Univ. u Novom Sadu
Zb. Rad. Prirod.–Mat. Fak.
Ser. Mat. 23, 2 (1993), 321 - 330

Review of Research
Faculty of Science
Mathematics Series

AN ALGORITHM FOR DETERMINING THE ADMISSIBLE REGION FOR A TWO - LINK MECHANISM IN THE PRESENCE OF POLYGONAL OBSTACLES

Nataša Divljak
Institute of Mathematics, University of Novi Sad
Trg Dositeja Obradovića 4, 21000 Novi Sad, Yugoslavia

Abstract

An algorithm is formed for determining the region of admissible positions for a two-link mechanism in the presence of poligonal obstancles. The algorithm is implemented in Pascal and an illustrative numerical example is given.

AMS Mathematics Subject Classification (1991): 70B15, 53A17.
Key words and phrases: Collision-free region, obstacle avoidance, robotics.

1. Introduction

The problem of determining a set of admissible positions for the tip of a two-link mechanism in the presence of obstacles has been dealt in [1] and [2]. The problem of planning an admissible trajectory for the tip of the two-link mechanism is reduced to solving the problem of determining the shortest distance between two points in a plane containing obstacles. To solve this problem, a suboptimal algorithm based on a searching technique with the criterion of the shortest distance to the target point has been described [3]. On the other hand, the problem of determining the shortest distance between two points in the presence of an obstacle in the form of a circumference has benn solved in [4].

In [2], a theorem has been proved on the basis of which the following statement holds: If a position of the two-link mechanism is admissible when the mechanism tip describes a closed Jordan curve, then the position of the mechanism is also admissible when the mechanism tip is in the interior region of that curve. Starting from this statement we shall present here an algorithm for determining the region of the admissible position for the tip of a mechanism in the presence of polygonal obstacles. This region is determined by the polygonal lines and the circumference arcs with radii equal to the sum and/or the absolute values of the difference of the mechanism link lengths.

2. Two-link mechanism

Let O be a fixed point in the plane. Consider the linkage OAB, such that $OA = \ell_1, AB = \ell_2$. By the rotation about the center O, the end point A of the segment OA can reach any point of the circle (O, ℓ_1). Similarly, by rotation about A, the end point B of the segment AB can reach any point of the circle (A, ℓ_2).

The coordinates of point B (Fig.1) are:

(1) $$x = \ell_1 \cos q_1 + \ell_2 \cos(q_1 + q_2)$$

(2) $$y = \ell_1 \sin q_1 + \ell_2 \sin(q_1 + q_2)$$

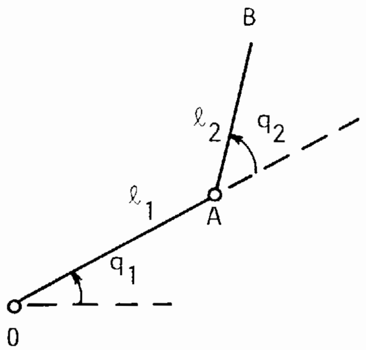

Fig.1

Since small changes in Δx and Δy imply small changes in Δq_1 and Δq_2, then on the basis of (1) and (2), the following holds:

$$(3) \qquad \Delta q_1 = (\cos(q_1 + q_2)\Delta x + \sin(q_1 + q_2)\Delta y)/(\ell_1 \sin q_2)$$

$$(4) \qquad \Delta q_2 = -(\ell_1 \cos q_1 + \ell_2 \cos(q_1 + q_2))\Delta x/(\ell_1\ell_2 \sin q_2) - \\ -(\ell_1 \sin q_1 + \ell_2 \sin(q_1 + q_2))\Delta y/(\ell_1\ell_2 \sin q_2)$$

3. Algorithm

On the basis of values ℓ_1 i ℓ_2 we can determine a set of achievable points in the plane, i. e. the ring

$$R = \{X : |\ell_1 - \ell_2| \leq OX \leq \ell_1 + \ell_2\}$$

Let us cover the ring R by a grid of squares of a given dimension which are represented by a matrix. The matrix element $a[i, j]$ corresponds to a square with a lower left vertex (i, j).

First, the achievable squares whose intersection with the ring is nonempty, are determined. For each square, it is sufficient to examine whether the closest or the farthest vertex w.r.t. ring centre is within the ring (Fig. 2).

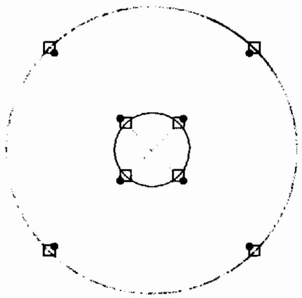

Fig.2

Then, it is necessary to test if the achievable squares are admissible. On the basis of the theorem from [2], given in the Introduction, it follows that to test the admissibility of its edges (in the case of squares that do not belong completely to the ring, only the edges in the ring and a part of the circumference) are relevant. The part of the ring which is outside is neglected, so that the admissibility of a square is determined only by its part which is in the ring.

The algorithm for testing the admissibility of a square is of the form:
n - number of divisions of a square
p - obstacles, represented by an array of polygons

```
function square (A, B, q1, q2): 0..1;    (1 - admissible)
(tests if the square B - the lower left vertex is admissible)
 function admissible (Δx, Δy, q1, q2): Boolean;
 (q1, q2 - the angles of the initial position)
    begin
       admissible: = false;
       for i: =1 to n do
         begin
           Bx := Bx + Δx;
           By := By + Δy;
           if B is achievable then
             begin
               if outside
                 begin
                   C1 := B;
                   outside: = false;
                   angles (B, q1, q2)  (the angles are determined via
                                        intersection of the circumference)
                 end
               else
                 begin
                   calculations for Δq1 and Δq2 ((3) and (4));
                   q1 := q1 + Δq1;
                   q2 := q2 + Δq2;
                   Ax := ℓ1 cos q1;
                   Ay := ℓ1 sin q1
                 end;
               if (OA ∩ p ≠ ∅) or (AB ∩ p ≠ ∅) then exit
           end
           else
```

```
            if not outside
              begin
                C_2 := B;
                outside:=true
              end
        end;
     admissible:=true
  end;


begin
  square:=0;
  outside:=false;
  if B is achievable then
    if (OA ∩ p ≠ ∅) or (AB ∩ p ≠ ∅) then exit
    else outside:=true;
  if admissible (1/n,0,q_1,q_2) then
    if admissible (0,1/n,q_1,q_2) then
      if admissible (-1/n,0,q_1,q_2) then
        if admissible (0,-1/n,q_1,q_2) then
          if arc (C_1,C_2) then     (tests the admissibility
                                     the circumference arc)
            square:=1
end;
```

As a result of testing the admissibility of a square, a matrix is obtained whose elements are 0 and 1 (0-not admissible, 1-admissible). On the basis of the given matrix, coordinates of the vertices of the polygons representing the non-admissible region are determined. The vertices are determined in the following order: the vertex with the lowest x coordinate among those with the lowest y coordinate is determined first, and then the subsequent vertices going along the polygon edges in the counterclockwise direction. When a particular polygon is tested, it is marked in order to exclude it from further searching, which is achieved by setting the matrix elements to 2. The algorithm is of the form:

r - the radius of the lerger circumference

```
outside:=true; (denotes that the given square is outside
                the tested polygon)
for j: = -r to r-1 do
  for i:= -r to r-1 do
    if a[i, j] = 2 then outside:= not outside
    else
      if (a [i,j] = 0) and outside then polygon (i,j);
```

direction - Boolean variable whose value is true when the subsecvent vertex has a highher y coordinate

```
procedure polygon (i, j);
  begin
    vertex(i,j)       (first vertex)
    a[i, j]:=2;
    i:=i + 1;
    while a[i, j]=0 do i:=i + 1;     (second vertex)
    vertex (i, j);
    a[i, j]:=2;
    direction:=true;
    repeat
      if direction then up
      else down
    until current vertex = first vertex
  end;
```

```
procedure up;
  begin
    j:=j + 1;
    if a[i, j]=0 then      (next vertex - higher x coordinate )
      begin
        vertex (i, j);
        while (a[i, j]=0) and (a[i,j-1] ≠ 0) do i:=i+1;
        vertex (i, j);
        direction:=a[i-1,j] ≠ 0;
        if direction then a[i,j]:=2
      end
    else
```

```
      if a[i,j]=1 then       (next vertex - lower x coordinate)
        begin
          vertex (i,j);
          while (a[i-1,j]=1) and (a[i-1,j-1] ≠ 1) do i:=i-1;
          vertex (i,j);
          direction:=a[i-1, j-1] ≠1;
          if direction then a[i, j]:=2
        end
        else a[i,j]:=2       (nex vertex - same x coordinate)
  end;


procedure down;
  begin
    j:=j-1;
    if a[i-1,j+1] =0 then   (next vertex - lower x coordinate)
      begin
        vertex (i,j+1);
        while (a[i-1,j] ≠1) and (a[i-1,j+1]=1) do i:=i-1;
        vertex (i,j);
        direction:=a[i-1,j+1]=0;
        if not direction then a[i,j]:=2
      end
    else
      if a[i,j]=1 then       (next vertex - higher x coordinate)
        begin
          vertex (i,j+1);
          while (a[i,j+1]=1) and (a[i,j+1]≠ 1) do i:=i+1;
          vertex (i,j+1);
          direction:=a[i,j+1]=1;
          if not direction then a[i,j]:=2
        end
      else a[i,j]:=2       (next vertex - same x coordinate)
  end;
```

4. Numerical example

The algorithmic structure described above has been implemented in Pascal.

The link lenght: $\ell_1 := 6; \ell_2 := 4$

The square edge is 1, and the number of divisions of the square n = 10.

The vertices of polygonal obstacles are:

First obstacle: (1,3), (6,4), (8,2.5), (8.5,4), (5,6), (4,8)

Second obstacle: (-2,-9), (1,-7), (-1,-7), (-1,-5), (-5,-5.5), (-5.5,-7.5)

In Fig.3 and Fig.4 are shown the obstacles (shaded area) and the region of non-admissible positions for the mechanism tip.

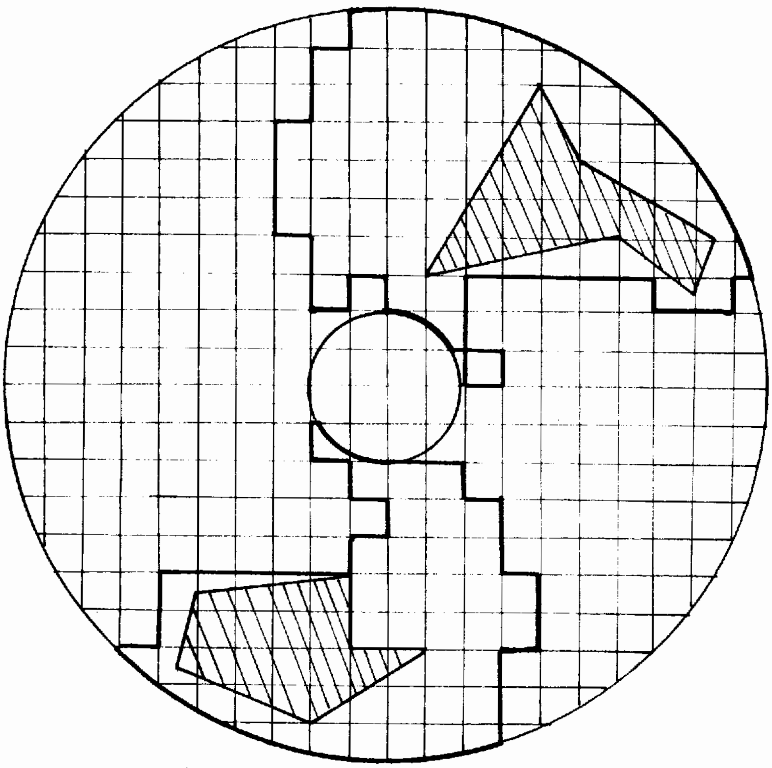

Fig.3 ($q_2 > 0$)

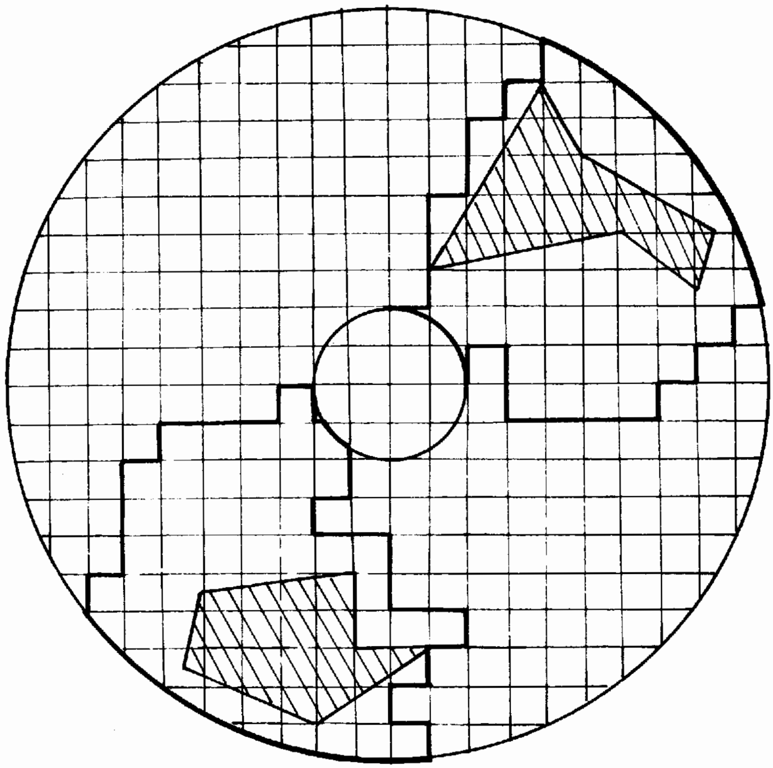

Fig.4 ($q_2 < 0$)

5. Conclusion

By determining the region of admissible positions of the two-link mechanism, the task of planning the shortest distance between the two points in the plane with the obstacles. The obstacles are defined by polygonal lines and the circumference segments.

References

[1] Surla, D., Konjović, Z., Determination of the collosion-free region for two-link mechanism in the presence of obstacles, Automatika 29(1988), 1-2, A.363-A.369.

[2] Tošić, R., Surla, D., The set of admissible positions for a two-linkage in the presence of obstacles, Z. angew Math. Mech. (to appear).

[3] Surla, D., Racković, M., An algorithm for determining near-optimal distance between two points in a plane containing obstacles, Automatica 31 (1990), 1-, 55-58.

[4] Surla, D., Racković, M., An algorithm for determining the shortest path for two-link mechanism tip between two points in the presence of polygonal obstacles, XXXIV Yugoslav Conference of ETAN, Zagreb, 1990. (in Serbo-Croat. to appear).

REZIME

ALGORITAM ZA ODREDJIVANJE DOPUSTIVE OBLASTI DVO-SEGMENTNOG MEHANIZMA U PRISUSTVU POLIGONALNIH PREPREKA

Formiran je algoritam za odredjivanje oblasti dopustivog položaja vrha dvo-segmentnog mehanizma u prisustvu poligonalnih prepreka. Algoritam je implementiran u Pascalu i prikazani su ilustrativni numerički primeri.

Received by the editors July 5, 1990