and (20), from which and $\mathbf{z}_1$, the section 1's circle and configuration variables ϕ_1, κ_1, and s_1 can be solved in a similar fashion as described in Section IV.B.2. There are at most two grasping configurations to realize a given Grasping Model 2.

Note that with analysis and derivation similar to the above, we can find at most two grasping configurations of the special situation where section 1 and section 2 share the same circle for a given Grasping Model 1 (see the beginning of Section IV).

VI. Implementation and Discussion

Algorithm 1 implements the method introduced above for finding grasping configurations for Grasping Models 1 and 2. For Grasping Model 1, since there are fewer constraint equations than the variables in the general situation, solutions of grasping configurations (section IV) depend on values of κ_1 and ϕ_1, which are input to the algorithm.

Algorithm 1 also include checking for penetrations of the robot into the target object for a found grasping configuration. Specifically, it checks if section 2 or section 1 of the robot penetrates into the object. See Fig. 9 for an example. If so, the grasping configuration is not valid and should be discarded. However, if section 2 or section 1 almost touch the object, the grasping configuration is even preferred because it means a tighter wrap, as shown in Fig. 4(b), recalling that the object circle is made deliberately smaller (by δ) to allow tight wrap with compliance (see Section III). The collision-checking method is similar to that in [18]. Other efficient collision-checking methods can also be applied [5].

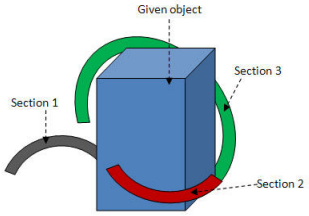


Fig. 9. A grasping configuration in collision with the object

Algorithm 1 Finding grasping configurations

input object circle with known $\mathbf{c}_3$, $\mathbf{y}_3$, and r_3
if Grasping Model 1 **then**
  input κ_1 and ϕ_1
  choose suitable initial guesses θ_{1guess}, θ_{3guess};
  solve for θ_1 and θ_3;
else
  if Grasping Model 2 **then**
    $\mathbf{c}_2 = \mathbf{c}_3$, $\mathbf{y}_2 = \mathbf{y}_3$, and $r_2 = r_3$;
    choose suitable initial guesses θ_{2guess};
    solve for θ_2;
  end if
end if
compute the corresponding grasping configurations $(\kappa_1, \phi_1, s_1, \kappa_2, \phi_2, s_2, \kappa_3, \phi_3, s_3)$;
for each grasping configuration **do**
  if the configuration passes penetration check **then**
    record this grasping configuration;
  end if
end for
return all recorded grasping configurations.

Given a Grasping Model 1, for one given pair of values κ_1=a and ϕ_1=ψ, if there are corresponding grasping configurations found by Algorithm 1, then because κ_1 and ϕ_1 can change values continuously, for each found grasping configuration $\mathbf{C}(a, \psi)$, there exists a small continuous neighborhood $B(\mathbf{C})$ of configurations corresponding to a small continuous neighborhood:

$$D(a, \psi) = \{\kappa_1, \phi_1 | \delta\kappa_1 > |\kappa_1 - a|, \delta\phi_1 > |\phi_1 - \psi|\}$$

where $\delta\kappa_1$ and $\delta\phi_1$ are small positive values. All configurations in $B(\mathbf{C})$ have very similar shapes to the found configuration $\mathbf{C}(a, \psi)$. However, for κ_1 and ϕ_1 values outside $D(a, \psi)$, their corresponding grasping configurations can be disconnected from $B(\mathbf{C})$ and belong to another continuous neighborhood of configurations, as illustrated in Fig. 10. Note that for the same pair of κ_1 and ϕ_1 values, there can be multiple solutions of grasping configurations, and each belongs to a different continuous neighborhood. This is also illustrated in Fig. 10, where there exists points in $D(a_1, \psi_1)$, each of which corresponds to two grasping configurations in $B(\mathbf{C}_1)$ and $B(\mathbf{C}_2)$ respectively.

There are a finite number of such neighborhoods of grasping configurations for a Grasping Model 1.

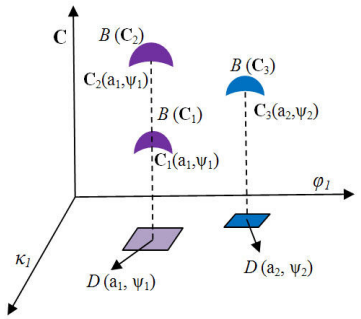


Fig. 10. Grasping configurations vs. κ_1 and ϕ_1. For $(\kappa_1, \phi_1) = (a_i, \psi_i)$, $i = 1, 2$, the corresponding configurations are $\mathbf{C}_j(a_i, \psi_i)$ in neighborhood $B(\mathbf{C}_j)$, $j = 1, 2, 3$, which corresponds to neighborhood $D(a_i, \psi_i)$. $B(\mathbf{C}_1)$ and $B(\mathbf{C}_2)$ both correspond to $D(a_1, \psi_1)$.

By discretizing the value ranges of κ_1 and ϕ_1 with a proper resolution, we can solve for grasping configurations representing all the neighborhoods for the case of Grasping Model 1. Based on the value ranges of κ_1 and ϕ_1 (for the OctArm – see section II), we find it reasonable to set the discretization interval to be 3% of each range. For each given Grasping Model 1, by running Algorithm 1 for each pair of κ_1 and ϕ_1 from the discretization, at least one representative configuration in each neighborhood of grasping configurations can be found. Note that for a given object circle, the actual value ranges of κ_1 and ϕ_1 that can possibly lead to grasping configurations can be smaller than the respective maximum value ranges for those parameters.

VII. Results

We present three examples here. We first describe types of grasping configurations and then provide representative grasping configurations found for those examples.

We can classify the OctArm grasping configurations based on the orientations ϕ_2 (i.e., the angle between planes of section 1 and section 2) and ϕ_3 into the following four types:

(1) ϕ_2 is negative and ϕ_3 is positive.
(2) ϕ_2 is positive and ϕ_3 is negative.
(3) ϕ_2 and ϕ_3 are both positive.
(4) ϕ_2 and ϕ_3 are both negative.

Table I shows the descriptions of three object circles, two of them are of Grasping Model 1, in terms of circles for section 3 (see Fig. 11), and the other one is of Grasping Model 2, in terms of a circle that both section 3 and section 2 share.

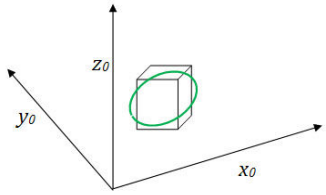
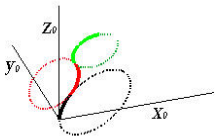
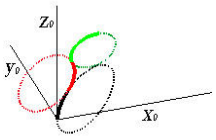


Fig. 11. Grasping Model 1: a section 3's circle bounding an object

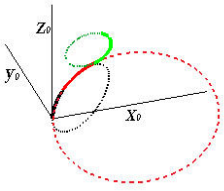
For the object circle specified in row 1 of Table I, four valid solutions are obtained by Algorithm 1 as illustrated in Fig. 12, where solutions 1 and 2 belong to the type (1), and solutions 3 and 4 belong to the type (2), and corresponding configurations are shown in Table II.



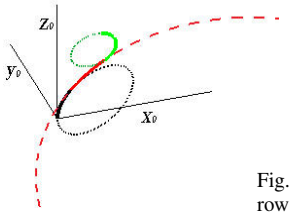
(a) sol.1 belongs to type (1)



(b) sol.2 belongs to type (1)



(c) sol.3 belongs to type (2)



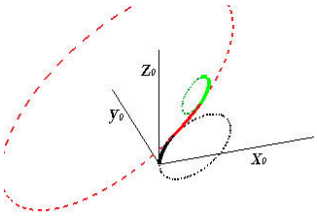
(d) sol.4 belongs to type (2)

Fig. 12. Representative solutions for the object circle in row 1 of Table I.

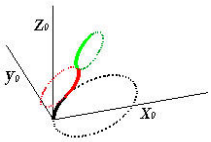
Note that solutions 1 and 2 belong to the same neighborhood where ϕ_1 changes continuously from 0.4 to 0.6 when $\kappa_1 = 0.0228$. Note also that solutions 2 and 3 are of different types but share the same pair of κ_1 and ϕ_1 values. This is one example to show that there can be multiple solutions for the same κ_1 and ϕ_1, which belong to separate (continuous) neighborhoods (see Fig. 10 for an illustration). For this particular case, there are at most three neighborhoods of grasping configurations.

For the object circle specified in row 2 of Table I, two solutions found by Algorithm 1 are illustrated in Fig. 13, where solution 1 belongs to type (3) and solution 2 belongs to type (4). Therefore, in this case, there are only two separate neighborhoods of grasping configurations. Corresponding configurations are also shown in Table II.

For the object circle of Grasping Model 2 in row 3 of Table I, there is a unique solution, and the solution is shown in Fig. 14, where the grasping configuration is shown in Table II.



(a) sol.1 belongs to type (3)



(b) sol.2 belongs to type (4)

Fig. 13. Representative solutions for the object circle in row 2 of Table I.

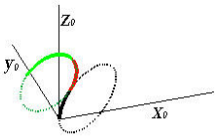


Fig. 14. The unique solution for the object circle of Grasping Model 2 in row 3 of Table I.

For the example object circles of Grasping Model 1 in Table I, the average time to run Algorithm 1 for finding grasping configurations corresponding to one pair of κ_1 and ϕ_1 is $0.05s$. The total time to find representative grasping configurations of all neighborhoods of configurations is $4s$ to $7s$ per example. This shows that our method is suitable for on-line determination of grasping configurations for the OctArm for a given object.

TABLE I
EXAMPLE OBJECT CIRCLES

Grasping Model	Circle Center	Radius	Normal
1	$[45.0cm, 37.5cm, 60cm]^T$	$22.5cm$	$[-0.55cm, 0cm, 0.83cm]^T$
1	$[45.0cm, 37.5cm, 60cm]^T$	$22.5cm$	$[-0.84cm, 0.24cm, 0.48cm]^T$
2	$[6.0cm, 34.5cm, 24.75cm]^T$	$30.83cm$	$[-0.76cm, 0.013cm, 0.64cm]^T$

TABLE II
GRASPING CONFIGURATIONS IN FIGURES 12– 14

Grasping Model 1	Fig.&Sol.	Section 1 $(\kappa_1(1/cm), s_1(cm), \phi_1$(rad))	Section 2 $(\kappa_2(1/cm), s_2(cm), \phi_2(rad))$	Section 3 $(\kappa_3(1/cm), s_3(cm), \phi_3(rad))$
Type (1)	Fig. 12 Sol. 1	$[0.0228, 34.0, 0.4]^T$	$[0.0324, 41.6, -1.85]^T$	$[0.0443, 53.5, 0.23]^T$
Type (1)	Fig. 12 Sol. 2	$[0.0228, 37.1, 0.6]^T$	$[0.0284, 38.2, -2.01]^T$	$[0.0441, 53.5, 0.20]^T$
Type (2)	Fig. 12 Sol. 3	$[0.0228, 37.1, 0.6]^T$	$[0.0143, 41.2, 1.57]^T$	$[0.0442, 53.5, -0.37]^T$
Type (2)	Fig. 12 Sol. 4	$[0.0213, 39.1, 0.4]^T$	$[0.0056, 43.7, 2.16]^T$	$[0.0443, 53.5, -0.82]^T$
Type (3)	Fig. 13 Sol. 1	$[0.0228, 39.9, 0.4]^T$	$[0.0085, 41.7, 0.23]^T$	$[0.0444, 53.5, 0.77]^T$
Type (4)	Fig. 13 Sol. 2	$[0.0228, 33.2, 0.2]^T$	$[0.0373, 43.1, -2.00]^T$	$[0.0444, 53.5, -0.18]^T$
Grasping Model 2	Fig. 14	$[0.0228, 34.1, 0.4]^T$	$[0.0324, 41.6, -1.85]^T$	$[0.0324, 53.5, 0.0]^T$

VIII. CONCLUSIONS AND FUTURE WORK

This paper presents a general analysis and method to determine grasping configurations for a given 3D object by a spatial continuum manipulator consisting of three constant-curvature sections and with a fixed base. The curvature, length, and the angle of each section of the manipulator are continuous variables that can be changed. Thus, a configuration of the three-section manipulator is a 9-dimensional vector of those variables. Our approach first defines grasping models for a given object to facilitate grasping by the continuum manipulator. It then uses inter-section constraints to analyze and solve for possible grasping configurations of the robot manipulator. We have completely characterized all possible grasping configurations for a given object grasping model and implemented the approach to determine valid grasping configurations. Our formulation of inter-section constraints and method of finding solutions apply to general continuum manipulators of n constant-curvature sections (where $n \geq 3$).

Our next step includes testing our results on object grasping with the real OctArm manipulator and integrating this approach of determining valid grasping configurations into online motion planning for continuum manipulation.

REFERENCES

[1] G.S. Chirikjian and J. W. Burdick, "Kinematically optimal hyperredundant manipulator configurations," *IEEE Trans. Robot. and Automat.*, 11(6):794-798, Dec. 1995.

[2] R. Cieslak and A. Morecki, "Elephant trunk type elastic manipulator a tool for bulk and liquid type materials transportation," *Robotica*, 17:11-16, 1999.

[3] C. Davidson and A. Blake, "Caging planar objects with a three-finger one-parameter gripper," *IEEE Trans. Robot. and Automation*, pp. 2722-2727, 1998.

[4] M. Ivanescu, M. Florescu, N. Popescu, and D. Pupescu, "Position and force control of the grasping function for a hyperredundant arm," *Proc. IEEE Int. Conf. on Robotics and Automation*, May 2008.

[5] P. Jimnez, F. Thomas, and C. Torras, "3D collision detection: A survey," *Computers and Graphics*, 25: 269-285, 2000.

[6] B.A. Jones and I.D. Walker, "Kinematics for multisection continuum robots," *IEEE Trans. Robot.*, 22(1):43-55, Feb. 2006.

[7] N. Magiddo, "Linear-time algorithms for linear programming in R^3 and related problems," *SIAM Journal on Comp.*, 12(4):759-776, 1983.

[8] S. Makita and Y. Maeda, "3D multifingered caging: Basic formulation and planning," *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Systems*, pp. 2697-2702, 2008.

[9] W. McMahan, B.A. Jones, I.D. Walker, V. Chitrakaran, A. Seshadri, and D. Dawson, "Robotic manipulators inspired by cephalopod limbs," *Proc. CDEN Design Conf.*, Montreal, Canada, pp. 1-10, July 2004.

[10] W. McMahan and I.D. Walker, "Octopus-inspired grasp-synergies for continuum manipulators," *Proc. of the IEEE Int. Conf. on Robotics and Biomimetics*, Bangkok, Thailand, pp. 945-950, Feb. 2009.

[11] W. McMahan, *et al.*, "Field trials and testing of the OctArm continuum manipulator," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Orlando, Florida, pp. 2336-2341, May 2006.

[12] H. Mochiyama and H. Kobayashi, "The shape Jacobian of a manipulator with hyper degrees of freedom," *Proc. IEEE Int. Conf. on Robotics and Automation*, Detroit, May 1999.

[13] K. Moore, W. Rhodes, M. Csencsits, M. Kowka, J. Gomer, and C. Pagano, "Interface evaluation for soft robotic manipulators," *Unmanned Systems Technology VIII: Proceedings of SPIE The International Society of Optical Engineering Defense and Security Symposium*, April 17-21, 2006, Kissimmee, Florida.

[14] S. Neppalli, M.A. Csencsits, B.A. Jones, and I.D. Walker, "A Geometrical approach to inverse kinematics for continuum manipulators," *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, France, Sep. 2008.

[15] S. Neppalli, M.A. Csencsits, B.A. Jones, and I.D. Walker, "Closed-form inverse kinematics for continuum manipulators," *Advanced Robotics*, 23:2077-2091, 2009.

[16] G. Robinson and J.B.C. Davies, "Continuum robots - a state of the art," *ICRA*, pp. 2849-2854, May 1999.

[17] D. Trivedi, C.D. Rahn, W.M. Kier, and I.D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Applied Bionics and Biomechanics*, 5(3):99-117, Sept. 2008.

[18] J. Xiao and R. Vatcha, "Real-time adaptive motion planning for a continuum manipulator," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Oct. 2010.