

# An Efficient Multi-solution Solver for the Inverse Kinematics of 3-Section Constant-Curvature Robots

Ke Qiu, Jingyu Zhang, Danying Sun, Rong Xiong, Haojian Lu, Yue Wang  
State Key Laboratory of Industrial Control and Technology,  
Zhejiang University, China

**Abstract**—Piecewise constant curvature is a popular kinematics framework for continuum robots. Computing the model parameters from the desired end pose, known as the inverse kinematics problem, is fundamental in manipulation, tracking and planning tasks. In this paper, we propose an efficient multi-solution solver to address the inverse kinematics problem of 3-section constant-curvature robots by bridging both the theoretical reduction and numerical correction. We derive analytical conditions to simplify the original problem into a one-dimensional problem. Further, the equivalence of the two problems is formalised. In addition, we introduce an approximation with bounded error so that the one dimension becomes traversable while the remaining parameters analytically solvable. With the theoretical results, the global search and numerical correction are employed to implement the solver. The experiments validate the better efficiency and higher success rate of our solver than the numerical methods when one solution is required, and demonstrate the ability of obtaining multiple solutions with optimal path planning in a space with obstacles.

## I. INTRODUCTION

Continuum robots are able to band, elongate or twist when actuated [1, 2], granting applications in various tasks like bioinspired grasping [3] and minimally invasive surgery [4, 5, 6]. The piecewise constant curvature is a well-known kinematic framework by modelling the robot with concatenated constant-curvature arcs [7]. In manipulation [8], tracking [9, 10], and planning tasks [11], the set of model parameters should be computed from the desired end pose to guide the motion, which is the inverse kinematics problem [12]. Since many continuum robots composed of an inextensible central backbone have fixed section lengths [9, 13, 14], they need at least 3 sections to reach a specified end pose using the piecewise constant curvature model. Nevertheless, the multi-section inverse kinematics remains an open problem.

An analytical method for the inverse kinematics is always preferred because it is fast and able to find all solutions. In [15], a closed-form geometric method is proposed for specified end translations, but it does not consider the tip orientations. Explicit analytical expressions for 2 extensible sections are derived in [16]. However, This method cannot be directly generalised to robots with inextensible sections [16].

The inverse kinematics is often formulated as a local optimisation problem over model parameters when analytical expressions are absent, such as the Newton-Raphson method

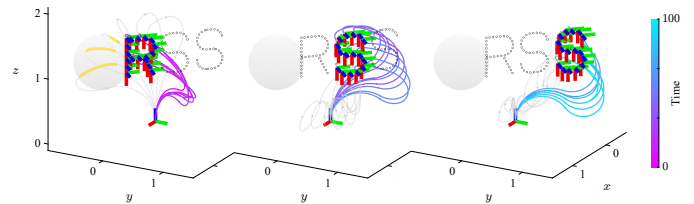


Fig. 1. We achieve the optimal RSS-shaped path planning in the parameter space with our multi-solution inverse kinematics solver. The solid line indicates the resultant motion after dynamic programming, while the dashed line indicates the multiple solutions. Collisions are painted yellow.

[10] and damped least square method [17]. Control schemes can also be implemented to drive the parameters until the tip arrives at the target position [9]. However, numerical inverse kinematics is to find only one solution close to the initial value and, additionally, its iteration process depends sensitively on the initial guess, causing inefficient performance [18].

In this paper, we propose an efficient multi-solution solver to address the inverse kinematics problem of 3-section constant-curvature robots. The idea is to simplify the numerical search by theoretical reduction. Specifically, in a theoretical aspect, we derive the analytical conditions to reduce the dimensionality of the 3-section inverse kinematics and finally arrive at an equivalent one-dimensional problem. By further introducing an approximation, the only dimension becomes traversable and the remaining parameters are derived analytically. We show that there exists an error bound of the approximation. In a numerical aspect, we employ a global search on the one dimension to ensure the resolution completeness of the approximated problem without hurting the efficiency. With a few steps of numerical iterations, we are able to find multiple solutions to the original inverse kinematics problem. The performance in experiments validates the better efficiency and higher success rate of our solver than the pure numerical methods when one solution is required. Besides, the ability to obtain multiple solutions allows us to achieve optimal path planning in a space with obstacles. As shown in Figure 1, we solve the configuration at each via point, yielding a path where we later allocate time. In summary, the contributions are presented as follows.

- 1) We propose two analytical conditions for the 2-section robot when the end rotation or translation is specified. We further propose the necessary conditions for the 3-

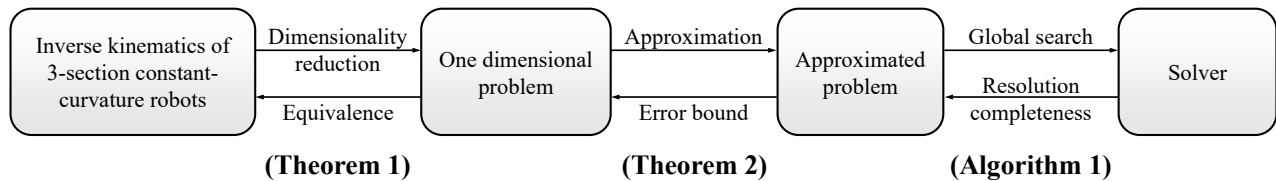


Fig. 2. The flowchart of this work. Rightwards arrows are the procedures. Leftwards arrows are the theoretical guarantees.

- section robot when the end pose is specified.
- 2) We reduce the dimensionality of the original inverse kinematics problem from six to one with these conditions. We formalise the one-dimensional problem and show their equivalence.
  - 3) We introduce an approximation to bring traversability to the reduced problem. An error bound is derived for the approximation.
  - 4) We present the solver, which applies a global search and a few steps of iterations. The computational efficiency and multi-solution accessibility are validated by different tasks in experiments. The code is released.<sup>1</sup>

## II. RELATED WORK

**Modelling and Applications.** In order to better manoeuvre a particular continuum manipulator, the strategy employed for modelling the kinematics is essential, concerning the mechanical design, material properties and geometrical characteristics. It is also indispensable for fitting sensor observation [19] or state estimation [20]. Three popular techniques of modelling include (1) The piecewise constant curvature model (see Webster III and Jones [21] for a review), (2) The Cosserat rod model [22, 23, 24], and (3) The finite element model [25]. Among them, the piecewise constant curvature model is perhaps the most well-known and widely used kinematic framework for continuum robots [7, 20, 26]. In practice, many tasks such as motion planning [11] and trajectory tracking [9, 10, 13] require the solutions to the inverse kinematics. The Cosserat rod model and the finite element model solve forward partial differential equations, considering the mechanics nature of robots and usually taking seconds to minutes for an accurate trajectory [26], while the piecewise constant curvature model is a pure geometric model solving the configuration at each via point, yielding a path where we should later allocate time and interpolate. It is usually computed in milliseconds.

**Analytical Methods.** A closed-form geometric method is given for multiple sections by treating each section as a straight rigid link together with a spherical joint and then using trigonometric relationships [15]. Tip orientations are not accounted for in this method, and approximations are unavoidable when robots have fixed section lengths. Applying the quaternion description yields a simple formulation of the constant-curvature model, and this leads to explicit analytical expressions for the inverse kinematics [16]. Unfortunately, this work is for constant-curvature continuum robots with 2

extensible sections and cannot be directly generalised to those with inextensible sections [16].

**Numerical Methods.** It is natural to resort to numerical methods when there are no analytical approaches available. Nonlinear optimisations can be used to find arc parameters that minimise the pose error. The Newton-Raphson method is a common choice with the pseudoinverse of the Jacobian [10, 17, 27]. Alternatively, the damped least square method is usually employed for better convergence [17]. However, these methods are likely to be influenced by the nonlinearity of the problem, converging to local minima or thrashing about. Multiple initial guesses are often required to get a solution or try to find an alternative solution. A solution to the problem can also be obtained in the sight of control. Recently, a proportional-integral-derivative control scheme is applied and achieves accurate tracking [9]. The steady-state values of parameters are the solution to the inverse kinematics. However, control constants for each parameter need to be tuned individually to account for differences in response and steady-state error caused by coupling effects. Though numerical methods make it possible to find proper model parameters, they rely on the selection of initial guesses, require plenty of computations, and are unable to find multiple solutions. Therefore, a new solver to handle the inverse kinematics problem of the piecewise constant-curvature model is desired.

## III. INVERSE KINEMATICS OF 3-SECTION CONSTANT-CURVATURE ROBOTS

**Notations.** The quaternion algebra  $\mathbb{H}$  is the four-dimensional associative algebra over  $\mathbb{R}$ , and  $\mathbb{S}^n$  denotes the  $n$ -dimensional unit sphere sitting inside  $\mathbb{R}^{n+1}$  [28]. A unit quaternion  $q \in \mathbb{S}^3 \subset \mathbb{H}$  can be written as a sum of real and imaginary parts, i.e.,

$$q = a + bi + cj + dk, \quad (1)$$

where

$$a, b, c, d \in \mathbb{R}, \quad \sqrt{a^2 + b^2 + c^2 + d^2} = 1. \quad (2)$$

The quaternion multiplication is associative,

$$\begin{aligned} \otimes : \mathbb{H} \times \mathbb{H} &\rightarrow \mathbb{H}, \\ q_1, q_2 &\rightsquigarrow q_1 \otimes q_2. \end{aligned}$$

For a unit quaternion  $q$ , the multiplicative inverse  $q^{-1}$  is identical to its conjugate  $q^*$ , which is defined by

$$q^* = a - bi - cj - dk. \quad (3)$$

<sup>1</sup>Homepage: <https://sites.google.com/view/micsolver>.

We use  $(\cdot)^T$  to denote the transpose operation on matrices. For a more elaborate elucidation, see Appendix A.

**Forward Kinematics of Piecewise Constant-Curvature Robots.** We follow the conventional assumptions of the piecewise constant curvature model [21]: (1) sections are modelled as a series of circular arcs, whose bending directions and angles are independent; (2) attachments between two adjacent sections are negligible; (3) adjacent circular arcs are mutually tangential.

Consider a robot modelled in  $N$  arc sections. The Cartesian coordinate systems are denoted by  $\{F_1\}, \{F_2\}, \dots, \{F_N\}$ , respectively, with the origin located at the proximal endpoint of each arc section. There is also a Cartesian coordinate system  $\{F_{N+1}\}$  attached to the distal endpoint of the  $N$ -th arc. For  $\lambda = 1, \dots, N$ , the relative rotation and translation from  $\{F_\lambda\}$  to  $\{F_{\lambda+1}\}$  is denoted by a unit quaternion  $q_\lambda$  and a vector  ${}^{\lambda+1}\mathbf{r} = \mathbf{r}_\lambda$  with respect to  $\{F_\lambda\}$ , and section lengths  $L_\lambda$  is known. Then the forward kinematics can be written as

$$q = q_1 \otimes q_2 \otimes \dots \otimes q_N, \quad (4)$$

$$\mathbf{r} = {}^{N+1}\mathbf{r}, \quad (5)$$

where the translation  $\mathbf{r}$  is calculated by

$${}^{\lambda+1}\mathbf{r} = {}^{\lambda+1}\mathbf{r} + q_\lambda \otimes {}^{\lambda+1}\mathbf{r} \otimes q_\lambda^*, \quad (6)$$

for  $\lambda$  from  $N$  to 1.

**Model Parameterisation.** Figure 3 presents the parameterisation of a 1-section robot using the constant curvature model, where  $\kappa_1$  is the curvature and  $\phi_1$  indicates the bending plane. Assume  $\kappa_1 L_1 \in [0, \pi]$  and  $\phi_1 \in [0, 2\pi]$  because in most cases a continuum robot is unable to exceed this limitation, and according to the geometric relationships, we obtain

$$\mathbf{r}_1 = \frac{1}{\kappa_1} \begin{pmatrix} \cos \phi_1 - \cos \kappa_1 L_1 \cos \phi_1 \\ \sin \phi_1 - \cos \kappa_1 L_1 \sin \phi_1 \\ \sin \kappa_1 L_1 \end{pmatrix} \quad (7)$$

Because the rotational axis lies exactly on the  $(\mathbf{i}, \mathbf{j})$ -plane, the element in the last dimension of the corresponding quaternion must be zero. In fact, we have

$$q_1 = \cos \frac{\kappa_1 L_1}{2} + \sin \frac{\kappa_1 L_1}{2} (-\sin \phi_1 \mathbf{i} + \cos \phi_1 \mathbf{j}). \quad (8)$$

Comparing it with (1) we can get

$$\begin{aligned} a_1 &= \cos \frac{\kappa_1 L_1}{2}, & b_1 &= -\sin \frac{\kappa_1 L_1}{2} \sin \phi_1, \\ c_1 &= \sin \frac{\kappa_1 L_1}{2} \cos \phi_1, & d_1 &= 0. \end{aligned} \quad (9)$$

We have  $a_1 \geq 0$  since the bending angle  $\kappa_1 L_1 \leq \pi$ . From (9) we can see that

$$\kappa_1 = \frac{2}{L_1} \arccos a_1, \quad \phi_1 = \arctan 2(-b_1, c_1). \quad (10)$$

As a result, the parameters  $q_1, \mathbf{r}_1$  are equivalent to the widely used arc parameters  $\kappa_1, \phi_1$ . Furthermore, we note that

$$\hat{\mathbf{r}}_1 = \mathbf{r}_1 / \|\mathbf{r}_1\| = (c_1 \quad -b_1 \quad a_1)^T, \quad (11)$$

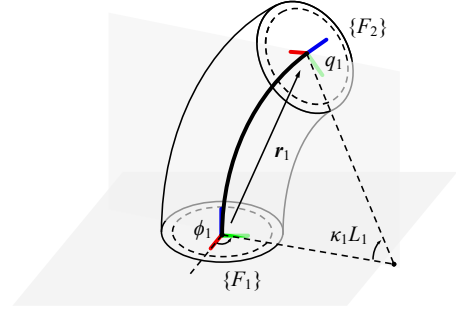


Fig. 3. Continuum robots modelled in constant curvature kinematics. The bold line with length  $L_1$  indicates the circular backbone curve. The parameters describe the angle  $\phi_1$  of the bending plane containing the arc, curvature  $\kappa_1$ , end rotation  $q_1$  and translation  $\mathbf{r}_1$ .

and

$$\|\mathbf{r}_1\| = \frac{L_1 \sqrt{1 - a_1^2}}{\arccos a_1}. \quad (12)$$

So  $q_1$  and  $\mathbf{r}_1$  can be computed from the three entries of  $\hat{\mathbf{r}}_1$  and a constant length  $L_1$ . Therefore,  $\hat{\mathbf{r}}_1$  parameterises the 1-section robot *alone*. The same argument shows that if  $\hat{\mathbf{r}}_\lambda$  is known, then both  $q_\lambda$  and  $\mathbf{r}_\lambda$  are uniquely determined. Consequently, we use  $\hat{\mathbf{r}}_\lambda$  as the model parameter.

**Problem Statement.** Since  $\hat{\mathbf{r}}_\lambda \in \mathbb{S}^2$ , each section offers 2 degrees of freedom. When  $N = 2$ , whether given an end translation or rotation, the robot will always have one redundant degree of freedom. When  $N = 3$ , the robot composed of 3 sections offers 6 degrees of freedom at the end. The number matches that of a pose in the task space, hence the robot is able to satisfy both rotational and translational constraints without redundancy. This is the reason that the 3-section model is widely used. Expand (4) and (5) and we have the problem of the inverse kinematics of 3-section constant-curvature robots.

**Problem 1 (Inverse Kinematics for 3-Section Robots).** *Given the end rotation  $q = (a \quad b \quad c \quad d)^T$  and translation  $\mathbf{r}$ , for  $\lambda = 1, 2, 3$ , let  $\hat{\mathbf{r}}_\lambda = (c_\lambda \quad -b_\lambda \quad a_\lambda)^T$ , then  $q_\lambda$  and  $\mathbf{r}_\lambda$  are determined by  $\hat{\mathbf{r}}_\lambda$ . Find  $\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2, \hat{\mathbf{r}}_3$  such that*

$$q = q_1 \otimes q_2 \otimes q_3, \quad (13)$$

$$\mathbf{r} = \mathbf{r}_1 + q_1 \otimes (\mathbf{r}_2 + q_2 \otimes \mathbf{r}_3 \otimes q_2^*) \otimes q_1^*. \quad (14)$$

#### IV. THEORETICAL RESULTS AND THE SOLVER

As shown in Figure 2, theoretical results are derived so that the range of solutions to Problem 1 can be restricted on a one-dimensional implicit curve (Section IV-A). By approximating the curve as a circle on the unit sphere with bounded error, we are able to traverse the circle and solve the remaining variables analytically (Section IV-B). A traversal on that circle ensures the resolution completeness. Then numerical correction is employed to arrive at the more accurate solution to Problem 1. Bridging the analytical and numerical approaches, the algorithm of our solver is accomplished (Section IV-C).

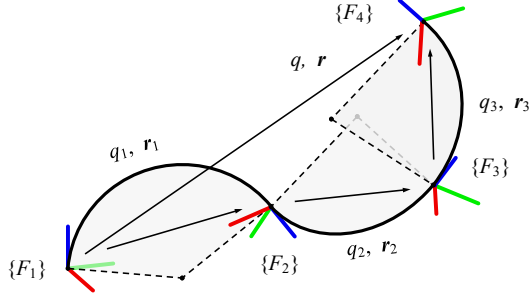


Fig. 4. An illustration of the variables  $q_\lambda, r_\lambda$  and coordinate systems  $\{F_\lambda\}$  in a robot composed of 3 circular arcs with a total of six degrees of freedom.

### A. Dimensionality Reduction

**2-Section Robots.** We begin from the inverse kinematics of 2-section robots, which is formulated as

$$q = q_1 \otimes q_2, \quad (15)$$

$$r = r_1 + q_1 \otimes r_2 \otimes q_1^*. \quad (16)$$

The next lemma gives a property of the 2-section constant curvature inverse kinematics when the end rotation is specified.

**Lemma 1** (Rotational Constraint). *Given an end rotation  $q = (a \ b \ c \ d)^T$ , the solutions to (15) are those  $\hat{r}_1 \in \mathbb{S}^2$  satisfying a linear constraint*

$$\mathbf{n} \cdot \hat{r}_1 = 0, \quad (17)$$

where  $\mathbf{n} = (b \ c \ d)^T$ . When  $\hat{r}_1$  is specified,  $\hat{r}_2$  can also be obtained linearly through

$$\hat{r}_2 = \mathbf{A} \hat{r}_1, \quad (18)$$

where

$$\mathbf{A} = \begin{pmatrix} -a & -d & c \\ d & -a & -b \\ c & -b & a \end{pmatrix}. \quad (19)$$

*Proof:* See Appendix B. ■

Next we analyse the constraint of  $\hat{r}_1$  when the end translation is given. Following (12) we define

$$\rho(a_\lambda, L_\lambda) = \|\mathbf{r}_\lambda\| = \frac{L_\lambda \sqrt{1 - a_\lambda^2}}{\arccos a_\lambda}, \quad (20)$$

then we have the next lemma.

**Lemma 2** (Translational Constraint). *Given an end translation  $\mathbf{r}$ , the solutions to (16) are those  $\hat{r}_1$  satisfying the constraint*

$$\|\mathbf{v}\| = \rho\left(\frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{v}\|}, L_2\right), \quad (21)$$

where

$$\mathbf{u} = (2a_1c_1 \quad -2a_1b_1 \quad 2a_1^2 - 1)^T, \quad (22)$$

and

$$\mathbf{v} = \mathbf{r} - \rho(a_1, L_1) \hat{r}_1. \quad (23)$$

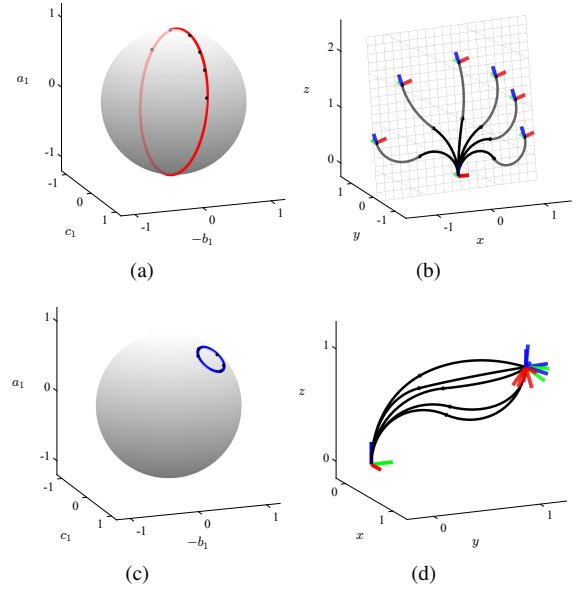


Fig. 5. With selected end rotation  $q = \cos(\alpha/2) + \sin(\alpha/2)\omega$ , where  $\alpha = \pi/10, \omega = (\mathbf{i} - 4\sqrt{6}\mathbf{j} + \sqrt{3}\mathbf{k})/10$ , we have (a)  $\hat{r}_1$  satisfies (17), indicated by the red line (Lemma 1); (b) for these  $\hat{r}_1$ , end rotations are identical, and the endpoints fall on the plane (56) (Corollary 1 in Appendix C). With selected end translation  $\mathbf{r} = (1 \ 1 \ 1)^T$ , we have (c)  $\hat{r}_1$  satisfies (21), indicated by the blue line, and (d) for these  $\hat{r}_1$ , end translations are identical (Lemma 2). Black dots on the left and robot configurations on the right are in a one-to-one correspondence.

After choosing an  $\hat{r}_1$ , we can calculate  $\hat{r}_2$  through

$$\hat{r}_2 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \frac{(2\hat{r}_1\hat{r}_1^T - \mathbf{I})\mathbf{v}}{\|\mathbf{v}\|}. \quad (24)$$

*Proof:* See Appendix D. ■

Note that  $\|\mathbf{u}\| = 1$ , if we approximate  $\rho(a_1, L_1) \approx L_1$  and  $\rho(\mathbf{u}^T \mathbf{v} / \|\mathbf{v}\|, L_2) \approx L_2$ , then the constraint (21) becomes the one derived in Neppalli et al. [15].

**Illustrative Example.** The behaviour of  $\hat{r}_1$  that predicted by Lemma 2, Corollary 1 (in Appendix C), and Lemma 3 are illustrated in Figure 5. When fixing an end rotation  $q$ , the rotational constraint indicates that  $\hat{r}_1$  lies on a great circle. When fixing an end translation  $\mathbf{r}$ , the translational constraint shows that  $\hat{r}_1$  lies approximately on a circle.

**3-Section Robots.** We now move to the 3-section inverse kinematics. Based on our previous results, we find a constraint for both  $\hat{r}_3$  and  $\hat{r}_1$ , which is formalised in the next lemma.

**Lemma 3.** *Given an end rotation  $q$  and translation  $\mathbf{r}$ , suppose  $\{\hat{r}_1, \hat{r}_2, \hat{r}_3\}$  is a solution to Problem 1, then for  $\lambda = 1, 3$ ,  $\hat{r}_\lambda$  satisfies the constraint*

$$\mathbf{r}^T \mathbf{B} \hat{r}_\lambda - \rho(a_\lambda, L_\lambda) d = 0. \quad (25)$$

where

$$\mathbf{B} = \begin{pmatrix} d & a & b \\ -a & d & c \\ -b & -c & d \end{pmatrix}. \quad (26)$$

*Proof:* See Appendix E. ■

From (25) we know that  $\hat{\mathbf{r}}_1$  and  $\hat{\mathbf{r}}_3$  are restricted to *closed* spherical curves with the same form, but no explicit expression has been found yet due to the nonlinearity. We parameterise the curve of  $\hat{\mathbf{r}}_3$  by one variable  $t$  formally to help the subsequent analysis,

$$\hat{\mathbf{r}}_3 = \hat{\mathbf{r}}_3(t), t \in [0, 1]. \quad (27)$$

For any  $t$ , we specify the end rotation and translation of the first two sections by  $\hat{\mathbf{r}}_3(t)$ , formulating a 2-section robot inverse kinematics problem,

$$q_e(t) = q \otimes q_3^{-1}(t) = q \otimes q_3^*(t), \quad (28)$$

$$\mathbf{r}_e(t) = \mathbf{r} - q_e(t) \otimes \mathbf{r}_3(t) \otimes q_e^*(t). \quad (29)$$

We have  $q_e(t) = (a_e \ b_e \ c_e \ d_e)^T$  and we denote

$$\mathbf{n}_e(t) = (b_e \ c_e \ d_e)^T, \quad (30)$$

and

$$\mathbf{v}_e(t) = \mathbf{r}_e(t) - \rho(a_1, L_1) \hat{\mathbf{r}}_1. \quad (31)$$

We now present the main results of this paper: the statement of the one variable problem and the theorem showing its equivalence to Problem 1.

**Problem 2.** *Given the end rotation  $q$  and translation  $\mathbf{r}$ , denote  $\mathbf{u}, \mathbf{n}_e, \mathbf{v}_e$  as in (22), (30), (31), respectively. Let  $S$  be the set of  $\hat{\mathbf{r}}_1 \in \mathbb{S}^2$  that satisfies*

$$\mathbf{n}_e(t) \cdot \hat{\mathbf{r}}_1 = 0, \quad (32)$$

$$\|\mathbf{v}_e(t)\| = \rho \left( \frac{\mathbf{u} \cdot \mathbf{v}_e(t)}{\|\mathbf{v}_e(t)\|}, L_2 \right), \quad (33)$$

$$\mathbf{A} \hat{\mathbf{r}}_1 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \frac{(2\hat{\mathbf{r}}_1 \hat{\mathbf{r}}_1^T - \mathbf{I}) \mathbf{v}_e}{\|\mathbf{v}_e\|}, \quad (34)$$

find  $t$  that makes  $S$  a nonempty set.

**Theorem 1.** *The following statements are equivalent:*

- (a) Finding  $\hat{\mathbf{r}}_\lambda, \lambda = 1, 2, 3$  in Problem 1.
- (b) Finding the one variable  $t$  in Problem 2.

*Proof:* Given  $q, \mathbf{r}$ . Suppose  $\{\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2, \hat{\mathbf{r}}_3\}$  is the solution to Problem 1, then we can first obtain  $t$  from the parametrisation, and according to Lemma 1, 2, 3, this  $\hat{\mathbf{r}}_1$  satisfies (32)-(34) simultaneously. Therefore,  $S$  is not empty and  $t$  is the solution to Problem 2. This shows that (a) implies (b). Inversely, if we know  $t$ , then  $\hat{\mathbf{r}}_3$  is specified, and since  $S \neq \emptyset$ , we take  $\hat{\mathbf{r}}_1 \in S$ . From (32) and Lemma 1 we can get  $\hat{\mathbf{r}}'_2$  such that  $\{\hat{\mathbf{r}}_1, \hat{\mathbf{r}}'_2, \hat{\mathbf{r}}_3\}$  satisfies (13). From (33) and Lemma 2 we can also get  $\hat{\mathbf{r}}''_2$  such that  $\{\hat{\mathbf{r}}_1, \hat{\mathbf{r}}''_2, \hat{\mathbf{r}}_3\}$  satisfies (14). We have  $\hat{\mathbf{r}}'_2 = \hat{\mathbf{r}}''_2 := \hat{\mathbf{r}}_2$  because of (34). Therefore  $\{\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2, \hat{\mathbf{r}}_3\}$  is the solution to Problem 1. This shows that (b) implies (a). ■

**Illustrative Example.** In Figure 6, we visualise (25) for both  $\hat{\mathbf{r}}_3$  and  $\hat{\mathbf{r}}_1$  in magenta. When  $\hat{\mathbf{r}}_3$  is at  $P_1$ , the rotational and translational constraints of  $\hat{\mathbf{r}}_1$  are presented in red and blue, respectively. In this example, the point  $P_2$  is the only intersection that satisfies all three constraints. Consequently, we have to check if the condition (34) holds for  $P_1$  and  $P_2$ . If it does, then by Theorem 1 we have found a solution of Problem 1, namely, the 3-section inverse kinematics problem.

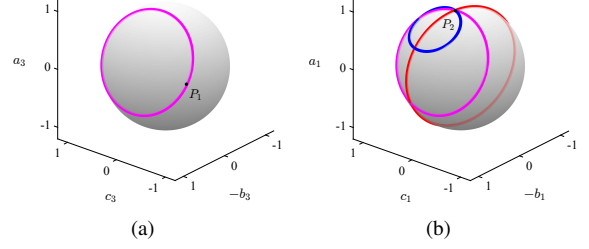


Fig. 6. In (a) and (b), the closed spherical curves are visualised in magenta (Lemma 3). When  $\hat{\mathbf{r}}_3$  is at the point  $P_1$ , the rotational constraint of  $\hat{\mathbf{r}}_1$  is indicated with the red line in (b), as well as the translational constraint with the blue line. Three constraints intersect at  $P_2$ , which is a candidate solution.

## B. Approximation

**Approximated Constraint of  $\hat{\mathbf{r}}_3$ .** The constraint of  $\hat{\mathbf{r}}_3$  is given by (25). Let  $\mathbf{n}_0 = \mathbf{B}^T \mathbf{r}$ , and we approximate (25) as

$$\mathbf{n}_0 \cdot (\hat{\mathbf{r}}_3^{(0)} - \gamma L_3 \mathbf{r}_0) = 0 \quad (35)$$

by relaxing the second term in (25) to  $\gamma L_3 d$ . It refers to a circle on the unit sphere so that  $\hat{\mathbf{r}}_3$  can be analytically traversed by one variable  $t$ . In (35) the factor  $\gamma \in (0, 1)$  is a constant determined later to minimise the approximation error throughout the domain  $a_3 \in [0, 1]$ , and

$$\mathbf{r}_0 = \frac{d\mathbf{n}_0}{\|\mathbf{n}_0\|^2}. \quad (36)$$

**Approximated Constraint of  $\hat{\mathbf{r}}_1$ .** We have known that  $\hat{\mathbf{r}}_1$  satisfies (32), which is a linear equation. Since the solution to Problem 1 satisfies (25) according to Lemma 3, the solution to Problem 2 must also satisfies (25) by Theorem 1, hence we make an approximation analogous to (35) and combine it with (32), which leads to the approximation of  $\hat{\mathbf{r}}_1$ ,

$$\begin{cases} \mathbf{n}_e \cdot \hat{\mathbf{r}}_1 = 0, \\ \mathbf{n}_0 \cdot (\hat{\mathbf{r}}_1^{(0)} - \gamma L_1 \mathbf{r}_0) = 0. \end{cases} \quad (37)$$

To evaluate the approximation, we have the following theorem.

**Theorem 2.** *The error of approximation (35) is bounded, i.e.,*

$$\left| \hat{\mathbf{n}}_0 \cdot (\hat{\mathbf{r}}_3^{(0)} - \hat{\mathbf{r}}_3) \right| \leq \frac{(\pi - 2) L_3 |d|}{2\pi \|\mathbf{B}^T \mathbf{r}\|}, \quad (38)$$

and when  $\|\mathbf{r}\| > L_1$ , the error of approximation (37) is also bounded, i.e.,

$$\left\| \hat{\mathbf{r}}_1^{(0)} - \hat{\mathbf{r}}_1 \right\| \leq \arccos \frac{2L_1 |d|}{\pi \|\mathbf{B}^T \mathbf{r}\|} - \arccos \frac{L_1 |d|}{\|\mathbf{B}^T \mathbf{r}\|}. \quad (39)$$

*Proof:* See Appendix F. ■

In Figure 7, the black dotted lines show the error bound (38) when we traverse  $\hat{\mathbf{r}}_3$  using (35) instead of (25). From (37) we know that  $\hat{\mathbf{r}}_1$  is the intersection of two circles on the unit sphere. We denote the angle between  $\hat{\mathbf{r}}_1$  and  $\hat{\mathbf{r}}_1^{(0)}$  by  $\beta$ , then  $\max |\beta|$  is an error bound for  $\hat{\mathbf{r}}_1$ , geometrically.



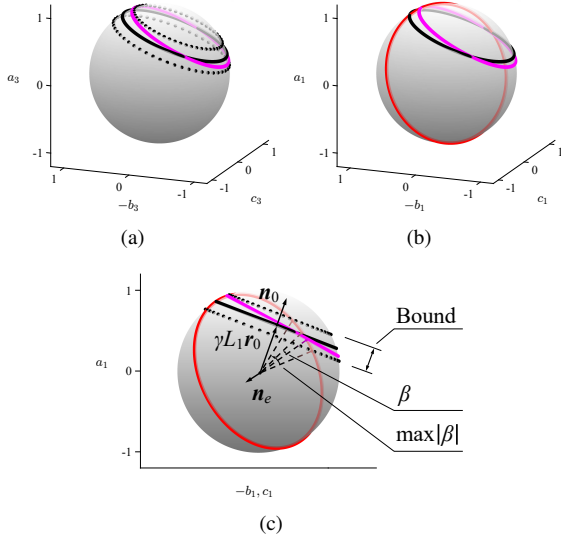


Fig. 7. The black solid lines suggest the approximation of the magenta lines, which represent (25) as previously, while the black dotted lines indicate the error bound. (a) The error is bounded when we traverse  $\hat{r}_3$  along the black solid line instead of the magenta. (b) We use the intersections of the black and red lines to estimate that of the magenta and red line. (c) Viewing from a line of sight that goes along the direction perpendicular to  $\mathbf{n}_0$ , we see that  $\beta$  is the angle between exact  $\hat{r}_1$  and approximated  $\hat{r}_1^{(0)}$ , so an error bound can be given by  $\max|\beta|$ .

### C. Inverse Kinematics Solver

**Circular Traversal.** To make sure resolution completeness, we need to go through all possible discretised  $\hat{r}_3$ . Luckily, the approximation in the previous subsection allows us to make an analytical traversal. It follows from (35) that when  $\mathbf{n}_0$  is not parallel with the  $z$ -axis, we construct

$$\hat{r}_3^{(0)}(t) = \gamma L_3 \mathbf{r}_0 + \sqrt{1 - \|\gamma L_3 \mathbf{r}_0\|^2} \cdot \mathbf{P} \begin{pmatrix} \sin 2\pi t \\ \cos 2\pi t \\ 0 \end{pmatrix}, \quad (40)$$

where  $\hat{\mathbf{n}}_1 = \hat{\mathbf{n}}_0 \times (0 \ 0 \ 1)^T$ ,  $\hat{\mathbf{n}}_2 = \hat{\mathbf{n}}_0 \times \hat{\mathbf{n}}_1$ , and the orthogonal matrix  $\mathbf{P} = (\hat{\mathbf{n}}_1 \ \hat{\mathbf{n}}_2 \ \hat{\mathbf{n}}_0)$ .

**Error Evaluation.** For any  $t \in [0, 1)$ , a unique  $\hat{r}_3$  is specified, then with (37),  $\hat{r}_1$  can be found through analytical expressions when  $\mathbf{n}_0$  is not parallel with  $\mathbf{n}_e$ . We compute  $\hat{r}'_2$  by Lemma 1 and  $\hat{r}''_2$  by Lemma 2. At this point we have two sets of model parameters  $\{\hat{r}_1, \hat{r}'_2, \hat{r}_3\}$  and  $\{\hat{r}_1, \hat{r}''_2, \hat{r}_3\}$ , we choose the one with a smaller error. To illustrate the calculation of error, we denote the actual end pose built from  $\hat{r}_\lambda$  by  $\mathbf{T}$ , the desired end pose built from  $q, \mathbf{r}$  by  $\mathbf{T}_d$ . Define a map from  $\hat{r}_\lambda \in \mathbb{S}^2$  to  $\xi_\lambda \in \mathfrak{se}(3)$  as

$$\sigma : \mathbb{S}^2 \times \mathbb{S}^2 \times \mathbb{S}^2 \rightarrow \mathfrak{se}(3) \times \mathfrak{se}(3) \times \mathfrak{se}(3), \\ \hat{r}_1, \hat{r}_2, \hat{r}_3 \rightsquigarrow \xi_1, \xi_2, \xi_3,$$

where  $\xi_\lambda = L_\lambda (-\kappa_\lambda \sin \phi_\lambda \ \kappa_\lambda \cos \phi_\lambda \ 0 \ 0 \ 0 \ 1)^T$  is the twist coordinates [21]. From (10) we know that the map  $\sigma$  exists. With the product of exponentials formula [29], the end pose  $\mathbf{T} \in \text{SE}(3)$  has the form

$$\mathbf{T} = \mathbf{T}(\xi_1, \xi_2, \xi_3) = \exp \hat{\xi}_1 \exp \hat{\xi}_2 \exp \hat{\xi}_3. \quad (41)$$

We evaluate the error regarding the norm of the body twist from  $\mathbf{T}$  to  $\mathbf{T}_d$ , i.e.,

$$e = e(\xi_1, \xi_2, \xi_3) = \left\| (\log \mathbf{T}^{-1} \mathbf{T}_d)^\vee \right\|. \quad (42)$$

**Candidate Selection.** By traversing  $t$  in  $[0, 1)$ , we have a series of model parameter  $\{\hat{r}_1, \hat{r}_2, \hat{r}_3\}$ , for which we can build an error function  $e(t)$ . Suppose  $\bar{t}$  is a solution to Problem 2, then the derivative of the error function at  $\bar{t}$  must be zero, i.e.,  $\frac{d}{dt} \Big|_{t=\bar{t}} e(t) = 0$ . Therefore, when  $t$  is being traversed, we can collect all the local minima by checking

$$e(\bar{t}) < e(\bar{t} + \Delta t), \quad e(\bar{t}) \geq e(\bar{t} - \Delta t). \quad (43)$$

We calculate the model parameters  $\{\hat{r}_1, \hat{r}_2, \hat{r}_3\}$  from  $\bar{t}$  as candidate solutions to Problem 1.

**Numerical Correction.** The exponential coordinates of the candidates are regarded as the initial values for the optimisation problem below:

$$\begin{aligned} & \text{minimise}_{\xi_1, \xi_2, \xi_3} \left\| (\log \mathbf{T}(\xi_1, \xi_2, \xi_3)^{-1} \mathbf{T}_d)^\vee \right\|^2 \\ & \text{subject to } \xi_1, \xi_2, \xi_3 \in \mathfrak{se}(3). \end{aligned} \quad (44)$$

We employ the Newton-Raphson or damped least square method to reach the accurate solution of Problem 1. Filtering the candidates unable to converge to the accuracy threshold, we form successful ones as our final result.

Because of the error bound in Theorem 2, the initial values are actually very close to the real solutions, therefore only a few steps of iterations are required in the numerical correction. In summary, we present Algorithm 1.

---

#### Algorithm 1 Inverse Kinematics Solver.

---

**Require:**  $q \in \mathbb{S}^3, \mathbf{r} \in \mathbb{R}^3, L_1, L_2, L_3 > 0$

```

1: procedure SOLVER( $q, \mathbf{r}$ )
2:    $\Delta t \leftarrow$  A search resolution
3:   for  $t = 0, t \leftarrow t + \Delta t, t < 1$  do
4:      $\hat{r}_3, \hat{r}_1 \leftarrow \hat{r}_3^{(0)}(t), \hat{r}_1^{(0)}(t)$  ▷ (40), (37)
5:      $\hat{r}'_2, \hat{r}''_2 \leftarrow$  Compute (18) and (24)
6:      $e, \hat{r}_2 \leftarrow \min\{e(\hat{r}'_2), e(\hat{r}''_2)\}$  ▷ (42)
7:     if  $t$  is a local minimum then
8:       Collect  $\{\hat{r}_1(t), \hat{r}_2(t), \hat{r}_3(t)\}$  in  $S_c$ .
9:     end if
10:  end for
11:  for each elements in  $S_c$  do
12:    if numerical correction converges then
13:      output  $\hat{r}_1, \hat{r}_2, \hat{r}_3$ 
14:    end if
15:  end for
16:  if no solutions are found then
17:    goto line 3 with  $\Delta t$  a finer search resolution
18:  end if
19: end procedure

```

---

## V. SIMULATION RESULTS

We showcase our 3-section inverse kinematics solver in two scenarios: one collision-free solution for a randomly given pose in a free task space or with obstacles (Section V-C, V-D), and path planning under the given sequence of via points (Section V-E).

### A. Setup

The robot used in our experiments is composed of 3 sections with an identical length 1. The evaluation of benchmark algorithms and our solver contains the following steps: (1) Sample random arc parameters  $\{(\kappa_\lambda, \phi_\lambda) : \lambda = 1, 2, 3.\}$  with uniform distributions and compute  $\xi_1, \xi_2, \xi_3$ . (2) Generate the desired pose  $T_d$  using the forward kinematics (41), then compute  $q, r$  from  $T_d$ . (3) Solve the inverse kinematics problem with different algorithms. (4) Repeat the above steps for 2000 times and evaluate the statistics on the results.

For the numerical correction, the criterion of success is that the error (42) is less than 0.01 and the total number of iterations does not exceed the maximum allowed value.

Since the overall average runtime is strongly influenced by the unsuccessful runs that require the maximum number of iterations, we pick out the successful portion to provide a more qualitative indicator of the performance for comparison.

The code for simulations is implemented in MATLAB R2022b and carried on a laptop computer with a 2.10 GHz Intel Core i7-1260P processor.

### B. Benchmarks

As mentioned, solving the inverse kinematics problem always relies on numerical methods. It is essentially a root-finding problem for quaternion equations (13) and (14). Especially, among various root-locating methods, we select several typically representative ones. The Newton-Raphson method is commonly used [10, 17, 27], so we choose it as one of the benchmarks. Additionally, we implement the gradient descending algorithm for this problem. A built-in MATLAB optimiser `fminsearch`, which is a derivative-free method using the Nelder-Mead simplex algorithm [30], is also adopted as a benchmark.

Our solver does not require an initial guess, while for benchmarks we provide random arc parameters as the initial guess because the straight configuration is essentially a singularity [31]. The Jacobian matrix in the iterative procedures has been derived in [32]. We put the expressions in Appendix G.

### C. One Solution in the Task Space

**Free Space.** We start from a free task space. Since the target pose is computed from the forward kinematics with random model parameters, the inverse kinematics problem is ensured to be solvable. Different methods may find either the parameters generating the desired pose, or an alternative solution. Regardless of which one, all the methods stop as soon as one solution is found. The result is shown in Table I.

**With Obstacles.** Structured obstacles are introduced in this scenario. The obstacles are spheres with a diameter of 0.4,

and their centre points form a square lattice in the task space with 0.8 in  $x$ - and  $y$ -axis, 1.0 in  $z$ -axis. In each repeat, we generate parameters whose corresponding configuration should be collision-free at the same time, so every inverse kinematics problem is still solvable here. The methods stop as soon as one arbitrary collision-free solution is found. For pure numerical methods that execute only once, the success rate would definitely decrease due to the influence of obstacles. So the Newton-Raphson method is allowed to execute for at most 5 times, labelled as Newton-Raphson (5). The result is presented in Table II.

**Comparison on Efficiency.** We judge the computational efficiency by the runtime. When there are no obstacles, our solver takes about 15% runtime relative to the Newton-Raphson method, which is the best among all benchmarks. Our solver performs additional computations before starting the numerical correction, which is not needed in benchmark algorithms. Yet the result shows that the operations benefit the overall efficiency because the provided initial value decreases plenty of iterations in the following numerical correction.

When obstacles exist, less than 19% runtime is achieved relative to the Newton-Raphson (5) method. We note that in this experiment our solver takes almost double the time. The increase in runtime is caused by two reasons. On the one hand, when a solution satisfying the desired pose but colliding with obstacles is found, the numerical correction is applied to the next candidate in order to find an alternative collision-free solution. This proportion is 9.35%. On the other hand, when no solutions are found, the solver would employ a retraversal with a finer resolution. The proportion of retraversal is 0.50% in free space, and increases to 2.90% when considering obstacles.

Notably, with our solver, we observe that some initial values are directly output as solutions *without* any iterations. The proportions are 17.55% and 21.75% in the two experiments, respectively. This strongly reduces the computations and meanwhile validates our theoretical results.

**Comparison on Success Rate.** The success rate reflects the dependency of a numerical algorithm on initial values and the ability to obtain multiple solutions when obstacles exist. The derivative-free method and gradient method are more likely to be trapped by local minima. The Newton-Raphson method is sometimes able to escape from local minima yet is more likely to thrash about until running out of all iterations.

Our solver maintains the full percentage of success rate in both experiments, which owes to the efficient multi-solution finding. The results validate the resolution completeness for the approximated problem, as well as the error bound that allows the numerical correction to converge in a few steps inside the bound.

### D. An Example of Multiple Solutions

We present an example of multiple solution finding of our solver in a space with obstacles. For a selected end rotation  $q$  and translation  $r$ , Figure 8(a) shows the error curve  $e(t)$ ,  $t \in [0, 1)$  obtained from traversing  $\hat{r}_3$  with a search resolution  $\Delta t = 0.01$ . We box the local minima as candidates for

TABLE I  
PERFORMANCE OF DIFFERENT METHODS FOR SOLVING THE INVERSE KINEMATICS PROBLEM.

Method	Success Rate (%)	Runtime (ms)	
		Successful	Total
Derivative-free	47.10	59.44	68.16
Gradient	51.05	95.23	178.41
Newton-Raphson	94.20	10.54	12.18
Our Solver	<b>100.00</b>	<b>1.59</b>	<b>1.59</b>

TABLE II  
PERFORMANCE OF DIFFERENT METHODS FOR SOLVING THE INVERSE KINEMATICS PROBLEM WITH LATTICE OBSTACLES.

Method	Success Rate (%)	Runtime (ms)	
		Successful	Total
Derivative-free	40.00	57.28	67.40
Gradient	45.65	95.22	181.11
Newton-Raphson	82.75	10.83	12.09
Newton-Raphson (5)	98.75	16.15	17.00
Our Solver	<b>100.00</b>	<b>3.05</b>	<b>3.05</b>

the following corrections. Finally, a total of 4 candidates all converge in less than 2 steps of iterations. As a result, our solver gets 4 multiple solutions for the desired end pose, as presented in Figure 8(b)-8(e).

Interestingly, 2 to 4 solutions are the most frequently observed results during our experiments. We speculate that the number of solutions of the 3-section inverse kinematics problem belongs to this range.

### E. Path Planning

In this scenario, instead of random sampling, we specify a path composed of a sequence of via points by giving the position and orientation of each point.

**Pentagram-Shaped Path.** Figure 9 illustrates the planning result with 3 different methods. The robot passes through points  $P_1, P_2, \dots, P_6$  in order. For the gradient method and Newton-Raphson method, the solution of the current pose would be used as the initial value for the next pose. For our solver, with multiple solutions obtained at each point, the one with minimal cost is chosen greedily. The cost is calculated by summing up the squared differences of model parameters.

**Comparison on Cost and Efficiency.** We repeat the experiment with 10 random seeds, and the error bars represent the 95% Jeffreys confidence intervals [33]. The total cost of planning varies from different seeds, revealing the degree of dependency on initial values. Hence the 0 deviation of cost reports that our algorithm is deterministic. The solver reduces about 10% of the total cost relative to the gradient method. In terms of efficiency, the runtime is reduced by about 73% compared with the Newton-Raphson method. Numerical methods do perform better when the previous solution is used as the initial value rather than a random one. Nevertheless, the smaller deviation of runtime reflects that our solver is more stable.

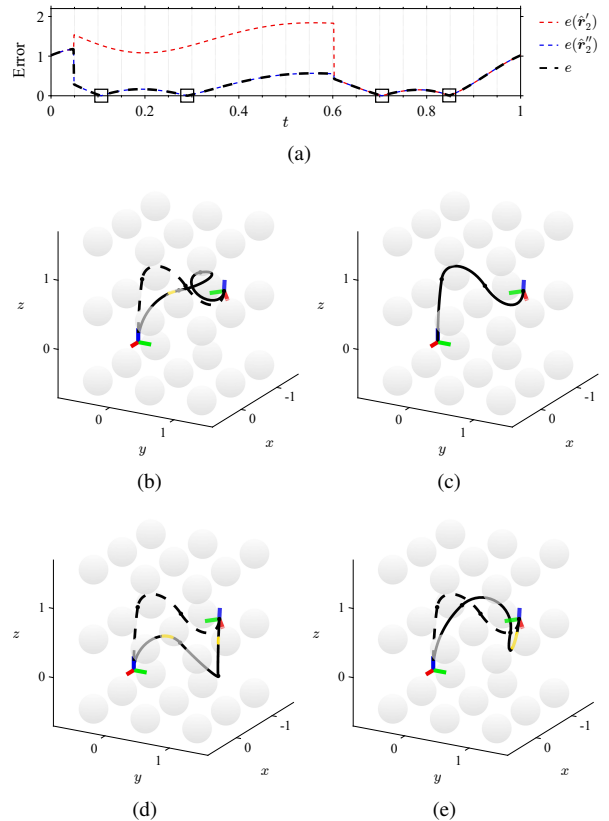


Fig. 8. We choose an end rotation  $q = \cos(\alpha/2) + \sin(\alpha/2)\omega$  and translation  $r = (-0.4 \ 1.1 \ 0.8)^T$ , where  $\alpha = 15\pi/16$ ,  $\omega = 0.48i + 0.1\sqrt{3}j - 0.86k$ . With our solver, we obtain (a) the error curve with a total of 4 equilibria, which are marked in boxes; (b-e) multiple solutions corresponding to equilibria from left to right. Collisions are painted yellow.

**Optimal RSS-Shaped Path.** We construct a sequence with 100 via points that form the shape of RSS. The orientations are all identical, whose  $x$ -axis points vertically downwards and is then rotated by  $-\pi/4$  radians. A spherical obstacle with a diameter of 0.8 is placed by the side of the path. While the solver calculates the solutions at each via point, we discard those that collide with the obstacle. The cost is the same as defined in the previous path planning experiment. By applying dynamic programming, we achieve the *optimal* path as shown in Figure 1. For clarity, we plot the solutions every 3 points.

## VI. CONCLUSION

In this paper, we present an efficient multi-solution solver to deal with the inverse kinematics problem of 3-section constant-curvature robots. Our theory simplifies the problem to an equivalent one-dimensional problem and gives an error bound for the following approximation. Furthermore, a circular traversal is applied to the approximated problem, which brings the resolution completeness, then through a few steps of iterations, we arrive at the solution of the inverse kinematics problem. The experimental results demonstrate a better efficiency and higher success rate of our solver than benchmark approaches when one solution is required. In addition, the



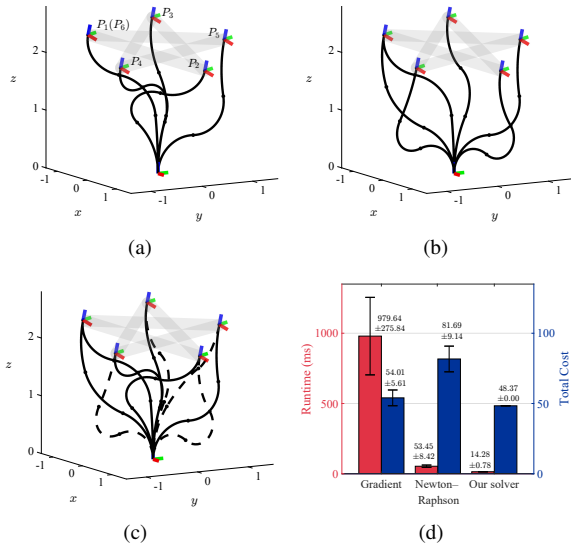


Fig. 9. Planning Results of a pentagram-shaped path ( $P_1, P_2, \dots, P_6$ ) using (a) the gradient method, (b) the Newton-Raphson method, (c) our solver. The dashed lines represent the multiple solutions. Using different random seeds, the runtime and total cost are summarised in (d).

ability of multi-solution finding allows optimal path planning in a space with obstacles.

#### ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (2021ZD0114500), the National Natural Science Foundation of China (T2293724, 62173293), Key R&D Program of Zhejiang (2022C01022), Zhejiang Provincial Natural Science Foundation of China (LD22E050007), Fundamental Research Funds for the Zhejiang Provincial Universities (2021XZZX021).

#### REFERENCES

- [1] Ernar Amanov, Thien-Dang Nguyen, and Jessica Burgner-Kahrs. Tendon-driven continuum robots with extensible sections—a model-based evaluation of path-following motions. *The International Journal of Robotics Research*, 40(1):7–23, 2021.
- [2] Long Wang and Nabil Simaan. Geometric calibration of continuum robots: Joint space and equilibrium shape deviations. *IEEE Transactions on Robotics*, 35(2):387–402, 2019.
- [3] Jinglin Li and Jing Xiao. Progressive planning of continuum grasping in cluttered space. *IEEE Transactions on Robotics*, 32(3):707–716, 2016.
- [4] Farshid Alambeigi, Sahba Aghajani Pedram, Jason L. Speyer, Jacob Rosen, Iulian Iordachita, Russell H. Taylor, and Mehran Armand. Scade: Simultaneous sensor calibration and deformation estimation of fbg-equipped unmodeled continuum manipulators. *IEEE Transactions on Robotics*, 36(1):222–239, 2020.
- [5] Andrea Bajo and Nabil Simaan. Hybrid motion/force control of multi-backbone continuum robots. *The Inter-*

- national Journal of Robotics Research*, 35(4):422–434, 2016.
- [6] Aditya Brij Koolwal, Federico Barbagli, Christopher Carlson, and David Liang. An ultrasound-based localization algorithm for catheter ablation guidance in the left atrium. *The International Journal of Robotics Research*, 29(6):643–665, 2010.
- [7] Jessica Burgner-Kahrs, D Caleb Rucker, and Howie Choset. Continuum robots for medical applications: A survey. *IEEE Transactions on Robotics*, 31(6):1261–1280, 2015.
- [8] Robert K Katzschmann, Andrew D Marchese, and Daniela Rus. Autonomous object manipulation using a soft planar grasping manipulator. *Soft Robotics*, 2(4): 155–164, 2015.
- [9] David Alatorre, Dragos Axinte, and Amir Rabani. Continuum robot proprioception: The ionic liquid approach. *IEEE Transactions on Robotics*, 38(1):526–535, 2022.
- [10] Phanideep S. Gonthina, Michael B. Wooten, Isuru S. Godage, and Ian D. Walker. Mechanics for tendon actuated multisection continuum arms. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3896–3902, 2020.
- [11] Vincent Duintam; Ron Alterovitz; Shankar Sastry; Ken Goldberg;. Three-dimensional motion planning algorithms for steerable needles using inverse kinematics. *The International Journal of Robotics Research*, 29(7):789–800, 2010.
- [12] Daniela Rus and Michael T Tolley. Design, fabrication and control of soft robots. *Nature*, 521(7553):467–475, 2015.
- [13] Jiewen Lai, Kaicheng Huang, Bo Lu, Qingxiang Zhao, and Henry K Chu. Verticalized-tip trajectory tracking of a 3d-printable soft continuum robot: Enabling surgical blood suction automation. *IEEE/ASME Transactions on Mechatronics*, 27(3):1545–1556, 2022.
- [14] Weiming Ba, Xin Dong, Abdelkhalick Mohammad, Mingfeng Wang, Dragos Axinte, and Andy Norton. Design and validation of a novel fuzzy-logic-based static feedback controller for tendon-driven continuum robots. *IEEE/ASME Transactions on Mechatronics*, 26(6):3010–3021, 2021.
- [15] Srinivas Neppalli, Matthew A. Csencsits, Bryan A. Jones, and Ian D. Walker. Closed-form inverse kinematics for continuum manipulators. *Advanced Robotics*, 23(15): 2077–2091, 2009.
- [16] Arnau Garriga-Casanovas and Ferdinando Rodriguez y Baena. Kinematics of continuum robots with constant curvature bending and extension capabilities. *Journal of Mechanisms and Robotics*, 11(1):011010, 2019.
- [17] Inderjeet Singh, Othman Lakhali, Yacine Amara, Vincent Coelen, Pushparaj Mani Pathak, and Rochdi Merzouki. Performances evaluation of inverse kinematic models of a compact bionic handling assistant. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 264–269, 2017.

- [18] Kevin M Lynch and Frank C Park. *Modern robotics*. Cambridge University Press, 2017.
- [19] Arthur W Mahoney, Trevor L Bruns, Philip J Swaney, and Robert J Webster. On the inseparable nature of sensor selection, sensor placement, and state estimation for continuum robots or “where to put your sensors and how to use them”. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4472–4478. IEEE, 2016.
- [20] Sven Lilge, Timothy D. Barfoot, and Jessica Burgner-Kahrs. Continuum robot state estimation using gaussian process regression on se(3). *The International Journal of Robotics Research*, 41(13-14):1099–1120, 2022.
- [21] Robert J Webster III and Bryan A Jones. Design and kinematic modeling of constant curvature continuum robots: A review. *The International Journal of Robotics Research*, 29(13):1661–1683, 2010.
- [22] Frederic Boyer, Vincent Lebastard, Fabien Candelier, and Federico Renda. Dynamics of continuum and soft robots: A strain parameterization based approach. *IEEE Transactions on Robotics*, 37(3):847–863, 2021.
- [23] Federico Campisano, Simone Caló, Andria A Ramirez, James H Chandler, Keith L Obstein, Robert J Webster III, and Pietro Valdastri. Closed-loop control of soft continuum manipulators under tip follower actuation. *The International Journal of Robotics Research*, 40(6-7):923–938, 2021.
- [24] John Till, Vincent Aloï, Katherine E. Riojas, Patrick L. Anderson, Robert James Webster III, and Caleb Rucker. A dynamic model for concentric tube robots. *IEEE Transactions on Robotics*, 36(6):1704–1718, 2020.
- [25] Kaitlin Oliver-Butler, Jake A. Childs, Adam Daniel, and D. Caleb Rucker. Concentric push-pull robots: Planar modeling and design. *IEEE Transactions on Robotics*, 38(2):1186–1200, 2022.
- [26] Federico Renda, Frédéric Boyer, Jorge Dias, and Lakmal Seneviratne. Discrete cosserat approach for multisection soft manipulator dynamics. *IEEE Transactions on Robotics*, 34(6):1518–1533, 2018.
- [27] Isuru S. Godage, Emanuele Guglielmino, David T. Branson, Gustavo A. Medrano-Cerda, and Darwin G. Caldwell. Novel modal approach for kinematics of multisection continuum arms. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1093–1098, 2011.
- [28] Brian C Hall. *Lie groups, Lie algebras, and representations*. Springer, 2013.
- [29] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.
- [30] Jeffrey C Lagarias, James A Reeds, Margaret H Wright, and Paul E Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal on optimization*, 9(1):112–147, 1998.
- [31] Cosimo Della Santina, Antonio Bicchi, and Daniela Rus. On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control. *IEEE Robotics and Automation Letters*, 5(2): 1001–1008, 2020. doi: 10.1109/LRA.2020.2967269.
- [32] Gregory S Chirikjian. *Stochastic models, information theory, and Lie groups, volume 2: Analytic methods and modern applications*. Springer Science & Business Media, 2011.
- [33] T Tony Cai. One-sided confidence intervals in discrete distributions. *Journal of Statistical planning and inference*, 131(1):63–88, 2005.
- [34] Joan Sola. Quaternion kinematics for the error-state kalman filter. *arXiv preprint arXiv:1711.02508*, 2017.
- [35] Timothy Barfoot, James R Forbes, and Paul T Furgale. Pose estimation using linearized rotations and quaternion algebra. *Acta Astronautica*, 68(1-2):101–112, 2011.

APPENDIX

A. A Catalog of Common Formulae

A unit quaternion  $q$  is written as a sum of real and imaginary parts in (1), but this kind of notation is not always convenient in our proof. Equivalently, we can also represent the same quaternion  $q$  in the form of a column vector, namely,

$$q = \begin{pmatrix} \delta \\ \boldsymbol{\varepsilon} \end{pmatrix}, \quad \delta \in \mathbb{R}, \quad \boldsymbol{\varepsilon} \in \mathbb{R}^3, \quad \sqrt{\delta^2 + \|\boldsymbol{\varepsilon}\|^2} = 1. \quad (45)$$

Then the conjugate of  $q$  becomes

$$q^* = \begin{pmatrix} \delta \\ -\boldsymbol{\varepsilon} \end{pmatrix}. \quad (46)$$

By definition, we compute the multiplication of two quaternions  $q_1$  and  $q_2$  as

$$q_1 \otimes q_2 = \begin{pmatrix} \delta_1 \delta_2 - \boldsymbol{\varepsilon}_1^T \boldsymbol{\varepsilon}_2 \\ \delta_1 \boldsymbol{\varepsilon}_2 + \delta_2 \boldsymbol{\varepsilon}_1 + \boldsymbol{\varepsilon}_1^\wedge \boldsymbol{\varepsilon}_2 \end{pmatrix}, \quad (47)$$

where the wedge operator  $(\cdot)^\wedge$  produces the usual skew-symmetric matrix,

$$\boldsymbol{\varepsilon}^\wedge = \begin{pmatrix} 0 & -\varepsilon_3 & \varepsilon_2 \\ \varepsilon_3 & 0 & -\varepsilon_1 \\ -\varepsilon_2 & \varepsilon_1 & 0 \end{pmatrix} \quad (48)$$

Since the quaternion multiplication is non-commutative, we introduce the left and right compound operators [34, 35] by defining  $(\cdot)^+ : \mathbb{H} \rightarrow \mathbb{R}^{4 \times 4}$  and  $(\cdot)^\oplus : \mathbb{H} \rightarrow \mathbb{R}^{4 \times 4}$ ,

$$q_1^+ = \begin{pmatrix} \delta_1 & -\boldsymbol{\varepsilon}_1^T \\ \boldsymbol{\varepsilon}_1 & \delta_1 \mathbf{I} + \boldsymbol{\varepsilon}_1^\wedge \end{pmatrix}, \quad q_2^\oplus = \begin{pmatrix} \delta_2 & -\boldsymbol{\varepsilon}_2^T \\ \boldsymbol{\varepsilon}_2 & \delta_2 \mathbf{I} - \boldsymbol{\varepsilon}_2^\wedge \end{pmatrix}, \quad (49)$$

so that the quaternion multiplication can be expressed in matrix notations, i.e.,

$$q_1 \otimes q_2 = q_1^+ q_2 = q_2^\oplus q_1. \quad (50)$$

It can be extended to a vector  $\mathbf{r} \in \mathbb{R}^3$  if it is written in the form of a pure quaternion  $\mathbf{r} = r_x \mathbf{i} + r_y \mathbf{j} + r_z \mathbf{k}$ , then the following product represents the rotation of  $\mathbf{r}$ , namely,

$$q \otimes \mathbf{r} \otimes q^* = q^+ (q^*)^\oplus \mathbf{r} = \left( \boldsymbol{\varepsilon} \boldsymbol{\varepsilon}^T + (\delta \mathbf{I} + \boldsymbol{\varepsilon}^\wedge)^2 \right) \mathbf{r}. \quad (51)$$

B. Proof of Lemma 1

*Proof:* We first reorder (15) as  $q_2 = q_1^* \otimes q$  since  $q_1^{-1} = q_1^*$  for unit quaternions, then apply the matrix notation to get

$$q_2 = q^\oplus q_1^*, \quad (52)$$

or

$$\begin{pmatrix} a_2 \\ b_2 \\ c_2 \\ d_2 \end{pmatrix} = \begin{pmatrix} a & -b & -c & -d \\ b & a & d & -c \\ c & -d & a & b \\ d & c & -b & a \end{pmatrix} \begin{pmatrix} a_1 \\ -b_1 \\ -c_1 \\ -d_1 \end{pmatrix}. \quad (53)$$

It has already been determined that  $d_\lambda = 0$ . So we separate the last row of (53) from another three to get two independent constraints. By rearranging the elements in the order of  $(c_\lambda \ -b_\lambda \ a_\lambda)^T = \hat{\mathbf{r}}_\lambda$ , we obtain an identity of  $\hat{\mathbf{r}}_1$  from the last row,

$$(b \ c \ d) \hat{\mathbf{r}}_1 = 0,$$

as well as a relationship of  $\hat{\mathbf{r}}_1$  and  $\hat{\mathbf{r}}_2$  from another three rows,

$$\mathbf{A} \hat{\mathbf{r}}_1 = \hat{\mathbf{r}}_2.$$

This shows the necessity.

Geometrically, (17) refers to a plane through the origin in the space  $\mathbb{R}^3$  with a normal vector  $\mathbf{n} = (b \ c \ d)^T$ . This plane intersects the unit sphere in the space since  $\hat{\mathbf{r}}_1 \in \mathbb{S}^2 \subseteq \mathbb{R}^3$ . The intersection is simply a circle

$$C(\mathbf{n}) = \{\mathbf{v} \in \mathbb{S}^2 : \mathbf{n} \cdot \mathbf{v} = 0\}. \quad (54)$$

Next we show the sufficiency to complete this proof. The proof of necessity is invertible, so we only need to verify that the solution is feasible by proving  $\hat{\mathbf{r}}_2 \in \mathbf{AC}(\mathbf{n}) \subseteq \mathbb{S}^2$ . Let  $\hat{\mathbf{r}}_1 = (c_1 \ -b_1 \ a_1)^T \in C(\mathbf{n})$ , then

$$\begin{aligned} \|\mathbf{A} \hat{\mathbf{r}}_1\|^2 &= \hat{\mathbf{r}}_1^T \mathbf{A}^T \mathbf{A} \hat{\mathbf{r}}_1 \\ &= (1 - b^2) c_1^2 + 2bc b_1 c_1 \\ &\quad + (1 - c^2) b_1^2 + 2cd a_1 b_1 \\ &\quad + (1 - d^2) a_1^2 - 2bd a_1 c_1 \\ &= c_1^2 + b_1^2 + a_1^2 - (bc_1 - cb_1 + da_1)^2. \end{aligned} \quad (55)$$

Since  $\hat{\mathbf{r}}_1 \in C(\mathbf{n})$ , we have  $\|\hat{\mathbf{r}}_1\| = \sqrt{c_1^2 + b_1^2 + a_1^2} = 1$  and  $\mathbf{n} \cdot \hat{\mathbf{r}}_1 = bc_1 - cb_1 + da_1 = 0$ , therefore

$$\|\hat{\mathbf{r}}_2\| = \|\mathbf{A} \hat{\mathbf{r}}_1\| = 1.$$

This shows the sufficiency. ■

C. Corollary

We observe a consequence after proving Lemma 1.

**Corollary 1.** *Given an end rotation  $q = (a \ b \ c \ d)^T$ , for those 2-section continuum robots with identical end rotation, their endpoints fall on the plane passing through the origin and with a normal vector  $\mathbf{n} = (b \ c \ d)^T$ , i.e.,*

$$\mathbf{n} \cdot \mathbf{r} = 0. \quad (56)$$

*Proof:* Rewrite (16) as

$$\begin{aligned} \mathbf{r} &= \rho(a_1, L_1) \hat{\mathbf{r}}_1 + \rho(a_2, L_2) q_1 \otimes \hat{\mathbf{r}}_2 \otimes q_1^* \\ &= \rho(a_1, L_1) \hat{\mathbf{r}}_1 + \rho(a_2, L_2) q_1^+ (q_1^*)^\oplus \hat{\mathbf{r}}_2. \end{aligned} \quad (57)$$

We have known that the 2-section continuum robots with identical end rotation must all meet the perpendicular constraint  $\mathbf{n} \cdot \hat{\mathbf{r}}_1 = 0$  referred to in (17). So the first term on the right side of (57) is perpendicular to  $\mathbf{n}$ . So does the second term because if we consider the inner product  $\mathbf{n} \cdot q_1^+ (q_1^*)^\oplus \hat{\mathbf{r}}_2$ , then plug (18) into this expression and it becomes

$$\begin{aligned} &\mathbf{n} \cdot q_1^+ (q_1^*)^\oplus \mathbf{A} \hat{\mathbf{r}}_1 \\ &= ada_1 - abc_1 + acb_1 + 2bdb_1 + 2cdc_1 \\ &\quad + 2(abc_1 - acb_1 - bdb_1 - cdc_1) (a_1^2 + b_1^2 + c_1^2) \\ &= a(da_1 + bc_1 - cb_1). \end{aligned} \quad (58)$$

As we have encountered in the proof of Lemma 1, we get  $\mathbf{n} \cdot q_1^+ (q_1^*)^\oplus \hat{\mathbf{r}}_2 = a \cdot \mathbf{n}^T \hat{\mathbf{r}}_1 = 0$ . This proves  $\mathbf{n} \cdot \mathbf{r} = 0$ . Hence, the endpoint falls on that given plane. ■

#### D. Proof of Lemma 2

*Proof:* Let  $\mathbf{w}_2 = (-c_2 \ b_2 \ a_2)^T$ , then by using matrix notations (49) and (3), we find that

$$\begin{aligned} & q_1^+ (q_1^*)^\oplus \hat{\mathbf{r}}_2 \\ &= \begin{pmatrix} 2c_1^2 - 1 & -2b_1c_1 & 2a_1c_1 \\ -2b_1c_1 & 2b_1^2 - 1 & -2a_1b_1 \\ 2a_1c_1 & -2a_1b_1 & 2a_1^2 - 1 \end{pmatrix} \begin{pmatrix} -c_2 \\ b_2 \\ a_2 \end{pmatrix} \\ &= (2\hat{\mathbf{r}}_1\hat{\mathbf{r}}_1^T - \mathbf{I})\mathbf{w}_2. \end{aligned} \quad (59)$$

For the last equality above, we use the fact that  $\hat{\mathbf{r}}_1^T\hat{\mathbf{r}}_1 = \|\mathbf{r}_1\|^2 = 1$ . Combine (57) and (59) to get

$$\mathbf{r} - \rho(a_1, L_1)\hat{\mathbf{r}}_1 = \rho(a_2, L_2) \left( 2\hat{\mathbf{r}}_1\hat{\mathbf{r}}_1^T - \mathbf{I} \right) \mathbf{w}_2. \quad (60)$$

We also notice the matrix  $2\hat{\mathbf{r}}_1\hat{\mathbf{r}}_1^T - \mathbf{I}$  is symmetric and involutory, namely,

$$\left( 2\hat{\mathbf{r}}_1\hat{\mathbf{r}}_1^T - \mathbf{I} \right)^T = \left( 2\hat{\mathbf{r}}_1\hat{\mathbf{r}}_1^T - \mathbf{I} \right), \left( 2\hat{\mathbf{r}}_1\hat{\mathbf{r}}_1^T - \mathbf{I} \right)^2 = \mathbf{I}.$$

So if we multiply (60) by  $2\hat{\mathbf{r}}_1\hat{\mathbf{r}}_1^T - \mathbf{I}$  and then denote  $\mathbf{v} = \mathbf{r} - \rho(a_1, L_1)\hat{\mathbf{r}}_1$ , we can get

$$\left( 2\hat{\mathbf{r}}_1\hat{\mathbf{r}}_1^T - \mathbf{I} \right) \mathbf{v} = \rho(a_2, L_2) \mathbf{w}_2. \quad (61)$$

This equation is equivalent to (16) because the derivation above can be inverted. Note that the left side only involves variables of the first arc section and the right side only of the second. To make it solvable, a necessary condition is the norm of both sides are equal. According to the definition of  $\mathbf{w}_2$ , the norm of  $\mathbf{w}_2$  equals to the norm of  $\mathbf{r}_2$ , which is exactly 1, so we must have

$$\|\mathbf{v}\| = \rho(a_2, L_2). \quad (62)$$

Now  $a_2$  can be calculated from the last row (the third dimension) of (61) by

$$a_2 = \frac{\mathbf{u} \cdot \mathbf{v}}{\rho(a_2, L_2)} = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{v}\|}, \quad (63)$$

where  $\mathbf{u} = (2a_1c_1 \ -2a_1b_1 \ 2a_1^2 - 1)^T$  is the transpose of the third row of the matrix  $2\hat{\mathbf{r}}_1\hat{\mathbf{r}}_1^T - \mathbf{I}$ , as defined in (26). So the inner product  $\mathbf{u} \cdot \mathbf{v}$  extracts the last component of the vector of the left part of (61).

Substitute (63) into (62) yields (21). Substitute (62) into (61) and then we can calculate  $\mathbf{w}_2$  by

$$\mathbf{w}_2 = (2\mathbf{r}_1\mathbf{r}_1^T - \mathbf{I}) \frac{\mathbf{v}}{\|\mathbf{v}\|}, \quad (64)$$

so  $\hat{\mathbf{r}}_2$  can be recovered from  $\mathbf{w}_2$  using elementary operations. This shows (24). ■

#### E. Proof of Lemma 3

We list some identities before we turn to the proof.

**Proposition 1.** Let  $\mathbf{B}$  be defined as (26), and denote  $\mathbf{m} = (c \ -b \ a)^T$ , then for  $\lambda = 1, 2, 3$ , we have

- (a)  $\hat{\mathbf{r}}_\lambda^T \mathbf{B} \hat{\mathbf{r}}_\lambda = \hat{\mathbf{r}}_\lambda^T \mathbf{B}^T \hat{\mathbf{r}}_\lambda = d.$
- (b)  $\hat{\mathbf{r}}_\lambda^T (\mathbf{B}^T \mathbf{B} + \mathbf{m} \mathbf{m}^T) \hat{\mathbf{r}}_\lambda = 1.$

*Proof:* The proof of the identities is straightforward using the fact that  $q$  is a unit quaternion and  $\hat{\mathbf{r}}_\lambda$  is a unit vector. (a) Expand  $\hat{\mathbf{r}}_\lambda^T \mathbf{B} \hat{\mathbf{r}}_\lambda$  and we have  $\hat{\mathbf{r}}_\lambda^T \mathbf{B} \hat{\mathbf{r}}_\lambda = d(a_\lambda^2 + b_\lambda^2 + c_\lambda^2) = d$ . Since  $\mathbf{B} - \mathbf{B}^T$  is antisymmetric, we have  $\hat{\mathbf{r}}_\lambda^T (\mathbf{B} - \mathbf{B}^T) \hat{\mathbf{r}}_\lambda = 0$ , so  $\hat{\mathbf{r}}_\lambda^T \mathbf{B} \hat{\mathbf{r}}_\lambda = \hat{\mathbf{r}}_\lambda^T \mathbf{B}^T \hat{\mathbf{r}}_\lambda = d$ .

(b) Expand the left and we have  $\hat{\mathbf{r}}_\lambda^T (\mathbf{B}^T \mathbf{B} + \mathbf{m} \mathbf{m}^T) \hat{\mathbf{r}}_\lambda = (a^2 + b^2 + c^2 + d^2)(a_\lambda^2 + b_\lambda^2 + c_\lambda^2) = 1 \cdot 1 = 1$ . ■

Knowing this, the proof of Lemma 3 can be completed.

*Proof of Lemma 3:* As mentioned, we partition the three sections into two groups. There are only two ways to make the partition, namely,  $\{1, 2\} \cup \{3\}$  and  $\{1\} \cup \{2, 3\}$ . We will see soon that the former leads to the case  $\lambda = 3$ , and the latter to the case  $\lambda = 1$ . First we show the case  $\lambda = 3$ . Let

$$q_e = q_1 \otimes q_2 = q \otimes q_3^{-1} = q \otimes q_3^*, \quad (65)$$

then through basic matrix multiplications we find

$$q_e = q^+ q_3^* = \begin{pmatrix} \delta\delta_3 + \varepsilon^T \varepsilon_3 \\ \delta_3 \varepsilon - \delta \varepsilon_3 - \varepsilon^\wedge \varepsilon_3 \end{pmatrix} = \begin{pmatrix} \mathbf{m}^T \hat{\mathbf{r}}_3 \\ \mathbf{B} \hat{\mathbf{r}}_3 \end{pmatrix}. \quad (66)$$

Here we keep adopting the symbols  $\mathbf{m}$  and  $\mathbf{B}$  defined in Proposition 1. According to Corollary 1, we know that the endpoint of the first two arc sections lies on the plane passing the origin with a normal vector  $\mathbf{n}_e = (b_e \ c_e \ d_e)^T = \mathbf{B} \hat{\mathbf{r}}_3$ . Denote

$$\mathbf{r}_e = \mathbf{r} - q_e \otimes \mathbf{r}_3 \otimes q_e^*, \quad (67)$$

then Lemma 1 tells us that

$$\mathbf{n}_e \cdot \mathbf{r}_e = 0. \quad (68)$$

Replace  $\mathbf{r}_e$  in (68) with (67) we get

$$\mathbf{r}^T \mathbf{B} \hat{\mathbf{r}}_3 - \rho(a_3, L_3) \mathbf{n}_e \cdot q_e \otimes \mathbf{r}_3 \otimes q_e^* = 0, \quad (69)$$

which is close to the result we want. It suffices to show that

$$\mathbf{n}_e \cdot q_e \otimes \mathbf{r}_3 \otimes q_e^* = d. \quad (70)$$

Note that

$$\begin{aligned} & q_e \otimes \mathbf{r}_3 \otimes q_e^* \\ &= (\mathbf{B} \hat{\mathbf{r}}_3) (\mathbf{B} \hat{\mathbf{r}}_3)^T \hat{\mathbf{r}}_3 + (\mathbf{m}^T \hat{\mathbf{r}}_3)^2 \hat{\mathbf{r}}_3 \\ &\quad + 2 (\mathbf{m}^T \hat{\mathbf{r}}_3) (\mathbf{B} \hat{\mathbf{r}}_3)^\wedge \hat{\mathbf{r}}_3 + (\mathbf{B} \hat{\mathbf{r}}_3)^\wedge ((\mathbf{B} \hat{\mathbf{r}}_3)^\wedge \hat{\mathbf{r}}_3), \end{aligned} \quad (71)$$

and therefore we have

$$\begin{aligned} & (\mathbf{B} \hat{\mathbf{r}}_3) \cdot q_e \otimes \mathbf{r}_3 \otimes q_e^* \\ &= (\mathbf{B} \hat{\mathbf{r}}_3)^T (\mathbf{B} \hat{\mathbf{r}}_3) (\hat{\mathbf{r}}_3^T \mathbf{B}^T \hat{\mathbf{r}}_3) + (\mathbf{m}^T \hat{\mathbf{r}}_3)^2 (\hat{\mathbf{r}}_3^T \mathbf{B}^T \hat{\mathbf{r}}_3) + 0 + 0 \\ &= (\hat{\mathbf{r}}_3^T \mathbf{B}^T \mathbf{B} \hat{\mathbf{r}}_3 + \hat{\mathbf{r}}_3^T \mathbf{m} \mathbf{m}^T \hat{\mathbf{r}}_3) (\hat{\mathbf{r}}_3^T \mathbf{B}^T \hat{\mathbf{r}}_3) \\ &= d \end{aligned} \quad (72)$$

The last equality follows from Proposition 1. So (25) holds for  $\lambda = 3$ .

The proof of case  $\lambda = 1$  is simpler. We use the partition  $\{1\} \cup \{2, 3\}$ . Let

$$q'_e = q_2 \otimes q_3 = q_1^{-1} \otimes q = q_1^* \otimes q, \quad (73)$$

and we can find a matrix  $\mathbf{B}'$  such that

$$q'_e = (q_1^*)^+ q = \begin{pmatrix} \delta_1 \delta + \varepsilon_1^T \varepsilon \\ \delta_1 \varepsilon - \delta \varepsilon_1 - \varepsilon_1^\wedge \varepsilon \end{pmatrix} = \begin{pmatrix} \cdot \\ \mathbf{B}' \hat{\mathbf{r}}_1 \end{pmatrix}. \quad (74)$$

Rearrange (14) and we have

$$q_1^* \otimes (\mathbf{r} - \mathbf{r}_1) \otimes q_1 = \mathbf{r}_2 + q_2 \otimes \mathbf{r}_3 \otimes q_2^*. \quad (75)$$

Again, Corollary 1 specifies a plane through the origin with a normal vector  $\mathbf{n}'_e = \mathbf{B}' \hat{\mathbf{r}}_1$ , and says the vector in  $\mathbb{R}^3$  to the left of (75) lies on that plane. If  $\mathbf{R}_1^* \in \text{SO}(3)$  is the rotation matrix of  $q_1^*$ , then we have

$$(\mathbf{B}' \hat{\mathbf{r}}_1) \cdot \mathbf{R}_1^* (\mathbf{r} - \mathbf{r}_1) = 0. \quad (76)$$

Since  $(q_1^*)^{-1} = q_1$ , we know that the corresponding rotation matrix of  $q_1$  is  $\mathbf{R}_1 = (\mathbf{R}_1^*)^{-1} = (\mathbf{R}_1^*)^T$ . So we have

$$(\mathbf{R}_1^*)^T \mathbf{B}' \hat{\mathbf{r}}_1 = \mathbf{R}_1 \mathbf{B}' \hat{\mathbf{r}}_1 = q_1 \otimes (\mathbf{B}' \hat{\mathbf{r}}_1) \otimes q_1^*. \quad (77)$$

Using (74) we find that

$$\begin{aligned} & q_1 \otimes (\mathbf{B}' \hat{\mathbf{r}}_1) \otimes q_1^* \\ &= (\varepsilon_1 \varepsilon_1^T + (\delta_1 \mathbf{I} + \varepsilon_1^\wedge)^2) (\delta_1 \varepsilon - \delta \varepsilon_1 - \varepsilon_1^\wedge \varepsilon) \\ &= \delta_1 \varepsilon - \delta \varepsilon_1 - \varepsilon_1^\wedge \varepsilon \\ &= \mathbf{B}' \hat{\mathbf{r}}_1. \end{aligned} \quad (78)$$

So (77) and (78) leads to

$$(\mathbf{R}_1^*)^T \mathbf{B}' \hat{\mathbf{r}}_1 = \mathbf{B}' \hat{\mathbf{r}}_1. \quad (79)$$

We combine (76) and (79) and apply Proposition 1(a) to deduce that

$$\begin{aligned} 0 &= (\mathbf{B}' \hat{\mathbf{r}}_1) \cdot \mathbf{R}_1^* (\mathbf{r} - \mathbf{r}_1) \\ &= \mathbf{r}^T (\mathbf{R}_1^*)^T \mathbf{B}' \hat{\mathbf{r}}_1 - \mathbf{r}_1^T (\mathbf{R}_1^*)^T \mathbf{B}' \hat{\mathbf{r}}_1 \\ &= \mathbf{r}^T \mathbf{B}' \hat{\mathbf{r}}_1 - \rho(a_1, L_1) \hat{\mathbf{r}}_1^T \mathbf{B}' \hat{\mathbf{r}}_1 \\ &= \mathbf{r}^T \mathbf{B}' \hat{\mathbf{r}}_1 - \rho(a_1, L_1) d. \end{aligned} \quad (80)$$

Therefore (25) also holds for  $\lambda = 1$ .

#### F. Proof of Theorem 2

*Proof:* Exactly,  $\hat{\mathbf{r}}_3$  satisfies the equation

$$\mathbf{n}_0 \cdot \hat{\mathbf{r}}_3 - \rho(a_3, L_3) d = 0. \quad (81)$$

Subtract this from (35) and we have the projection error

$$\begin{aligned} \left| \hat{\mathbf{n}}_0 \cdot (\hat{\mathbf{r}}_3^{(0)} - \hat{\mathbf{r}}_3) \right| &= \frac{\gamma L_3 |d| - \rho(a_3, L_3) d}{\|\mathbf{n}_0\|} \\ &= \frac{L_3 |d|}{\|\mathbf{B}^T \mathbf{r}\|} \left| \gamma - \frac{\sqrt{1 - a_3^2}}{\arccos a_3} \right|. \end{aligned} \quad (82)$$

We find that

$$\min_{\gamma} \left\{ \max_{a_3 \in [0,1]} \left\{ \left| \gamma - \frac{\sqrt{1 - a_3^2}}{\arccos a_3} \right| \right\} \right\} = \frac{1}{2} - \frac{1}{\pi} \quad (83)$$

when taking  $\gamma = \frac{1}{2} + \frac{1}{\pi}$ , so we get

$$\left| \hat{\mathbf{n}}_0 \cdot (\hat{\mathbf{r}}_3^{(0)} - \hat{\mathbf{r}}_3) \right| \leq \frac{(\pi - 2) L_3 |d|}{2\pi \|\mathbf{B}^T \mathbf{r}\|}. \quad (84)$$

Next we turn to find the bound of the approximation of  $\mathbf{r}_1$ . The exact equation is

$$\begin{cases} \mathbf{n}_e \cdot \hat{\mathbf{r}}_1 = 0, \\ \mathbf{n}_0 \cdot \hat{\mathbf{r}}_1 - \rho(a_1, L_1) d = 0, \end{cases} \quad (85)$$

where  $\mathbf{n}_e = \mathbf{B} \hat{\mathbf{r}}_3$ ,  $\mathbf{n}_0 = \mathbf{B}^T \mathbf{r}$ . Since both  $\hat{\mathbf{r}}_1^{(0)}$  and  $\hat{\mathbf{r}}_1$  are elements in  $\mathbb{S}^2$  and perpendicular with  $\mathbf{n}_e$ , we assume

$$\hat{\mathbf{r}}_1 = \exp(\mathbf{n}_e^\wedge \beta) \hat{\mathbf{r}}_1^{(0)}, \quad (86)$$

and therefore,

$$\left\| \hat{\mathbf{r}}_1^{(0)} - \hat{\mathbf{r}}_1 \right\| \leq |\beta|. \quad (87)$$

By following the pattern of proving (84), we have the similar bounded projection error for  $\hat{\mathbf{r}}_1$

$$\left| \hat{\mathbf{n}}_0 \cdot (\hat{\mathbf{r}}_1^{(0)} - \hat{\mathbf{r}}_1) \right| \leq \frac{(\pi - 2) L_1 |d|}{2\pi \|\mathbf{B}^T \mathbf{r}\|}. \quad (88)$$

According to Figure 7(c), we have

$$\begin{aligned} |\beta| &= \arccos \left( \|\gamma L_1 \mathbf{r}_0\| - \frac{(\pi - 2) L_1 |d|}{2\pi \|\mathbf{B}^T \mathbf{r}\|} \right) \\ &\quad - \arccos \min \left\{ 1, \|\gamma L_1 \mathbf{r}_0\| + \frac{(\pi - 2) L_1 |d|}{2\pi \|\mathbf{B}^T \mathbf{r}\|} \right\}. \end{aligned} \quad (89)$$

If we assume the distance of translation  $\mathbf{r}$  is larger than the length of the first section  $L_1$ , i.e.,  $\|\mathbf{r}\| > L_1$ , then we have

$$\begin{aligned} \|\mathbf{B}^T \mathbf{r}\| &= \sqrt{\|\mathbf{r}\|^2 d^2 + (\cdot)^2 + (\cdot)^2 + (\cdot)^2} \\ &\geq \|\mathbf{r}\| |d| > L_1 |d|. \end{aligned} \quad (90)$$

Thus, combining (87), (89) and (90) we know that the error of  $\hat{\mathbf{r}}_1$  is also bounded, namely,

$$\left\| \hat{\mathbf{r}}_1^{(0)} - \hat{\mathbf{r}}_1 \right\| \leq \arccos \frac{2L_1 |d|}{\pi \|\mathbf{B}^T \mathbf{r}\|} - \arccos \frac{L_1 |d|}{\|\mathbf{B}^T \mathbf{r}\|}. \quad (91)$$

#### G. Jacobian Matrix

According to [32], the left Jacobian and the right Jacobian for  $\text{SO}(3)$  are computed as

$$\begin{aligned} \mathbf{J}_l(\boldsymbol{\omega}) &= \mathbf{I} + \frac{1 - \cos \|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|^2} \boldsymbol{\omega}^\wedge + \frac{\|\boldsymbol{\omega}\| - \sin \|\boldsymbol{\omega}\|}{\|\boldsymbol{\omega}\|^3} (\boldsymbol{\omega}^\wedge)^2, \\ \mathbf{J}_r(\boldsymbol{\omega}) &= \mathbf{J}_l^T(\boldsymbol{\omega}). \end{aligned} \quad (92)$$

Hence the right Jacobian for  $\text{SE}(3)$  is

$$\mathbf{J}(\boldsymbol{\xi}) = \begin{pmatrix} \mathbf{J}_r(\boldsymbol{\omega}) & \mathbf{0}_{3 \times 3} \\ \exp(-\boldsymbol{\omega}^\wedge) \frac{\partial}{\partial \boldsymbol{\omega}} (\mathbf{J}_l(\boldsymbol{\omega}) \mathbf{v}) & \mathbf{J}_r(\boldsymbol{\omega}) \end{pmatrix}. \quad (93)$$