Proceedings of the 1997 IEEE
International Conference on Robotics and Automation
Albuquerque, New Mexico - April 1997

An Obstacle Avoidance Scheme for Hyper-Redundant Manipulators — Global Motion Planning in Posture Space —

Shugen Ma *and* Mototsugu Konno

Department of Systems Engineering, Faculty of Engineering
Ibaraki University, 4-12-1 Naka-Narusawa-Cho, Hitachi-Shi, 316 Japan

Abstract

Hyper-redundant manipulator has a very large or infinite degrees of kinematic redundancy, thus is possessed of unconventional features such as the ability to enter a narrow space while avoiding obstacles. In this paper, we propose a novel obstacle avoidance technique for the hyper-redundant manipulator to perform a payload-location task from point to point while avoiding the existing static obstacles in environment. The scheme is based on analysis in the defined posture space, where three parameters were used to determine the hyper-redundant manipulator configurations. The scheme is verified by computer simulation in case of using the model of the developed Hyper-R Arm. It shows that our method works perfect and the obstacles are well avoided globally.

1 Introduction

Hyper-redundant manipulators have a very large relative degrees of kinematic redundancy. These robots, which are analogous in design and operation to the "Trunk of an Elephant", have unconventional features such as the ability to enter a narrow space while avoiding obstacles. They, thus, are suitable for applications where a conventional industrial robot could never be introduced. However, the realization of such a hyper-redundant manipulator is difficult because there are serious engineering problems involved. One is weight, another is control of their large degrees of redundancy. A hyper-degree-of-freedom driving mechanism mounted on the arm makes the manipulator quite heavy. A powerful driving system needed to support such a long and heavy arm makes the arm even heavier. A novel tendon-driven manipulator mechanism [1] was thus proposed, which enables the lightweight manipulator to exhibit enormous payload capability. Further on, a hyper-redundant manipulator called "Hyper-R Arm" has also been developed by using this mechanism [3].

To solve another problem of controlling the hyper degrees of freedom, algorithms for the position-coordination of a hyper-redundant manipulator were introduced to resolve its kinematic redundancy [2]–[4]. In this paper, a novel scheme of obstacle avoidance will be introduced for hyper-redundant manipulator to perform a task in unconstructive environment. The scheme is based on analysis in the defined posture space, and has a number of advantages over other possible ones. First, this scheme can be used for real-time control of hyper-redundant manipulator while it is working in a static environment. Second, distributed control is easily applied by this scheme. In this paper, we shall also show the validity of the technique by computer simulation in the case of using the Hyper-R Arm model.

The structure of the remainder of the paper is as follows: in Section 2, we shall introduce the mechanism and control method of a hyper-redundant manipulator. In Section 3, we shall discuss the obstacle avoidance algorithms, and propose a novel obstacle avoidance technique for hyper-redundant manipulators. The computer simulations were performed to verify the validity of the proposed technique and its result is given in Section 4. The conclusion of the paper is given at Section 5.

2 Mechanism and Control of a Hyper-Redundant Manipulator

In this section, we shall introduce the mechanism of a mechanical model of hyper-redundant manipulator called "Hyper-R Arm". The control scheme of the hyper-redundant manipulator, which is based on the curvilinear theory, will be also introduced here.

2.1 Mechanism of a Hyper-redundant manipulator

To solve the weight problem, a hyper-redundant manipulator named "Hyper-R Arm" has been developed [3], which is shown in figure 1. It consists of 10 free connected serial links. One pair of tendons for driving link i ($i = 1, \cdots, n$) around joint i is fixed at pulley i, which is riveted on link i while freely rotating around joint i, at one end of them; then they are wound along the way of the pulley of joints $i-1$ through 1 in the opposite direction, to be extended to the base. In this mechanism, when generation of torque τ_i at joint i is attempted, the same torque is inevitably generated to joints 1 through $i-1$ (*in case of uniform pulleys' radius*). As far as joint i is connected, power transmission pulleys to drive joints $i-1, \cdots, 1$ of the tip side are pivoted to this joint and therefore torques produced to drive these joints are accumulated at joint i. As a result, we know that the tendon tractions are positively coupled in the mechanism [1].

The Hyper-R Arm shown in figure 1 has 10 degrees of freedom at the arm, and is driven by 20 DC motors

through tendons and pulleys; the length of the arm is about 0.8 meters and the total arm weight is about 4.047 kilograms.

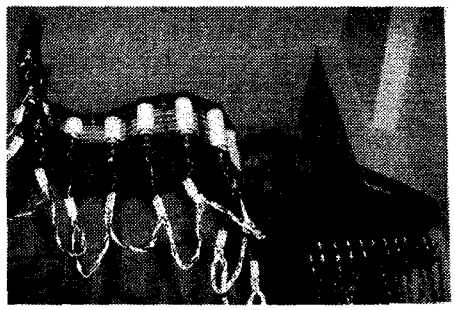


Figure 1: Model of a hyper-redundant manipulator: Hyper-R Arm

Preliminary tests on the Hyper-R Arm confirmed the excellence of its basic movement performance and the following characteristics were demonstrated [3].

- Large payload capability
- Variable compliance arm structure

2.2 Control of a Hyper-redundant manipulator

Traditionally, kinematic analysis and motion planning for redundant manipulators has relied upon a pseudoinverse [7] [8], generalized inverse [9] [10], or extended inverse [11] of the manipulator Jacobian matrix. These schemes, however, are difficult to the real-time control of hyper-redundant manipulators, because of their computational inefficiency in manipulation of matrices. A position-coordination scheme to resolve the kinematic redundancy in hyper-redundant manipulators has been proposed on the basis of curvilinear theory [2]-[6]. This scheme has been used to our Hyper-R Arm, to clarify its capability to the real-time control of the manipulator.

Continuous curvilinear theory models the posture of hyper-redundant manipulator arm with the curvature function $\kappa(s)$, which represents the curvature of the arm at distance s along the curve measured from the base. As we explored, the serpenoid curve, which is defined for the snake wave motion [5] [6], is better used to define the arm posture of hyper-redundant manipulators, because of the easily obtained inverse solution from the given boundary position and the length of curve to the structure of curve, in the case that we define the solution of boundary position derived from the known curve structure as direct solution.

Consider a serpenoid curve, the curvature of which is defined as[1]

$$\kappa(s) = \frac{2\pi}{\ell} a_1 \cos(\frac{2\pi}{\ell} s) + \frac{2\pi}{\ell} a_2 \sin(\frac{2\pi}{\ell} s) \qquad (1)$$

where a_1 and a_2 are the coefficients to define the curvature function, and ℓ is the curve length which is equal to the length of the manipulator arm that is to be configured. The angle $\alpha(s)$, which represents the inclination angle of the vector in relation to the x-axis on the curvilinear length s, and the end-point position of x, y coordination $(x(\ell), y(\ell))$ can be derived and obtained by

$$\begin{aligned} \alpha(s) &= \alpha_0 + \int_0^s \kappa(s) ds \\ &= \alpha_0 + a_1 \sin\frac{2\pi}{\ell} s - a_2 \cos\frac{2\pi}{\ell} s + a_2, \quad (2) \\ x(\ell) &= \int_0^\ell \cos(\alpha(s)) ds \\ &= \cos(a_2 + \alpha_0) J_0 \left(\sqrt{a_1^2 + a_2^2} \right) \ell, \quad (3) \\ y(\ell) &= \int_0^\ell \sin(\alpha(s)) ds \\ &= \sin(a_2 + \alpha_0) J_0 \left(\sqrt{a_1^2 + a_2^2} \right) \ell \quad (4) \end{aligned}$$

where α_0 is the initial inclination angle of the vector in relation to the x-axis on the start point, and J_0 is the zero-order Bessel function [12].

While the end position of the curve is given, its structure (*or posture*) given by the curvature $\kappa(s)$ corresponding to the given initial inclination angle α_0 can be defined by deriving the coefficients a_1 and a_2. The coefficients a_1 and a_2 are obtained by inverting equation (3) and (4), and are given by

$$\begin{aligned} a_2 &= \tan^{-1}\left(\frac{y(\ell)}{x(\ell)}\right) - \alpha_0, \\ a_1 &= \left(\left[J_0^{-1}\left(\left(x(\ell)^2 + y(\ell)^2\right)^{\frac{1}{2}} / \ell \right) \right]^2 - a_2^2 \right)^{\frac{1}{2}} \quad (5) \end{aligned}$$

where J_0^{-1} is the restricted inverse zero-order Bessel function [12]. Thus, the inverse solution of the curve (*the structure of the curve*) is derived and defined by the coefficients a_1 and a_2. Of course, if we changed the initial inclination angle α_0, the structure of the curve would be changed too, as shown in equation (6). Thus, the structure of the serpenoid curve is decided by three variables a_1, a_2, and α_0.

The hyper-redundant manipulator posture can be configured by restricting the arm on the defined serpenoid curve. In this case, the joint angles of the manipulator to be configured can be expressed as

$$\begin{aligned} q_1 &= \alpha(\frac{L}{2}) = a_1 \sin(\frac{\pi}{N}) + a_2 \left(1 - \cos(\frac{\pi}{N})\right) + \alpha_0, \\ q_i &= \alpha\left((i - 1 + \frac{1}{2})L\right) - \alpha\left((i - 1 - \frac{1}{2})L\right) \\ &= a_1 \left[\sin(\frac{\pi}{N}(2i-1)) - \sin(\frac{\pi}{N}(2i-3))\right] \quad (6) \\ &\quad -a_2 \left[\cos(\frac{\pi}{N}(2i-1)) - \cos(\frac{\pi}{N}(2i-3))\right], \\ &\quad \textit{where} \quad i = 2, 3, \cdots . N \end{aligned}$$

[1] $\kappa(s) = A \sin(\frac{2\pi}{\ell} s + \phi)$; $A = \frac{2\pi}{\ell}\sqrt{a_1^2 + a_2^2}$, $\phi = \tan^{-1}\frac{a_1}{a_2}$

Wherein N is the joint number of the manipulator and L is the length of the link, equal to ℓ/N.

As a result, the inverse kinematic solution, in which the joint angles of the hyper-redundant manipulator that needs to be configured are derived from the given end position $(x(\ell), y(\ell))$, has been derived by equations (5) and (6). It should be understandable that the scheme is advanced in the computational cost, and makes the real-time position coordination control of hyper-redundant manipulators possible.

3 Obstacle Avoidance Motion Planning in the Posture Space

As stated in introduction, the hyper-redundant manipulators have unconventional features such as the ability to enter a narrow space while avoiding obstacles. In this section, we discuss the obstacle avoidance algorithms of the hyper-redundant manipulator.

Obstacle avoidance approaches for non-redundant manipulator have widely discussed [13]–[20], and the trajectories for the manipulator to avoid obstacles were generated in the workspace [13]–[16], and/or some in the configuration space (*joint space generated by joint angles changes*) [17]–[20]. Moreover, the algorithms to generate obstacle avoidance trajectory for traditional redundant manipulators were also proposed from analyzing the workspace [21]–[24]. However, no more obstacle avoidance algorithms for hyper-redundant manipulator have been discussed, except one in the workspace analysis [6]. In this paper, we propose a new, novel obstacle avoidance scheme, which is based on the analysis of the posture space that will be defined later.

The posture of the hyper-redundant manipulator is decided through three variables a_1, a_2, and α_0. That is, while giving the values of the a_1, a_2, and α_0, the posture of the manipulator arm is decided directly through the equation (6). The posture space of the hyper-redundant manipulator is thus defined as follows:

> *Definition:* A **posture** æ (a_1, a_2, α_0) of hyper-redundant manipulator is a specification of the posture (*or structure*) of the hyper-redundant manipulator in workspace. The **posture space** of hyper-redundant manipulator is the space Æ of all possible postures of the hyper-redundant manipulator. A unique posture of Æ is arbitrarily selected (*But* $a_1 = 0$, $a_2 = 0$, $\alpha_0 = 0$ *is generally selected*) and is called the **reference posture** of hyper-redundant manipulator. It is denoted by ø.

The algorithm for hyper-redundant manipulator to avoid the static obstacles, proposed in this paper can be briefly described as

- Map the workspace of hyper-redundant manipulator onto its posture space;
- Create the obstacle avoidance trajectories in the posture space;
- Find out the optimal one from the all possible obstacle avoidance trajectories;
- Perform the task in the workspace where the static obstacles exist.

Next, we discuss them step by step.

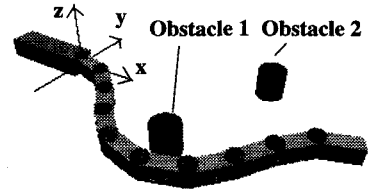


Figure 2: A hyper-redundant manipulator in a workspace where the static obstacles exist

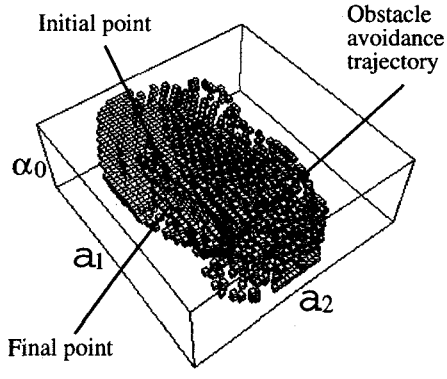


Figure 3: Obstacles in the posture space, mapped from the obstacles in workspace shown in figure 2, and the obtained obstacle avoidance trajectory in the posture space

3.1 Obstacle map in the posture space

The variables a_1, a_2, and α_0, which configure the manipulator posture, generate the posture space Æ of the hyper-redundant manipulator. Through changing arbitrarily the values of a_1, a_2, and α_0 between their boundary and finding out the collision data with obstacles, obstacle map in posture space can be generated. Figure 3 show one example of the obstacle map in posture space corresponding to that in workspace shown in figure 2. It should be noted here that, the obstacle data shown in figure 3 is one after cell decomposition [25] [26]. Through the cell decomposition of the original continuous data, the operation to generate the obstacle avoidance trajectories would become easier. In this paper, we thus use the cell data to plan a collision free trajectory of the hyper-redundant manipulator from obstacles.

3.2 Obstacle avoidance trajectory generation algorithms in the posture space

To generate an obstacle-collision free trajectory in 3-dimensional posture space, generally, is extremely complicated. Here, we first discuss an obstacle avoidance problem in 2-dimensional posture space, then extend it to 3-dimensional posture space.

3.2.1 2-D obstacle-collision free trajectory

In the case of 2-D posture space, shown in figure 4, the obstacle avoidance trajectory (OAT) is generated by following algorithm.

2-D OAT Generation Algorithm

1) Draw a straight line from initial point **S** to end point **E**, and find out all intersection points $\mathbf{S}'_i$ and $\mathbf{E}'_i$ with obstacle i ($i = 1, \cdots, K$; K: number of obstacles);
2) Calculate two parts of each obstacle's contours connected the points $\mathbf{S}'_i$ and $\mathbf{E}'_i$;
3) Connect each part of trajectories (*straight line from* **S** *to* $\mathbf{S}'_1$, *each obstacle's contours from* $\mathbf{S}'_i$ *to* $\mathbf{E}'_i$, *straight line from* $\mathbf{E}'_{i-1}$ *to* $\mathbf{S}'_i$, *and straight line from* $\mathbf{E}'_K$ *to* **E**) to generate the 2^K obstacle-collision free trajectories;
4) Select the shortest one among the above 2^K trajectories.

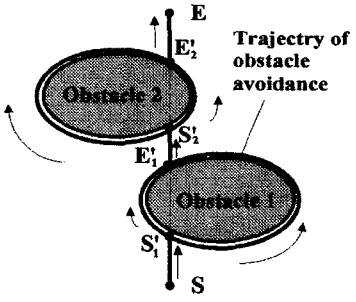


Figure 4: 2-D obstacle avoidance trajectories in the case of two obstacles

3.2.2 3-D obstacle-collision free trajectory

In case of the simple shape of obstacles, the OAT generation algorithm like the 2-D one would be work. However, it is not the case for common obstacles because the obstacle's shape is almost complicated. Here we propose the following algorithm to generate 3-D OAT. Before we introduce the algorithm, let us define some terms used in the algorithm. The coordinate of the initial point **S** in posture space, as shown in figure 5 is defined by $(a_1^s, a_2^s, \alpha_0^s)$, the coordinate of the end point **E** by $(a_1^e, a_2^e, \alpha_0^e)$, and the coordinates of the initial & end points $\mathbf{S}_p, \mathbf{E}_p$ projected onto the α_{0p} plane by $(a_{1p}^s, a_{2p}^s, \alpha_{0p}^s)$ & $(a_{1p}^e, a_{2p}^e, \alpha_{0p}^e)$, respectively.

3-D OAT Generation Algorithm

1) Project the initial point **S** and the end point **E** to an arbitrarily plane $\alpha_0 = \alpha_{0p}$, to generate the projected the initial point $\mathbf{S}_p$ and the end point $\mathbf{E}_p$ on the plane; If the points $\mathbf{S}_p$ or $\mathbf{E}_p$ exist in the obstacles, then find out the nearest point of them on the contour of the obstacles;
2) Generate the OAT in the plane $\alpha_0 = \alpha_{0p}$ by 2-D OAT generation algorithm mentioned-above[2];
3) Shift the point from the initial point **S** to $\mathbf{S}_p$ and from $\mathbf{E}_p$ to the end point **E** along the straight lines connected them, to generate the OAT from **S** to $\mathbf{S}_p$ and that from $\mathbf{E}_p$ to **E**; In the shifting process, if the point on the line go in the obstacles, find out the nearest point $\mathbf{P}_i$ of it on the contour of the obstacles, then reshift the point from $\mathbf{P}_i$ along the straight line connected $\mathbf{P}_i$ & $\mathbf{S}_p$ and/or $\mathbf{P}_i$ & **E**;
4) Connect each part of trajectories to generate the 3-D obstacle-collision free trajectory.

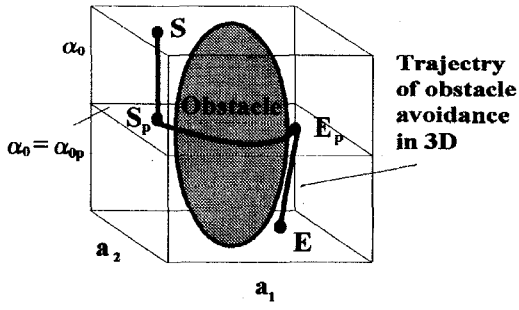
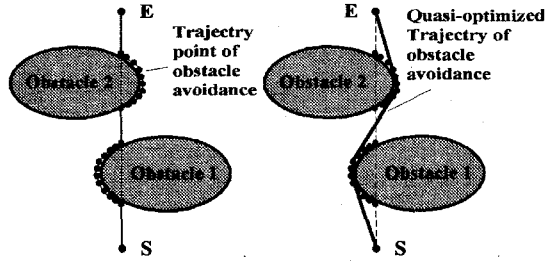


Figure 5: 3-D obstacle avoidance trajectories

3.3 Quasi-optimization of the obstacle avoidance trajectories

The obstacle avoidance trajectories in the posture space generated by above-stated algorithms are along the contour of the obstacles in part. They must be not the shortest ones in global meaning. In this subsection, we introduce a method to quasi-optimize the obtained OATs.



(a) Before optimization (b) After optimization

Figure 6: Obstacle avoidance trajectories in the posture space

Suppose that we have an OAT shown in figure 6 (a), then the quasi-optimized OAT can be easily derived by finding out the points on the obstacles' contour that are tangential to the obstacles by the lines, and connecting each part of trajectories (*tangential lines*

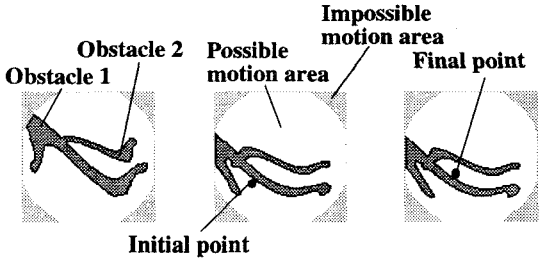
[2]It should be pointed out here, that the 3-D OAT generation algorithm proposed only works in the case that the 2-D OAT must exist in the plane $\alpha_0 = \alpha_{0p}$.

and obstacles' contours). Figure 6 (b) shows one simple example, the method, however, can be also used to the complicated 2-D cases and the most 3-D cases.

While the obstacle avoidance trajectory in the posture space is obtained, as stated-above, the OAT in the joint space (*or configuration space*) is easily generated from the equation (6) and that in the workspace from the equations (3) and (4).

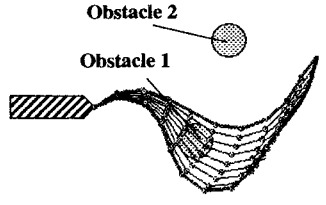
4 Simulation Examples

We shall use a hyper-redundant manipulator "Hyper-R Arm" shown in figure 1 to simulate the proposed obstacle avoidance trajectory generation algorithms. The length of each link of the arm is $0.08[m]$, and the obstacles are simply given by two circles with radius $0.05[m]$ centered at $(0.3[m], -0.1[m])$ and $(0.4[m], 0.2[m])$, as shown in figure 2.

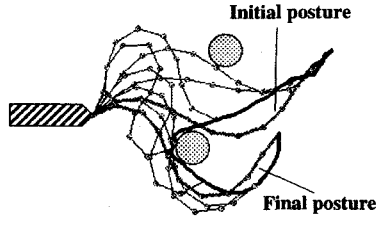


$\alpha_0 = 0.0[rad]$ $\alpha_0 = 0.698[rad]$ $\alpha_0 = 0.785[rad]$

Figure 7: Examples of the obstacles on the some (a_1, a_2) planes



(a) Without obstacle avoidance scheme



(b) With obstacle avoidance scheme

Figure 8: Resulted manipulator postures in the workspace

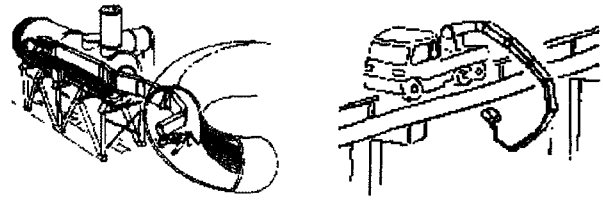
Figure 3 shows the obstacles and the obstacle avoidance trajectory derived by the proposed algorithm in the posture space, where the cells, were decomposed with the range $-2.0 < a_1, a_2 < 2.0$, $-1.571[rad] < \alpha_0 < 1.571[rad]$ by the interval $\delta a_1 = \delta a_2 = 0.04$ and $\delta\alpha_0 = 0.087[rad]$. Their shapes on the (a_1, a_2) planes with $\alpha_0 = 0.0[rad]$, $\alpha_0 = 0.698[rad]$, and $\alpha_0 = 0.785[rad]$ are also shown in figure 7 for clearly understanding.

The manipulator postures in workspace, resulted from the trajectory shown in figure 3, are shown in figure 8 (b). For comparison, the manipulator postures without the obstacle avoidance algorithm are shown in figure 8 (a). From the result, we clearly know that our algorithm works perfect, and the obstacles existed in static state are well avoided globally.

At last, we make an issue of the computation time. In case of the NEC PC-9821Bp (486DX2-66MHz), the mapping of the obstacles and the possible motion area from the workspace to the posture space totally takes about 20 minutes, however, the generation of the obstacle avoidance trajectory only takes 0.36 seconds. This trajectory, moreover, is generate in advance. After we have the trajectory, the manipulator can be distributedly controlled to track it. Actually, the trajectory-tracking time is only 0.21 seconds. Thus, the real-time control should be possible. This was also proven by experimental test, in the case of using the developed Hyper-R Arm shown in figure 1.

5 Conclusions

In this paper, we proposed a novel obstacle avoidance technique for the hyper-redundant manipulator to perform a payload-location task from point to point while avoiding the existing static obstacles in environment. The new obstacle avoidance scheme is based on analysis in the posture space, where three parameters were used to determine the hyper-redundant manipulator posture. The computer simulations were executed, and show that our method works good and it makes the real-time obstacle avoidance control possible in the case of the static obstacles in environment.



(a) Maintenance robot for nuclear reactor

(b) Maintenance robot for highway/freeway

Figure 9: Application examples of the 2D hyper-redundant manipulator

The discussion in this paper was limited to 2 dimensional problems in order to understand clearly the analysis technique of the posture space. However, if we consider the applications shown in the figure 9, the algorithm proposed here can be directly used and should be very useful. In addition, we have concentrated the discussion onto the real-time obstacle avoidance control. The shortest collision free trajectory in the posture space stated in section 3 should be not the shortest one in the workspace, and is not optimal

in energy consumption. The following topics will be discussed in future studies.

- Obstacle avoidance control problem of the hyper-redundant manipulator in the 3 dimensional environment;
- Obstacle avoidance control problem with consideration of the energy consumption and manipulator dynamics.

References

[1] S. Hirose and S. Ma, "Coupled tendon-driven multijoint manipulator", Proc. IEEE Int. Conf. on Robotics and Automation, vol.2, pp.1268-1275, 1991

[2] S. Hirose, K. Yokoshima, and S. Ma, "2-DOF Moray drive for hyper-redundant manipulators", Proc. Int. Conf. on Intelligent Robots and Systems, vol.2, pp.1753-1740B, 1992

[3] S. Ma, M. Konno, H. Yoshinada, and Y. Tsutsumi, "Development of a hyper-redundant manipulator", Proc. 7th JSME Annual Conf. on Robotics and Mechatronics, vol.B, pp.760-763, 1995 (in Japanese)

[4] S. Ma, S. Hirose, and H. Yoshinada, "Development of a hyper-redundant manipulator for maintenance of nuclear reactors", Int. J. of Advanced Robotics, vol.9, no.3, pp.281-300, 1995

[5] S. Hirose, *Biologically Inspired Robots*, Oxford, Oxford University Press, 1993

[6] G. S. Chirikjian and J. W. Burdick, "An obstacle avoidance algorithm for hyper-redundant manipulators", Proc. IEEE Int. Conf. on Robotics and Automation, vol.1, pp.625-631, 1990

[7] C. A. Klein and C. H. Huang, "Review of the pseudoinverse for control of kinematically redundant manipulators", IEEE Trans. on Syst., Man, Cyber., vol.13, no.2, pp.245-250, 1983

[8] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multi-body mechanisms", IEEE Trans. on System, Man, and Cybernetics, vol.7, no.12, pp.868-871, 1977

[9] D. N. Nenchev, "Redundancy resolution through local optimization: a review", Journal of Robotic Systems, vol.6, no.6, pp.769-798, 1989.

[10] T. Shamir and Y. Yomdin, "Repeatability of redundant manipulators: Mathematical solution of the problem", IEEE Trans. on Automatic Control, vol.33, no.11, pp.1004-1009, 1988

[11] J. Baillieul, "Kinematic programming alternatives for redundant manipulators", Proc. IEEE Int. Conf. on Robotics and Automation, vol.2, 1985

[12] I. N. Sneddon, *Special Functions of Mathematical — Physics and Chemistry —*, London: Oliver & Boyd, 1961

[13] K. Kudo, M. Takata, and T. Sasaki, "An algorithm for searching the route of an artificial hand", Trans. of the Society of Instrument and Control Engineer, vol.11, no.4, pp.29-34, 1975 (in Japanese)

[14] H. Ozaki, A. Mohri, and M. Takata, "Planning of collision free movement of a manipulator considering its body space", Trans. of the Society of Instrument and Control Engineer, vol.18, no.9, pp.72-79, 1982 (in Japanese)

[15] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", Int. J. of Robotics Research, vol.5, no.1, pp.90-98, 1986

[16] T. Tsuji, P. G. Morasso, K. Shigehashi, and M. Kaneko, "Motion planning for manipulators using artificial potential field approach that can adjust convergence time of generated arm trajectory", J. of the Robotics Society of Japan, vol.13, no.2, pp.285-290, 1995 (in Japanese)

[17] T. Lozano-Perez, "Spatial planning: A configuration space approach", IEEE Trans. on Computers, vol.32, no.2, pp.108-120, 1983

[18] T. Lozano-Perez, "A simple motion-planning algorithm for general robot manipulators", IEEE Trans. on Robotics and Automation, vol.3, no.3, pp.224-238, 1987

[19] T. Nagata and I. Hara, "Describing moving obstacles in a configuration space", Proc. Annual Conf. of the Robotics Society of Japan, pp.513-514, 1991 (in Japanese)

[20] N. Kida, H. Yano, and Noborio, "A path-planning algorithm of a six-freedom robotic manipulator for avoiding some obstacles or singular configuration based on the configuration graph", Proc. Annual Conf. of the Robotics Society of Japan, pp.517-520, 1991 (in Japanese)

[21] M. Kircanski and M. Vukobratovic, "Contribution to control of redundant robotic manipulators in an environment with obstacles", Int. J. of Robotics Research, vol.5, no.4, pp.112-123, 1986

[22] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based on redundancy control of robot manipulators", Int. J. of Robotics Research, vol.6, no.2, pp.3-15, 1987

[23] H. Zghal, R. V. Dubey, and J. A. Euler, "Collision avoidance of a multiple degree of redundancy manipulator operating through a window", ASME Trans. on Dyn. Syst., Meas. Control, vol.114, pp.717-720, 1992

[24] R. V. Mayorga, F. Janabi-Sharifi, and A. K. C. Wong, "A fast approach for the robust trajectory planning of redundant manipulators", J. of Robotic Systems, vol.12, no.2, pp.147-161, 1995

[25] Y. Liu and H. Onda, "Constructing an approximate representation of a configuration space without using any intersection check", Trans. of the Society of Instrument and Control Engineer, vol.31, no.1, pp.89-97, 1995 (in Japanese)

[26] C. Marolo and E. Pagello, "A cell decomposition approach to motion planning based on collision detection", Proc. Int. Conf. on Advanced Robotics, vol.1, pp.481-488, 1995