Motion planning of hyper-redundant robot through confined ducts

K. P. Ashwin*, Arkadeep N. C. *, and A. Ghosal †

Abstract

??

Keywords:??,??,??.

1   Introduction

1.
- What are redundant robots

- What is the use of redundancy

- examples of applications of redundant robots

- what is the downside?

- what is called redundancy resolution. Say that the displacement of tip is specified and the co-ordinates(or the joint angles) of the subsequent links should be found out.

2.
- State that the major advantage of red. robots is obstacle avoidance

- What is the first redundacy resolution scheme? Did it avoid obstacles?

- How is obstacle avoidance included in red. resolution

- Give a review of obstacle avoidance algorithms

- state why tractrix is better, what is tractrix

---
*Graduate Student at the Robotics and Design Lab, Department of Mechanical Engineering, Indian Institute of Science, Bangalore 560012, India, email: ashwinkp@iisc.ac.in

†Corresponding Author, Professor, Department of Mechanical Engineering, Indian Institute of Science, Bangalore, email: asitava@iisc.ac.in.

- how obstacle avoidance is easily implemented

3.
- State that particularly interesting is the case of motion through ducts, endoscopy and inspection
- How the tractrix method as is is difficult to implement
- Explain the motive of this paper
- Contents of the paper

2 Overview of tractrix based motion planning

Consider a rigid link of length L_0 positioned in a 2D plane, initially aligned to the Y-axis as shown in the figure[ref fig]. The co-ordinates of the 'head' of the link is given as $\mathbf{X}_h = [X_h, Y_h]^T$ and the co-ordinates of the 'tail' as $\mathbf{X}_t = [X_t, Y_t]^T$. If the head is displaced to the co-ordinate $\mathbf{x}_h = [x_h, y_h]$ along the positive $X-$ axis by t units, the tail of the link can lie anywhere on the circumference of a circle centered at the co-ordinate $(t, 0)$ with radius L_0. If we assign a rule that the velocity of the tail of the link is always directed towards the length of the link, we get two diametrically opposite points on the circle. The continuous path traced by the tail point is the well known tractrix curve given by the expressions[ref]:

$$\mathbf{x}_t = [x(t), y(t)] = [t - L_0 \tanh \frac{t}{L_0}, L_0 \operatorname{sech} \frac{t}{L_0}] \tag{1}$$

The extension of tractrix equation along motion in arbitrary direction as well as an algorithm to calculate the same in 3-D can be found in [ref] and [ref]. In case of multiple links connected to each other as in the case of hyper-redundant robot, or a one dimensional object approximated as a series of connected linkages the algorithm can be applied iteratively from the head to tail as shown in [ref]. By moving along the tractix curve, the tail moves the minimum distance with respect to its initial position. Also, the displacement $\|\mathbf{x}_h - \mathbf{X}_h\| \geq \|\mathbf{x}_t - \mathbf{X}_t\|$ which means that the displacement attentuates from the displaced link to end of the chain in case of serially connected links[fig2]. Due to the minimal displacement of the tail, the motion can be imagined as the one with high lateral resistance on the link and less resistance in the direction of motion of link.

It is shown in [ref] that for a single rigid link, tractrix curve can also be obtained by minimizing an L^2 metric which is essentially the displacement of the tail from its initial

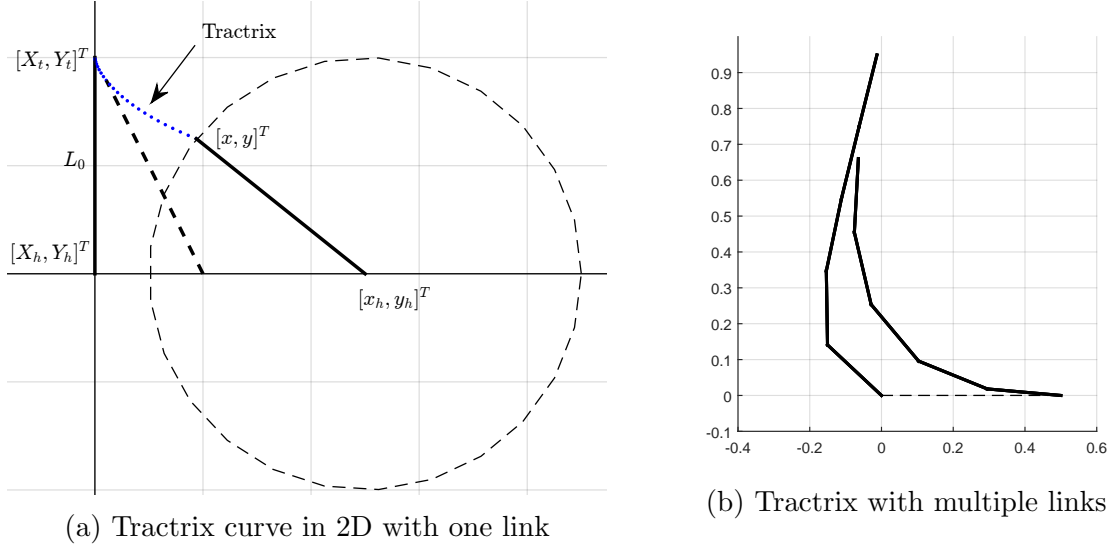(a) Tractrix curve in 2D with one link

(b) Tractrix with multiple links

Figure 1: Tractrix in a plane

position subject to the condition that the length of link is always preserved. i.e, the co-ordinates of the tail can be obtained from the following minimization problem:

$$\underset{\mathbf{x}_t}{\arg\min} \|\mathbf{x}_t - \mathbf{X}_t\| \tag{2}$$
$$\text{Subject to: } \|\mathbf{x}_h - \mathbf{x}_t\| - L_0 = 0$$

An advantage of expressing tractrix as a minimization problem is that we can add more constraints to the above expression and hence, control the position of the tail point. Though the resulting curve may not necessarily be a tractix curve, the motion of the tail will appear realistic [ref]. For the motion of a link which minimizes its tail velocity (displacement of the tail co-ordinates), obstacle avoidance is achieved by formulating the problem as:

$$\underset{\mathbf{x}_t}{\arg\min} \|\mathbf{x}_t - \mathbf{X}_t\| \tag{3}$$
$$\text{Subject to:} \|\mathbf{x}_h - \mathbf{x}_t\| - L_0 = 0$$
$$f(\mathbf{x}) \succeq 0$$

where $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ is the analytical equations of the boundaries of the surfaces which are to be avoided. For example, if the tail is to avoid a single obstacle represented by a circle with center (x_c, y_c), the expression $f(x) = (x - x_c)^2 + (y - y_c)^2 - r^2 > 0$ ensures that the point x always lies outside the circle of radius r. Complex objects can be modelled as a combination of super-ellipses as shown in [ref]. In this case, $f(x)$ will be a vector of all boundary equations

3

$\mathbf{f}(x) = [f_1(x), f_2(x), ...f_m(x)]^T$ [fig3]. It is also worth noting that the value of function will increase or decrease as the point is farther from the curve $f(x)$; the value being zero on the curve. Hence, this approach can also be imagined as a geometric potential field, with zero potential only at the surface of the obstacle.
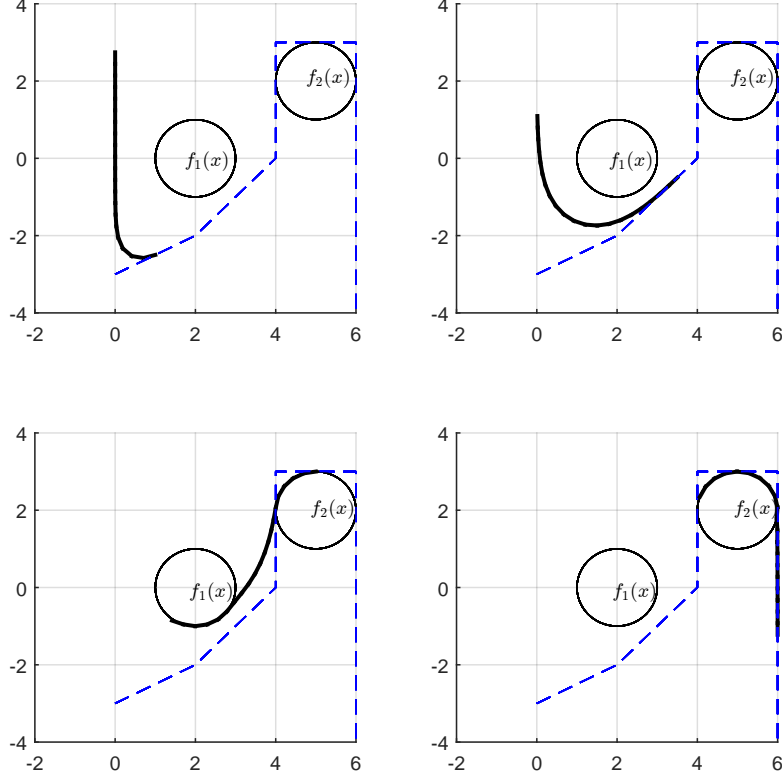

Figure 2: Obstacle avoidance in a plane

In the case of path planning of hyper-redundant manipulator, a majority of problems deal with simulating the motion of the robot while traversing through a confined space within a narrow bounded path, which we term as a "*duct*" for the remainder of the paper. The problem of planning the motion of the robot through a duct may be specified as:

$$\underset{\mathbf{x}_t}{\arg\min} \|\mathbf{x}_t - \mathbf{X}_t\| \tag{4}$$

$$\text{Subject to:} \|\mathbf{x}_h - \mathbf{x}_t\| - L_0 = 0$$

$<<<<<<< HEAD \quad f(x) < 0 ======= \quad f(x) \leq 0 >>>>>>> 1812e30da739acd007c54599205592$

While this expression is applicable for a duct represented by a single surface with the boundary $f(x)$, unlike the obstacle avoidance problem, the same will not work in the case of complex

4

surfaces represented by combination of simpler analytical shapes. This is because if a point is classified as inside one of the simpler shapes, then it should be classified as outside the other shapes forming the duct. In other words, if one constraint function $f_k(x) < 0$, then the other constraint functions $f_{i \neq k} \geq 0$. In the next sections, we present different methods to represent the ducts and how confined space motion is achieved in different cases.

2.1 Nature of the optimization problem

In the previous section we have discussed how tractrix based motion planning can be formulated as a constrained optimization problem, which entails a discussion on the suitability of the posed optimization problem and guarantee that the solution is unique and physically tractable. To address this, we will prove that the problem in equation (4) and the variations of the same used in the current work can be posed as convex problems. A function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* if and only if the following two conditions hold:

    a The domain of f, $\mathbf{dom} f$ is a convex set, and,

    b For all $x, y \in \mathbf{dom} f$ and θ with $0 \leq \theta \leq 1$

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y) \tag{5}$$

In our problem, the function is the L_2 norm of a vector in $\mathbb{R}^2$ or $\mathbb{R}^3$. Though the L_2 norm is defined for all real vectors, the constraints ensuring that the object moves within the confinement of the duct restricts the domain to a feasible closed set $\mathcal{S}$[1]. Therefore, if the conditions a and b associated with the definition of a convex function are satisfied for $\mathcal{S}$ and the L_2 norm, then the solution obtained for (put optimization equation here) is unique.

It is a well known result that any p-norm $||x||_p = (\sum\limits_{i=1}^{n} |x_i|^p)^{1/p}, \ x \in \mathbb{R}$ is a convex function for $p \geq 1$. The fact that a p-norm is convex can be shown by using the scalability and the triangle inequality associated with a normed vector-space. With θ being a scaling parameter, equation (5) simplifies exactly to the triangle equality for norms. Also, any p-norm is a valid norm on $\mathbb{R}$ for $p \geq 1$ follows from the Minkowski inequality[2]. This leaves us to show that the feasible set $\mathcal{S}$ is convex. For all our modeling examples, we choose $\mathcal{S}$ to be convex by assigning convex boundaries to it or, expressing $\mathcal{S}$ as intersections of closed convex sets. However, as is, none of the problems posed in equations (2) to (4) are convex because of the

---

[1]Assuming $\mathcal{S}$ to be a closed set allows the object to physically touch the boundaries of the duct
[2]For a measure space $\mathbb{S}$, $p \geq 1$ and f, g are members of the Lebesgue space $L_P(\mathbb{S})$, the following inequality holds: $||f + g||_p \leq ||f||_p + ||g||_p$

non-linear equality constraints associated with them, which guarantees that the length of a segment is constant. To pose it as a convex problem we approximate the constraint with 2 linear constraints as described in section(**put complexity section here**). For the cases of path planning in 2-D and 3-D, the paths chosen are either entirely convex or are discretized as such. As discussed in the following sections, we do not represent a "duct" as a union of closed convex sets as the first condition of convexity cannot be guaranteed for sets formed by union of convex sets.

3 Motion planning through planar ducts

In this section, we propose different methods to represent a duct in 2D planar surface.

3.1 Representation of duct using super-ellipses

One method to represent a duct is by overlapping a series of super-ellipses as shown in [fig5b]. This is the most simple and straightforward means of representation as shown in [ref Midhun]. In Cartesian co-ordinate system R^2, the contour of super-ellipse obeys the following equation:

$$f(\mathbf{x}) = f(x,y): \quad \left|\frac{x - x_c}{a}\right|^n + \left|\frac{y - y_c}{b}\right|^n - 1 = 0 \tag{6}$$

The condition $f(\mathbf{x}_t) < 0$ will ensure that the co-ordinates of the tail of the link (x_t, y_t) always lie inside the bounding curve of a super-ellipse. However, in case of multiple equations $(f_i(x), \; i = 1, 2, ..., m)$, only one of them will be satisfied for the point to be inside the duct. In practical implementation, we can say that this translates to saying that the least value amongst all the values of $f_i(x)$ should be less than zero. For the ith super-ellipse which is rotated by an angle ϕ_i about the $z-$ axis and whose center is translated to the co-ordinates (x_i, y_i) so as to fit a portion of a duct, the co-ordinates of boundary should be multiplied with a transformation matrix

$$T_i = \begin{bmatrix} \cos\phi_i & -\sin\phi_i & 0 & x_i \\ \sin\phi_i & \cos\phi_i & 0 & y_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

The constraint equation now becomes $g_i(\mathbf{x}_t): f_i(T_i^T \mathbf{x}_t) < 0, \; i = 1, 2, ..., m$ and, the complete
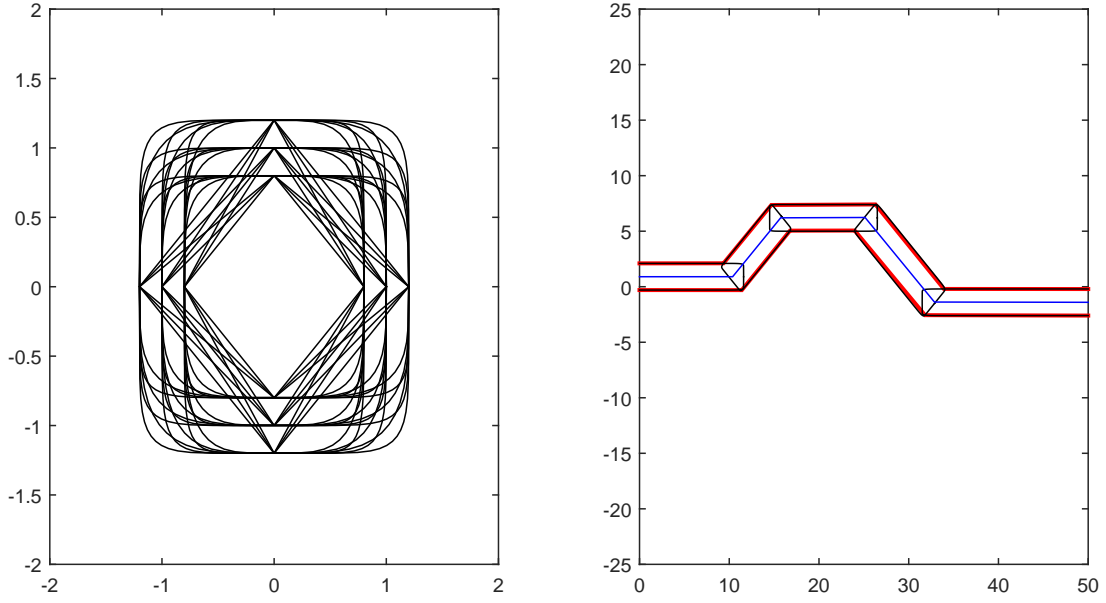
Figure 3: Super-ellipses and a duct modelled as combination of super-ellipses

formulation is given as:

$$\underset{\mathbf{x}_t}{\arg\min} \|\mathbf{x}_t - \mathbf{X}_t\| \tag{8}$$

$$\text{Subject to:} \quad \|\mathbf{x}_h - \mathbf{x}_t\| - L_0 = 0$$

$$\min(g_i(\mathbf{x}_t)) < 0, i = 1, 2, ..., m$$

An example of single link and multi-segmented chain passing through the duct is shown in [fig5]. Motion of a unit link with and without constraint is shown in [fig6]. The negative gradient (descend direction) of the inequality constraint function is also shown in the figure. The method shown here is quite fast and scalable as explained in section ??, while the majority of time taken for the scheme being in identifying the super-ellipsoids which fit the duct profile. For the example shown in this section, this identification is done by manually selecting clusters of points in the duct and fitting ellipses which will reduce the fitting error in a least squared sense.

3.2 Representation of duct as a set of connected quadrilaterals

Since the profile of a super-ellipse is always symmetric, for complex and non-symmetric ducts, representation using the previous method might require a large number of shapes. In
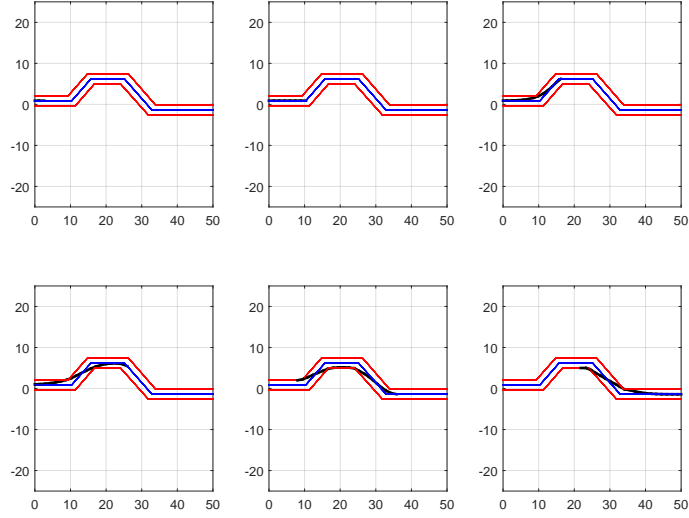
7

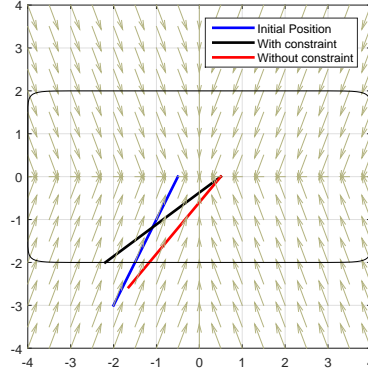Figure 4: Motion through duct modelled as combination of super-ellipses

Figure 5: Effect of gradient of inequality constraint in pulling the tail into the duct

such cases, a complex duct shape can be represented as a continuous array of discrete patches formed by piecewise continuous curves as shown in [fig6]. The individual quadrilateral patches may be denoted as $A_1, A_2, ..., A_n$, each bounded by the line segments defined by the points $(\mathbf{P_0}, \mathbf{P_1}), (\mathbf{P_1}, \mathbf{P_2}), ..., (\mathbf{P_{n-1}}, \mathbf{P_n})$ for the curve $\zeta_\mathbf{1}$ and $(\mathbf{Q_0}, \mathbf{Q_1}), (\mathbf{Q_1}, \mathbf{Q_2}), ..., (\mathbf{Q_{n-1}}, \mathbf{Q_n})$ for the curve $\zeta_\mathbf{2}$. Then the co-ordinates of a point inside the surface patch A_i is given by the parametric expression

$$\mathbf{x}_i(u,v) = [\mathbf{P}_{i-1} + (1-u)\,\mathbf{P}_i]\,(1-v) + [\mathbf{Q}_{i-1} + (1-u)\,\mathbf{Q}_i]\,(v) \tag{9}$$

in parameters u and v. If the vertices of the quadrilateral are given by $\mathbf{P}_i = [{}^xP_i, {}^yP_i]^T$ and $\mathbf{Q}_i = [{}^xQ_i, {}^yQ_i]^T$, then the analytical expressions for the terms u and v, given the value of

8

$\mathbf{x}_i$, can be obtained by solving 9 (see Appendix). The values of u, v can be used to classify the point with respect to the surface patch A_i[3].
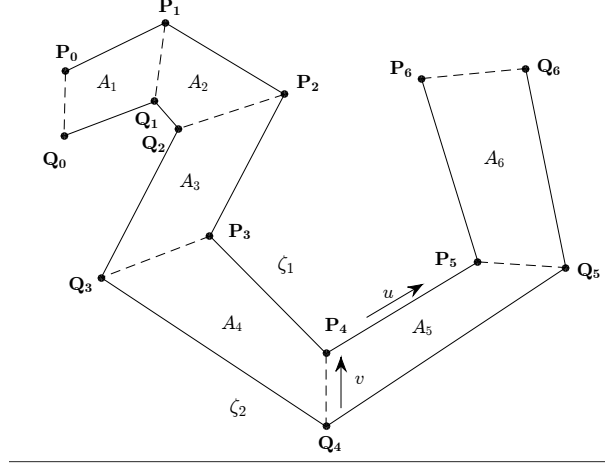

Figure 6: Duct represented by stitched quadrilaterals

In case of a single quadrilateral patch, we can now write the optimization problem as:

$$\min_{\mathbf{x}_t} \|\mathbf{x}_t - \mathbf{X}_t\| \tag{10}$$

$$\text{Subject to:} \quad \|\mathbf{x}_h - \mathbf{x}_t\| - L_0 = 0 \tag{11}$$

$$0 \leq u \leq 1 \tag{12}$$

$$0 \leq v \leq 1 \tag{13}$$

for real values of u and v. In case of multiple patches, classifying one point with respect to all the patches return the values $(u_1, v_1), (u_2, v_2), ..., (u_m, v_m)$ etc. for the m number of patches $A_1, A_2, ..., A_m$ and consequently, m set of conditions. But out of the m condition sets, only one set should be satisfied since the point will belong to only one patch at a given instance of motion through the duct. Including a switching statement in the optimization code will be inefficient especially when the point to be classified is close to the common boundary between two patches.

In order to provide a gradient to the constraint which will direct the point into the duct, an inequality constraint is to be included as in the case of super-ellipses described in the

---

[3]It may be noted that there will be two sets of solution and they are not always real and unique. For example, the point $\mathbf{P} = (10, -5)$ when classified with respect to the area A given by the points $\mathbf{P}_1 = (0, 15), \mathbf{P}_2 = (10, 10), \mathbf{Q}_1 = (0, 0)$ and $\mathbf{Q}_2 = (4, 1)$, returns the values $u = (1.0 + 0.6\ i,\ 1.0 - 0.6\ i)$ and $v = (2.0 + 1.9\ i,\ 2.0 - 1.9\ i)$. However, it is a trivial task to filter out the imaginary set of solutions, should the algorithm encounters the same.

previous subsection. If $\hat{u}$ and $\hat{v}$ represent the parameters obtained for a point $\mathbf{x}_t$ classified with respect to the quadrilateral A_i, $x_{\zeta_1}(t) = \mathbf{P}_{i-1} + (1 - \hat{u})\mathbf{P}_i$ and $x_{\zeta_2}(t) = \mathbf{Q}_{i-1} + (1 - \hat{u})\mathbf{Q}_i$ will give the two points on the duct boundary curves corresponding to the parameter $\hat{u}$. Then using triangular inequality theorem ??, we can see that the value

$$h = \|\mathbf{x}_{\zeta_1} - \mathbf{x}_t\|^2 + \|\mathbf{x}_{\zeta_2} - \mathbf{x}_t\|^2 - \|\mathbf{x}_{\zeta_1} - \mathbf{x}_{\zeta_2}\|^2 \tag{14}$$

will always assign a negative real value for h when point is inside the duct and a positive real value when the point is outside the duct. The value will be zero only at the boundaries. Hence, for an array of quadrilaterals, it is only necessary that the minimum value of the vector $\mathbf{h} = [h_1, h_2, ..., h_m]$ should be negative for classifying the point with respect to the duct, as in the case of previous section. But as we can see, the inequality only takes into account the parameter variation across the boundaries (along the parameter v) and not in the direction of u which could result in erroneous classification. For example, with reference to the figure [fig 9b], the value of h for the point $\mathbf{P}$ classified with respect to the two quadrilaterals are $h_1 = ??$ and $h_2 = ??$. Hence, a minimum of both the values will wrongly classify the point as inside. To account for the same, we make use of the function χ which is necessarily a linear combination of two Heaviside step functions $H(0) - H(1)$, defined as:

$$\chi(t) = \begin{cases} 0, & t < 0 \\ 1, & 0 \leq t \leq 1 \\ 0, & t > 1 \end{cases} \tag{15}$$

The function χ applied on the quantity $\hat{u}_i$ (which is the value of parameter u classified with respect to quadrilateral A_i), will return 0 only if the point satisfies the constraint $0 \leq \hat{u}_i \leq 1$. Now, multiplying this quantity $\chi(\hat{u})$ with h_i will return a non-zero negative value only if the point is inside the duct. Including the inequality constraint, the complete optimization problem is given as:

$$\min_{\mathbf{x}_t} \|\mathbf{x}_t - \mathbf{X}_t\| \tag{16}$$

$$\text{sub:} \quad \|\mathbf{x}_h - \mathbf{x}_t\| - L_0 = 0$$

$$[\chi(\hat{\mathbf{u}})]^T \mathbf{h} < 0 \tag{17}$$

where $\hat{\mathbf{u}} = [\hat{u}_1, \hat{u}_2, ..., \hat{u}_m]^T$

Motion of a unit link passing through the duct is shown in [fig11] and the effect of the added inequality constraint to pull the tail end of the link which is initially positioned outside the duct, is shown in [fig12]
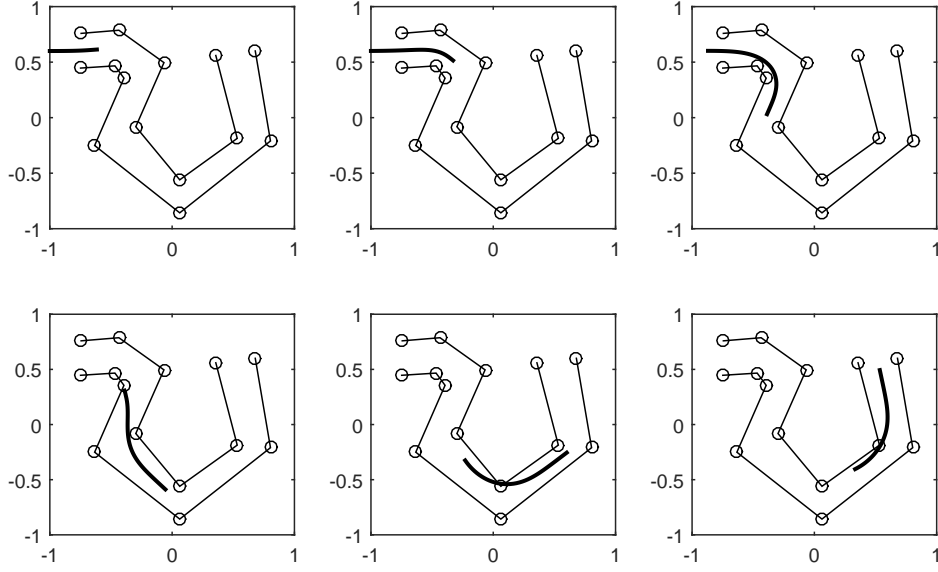
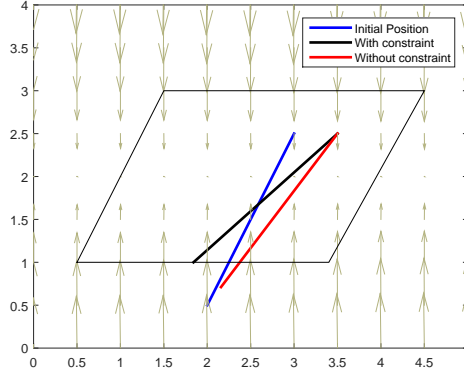Figure 7: Example of constrained motion with stitched quadrilaterals

Figure 8: Effect of gradient of inequality constraint in pulling the tail into the quadrilateral duct

3.3 Representation of duct using two non-intersecting continuous curves

If the non-intersecting border curves of the duct can be analytically expressed, then the equation of the surface patch will simply be,
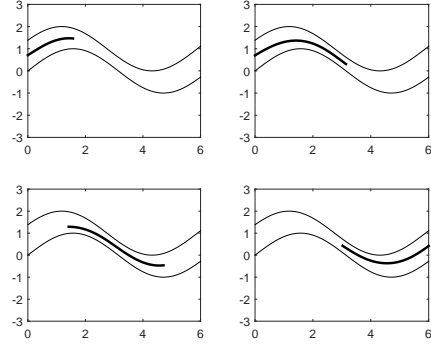
$$\mathbf{x}_i(u, v) = \zeta_1(u)(1 - v) + \zeta_2(u)(v) \tag{18}$$

For example, [fig11] shows a 2D duct defined by two curves $\zeta_1(u) = [u, \ \sin(u)]^T$ and $\zeta_2(u) = [u, \ \sin\left(u + \frac{\pi}{8}\right) + 1]^T$ and a path chosen midway between the two curves. The equation of

11

the surface generated by this curves will be
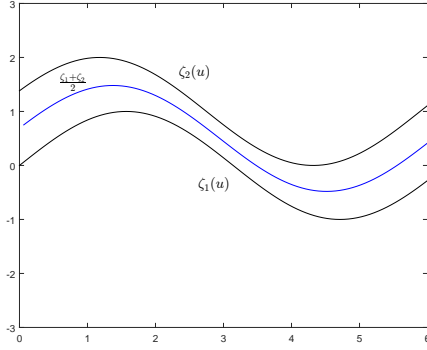
(a) Example of analytical duct

(b) Motion through analytical duct

Figure 9: Tractrix based algorithm on analytical duct

$$\begin{bmatrix} x(u,v) \\ y(u,v) \end{bmatrix} = \begin{bmatrix} u \\ \sin\left(u\right) + \left[\sin\left(u + \frac{\pi}{8}\right) - \sin\left(u\right) + 1\right]v \end{bmatrix} \tag{19}$$

which has the analytical solution for u and v:

$$u = x$$

$$v = \frac{y - \sin\left(x\right)}{\sin\left(x + \frac{\pi}{8}\right) - \sin\left(x\right) + 1}$$

In this case, we will solve the equations:

$$\min_{\mathbf{x}_t} \|\mathbf{x}_t - \mathbf{X}_t\| \tag{20}$$

$$\text{sub:} \quad \|\mathbf{x}_h - \mathbf{x}_t\| - L_0 = 0$$

$$0 \le v = \frac{y_t - \sin\left(x_t\right)}{\sin\left(x_t + \frac{\pi}{8}\right) - \sin\left(x_t\right) + 1} \le 1 \tag{21}$$

An example movement of hyper-redundant manipulator through the duct is shown in [fig12]. However, analytical solution is not always viable for complex equations and numerical procedure must be employed to find the values of u and v corresponding to the given tail point to be classified. The equation to be used is 10.

4   Motion planning through 3D ducts

In this section, we will explain a few methods to represent ducts in 3D and how motion planning is achieved in the same.

4.1 Representation of duct using combination of super-ellipsoids

In Cartesian co-ordinate system in R^3, the surface of a super-ellipsoid follows the equation:

$$f(x,y,z): \left[\left\{\left(\frac{x}{a}\right)^{\frac{2}{e}} + \left(\frac{y}{b}\right)^{\frac{2}{e}}\right\}^{\frac{e}{n}} + \left(\frac{z}{c}\right)^{\frac{2}{n}}\right]^{\frac{n}{2}} - 1 = 0 \tag{22}$$

By changing the parameters a, b, c and n, we get different closed surfaces as shown in [fig13]. By combining different super-ellipse shapes, we can generate a 3D duct as shown in [fig14]

[fig13] : What are super-ellipsoids. [fig14] : Duct represented by super-ellipsoids.

The procedure to calculate the inside-outside condition is same as that of the method described in section [ref sec]. The final equations will be same as [**??**]. An example problem with duct approximated using super-ellipsoids is shown in [fig15]. As mentioned in the case for super-ellipses, solution to the path planning problem with ducts represented by super-ellipsoids is fast as explained in section ??. Identifying the shapes which fit the duct, is also same as the method mentioned in section ??.

[fig15] : Motion through super-ellipsoid ducts

4.2 Representation of duct as a set of connected cylinders

Another more involved method in representing duct is by series of connected cylinders. By linearly interpolating two circles in 3D, we get the parametric equation of surface of the cylinder as (refer Appendix for the detailed expressions):

$$x = C_1(u, t, \theta), \quad y = C_2(u, t, \theta), \quad z = C_3(u, t, \theta) \tag{23}$$

where the parameters u, t and θ varies along the radial, axial and circumferential direction of the cylinder respectively (refer [fig16]). Similar to the representation in [sec], $0 \le u < 1$ and $0 < t < 1$ classifies the point as inside the cylinder. The final minimization problem is same as that of [16]. One major advantage of using this formulation is that by changing the radius of cylinder from r to $r - \delta$, we can easily implement a clearance of δ units from the walls of the cylinder. Motion of link chain through the duct connected using cylinders is shown in [fig17]

[fig16] : Connected cylinder forming duct [fig17] : Motion of HR manipulator through connected-cylinders duct

4.3 Representation of duct as point clouds

Most practically convenient way of representing the duct would be as the a collection of point clouds ($\mathbf{R}_i, i = 1, 2, ..., N$ where N is the number of points in the cloud) such as obtained from Stereolithographic formatted file (STL). It is possible to pose the optimization problem in the form:

$$\min_{\mathbf{x}_t} \|\mathbf{x}_t - \mathbf{X}_t\| \tag{24}$$

$$\text{sub:} \quad \|\mathbf{x}_h - \mathbf{x}_t\| - L_0 = 0$$

$$[\mathbf{A}]\mathbf{x}_t + \mathbf{B} \leq 0 \tag{25}$$

where $\mathbf{A}$ is a $m \times 3$ matrix and $\mathbf{B}$ is a $m \times 1$ vector. The left hand side of m inequalities represent the equations of m number of planes spanned by three adjacent points in the cloud. The ith equation, $A_i^1 x_t + A_i^2 y_t + A_i^3 z_t + B_i$ takes a value less than zero when the point $\mathbf{x}_t$ is in the half space which contains the origin and is greater than zero otherwise. The value also provides the attractive gradient which will ensure that the point stays inside the duct. However, in actual implementation, this procedure will be tedious and for practical convenience, it is possible to classify the point $\mathbf{x}_t$ as inside or outside the hull using the algorithm proposed by ??. The attracting gradient which ensures the point to be inside the duct–as the case with the previous methods– can be provided using the artificial potential field generated from the centroid of the point cloud in conjunction with the output of the in-out function. The inequality constraint then becomes

$$w(\mathbf{R}) \frac{1}{\|(\overline{\mathbf{R}}) - \mathbf{x}_t\|} \leq 0 \tag{26}$$

where $w(\overline{\mathbf{R}})$ represents the output from in-out function which is either 1 for the point being outside and 0 for the point being inside the cloud or on the bounding surface[4].

Another interesting prospect of this method is its application in obstacle avoidance where the left hand side of the equation Equation (26) will be greater than zero. The obstacle avoidance methods which follow the equation given by Equation (3) is often limited to shapes which can be modelled analytically. Though the algorithm shown here is limited to convex shapes, they can be applied to more complex and non-symmetric shapes.

4.4 Analytical ducts

As mentioned in the section??, it is possible to express same

---

[4]Unlike the previous classification problems, the bounding surface will also be considered as inside the surface in this case.

5 Examples and discussion

In this section, we present two practical examples of the above mentioned methods and some discussion on the implementation of the same.

5.1 Examples of motion planning through ducts

5.1.1 Motion planning through a GI tract

There has been a growing interest in simulating motion of endoscope through GI tract for developing simulators for endoscopy and for implementing path planning for endoscopic and laparoscopic surgical robots. In this section, we simulate the natural motion of an endoscope through GI tract. For simulation, we use the stereolithographic data of GI tract obtained from ??. We have demonstrated both the methods presented in [sec] and [sec] for approximating the GI tract. In the first method, a collection of points are manually selected from the STL file where super-elliposids are fit based on least square error minimization techniques. Representation of GI tract as series of super-ellipsoids is shown in [fig19]

[fig19] : Representation of GI tract as super-ellipsoids.

For representing GI tract as cylinders, we first found out the medial axis of the duct using ??. Then at equal intervals of distance along the medial axis, planes are drawn normal to the same. The collection of points which are in the close proximity of the plane are selected and a circle is fitted on the points using least square error minimization. The parameters so obtained are used for the cylinder equations in [23]. Representation of GI tract as a series of connected cylinders is shown in [fig20]

[fig20] : Representation of GI tract as cylinders.

The realistic motion simulation of endoscope through GI tract is shown in [fig21]

[fig21] : Motion through GI tract

5.1.2 Motion planning for search and rescue operations

Exploring obstacle-cluttered environments such as in an earthquake hit zone is one of the major applications that necessitates the use of hyper-redundant manipulators. In this example, we show how the proposed algorithms can be effectively used in such applications. With reference to [fig ??], the objective of the hyper-redundant manipulator, consisting of 20 links is to 1) enter the scene through the hole in the outer wall, 2) Pass through the vent to get inside the room, 3) Explore the objects 1,2 and 3 located inside the blue hull and 4) Exit without colliding with the complex shapes given by objects 4 and 5. Here, the

exploration problem is effectively tackled using the following constraints added to the optimization problem (refer Fig ??): 1) An ellipsoid of the size of the hole located in the outer wall 2) collection of cylindrical ducts to represent the vent, 3) STL models for the objects 1,2 and 3 and 4) Ellipsoid at the last bend in the path to avoid collision with the complex shapes of objects 4 and 5. It is easy to verify that the existing geometry based motion planning of robot will be quite complicated for this scene while using the methods proposed in this paper, the problem can be addressed with only 6 constraints. The six constraints in this case are actively selected so that they become activate only when the position of the head is in the vicinity of the objects/ shapes. Figure [??] shown the simulated results for the motion of manipulator along a chosen path.

5.2 Computational complexity and limitations of tractrix based scheme

... for now the procedure takes ?? takes to compute. We plan to reduce this in the future. Another limitation of tractrix based scheme proposed is the locking of a link which traverses sharp bends for non-differentiable ducts given by representations proposed in ?? and ??, as shown in [fig ??]. Here, the expected subsequent position of the link is the one which bends around the corner following the head, while the tractrix based approach will result in the link being pulled downwards (like bicycle chain on sprocket) to the position 2 shown in the figure, which is the theoretical minimum for the optimization problem satisfying the constraints. Hence, the bending angle at the corners in the most general case may be limited to 90°. In order to avoid this issue, the sharp corners may be blunted off by adding extra points in the corners as shown in [fig ??]. For the curve ζ_1 which includes a corner represented by the points $\mathbf{P}_1, \mathbf{P}_2$ and $\mathbf{P}_3$, and a given link length L_0 such that $\|\mathbf{P}_1 - \mathbf{P}_2\| \geq L_0$ and $\|\mathbf{P}_3 - \mathbf{P}_2\| \geq L_0$, the curvature may be increased by adding another line segment so that the final points be $\mathbf{P}_1, \mathbf{P}_2^a, \mathbf{P}_2^b, \mathbf{P}_3$ where the point $\mathbf{P}_2^a = \mathbf{P}_2 + \frac{L_0}{\|\mathbf{P}_2 - \mathbf{P}_1\|} \| (\mathbf{P}_1 - \mathbf{P}_2\|)$ and $\mathbf{P}_2^b$ may be obtained form solving the equations

$$\|\mathbf{P}_2 - \mathbf{P}_2^b\| = L_0, \quad \|\mathbf{P}_2^a - \mathbf{P}_2^b\| = L_0 \tag{27}$$

Needless to say, this rounding off is subject to practical feasibility.

5.3 Tractrix emulates natural motion

Motion simulated using the approach discussed in this paper imparts realism due to the minimal movement of tail with respect to the head. In nature this effect is created due to

the friction or other resistive forces acting in the lateral direction of the link. In the absence of friction, displacement given in the head will result in a simple translation of link. In other words, $\mathbf{x}_t$ in this case, will be same as $\mathbf{x}_h$. In figure [fig22], we can see the effect of friction on a connected link chain which is pulled in one direction. The discrepancy between natural movement and the simulation based on the tractrix approach is because tractrix represents the ideal scenario with motion of tail limited only in the direction of motion of the link due to very high friction while in natural motion, there is always some degree of translation possible. This slippage could be included in the formulation, however, by adding a slip vector to the tail displacement. Since the slippage amounts to the translation of link in the direction of link, slip vector will be in the direction of head displacement and can be written as as $\mathbf{s} = \nu\mathbf{X}_h$, where $\nu < 1$ is a constant. $\nu = 0$ represents the ideal tractrix case with friction and $\nu = 1$ represents pure translation of the link. Including this vector, the minimization function may be written as:

$$\min_{\mathbf{x}_t}\|\mathbf{x}_t - (\mathbf{X}_t + \nu\mathbf{X}_h)\,\| \tag{28}$$

[fig22] : Experiments using bicycle chain

Using a constant value of $\nu =$??, we can see from [fig23] that the formulation conforms with the actual final pose with good accuracy.

[fig23] : Tractrix with ν.

6   Conclusions

??

6.1   Contributions and scope of future work

uses LAPACK and heavy solvers.. not really required. instead a stripped version of the interior point will do quite well for fast near real-time implementation.

7 Appendix

7.1 Analytical expressions for u, v for quadrilateral patch

$$v = \frac{k_1 - k_2 \pm \sqrt{(k_2 - k_1)^2 - 4k_3 k_4}}{2k_3}, \quad u = \frac{(x - {}^x P_{i-1} + v)}{(a_2 + v a_3)} \tag{29}$$

where, $k_1 = (b_1 b_2 + b_0 b_3)$, $k_2 = (a_1 a_2 + a_0 a_3)$, $k_3 = (a_1 a_3 - b_1 b_3)$, $k_4 = (a_0 a_2 - b_0 b_2)$

$a_0 = y - {}^y P_{i-1}$, $a_1 = {}^y P_{i-1} - {}^y P_i$, $a_2 = {}^x Q_{i-1} - {}^x P_{i-1}$, $a_3 = ({}^x Q_i - {}^x P_i) - ({}^x Q_{i-1} - {}^x P_{i-1})$

$b_0 = x - {}^x P_{i-1}$, $b_1 = {}^x P_{i-1} - {}^x P_i$, $b_2 = {}^y Q_{i-1} - {}^y P_{i-1}$, $b_3 = ({}^y Q_i - {}^y P_i) - ({}^y Q_{i-1} - {}^y P_{i-1})$

If ${}^x P_{i-1} = {}^x P_i$ and ${}^x Q_{i-1} = {}^x Q_i$,

$$u = \frac{b_0}{a_2}, \quad v = \frac{a_0 a_2 - (a_3 + a_2)}{a_1 (b_0 - a_2) + b_0 (b_3 - b_2)} \tag{30}$$

7.2 Parametric equation of solid cylinder

$$x = C_1(u, t, \theta) = (r_1 m_1 u \cos \theta + m_2)(1 - t) + (r_2 n_1 u \cos \theta + n_2) t \tag{31}$$

$$y = C_2(u, t, \theta) = (r_1 m_3 u \cos \theta + r_1 m_4 u \sin \theta + m_5)(1 - t) + (r_2 n_3 u \cos \theta + r_2 n_4 u \sin \theta + n_5) t \tag{32}$$

$$z = C_3(u, t, \theta) = (r_1 m_6 u \cos \theta + r_1 m_7 u \sin \theta + m_8)(1 - t) + (r_2 n_6 u \cos \theta + r_2 n_7 u \sin \theta + n_8) t \tag{33}$$

where

$$m_1 = \cos {}^1\phi_2, \ m_2 = {}^1 x_c, \ m_3 = \sin {}^1\phi_1 \sin {}^1\phi_2, \ m_4 = \cos {}^1\phi_1,$$

$$m_5 = {}^1 y_c, \ m_6 = -\cos {}^1\phi_1 \sin {}^1\phi_2, \ m_7 = \sin {}^1\phi_1, \ m_8 = {}^1 z_c$$

and

$$n_1 = \cos {}^2\phi_2, \ n_2 = {}^2 x_c, \ n_3 = \sin {}^2\phi_1 \sin {}^2\phi_2, \ n_4 = \cos {}^2\phi_1,$$

$$n_5 = {}^2 y_c, \ n_6 = -\cos {}^2\phi_1 \sin {}^2\phi_2, \ n_7 = \sin {}^2\phi_1, \ n_8 = {}^2 z_c$$

The quantity ${}^1(\cdot)$ and ${}^2(\cdot)$ represent the corresponding parameters of the circles at the ends of the cylinder. ϕ_1 and ϕ_2 are the angles about the Y and X-axes which the plane of the circle is rotated, (x_c, y_c, z_c) is the co-ordinate of the center of the circle.

7.3 An algorithm for classifying a point with respect to a polyhedron

Classification a point with respect to a triangulated domain is a very well known problem and many methods exist, which offer various advantages in terms of computational complexities. In the algorithm algorithm 1, we describe a method, to classify a given set of points O with respect to $\mathcal{S}$ as inside (O_{in}) and outside (O_{out}). Our algorithm and it's Matlab implementation[5] is not very efficient at higher dimensions but seems to perform reasonably well at 3D, by taking less than 8 seconds to classify about 10,000 points with respect to a meshed domain.

Purpose : To classify a set of points O with respect to $\partial\mathcal{S}$
Input: The set O, $O \in \Re^3$ and $O \equiv O_{in} \cup O_{out}$
Output: O_{in}, O_{out}
  1: Obtain the α shape $\mathcal{S}$ of the set of points $\mathbb{S}$
  2: Obtain C, the center of $\mathcal{S}$ by taking the mean of the vertices of the N facets constituting $\partial\mathcal{S}$
  3: **for** $i = 1, 2, ..., N$ **do**
  4:     Calculate the normal to the ith facet, $A_n^i = (f_i^1 - f_i^2) \times (f_i^2 - f_i^3)$
  5:     Choose a point A^i on the ith facet and move A_n^i to A^i
  6:     Ensure that A_n^i is directed inside, towards C
  7:     $H(i) = \langle \vec{O_i - A_i}, A_n^i \rangle$
  8: **end for**
  9: **if** $H(i) \geq 0\ \forall i$ **then**
 10:     Classify $O_1 \in O_{in}$ as inside, otherwise classify $O_1 \in O_{out}$
 11: **end if**
Algorithm 1: Algorithm for classifying points as inside (O_{in}) or outside (O_{out}) of a triangulated domain

References

We have to see IEEE reference format.

---

[5]We have used a Matlab function inhull.m, a fully vectorized implementation of the algorithm, made available by John D'Errico for free usage.