Robotica (2017) volume 35, pp. 224–240. © Cambridge University Press 2015
doi:10.1017/S0263574715000946

Kinematic model to control the end-effector of a continuum robot for multi-axis processing

Salvador Cobos-Guzman*, David Palmer and Dragos Axinte

University of Nottingham, Machining and Condition Monitoring Research Group, University Park, Nottingham, NG7 2RD, UK.

(Accepted October 20, 2015. First published online: November 24, 2015)

SUMMARY

This paper presents a novel kinematic approach for controlling the end-effector of a continuum robot for *in-situ* repair/inspection in restricted and hazardous environments. Forward and inverse kinematic (IK) models have been developed to control the last segment of the continuum robot for performing multi-axis processing tasks using the last six Degrees of Freedom (DoF). The forward kinematics (FK) is proposed using a combination of Euler angle representation and homogeneous matrices. Due to the redundancy of the system, different constraints are proposed to solve the IK for different cases; therefore, the IK model is solved for bending and direction angles between ($-\pi/2$ to $+\pi/2$) radians. In addition, a novel method to calculate the Jacobian matrix is proposed for this type of hyper-redundant kinematics. The error between the results calculated using the proposed Jacobian algorithm and using the partial derivative equations of the FK map (with respect to linear and angular velocity) is evaluated. The error between the two models is found to be insignificant, thus, the Jacobian is validated as a method of calculating the IK for six DoF.

KEYWORDS: Continuous robots; Inverse kinematic; Kinematics model; Jacobian for hyper-redundant systems.

1. Introduction

Inside the human body and some animals, there are specific muscular organs responsible for controlling the movements, manipulations and different types of locomotion in a continuum bendable form. These muscles are called muscular hydrostats[1,2] such as mammalian tongues, elephant trunks, cephalopod tentacles and so on. Hydrostatic skeletons do not possess rigid elements but the stiffness is created by internal pressure in the muscles. Other animals such as snakes, worms and slugs are considered as hyper-redundant morphologies. In snake animals, the locomotion is produced by the lateral vertebral flexion that generates the adequate propulsive force for three types of locomotion: lateral undulation, concertina and sidewinding.[3] These types of locomotion have inspired some researchers in the design of snake robots.[4]

The biomechanical behaviour of the previous biological examples represents a complex redundant kinematic behaviour due to the 'hyper-redundancy' of continuous rotations in the same plane of movement. The term 'hyper-redundancy' was introduced by Chirikjian and Burdick,[5,6] and refers 'to have a very large or infinite redundant degrees of freedom'. The first hyper-redundant manipulator (tensor arm) was designed in the late of 1960s by Anderson and Horn.[7] Since then, numerous hyper-redundant robots have been designed based on biological inspiration.

The elephant's trunk robot[8] is designed as a segmented backbone with a total of 32 DoF; the actuation is provided via series of tendons routed through the structure. Another example of an elephant's trunk robot is the bionic handling assistant (BHA),[9] a flexible continuum manipulator based on pneumatic actuators that uses three sections of three DoF each. The octopus tentacle was

* Corresponding author. E-mail: Salvador.Cobos_Guzman@nottingham.ac.uk, epxdp1@nottingham.ac.uk, dragos.axinte@nottingham.ac.uk

the inspiration behind the design of the Tentacle Manipulator,[10] a highly flexible robot consisting of an elastic backbone actuated with four DoF. A snake inspired manipulator from OC Robotics (OCR) is used for nuclear inspection and maintenance;[11] comprised of a number of sub-sections, the arm is driven by a set of actuators that pulls the wires to control the bend angles of the snake arm. Another OCR demonstrator consists of five sections and 10 DoF. An additional example of a snake-like manipulator[12] is controlled by three linear actuators and provides tunable-stiffness by a layer jamming mechanism.

For constrained environments, a hyper-redundant miniature manipulator driven by high-strain shape memory alloy (SMA) was presented in ref. [13], composed of four identical modules with a total of 12 DoF. While for medical applications, the Active Cannulas (Re-curved Tube)[14] is composed of super-elastic tubes; this device is very small with diameters of (0.8 mm–2.39 mm), controlled by a simple manual actuation of six DoF.

This paper describes a novel method for controlling the end-effector of a Hyper-Redundant Snake Arm (HRSA) based on a proposed kinematic model. The difference between several IK and FK approaches for hyper-redundant manipulators is described below.

In reference to the kinematic calculations for snake arms, the Compact Formulation[15] is a method of solving the IK of hyper-redundant robots. Compact Formulation is derived by applying Gaussian elimination to transform the underspecified matrix into a row-reduced echelon form. This method uses the pseudo-inverse formulation to solve an underspecified set of linear equations and therefore leads to longer computational times, depending on the number of variables that the algorithm has to process.

The spline solution method[16] is based on cubic and quartic splines in order to perform a parameterization of the backbone curve that allows for the inclusion of the constraints of the end-effector pose into the parametric equations of the backbone curve; this method requires parallel processing techniques that require excessive computational resources from the controller. A faster technique using the Recursive Fitting Method is proposed[17] to solve nonlinear algebraic equations that guarantee the existence of solutions for the IK problem at the velocity level; here, the backbone curve is approximated with a number of lines of specified lengths, thus causing small errors in the end-effector depending on the number of segments and DoF.

A continuous-variable-based optimization method for the IK of binary manipulators was presented in ref. [18]. In this optimization, the distance between the end-effector and the target is constrained, leading to solutions of low accuracy. A geometric method for solving the equations pertaining to a planar hyper-redundant manipulator has been proposed by;[19] however, this approach fails for backbones with more than one DoF in each disc, while some solutions result in errors due to the redundancy in the system. Also, closed form solutions for continuum robots composed of a continuously bendable backbone have been proposed;[20] however, the method is presented as a single-section solution and no information was provided about the errors produced in multiple sections for desired vs. calculated trajectories.

Another approach is to express the kinematics as a constrained optimization problem, minimizing the position error and simultaneously avoiding any collision of the manipulator with either the obstacles or within its links;[21] this technique is based on the classical augmented Lagrangian method for solving the resulting constrained optimization problem. Similarly, the Sequential Quadratic Programming (SQP) algorithm[22] can be used to solve the IK, minimizing a function with local kinematic constrains.

In ref. [23], the Product-Of-Exponentials (POE) formula[24,31] and the Lie Bracket method from group theory is employed to derive the instantaneous kinematic model while the IK are solved using the Newton–Raphson numeric method. Also, screw theory was proposed for a modular spatial hyper-redundant manipulator[25] to analyse the kinematics, velocity and acceleration; while in ref. [26], the closed-loop IK algorithm is based on the Jacobian pseudo-inverse and Jacobian transpose.

The main contribution of this paper is the provision of the forward and IK models between $\pm \pi/2$ rad for two, four and six DoF respectively, to enable the use of continuum robots for multi-axis machining processes in restricted/hazardous environments; these algorithms allow for the generation of tool path trajectories for the end-effector.

The proposed kinematics of the HRSA robot is composed of 24 DoF in total, divided into four stages, each of which is controlled in six DoF. The objective is to move the robot in a specific area for a machining process and, in particular, to control the last segment where the end-effector is located.

https://doi.org/10.1017/S0263574715000946 Published online by Cambridge University Press

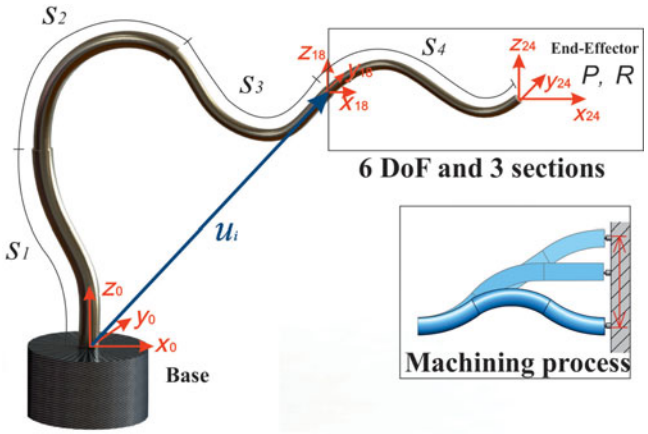

Fig. 1. Kinematic representation of the HRSA.

In this way, the previous segments are fixed and only the last segment needs to be controlled to enable multi-axis machining. Thus, the end-effector, mounted at the extremity of the last segment, can be controlled with dexterity and precision in order to perform complex *in-situ* inspections/machining tasks in industrial applications reducing the direct exposure of the human operator.

The FK model is presented using a combination of multiplication of homogenous transformation ($4\times$ 4) matrices and an original method to eliminate important singularity points. In this case, the methodology of Hartenberg and Denavit[27] is not useful for the type of joint configuration that describes the arc of each section produced by the bend angle of the continuum robot. The combination of matrices proposed provides the kinematic behaviour of the compliant joints in the flexible backbone construction; the FK eliminates any twist motion around the central axis of each joint.

A novel method for solving the IK is presented only for the last six DoF. This problem was addressed using analytical equations and constraints for bending angles. The range of movement of each DoF is from $-\pi/2$ to $+\pi/2$ radians; the solutions of the IK are in a closed form for this range of movement. The Jacobian of the last six DoF is designed based on a novel algorithm for the specific transformations in the FK. In this case, a Jacobian algorithm with the dimensions ($6\times$ 6) is proposed for the last section. This algorithm proves that this methodology can be used for real-time applications of similar hyper-redundant systems.

2. Kinematic Description

The full HRSA is divided in four stages (S_i), each with a different diameter (Fig. 1) and containing three sections $S_{i,k} \in \Re^+$; where $i = \{1, 2, 3, 4\}$ is the number of stages and $k = \{1, 2, 3\}$ is the number of sections. Each section is afforded two DoF by means of a compliant joint construction such as it is described in ref. [32] (these types of joints are not universal joints). Thus, each stage contains six DoF and the entire HRSA has 24 DoF in total. The kinematic analysis is based on the behaviour of a backbone curvature structure.

Vector u_i denotes the position of the frame of reference (x_{18}, y_{18}, z_{18}) with respect to the frame of reference of the base (x_0, y_0, z_0); note that the six DoF of the stage 4 (S_4) are located in frames of refs. [18]–[24]. In this paper, only the last stage S_4 will be used for machining tasks because moving the lower sections ($S_1 - S_3$) may induce vibrations in the end-effector; therefore, when the robot is in an optimal position for machining tasks, the position of sections ($S_1 - S_3$) are fixed. Thus, this part of the robot can considered rigid due to the use of a rigidizing system[28] up until point 18 and the end-effector can be controlled for machining processes. The initial navigation path is generated using a tip-following algorithm[30] for navigation tasks. However, the navigation of the entire HRSA and the rigidizing system is not the objective of this paper. Finally, $P = (P_X, P_Y, P_Z)^T$ and $R \in \Re^{3\times3}$ represent the final position and orientation of the end-effector.

The general kinematic representation of the bend and direction angles for a section is shown in Fig. 2. The arc of $S_{i,k}$ is produced by the bend angle ($qb_{i,k} \in \Re$) and the radius ($r_{i,k} \in \Re^+$). By

https://doi.org/10.1017/S0263574715000946 Published online by Cambridge University Press

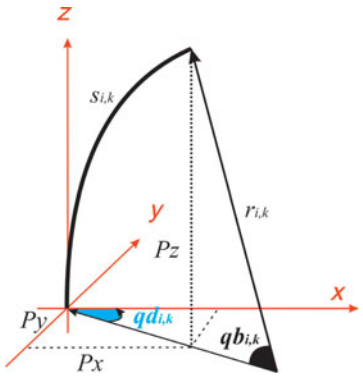

Fig. 2. Kinematic representation for a section $S_{i,k}$ formed by bend $qb_{i,k}$ and direction $qd_{i,k}$ angles.

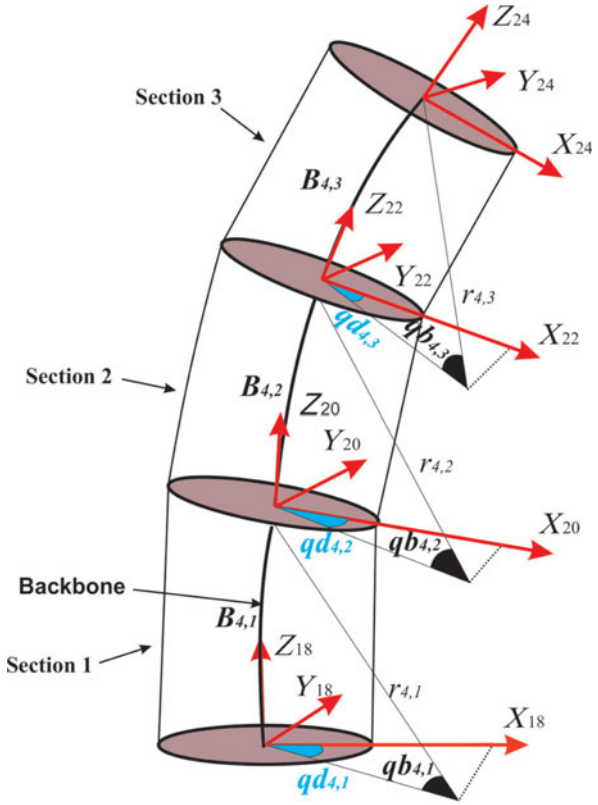

Fig. 3. General kinematic representation of stage 4 controlled by six DoF. The backbone of each section is represented by $B_{i,j}$ and each two degrees of freedom of each section by $qd_{i,k}$ and $qb_{i,k}$.

definition, the arc of $S_{i,k}$ is given by the product of the bend angle ($qb_{i,k}$) and the radius ($r_{i,k}$). Moreover, the direction angle ($qd_{i,k} \in \Re$) is the matrix product of two rotations about the Z axis (clockwise and anticlockwise) as described in section 3.1. In this model, the direction angle does not correspond to a pure rotation about the Z axis; the movement of the robot is in the XZ plane when the direction angle is equal to zero.

The kinematic representation of stage S_4 is shown in Fig. 3. The stage is divided in three sections and the length of the backbone ($B_{i,j} \in \Re^+$) between each section/disc is the same; where $i = \{1, 2, 3, 4\}$ is the number of stages and $j = \{1, 2, 3\}$ is the number of the backbone segment. In this representation, the length of the backbone between each joint is represented by $B_{4,1}$ to $B_{4,3}$. In each section, two DoF are controlled (see Fig. 2): the direction angle ($qd_{i,k}$) and the bending angle ($qb_{i,k}$). Finally, $r_{i,k}$ is the radius formed by the bending angle and the arc trajectory between each joint. The bending angle ($b_{i,k}$) can be considered to be in the XZ plane when $qd_{i,k} = 0$. Hence, a combination of

https://doi.org/10.1017/S0263574715000946 Published online by Cambridge University Press

matrices is proposed in order to eliminate the rotation about the central axis of the backbone induced by $qd_{i,k}$; this is very important in order to avoid twist about the backbone. Similar solutions of three sequential rotations with dimensions of (3 × 3) (*Rz, Ry, R-z*) (where positive rotations are clockwise) can be seen in refs. [19] and [21]. The difference between these approaches and the proposed method is the use of homogenous matrices and the proposed equations. In the next section, the kinematic model is described in detail.

3. Novel Approach to Calculating the Kinematic Model

The FK of the last stage of the HRSA can be written as

$$x_{ef} = f(r_4, q_4). \tag{1}$$

Equation (1) maps the six joint angles of the last segment of the end-effector in terms of position and orientation. In this equation, $x_{ef} \in \Re^{4\times4}$ is a homogeneous matrix that represents the position and orientation. Similarly, x_{ef} can be described in the form

$$x_{ef} = \begin{bmatrix} n & o & a & P \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

where $n \in \Re^{3\times1}$, $o \in \Re^{3\times1}$ and $a \in \Re^{3\times1}$ are an orthonormal triple, representing the orientation; $q_6^0 = (qd_{4,1}, qb_{4,1}, qd_{4,2}, qb_{4,2}, qd_{4,3}, qb_{4,3})^T$ is the vector of the six bending/direction angles and $r_3^0 = (r_{4,1,}, r_{4,2}, r_{4,3})^T$ is the vector of the three instantaneous radii that change for different combinations of bending angles. Thus, the FK of the end-effector can be controlled in task space of six dimensions.

3.1. Forward kinematics for a general kth section

The proposed method calculates the FK based on the combination of Euler angles and homogeneous transformation matrices. The matrices are presented in the following way:

$$T_1 = \begin{bmatrix} \cos(qd_{i,k}) & -\sin(qd_{i,k}) & 0 & r_{i,k}\cos(qd_{i,k}) \\ \sin(qd_{i,k}) & \cos(qd_{i,k}) & 0 & r_{i,k}\sin(qd_{i,k}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$T_2 = \begin{bmatrix} \cos(qb_{i,k}) & 0 & \sin(qb_{i,k}) & -r_{i,k}\cos(qb_{i,k}) \\ 0 & 1 & 0 & 0 \\ -\sin(qb_{i,k}) & 0 & \cos(qb_{i,k}) & r_{i,k}\sin(qb_{i,k}) + b_{i,j} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$T_3 = \begin{bmatrix} \cos(qd_{i,k}) & \sin(qd_{i,k}) & 0 & 0 \\ -\sin(qd_{i,k}) & \cos(qd_{i,k}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

Therefore, the FK of a *k*th section is obtained multiplying Eqs. (3), (4) and (5).

$$T_k = T_1 T_2 T_3. \tag{6}$$

https://doi.org/10.1017/S0263574715000946 Published online by Cambridge University Press

In Eqs. (3) and (4), the radius ($r_{i,k}$) can be calculated as

$$r_{i,k} = \frac{S_{i,k}}{qb_{i,k}}. \tag{7}$$

The reader can check that this equation depends on the bend angle. However, in this equation, if the bend angle is equal to zero the result is infinite. In order to avoid this problem, the variable $b_{i,j} \in \Re^+$ has been added to Eq. (4). The variable $b_{i,j}$ is defined using the following algorithm:

$$\begin{aligned}
&\text{If } \boldsymbol{q}b_{i,k} == 0,\\
&r_{i,k} = 0;\\
&b_{i,j} = S_{i,k};\\
&\text{Else}\\
&r_{i,k} = \frac{S_{i,k}}{qb_{i,k}}\\
&b_{i,j} = 0; \text{End}.
\end{aligned}$$

In this algorithm, the logic structure indicates that when the bending angle is equal to zero, the distance between the frame of reference *j* and the next frame of reference *j+1* is equal to the distance $S_{i,k}$, i.e. a straight line is produced when $qb_{i,k} = 0$.

3.2. Forward kinematics of stage 4 (S_4)

The FK of last segment is obtained using the concept of the FK for *kth* section. In this case, six DoF are considered. The matrices are presented in the following way:

$$T_1 = \begin{bmatrix} \cos(qd_{4,1}) & -\sin(qd_{4,1}) & 0 & r_{4,1}\cos(qd_{4,1}) \\ \sin(qd_{4,1}) & \cos(qd_{4,1}) & 0 & r_{4,1}\sin(qd_{4,1}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$T_2 = \begin{bmatrix} \cos(qb_{4,1}) & 0 & \sin(qb_{4,1}) & -r_{4,1}\cos(qb_{4,1}) \\ 0 & 1 & 0 & 0 \\ -\sin(qb_{4,1}) & 0 & \cos(qb_{4,1}) & r_{4,1}\sin(qb_{4,1}) + b_{4,1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

$$T_3 = \begin{bmatrix} \cos(qd_{4,1}) & \sin(qd_{4,1}) & 0 & 0 \\ -\sin(qd_{4,1}) & \cos(qd_{4,1}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

$$T_4 = \begin{bmatrix} \cos(qd_{4,2}) & -\sin(qd_{4,2}) & 0 & r_{4,2}\cos(qd_{4,2}) \\ \sin(qd_{4,2}) & \cos(qd_{4,2}) & 0 & r_{4,2}\sin(qd_{4,2}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

https://doi.org/10.1017/S0263574715000946 Published online by Cambridge University Press

$$T_5 = \begin{bmatrix} \cos(qb_{4,2}) & 0 & \sin(qb_{4,2}) & -r_{4,2}\cos(qb_{4,2}) \\ 0 & 1 & 0 & 0 \\ -\sin(qb_{4,2}) & 0 & \cos(qb_{4,2}) & r_{4,2}\,(\sin(qb_{4,2}) + b_{4,2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

$$T_6 = \begin{bmatrix} \cos(qd_{4,2}) & \sin(qd_{4,2}) & 0 & 0 \\ -\sin(qd_{4,2}) & \cos(qd_{4,2}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

$$T_7 = \begin{bmatrix} \cos(qd_{4,3}) & -\sin(qd_{4,3}) & 0 & r_{4,3}\cos(qd_{4,3}) \\ \sin(qd_{4,3}) & \cos(qd_{4,3}) & 0 & r_{4,3}\sin(qd_{4,3}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

$$T_8 = \begin{bmatrix} \cos(qb_{4,3}) & 0 & \sin(qb_{4,3}) & -r_{4,3}\cos(qb_{4,3}) \\ 0 & 1 & 0 & 0 \\ -\sin(qb_{4,3}) & 0 & \cos(qb_{4,3}) & r_{4,3}\,(\sin(qb_{4,3}) + b_{4,3} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

$$T_9 = \begin{bmatrix} \cos(qd_{4,3}) & \sin(qd_{4,3}) & 0 & 0 \\ -\sin(qd_{4,3}) & \cos(qd_{4,3}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{16}$$

Using Eqs. (8) to (16), the end-effector positions and orientations can be expressed as

$$x_{ef} = T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9. \tag{17}$$

The algorithm presented for the *kth* section has to be executed first for each bending ($qb_{i,k}$) angle prior to using Eq. (17) in order to avoid singularity points. Finally, the FK was tested in several simulations in a Matlab environment, an example of which is shown in Fig. 4. This simulation was tested by introducing $S_{i,k} = 100$ mm for each section.

3.3. Work space generation

The validity of the FK proposed is proven by the Work Space Generation (WSG) that the FK can map.[33,34] This WSG represents the possible end positions and orientations that the stage S_4 can reach. Figure 5(a) shows the possible end positions of stage S_4 in one dimension. The dashed lines in black represent the trajectory of the HRSA, while the dots in red, blue and green represent the end positions of section 1, 2 and 3 respectively. This figure proves the expected position for each section. Figure 5(b) shows the work volume of the end-effector. In this figure, the possible positions that the end-effector can reach are represented in red. Figure 5(c) shows that the maximum height is 363.66 mm and the maximum diameter is 423.2 mm.

3.4. Inverse kinematics of section 4 (S_4)

The IK is presented for two, four and six DoF; therefore, three methods are solved in this manner because the manipulation space is restricted and the range of motion of the HRSA is affected. Thus, depending on the restricted environment, two, four or six DoF can be selected to perform machining tasks. Figure 6 shows a general representation of the vectors used to solve the IK.

https://doi.org/10.1017/S0263574715000946 Published online by Cambridge University Press

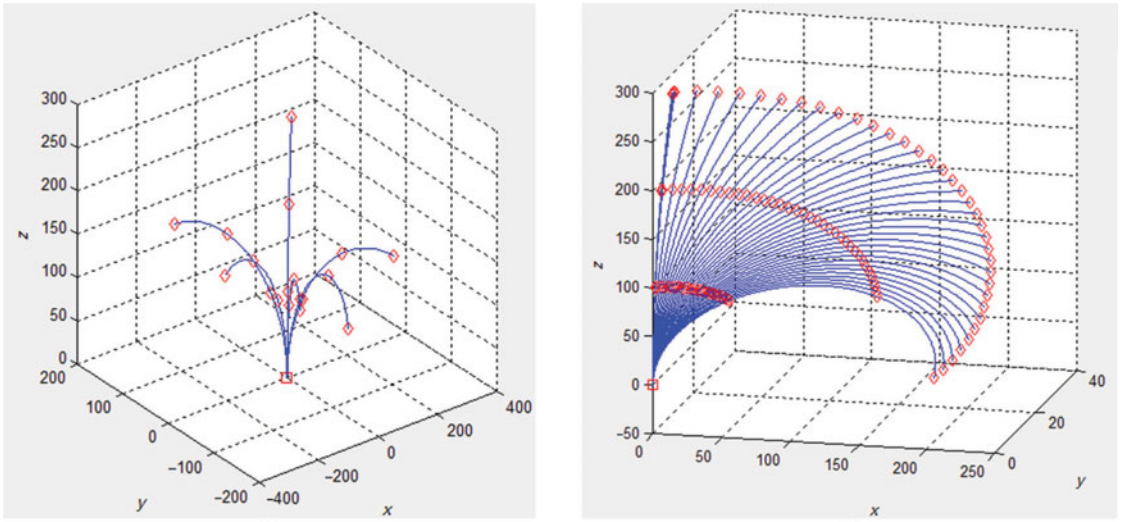

Fig. 4. Virtual model of stage 4 in Matlab. Several positions are represented using the forward kinematics. The blue curves represent the backbone trajectory of the HRSA and the red marks the end position of each section. The units of each axis are in mm.

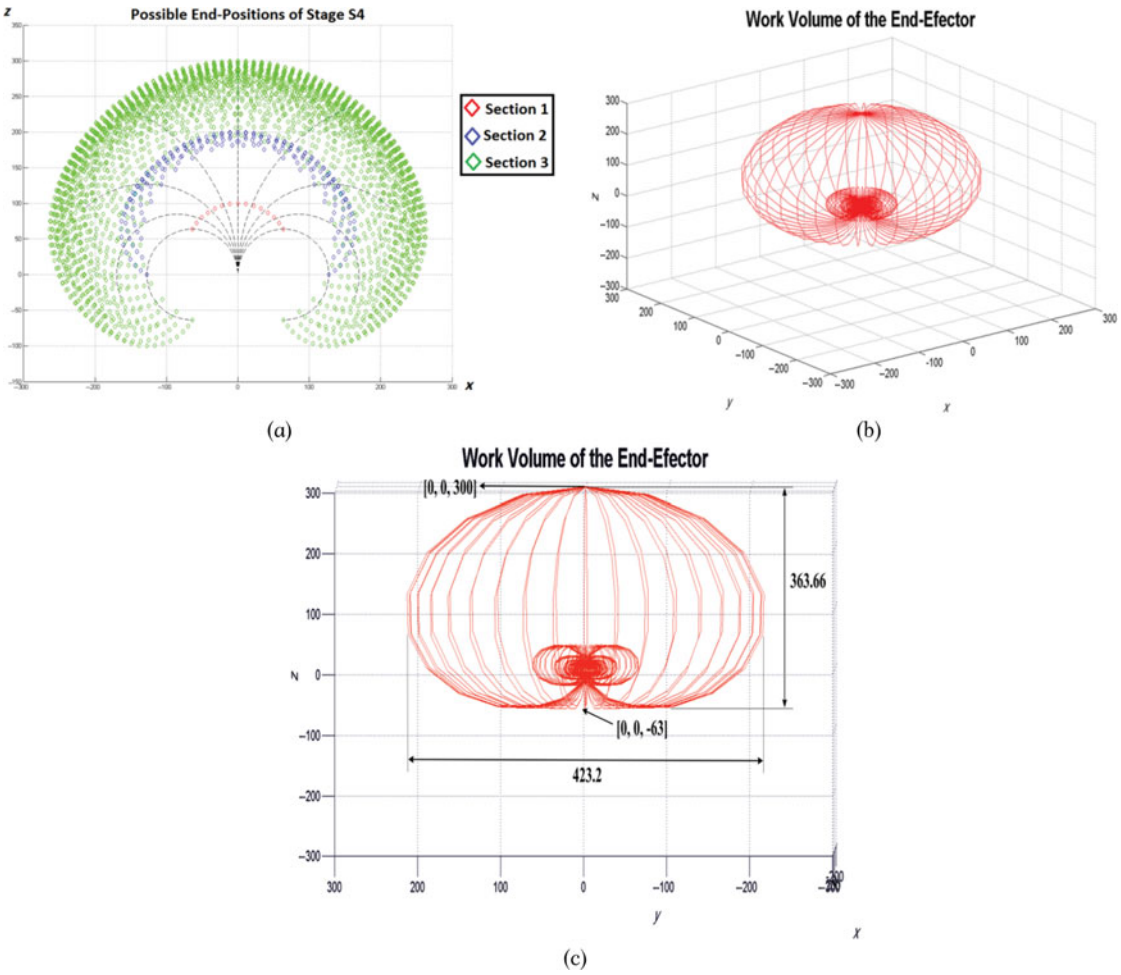

Fig. 5. (a) Work space representation in one dimension of stage S_4 in two dimensions. These final points were obtained using the forward kinematics for a range of $-\pi/2$ to $\pi/2$ radians. (b) Work volume of the end-effector in 3D. (c) The units of each axis are in mm.

https://doi.org/10.1017/S0263574715000946 Published online by Cambridge University Press

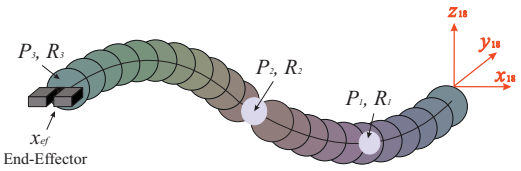

Fig. 6. General representation of the kinematic problem. The matrix x_{ef} represents the final orientation and position of the end-effector; each position and orientation are represented for each section (P_1, R_1, P_2, R_2, P_3 and R_3).

3.4.1. Inverse kinematics for the last two DoF. The proposed method is based on the Taylor Series approximation of order 4[29] to obtain the bending angle $qb_{4,3}$ using only the information of P_{z3} with respect the previous frame of reference (see Fig. 6). The vector $P_3 = (P_{x3}, P_{y3}, P_{z3})^{\mathrm{T}}$ of the last section ($S_{4,3}$) is obtained from Eq. (18)

$$T_9^7 = x_{ef}(T_6^1)^{-1}. \tag{18}$$

The vector P_{z3} can be expressed as Eq. (19) and the radius as Eq. (20)

$$P_{Z3} = r_{4,3}\ \sin(qb_{4,3}) \tag{19}$$

$$r_{4,3} = \frac{S_{4,3}}{q_2}\sin(qb_{4,1}). \tag{20}$$

Substituting Eq. (20) into Eq. (19), yields

$$\frac{\sin(qb_{4,3})}{qb_{4,3}} = \frac{P_{Z3}}{S_{4,3}}. \tag{21}$$

The Taylor Series approximation for sine is given by

$$\sin(x) \approx x + \frac{x^5}{5!} - \frac{x^3}{3!}\ldots. \tag{22}$$

Using Eq. (22), Eq. (21) can be expressed as

$$qb_{4,3} + \frac{qb_{4,3}^5}{120} - \frac{qb_{4,3}^3}{6} \approx qb_{4,3}\frac{P_{Z3}}{S_{4,3}}. \tag{23}$$

Therefore, $qb_{4,3}$ is solved by the roots of Eq. (23)

$$qb_{4,3}^4 - 20qb_{4,3}^2 + 120 - 120\frac{P_{Z3}}{S_{4,3}} = 0. \tag{24}$$

In this case, $S_{4,3}$ is the length of the arc and is constant. For our experiments, $S_{4,3} = 100$ mm.

The roots of Eq. (24) generate four solutions; the best solution is the minimum error between $P_{z3}/S_{4,3}$ and $sin(qb_{4,3})/qb_{4,3}$.

The direction angle $qd_{4,3}$ can be obtained as

$$\begin{aligned} &\text{If } Px3 == 0 \\ &qd_{4,3} = 0 \\ &\text{Else} \end{aligned} \tag{25}$$

https://doi.org/10.1017/S0263574715000946 Published online by Cambridge University Press

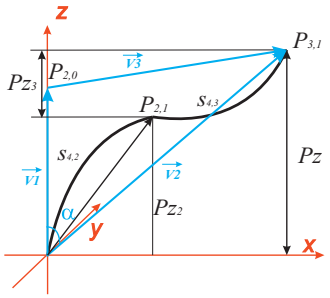

Fig. 7. Vector representation of two bending angles when the direction angles have the same value.

$$qd_{4,3} = \tan^{-1}\left(\frac{P_{y3}}{P_{x3}}\right) \tag{26}$$

End.

This algorithm avoids the singularity point when P_{x3} is equal to zero.

3.4.2. Inverse kinematics for the last four DoF. In the same way, as section 3.4.1, the vector P_3 for the last two sections (2 and 3) is obtained from Eq. (27)

$$T_9^4 = x_{ef}(T_3^1)^{-1}. \tag{27}$$

In this case, the IK problem becomes very complex due to the redundancy of the system; therefore, it is proposed to control two bending angles using the direction angles given below:

1. Case I: All direction angles are the same, in other words, $qd_{4,2} \equiv qd_{4,3}$. Therefore, $qd_{4,2}$ is obtained as

If $P_{x3} == 0$

$qd_{4,2} = 0;$

Else

$$qd_{4,2} = \tan^{-1}\left(\frac{P_{y3}}{P_{x3}}\right) \tag{28}$$

End

2. Case II: The relationship between the bending angles $\phi = qb_{4,2} + qb_{4,3}$ is given by

If $P_{x3} >= 0$

$$\varphi = \cos^{-1}(a_z) \tag{29}$$

Else

$$\varphi = -\cos^{-1}(a_z) \tag{30}$$

End,

where $a_z \in \Re$ is the third element of vector $a = (a_x, a_y, a_z)^T$ (see Eq. (2)).

Let us consider the initial position, $P_{2,0}$, when $qb_{4,2} = 0$ *rad* (see Fig. 7). In this case, $S_{4,2}$ is considered to be 100 mm. The modulus of vector $|\vec{v}_1|$ represents the length of the arc $S_{4,2}$. The rest of the vector modules of Fig. 7 can be calculated as

$$|\vec{v}_1| = S_{4,2} \tag{31}$$

https://doi.org/10.1017/S0263574715000946 Published online by Cambridge University Press

Table I. Constrains between bending angles.

Constrain: $c = qb_{4,3}/qb_{4,2}$	κ
1	1
0.9	1.0249
0.8	1.0512
0.7	1.0791
0.6	1.1086
0.5	1.1399
0.4	1.1732
0.3	1.2087
0.2	1.2464
0.1	1.2867
−0.1	1.3760
−0.2	1.4255
−0.3	1.4787
−0.4	1.5361
−0.5	1.5981
−0.6	1.6653
−0.7	1.7382
−0.8	1.8177
−0.9	1.9046
−1	2

$$|\vec{v}_2| = \sqrt{P_x^2 + P_y^2 + P_z^2} \tag{32}$$

$$|\vec{v}_3| = \sqrt{P_x^2 + P_y^2 + (P_z - S_{4,2})^2}. \tag{33}$$

The angle α is calculated using the law of cosines as

$$\alpha = \cos^{-1}\left(\frac{\vec{v}_2^2 + \vec{v}_1^2 - \vec{v}_3^2}{2\,|\vec{v}_1|\,|\vec{v}_2|}\right). \tag{34}$$

Angle α will be used to calculate the bending angle $qb_{4,2}$ according to Eq. (35) and the constraints of Table I.

$$qb_{4,2} = \kappa\alpha, \tag{35}$$

where $\kappa \in \Re$ is correlated to the constraint, $c \in \Re$. The κ values of Table I were obtained, minimizing the error between the desired constraint (c) by means of the FK and the output of Eqs. (34) and (35). The following polynomial regression model is proposed to generate the values of κ according to Table I:

$$\kappa = 0.0183c^4 - 0.0555c^3 + 0.1518c^2 - 0.4441c + 1.3299. \tag{36}$$

The coefficients of Eq. (36) were obtained in order to generate an exact correlation between the predictors κ and c using the values given in Table I.

The various values of c can be implanted in a for-loop algorithm evaluating the maximum error between the desired position and orientation and the result obtained by Eq. (36).
$qb_{4,3}$ can be obtained using Eqs. (29) or (30):

$$qb_{4,3} = \phi - qb_{4,2}. \tag{37}$$

Finally, the last direction angle, $qd_{4,3}$, can be obtained using Eqs. (18) and (26).

https://doi.org/10.1017/S0263574715000946 Published online by Cambridge University Press

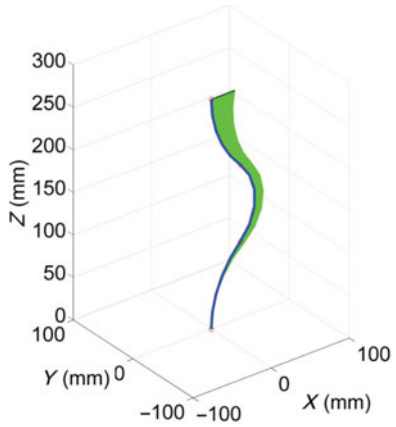

Fig. 8. Linear trajectories using the inverse of the Jacobian.

3.4.3. Inverse kinematics for the last six DoF. In section 3.4.2, it was shown that many cases can be obtained due to the lack of information relating to the final positions of each section and the fact that the radius of the bending angle is not constant for different values of $qb_{4,1}$, $qb_{4,2}$ and $qb_{4,3}$. For this reason, traditional approaches are impractical or unfeasible when the numbers of DoF are increasing. However, a numerical implementation based on the inverse of the Jacobian, $J^{-1}(q_4)$, can be implemented for a previous instant in time, such as

$$q_4(t_{k+1}) = q_4(t_k) + J^{-1}(q_4(t_k))v_e(t_k)\Delta t, \tag{38}$$

where $t_k \in \Re^+$ is the time instant; $t_{k+1} \in \Re^+$ is the next period of time; $J^{-1}(q_4(t_k)) \in \Re^{6\times6}$ is the inverse of the Jacobian for stage 4 in the time instant; $v_e(t_k) \in \Re^{6\times1}$ is the vector of instant velocities of the end-effector and $\Delta t \in \Re$ is an scalar that minimize the error.

This method implies that the joint variables, q_4, corresponding to a given desired end-effector pose (x_{efd}) are precisely computed only when the error between the desired end-effector pose and the candidate end-effector pose (x_{efc}) is reduced to within a given threshold; therefore, the computing time depends on the dynamic characteristics that are produced by the error of the differential equations. Figure 8 shows a linear trajectory of the end-effector (black line) with a length of 30 mm. The blue pose is the initial position and the green lines are the result of the IK. This figure shows precise poses following the straight line using the inverse of the Jacobian. The maximum vertical error produced is less than 30 μ. The next section shows the method proposed to compute the Jacobian used for this experiment.

4. Approach to Calculate the Jacobian

The method proposed uses the vectors (n_k, o_k, a_k) that correspond to the matrix transformation of each DoF. In other words, the vectors selected represent the axis of each virtual joint around the backbone.

The matrix transformation is represented as $T_k^0 \in \Re^{4\times4}$ for joint k; where

T_k^0 is the transformation from the first frame to the next k joint frame;
$n_k \in \Re^{3\times1}$ is the vector used for joints about the X axis;
$o_k \in \Re^{3\times1}$ is the vector used for joints about the Y axis;
$a_k \in \Re^{3\times1}$ is the vector used for joints about the Z axis;
$k \in \{1, 2, \ldots, n\}$ is the joint number.

The virtual joints along the backbone for each disc are represented as combinations of o_k and a_k for the purpose of obtaining the Jacobian. The overall Jacobian of linear and angular velocities is defined as

$$J_i = \begin{bmatrix} J_{v,k} \ldots J_{v,n} \\ J_{\omega,k} \ldots J_{\omega,n} \end{bmatrix}. \tag{39}$$

https://doi.org/10.1017/S0263574715000946 Published online by Cambridge University Press

For linear velocities, the *kth* column $J_{v,k}$ is defined according to the joint axis of each joint *k*. In this instance, two cases are proposed to take into account the fact that all the virtual points are considered joints about the *Y* and *X* axis. The proposed definitions are defined as

1. For direction angles:

$$J_{v,k} = (a_{k+\varphi} - a_{k+1+\varphi}) \times (P - P_{k+\varphi}) \tag{40}$$

$$J_{\omega,k} = (a_{k+\varphi} - a_{k+1+\varphi}) \tag{41}$$

2. For bending angles:

$$J_{v,k} = o_{k+\varphi} \times (P - P_{(k-1+\varphi)}) \tag{42}$$

$$J_{\omega,k} = o_{k+\varphi}. \tag{43}$$

In these two equations, the new term ϕ is given by

$$\phi = ns - 1, \tag{44}$$

where $ns \in \Re^+$ is the number of sections.

The symbol $\times$ represents the cross product of Eqs. (40) and (42). The Jacobian for the last stage can be obtained using the components of $J_{v,k}$ and $J_{\omega,k}$

$$J_{s,4} = \begin{bmatrix} J_{v,1} & \cdots & J_{v,6} \\ J_{\omega,1} & \cdots & J_{\omega,6} \end{bmatrix}. \tag{45}$$

The Jacobian for each section is calculated using the previous method, as described below: the first direction angle ($k = 1$) is located in section 1, hence, $ns = 1$, using the vector a_k and $\phi = 0$. Thus, J_1 is computed using Eqs. (40) and (41)

$$J_1 = \begin{bmatrix} J_{v,1} \\ J_{\omega,1} \end{bmatrix} = \begin{bmatrix} (a_1 - a_2) \times (P - P_1) \\ a_1 - a_2 \end{bmatrix}, \tag{46}$$

where a_1 is the vector along the *X* axis of T_1^0; P_1 is the position of T_1^0; a_2 is the vector along the *X* axis of T_2^0 and *P* is the final position of the end-effector T_9^0.

The first bend angle ($k = 2$) is located in section 1, hence, $ns = 1$, using the vector o_k and $\phi = 0$; J_2 is computed using Eqs. (42) and (43)

$$J_2 = \begin{bmatrix} J_{v,2} \\ J_{\omega,2} \end{bmatrix} = \begin{bmatrix} o_2 \times (P - P_1) \\ o_2 \end{bmatrix}, \tag{47}$$

where o_2 is the vector along the *Y* axis of T_2^0.

The second direction angle ($k = 3$) is located in section 2, hence, $ns = 2$, using the vector a_k and $\phi = 1$; J_3 is computed using Eqs. (40) and (41) such as

$$J_3 = \begin{bmatrix} J_{v,3} \\ J_{\omega,3} \end{bmatrix} = \begin{bmatrix} (a_4 - a_5) \times (P - P_4) \\ a_4 - a_5 \end{bmatrix}, \tag{48}$$

where a_4 is the vector along the *X* axis of T_4^0, P_4 is the position of T_4^0 and a_5 is the vector along the *X* axis of T_5^0.

https://doi.org/10.1017/S0263574715000946 Published online by Cambridge University Press

The second bend angle ($k = 4$) is located in section 2, hence, $ns = 2$, using the vector o_k and $\phi = 1$; J_4 is computed using Eqs. (42) and (43)

$$J_4 = \begin{bmatrix} J_{v,4} \\ J_{\omega,4} \end{bmatrix} = \begin{bmatrix} o_5 \times (P - P_4) \\ o_5 \end{bmatrix}, \tag{49}$$

where o_5 is the vector along the Y axis of T_5^0.

The third direction angle ($k = 5$) is located in section 3, hence, $ns = 3$, using the vector a_k and $\phi = 2$; J_5 is computed using Eqs. (40) and (41):

$$J_5 = \begin{bmatrix} J_{v,5} \\ J_{\omega,5} \end{bmatrix} = \begin{bmatrix} (a_7 - a_8) \times (P - P_7) \\ a_7 - a_8 \end{bmatrix}, \tag{50}$$

where a_7 is the vector along the X axis of T_7^0, P_7 is the position of T_7^0 and a_8 is vector along the X axis of T_8^0. Finally, the third bend angle ($k = 6$) is located in section 3, hence, $ns = 3$, using the vector o_k

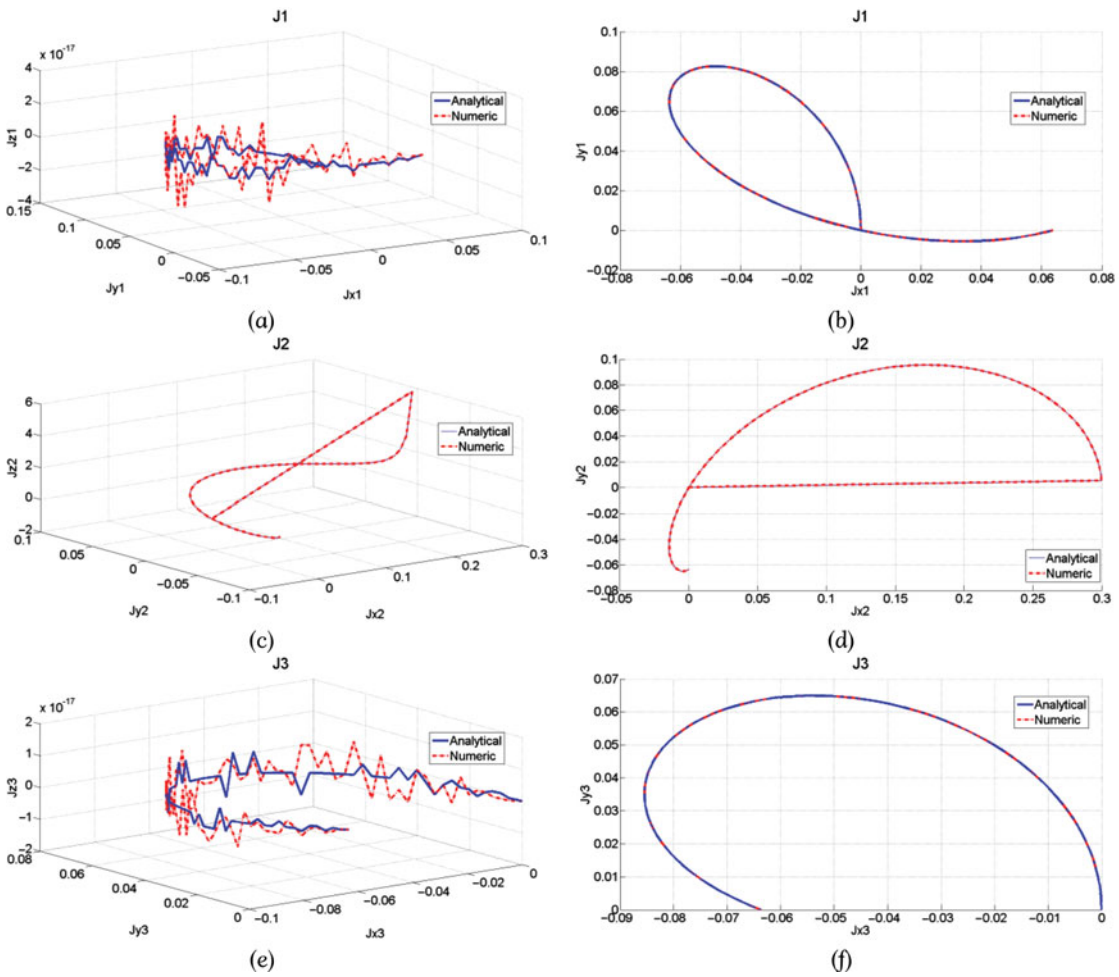

Fig. 9. Trajectories of analytical (blue) and proposed (red) models for sections 1, 2 and 3. Panels (a), (c), (e), (g), (i) and (k) show the two trajectories in three dimensions for J_1, J_2, J_3, J_4, J_5 and J_6 respectively. Panels (b), (d), (f), (h), (j) and (l) show the two trajectories in two dimensions.

https://doi.org/10.1017/S0263574715000946 Published online by Cambridge University Press

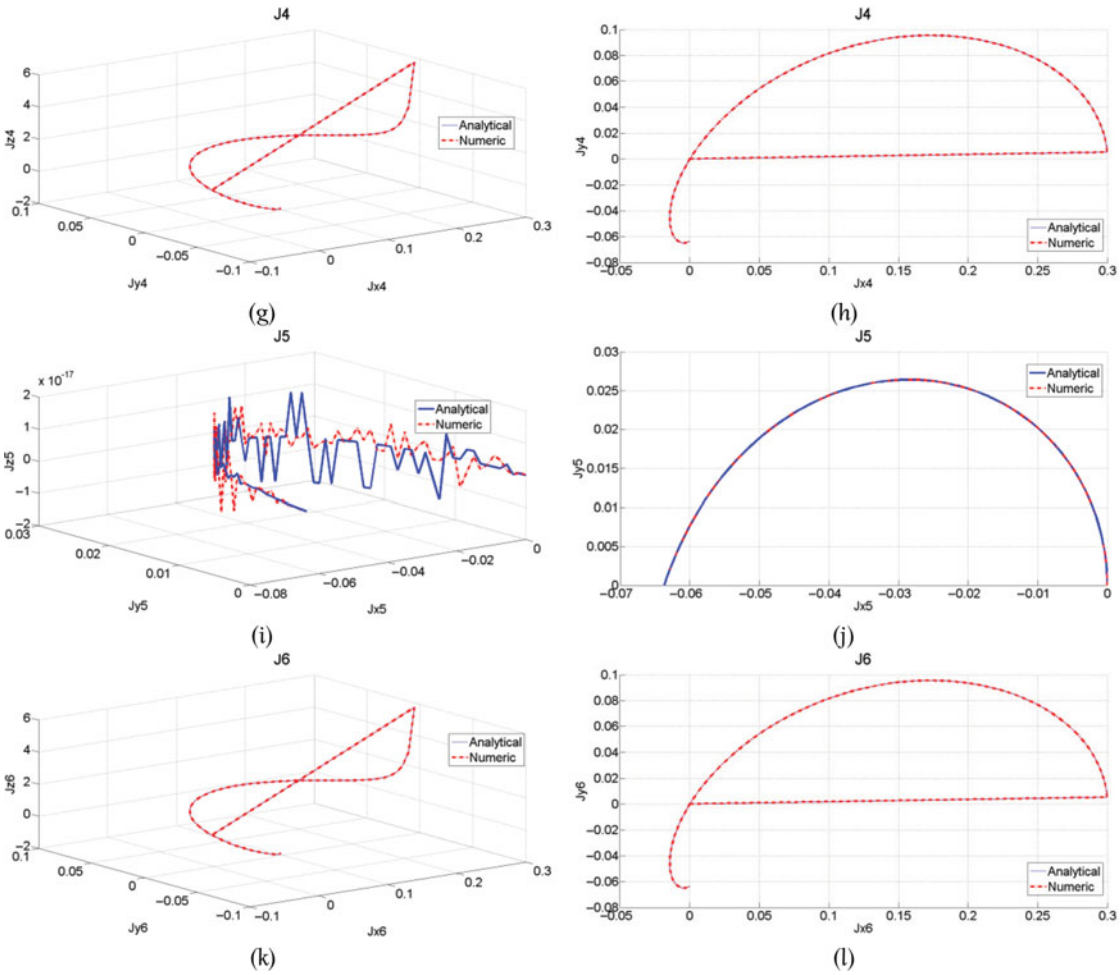

Fig. 9. Continued

Table II. Mean average error between the analytical and proposed models.

Jacobian	Mean average error
$[Jx_1, Jy_1, Jz_1]$	[0.0505E-17, −0.2983E-17, −0.0081E-17]
$[Jx_2, Jy_2, Jz_2]$	[0.0246E-16, −0.0049E-16, −0.1954E-16]
$[Jx_3, Jy_3, Jz_3]$	[0.0951E-17, −0.1029E-17, 0.0314E-17]
$[Jx_4, Jy_4, Jz_4]$	[−0.1489E-17, 0.0555E-17, −0.0760E-17]
$[Jx_5, Jy_5, Jz_5]$	[−0.0067E-16, −0.1028E-16, −0.0060E-16]
$[Jx_6, Jy_6, Jz_6]$	[−0.0315E-16, 0.0034E-16, −0.3227E-16]

and $\phi = 2$, J_6 is computed using Eqs. (42) and (43):

$$J_6 = \begin{bmatrix} J_{v,6} \\ J_6 \end{bmatrix} = \begin{bmatrix} o_8 \times (P - P_7) \\ o_8 \end{bmatrix}, \tag{51}$$

where o_8 is the vector along the Y axis of T_8^0.

The matrix of the Jacobian allows the determination of joint velocities $\dot{q}_{i,k}$ necessary to achieve a desired end-effector velocity ξ. In this case, the minimum norm solution is given by

$$\dot{q}_{i,k} = J\xi. \tag{52}$$

https://doi.org/10.1017/S0263574715000946 Published online by Cambridge University Press

4.1. Justification

Equations (40)–(43) are proposed in this form because the kinematic model is composed of the product of the two matrices for the direction angle. These types of multiplications produce quadratic terms in sine and cosine and the correct behaviour between each section has to be identified correctly. Therefore, using this method, the vectors of position ($P_{k+\phi}$ or $P_{k-1+\phi}$) and the rotation vectors ($a_{k+\phi}$, $a_{k+1+\phi}$, $o_{k+\phi}$) correspond to the correctly selected vectors required for the cross product for each section. As proof of verification, this method is compared with the partial differential equations (analytical method) of the Jacobian and the error is measured between the two methods. Figure 9 shows the trajectories between the analytical (blue line) and proposed (red line) models for each element of the Jacobian, J_i. The error between the two models is quantified in Table II and Fig. 9. The errors are very small; proving that the method proposed is useful for similar hyper redundant HRSA systems.

5. Conclusion

The kinematic model to control the end-effector of a HRSA based on backbone curvature was presented in detail; the equations reported provide the forward and inverse models to control an efficient trajectory in a closed form between $\pm\,\pi$ radians, consequently, the singularity points are eliminated for the range of movement of each section, $S_{i,k}$, for both kinematic models. The proposed IK method is useful for real-time applications that require precision control and a fast computing process; therefore, it can be used for machining processes where the base stages of the HRSA are fixed in a specific pose. The method for obtaining the Jacobian is of key importance for hyper-redundant robots where the number of DoF is high and difficulties are encountered in finding correct analytical equations or numerical approximations, due to the complexity of the elements of the matrix. These findings are useful for multi-axis manipulations of the end-effectors of continuum robots so that processing tasks are possible in closely confined environments (e.g. aero-engines) where complex snake motions are required to reach specific areas to repair internal parts. For future work, these methods will be used to control a machining process, where interpolations of trajectories or control forces are required for implementation.

Acknowledgements

This research work was led by Dr. Salvador Cobos Guzman under the EU Seventh Framework Programme in the theme: Robots for automation of post-production and other auxiliary processes, collaborative (Grant No. 284959), MiRoR project http://www.miror.eu/.

Supplementary Material

To view supplementary material for this article, please visit http://dx.doi.10.1017/S0263574715000946.

References

1. Y. Yekutieli, T. Flash and B. Hochner, *Biomechanics: Hydroskeletal. Encyclopedia of Neuroscience* (R. S. Larry, ed.) (Elsevier Academic Press, Oxford, UK, 2009), vol. 2, pp. 189–200.
2. W. M. Kier and K. K. Smith, "Tongues, tentacles and trunks: The biomechanics of movement in muscular-hydrostats," *Zool. J. Linn. Soc.* **83**, 307–324 (1985).
3. B. C. Jayne, "kinematics of terrestrial snake locomotion," *Copeia* **4**, 915–927 (1986).
4. A. A. Transeth and K. Y. Pettersen, "A survey on snake robot modeling and locomotion," *Robotica* **27**(7), 999–1015 (2009).
5. G. S. Chirikjian and J. W. Burdick, "Kinematics of Hyper-Redundant Robot Locomotion with Applications to Grasping," *IEEE International Conference on Robotics and Automation,* Sacramento, CA, USA (1991) pp. 720–725.
6. G. S. Chirikjian and J. W. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Trans. Robot. Autom.* **10**(3), 343–354 (1994).
7. V. C. Anderson and R. C. Horn, "Tensor arm manipulator design," *Transactions of the ASME*, vol. 67, DE-57 (1967) pp. 1–12.

8. I. D. Walker and M. W. Hannan, "A Novel Elephant's Trunk Robot," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Atlanta, USA (1999) pp. 410–415.
9. T. Mahl, A. Hildebrandt and O. Sawodny, "Forward Kinematics of a Compliant Pneumatically Actuated Redundant Manipulator," *7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Singapore (2012) pp. 1267–1273.
10. I. A. Gravagne and I. D. Walker, "Manipulability, force, and compliance analysis for planar continuum manipulators," *IEEE Trans. Robot. Autom.* **18**(3), 263–273 (2002).
11. OC robotics, http://www.ocrobotics.com/.
12. Y.-J. Kim, S. Cheng, S. Kim and K. Iagnemma, "Design of a Tubular Snake-like Manipulator with Stiffening Capability by Layer Jamming," *IEEE/RSJ International Conference on Intelligent Robots and Systems Vilamoura*, Portugal (2012) pp. 4251–4256.
13. E. Lanteigne and A. Jnifene, "Design of a Link-Less Hyper-Redundant Manipulator and Composite Shape Memory Alloy Actuator," *Conference on Electrical and Computer Engineering, CCECE '06*, Ottawa, Canada (2006) pp. 1180–1183.
14. R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Robot. Res.* **29**(13), 1661–1683 (2010).
15. F.-T. Cheng, T.-H. Chen and Y.-Y. Sun, "Inverse Kinematic Solutions for Redundant Manipulators using Compact Formulation," *IEEE/RSJ International Workshop on Intelligent Robots and Systems, Intelligence for Mechanical Systems*, Osaka, Japan (1991) pp. 153–158.
16. K. E. Zanganeh and J. Angeles, "The Inverse Kinematics of Hyper-Redundant Manipulators using Splines," *IEEE International Conference on Robotics and Automation*, Nagoya, Japan (1995) pp. 2797–2802.
17. F. Fahimi, H. Ashrafiuon and C. Nataraj, "An improved inverse kinematic and velocity solution for spatial hyper-redundant robots," *IEEE Trans. Robot. Autom.* **18**(1), 103–107 (2002).
18. Y. Y. Kim, G.-W. Jang and S. J. Nam, "Inverse kinematics of binary manipulators by using the continuous-variable-based optimization method," *IEEE Trans. Robot.* **22**(1), 33–42 (2006).
19. S. Yahya, H. A. F. Mohamed, M. Moghavvemi and S. S. Yang, "A Geometrical Inverse Kinematics Method for Hyper-Redundant Manipulators," *10th International Conference on Control, Automation, Robotics and Vision, ICARCV 2008*, Hanoi, Vietnam (Dec. 17–20, 2008) pp. 1954–1958.
20. S. Neppalli, M. A. Csencsits, B. A. Jones and I. D. Walker, "Closed-form inverse kinematics for continuum manipulator," *Adv. Robot.* **23**(15), 2077–2091 (2009).
21. E. Singla, S. Tripathi, V. Rakesh and B. Dasgupta, "Dimensional synthesis of kinematically redundant serial manipulators for cluttered environments," *Robot. Auton. Syst.* **58**(5), 585–595 (2010).
22. R. Mutlu, G. Alici and W. Li, "An effective methodology to solve inverse kinematics of electroactive polymer actuators modelled as active and soft robotic structures," *Mech. Mach. Theory* **67**, 94–110 (2013).
23. Z. Zhang, G. Yang and S. H. Yeo, "Inverse Kinematics of Modular Cable-Driven Snake-Like Robots with Flexible Backbones," *IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Qingdao, China (2011) pp. 41–46.
24. C. He, S. Wang, Y. Xing and X. Wang, "Kinematics analysis of the coupled tendon-driven robot based on the product-of-exponentials formula," *Mech. Mach. Theory* **60**, 90–111 (2013).
25. J. Gallardo, R. Lesso, J. M. Rico and G. Alici, "The kinematics of modular spatial hyper-redundant manipulators formed from RPS-type limbs," *Robot. Auton. Syst.* **59**(1), 12–21.
26. P. Falco and C. Natale, "On the stability of closed-loop inverse kinematics algorithms for redundant robots," *IEEE Trans. Robot.* **27**(4), 780–784 (2011).
27. R. S. Hartenberg and J. Denavit, "A kinematic notation for lower pair mechanisms based on matrices," *J. Appl. Mech.* **77**(2), 215–221 (1955).
28. J. Kell, S. Davies, I. McGill, G. Rigg, M. Raffles, M. Daine, M. Kong, D. Axinte, I. Marinescu and C. Herbert, Patent: EP2431140A1 (2012) pp. 1–10, 21.03.
29. G. B. Thomas and R. L Finney, *Calculus and Analytic Geometry*, 9th ed. (Addison Wesley, USA, 1996), ISBN 0-201-53174-7.
30. D. Palmer, S. Cobos Guzman and D. Axinte, "Real-time method for tip following navigation of continuum snake arm robots," *Robot. Auton. Syst.* **62**(10), 1478–1485 (2014).
31. R. Chhabra and M. R. Emami, "A generalized exponential formula for forward and differential kinematics of open-chain multi-body systems," *Mech. Mach. Theory* **73**, 61–75 (2014).
32. X. Dong, M. Raffles, S. Cobos Guzman, D. Axinte and J. Kell, "Design and analysis of a family of snake arm robots connected by compliant joints," *Mech. Mach. Theory* **77**, 73–91 (2014).
33. G. S. Chirikjian and I. Ebert-Uphoff, "Numerical convolution on the Euclidean group with applications to workspace generation," *IEEE Trans. Robot. Autom.* **14**(1), 123–136 (1998).
34. Y. Wang and G. S. Chirikjian, "Workspace generation of hyper-redundant manipulators as a diffusion process on SE(N)," *IEEE Trans. Robot. Autom.* **20**(3), 399–408 (2004).