Redundancy resolution using tractrix – simulations and experiments

V C Ravi*, Subrata Rakshit† and Ashitava Ghosal‡
Department of Mechanical Engineering
Indian Institute of Science
Bangalore, 560012, India

Abstract

Hyper-redundant robots are characterized by the presence of a large number of actuated joints, many more than the number required to perform a given task. These robots have been proposed and used for many applications involving avoiding obstacles or, in general, to provide enhanced dexterity in performing tasks. Making effective use of the extra degrees of freedom or resolution of redundancy has been an extensive topic of research and several methods have been proposed in literature. In this paper, we compare three known methods and show that an algorithm based on a classical curve called the tractrix leads to a more 'natural' motion of the hyper-redundant robot, with the displacements diminishing from the end-effector to the fixed base. In addition, since the actuators nearer the base 'see' a greater inertia due to the links farther away, smaller motion of the actuators nearer the base results in better motion of the end-effector as compared to other two approaches. We present simulation and experimental results performed on a prototype eight link planar hyper-redundant manipulator.

1 Introduction

A rigid body has three degrees of freedom when moving in a plane and six degrees of freedom when moving in three dimensional space. In general, one can position and orient the end-effector of a planar/spatial manipulator in its workspace, with three/six actuated joints. A hyper-redundant robot or manipulator is constructed with a larger number of links and actuated joints than are required for the required tasks – more than three for planar and more than six for spatial motion. This results in the well-known difficulty in the kinematic analysis of hyper-redundant manipulators, viz. given a desired motion of the end-effector or a point on the manipulator, there exist an *infinite* number of solutions for the motion of the joints. The problem of obtaining or choosing a unique solution from this infinite set is called *resolution of redundancy*. There exists a vast amount of literature, starting in the early 1980's, on this topic. In the recent past, there has been a renewed interest in this subject due to its possible application in modeling, simulating and predicting protein conformations[1]. We briefly review the main existing approaches and present the not-so-widely-used tractrix based approach in more detail in the next section.

One of the first approaches that was tried for the resolution of redundancy involved the use of the *least-squares* technique and its variants. In this approach, first the end-effector linear and angular velocities are related to the joint rates by the well-known manipulator Jacobian matrix. In the case of a redundant manipulator, with more joint variables than Cartesian or task space variables, the manipulator Jacobian matrix is non-square. In the second step, for a given end-effector velocity, the joint rates are obtained by computing a pseudo-inverse[2] of the non-square manipulator Jacobian matrix. In the final step, the joint values are updated by using a finite difference approximation of the computed joint rates. In its basic form, the pseudo-inverse based method is at the level of velocity and has the interesting property of minimizing joint rates in a least-square sense. In other variants, the pseudo-inverse method has been applied with a weighting matrix, applied at the joint acceleration level and extended to include a null-space term for optimizing additional desired quantities such as singularity [3], joint limits [4] and obstacle avoidance [5], to minimize joint torques [6] or to maximize a manipulability index [7] (for details of various pseudo-inverse based schemes, see the review paper by Klein and Huang [8] and the textbook by Nakamura [9] and the references contained therein). The pseudo-inverse based resolution scheme is a purely numerical and a local approach. Moreover, since it involves

*Centre for AI and Robotics, Bangalore, India, Email: vcravi@gmail.com

†Centre for AI and Robotics, Bangalore, India, Email: srakshit@cair.drdo.in

‡Corresponding author. Email: asitava@mecheng.iisc.ernet.in,
Paper No. DETC2009-86934, IDETC/CIE 2009

[1]Proteins are linear hetero bio-polymers comprising of amino acid residues [1]. A simplistic and *classical* physics based model of a protein backbone consists of 50 to 500 amino acid residues connected by two degree-of-freedom joints and hence can be thought of as a large hyper-redundant serial manipulator.

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 hyper-redundant manipulators with a large number of links and joints or for problems relating to protein kinematics.

In the second approach, called the *modal* approach, appropriate continuous curves are used to *approximate* the 'backbone' of a 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 [10] and use linear combination of modes [11]. The main drawback in this approach is that the motion planning is done on the backbone curve and hence the axial length of the hyper-redundant manipulator can only be *approximately* preserved. It is also not clear how computationally efficient the method is when the number of links and joints is very large.

In a set of papers in the early 1990's, Reznik and Lumelsky [12, 13, 14] present a sensor-based motion planning algorithm for a highly redundant manipulator, based on a classical curve called the tractrix. They show that the tractrix curve has the attractive property of uniformly distributed motion, with the motion 'dying' out from the end-effector to the fixed end. They proposed a real-time, $\mathcal{O}(n)$, iterative algorithm to obtain the joint motions for a given end-effector motion. Additionally, they proposed strategies to avoid collisions with obstacles, based on sensing data obtained from sensors on the robot. In this paper, we revisit the tractrix based resolution scheme and show that the motion at the joints indeed dies out. We propose a slightly different algorithm for resolution of redundancy for a hyper-redundant manipulator and present simulations and experiments which compare the results obtained using pseudo-inverse, modal and the tractrix based approaches. In any robot the joints nearer the base must be able to move all outward links and joints and thus 'see' large inertias. Hence the 'dying' out property of a tractrix is advantageous as the joints nearer the base move the least. This is shown in simulations and experiments done using a prototype 8-link planar hyper-redundant manipulator.

The paper is organized as follows: in section 2 we present a brief overview of the tractrix curve and list its attractive properties. In section 3, we first extend the notion of the tractrix to a case where the head moves along an arbitrary direction in a plane and then to spatial 3D motion. In section 4, we present an algorithm based on the tractrix to resolve redundancy in hyper-redundant manipulators. In section 5, we present numerical simulation results on resolution of redundancy of a 8-link hyper-redundant manipulator. We compare the results obtained from the tractrix based resolution scheme with those obtained from the pseudo-inverse and modal approaches. In section 6, we present experimental results obtained from a prototype 8-link, planar hyper-redundant manipulator and show that the motion based on a tractrix is superior to that obtained by the two other methods. Finally, we present the conclusions in section 7.

2 A brief review of the tractrix curve

Historically, the tractrix curve arose in the following problem posed to the famous German mathematician Leibniz: What is the path of an object starting off with a vertical offset when a string of constant length drags it along a 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 X axis is an asymptote to the tractrix [15].

The above concept of the curve traced by the dog is also valid for a a single link moving in the plane, as first recognized by Reznik and Lumelsky [12, 13, 14]. In figure 1, if the head P, denoted by j_1, is made to move along a straight line ST parallel to the X-axis, the motion of the tail, denoted by j_0, such that the velocity of j_0 is *along* the link, is a tractrix curve.

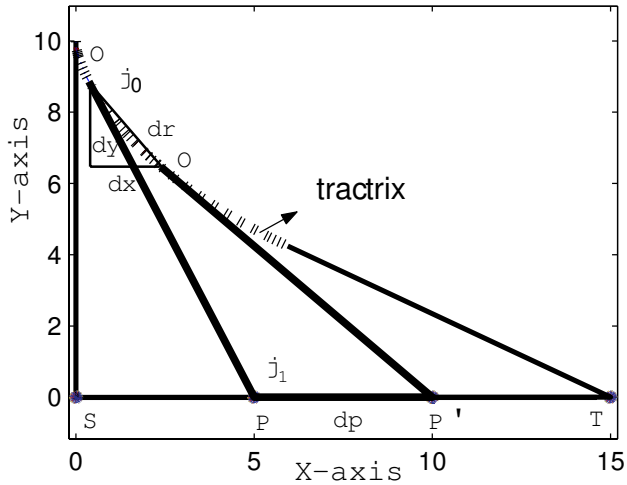

Figure 1: Motion of a link when one end is pulled along the line ST parallel X-axis

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 as

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

where L is the 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

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

2.1 Some important 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 curves. These follow from the differential equation (1) and its closed-form solutions given in equations (2) and (3)(see also [12, 13]).

- Since the instantaneous motion of the tail j_0 is directed along the link, the following optimality property holds: given an infinitesimal displacement dp of j_1 along ST, the length of the path traversed by the tail j_0, denoted by a vector dr, presents a local minimum of all possible paths for j_0.

- Furthermore, the ratio between dr and dp obeys an inequality $dr \leq dp$. The inequality follows from the following reasoning:

  Let x and y be the coordinates of point j_0, and p be the x-coordinate of j_1. From figure 1 we get

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

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

  $$\frac{dr}{dx} = \frac{L}{\sqrt{L^2 - y^2}} \tag{5}$$

  and dr/dp can be obtained as

  $$\frac{dr}{dp} = \frac{\sqrt{L^2 - y^2}}{L} \leq 1 \tag{6}$$

  where we get an equality if the link is along the line ST.

- The location of the tail j_0 for a given motion of the head j_1 along ST can be computed in terms of hyperbolic functions as shown in equation (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 given by $y_e = mx_e$ where m is the slope of the line connecting the initial position of the head and the destination point of the head. We define the new X' axis along the desired path of the head (see figure 2). The differential equation for the tangent can now be written as

$$\frac{dy}{dx} = \frac{y - y_e}{x - x_e} \tag{7}$$

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

$$x_e = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \tag{8}$$

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 ($X' - Y'$) is negative and vice versa. Substituting the expressions obtained for x_e from equation (8) and $y_e = mx_e$ in equation (7), and integrating it we get the tractrix curve shown in figure 2. It may be noted that as the head moves along X' axis from $(0, 0)$, initially the tail goes *backward* along the curve **1**. After the point A, where the link is perpendicular to X' axis, the tractrix follows the path **2**. The two paths meet at a cusp as shown in figure 2.

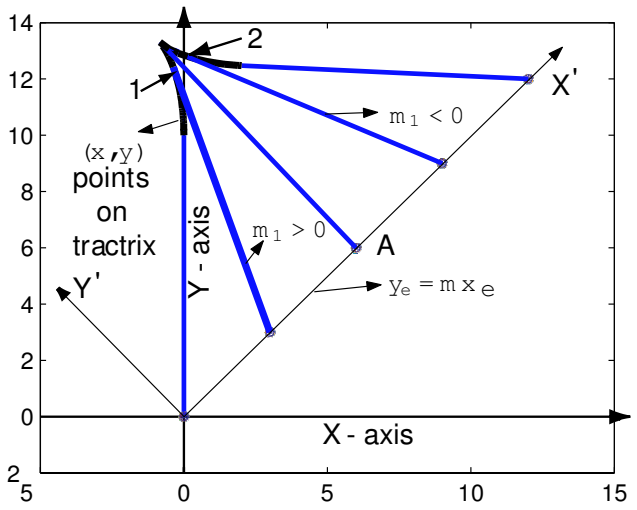

Figure 2: Motion of a link when one end is pulled along the line $y_e = mx_e$

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

equations of the form

$$\begin{aligned} \frac{dy}{dx} &= \frac{y-y_e}{x-x_e} \\ \frac{dz}{dx} &= \frac{z-z_e}{x-x_e} \end{aligned} \tag{9}$$

The equations of the path followed by the head are

$$y_e = m_1 x_e, \quad z_e = m_2 x_e \tag{10}$$

where, $m_1 = y_p/x_p$, $m_2 = z_p/x_p$, and $\mathbf{X}_p = (x_p, y_p, z_p)^T$ is the destination point of the head. It may be noted that the above equations assume that the link is initially lying along the 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

$$L^2 = (x-x_e)^2 + (y-y_e)^2 + (z-z_e)^2 \tag{11}$$

One obvious way to obtain the tractrix in 3D space would be to numerically integrate the differential equations (9) after using equation (9) and (10) to eliminate x_e, y_e and z_e. The solution will give the path taken by the tail in 3D space. This would be computationally intensive, especially if we have to numerically integrate a large number of these equations for many rigid links. Instead of the numerical solution of the differential equations, we present next an algorithm for obtaining points on the tractrix in 3D space.

First, we define a reference plane using the initial positions of the head and 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 the head. In this reference plane, we can solve the 2D parametric equations (3) of the tractrix and obtain the the position of the tail (x_r, y_r) in the reference plane. To obtain the position of the tail in global co-ordinates, the points (x_r, y_r) are transformed from the reference (local) to the global co-ordinate system. These steps are presented below as an algorithm.

Algorithm *TRACTRIX3D*

1 Define the vector $\mathbf{S} = \mathbf{X}_p - \mathbf{X}_h$ where $\mathbf{X}_h$ is the current location and $\mathbf{X}_p$ is the destination 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 $\mathbf{S}$ and Z-axis as $\hat{\mathbf{Z}}_r = \frac{\mathbf{S}\times\mathbf{T}}{|\mathbf{S}\times\mathbf{T}|}$. Finally define rotation matrix ${}^0_r[\ R\ ] = [\hat{\mathbf{X}}_r \quad \hat{\mathbf{Z}}_r \times \hat{\mathbf{X}}_r \quad \hat{\mathbf{Z}}_r]$.

4 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 $p = L\ \mathrm{sech}^{-1}(\frac{y}{L}) \pm |\mathbf{S}|$.

5 From p, obtain the X and Y coordinate of the point on the tractrix in the reference coordinate system as

$$\begin{aligned} x_r &= \pm|\mathbf{S}| - L\tanh(\frac{p}{L}) \\ y_r &= L\ \mathrm{sech}(\frac{p}{L}) \end{aligned} \tag{12}$$

6 From x_r and y_r obtain the point on the tractrix $(x, y, z)^T$ in the global fixed coordinate system $\{0\}$ as

$$(x, y, z)^T = \mathbf{X}_h +^0_r [\ R\ ](x_r, y_r, 0)^T \tag{13}$$

4 Algorithm for resolution of redundancy

The algorithm *TRACTRIX3D* can be used for resolution of redundancy for any serial hyper-redundant manipulator. Consider a hyper-redundant manipulator with n rigid links $l_1, l_2, ..., l_n$ with joints $j_1, j_2, ..., j_{n-1}$ where j_i is the joint connecting link l_i and link l_{i+1}. For spatial motion, we assume that the links are connected by spherical joints and for planar motion, the joints are assumed to be 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[2] $j_{n_{\text{new}}}$ given by $\mathbf{X}_p = (x_p, y_p, z_p)^T$. From the steps given in the algorithm *TRACTRIX3D* we can obtain the new displaced location of the tail point j_{n-1} denoted by $\mathbf{X} = (x, y, z)^T$ as it follows a tractrix (see equation (13)). 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.

Algorithm *RESOLUTION-TRACTRIX*

1 Input desired location of head of link l_n, i.e., the point $(x_p, y_p, z_p)^T$ and set $j_{n_{\text{new}}} = (x_p, y_p, z_p)^T$.

2 for $i : n \rightarrow 1$

2.1 Call *TRACTRIX3D* and obtain location of the tail of link i, i.e., obtain $(x, y, z)^T_{i-1}$

[2] We can discretize a 'large' step into 'smaller' steps for numerical simulation.

2.2 Set new location of head of link $i-1$, i.e., $j_{i-1_{\text{new}}} \leftarrow (x,y,z)^T_{i-1}$

3 At the end of step **2**, the tail of the first link, i.e., j_0, would have moved although by a small amount. To *fix* j_0 at the origin

3.1 Move j_0 to the origin $(0,0,0)^T$ and translate 'rigidly' *all* other links with no rotations at the joints. At the end of the 'rigid' translation, the end-effector will not be at the desired (x_p, y_p, z_p).

3.2 Repeat step **2** and **3** until the head reaches (x_p, y_p, z_p) and the point j_0 is within a prescribed error bound of $(0,0,0)$.

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. This fact makes the algorithm amenable to real time computation.

2) The joint angles can be easily obtained since the initial and final position of all the links are known. The joint angle θ_i is given by

$$\theta_i = \cos^{-1}(\overrightarrow{j_{i-1}j_i}(k+1) \cdot \overrightarrow{j_{i-1}j_i}(k)) \qquad (14)$$

where $\overrightarrow{j_{i-1}j_i}(k)$ is the unit vector from the tail to the head of the i-th link at k-th 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 $k+1$-th and k-th 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.

3) 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 $dr_0 \leq dr_1 \leq ... \leq dr_{n-1} \leq dr_n$, 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 equation (6). A consequence of this observation is that the motion of the links away from the end-effector gets progressively smaller and appears to 'die' out as we move towards the first link. This is a desirable feature for hyper-redundant manipulators since the joints towards the base see larger inertia and a desirable strategy would be to move these joints the least.

4) The property given in equation (6) also implies that for a tractrix motion, the *sum* of the motion of all links except the end-effector, $\sum_{i=0}^{i=n-1} |dr_i|$, is minimized. If the manipulator Jacobian is not singular, one can show that the sum of all joint motions is also minimized. This is in contrast to pseudo-inverse based resolution of redundancy where the *infinitesimal* motion of the joints are minimized in the *least squares* sense. The results obtained from the tractrix based resolution of redundancy are thus different from pseudo-inverse based methods.

5) To 'fix' the tail of the first link, iterations as described in step **3** needs to be performed. It may be noted that convergence of this iterative process is guaranteed if the desired end-effector position $(x_p, y_p, z_p)^T$ can be *reached* by the hyper-redundant manipulator. This is due to the above mentioned property, namely that the motion of the links dies down as we progress from the end-effector to the first link and the motion of j_0 to $(0,0,0)^T$ and the rigid translation of the entire manipulator will tend to zero with each iteration. In our simulations, two or three iterations were always found to be enough to bring j_0 to $(0,0)$ within an error bound of 10^{-3} mm.

5 Simulation results

The tractrix based redundancy resolution has been applied to a 8-link planar hyper-redundant planar manipulator whose links are 70 mm long. For the planar hyper-redundant manipulator, the desired (x_p, y_p) of the end-effector, *inside* the workspace of the manipulator, is specified. We obtain the motion of all the links by three approaches, namely the *pseudo-inverse*, *modal* and the *tractrix* based approach described in section 4.

We performed simulations for two representative motions – a) a set of arbitrarily chosen five straight line motions of end-effector and b) a closed circular trajectory of the end-effector as shown in figure 4. In the figure 4, the final configuration of the hyper-redundant robots obtained using the three approaches are also shown. The three plots in figure 5 show the joint angles for the straight line trajectory, computed using the pseudo-inverse, modal and tractrix approaches, respectively. For all the three approaches, the entire trajectory (including the motion from the home position) was discretised into 1022 steps and the total time taken for the motion was 176 seconds. The three plots in figure 6 show the joint angles for a circular trajectory. In the case of the circular trajectory, the trajectory (including the straight line motion from the home position) for all three approaches was discretised into 858 steps and the total time taken for the motion was 152 seconds. In the plots, joint 1 is

the base joint or the 'tail' and joint 8 is the end-effector or the 'head'.

We can make the following observations from the simulation results.

a) The joint values obtained from the three approaches for the same Cartesian trajectory are widely different. As a consequence, the final configurations obtained are very different for the three approaches.

b) The motion of the first two joints is least in tractrix based method. The motion of the last two joints are larger than in the other two approaches. This is as expected.

c) In all three approaches, for the closed circular trajectory of the end-effector, the joint values are not the same at the start and end of the motion.

d) The time taken to compute the joint variables by all the three approaches are quite different. On a Pentium IV PC, the calculations for the modal approach require approximately 0.24 seconds for the straight line trajectories and approximately 1.8 seconds for the circular trajectory. In contrast, the tractrix based approach takes approximately 0.03 seconds for the straight line and 0.24 seconds for circular trajectory. The pseudo-inverse approach takes the least amount of time – approximately 0.006 seconds for straight line and 0.06 seconds for the circular trajectory. This is probably due to the fact that the pseudo-inverse of matrices is implemented very efficiently in Matlab [16] using optimised C language codes, whereas our implementation of modal and tractrix based approaches in Matlab has large overheads which have not been optimized. In addition, the number of links are not very large, and we expect that the tractrix based approach would be faster when the number of links is much larger.

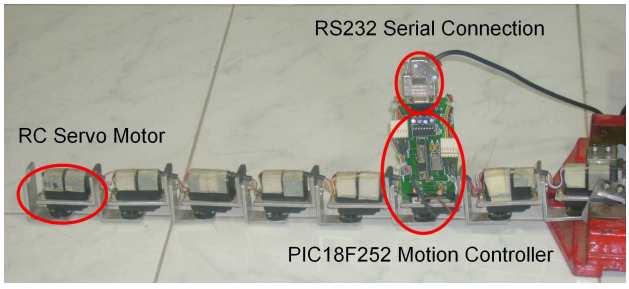

Figure 3: Experimental 8-link hyper-redundant manipulator.

6 Experimental results

To validate the numerical simulation results, we have built a prototype 8-link planar hyper-redundant manipulator. This is described next.

6.1 Experimental Set-up

The experimental prototype set-up consists of an 8-link hyper-redundant robot with single degree-of-freedom revolute joints connecting two links. Each link consists of two 70 mm long L-angle aluminum brackets joined back-to-back. One bracket holds the motor while the other connects to the motor in the previous link. The robot was clamped at one end and placed on a smooth (tiled) floor to reduce friction between the robot and the ground during the motion. In its current version, the robot does not have wheels and the links slide on a smooth floor. In addition, the mechanical fabrication is not very sophisticated with no compensation for the dynamics of the robot motion. Consequently, the tracking accuracy is quite low; however, it is good enough to provide a reasonable visualization of the implementation of the pseudo-inverse, modal and tractrix based resolution of redundancy in hyper-redundant robots.

The joints are driven by standard Futaba S3003 RC hobby servos [17, 18]. The RC servos take a command position in the form of a PWM pulse with a nominal time period of 30 ms; the pulse width determines the commanded position of the motor. The motors have a 90° range of motion on either side of the nominal position. A pulse width of 1 ms corresponds to the $-90°$ position and a pulse width of 2 ms corresponds to the $+90°$ position, with the motor centered at 1.5 ms. The motors have a potentiometer on the output shaft and an integrated closed-loop controller to maintain the position according to the input pulse. The controller electronics do not provide an intrinsic interface for higher level monitoring of the motor position. A custom designed PIC18F252 micro-controller-based board is used to generate the command pulses for all the 8 RC servo motors. The board also has an RS232 serial interface port to communicate with a PC.

The experimental 8-link hyper-redundant manipulator is shown in figure 3. During the motion, we placed a ink marker pen at the end-effector and the obtained motion is traced by the marker on the white tiled floor. We also chose the desired trajectory such that the links do not intersect or collide with each other and the end-effector trajectory can be realized with the $\pm 90°$ restriction on the motion of the rotary joints.

6.2 Results

A video of the full motion of the end-effector was taken for all three resolution schemes and for the two kinds of representative trajectories used for numerical simulations, namely, a) a straight line motion from a home position followed by a set of five straight line trajectories and b) a straight motion from a home position followed by a closed circular trajectory. The desired time of the entire motion was same as in numerical simulations, i.e., 176 seconds and 152 seconds for the straight and circular case, respectively. In figure 7, we present snapshots of the hyper-redundant manipulator, following the sequence of desired straight line trajectories. The snapshots show the configuration of the hyper-redundant manipulator at the final position and a trace of the actual trajectory of the robot using the pseudo-inverse, modal approach and tractrix based approaches. Likewise in figure 8, we present snapshots of the hyper-redundant manipulator at the end of the desired circular trajectory, using the three approaches.

From the video and the snap shots it can be clearly seen that the joint angles obtained by each of the three approaches are very different. In particular, we can observe that the joints nearer to the fixed base move the least in the tractrix based scheme (see bottom most snapshots in figures 7) and 8. It is also clear from the video that the tractrix based resolution scheme gives a smoother motion when compared to the modal approach. Even though we were not able to traverse the circle completely due to hardware limitations, the circular arc obtained from the tractrix based scheme looks closest to the desired trajectory and the motion of the joints are smoother for the arbitrary linear segments. This is expected since the joints closest to the base, which see the maximum inertia and frictional forces, move the least in the tractrix based resolution scheme.

7 Conclusion

In this paper, we use a classical tractrix curve and its extension to 3D space for resolution of redundancy in serial multi-body systems. The resolution scheme has been used for hyper-redundant manipulators. For an arbitrary chosen motion of the end-effector, the motion of all other links are computed by using the equations of a tractrix. Once the motion of the links are known in Cartesian space, the joint angles are computed using simple dot products and in this sense, the resolution of redundancy is at the Cartesian position level.

One of the key properties of a tractrix is that the motion of the links decreases as one moves from the end-effector towards the base. This is very desirable for hyper-redundant manipulators since the actuators near the fixed base 'see' a greater inertia. A second very attractive feature of the tractrix-based scheme is that the computations involve simple vector algebra and evaluation of hyperbolic functions, thereby making it amenable to real-time computations.

The tractrix based algorithm was tried out on an prototype 8-link hyper-redundant manipulator. The results obtained appear to be superior to those obtained using the pseudo-inverse and modal approaches. For future we intend to fine tune the experimental set-up and carry out more precise experiments.

Acknowledgment

The authors would like to thank their colleagues Sartaj Singh and Nikhil Mahale for developing the hardware. The hyper-redundant manipulator was made possible by funding from CAIR, DRDO, Govt. of India.

References

[1] Branden, C. and Tooze, J. (1999), *Introduction to Protein Structure*, Garland Publishing, New York.

[2] Rao, C. R. and Mitra, S. K. (1971), *Generalised Inverse of Matrices and its Applications*, Wiley.

[3] Baillieul, J., Hollerbach, J. and Brockett, C. W. (1984), 'Programming and control of kinematically redundant manipulators', *Proc. of 23rd IEEE Conf. on Decision and Control*, pp. 768-774.

[4] Liegeois, A. (1977), 'Automatic supervisory control of the configuration and behavior of multi-body mechanisms', *IEEE Trans., Systems, man and Cybernetics*, **7**, pp. 868-871.

[5] Khatib, O. (1985), 'Real-time obstacle avoidance for manipulators and mobile robots', *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 500-505.

[6] Hollerbach, J. M. and Suh, K. C. (1987), 'Redundancy resolution of manipulators through torque optimisation', *IEEE Trans., Jou. of Robotics and Automation*, **3**, pp. 308-316.

[7] Yoshikawa, T. (1985), 'Manipulability of robotic mechanisms', *The Int. Journal of Robotics Research*, **4**, pp. 3-9.

[8] Klein, C. A. and Huang, C. H. (1983), 'Review of the pseudo-inverse for control of kinematically redundant manipulators', *IEEE Trans. on Systems, Man, and Cybernetics*, **SMC-13**(3), pp. 245-250 .

[9] Nakamura, Y. (1991), *Advanced Robotics: Redundancy and Optimization*, Addison-Wesley.

[10] Zanganeh, K. E. and Angeles, J. (1995), 'The inverse kinematics of hyper-redundant manipulators using splines', *Proc. IEEE Int. Conf. on Robotics and Automation*, **3**, pp. 2797-2802.

[11] Chirikjian, G. S., Burdick, J. W. (1994), 'A modal approach to hyper-redundant manipulator kinematics', *IEEE Trans. on Robotics and Automation*,**10**(3), pp. 343-354.

[12] Reznik, D. and Lumelsky, V. (1992), 'Motion planning with uncertainty for highly redundant kinematic structures: I. "Free snake" motion', *Proc. of 1992 IEEE/RSJ, Int. Conf. on Intelligent Robots and Systems*, pp. 1747-1752.

[13] Reznik, D. and Lumelsky, V. (1993), 'Motion planning with uncertainty for highly redundant kinematic structures: II. The case of a snake arm manipulator', *Proc. of 1993 Int. Conf. on Robotics and Automation*, Vol. 3, pp. 889-894.

[14] Reznik, D. and Lumelsky, V. (1995), 'Sensor-based motion planning in three dimensions for a highly redundant snake robot', *Advanced Robotics*, **9**(3), pp. 255-280.

[15] http://mathworld.wolfram.com/Tractrix.html

[16] *Matlab Users Manual*, The MathWorks, Inc.

[17] www.futaba-rc.com/servos/servos.html Futaba Servos.

[18] www.futaba-rc.com Futaba S3003 Servo Tech Notes.

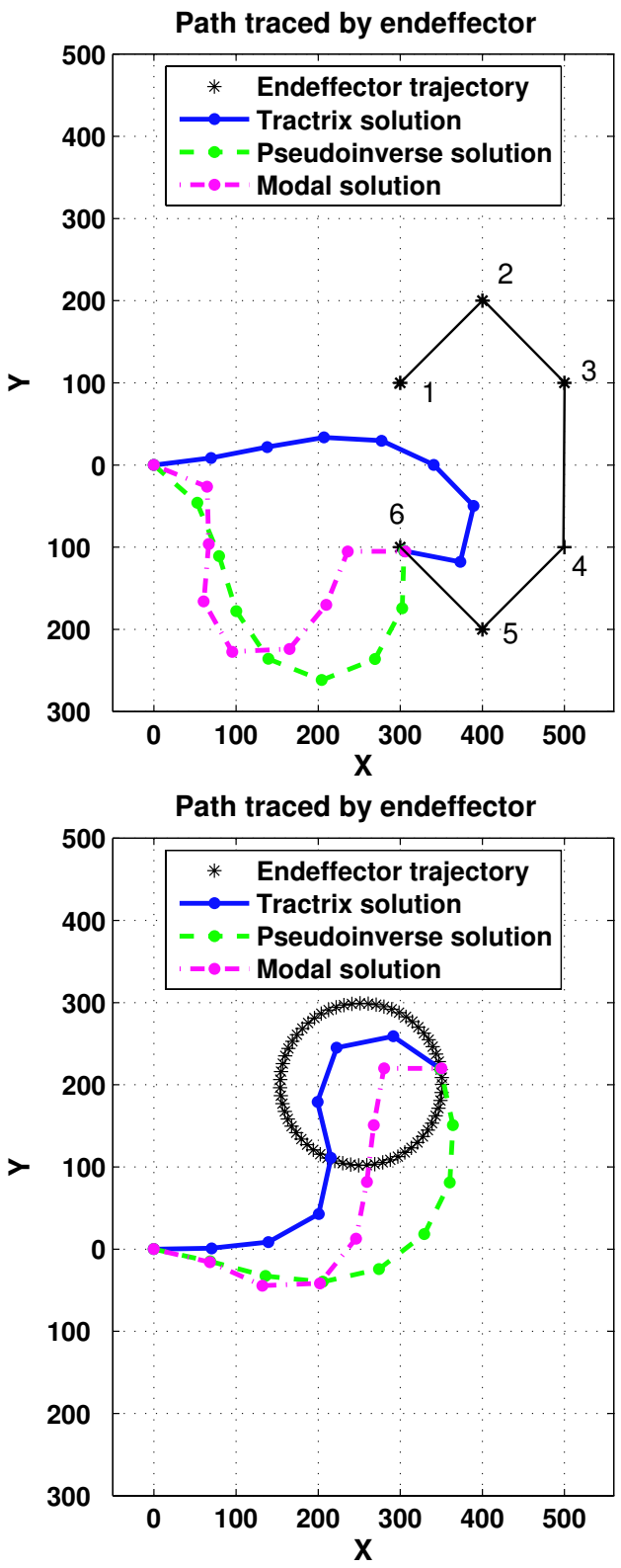

Figure 4: Desired straight line and circular end-effector trajectories.

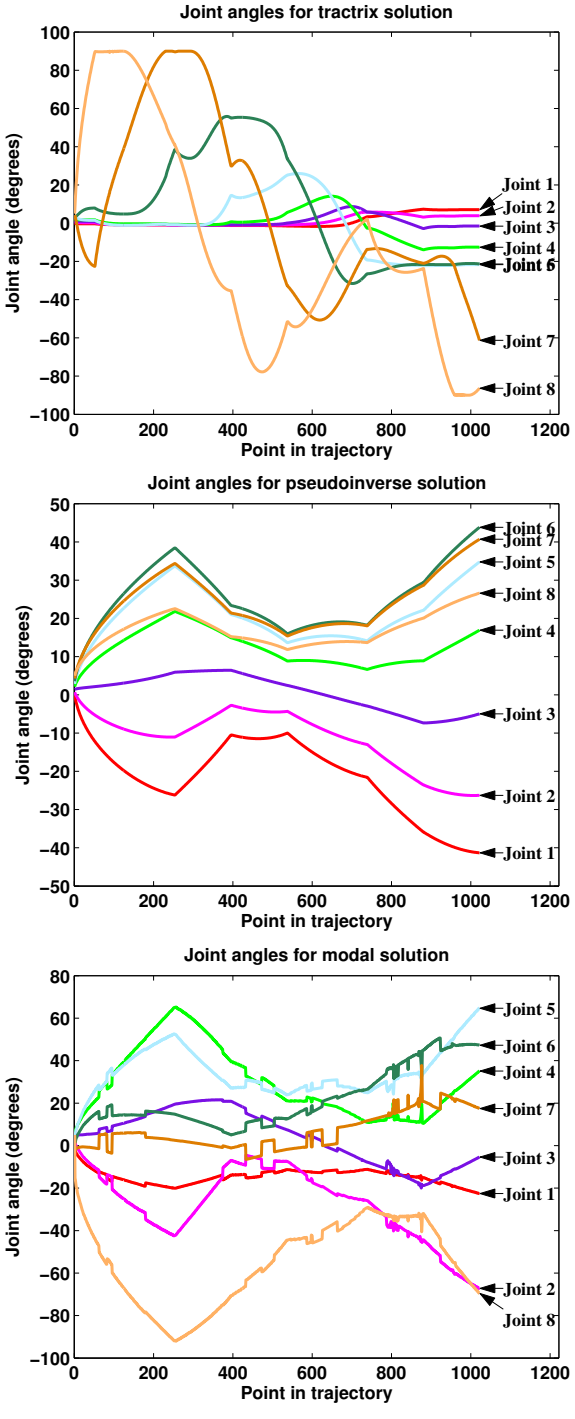

Figure 5: Plot of joint variables for straight line trajectories.

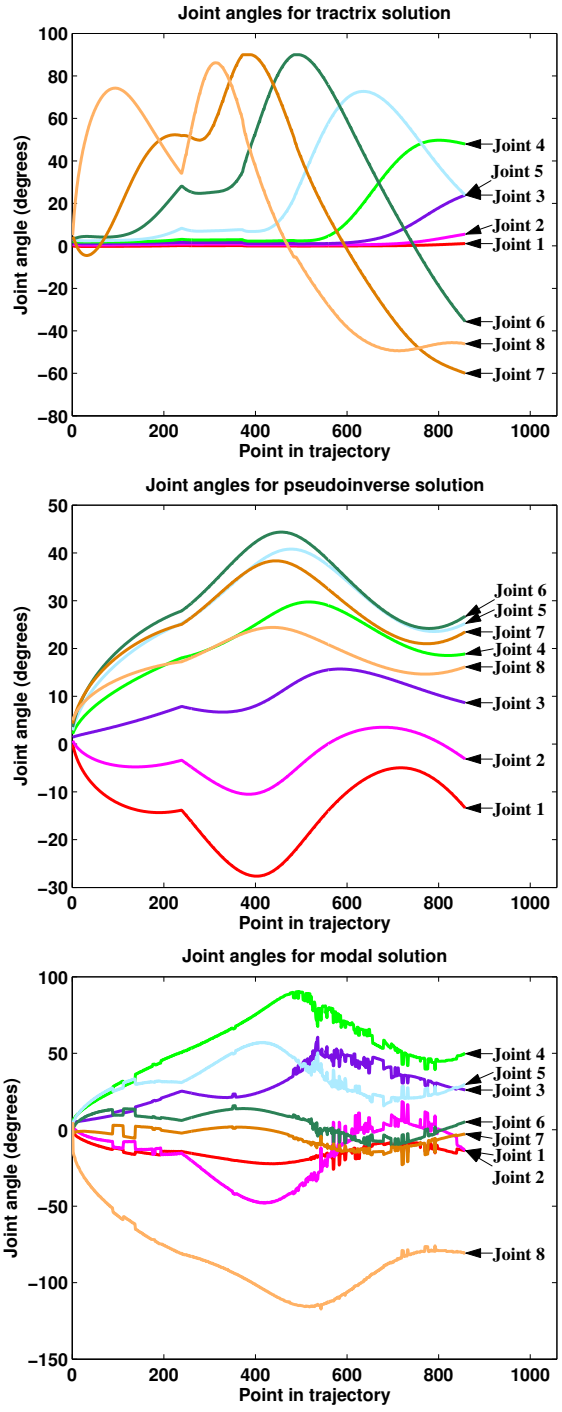

Figure 6: Plot of joint variables for circular trajectory.

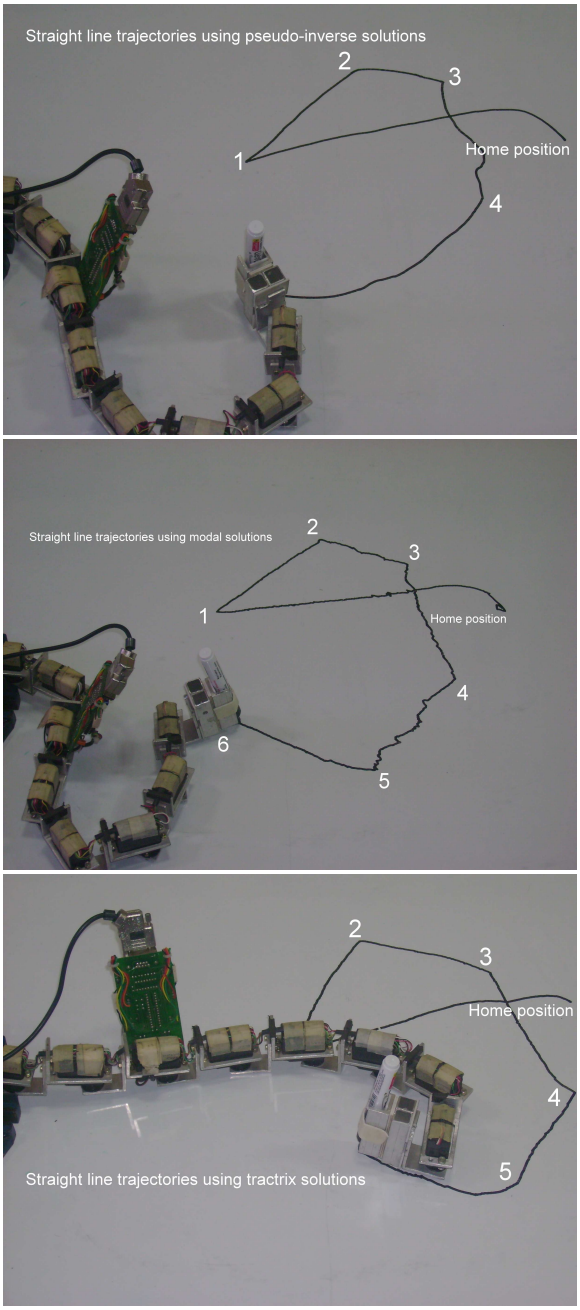

Figure 7: Snapshots of hyper-redundant manipulator executing straight line trajectories.

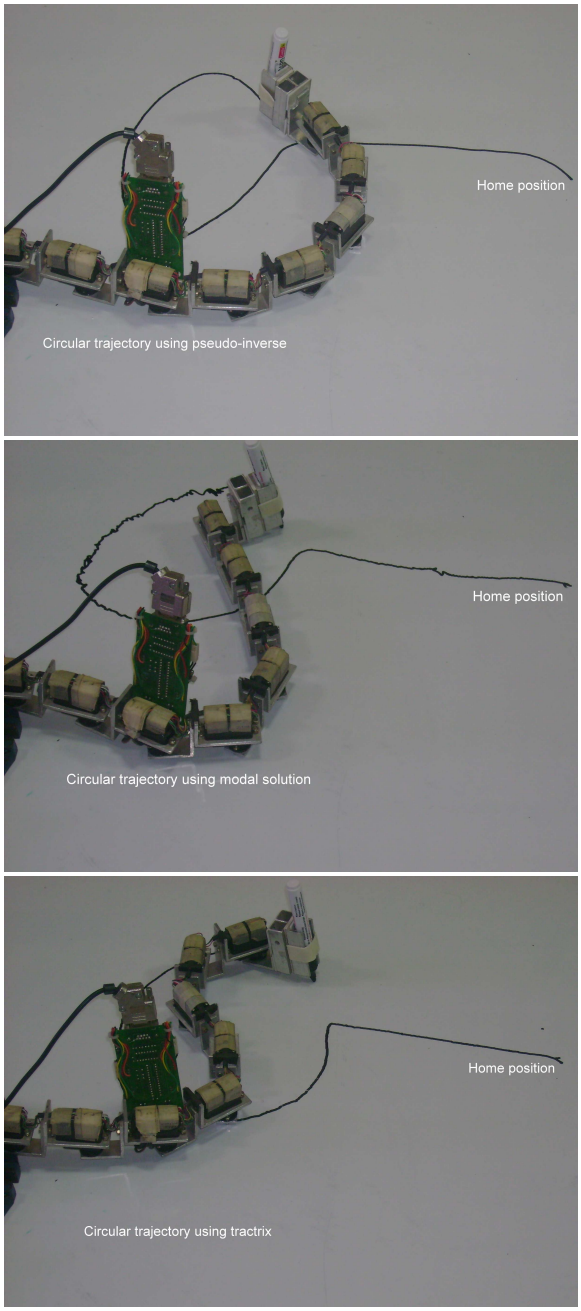

Figure 8: Snapshots of hyper-redundant manipulator executing circular trajectories.