Inverse Kinematics and Dexterous Workspace Formulation for 2-Segment Continuum Robots with Inextensible Segments

Yifan Wang, Zhonghao Wu, Longfei Wang, Bo Feng and Kai Xu*, Member, IEEE

Abstract—The inverse kinematics (IK) problem of continuum robots has been investigated in depth in the past decades. Under the constant-curvature bending assumption, closed-form IK solution has been obtained for continuum robots with variable segment lengths. Attempting to close the gap towards a complete solution, this paper presents an efficient solution for the IK problem of 2-segment continuum robots with one or two inextensible segments (a.k.a., constant segment lengths). Via representing the robot’s shape as piecewise line segments, the configuration variables are separated from the IK formulation such that solving a one-variable nonlinear equation leads to the solution of the entire IK problem. Furthermore, an in-depth investigation of the boundaries of the dexterous workspace of the end effector caused by the configuration variables limits as well as the angular velocity singularities of the continuum robots was established. This dexterous workspace formulation, which is derived for the first time to the best of the authors’ knowledge, is particularly useful to find the closest orientation to a target pose when the target orientation is out of the dexterous workspace. In the comparative simulation studies between the proposed method and the Jacobian-based IK method involving 500,000 cases, the proposed variable separation method solved 100% of the IK problems with much higher computational efficiency.

Index Terms—Continuum Robots, Inverse kinematics, Dexterous workspace.

I. INTRODUCTION

Continuum robots demonstrate potentials for applications in industrial inspection [1], rescue [2], and healthcare [3] due to their advantages, such as dexterity in confined spaces, inherent compliance, and structural compactness.

Kinematics modeling of multi-segment continuum robots usually adopts a common assumption of constant curvature bending [4]. This approach has been verified analytically and experimentally [5, 6]. It is widely used due to its analytic formulation of the forward kinematics. The inverse kinematics (IK) problem, on the other hand, is not straightforward given the fact that segment bending changes the position and orientation of the end effector at the same time to introduce strong coupling between the end effector’s position and orientation. The existence of a closed-form IK solution depends on the robot’s specific structure and is valid only for the ones with variable-length segments as in [7-9] and the ones with an inverted dual continuum mechanism and inextensible continuum segments as in [10]. When the closed-form IK solution is not available, numerical approaches are often adopted, e.g., the Jacobian-based methods [5, 11-13]. However, the Jacobian-based methods do not always converge to a solution if improper initial guesses of the configuration variables were adopted. Besides, its computational requirement is also relatively high. Hence, an efficient solution to the IK problem of continuum robots with inextensible bending segments is still in great need, since inextensible continuum segments can be much more reliably fabricated in practice.

The challenge of solving the IK problem of a continuum robot comes from the coupling between the position and orientation of its end effector when continuum segments are bent. Due to this strong coupling, all the existing Jacobian-based numerical methods solve all configuration variables simultaneously from the multi-dimensional IK formulation.

In this paper, an efficient Variable Separation IK (VS-IK) method for 2-segment constant-curvature continuum robots with one or two inextensible bending segments is hence presented. By representing the robot shape as piecewise line segments and utilizing the formulation of the line segment geometry, one key configuration variable is separated from the IK formulation. And a one-dimensional nonlinear equation is solved subsequently to obtain one and then all the configuration variable values. The computational efficiency is greatly improved since the to-be-solved equation is only one-dimensional. The conducted simulation studies showed that the VS-IK method significantly outperformed the Jacobian-based method in terms of computation time, number of iterations, and success rate.

Furthermore, the boundaries of the dexterous workspace (i.e., reachable orientations of the end effector at a target position) are categorized into two types. One type of the boundaries caused by the configuration variables limits is analytically formulated, while the other type of the boundaries that occur at singularities of the angular velocity is formulated implicitly. To the best of the authors’ knowledge, this categorization is the first one to analyze the dexterous workspace of a continuum robot. The existing studies evaluate the dexterous workspace of the continuum robots via exhausting numerical methods [14-16]. Based on the presented investigation, handling the target poses with unreachable orientations becomes much easier.
This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

The proposed VS-IK method is shown applicable to continuum robots with one or two inextensible segments, even when the two segments are not directly connected.

This paper is organized as follows. Section II summarizes the kinematics of the 2-segment continuum robot. Next, the geometry-based VS-IK method is elaborated in Section III. In Section IV, the analytic formulation of the dexterous workspace is detailed, and the IK solution that considers the unreachable orientations is described. Simulation study of the VS-IK method compared to the Jacobian-based method is reported in Section V. The conclusion and future works are summarized in Section VI.

II. KINEMATICS NOMENCLATURE AND COORDINATES

The continuum robot under investigation consists of two bending segments, each with 2 bending degrees of freedom (DoFs), a rigid straight base stem, a rigid straight middle stem, and an end effector. A practical embodiment of the continuum robot is as in [17]. The robot driven by an actuation unit can be deployed into a working cavity through a feed channel, and can work in both partially or fully inserted configurations.

When the 1st bending segment is partially inserted into the cavity, this inserted portion is equivalent to a segment with variable length. Hence, the 1st segment possesses 2-DoF bending and 1-DoF length varying, while the 2nd segment possesses 2-DoF bending with a constant length. The actuation unit provides 1-DoF rotation about the neutral axis of the 1st segment. This configuration only has the 2nd segment as an inextensible segment, as shown in Fig. 1(a.1), and is therefore referred to as configuration CI-1.

When the 1st bending segment is fully inserted, a configuration transition occurs such that the base stem now introduces a translation along the axis of the feed channel. Hence, the two bending segment each possesses 2-DoF bending, and the actuation unit provides 1-DoF rotation about the axis of the base stem. This configuration has two inextensible segments, as shown in Fig. 1(a.2), and is referred to as configuration CI-2.

A. Nomenclature and Coordinate

The coordinate attachment for the entire robot are shown in Fig. 1(a), while the coordinate attachment for the \( i \)th segment is shown in Fig. 1(b). The coordinates are defined as follows, and the nomenclature is defined in Table I.

- **Base coordinate \( \{ tb \} \equiv \{ \dot{x}_{tb}, \dot{y}_{tb}, \dot{z}_{tb} \} \) is attached to the center of the base cross section of the \( j \)th segment with \( \dot{z}_{tb} \) perpendicular to the base cross section.**

- **End coordinate \( \{ te \} \equiv \{ \dot{x}_{te}, \dot{y}_{te}, \dot{z}_{te} \} \) is attached to the center of the end cross section of the \( i \)th segment. \( \dot{z}_{te} \) is perpendicular to the end cross section, and \( \dot{x}_{te} \) points to the same segment surface region as \( \dot{x}_{tb} \) such that the segment does not undergo twisting.**

- **Bending plane coordinate \( \{ t1 \} \equiv \{ \dot{x}_{t1}, \dot{y}_{t1}, \dot{z}_{t1} \} \) is defined such that the \( i \)th continuum segment bends in its XY plane, and its origin coincides with that of \( \{ te \} \). The XY plane of \( \{ t1 \} \) is referred to as the bending plane.**

- **Bending plane coordinate \( \{ t2 \} \equiv \{ \dot{x}_{t2}, \dot{y}_{t2}, \dot{z}_{t2} \} \) is obtained from \( \{ t1 \} \) via a rotation about \( \dot{z}_{t1} \) for an angle \( \theta_t \) so that \( \dot{z}_{t2} \) is aligned with \( \dot{z}_{te} \).**

- **End effector coordinate \( \{ g \} \equiv \{ \dot{x}_g, \dot{y}_g, \dot{z}_g \} \) is attached to the tip of the end effector, obtained by moving \( \{ 2e \} \) for the length \( L_g \) along \( \dot{z}_{2e} \).**

- **World coordinate \( \{ w \} \equiv \{ \dot{x}_w, \dot{y}_w, \dot{z}_w \} \) is fixed to the feed channel with \( \dot{z}_w \) aligned with the channel axis. It should be noticed that \( \{ 1b \} \) is obtained from \( \{ w \} \) via a rotation about \( \dot{z}_w \) for an angle \( \phi \).**

According to the nomenclature, the configuration variables for CI-1 are \( \phi, \theta_1, L_1, \delta_1, \theta_2, \delta_2 \), while the configuration variables for CI-2 are \( L_s, \phi, \theta_1, \delta_1, \theta_2, \delta_2 \).

B. Forward Kinematics

The position and orientation of \( \{ t \} \) for the \( i \)th segment are given by (1) and (2).

\[
\begin{align*}
\delta \mathbf{p}_w &= \frac{L}{\theta_1} \left[ \cos \delta_1 (1 - \cos \theta_1) \quad \sin \delta_1 (1 - \cos \theta_1) \quad \sin \theta_1 \right]^T, \quad (1) \\
\delta \mathbf{R}_w &= \text{Rot}(\dot{z}_w - \dot{z}_g) \text{Rot}(\dot{y}_g, \theta_g) \text{Rot}(\dot{z}_g). \quad (2)
\end{align*}
\]

where \( \delta \mathbf{p}_w = [0 \ 0 \ L_1]^T \) when \( \theta_1 = 0 \), and \( \text{Rot}(\mathbf{m}, \alpha) \) represents the rotation matrix about the axis \( \mathbf{m} \) by an angle \( \alpha \).
The complete forward kinematics of the continuum robot can be referred to [17].

III. VARIABLE SEPARATION INVERSE KINEMATICS

The VS-IK method for the configurations CI-2 and CI-1 are discussed in Section III.A and Section III.B, respectively.

A. Inverse Kinematics for the Configuration CI-2

To separate the configuration variables, the robot is characterized using piecewise line segments, as shown in Fig. 2 (a). A constant-curvature continuum segment is represented by two line segments that are tangent to the arc at its two ends, as shown in Fig. 2 (b). Then, the shape of the entire continuum robot is represented by three consecutive line segments. The intersections of the line segments are $p_1$ and $p_2$, respectively.

The line segment length of the $l_i$ segment is given by (3).

$$l_i = L_i \tan(\theta_i / 2)/\theta_i$$

where $l_i = L_i/2$ when $\theta_i = 0$. Note that $l_i$ approaches infinity for $\theta_i = \pi$. In the following content, the maximum bending angle $\theta_i$ is assumed to be less than $\pi$, which is practically adopted for most continuum robot designs.

The pose of the end effector is represented by (4).

$$^w T = \begin{bmatrix} ^w R & ^w p \end{bmatrix} = \begin{bmatrix} n & s & a & ^w p \end{bmatrix}$$

where $^w p = [p_x, p_y, p_z]^T$ denotes the position of the end effector, while $n, s,$ and $a$ are unit vectors.

In the line segment representation, the position of $p_1$ is expressed from the origin of $[w]$, while the position of $p_2$ is expressed backward from $^w p_1$ as follows.

$$p_1 = \begin{bmatrix} 0 & 0 & l_i + L_i \end{bmatrix}^T$$

$$p_2 = ^w p - L_{2x}a = \begin{bmatrix} p_x - L_{2x}a_x & p_y - L_{2x}a_y & p_z - L_{2x}a_z \end{bmatrix}^T$$

where $L_{2x} = l_2 + L_y$.

And it follows that

$$\|p_2 - p_1\| = L_{11/2}$$

where $L_{11/2} = l_1 + L_2 + l_2$.

Then, $\theta_i$ is expressed using the inner products of the vectors available in the line segment representation as follows.

$$\cos \theta_i = (p_2 - p_1)^T \hat{z}_i / L_{11/2} = (p_z - p_{z1}) / L_{11/2}$$

$$\cos \theta_2 = (p_2 - p_1)^T a / L_{11/2}$$

Expanding (8) gives an expression of $L_2$ with respect to $l_1$, $l_2$ and $\theta_2$, as in (10). Substituting (10) into (7) yields (11).

$$L_2 = p_z - L_{2x}a_z - L_{11/2} \cos \theta_i - l_1$$

$$b x = p_i + p_j$$

$$b y = p_i + p_j$$

Substituting (10) into (9) gives an expression of $\cos \theta_2$ with respect to $l_1$, $l_2$ and $\theta_i$, as in (12). Then, substituting $\theta_2$ and $\tan(\theta_2/2)$ in (3) yields (13).

$$\cos \theta_2 = (p_i^T a + L_{2x}(a_x^2 - 1) - a_x(p_x - L_{11/2} \cos \theta_i)) / L_{11/2}$$

$$l_2 \cos \theta_2 = (b_x - a_x \cos \theta_i) / L_{11/2}$$

where $b_x = b_z + (a_x^2 - 1)l_2$.

Since $l_1$ and $l_2$ are functions of $\theta_i$, as shown in (3) and (11), equation (13) only contains the configuration variable $\theta_i$, and can be solved using general nonlinear equation solving methods. Please note that equation (11) gives two possible solutions of $l_2$. While solving (13), one solution of $l_2$ from (11) is firstly used; if no solution of $\theta_i$ is found, the other solution of $l_2$ is used to continue the iteration.

After solving $\theta_i$, $l_1$ and $l_2$ are obtained from (3) and (11) respectively. Then, $L_2$ is obtained from (7) or (8), and $\theta_2$ is obtained from (9). The remaining configuration variables $\delta_1, \delta_2$ and $\varphi$ are solved as follows.

Using (5) and (6), $p_1$ and $p_2$ are obtained. Points $p_1$, $p_2$ and $p_3$ lie in the bending plane of the $1^{st}$ segment, while $p_1$, $p_2$, and $p_3$ lie in the bending plane of the $2^{nd}$ segment.

The bending direction of the $1^{st}$ segment is calculated by:

$$\varphi = \arctan \left( \frac{2(p_2 \times p_3^T) p_4}{\|p_2\| \|p_3\| \|p_4\|} \right)$$

The relative bending direction between the $1^{st}$ and the $2^{nd}$ segments is calculated by:

$$\delta_1 - \delta_2 = \text{sgn} \left( \frac{p_1 \times p_2^T p_3}{\|p_1\| \|p_2\| \|p_3\|} \right) \left( \frac{\arccos \left( \frac{z_2 \times p_2^T (p_2 - p_1) \times a}{\|p_2\| \|p_3\|} \right)}{L_{11/2}} \right)$$

where the sgn function indicates the direction from $\delta_1$ to $\delta_2$. The sgn function is zero when $(p_1 \times p_2)^T p_3 = 0$, indicating that both segments are bending to the same direction and $\delta_1 = \delta_2$.

The forward kinematics for the end effector orientation is given as follows:
This work has been submitted to the IEEE for possible publication. May be transferred without notice, after which this version may no longer be accessible.

Substituting (2) into (16) yields (17).

\[
\text{Rot}(\mathbf{\hat{z}}, \varphi)_{\text{ub}} R_{w_{\text{ub}}} = w R_{g_{\text{ub}}}.
\]  

A. Dexteroes Workspace of the Configuration CI-1

The rotation of the end effector about its axis \( \mathbf{\hat{z}}_s \) can be independently generated by changing \( \varphi \) without affecting \( \mathbf{\hat{z}}_s \).

Therefore, the dexterous workspace only concerns about \( \mathbf{\hat{z}}_s \) (the pointing direction), which is characterized by a unit spherical surface centered at the target point.

Since the configuration variables \( \delta_i \) and \( \delta_j \) are not limited, there exist two feasible configurations with mirrored shapes about a symmetry plane for a given end effector position, as illustrated in Fig. 3(a.1). The symmetry plane is defined by the target end effector position \( \mathbf{\hat{p}}_s \) and the \( \mathbf{\hat{z}}_s \) axis of the world coordinates. The \( \mathbf{\hat{z}}_s \) axis of the two mirrored poses are therefore symmetric about the plane, indicating the symmetry of the dexterous workspace with respect to the plane. Hence, it is sufficient to investigate the dexterous workspace by projecting half of the unit spherical surface onto the symmetry plane to form a concentric unit circular area. And the dexterous workspace can be described in the unit circular area, as shown in Fig. 3(a.2 and a.3). A symmetry plane coordinate \( \{s\} \) is then defined as in Fig. 3, rotating \( \{w\} \) around \( \mathbf{\hat{z}}_s \) such that the XZ-plane of \( \{s\} \) coincides with the symmetry plane. Please note that all equations derived in Section II still hold while expressed in \( \{s\} \), with the target position \( \mathbf{\hat{p}}_s \) and the \( \mathbf{\hat{z}}_s \) axis of the target orientation \( \mathbf{\hat{a}} \) represented by \( \mathbf{\hat{p}}_s = [p_x, p_y, p_z]^T \) and \( \mathbf{\hat{a}} = [a_x, a_y, a_z]^T \). They are obtained as follows:

\[
\mathbf{\hat{p}}_s = \text{Rot}(\mathbf{\hat{z}}, -\gamma)\mathbf{\hat{p}}_s, \quad \mathbf{\hat{a}} = \text{Rot}(\mathbf{\hat{z}}, -\gamma)\mathbf{\hat{a}}.
\]  

where \( \gamma = \arctan(p_y/p_x) \), and \( p_y = 0 \), referring to Fig. 3(a.1).

Each boundary of the dexterous workspace is projected as a curve or a line segment on the unit circular area. As shown by later derivations, using this projected unit circular area facilitates the representation of the dexterous workspace from the workspace boundaries.

In the configuration CI-1, there are four type-I boundaries obtained when #1) \( \theta_1 = \theta_{s,2} \), #2) \( \theta_1 = \theta_{s,3} \), #3) \( L_1 = (r_{1s})/\theta_1 \), and #4) \( L_1 = L_{s,1} \). Note that although \( \theta_1 \) and \( \theta_2 \) have a lower limit 0 by definition, this limit does not constrain the dexterity of the end effector. This is because that when the \( i \)-th segment is straight (namely \( \theta_1 = 0 \)), it can bend and change the end effector orientation to all directions. Since \( \theta_2 \) is the only configuration variable in (19), the boundary #1 corresponds to (19) when \( \theta_1 = \theta_2 \). It is noted that \( a_{s_{\theta}} \) is multiplied by \( p_{s_y} \) in (19) and \( p_{s_y} = 0 \) from (20). Hence, \( a_{s_{\theta}} \) disappears from (19). Rearranging (19) in terms of \( a_{s_{\theta}} \) and \( a_{s_{\omega}} \) gives (21). The boundary #1 is hence a straight line on the symmetry plane, corresponding to a circle on the unit spherical surface, as \( \theta_2 \) set to \( \theta_{s,2} \) in (21).

\[
A_1(\theta_{s,2})a_{s_{\omega}} + B_1(\theta_{s,2})a_{s_{\theta}} + C_1(\theta_{s,2}) = 0.
\]  

where \( A_1(\theta_{s,2}) \equiv -2p_{s_{\omega}}(L_{s_{2c}} + L_{s_2} \cos \theta_{s,2}) \), \( B_1(\theta_{s,2}) \equiv -L_{s_{2c}}^2 + p_{s_{\omega}}^2 - 2L_{s_{2c}}L_{s_2} \cos \theta_{s,2} \), \( C_1(\theta_{s,2}) \equiv 2L_{s_{2c}}^2 - L_{s_{2c}}^2 + (L_{s_{2c}}^2 + p_{s_{\omega}}^2 + L_{s_{2c}}^2) \cos \theta_{s,2} \), and \( L_{s_{2c}} = l_2 + L_2 + p_{s_2} \).

The boundary #2 is obtained by first substituting \( l_1 \) in (8) with (18), \( a_{s_{\theta}} \) disappears from (18) due to its multiplication with \( p_{s_y} = 0 \). Rearranging (8) in terms of \( a_{s_{\omega}} \) and \( a_{s_{\omega}} \) gives:
This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

\[ A_2(\theta_2) a^{\text{a}}_s + B_2(\theta_1, \theta_2) a^{\text{a}}_c + C_2(\theta_1, \theta_2) a^{\text{a}}_s + D_2(\theta_1, \theta_2) = 0, \quad (22) \]

where

\[ A_2(\theta_2) \triangleq -2L_{2s}^2, \quad B_2(\theta_1, \theta_2) \triangleq -2L_{2c}^2 (\cos \theta_1 - 1), \]

\[ C_2(\theta_1, \theta_2) \triangleq -2L_{2c}^2 \alpha x_1^2 (1 + \cos \theta_1), \quad \text{and} \quad D_2(\theta_1, \theta_2) \triangleq L_{2c}^2 - L_{2c}^2 \]

\[ + p_{a}^2 + (L_{2s}^2 + L_{2c}^2 + p_{a}^2) \cos \theta_1. \]

Equation (22) represents a family of curves controlled by \( \theta_2 \). Since the end effector orientation \( \hat{a} \) also satisfies (21), the boundary \#2 is the trajectory of the intersection of the line (21) and the curve (22). Substituting \( a^{\text{a}}_a \) in (22) with (21) gives (23). Given \( \theta_2, \alpha_x \) and \( \alpha_c \) can be solved from (23) and (21) respectively. Therefore, the boundary \#2 is a parametric curve driven by \( \theta_2 \).

\[ A_2(\theta_2) a^{\text{a}}_s + (B_2(\theta_1, \theta_2) - C_2(\theta_1, \theta_2) B_1(\theta_2)/A_1(\theta_2)) a^{\text{a}}_c \]

\[ + (D_2(\theta_1, \theta_2) - C_2(\theta_1, \theta_2) C_1(\theta_2)/A_1(\theta_2)) = 0. \quad (23) \]

Next, substituting (9) into (7) yields (24).

\[ l = -L_{2s}^2 - L_{1s}^2 - 2L_{2s}^2 (L_{1s}^2 + L_{1s}^2) \cos \theta_1 + p_{a}^2 + p_{a}^2 \]

\[ 2(L_{2s}^2 \cos \theta_1 + L_{2c}^2) . \quad (24) \]

For the boundaries \#3 and \#4, \( \theta_1 \) is given by (25) and (26), substituting \( L_1 \) in (3) with its lower limit \( (r_1 - \theta_1) \), and higher limit \( L_{1s} \), respectively.

\[ \theta_1 = 2 \arctan(l_1/r_1) , \quad (25) \]

\[ \theta_1 / \tan(\theta_1/2) = L_{1s} / l_1 . \quad (26) \]

Rearranging (8) in terms of \( a_{s} \) and \( a_{s} \) gives (27).

\[ A_3(l_1, \theta_1) a_{s} + B_3(l_1, \theta_1) a_{c} + C_3(l_1, \theta_1) = 0, \quad (27) \]

where

\[ A_3(l_1, \theta_1) \triangleq 2p_{a}^2 (L_{s}^2 + L_{1s}^2 \cos \theta_1), \quad B_3(l_1, \theta_1) \triangleq p_{a}^2 \]

\[ -L_{1s}^2 - L_{1s}^2 - 2L_{1s}^2 L_{1s} \cos \theta_1, \quad L_{1s} = l_1 - p_{a}^2, \quad L_{1s} = L_{1s} - L_{s} \]

and \( C_3(l_1, \theta_1) \triangleq 2L_{1s} \cos \theta_1 + (p_{a}^2 + L_{1s}^2) \cos \theta_1. \)

Since \( l_1 \) and \( \theta_1 \) are expressed as functions of \( \theta_1 \) using (24), (25) and (26), the straight line (27) is in fact controlled by \( \theta_1 \). The boundaries \#3 and \#4 are trajectories of the intersection point of (21) and (27), as parametric curves driven by \( \theta_2 \).

Please note that \( \theta_1 \) is solved from (26) via a numerical process.

A particular case of \( L_1 = (r_1 - \theta_1) \) is that \( \theta_1 = 0 \). Then \( L_1 = 0 \) corresponds to a configuration that has one bending segment, and there exists only one IK solution for a given end effector position. Therefore, \( L_1 = 0 \) is represented by a dot in the unit circular area rather than a curve.

As for the type-II boundary, an enveloping curve is expected as none of the configuration variables are at their limits. It is noted that (21) gives a set of end effector directions expressed in \( \{s\} \) when \( \theta_2 \) is set to a specific value. Therefore, the set of end effector directions is swept by (21) from \( \theta_2 = 0 \) to \( \theta_2 = \pi \). The type-II boundary is thus the envelope of the family of straight line (21). This boundary is obtained by analytically solving (28), referring to the singularity theory in [18].

\[ A_1 a_s + B_1 a_c + C_1 = 0, \quad \frac{DA_1}{d\theta_2} a_s + \frac{DB_1}{d\theta_2} a_c + \frac{DC_1}{d\theta_2} = 0. \quad (28) \]

Fig. 3 Dextorous workspace of the continuum robot. The symmetry plane, coordinate \{s\}, and the unit spherical surfaces with the continuum robot in CI-I and CI-2 are shown in (a.1) and (b.1), respectively. The projections of the unit spherical surfaces on the symmetry plane are shown in (a.2) & (a.3) for the configuration CI-1, and (b.2) & (b.3) for the configuration CI-2. The dextorous workspace is represented by the yellow areas.

To determine which side of the boundary curve corresponds to the feasible range of the configuration variable, the configuration limits are perturbed towards the feasible range and substituted into the type-I boundary curves to generate points on the feasible side. The area characterized by the feasible sides of all type-I boundaries and the type-II boundary is the dextorous workspace. The dextorous workspace and its boundaries for the configuration CI-1 are shown in Fig. 3(a.2 and a.3).

B. Dextorous Workspace of the Configuration CI-2

In the configuration CI-2, there are four type-I boundaries: \#1) \( \theta_1 = \theta_{1s} \), \#2) \( \theta_2 = \theta_{2s} \), \#3) \( L_1 = 0 \), and \#4) \( L_2 = L_{2s} \).

For the boundaries \#1 and \#2, equation (7) is substituted into (8) and (9) to eliminate \( L_{1s} \). \( a_{s} \) disappears from (8) and (9) due to its multiplication with \( p_{a}^2 = 0 \). Rearranging (8) and (9) in terms of \( a_{s} \) and \( a_{c} \) yields (29) and (30):
Therefore, the point closest to the target direction lies on the type-I boundaries. Such closest points are found by discretizing the type-I boundaries as specifying sequences of $\theta_1$ and $\theta_2$, and analytically calculating $a_{x1}$ and $a_{x2}$. If the VS-IK method gives a non-real solution or fails to converge, the target direction is infeasible since it violates the type-II boundary of the continuum robot. The closest feasible direction then lies on the type-II boundary. The type-II boundary can be solved from (38) for CI-1 and (39) for CI-2.

The proposed formulation has several advantages over the existing numerical methods. The existing methods only give an approximation to the dexterous workspace by discretizing the end-effector orientation into patches on the unit spherical surface. Also, these methods identify the dexterous workspace either by solving the IK for every patch with the Jacobian-based method [14] or by generating feasible end-effector poses with the Monte-Carlo method [15, 16], both of which are time-consuming. The proposed formulation, in contrast, gives exact boundaries that can be efficiently calculated.

V. SIMULATION STUDY

Numerical simulation was conducted to evaluate the performance of the proposed VS-IK method. The Jacobian-based damped least squares method [19] (the Jacobian-DLS method) was used for comparison.

The IK methods were considered to be solved when the position error and the orientation error of the end effector was less than 0.01 mm and 0.01 rad respectively, without exceeding the limits of the configuration variables and the number of iterations. The orientation error was calculated as the rotation angle between the target orientation and the end effector orientation.

The VS-IK method employed the Newton-Raphson method to solve the nonlinear equations (13) and (19). The iteration residuals of (13) and (19) were empirically adjusted to 0.01 and 0.0003, respectively, to match the designated pose error tolerances. The maximum iterations per target pose was set to 200 for both methods. The initial configurations for the Jacobian-DLS method were set to $L_1 = L_1/2$ for CI-1 and $L_2 = L_2/2$ for CI-2, with all other configuration variables equal to zero. For the VS-IK method, the initial values were set to $\theta_1 = \theta_2 = 0$ and $\theta_1 = \arccos(\alpha_1)$ for CI-2. Both methods were implemented in MATLAB (Mathworks Inc.) and ran on a laptop with a 2.3 GHz Intel Core i5-8300H CPU. The structural parameters of the continuum robot used in the simulation are given in Table II.

A. Case Study for Reachable Targets

This case study was conducted for 250,000 test cases (sets of initial and target poses for verifying the effectiveness of the IK approach) from a previous work [17] for comparison (namely, 125,000 cases for Configuration #3 in [17] as configuration CI-1 in this study, as well as 125,000 cases for Configuration #4 in [17] as configuration CI-2 in this study). The target poses were generated by assigning random values to the configuration variables within the feasible ranges and calculating forward kinematics, and hence are reachable.

The performance of the two IK methods is shown in Table III. The VS-IK method used substantially less computation...
This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

time and the average time for each iteration, compared to the Jacobian-DLS method. For the configurations CI-1 and CI-2, the total computation time of the VS-IK method for 125,000 cases was 2.86 s and 7.13 s, respectively, corresponding to 99.43% and 96.72% time reduction compared to the Jacobian-DLS method, respectively. In terms of the time per iteration, the VS-IK method is more than 10 times faster than the Jacobian-based method, referring to Table III, facilitating real-time trajectory tracking tasks. The major computational load of the VS-IK method only comes from computing the values and gradients of one-variable equations (13) and (19).

The Jacobian-DLS method calculates the forward kinematics and the pseudo-inverse of the Jacobian matrix at every iteration.

The VS-IK method achieved 100% success rate. The total success rate of the Jacobian-DLS method in this study is higher than that in our previous work (97.79%) because the configuration transition is not considered in this study. Nevertheless, the convergence of the Jacobian-DLS method relied on the proper setting of the initial configuration of the robot and the end effector twist (i.e., the linear and angular velocities). In contrast, all targets in the conducted simulation were successfully converged from a single initial value setting using the VS-IK method, without tuning extra parameters.

B. Case Study for Non-Guaranteed Reachable Targets

This case study was conducted on test cases generated by randomly sampling positions within the translational workspace that is generated according to [14], and generating uniformly distributed random Euler angles to obtain random orientations in SO(3). Therefore, all target positions were reachable within the configuration limits, but the corresponding orientations were not guaranteed to be reachable. This case study emulates a realistic teleoperation scenario where the target poses might not be completely reachable. Two test sets were generated for CI-1 and CI-2, respectively, each containing 125,000 poses.

The dimension reduced Jacobian formulation [20] was used as this Jacobian-DLS method with position reaching as the primary task [17] to converge to target poses that may be out of the dexterous workspace. Since the orientation might not converge successfully, the iteration was set to terminate if the orientation error was not smaller than the previous smallest error for 30 consecutive iterations after converging to the target position. For the VS-IK method, the iteration terminated if the configuration variables $\theta_1$ and $\theta_2$ exceeded $\pi$ for 3 consecutive iterations. If the target orientation was out of the dexterous workspace, 600 equally incremental values of $\theta_1$ and $\theta_2$ within their limits were used to generate the boundaries.

Table IV shows the performance of both IK methods. Similar to the simulation in Section V-A, the VS-IK method exhibited improved computational efficiency and 100% success rate, achieving smaller orientation error compared to the Jacobian-DLS method.

The success rate of the Jacobian-DLS method was significantly lower than that of the VS-IK method. Representative failed cases are shown in Fig. 4 (a), where the Jacobian-DLS method was stuck at the configuration variables limits while trying to reach the lower part of the translational workspace. In the CI-1 case, the Jacobian-DLS method first retracted the 1st segment to reduce the position error, as in Fig. 4(d). In the CI-2 case, the Jacobian-DLS method first retracted the base segment to reduce the position error, as in Fig. 4(e). To reach the target position, the robot had to extend and bend the 1st segment in the CI-1 case and bend the 1st segment in the CI-2 case. However, doing so would temporarily increase the position error in both cases, which is prohibited since the Jacobian-DLS method is gradient-based in the task space.

VI. Conclusions

In this paper, an efficient IK method for continuum robots with one or two inextensible bending segments is presented. By using the line segment representation to separate the configuration variables, the IK problem is formulated as solving a one-dimensional nonlinear equation for $\theta_1$ or $\theta_2$, and then solving other configuration variables in closed-form. By incorporating the configuration limits in the formulation of the VS-IK, the boundaries of the dexterous workspaces are formulated as parametric curves. A comparative simulation study was performed for the VS-IK method and the Jacobian-DLS method on a total of 500,000 test cases. The results showed that the VS-IK method achieved more than 96% computation time reduction and 100% success rate. The efficiency of the VS-IK method would facilitate real-time motion planning and control. The analytic formulation of the

| Table II. Structural Parameters |
|--------------------------------|
| $L_{10}$ | $L_{20}$ | $L_{1}$ | $L_{2}$ | $L_{3}$ | $\theta_{10}$ | $\theta_{20}$ | $r_{1}$ |
| 40 mm | 60 mm | 20 mm | 20 mm | 150 mm | $\pi/2$ rad | 2$\pi/3$ rad | 80/1$\pi$ mm |

| Table III. Performance Comparison for Reachable Targets |
|---------------------------------------------------------|
| Performance Index | Jacobian-DLS | VS-IK |
|-------------------|--------------|-------|
| CI-1              | CI-1         | CI-2  |
| Computation time (s) | 464.18  | 217.40 | 2.86 | 7.13 |
| Avg. iterations   | 26.73  | 14.59 | 3.86 | 7.05 |
| Avg. time per iteration (10^3 s) | 13.89 | 11.92 | 0.59 | 0.81 |
| Success rate      | 97.58% | 99.43% | 100% | 100% |
| Failure count     | 3025  | 715   | 0    | 0    |

| Table IV. Performance Comparison for Non-Guaranteed Reachable Targets |
|---------------------------------------------------------|
| Performance Index | Jacobian-DLS | VS-IK |
|-------------------|--------------|-------|
| CI-1              | CI-1         | CI-2  |
| Execution time (s) | 2218.02 | 2030.56 | 18.34 | 60.43 |
| Avg. iterations   | 72.48  | 84.07 | 3.47 | 14.75 |
| Success rate      | 64.89% | 90.19% | 100% | 100% |
| Failure count     | 43889 | 12268 | 0    | 0    |
| Avg. position error (mm) | 11.73 | 4.92 | 0.00073 | 0.0011 |
| Avg. orientation error (rad) | 1.1293 | 0.6128 | 0.9521 | 0.5980 |
Fig. 4 Representative failed cases of the Jacobian-DLS method and the solutions given by the VS-IK method are shown in (a). Target orientations, orientations given by the VS-IK method, and the dexterous workspace at the target positions are shown in (b) for the CI-1 case and (c) for the CI-2 case.\r
\r
The trajectories of the configuration variables of the Jacobian-DLS method are shown in (d) for the CI-1 case and (e) for the CI-2 case.
\r
dexterous workspaces can also benefit the design and kinematics performance evaluation of continuum robots.
\r
Extending the VS-IK method to continuum robots with more than two segments will be attempted in the future work.

References

[1] S. Liu, Z. Yang, Z. Zhu, L. Han, X. Zhu, and K. Xu, "Development of a Dexterous Continuum Manipulator for Exploration and Inspection in Confined Spaces," Industrial Robot: An International Journal, vol. 43, No.3, pp. 284-295, 2016.

[2] W. McMahan, V. Chittrakaran, M. Csenecits, D. M. Dawson, I. D. Walker, B. A. Jones, M. Pritts, D. Dienno, M. Grissom, and C. D. Rahn, "Field Trials and Testing of the OctArm Continuum Manipulator," in IEEE International Conference on Advanced Robotics (ICAR), Orlando, FL, USA, 2006, pp. 2336-2341.

[3] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum Robots for Medical Applications: A Survey," IEEE Transactions on Robotics, vol. 31, No.6, pp. 1261-1280, Dec 2015.

[4] R. J. Webster and B. A. Jones, "Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review," International Journal of Robotics Research, vol. 29, No.13, pp. 1661-1683, Nov 2010.

[5] B. A. Jones and I. D. Walker, "Kinematics for Multisection Continuum Robots," IEEE Transactions on Robotics and Automation, vol. 22, No.1, pp. 43-55, Feb 2006.

[6] K. Xu and N. Simaan, "Analytic Formulation for the Kinematics, Statics and Shape Resolution of Multibackbone Continuum Robots via Elliptic Integrals," Journal of Mechanisms and Robotics, vol. 2, No.011006, pp. 1-13, Feb 2010.

[7] P. Sears and P. Dupont, "A Steerable Needle Technology Using Curved Conical Tubes," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Beijing, China, 2006, pp. 2850-2856.

[8] S. Neppalli, M. A. Csenecits, B. A. Jones, and I. D. Walker, "Closed-Form Inverse Kinematics for Continuum Manipulators," Advanced Robotics, vol. 23, No.15, pp. 2077-2091, 2009.

[9] A. Garriga-Casanovas and F. Rodriguez y Baena, "Kinematics of Continuum Robots With Constant Curvature Bending and Extension Capabilities," ASME Journal of Mechanisms and Robotics, vol. 11, No.1, p. 011010, Feb 2019.

[10] B. Zhao, L. Zeng, B. Wu, and K. Xu, "A Continuum Manipulator with Closed-form Inverse Kinematics and Independently Tunable Stiffness," in IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 1847-1853.

[11] P. Sears and P. E. Dupont, "Inverse Kinematics of Concentric Tube Steerable Needles," in IEEE International Conference on Robotics and Automation (ICRA), Rome, Italy, 2007, pp. 1887-1892.

[12] L. S. Godage, E. Guglielmino, D. T. Branson, G. A. Medrano-Cerda, and D. G. Caldwell, "Novel Modal Approach for Kinematics of Multisection Continuum Arms," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) San Francisco, CA, USA, 2011, pp. 1093-1098.

[13] T. Mahl, A. Hildebrandt, and O. Sawodny, "A Variable Curvature Continuum Kinematics for Kinematic Control of the Bionic Handling Assistant," IEEE Transactions on Robotics, vol. 30, No.4, pp. 935-949, Aug 2014.

[14] K. Xu, J. Zhao, and X. Zheng, "Configuration Comparison among Kinematically Optimized Continuum Manipulators for Robotic Surgeries through a Single Access Port," Robotica, vol. 33, No.10, pp. 2025-2044, Dec 2015.

[15] L. Wu, R. Crawford, and J. Roberts, "Dexterity Analysis of Three 6-DoF Continuum Robots Combining Concentric Tube Mechanisms and Cable-Driven Mechanisms," IEEE Robotics and Automation Letters, vol. 2, No.2, pp. 514-521, 2016.

[16] C. Wang, S. Geng, D. T. Branson, C. Yang, J. S. Dai, and R. Kang, "Task-Space-Based Orientability Analysis and Optimization of a Wire-Driven Continuum Robot," Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, vol. 233, No.23-24, pp. 7638-7668, 2019.

[17] S. a. Zhang, Q. Li, H. Yang, J. Zhao, and K. Xu, "Configuration Transition Control of a Continuum Surgical Manipulator for Improved Kinematic Performance," IEEE Robotics and Automation Letters, vol. 4, No.4, pp. 3750-3757, Oct 2019.

[18] J. W. Bruce and P. J. Giblin, Curves and Singularities: a Geometrical Introduction to Singularity Theory: Cambridge University Press, 1992.

[19] S. Chiaverini, "Singularity-Robust Task-Priority Redundancy Resolution for Real-Time Kinematic Control of Robot Manipulators," IEEE Transactions on Robotics and Automation, vol. 13, No.3, pp. 398 - 410, June 1997.

[20] Z. Wu, H. Yang, X. Liu, and K. Xu, "Dimension Reduced Instantaneous Inverse Kinematics for Configuration Variable Limits of Continuum Manipulators," in IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, Yunnan, China, 2019, pp. 303-308.