Mechanism and Machine Theory 45 (2010) 454–466

Contents lists available at ScienceDirect

Mechanism and Machine Theory

journal homepage: www.elsevier.com/locate/mechmt

A real-time algorithm for simulation of flexible objects and hyper-redundant manipulators

S. Sreenivasan [a,b], Piyush Goel [a,c], Ashitava Ghosal [a,*]

[a] *Dept. of Mechanical Engineering, Indian Institute of Science, Bangalore 560 012, India*
[b] *Geometric Software Solution Ltd., Pune, India*
[c] *John Deere Technology Centre, Pune, India*

ARTICLE INFO

Article history:
Received 22 August 2008
Received in revised form 30 September 2009
Accepted 9 October 2009
Available online 14 November 2009

Keywords:
Redundancy resolution
Hyper-redundant manipulators
Tractrix
Flexible objects
Real-time simulation

ABSTRACT

Flexible objects such as a rope or snake move in a way such that their axial length remains almost constant. To simulate the motion of such an object, one strategy is to discretize the object into large number of small rigid links connected by joints. However, the resulting discretised system is highly redundant and the joint rotations for a desired Cartesian motion of any point on the object cannot be solved uniquely. In this paper, we revisit an algorithm, based on the classical tractrix curve, to resolve the redundancy in such hyper-redundant systems. For a desired motion of the 'head' of a link, the 'tail' is moved along a tractrix, and recursively all links of the discretised objects are moved along different tractrix curves. The algorithm is illustrated by simulations of a moving snake, tying of knots with a rope and a solution of the inverse kinematics of a planar hyper-redundant manipulator. The simulations show that the tractrix based algorithm leads to a more 'natural' motion since the motion is distributed uniformly along the entire object with the displacements diminishing from the 'head' to the 'tail'.

1. Introduction

In realistic computer animation of deformable objects such as snakes, ropes, trees, grass or human hair, the *axial length* of the object is preserved. To achieve this, one approach is to discretize the deformable object into a large number of rigid links connected by rotary or spherical joints and then apply motion at the joints to obtain a realistic animation. A deformable object discretised in such a way can be thought of as a hyper-redundant serial manipulator with a large number of links and joints. The main and well-known difficulty in the analysis of hyper-redundant manipulators is that given a desired motion of the end-effector or a point on the manipulator, there exists an *infinite* number of solutions for the motion of the joints (or the motion of links). The problem of obtaining or choosing a unique solution from this infinite set is called as the *resolution* of redundancy, and there exists a vast amount of literature on resolution of redundancy. In the following we briefly review the main approaches.

The first approach to be tried involved the use of *least-squares* technique or its variant. In this approach, the velocity of the end-effector is first related to the joint rates by the well-known manipulator Jacobian matrix. In the case of a redundant system, with more joint variables than Cartesian or task space variables, the manipulator Jacobian matrix is non-square. In the second step the joint rates are obtained, for a given Cartesian or task space velocity, by inverting the manipulator Jacobian using a pseudo-inverse [1]. In the final step, the joint values are updated from the computed joint rates. In its basic

* Corresponding author. Tel.: +91 80 2293 2956; fax: +91 80 2360 0648.
E-mail addresses: sreeenivasanss@yahoo.com (S. Sreenivasan), goelpiyush28@yahoo.co.in (P. Goel), asitava@mecheng.iisc.ernet.in (A. Ghosal).

0094-114X/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mechmachtheory.2009.10.005

form, the pseudo-inverse based method is at the level of velocity or infinitesimal motions and has the interesting property of minimising joint rates in the least-square sense. In other variants, the pseudo-inverse method has been applied with a weighting matrix, at the joint acceleration level and by including the null-space term to optimise additional desired quantities such as singularity [2], joint limits [3] and obstacle avoidance [4], minimisation of joint torques [5] or maximizing a manipulability index [6] (for details of various pseudo-inverse based schemes, see the review paper by Klein and Huang [7] and the textbook by Nakamura [8] and the references contained therein). The pseudo-inverse approach is a purely numerical and local approach. Moreover, since it involves inverting a matrix, it has a complexity of $\mathcal{O}(n^4)$, where n is the number of joint variables. This makes it computationally expensive for flexible objects such as a rope or a snake where one would need a large number of links and joints (in our simulations we have used up to 40 links) for realistic animations.

In another different approach, appropriate continuous curves were used to *approximate* the 'backbone' of a rigid link hyper-redundant manipulator. Motion planning is done with the continuous curve and then the rigid link robot is *fitted* to the updated curve. The backbone curves were chosen as splines [9] and linear combination of modes [10]. The main drawback in this approach is that the motion planning is done on the curve and hence the axial length of the hyper-redundant manipulator can only be *approximately* preserved and it is not clear how efficient the method is for simulating motion and animation of flexible objects such as ropes, snakes and human hair.

In a set of papers in early 1990's, Reznik and Lumelsky [11–13] present a sensor-based motion planning algorithm for highly redundant manipulator based on a classical curve called the *tractrix* [14]. They show that the tractrix curve has the attractive property of uniformly distributed motion with the motion 'dying' out from one end ('head') to the other end ('tail'). They pointed out that for n links, the algorithm has a complexity of $\mathcal{O}(n)$, i.e., it is linear in the number of links. For the case of a manipulator with one end fixed, they proposed an iterative algorithm to obtain the joint motions for a given end-effector motion. Additionally, they proposed strategies to avoid collision with obstacles based on sensing data. More recently, there is renewed interest in simulation and realistic animation of flexible objects such as ropes and strings due to the need for developing real-time simulation of suturing and knot tying in microsurgery simulations (see, for example, [15–17]). In Ref. [18], the authors have developed a 'follow the leader' approach to simulate tying of knots in microsurgery where the suture or string is discretised into large number of rigid links. One end of the string is held fixed and the other end is moved in a manner to tie a desired knot. The 'follow the leader' approach is similar to the tractrix based approach – in the tractrix based approach the motion of an intermediate link is such that the motion of the 'head' is *along* the link ahead of it whereas in the 'follow the leader approach' the motion of the intermediate links 'follows' the link ahead of it. In this paper, we revisit the tractrix based algorithm. We extend the tractrix equations to the case when the 'head' moves along an arbitrary direction in the plane or in 3D space and present a general algorithm for simulating motion of large hyper-redundant systems. We clearly show that the computation of the motion of the links can be done in real-time since it involves evaluation of simple algebraic, trigonometric and hyperbolic functions. We present a slightly different algorithm, in comparison to the work reported in [12], for the case of a serial robot in which one end has to be fixed. The simulations of the motion of a snake, tying of knots with a rope and a solution of the inverse kinematics of a planar hyper-redundant manipulator clearly show that the tractrix based strategy yields a more 'natural looking' motion.

The paper is organised as follows: in Section 2 we present an overview of the tractrix curve and its attractive properties. In Section 3, we first extend the notion of the tractrix when the head moves along an arbitrary direction in a plane and then to spatial 3D motion. We present an algorithm to obtain the points on the tractrix when the end of a link moves in 3D space. In Section 4, we present an algorithm based on the tractrix, to resolve redundancy in hyper-redundant systems. In Section 5, we present numerical simulation results for a snake, tying of knots with one and two hands, and resolution of redundancy of a hyper-redundant manipulator. We briefly discuss the advantages of the tractrix based approach in comparison to a 'physics' based approach. Finally, we present the conclusions in Section 7.

2. An overview of the tractrix curve

Historically, the tractrix arose in the following problem posed to famous German mathematician Leibniz: What is the path of an object starting of with a vertical offset when a string of constant length drags it along a straight horizontal line? By associating the object with a dog, the string with a leash, and the pull along a horizontal line with the dog's master, the curve has the descriptive name *hund* curve (hound curve) in German. Leibniz found the curve using the fact that the axis is an asymptote to the tractrix [14]. The above concept of the curve traced by the dog is also valid for a a single link moving in the plane and was first recognized by Reznik and Lumelsky [11]. If the 'head' of the link, denoted by j_1, is made to move along a straight line parallel to the X-axis, the path traced by the 'tail', denoted by j_0, when the *velocity of* j_0 *is along the link*, is a tractrix shown by the dotted curve in Fig. 1b.

2.1. Properties of the tractrix curve

We lists some of the important properties of a tractrix curve which are the basis of the attractive features of the resolution scheme based on the tractrix curve (see also [11]).

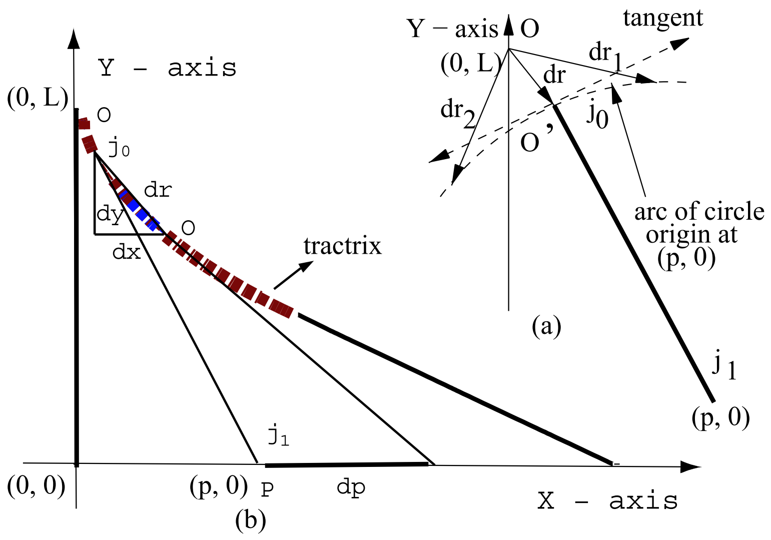

Fig. 1. Motion of a link when one end is pulled along a line parallel to X-axis.

- Given an infinitesimal displacement dp of j_1 parallel to the X-axis, the infinitesimal displacement of j_0, denoted by a vector dr, presents a *local minimum* of all possible paths for j_0 when dr is *along* the link. This follows from the following reasoning: Let the j_1 move to a point $(p,0)$ along the X-axis. The point j_0 can lie anywhere on a circle centered at $(p,0)$ with radius L where L is the length of the link. Fig. 1a shows three possible (exaggerated) infinitesimal displacements dr, dr_1 and dr_2 of the tail originally at point O with coordinates $(0,L)$. Clearly the infinitesimal displacement OO' or dr is shortest when it is normal to the tangent to the circle centered at $(p,0)$. Since the normal to the tangent to the circle is along the link, the vector dr is least when dr is along the link $j_0 - j_1$. Alternately dr is smallest when the velocity vector at j_0 is along the link. It maybe noted that we are dealing with infinitesimal displacements dr and not finite displacements.
- Using the fact that the velocity vector at j_0 is always aligned with the link, i.e., with the tangent to the tractrix, the tractrix equation can be derived from the differential equation of the tangent

$$\frac{dy}{dx} = -\frac{y}{\sqrt{L^2 - y^2}} \tag{1}$$

where L is length of the link. The above differential equation can be solved in *closed-form*, and we can write

$$x = L\ \log\frac{y}{L - \sqrt{L^2 - y^2}} - \sqrt{L^2 - y^2} \tag{2}$$

The solution of the differential equation can also be written in a parametric form, with p as the parameter, as

$$x(p) = p - L\ \tanh\left(\frac{p}{L}\right), \quad y(p) = L\ \mathrm{sech}\left(\frac{p}{L}\right) \tag{3}$$

- The ratio between dr and dp obeys an inequality $dr \leqslant dp$. The inequality follows from the following reasoning: Let (x,y) be the coordinates of point j_0, and $(p,0)$ be the coordinates of j_1. Since the link length L is constant, j_0 must lie on the circle $(x-p)^2 + y^2 = L^2$ (see Fig. 1a). From the equation of the circle, we get

$$p = x + \sqrt{L^2 - y^2} \tag{4}$$

and we can write

$$\frac{dp}{dx} = 1 - \frac{y}{\sqrt{L^2 - y^2}}\left(\frac{dy}{dx}\right) \tag{5}$$

The displacement dr can be written as $dr = \sqrt{dx^2 + dy^2}$, and using elementary calculus, we get

$$\frac{dr}{dp} = \frac{dr/dx}{dp/dx} = \frac{\sqrt{1 + (dy/dx)^2}}{1 - \frac{y}{\sqrt{L^2 - y^2}}(dy/dx)} \tag{6}$$

and using the equation of the tractrix (1), we get dr/dp as

$$\frac{dr}{dp} = \frac{\sqrt{L^2 - y^2}}{L} \leqslant 1 \tag{7}$$

where we get an equality if the link is lying on the X-axis and moves along it with $y = 0$.

- The location of the tail j_0 for a given motion of the head j_1, parallel to the X-axis, can be computed in terms of hyperbolic functions as shown in Eq. (3).

3. Extension of the tractrix to spatial motion

We first consider the case of the head j_1 moving along an arbitrary straight line, not necessarily the X-axis, given by $y_e = mx_e$ where $m = y_p/x_p$ is the slope of the line connecting the initial position of the head and the destination point of the head (x_p, y_p). The differential equation for the tangent can now be written as

$$\frac{dy}{dx} = \frac{y - y_e}{x - x_e} \tag{8}$$

From the length constraint, $L^2 = (x - x_e)^2 + (y - y_e)^2$, we can solve for x_e and we get,

$$x_e = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \tag{9}$$

where $A = 1 + m^2, B = 2my + 2x, C = x^2 + y^2 - L^2$. From the above expression for x_e, we can see that there are two possible values of x_e for every x and y. The positive sign is used when the slope of the link (m_1), with respect to a new coordinate system with the path of the head as the X-axis is negative and vice versa. Substituting the expressions obtained for x_e from Eq. (9) and $y_e = mx_e$ in Eq. (8), and integrating it we get the tractrix shown in Fig. 2.

The equations describing a tractrix can be extended to 3D space. In 3D space we will have two differential equations of the form

$$\begin{aligned} \frac{dy}{dx} &= \frac{y - y_e}{x - x_e} \\ \frac{dz}{dx} &= \frac{z - z_e}{x - x_e} \end{aligned} \tag{10}$$

The equations of the path followed by head are

$$y_e = m_1 x_e, \quad z_e = m_2 x_e \tag{11}$$

where, $m_1 = y_p/x_p, m_2 = z_p/x_p$, and (x_p, y_p, z_p) is the destination point of the head. It may be noted that the above equations assumes that the link is initially lying along Y-axis; however, similar equations can be obtained if the link is along the Z or the X-axis. We also have the constraint of length preservation

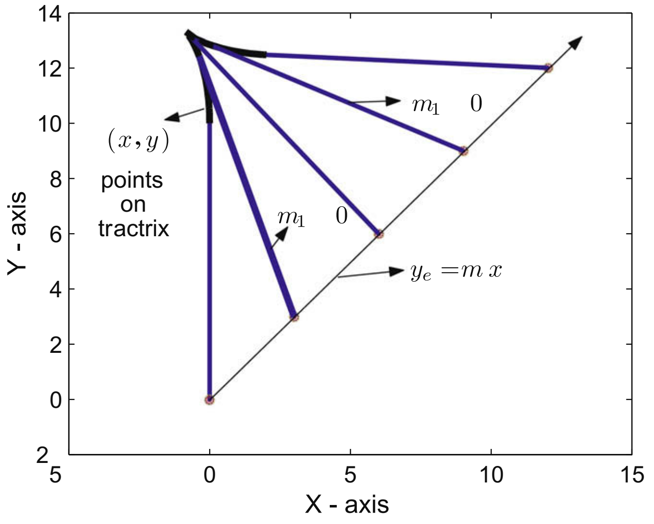

Fig. 2. Motion of a link when one end is pulled along the line $y_e = mx_e$.

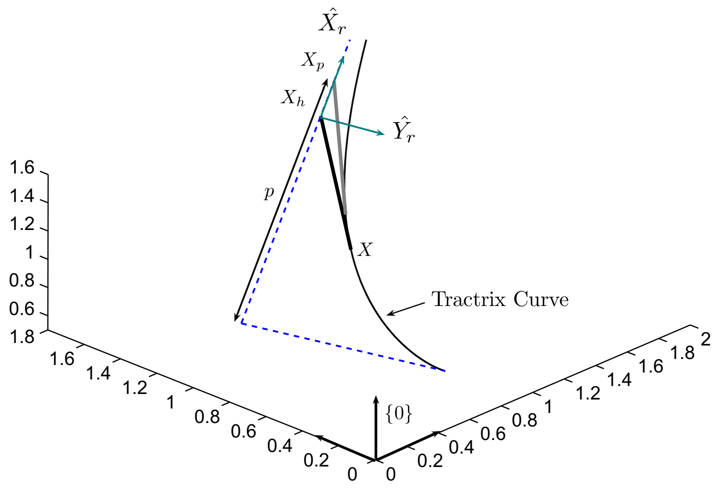

Fig. 3. Tractrix in global and local coordinates.

$$L^2 = (x - x_e)^2 + (y - y_e)^2 + (z - z_e)^2 \tag{12}$$

One obvious way to obtain the tractrix in 3D space would be to numerically integrate the differential Eqs. (10) and obtain the path taken by the tail in 3D space. This would be computationally intensive especially if we have to numerically integrate large number of equations resulting from discretising a flexible object into many rigid links. Instead of the numerical solution of the differential equations, we present an improved algorithm for obtaining the point on the tractrix in 3D space. For this purpose a reference plane is constructed using the three points, namely the initial positions of head, tail and the destination point of the head denoted by $\mathbf{X}_p = (x_p, y_p, z_p)^T$. The X-axis of the reference plane is aligned with the path of head. In this reference plane, we can solve the 2D parametric Eqs. (3) of the tractrix and obtain the position of the tail (x_r, y_r) in the reference plane. To obtain the position of the tail in global coordinates, the points (x_r, y_r) are transformed from the reference (local) to the global coordinate system. These steps are presented below as an algorithm.

3.1. *Algorithm TRACTRIX3D*

(1) Define the vector $\mathbf{S} = \mathbf{X}_p - \mathbf{X}_h$ where $\mathbf{X}_h$ is the current location of the head and $\mathbf{X}_p$ is the destination point of the head.
(2) Define the vector $\mathbf{T} = \mathbf{X} - \mathbf{X}_h$ where $\mathbf{X} = (x, y, z)^T$ is the tail of the link lying on the tractrix.
(3) Define the new reference coordinate system $\{r\}$ with the X-axis along $\mathbf{S}$. Hence $\hat{\mathbf{X}}_r = \frac{\mathbf{S}}{|\mathbf{S}|}$.
(4) Define the Z-axis as $\hat{\mathbf{Z}}_r = \frac{\mathbf{S}\times\mathbf{T}}{|\mathbf{S}\times\mathbf{T}|}$.
(5) Define rotation matrix ${}^0{}_r[R] = [\hat{\mathbf{X}}_r \; \hat{\mathbf{Z}}_r \times \hat{\mathbf{X}}_r \; \hat{\mathbf{Z}}_r]$.
(6) The Y-coordinate of the tail (lying on the tractrix) is given by $y = \hat{\mathbf{Y}}_r \cdot \mathbf{T}$ and the parameter p can be obtained as $p = L\ \text{sech}^{-1}(\frac{y}{L}) \pm |\mathbf{S}|$.
(7) From p, we can obtain the X and Y-coordinate of the point on the tractrix in the reference coordinate system as

$$x_r = \pm|\mathbf{S}| - L\ \tanh\left(\frac{p}{L}\right)$$
$$y_r = L\ \text{sech}\left(\frac{p}{l}\right) \tag{13}$$

(8) Once x_r and y_r are known, the point on the tractrix $(x, y, z)^T$ in the global fixed coordinate system $\{0\}$ is given by

$$(x, y, z)^T = \mathbf{X}_h + {}^0_r[R](x_r, y_r, 0)^T \tag{14}$$

The above steps are illustrated in Fig. 3.

4. Algorithm for resolution of redundancy

The algorithm *TRACTRIX3D* can be used for resolution of redundancy for any serially discretised deformable object. Consider a deformable object such as a snake or a rope discretised into n rigid links $l_1, l_2, \ldots, l_n$ with joints $j_1, j_2, \ldots, j_{n-1}$ where j_i is the joint connecting link i and link $i+1$. For spatial motion, we assume that the links are connected by spherical joints and for planar motion, the joints are rotary.

Consider the last two links l_n and l_{n-1}. The head of the link l_n denoted by the point j_n is required to be moved to a new position[1] $j_{n_{\text{new}}}$ given by $(x_p, y_p, z_p)^T$. From the steps given in the algorithm *TRACTRIX3D* we can obtain the new displaced location of the tail point j_{n-1} denoted by $(x, y, z)^T$ as it follows a tractrix (see Eq. (14)). The link l_{n-1} is attached to the link l_n and hence the tail of the link l_n can be considered to be the head of the link l_{n-1}. The head of the link l_{n-1} should now be moved from its existing location to $(x, y, z)^T$. The location of the tail of link l_{n-1}, following a tractrix, can again be obtained from the steps given in algorithm *TRACTRIX3D*. It maybe noted that the reference plane and the rotation matrix obtained in the steps described in *TRACTRIX3D* are *not* the same for the two links. Following similar steps, we recursively obtain the motion of the head and tail of all links down to the first link l_1. We present this resolution scheme as an algorithm.

4.1. Algorithm RESOLUTION-TRACTRIX

```
(1) Input desired location of head of link $l_n$, i.e., the point $(x_p, y_p, z_p)^T$.
(2) $j_{n_{\text{new}}} = (x_p, y_p, z_p)^T$.
(3) For $i: n \rightarrow 1$.
(3.1)   Call TRACTRIX3D and obtain location of the tail of link $i$, i.e., obtain $(x, y, z)^T_{i-1}$.
(3.2)   Set new location of head of link $i-1$, i.e., $j_{i-1_{\text{new}}} \leftarrow (x, y, z)^T_{i-1}$.
```

We can make the following remarks about the above algorithm.

(1) The algorithm for resolution of redundancy has a complexity of $\mathcal{O}(n)$ where n is the number of rigid links. This follows from the observation that the computation of the tractrix, for a link in 3D space (see algorithm *TRACTRIX3D*) is determined by a constant number of vector cross and dot products, computation of two hyperbolic functions, and a constant number of 3×3 matrix multiplication and additions. The number of computations is not dependent on n and hence the complexity is $\mathcal{O}(n)$. This fact makes the algorithm amenable for real-time computations.

(2) Instead of a flexible object, such as a rope, being moved from the end, if it is moved from any point on the body, then we can divide the object into two parts and apply the steps listed in algorithm *TRACTRIX3D* to the two parts individually. This can be done in parallel computations.

(3) In case of a hyper-redundant robot, the joint angles can be easily obtained since the initial and final position of all the links are known. In case of planar hyper-redundant robots, the joint angle θ_i is given by

$$\theta_i = \cos^{-1}\left(\overrightarrow{j_{i-1}j_i}(k+1) \cdot \overrightarrow{j_{i-1}j_i}(k)\right) \tag{15}$$

where $\overrightarrow{j_{i-1}j_i}(k)$ is the unit vector from the tail to the head of the link at kth instant. In the case of spatial motion with spherical joints connecting the links, the rotation angles at the spherical joint can be obtained from rotation matrices at $k+1$ and kth positions. In comparison to a pseudo-inverse based method, the resolution of redundancy is done in *Cartesian space* and then the joint angles are computed.

(4) Under a tractrix motion, when the head of the link l_n moves by dr_n the displacements of all the links obey the inequality $dr_0 \leqslant dr_1 \leqslant \ldots \leqslant dr_{n-1} \leqslant dr_n$, with the equality $dr_i = dr_{i-1}$ reached *only* when the line of motion of joint j_i coincides with link l_i. This observation follows from Eq. (7). A consequence of this observation is that the motion of the links away from the head gets progressively smaller and appears to 'die' out.

(5) The property given in Eq. (7) also imply that for a tractrix motion, the *sum* of the motion of all links except the head, $\sum_{i=0}^{i=n-1} |dr_i|$, is minimised. The sum of all joint motions is also minimised. This is in contrast to pseudo-inverse based resolution of redundancy where the *infinitesimal* motion of the joints are minimised in the *least square* sense. The results obtained from the tractrix based resolution of redundancy are thus different from pseudo-inverse based methods.

4.2. Obstacle avoidance

While moving the links of the deformable object such as in tying a knot, we must ensure that the discretised links do not intersect or collide with each other. To effectively simulate the motion of the discretised links, we need to detect collisions between the links and then develop a strategy to manage the collisions. There exists a wide variety of algorithms in literature for collision detection (see, for example, the review paper [19] and the references contained there in). In our implementation, we have used a simple conservative strategy of bounding each link by a sphere and then checking for the distance between the centre of the spheres. If the distance is such that there is collision, we move the centre of one of the links along the common normal by a distance slightly greater that $(l_i + l_{i+1})/2$ where l_i is the length of the ith link. Unfortunately this motion can result in a collision at some other link. Hence, we must recursively follow a strategy of detection and collision management for all links. If the number of iterations exceeds a given value the object is considered to be locked at that configuration. This collision detection and management scheme makes the tractrix based resolution scheme loose its desired $\mathcal{O}(n)$ characteristics when applied to simulations where collisions and self-intersections can occur.

[1] We can discretise a 'large' step into 'smaller' steps for animation.

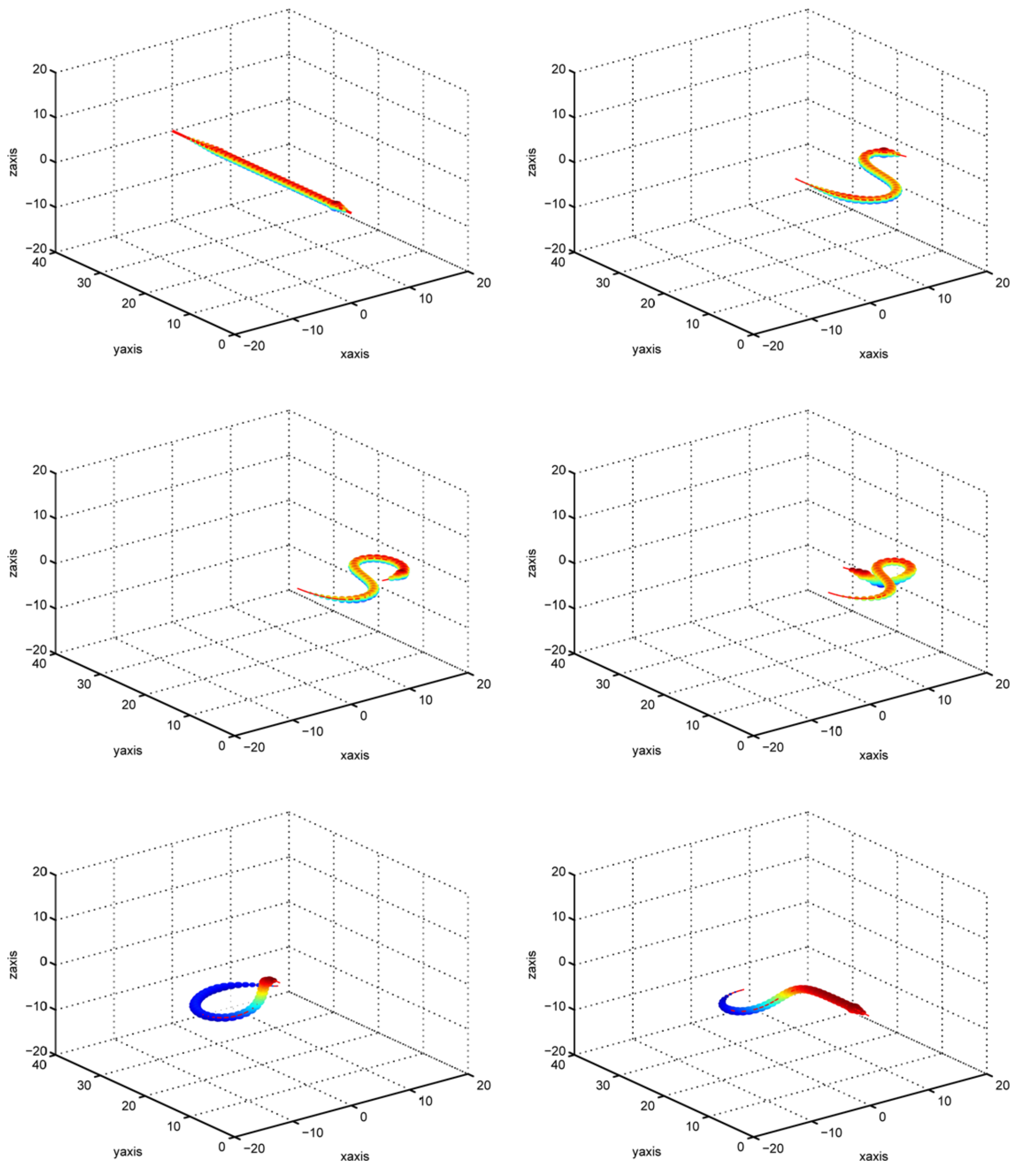

Fig. 4. Simulation of the motion of snake.

5. Simulation results and discussion

The tractrix based redundancy resolution has been applied to visualize the motion of a snake, tying a knot in a rope, and for solving the inverse kinematics of hyper-redundant planar manipulator. We present snap shots of animation results for each of these objects. The animations of a snake, tying of a single handed and two handed knot are available as supplementary material (see Appendix A).

5.1. Motion of a snake in 3D

We model a snake with 40 links. The head of the snake is moved along an arbitrary curve in 3D space and the motion of each of the subsequent links are obtained according to the tractrix strategy presented in Section 4. From the motion of the links, an animation is created using the commercial software Matlab [20]. Fig. 4 shows the configuration of the snake at few instances.

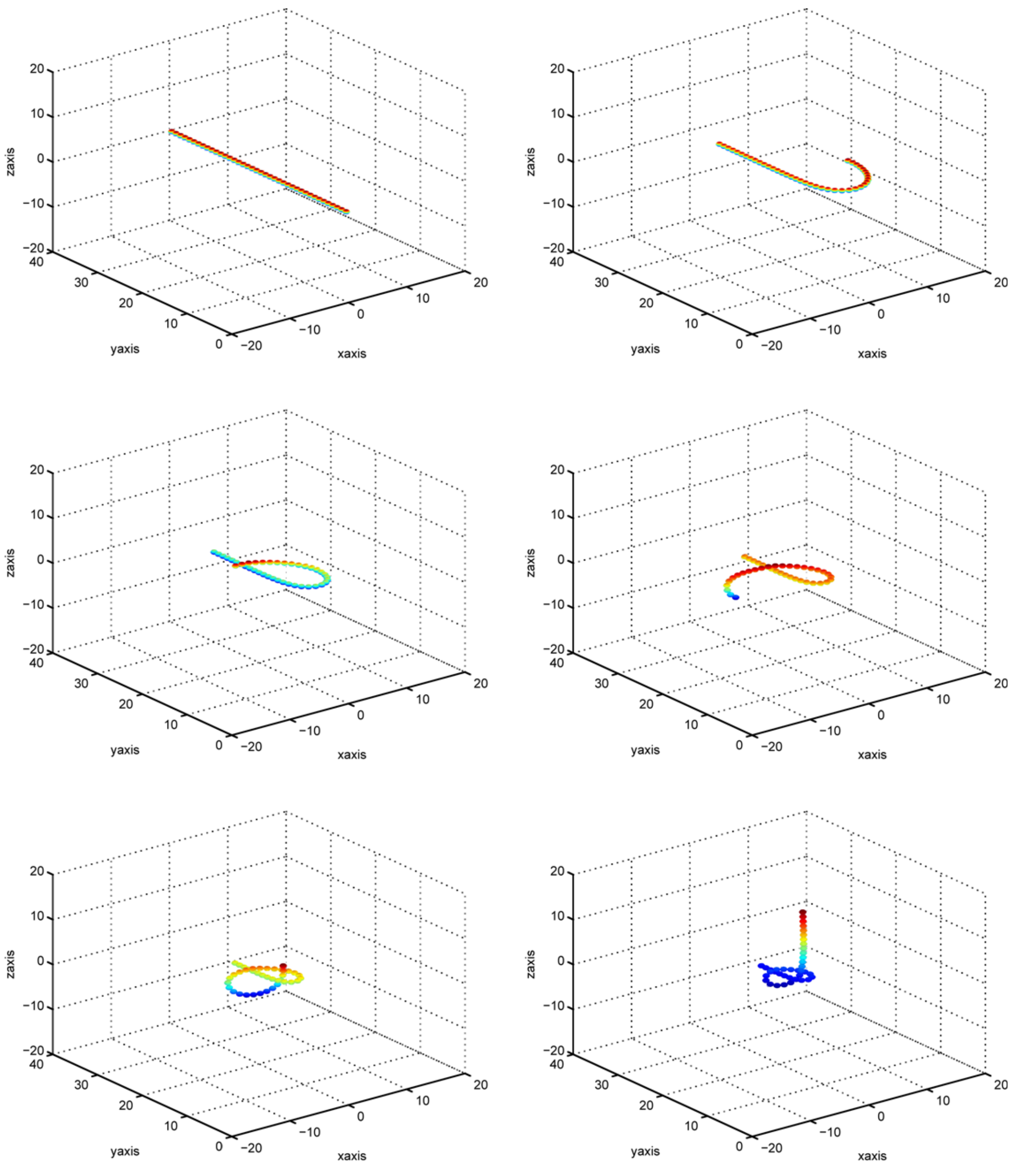

Fig. 5. Simulation of the single-handed knot tying.

5.2. A single-handed knot

We model a rope with 40 rigid links each one unit long. One end of the rope is moved in a fashion so that a knot is tied near the centre of the rope. Several configurations of the rope while tying the knot are shown in Fig. 5. An animation of the knot tying process was created in Matlab from the computed configurations and since the computations are fairly simple, this could be done in real-time.

5.3. Two-handed knot

In case the knot is to be tied by moving both the ends of the rope (by moving both hands) then the two ends are moved alternatively. Again, the motion of all the links are obtained in real-time by using the tractrix based approach. Various configurations of the rope while two-handed knot tying are shown in Fig. 6. It was observed that the collision avoidance takes most of the time in the simulation of single-handed and two-handed knots.

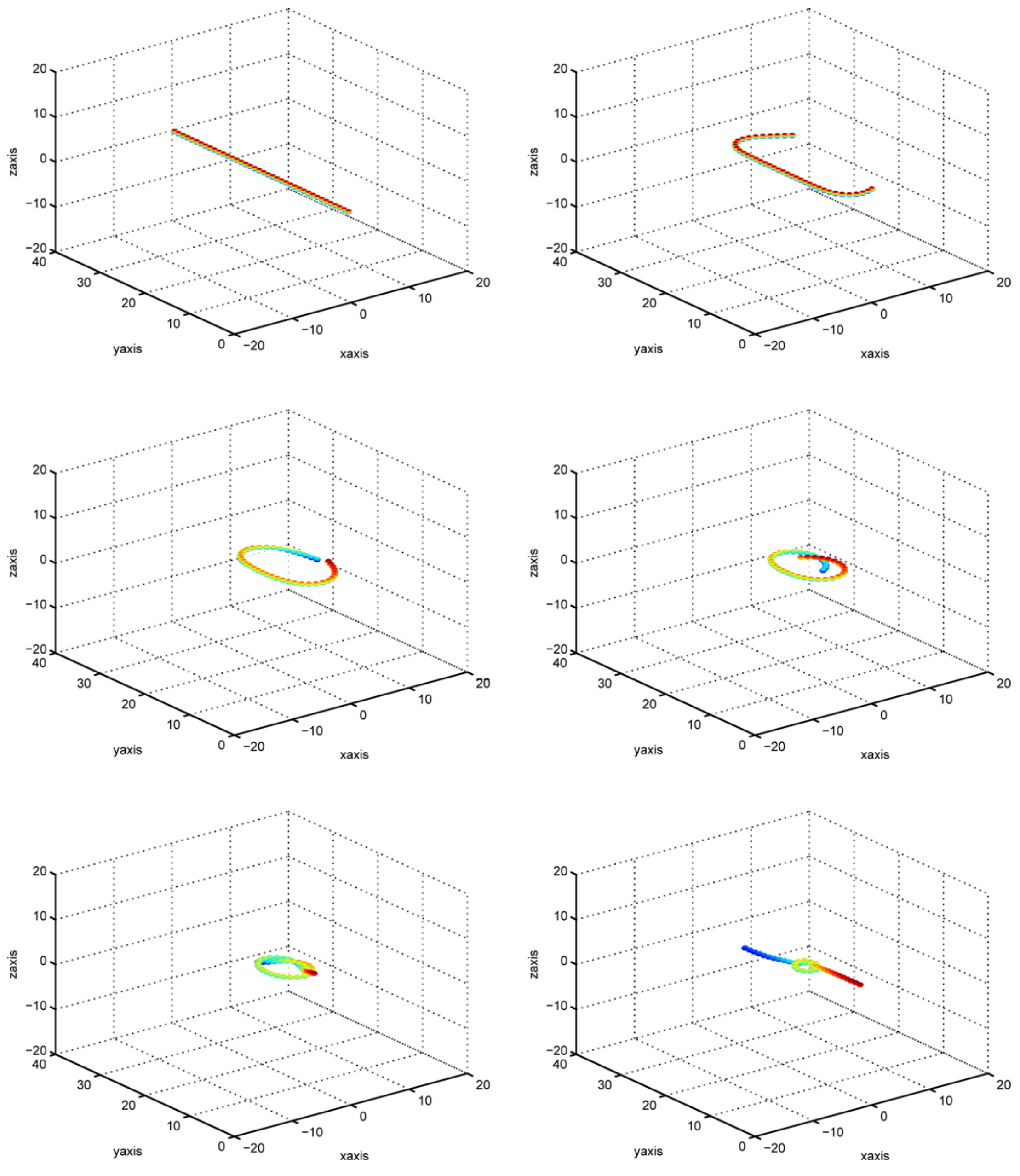

Fig. 6. Simulation of a two-handed knot tying.

5.4. Planar hyper-redundant manipulator

In the case of inverse kinematics of a hyper-redundant manipulator, the desired (x_p, y_p) of the end-effector (same as the head), *inside* the workspace of the manipulator, is specified. We obtain the motion of all the links by the tractrix approach. In a tractrix the tail of the first link also moves although by a small amount. However, for a manipulator the position j_0 (or the tail of the first link) should be fixed. To overcome this problem, we can adopt one of two strategies:

- We use the algorithm *RESOLUTION-TRACTRIX* till the tail of the *third* (or fourth) link and use the well-known inverse kinematics equations of planar 2R (3R) manipulator to solve for the joint angles of the first two (three) links.
- We compute the motion of all links using the algorithm *RESOLUTION-TRACTRIX* and find the location of the fixed base point j_0. Next we move j_0 to (0,0) and *all* other links are translated 'rigidly' with no rotations at the joints. This results in the end-effector moving away from the desired (x_p, y_p). We repeat the *RESOLUTION-TRACTRIX* till the head reaches (x_p, y_p) and the

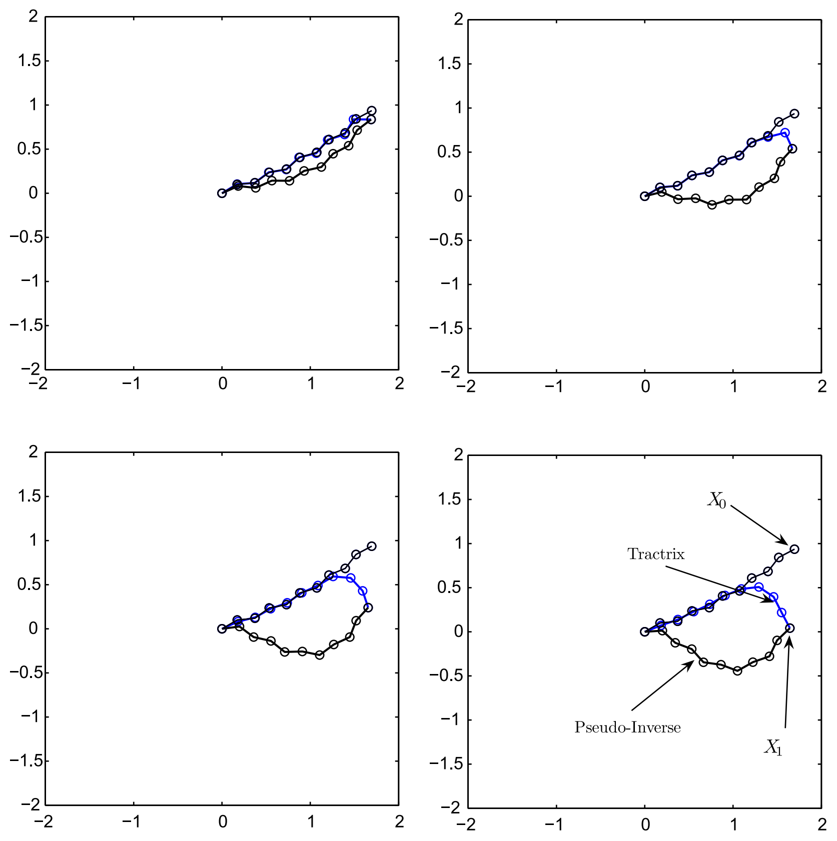

Fig. 7. Intermediate configurations of the planar hyper-redundant manipulator.

point j_0 is within a prescribed error bound of (0,0). It may be noted that convergence of this iterative process is guaranteed[2] since the motion of the links dies down as we progress from the head to the tail (see observation listed in the previous section) and the rigid translation of the entire manipulator will tend to zero with iteration.

In practice the second approach works better and not more than three iterations were required in the several simulations we tried. This is because in many simulations, the first two (three) links are 'stretched out' in a singular configuration, and the closed-form inverse kinematics solutions of a planar $2R$ (or $3R$) manipulator do not work very well when the links are fully stretched out. Fig. 7 gives intermediate configurations of a 10 link hyper-redundant planar manipulator where each link length is 0.2 units. The initial configuration and the final (desired) end-effector (x, y) coordinates are as marked in Fig. 7. The intermediate configuration with blue markers are obtained from the tractrix based approach and the configuration with black markers are from a pseudo-inverse formulation. The plot of joint rotations from the tractrix and pseudo-inverse based approaches are shown in Fig. 8.

6. Discussion

The theoretical development of the tractrix based resolution scheme (see observations (4) and (5) in Section 4) and the animations of motion of snakes and tying of knots clearly show that in the tractrix based resolution of redundancy, the motion tends to 'die out' from the head to the tail. For simulation of flexible objects such as ropes, snakes and human hair, this feature gives a more 'natural' and 'realistic' visualization of the motion of the flexible object. However, the 'natural' motion is *not* from a 'physics' based approach. There exists a fair amount of literature where authors have proposed models of flexible objects such as human hair and ropes with discretised rigid links and associated springs and dampers (see, for example, the work by Rosenblum et al. [21], Phillips et al. [22] and the references contained in them). The equations of motion of the rigid multi-body system are then solved using one of the many available $\mathcal{O}(n)$ algorithms for the forward dynamics, and it is conceivable that a proper choice of spring stiffness and damping can lead to a motion which 'dies out' from the head to the tail.

[2] The convergence is guaranteed if the desired point (x_p, y_p) can be *reached* by the hyper-redundant manipulator.

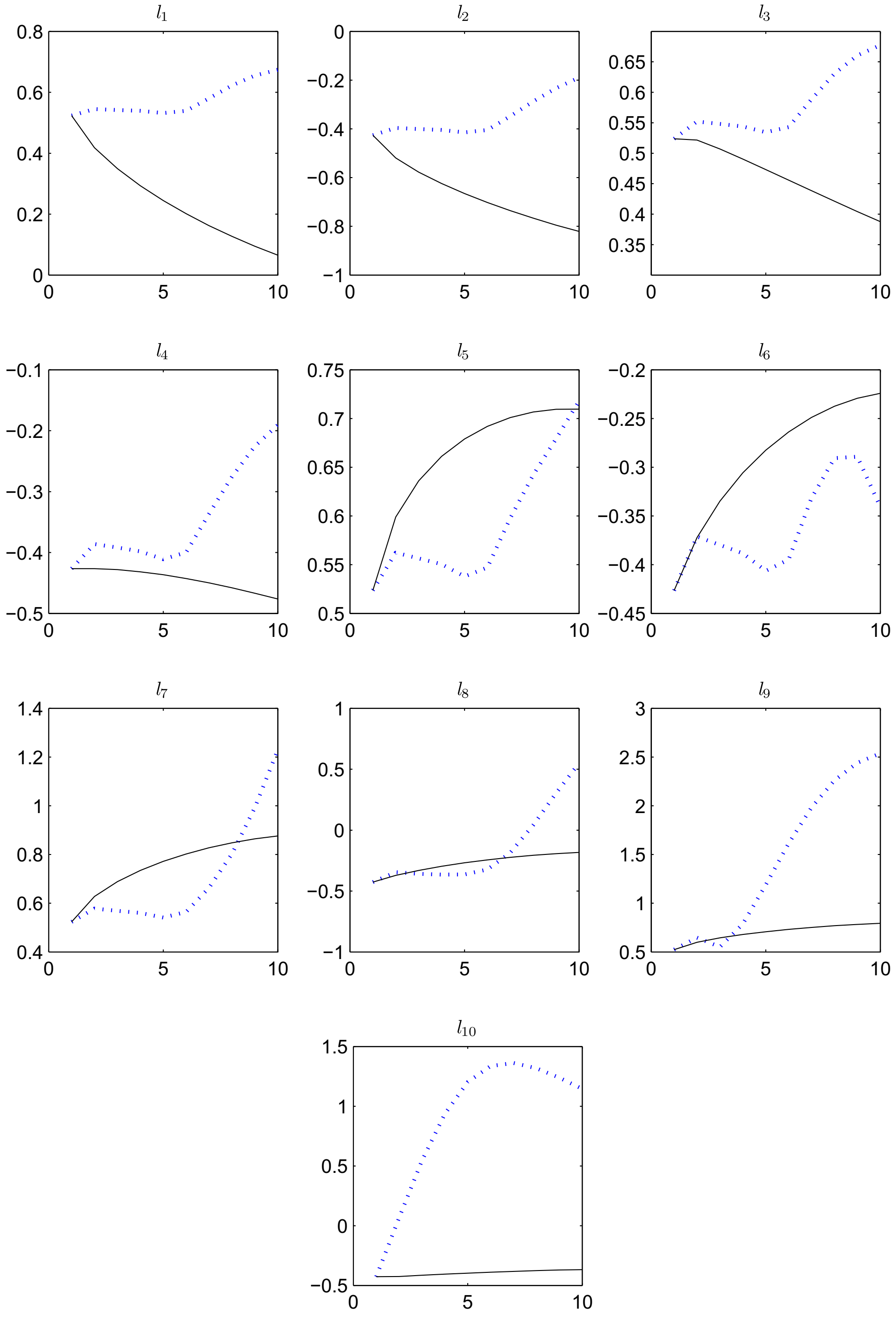

Fig. 8. Joint rotations in the planar hyper-redundant manipulator: dotted blue tractrix, continuous black pseudo-inverse. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

In all these ’physics’ based approaches, it is, however, not clear how to select appropriate values of spring stiffness or damping coefficients. In contrast, in the tractrix based approach no choices have to be made and no dynamic equations need to be solved. In addition, since it involves only simple matrix operations and evaluation of hyperbolic and trigonometric functions, it is expected to be fast and efficient and amenable for real-time simulation and visualization of the motion of flexible objects.

The tractrix based approach appears to be purely kinematic in nature – for a given motion of the head dp, the motion of the tail dr is less than or equal to dp for each link. This central idea, however, can be given a ’physics flavour’ and one can argue that the tractrix based approach is *not* completely an ad-hoc or a ’non-physics’ based approach. Consider the situation where all the mass of the discretised links are lumped at the head of the link. Since dp/dt and dr/dt are the velocities of the head and the tail of a link, observations (4) and (5) of Section 4 can be related to the decrease of the kinetic energy as one traverses the (discretised) flexible object from the head to the tail. In this sense, the tractrix based approach can be related to the minimisation of kinetic energy.

In the case of hyper-redundant manipulators, observations (4) and (5) of Section 4 is a very desirable feature. In all hyper-redundant serial manipulators, the joints and actuators toward the fixed base ’see’ larger inertia and the first joint and actuator sees the largest inertia – the first actuator must be capable of moving all the links. It is very desirable strategy to move the joints and actuators nearer the base the least. The simulations shown in Figs. 7 and 8 clearly show that the joints towards the fixed base rotate the least.

7. Conclusion

In this paper, we have used a classical planar curve, namely the tractrix, and its extension to 3D space for resolution of redundancy in serial multi-body systems. The resolution scheme can be used for for hyper-redundant manipulators and simulation and visualization of motion of flexible objects such as snakes and ropes where the axial length of the flexible object is preserved. In case of flexible objects such as snakes and ropes, the object is discretised into a sufficiently large number of rigid links and for an arbitrary chosen motion of the head of one link, the motion of all other links are computed by using the equations of a tractrix. In case of hyper-redundant manipulators, once the motion of the links are known in Cartesian space, the joint angles can be computed using simple dot products or from a rotation matrix and, in this sense, the resolution of redundancy is at the Cartesian position level.

One of the key property of a tractrix is that the motions of links decreases as one goes away from the head and this makes the visualization of tying knots and motion of a snake ’natural’ and more realistic. In addition, since the computations involve simple vector algebra and evaluation of hyperbolic functions, the simulation and visualization can be done in real-time.

The tractrix based approach has been implemented for simulating and visualization of the motion of a snake, tying of knots with one or two hands and resolution of redundancy for a 10 link, planar, hyper-redundant manipulator. The simulations and visualizations clearly show the advantages of a tractrix based approach.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.mechmachtheory.2009.10.005.

References

[1] C.R. Rao, S.K. Mitra, Generalised Inverse of Matrices and its Applications, Wiley, 1971.
[2] J. Baillieul, J. Hollerbach, C.W. Brockett, Programming and control of kinematically redundant manipulators, in: Proceedings of the 23rd IEEE Conference on Decision and Control, 1984, pp. 768–774.
[3] A. Liegeois, Automatic supervisory control of the configuration and behavior of multi-body mechanisms, IEEE Transactions, Systems, man and Cybernetics 7 (1977) 868–871.
[4] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, in: Proceedings of the IEEE International Conference on Robotics and Automation, 1985, pp. 500–505.
[5] J.M. Hollerbach, K.C. Suh, Redundancy resolution of manipulators through torque optimisation, IEEE Transactions, Journal of Robotics and Automation 3 (1987) 308–316.
[6] T. Yoshikawa, Manipulability of robotic mechanisms, The International Journal of Robotics Research 4 (1985) 3–9.
[7] C.A. Klein, C.H. Huang, Review of the pseudo-inverse for control of kinematically redundant manipulators, IEEE Transactions on Systems, Man, and Cybernetics SMC-13 (3) (1983) 245–250.
[8] Y. Nakamura, Advanced Robotics: Redundancy and Optimization, Addison-Wesley, 1991.
[9] K.E. Zanganeh, J. Angeles, The inverse kinematics of hyper-redundant manipulators using splines, in: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 3, 1995, pp. 2797–2802.
[10] G.S. Chirikjian, J.W. Burdick, A modal approach to hyper-redundant manipulator kinematics, IEEE Transactions on Robotics and Automation 10 (3) (1994) 343–354.
[11] D. Reznik, V. Lumelsky, Motion planning with uncertainty for highly redundant kinematic structures: I. Free snake motion, in: Proceedings of the 1992 IEEE/RSJ, International Conference on Intelligent Robots and Systems, 1992, pp. 1747–1752.
[12] D. Reznik, V. Lumelsky, Sensor-based motion planning for highly redundant kinematic structures: II. The case of a snake arm manipulator, in: Proceedings of the 1993 International Conference on Robotics and Automation, vol. 3, 1993, pp. 889–894.
[13] D. Reznik, V. Lumelsky, Sensor-based motion planning in three dimensions for a highly redundant snake robot, Advanced Robotics 9 (3) (1995) 255–280.
[14] <http://mathworld.wolfram.com/Tractrix.html>.
[15] U. Kühnapfel, H.K. Cakmak, H. Maaß, Endoscopic surgery training using virtual reality and deformable tissue simulation, Computers and Graphics 24 (2000) 671–682.
[16] H. Kang, J.T. Wen, Robotic knot tying in minimally invasive surgery, in: Proceedings of the IEEE/RSJ, International Conference on Intelligent Robots and Systems, 2002, pp. 1421–1426.

[17] J. Brown, S. Sorkin, J.-C. Latombe, K. Montgomery, M. Stephanides, Algorithmic tools for real-time microsurgery simulation, Medical Image Analysis 6 (3) (2002) 289–300.
[18] J. Brown, J.-C. Latombe, K. Montogomery, Real-time knot-tying simulation, The Visual Computer 20 (2004) 165–179.
[19] P. Jimenez, F. Thomas, C. Torras, 3D collision detection: a survey, Computers and Graphics 25 (2) (2001) 269–285.
[20] Matlab Users Manual, The MathWorks, Inc., 1992.
[21] R.E. Rosenblum, W.E. Carlson, E. Tripp, Simulating the structure and dynamics of human hair: modeling, rendering and animation, Journal of Visualization and Computer Animation 2 (4) (1991) 141–148.
[22] J. Phillips, A. Ladd, L.E. Kavaraki, Simulated knot tying, in: Proceedings of the IEEE, International Conference on Robotics and Automation, vol. 1, 2002, pp. 841–846.