

FABRIKc: an Efficient Iterative Inverse Kinematics Solver for Continuum Robots

Weihao Zhang, Zhixiong Yang, Tianlai Dong and Kai Xu*, *Member, IEEE*

Abstract—Continuum robots and manipulators attracted lots of attention in the past decade owing to their dexterity, intrinsic compliance and design compactness. A widely accepted approach in formulating the kinematics of a multi-segment continuum robot is to assume their shapes as serially connected arcs with constant curvature at different values for each segment. In spite of the simplification in kinematics, complete analytic solutions of the inverse kinematics (IK) problem of such a multi-segment continuum robot may not exist. Instead, a generalized inverse Jacobian method is often used for the IK problem. This Jacobian-based method is computationally demanding and sometimes fails to solve the IK problem. This paper proposes a heuristic approach to iteratively solve the IK problem of a continuum robot. The algorithm implementation, which is straightforward, is elaborated. Several simulation case studies show that the algorithm is highly effective in computing the IK solutions for continuum robots with different topologies, indicating the effectiveness of this algorithm.

I. INTRODUCTION

CONTINUUM robots, a term proposed in [1], attracted lots of attention in the past decade because of their advantages in terms of dexterity, intrinsic compliance and design compactness. Benefited by their useful features, continuum robots have been broadly employed for a variety of applications in confined spaces [2, 3].

Kinematics of continuum robots is often formulated by characterizing their shapes as smooth curves. A common approach is to assume the shapes as serially connected arcs with constant curvature at different values for each segment, while formulating the kinematics of a multi-segment continuum robot [4]. This approach, which is widely accepted, was verified analytically and experimentally [5-7].

Under the constant curvature assumption, each segment is usually characterized by three configuration variables (e.g., two for bending angles and one for segment length). The use of configuration space to bridge the Cartesian task space and the actuation space generalizes the kinematics model and makes the model independent of particular actuation schemes (e.g., the uses of cables, elastic rods, shape memory alloys,

hydraulic or pneumatic pressures, etc.).

When a few segments are serially connected with each other or with rigid-linked kinematic chains, the forward kinematics formulation is straightforward under this constant curvature assumption. This has enabled plenty of applications of continuum robots in healthcare and industries (e.g., the ones in [8-16]).

More sophisticated approaches to formulate the kinematics of continuum robots have also been attempted, for example, via the use of Hamilton's principle [17], elliptic integrals [7], Cosserat rod theory [18], virtual power [19], etc. However, applications of these methods are limited for serial continuum robots due to the difficulties in formulations and the challenges in real-time computation.

Just like a generic serial manipulator whose inverse kinematics (IK) problem usually does not have closed-form solutions (unless specifically designed), the IK problem of a continuum robot only has analytical solutions when the robots have particular forms (e.g., a variable curvature arc length followed by a fixed curvature arc length in [20]) or have unconstrained structural parameters (e.g., arbitrary segment lengths while neglecting the orientation of the robot's end effector in [21]).

In most cases, the IK problems of continuum robots are solved numerically, employing a generalized inverse Jacobian method (e.g., in [14, 22-25]). This generalized inverse Jacobian method, which is similar to the resolved motion rate control in [26], can also be constructed to provide singularity robustness and kinematic redundancy resolutions (e.g., enable self-motions, avoid joint limits, etc.).

The generalized inverse Jacobian method for the IK problem is computationally demanding. What's more, it takes even longer for the generalized inverse Jacobian method to converge to a solution when the target pose is singular: the singularity-robust formulation often adds small non-zero errors to maintain numerical stability but delays the convergence.

Sometimes, this inverse Jacobian method fails to find an IK solution that does exist, when the pose to be solved is very different from the robot's current pose. This may be due to the fact that the inverse Jacobian leads the iterations towards a local minimum, instead of finding the IK solution. Changing initial guesses for multiple times is often needed to find the IK solution.

Recently, a new heuristic iterative approach, the Forward And Backward Reaching Inverse Kinematics (FABRIK) method, which is based on conformal geometric algebra, has been proposed [27]. The FABRIK method finds the joints'

Manuscript received on Mar 15th, 2018. This work was supported in part by the National Natural Science Foundation of China (Grant No. 51435010, Grant No. 51722507 and Grant No. 91648103), and in part by the National Key R&D Program of China (Grant No. 2017YFC0110800).

Weihao Zhang and Kai Xu are with School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China (asterisk indicates the corresponding author, phone: +86-21-34206547; e-mails: zwh0813@sjtu.edu.cn and k.xu@sjtu.edu.cn).

Zhixiong Yang and Tianlai Dong are with the RII Lab (Lab of Robotics Innovation and Intervention), UM-SJTU Joint Institute, Shanghai Jiao Tong University, Shanghai, China (e-mails: yangzhixiong@sjtu.edu.cn and 5123709073@sjtu.edu.cn).

position in a geometric way rather than computing the joint values. This heuristic approach is proven convergent, efficient and versatile in various scenarios without suffering from singularity problems [28].

While the original FABRIK method only handles the IK problem of rigid-linked robots, this paper proposes the FABRIKc method, by reformulating the FABRIK procedure for the IK problems of continuum robots. Each segment of the continuum robot will be replaced by two virtual rigid links and a virtual joint. The process of replacing an arc segment with rigid links and joints in this paper is different from that in [21] where each segment was replaced by only one rigid link connecting the base and the tip of the segment. What's more, the FABRIKc method can solve the IK problems with prescribed segment lengths and end effector orientations, whereas these solutions were not guaranteed by the method in [21].

This paper is organized as follows. Section II summarizes the kinematics of a single segment and explains how to replace the segment with two virtual links and one virtual joint. The FABRIKc method is elaborated in Section III, while several simulation case studies for different continuum robots are detailed in Section IV to demonstrate the efficacy of this method. Section V summarizes the conclusions and the future work.

II. MODELING OF A SINGLE SEGMENT

Modeling of a single segment only concerns the mapping between the configuration space and the Cartesian workspace, in order to make the derivation actuation independent. Following the constant curvature assumption, kinematics of a single segment (the t th segment) is summarized in this section, referring to the details in [29].

The nomenclature is listed in Table I, while the assigned coordinate systems are defined as follows, referring to Fig. 1.

- **Base coordinate** $\{tb\} \equiv \{\hat{x}_{tb}, \hat{y}_{tb}, \hat{z}_{tb}\}$ locates its origin at the center of the base cross section with \hat{z}_{tb} perpendicular to the base cross section.
- **Bending plane coordinate 1** $\{t1\} \equiv \{\hat{x}_{t1}, \hat{y}_{t1}, \hat{z}_{t1}\}$ shares its origin with $\{tb\}$ and has the segment bending in its XY plane.
- **Bending plane coordinate 2** $\{t2\} \equiv \{\hat{x}_{t2}, \hat{y}_{t2}, \hat{z}_{t2}\}$ is obtained by rotating $\{t1\}$ about \hat{z}_{t1} for an angle θ_t so that \hat{x}_{t2} is perpendicular to the end cross section. The origin of $\{t2\}$ locates at the center of the end cross section.
- **End coordinate** $\{te\} \equiv \{\hat{x}_{te}, \hat{y}_{te}, \hat{z}_{te}\}$ locates its origin at the center of the end cross section. \hat{z}_{te} aligns with \hat{x}_{t2} while \hat{x}_{te} is obtained by rotating \hat{y}_{t2} about \hat{z}_{te} for an angle δ_t .

TABLE I

NOMENCLATURE USED IN KINEMATICS MODELING AND ALGORITHM

Symbol	Representation
n	Number of the segments
t	Index of the segments, $t = 1, 2, \dots, n$. $t = 1$ for the most proximal

	segment and $t = n$ for the most distal segment.
L_t	Length of the t th segment
θ_t	Bending angle of the t th segment in the bending plane; $\theta_t = 0$ indicates that the t th segment is straight.
δ_t	The right-handed rotation angle from \hat{y}_{t1} about \hat{z}_{t1} to \hat{x}_{te}
l_t	Length of the virtual link 1 and 2
$\mathbf{p}_{tb}, \mathbf{p}_{te}$	Center position of the t th segment's base cross section and end cross section in the world coordinate
${}^{tb}\mathbf{p}_{te}$	Center position of the t th segment's end cross section in $\{tb\}$
$\mathbf{R}_{tb}, \mathbf{R}_{te}$	The coordinate transformation matrix mapping from $\{tb\}$ and $\{te\}$ to the world coordinate, respectively.
${}^{tb}\mathbf{R}_{te}$	The coordinate transformation matrix relating $\{te\}$ and $\{tb\}$
\mathbf{p}^*	The target position in the world coordinate
$\hat{\mathbf{x}}^*, \hat{\mathbf{z}}^*$	The target orientation in the world coordinate
k, k_{\max}	The iteration index and the maximal allowable iteration number

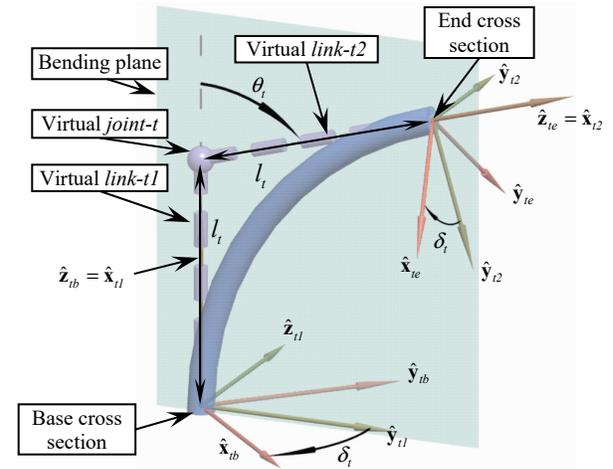


Fig. 1. Nomenclature and coordinates of the t th segment

With the nomenclature and the coordinates defined, the direct kinematics of a single segment can be formulated as follows referring to [29]. The center of the end cross section of the t th segment is in (1):

$${}^{tb}\mathbf{p}_{te} = \frac{L_t}{\theta_t} \begin{bmatrix} \cos \delta_t (1 - \cos \theta_t) \\ \sin \delta_t (\cos \theta_t - 1) \\ \sin \theta_t \end{bmatrix} \quad (1)$$

Where ${}^{tb}\mathbf{p}_{te} = [0 \ 0 \ L_t]^T$ when $\theta_t = 0$.

The coordinate transformation matrix relating $\{te\}$ and $\{tb\}$ is as follows:

$${}^{tb}\mathbf{R}_{te} = {}^{tb}\mathbf{R}_{t1} {}^{t1}\mathbf{R}_{t2} {}^{t2}\mathbf{R}_{te} \quad (2)$$

$$\text{Where } {}^{tb}\mathbf{R}_{t1} = \begin{bmatrix} 0 & \cos \delta_t & \sin \delta_t \\ 0 & -\sin \delta_t & \cos \delta_t \\ 1 & 0 & 0 \end{bmatrix},$$

$${}^{t1}\mathbf{R}_{t2} = \begin{bmatrix} \cos \theta_t & -\sin \theta_t & 0 \\ \sin \theta_t & \cos \theta_t & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } {}^{t2}\mathbf{R}_{te} = {}^{tb}\mathbf{R}_{t1}^T.$$

To implement the proposed FABRIKc algorithm in Section III, the shape and kinematics of a single segment shall be represented by a virtual joint and two virtual links as shown in

Fig. 1. The virtual joint (namely *joint-t*) is located at the intersection of the normals of the segment's base cross section and end cross section. The two virtual links (*link-t1* and *link-t2*) connect the *joint-t* and the origins of the base cross section and the end cross section, respectively. The *link-t1* and *link-t2* are in the directions of $\hat{\mathbf{z}}_{tb}$ and $\hat{\mathbf{z}}_{te}$, correspondingly.

It should be noted that when L_t is constant for a specific segment, the virtual links' lengths l_t is not constant and can be obtained as follows:

$$l_t = \frac{L_t}{\theta_t} \tan\left(\frac{\theta_t}{2}\right) \quad (3)$$

Where $l_t = L_t/2$ when $\theta_t = 0$.

Then the position of the *joint-t* can be written as in (4) or (5).

$$\mathbf{p}_{tj} = \mathbf{p}_{tb} + l_t \hat{\mathbf{z}}_{tb} \quad (4)$$

$$\mathbf{p}_{tj} = \mathbf{p}_{te} - l_t \hat{\mathbf{z}}_{te} \quad (5)$$

For a single segment, if the position of the center of the end cross section ${}^{tb}\mathbf{p}_{te} = [{}^{tb}x_{te} \quad {}^{tb}y_{te} \quad {}^{tb}z_{te}]^T$ is known, the configuration variable δ_t and θ_t can be obtained from (6) and (7):

$$\delta_t = \text{atan2}(-{}^{tb}y_{te}, {}^{tb}x_{te}) \quad (6)$$

Where $\text{atan2}(y, x)$ is the right-handed angle between the x-axis and a ray passing the origin and the point (x, y). δ_t is singular and can be any value when ${}^{tb}x_{te} = {}^{tb}y_{te} = 0$.

$$\theta_t = \arccos(\hat{\mathbf{z}}_{tb} \cdot \hat{\mathbf{z}}_{te}) \quad (7)$$

With δ_t and θ_t obtained, L_t can be calculated from (1).

In the FABRIKc algorithm presented in Section III, the information about the virtual joints and the virtual links of all the segments in a continuum robot can be rapidly computed by finding the virtual joints' locations \mathbf{p}_{tj} .

III. FABRIKc ALGORITHM

In this section, the FABRIKc method for solving the IK problem of continuum robots is presented, by re-formulating the FABRIK method for the IK problem of articulated robots.

The FABRIKc algorithm updates the robot's configuration variables through two phases in each iteration: the forward reaching phase and the backward reaching phase.

As shown in Fig. 2, The FABRIKc algorithm starts with the target position \mathbf{p}^* and orientation $\hat{\mathbf{z}}^*$ of the robot's end effector as well as its current pose (L_t , θ_t and δ_t , $t = 1, 2, \dots, n$). Then \mathbf{p}_{tj} and l_t for each segment are initialized using (3) and (4) for the current robot pose. The residual positioning error e is calculated from (8). Only the positioning error is concerned because the FABRIKc algorithm inherently guarantees the orientation of the robot's end effector.

$$e = \|\mathbf{p}^* - \mathbf{p}_{ne}\| \quad (8)$$

Where \mathbf{p}_{ne} is the center position of the end cross section of the n th segment.

When the error e is bigger than a threshold ε and the

iteration index k is smaller than the maximal allowable iteration number k_{max} , the FABRIKc algorithm goes through a two-phase-iteration process.

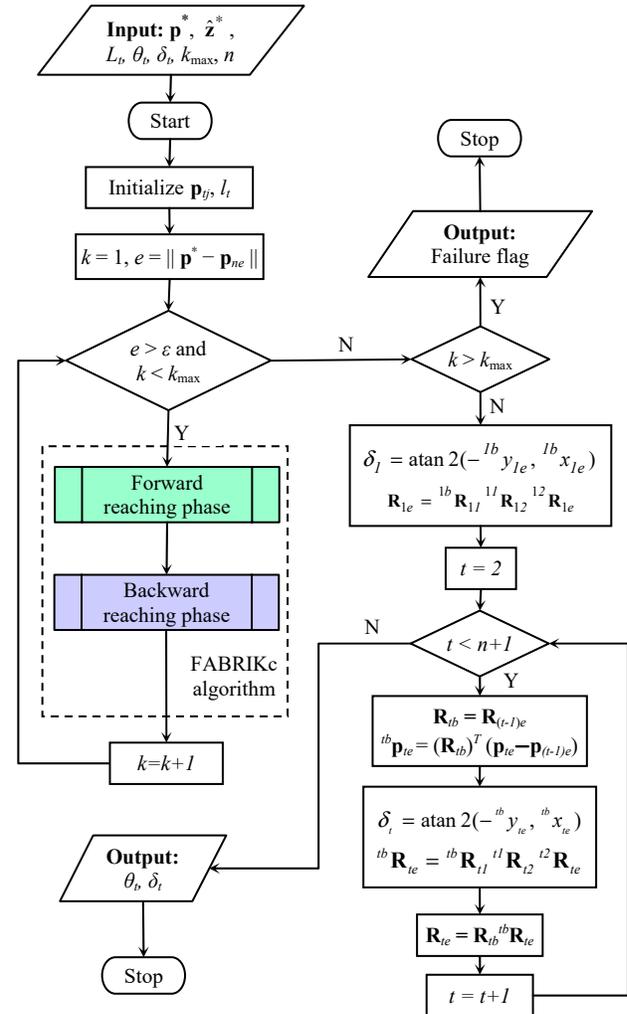


Fig. 2. Flowchart of the FABRIKc algorithm

In the forward reaching phase, the end cross section of the robot's n th segment is firstly moved to the target position with aligned orientation by setting $\mathbf{p}_{ne} = \mathbf{p}^*$ and $\hat{\mathbf{z}}_{ne} = \hat{\mathbf{z}}^*$. Then the position \mathbf{p}_{nj} of the *joint-n* can be obtained using (5). Next, the normal of the base cross section of the robot's n th segment can be obtained as in (9). Subsequently, θ_n , l_n and \mathbf{p}_{nj} should be updated one by one using (7), (3) and (5) respectively.

$$\hat{\mathbf{z}}_{nb} = (\mathbf{p}_{nj} - \mathbf{p}_{(n-1)j}) / \|\mathbf{p}_{nj} - \mathbf{p}_{(n-1)j}\| \quad (9)$$

The handling of the n th segment above should be repeated for from the $(n-1)$ th segment to the 1st segment. For example, the end cross section of the robot's t th segment should be moved to the base cross section of the $(t+1)$ th segment by setting $\mathbf{p}_{te} = \mathbf{p}_{(t+1)b} = \mathbf{p}_{(t+1)j} - l_{(t+1)} \hat{\mathbf{z}}_{(t+1)b}$ and $\hat{\mathbf{z}}_{te} = \hat{\mathbf{z}}_{(t+1)b}$. Then the position \mathbf{p}_{tj} of the *joint-t* can be obtained using (5). Next, the normal of the base cross section of the robot's t th segment can be obtained as in (10). Subsequently, θ_t , l_t and \mathbf{p}_{tj} should be updated one by one using (7), (3) and (5) respectively. The only special treatment for the 1st segment is

to let $\hat{\mathbf{z}}_{tb} = [0 \ 0 \ 1]^T$ before updating θ_l and l_l using (7) and (3).

$$\hat{\mathbf{z}}_{tb} = (\mathbf{p}_{ij} - \mathbf{p}_{(t-1)j}) / \|\mathbf{p}_{ij} - \mathbf{p}_{(t-1)j}\| \quad (10)$$

In the backward reaching phase, the first step is to move the base cross section of the robot's 1st segment to the origin of the world coordinate (a.k.a, $\{1b\}$) by setting $\mathbf{p}_{1b} = [0 \ 0 \ 0]^T$. Then \mathbf{p}_{1j} can be obtained using (4). Next, the normal of the end cross section of the robot's 1st segment can be obtained as in (11). Subsequently, θ_l , l_l and \mathbf{p}_{1j} should be updated one by one using (7), (3) and (4) respectively.

$$\hat{\mathbf{z}}_{1e} = (\mathbf{p}_{2j} - \mathbf{p}_{1j}) / \|\mathbf{p}_{2j} - \mathbf{p}_{1j}\| \quad (11)$$

The handling of the 1st segment above should be repeated for from the 2nd segment to the n th segment. For example, the base cross section of the robot's t th segment should be moved to the end cross section of the $(t-1)$ th segment by setting $\mathbf{p}_{tb} = \mathbf{p}_{(t-1)e} = \mathbf{p}_{(t-1)j} + l_{(t-1)}\hat{\mathbf{z}}_{(t-1)e}$ and $\hat{\mathbf{z}}_{tb} = \hat{\mathbf{z}}_{(t-1)e}$. Then the position \mathbf{p}_{ij} of the $joint-t$ can be obtained using (4). Next, the normal of the end cross section of the robot's t th segment can be obtained as in (12). Subsequently, θ_t , l_t and \mathbf{p}_{ij} should be updated one by one using (7), (3) and (4) respectively. The only special treatment for the n th segment is to let $\hat{\mathbf{z}}_{ne} = \hat{\mathbf{z}}^*$ before updating θ_l , l_l and \mathbf{p}_{nj} using (7), (3) and (4) respectively.

$$\hat{\mathbf{z}}_{te} = (\mathbf{p}_{(t+1)j} - \mathbf{p}_{ij}) / \|\mathbf{p}_{(t+1)j} - \mathbf{p}_{ij}\| \quad (12)$$

Lastly, \mathbf{p}_{ne} is updated from (5) using \mathbf{p}_{nj} , l_n and $\hat{\mathbf{z}}_{ne}$. With the positioning error e updated using (8), the iteration will terminate or repeat.

When the iteration terminates with $e < \varepsilon$ and $k < k_{max}$, δ_l and \mathbf{R}_{le} are first calculated using (6) and (2) respectively, as indicated in Fig. 2. Then for from the 2nd segment to the n th segment, \mathbf{R}_{tb} is firstly calculated as $\mathbf{R}_{(t-1)e}$. Next, ${}^{tb}\mathbf{p}_{te}$ is calculated using (13), while δ_t and ${}^{tb}\mathbf{R}_{te}$ are obtained using (6) and (2) respectively. Please note, all the θ_t values have been obtained during the two-phase iteration.

$${}^{tb}\mathbf{p}_{te} = \mathbf{R}_{tb}^T (\mathbf{p}_{te} - \mathbf{p}_{(t-1)e}) \quad (13)$$

With all the above variables obtained, the IK problem of the continuum robot is solved.

In order to further demonstrate the implementation of the FABRIKc algorithm, the forward and the backward reaching phases are depicted in Fig. 3 on a 3-segment continuum robot with constant segment lengths. Each segment possesses two bending DoFs (Degrees of Freedom).

The initial and target poses of the 3-segment continuum robot is shown in Fig. 3(a).

In the forward reaching phase, the end cross section of the 3rd segment is moved to the target position ($\mathbf{p}_{3e} = \mathbf{p}^*$) with the orientation aligned. The line from \mathbf{p}_{3e} along $-\hat{\mathbf{z}}^*$ with the length of l_3 gives the position of the $joint-3$ (\mathbf{p}_{3j}). Connecting \mathbf{p}_{3j} and \mathbf{p}_{2j} gives a line to determine θ_3 as in Fig. 3(b). Then l_3 shall be updated using the new θ_3 as in Fig. 3(c). This gives an updated value for \mathbf{p}_{3j} as well as \mathbf{p}_{3b} . Next, the end cross section of the 2nd segment is moved to the base cross section

of the 3rd segment. The line from \mathbf{p}_{2e} along $-\hat{\mathbf{z}}_{2e}$ with the length of l_2 gives the position of the $joint-2$ (\mathbf{p}_{2j}). Connecting \mathbf{p}_{2j} and \mathbf{p}_{1j} gives a line to determine θ_2 as in Fig. 3(d). Then l_2 shall be updated using the new θ_2 as in Fig. 3(e). This gives an updated value for \mathbf{p}_{2j} as well as \mathbf{p}_{2b} . After similar treatment for the 1st segment, the center of the base cross section will move away from the origin, as in Fig. 3(f). As the base of the 1st segment shall not move, the backward reaching phase is necessary. As shown in Fig. 3(g), the base cross section of the 1st segment should be moved to the origin. Then the 2nd and the 3rd segments shall be moved, while updating θ_2 , l_2 , \mathbf{p}_{2j} , θ_3 , l_3 and \mathbf{p}_{3j} subsequently. A full iteration completes after the backward reaching phase is finished.

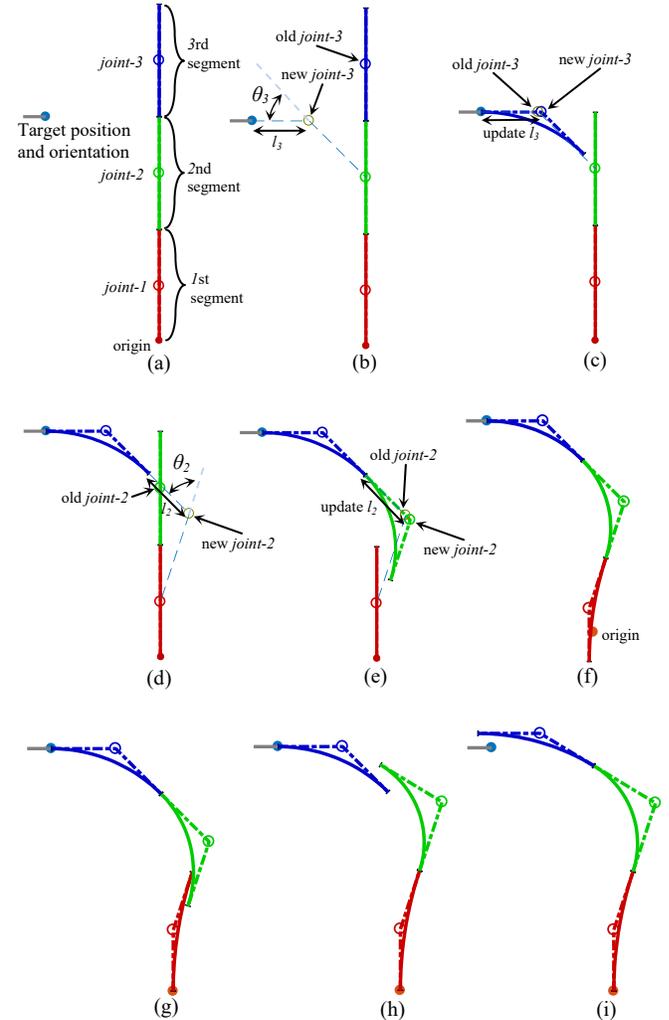


Fig. 3. An example of the two-phase iteration of the proposed FABRIKc algorithm on a 3-segment continuum robot: (a) the initial and the target poses, (b) update θ_3 , (c) update the $joint-3$ and l_3 , (d ~ f) continue the algorithm for the rest segments, (g) move the base cross section of the 1st segment to the origin, (h and i) move the 2nd and the 3rd segments while updating θ_2 , l_2 , \mathbf{p}_{2j} , θ_3 , l_3 and \mathbf{p}_{3j} subsequently.

As a heuristic algorithm, the proposed FABRIKc algorithm possesses quite a few advantages, besides its straightforward implementation. Representing the segments using the virtual joints and the virtual links avoids the calculation of the Jacobian matrix and its inverse. The singularity concern is hence eased since the Jacobian matrix is not involved. The

forward and the backward reaching phases dramatically reduced required computational load. By updating l_t using (3), the segment lengths (L_t) can be easily controlled (setting constant values or value ranges).

IV. SIMULATION CASE STUDIES

Two simulation case studies are presented in this section to demonstrate the effectiveness of the proposed FABRIKc algorithm.

The first simulation case study is for a three-segment continuum robot. Each segment has a constant length and possesses two bending DoFs. A robot with the same structural topology was implemented for palpation tasks in [30]. The second simulation case study is for a hybrid continuum robot comprised of two constant-length continuum segments and two straight rigid segments. A prismatic feeding joint translates the entire robot. This case study shows that the proposed FABRIKc algorithm can also handle rigid elements.

To demonstrate the efficiency of the FABRIKc algorithm, the inverse Jacobian method was also used to deal with the same tasks for comparison. All the simulations were carried out with MATLAB codes on a 3.5-GHz Core i5 processor on a Window 10 platform.

A. Case #1

The proposed FABRIKc algorithm was firstly tested on the 3-segment continuum robot with the segments lengths defined in Table II.

$L_1=50$ mm	$L_2=40$ mm	$L_3=30$ mm

The simulation drove the continuum robot from an initial configuration/pose to a target configuration/pose. The simulation results for two different choices of the initial configurations solved by the inverse Jacobian method and the FABRIKc algorithm are plotted in Fig. 4 and Fig. 5, respectively.

It can be seen from Fig. 4 that the inverse Jacobian method can drive the robot towards a target pose. But there do exist scenarios that the inverse Jacobian method can lead to a trapped configuration of the robot, for example in Fig. 4(b). When the robot is trapped, the position and the orientation errors will not continue to decrease, as shown in Fig. (d). The apparent reason for this trapping is that the θ_1 and θ_2 angles reached their motion limits that are 90° . The fundamental reason might be that the desired motion twist, which is mapped by the inverse Jacobian, generates the configuration variable rates with opposite directions. This wrongly pushes the configuration variables towards their range limits and eventually traps the inverse Jacobian method.

On the other hand, the FABRIKc algorithm so far shows good versatility and robustness. Both initial configurations in Fig. 5, which are identical to those in Fig. 4, can reach the target configurations. Even only after the first iteration, the robot has reached the poses very close to the target poses. The poses with the virtual joints and the virtual links are shown in

Fig. 5(a and b). The FABRIKc algorithm terminates after 12 iterations when the error is less than 0.001 mm. It can be seen from Fig. 5(c and d) that the positioning errors were reduced to around 1 mm even only after the first iteration.

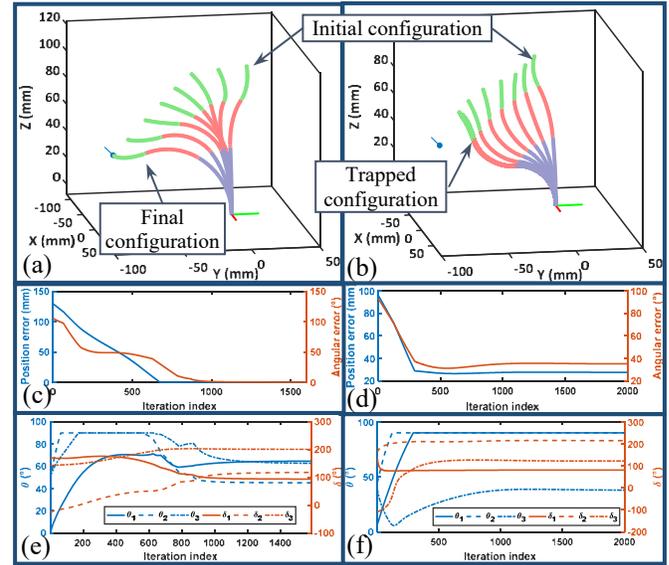


Fig. 4. Inverse Jacobian method for driving the continuum robot from the initial configurations to the target configurations: (a and b) robot poses, (c and d) position errors and angular errors of the end effector, and (e and f) configuration variables during the iterations

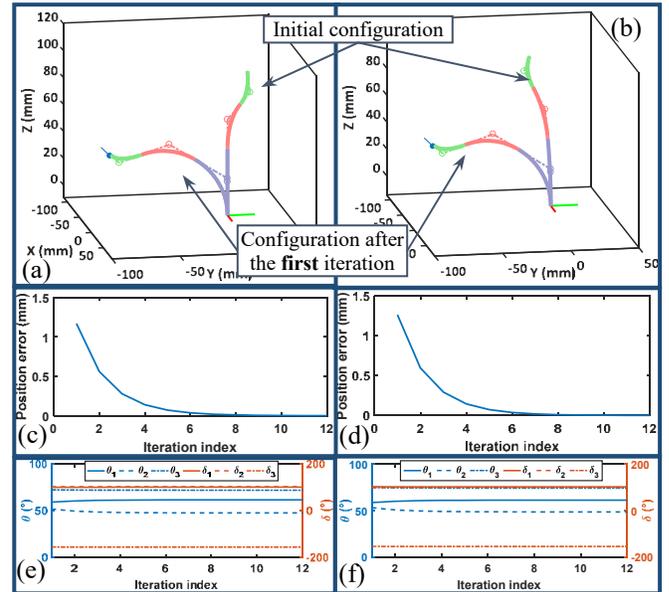


Fig. 5. The FABRIKc algorithm for the IK problem: (a and b) initial configurations and the configurations after the first iteration, (c and d) position errors, and (e and f) configuration variables during the iterations

B. Case #2

The proposed FABRIKc algorithm was implemented on a two-segment continuum robot as shown in Fig. 6(a). Within this robot, each of the two continuum segments possesses two bending DoFs. A length varying rigid segment can translate the entire robot, while the 1st rigid segment with a fixed length connects the 1st and the 2nd continuum segments. The structural parameters of the hybrid continuum robot are listed in Table III. This case study indicates that the FABRIKc

algorithm can also handle rigid elements.

While handling the rigid segment that connects the two continuum segments, trivial modifications were introduced: translating the rigid segments to append the base or end cross sections of the previous or the last continuum segments.

While handling the prismatic feeding joint, the joint is represented as a length-varying rigid segment. Then after the forward reaching phase, the base cross section of the feeding segment was moved away from the origin, as shown in Fig. 7(a). Since the length-varying rigid segment's length is changeable, the error component along the Z-axis can be corrected via varying the length as shown in Fig. 7(b). Then the segment is translated back to the origin as shown in Fig. 7(c) and the back reaching phase continues.

The simulation is about moving the end effector to point to the inspection target point from different orientations. First, the desired positions for the end effector were picked on a spherical surface. The radius of the spherical surface was the desired viewing distance. The orientation of the end effector was then determined by pointing to the inspection target. This simulation mimics an endoscopic inspection task. The results are illustrated in Fig. 6(b). The iterations and time for the simulations are presented in Table IV. The inverse Jacobian method was also used to solve the problem and the results are listed in Table IV for comparison. It can be clearly seen that the FABRIKc algorithm is very efficient: about 10 times faster than the inverse Jacobian method.

TABLE III
SEGMENTS LENGTHS OF THE HYBRID CONTINUUM ROBOT

$L_{c1} = 40$ mm	$L_{feeding} \in [0, 150]$ mm	$L_{c2} = 60$ mm	$L_{r1} = L_{r2} = 20$ mm
------------------	-------------------------------	------------------	---------------------------

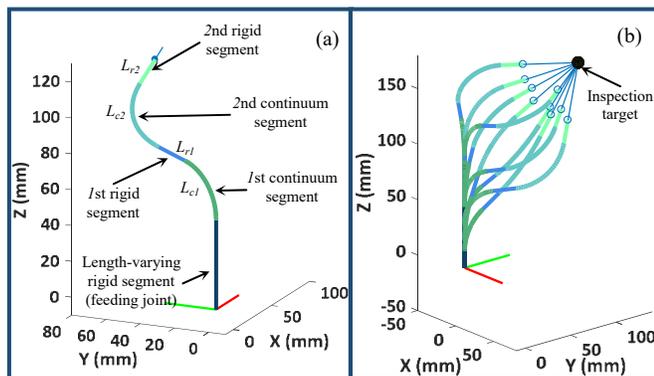


Fig. 6. (a) The hybrid continuum robot; (b) IK solutions, for possibly inspecting a target from different view

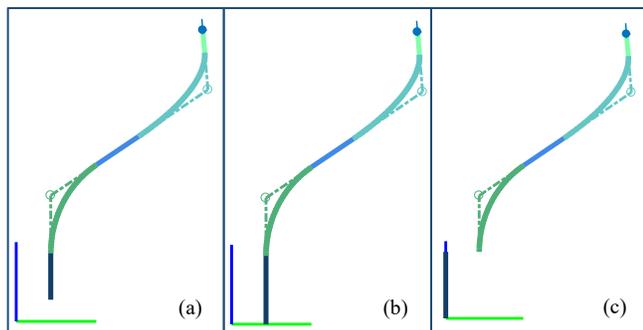


Fig. 7. (a) The length-varying rigid segment is moved away from the origin; (b) change the segment's length, and (c) translate the length-varying rigid segment back to the origin

TABLE IV
SIMULATIONS RESULTS OF THE HYBRID CONTINUUM ROBOT

Case	Average iterations	Average time (second)	Max. iterations	Max. time (second)	
#2	FABRIKc	44.5556	0.0222	112	0.0826
	Jacobian	1604.8	0.1926	1722	0.2177

V. CONCLUSIONS AND FUTURE WORK

This paper proposes the FABRIKc algorithm, an iterative IK solver for continuum robots, reformulating the FABRIK algorithm that was originally designed for articulated robots.

Implementation of the FABRIKc algorithm is elaborated, including the representation of the continuum segments using virtual joints and virtual links, as well as the development of the forward and the backward reaching phases.

The FABRIKc algorithm is a heuristic iterative process where the forward and backward reaching phases alternately change the robot configurations and quickly return the IK solutions. It can be potentially used for real-time control and motion planning since it has low computational cost.

The FABRIKc algorithm was tested on two continuum robots with different structures. The results suggest that the FABRIKc algorithm is about ten times faster than the inverse Jacobian method and can handle the scenarios where the IK problem fails to be solved by the inverse Jacobian method.

Future investigation on the FABRIKc algorithm include several aspects, including i) active incorporating joint limits (the joint limits are now checked after the solution was obtained), ii) formulating for obstacle avoidance and self-motions, and iii) obtaining complete IK solutions for a continuum robot.

REFERENCES

- [1] G. Robinson and J. B. Davies, "Continuum Robots - A State of the Art," in *IEEE International Conference on Robotics and Automation (ICRA)*, Detroit, Michigan, 1999, pp. 2849-2853.
- [2] I. D. Walker, "Continuous Backbone "Continuum" Robot Manipulators," *ISRN Robotics*, vol. 2013, No.726506, pp. 1-19, 2013.
- [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] D. B. Camarillo, C. R. Carlson, and J. K. Salisbury, "Configuration Tracking for Continuum Manipulators With Coupled Tendon Drive," *IEEE Transactions on Robotics*, vol. 25, No.4, pp. 798-808, Aug 2009.
- [7] K. Xu and N. Simaan, "Analytic Formulation for the Kinematics, Statics and Shape Restoration of Multibackbone Continuum Robots via Elliptic Integrals," *Journal of Mechanisms and Robotics*, vol. 2, No.011006, pp. 1-13, Feb 2010.
- [8] R. J. Webster, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, "Nonholonomic Modeling of Needle Steering," *The International Journal of Robotics Research*, vol. 25, No.5-6, pp. 509-525, May 2006.
- [9] W. McMahan, V. Chitrakaran, M. Csencsits, 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.
- [10] D. B. Camarillo, C. F. Milne, C. R. Carlson, M. R. Zinn, and J. K. Salisbury, "Mechanics Modeling of Tendon-Driven Continuum Manipulators," *IEEE Transactions on Robotics*, vol. 24, No.6, pp. 1262-1273, Dec 2008.
- [11] N. Simaan, K. Xu, A. Kapoor, W. Wei, P. Kazanzides, P. Flint, and R. H. Taylor, "Design and Integration of a Telerobotic System for Minimally Invasive Surgery of the Throat " *International Journal of Robotics Research*, vol. 28, No.9, pp. 1134-1153, 2009.
- [12] P. E. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and Control of Concentric-Tube Robots," *IEEE Transactions on Robotics*, vol. 26, No.2, pp. 209-225, April 2010.
- [13] J. Zhao, X. Zheng, M. Zheng, A. J. Shih, and K. Xu, "An Endoscopic Continuum Testbed for Finalizing System Characteristics of a Surgical Robot for NOTES Procedures," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Wollongong, Australia, 2013, pp. 63-70.
- [14] K. Xu, J. Zhao, and M. Fu, "Development of the SJTU Unfoldable Robotic System (SURS) for Single Port Laparoscopy," *IEEE/ASME Transactions on Mechatronics*, vol. 20, No.5, pp. 2133-2145, Oct 2015.
- [15] 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.
- [16] X. Dong, D. Axinte, D. Palmer, S. Cobos, M. Raffles, A. Rabani, and J. Kell, "Development Of a Slender Continuum Robotic System For On-Wing Inspection/Repair of Gas Turbine Engines," *Robotics and Computer-Integrated Manufacturing*, vol. 44, pp. 218-229, April 2017.
- [17] 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, No.2, pp. 299-307, June 2003.
- [18] D. C. Rucker and R. J. Webster, "Statics and Dynamics of Continuum Robots With General Tendon Routing and External Loading," *IEEE Transactions on Robotics*, vol. 27, No.6, pp. 1033-1044, Dec 2011.
- [19] W. S. Rone and P. Ben-Tzvi, "Continuum Robot Dynamics Utilizing the Principle of Virtual Power," *IEEE Transactions on Robotics*, vol. 30, No.1, pp. 275-287, Feb 2014.
- [20] P. Sears and P. Dupont, "A Steerable Needle Technology Using Curved Concentric Tubes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, 2006, pp. 2850-2856.
- [21] S. Neppalli, M. A. Csencsits, B. A. Jones, and I. D. Walker, "Closed-Form Inverse Kinematics for Continuum Manipulators," *Advanced Robotics*, vol. 23, No.15, pp. 2077-2091, 2009.
- [22] 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.
- [23] I. 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.
- [24] T. Zheng, D. T. Branson, E. Guglielmino, R. Kang, G. A. M. Cerda, M. Cianchetti, M. Follador, I. S. Godage, and D. G. Caldwell, "Model Validation of an Octopus Inspired Continuum Robotic Arm for Use in Underwater Environments," *ASME Journal of Mechanisms and Robotics*, vol. 5, No.2, p. 021004, Mar 2013.
- [25] 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.
- [26] K. A. O'Neil, Y.-C. Chen, and J. Seng, "Removing Singularities of Resolved Motion Rate Control of Mechanisms, Including Self-Motion," *IEEE Transactions on Robotics and Automation*, vol. 13, No.4, pp. 741-751, Oct 1997.
- [27] A. Aristidou and J. Lasenby, "FABRIK: A Fast, Iterative Solver for the Inverse Kinematics Problem," *Graphical Models*, vol. 73, No.5, pp. 243-260, 2011/09/01/ 2011.
- [28] A. Aristidou, Y. Chrysanthou, and J. Lasenby, "Extending FABRIK with Model Constraints," *Computer Animation and Virtual Worlds*, vol. 27, No.1, pp. 35-57, 2016.
- [29] K. Xu and N. Simaan, "An Investigation of the Intrinsic Force Sensing Capabilities of Continuum Robots," *IEEE Transactions on Robotics*, vol. 24, No.3, pp. 576-587, June 2008.
- [30] K. Xu and N. Simaan, "Intrinsic Wrench Estimation and Its Performance Index of Multi-Segment Continuum Robots," *IEEE Transactions on Robotics*, vol. 26, No.3, pp. 555-561, June 2010.