ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2008, Vol. 37, No. 2

PATH PLANNING ALGORITHM FOR SNAKE-LIKE ROBOTS

Erdinc Sahin Conkur

Mechanical Engineering Department, Pamukkale University
Denizli, 20017, Turkey

Riza Gurbuz

Ankara University Cankiri College, Cankiri Meslek Yuksekokulu
Cankiri, 18200, Turkey

Abstract. Snake-like robots have the ability of automatically performing various tasks that require man-equivalent capabilities by reaching areas difficult or impossible to reach for human beings. However, problems in path planning and design of such robots prevent them from being fully functional. In this paper, a path planning algorithm for snake-like robots is presented. Snake-like robots are modeled with discreet serial links employing many degrees of freedom. They are able to follow smoothly curved paths consisting of many points by determining their configurations to reach the goal while avoiding obstacles in the workspace. Simulations have been accomplished to show the effectiveness of the algorithm.

Keywords: Snake-like robot, path planning.

1. Introduction

Snake-like robots are typical examples of redundant robots with many degrees of freedom. They are intended to be intelligent and biologically-inspired mobile robots. Snake-like robots can change their configurations to achieve specific tasks using a number of links and joints on their long and thin bodies. They have become a widely used robot type and well-suited for a large number of applications [1].

Many issues on snake-like robots have been considered studying natural animals in the hope that structure and morphology of biological snakes may be applied to them [2]. Snake skeletons are made up of repeated small pieces whose motions are relatively limited with respect to each other. There are a few examples of snake-like robots having the ability of realistic snake-like motion simply because there are many problems on path planning and design of such serpentine robots [3].

Snake-like robots might be considered as a combination of mobile robots and highly redundant manipulators. They are mobile robots since their whole bodies move from one place to another. They are usually designed with wheels although there are examples of locomotion without the attributes of wheels [4]. Wheels do not appear in nature, but the idea of long body with many contact points has evolved from snakes. Snake-like robots are also highly redundant manipulators, because highly redundant manipulator motion resembles snake motion with relatively large number of degrees of freedom [5].

When a redundant robot has a large number of discrete links, it may be referred to as *serpentine* robot [6]. When a redundant robot bends continuously along its length, instead of having discrete joints and rigid links, it is known as *continuum* robot [7]. There has been lots of research on both types.

In this paper, a solution to the path planning problem for snake-like robots is proposed. The snake-like robot is presented with a highly redundant manipulator with discreet serial links. A smoothly curved path is determined from the snake-like robot's links to the goal using numerical potential fields. The snake-like robot performs a kind of serpentine motion on this path. The robot's head moves on the path while the rest of its body follows the head's trace by determining their configurations to reach the goal.

2. Finding global paths to the goal

A potential, $\Phi(r)$, is defined by the Laplace equation

$$\nabla^2\Phi = 0 \qquad (1)$$

in a closed region, Ω, of continuous, equal connectivity. $\Phi(r)$ does not have local minima. The Laplace equation in two dimensions can be represented on equally spaced and connected grid under Dirichlet boundary conditions by the following partial difference equation,

$$\Phi_{(i,j)} = (\Phi_{(i+1,j)} + \Phi_{(i-1,j)} + \Phi_{(i,j+1)} + \Phi_{(i,j-1)})/4\,, \qquad (2)$$

where i = position on the grid in the x direction, j = position on the grid in the y direction. The direction of the largest descent α is determined by means of the field values of surrounding grid points as follows;

$$\alpha = a\tan 2\left(\frac{\Phi_{(i,j-1)} - \Phi_{(i,j+1)}}{\Phi_{(i-1,j)} - \Phi_{(i+1,j)}}\right). \qquad (3)$$

Then using α, the x and y components of the straight line between two successive points are calculated; this line starting from the current point and ending at the next point is drawn in the direction of the largest descent.

3. Following the path points

Following the path means keeping the ends of the links on the path. Figure 1 shows a robot link and a small portion of the path. To follow the path, the end of the link, point B, is made to approach to the path points by calculating the shortest distance BC numerically.

Before the link moves, starting from the second point of the path, the distance from point B to this point is calculated and compared with the distance from point B to the first point. If the distance for the second point is shorter than the one for the first point, the procedure continues. The distance from point B to the third point is calculated and compared with the second one. If it is shorter than the second one, the next point is taken into account and the procedure continues. Otherwise, the procedure stops and the second point is taken as the shortest distance BC.

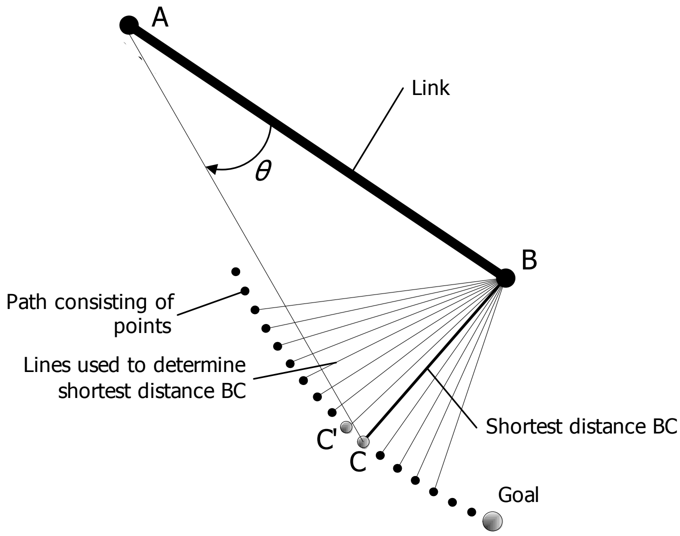


Figure 1. Finding shortest distance, angle and direction of rotation

Using the coordinates of point C and the shortest distance BC that have just been determined, distance AC can be calculated. Since all of the sides of triangle ABC are known, angle α is determined by means of cosines theorem. If α is more than a fixed value E, the link is rotated by E. If it is less than E, the link is rotated by α itself. In other words, the link is rotated as much as required, preventing shaking. One final note is that the above procedure is carried out for each link and its path at each increment.

3.1. Propagation procedure through links

The relationship between robot links is attained by way of a propagation procedure carried out through links. Suppose that the end-effector, i.e. snake's head, is taken as link i. It is moved repeatedly until its tip reaches the path. When the link reaches the path, the link is said to be settled. Then, link (i-1) is moved and link i is settled again. As soon as (i-1) is settled, link (i-2) is moved and both link (i-1) and link i are re-settled. This procedure continues until the base link which is the furthest link in the opposite side of the snake's head is reached. Therefore, the robot achieves first part of the snake-like motion.

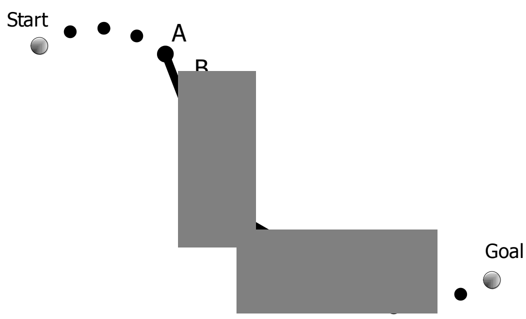


Figure 2. Two subsequent configurations of a two link snake-like robot

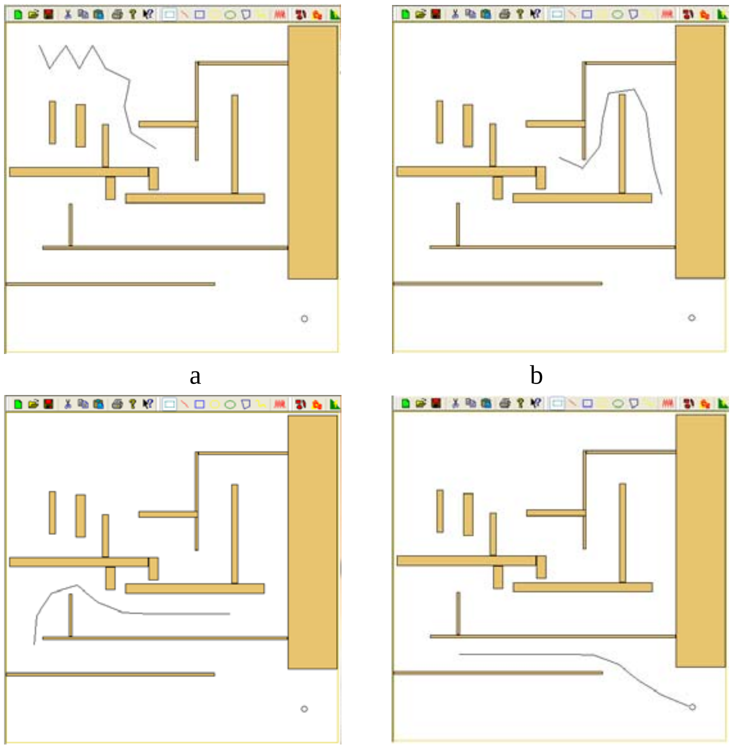
3.2. Moving the base of the snake-like robot

The snake-like robot starts to move and gradually takes the shape of the path (Figure 2). When the snake-like robot completely stretches, i.e. mapped on the path, the algorithm starts to move the robot's base from its current position to the next position on the path. Whenever the robot's base is moved to the next point on the path, all the links of the snake-like robot are adjusted in the way that the tip of each link is kept on the path points until the robot reaches the goal.

3.3. Computer Simulations

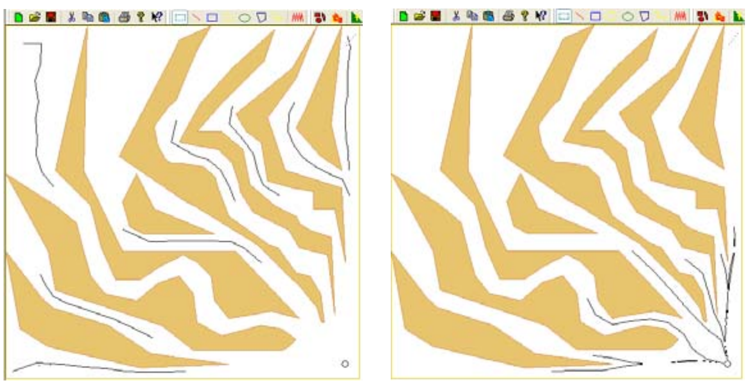
There are two examples of snake-like robots given in this section. A snake-like robot with 9 links in a W-space cluttered with many arbitrarily shaped obstacles is shown in Figure 3a.

Simulations are performed on a P4, 2.4 Mhz Laptop with 512 MB RAM. Figures 3a-d show the snapshots of the snake-like robot reaching the goal without colliding with obstacles.



a b c d

Figure 3. Snapshots of the snake-like robot for the first example



a b

Figure 4. Snapshots of the snake-like robot for the second example

The total time for the motion is about 2 seconds. In the second example, the workspace contains 8 snake-like robots with various numbers of links. Figures 4a-b show the snapshots of the snake-like robots reaching the goal point successfully without colliding with obstacles. As it can be seen from these examples, the algorithm presented here can employ many degrees of freedom.

4. Conclusions

In this paper, the path-planning problem for snake-like robots has been considered. The algorithm presented here guides snake-like robots to the goal through obstacles. These robots are modeled with highly redundant manipulators with discreet serial links. Using numerical potential fields, a smoothly curved path is determined from the snake-like robot's links to the goal. The snake-like robot performs a kind of serpentine motion on this path consisting of a number of points. As seen from the examples given, snake-like robots with many degrees of freedom follow the path to reach the goal. The algorithm is robust and implemented in real-time.

References

[1] **K.J. Kyriakopoulos, G. Migadis, K. Sarrigeorgidis.** The NTUA Snake: Design, Planar Kinematics, and Motion Planning. *Journal of Robotic Systems*, 16(1), 37-72, 1999.

[2] **K.J. Dowling.** Limbless Locomotion: Learning to Crawl with a Snake Robot. *PhD Thesis. Carnegie Mellon University. Pittsburgh, USA*, 1997.

[3] **A. Crespi, A. Badertscher, A. Guignard, A.J. Ijspeert.** AmphiBot I: an amphibious snake-like robot. *Robotics and Autonomous Systems*, 50(4), 2005, 163-175.

[4] **F. Matsuno, K. Suenaga.** Experimental Study on Control of Redundant 3D Snake Robot based on Kinematic Model. *Proceedings of the* 2*nd International Symposium on Adaptive Motion of Animals and Machine*, Kyoto, *March* 4-8, *Japan*, 2003.

[5] **M. Yamakita, M. Hashimoto, T. Yamada.** Control of Locomotion and Head Configuration of 3D Snake Robot (SMA). *Proceedings of the* 2003 *IEEE International Conference on Robotics & Automation, Taipei, Taiwan, September* 14-19, 2003.

[6] **E.S. Conkur, R. Buckingham.** Clarifying the definition of redundancy as used in robotics. *Robotica*, 15(5), 1997, 583-586.

[7] **M.W. Hannan, I.D. Walker.** Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots. *Journal of Robotic Systems*, 20(2), 2003, 45–63.

Received February 2008.