Obstacle avoidance of redundant discretely-actuated manipulators using workspace density functions

E. Lanteigne[1], A. Jnifene[2]

[1] *Department of Mechanical Engineering, Royal Military College of Canada, Eric.Lanteigne@rmc.ca*

[2] *Department of Mechanical Engineering, Royal Military College of Canada, Amor.Jnifene@rmc.ca*

Abstract

This paper presents an obstacle avoidance method for a discretely-actuated planar manipulator with a large number of serially connected modules. The method uses a combination of workspace density functions and obstacle density maps to solve the inverse kinematics. Module states are determined sequentially, starting at the base of the manipulator up to the end-effector. At each iteration, the workspace density at the target location is computed for each module state, these values are then scaled according to the frequency of perceived obstacles around the location of the current module, and the module state with the largest scaled density is selected. The advantage of the method is that the environmental information generated by range sensors on the manipulator surface could be used directly by the algorithm, thus avoiding the geometrical modelling of obstacles and the computation of the nearest point distance between the manipulator and the obstacles. This paper provides a description of the proposed algorithm and presents the simulation results of a seventeen module planar manipulator operating in an obstacle field.

Keywords: redundant manipulator, density functions, obstacle avoidance.

1 INTRODUCTION

Hyper-redundant manipulators are highly dexterous robotic devices typically composed of numerous serially connected modules and actuators. The large number of degrees of freedom allows the manipulator to avoid obstacles and to navigate through narrow passages. The high mobility of redundant manipulators increases the complexity of both the inverse kinematics and the obstacle avoidance problem. The challenge is determining a suitable solution in a reasonable amount of time.

There are various different approaches to both these problems in the literature. For only several degrees of freedom, resolved-motion rate inverse kinematics methods based on the pseudoinverse were proposed under various formulations [1]–[4]. In one way or another, these solutions yield a minimum norm solution to the primary motion task and a homogeneous solution used for secondary tasks. Obstacle avoidance strategies based on the Jacobian matrix generally involved maximizing the distance between the manipulator link closest to the obstacle vertex [5] [6].

The artificial potential field method initially developed for mobile robots [7] [8] was adapted to redundant manipulators in [9] [10]. This method is based on a field of forces interacting with the manipulator structure. The target coordinate acts as an attractive pole to the end-effector and obstacles act as repulsive surfaces to the manipulator. A typical disadvantage of applying this method to robotic manipulators is that obstacles must be modeled using analytical equations. Modelling the obstacles as mesh grids in which each mesh grid points exert a repulsive force to the manipulator structure was proposed to overcome this problem [11]. It is to be noted that many of these strategies treat obstacles as convex polyhedrons, spheres or ellipsoids to avoid local minimum solutions [15]-[17]. Although these methods are compact and concise, they all rely on the Jacobian and its pseudoinverse which has demonstrated to be computationally inefficient for manipulators with a large number of degrees of freedom. In either case a large number of constraint tasks or potential functions must be optimized to solve for the redundant modules or a large number obstacles.

The backbone curve method was developed to limit the number of parameter in the optimization problem. The inverse kinematics is reduced to finding the mode shape functions describing a continuous or piecewise continuous curve fixed to the manipulator structure or backbone [12] [13]. The application of this method involves fixing the number of mode shape functions according to the number of obstacles in the workspace. The shape functions are sequentially separated into "free" sections and obstacle avoiding sections which traverse "virtual tunnels" derived from the obstacle field [14]. The generation of the tunnels is non-intuitive for complex obstacle fields and not easily adaptable to time-varying obstacle fields.

Workspace density functions approximate the number of forward kinematic solutions of discretely-actuated hyper-redundant manipulators in the workspace by dividing the workspace into small volume elements and recording the number of poses in each element [18]. Workspace density functions describe how accurately a position and orientation can be reached by the end-effector. These functions are used in conjunction with a breadth-first search algorithm to quickly solve the inverse kinematics problem [19]. The method works sequentially from the base of the manipulator to the end-effector module. For each module, the state resulting in the highest density at the end-effector coordinate is selected as the optimal state and used in the computation of the subsequent module states in the manipulator chain.

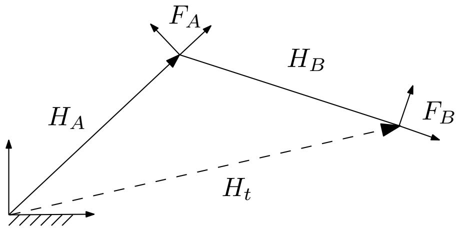


Figure 1: Concatenation of two homogeneous transformations

In light of the current interest in mobile robots, several heuristic methods have been proposed that are analogous to the workspace density based method. The vector field histogram method uses a two-dimensional Cartesian grid generated from range sensors aboard a mobile robot for the representation of the obstacles [24] [25]. From the Cartesian grid, a polar histogram function quantifying the obstacle density around the robot is generated, and the direction with the lowest obstacle density is chosen.

In this paper we present a variation on the breadth-first search inverse kinematics algorithm to included an obstacle avoidance strategy. The approach builds on the previous research by applying the information contained in the two-dimensional grid representing the obstacle field to the densities computed from the workspace density functions. For each module state, the density at the end-effector coordinate is scaled with an obstacle density factor representing the obstacle density around the current module. The following sections describe the generation of the workspace density functions, the breadth-first search inverse kinematics algorithm, and the obstacle avoidance strategy. Simulation results are presented at the end of the paper using a manipulator with seventeen modules operating in an obstacle field consisting of uniformly-distributed circles.

2 Workspace density functions

Workspace density functions quantify a manipulator's ability to reach a given target coordinate. The density at each location in Euclidian space reflects the number of reachable frames at that location [20]. The density function ρ of a discretely-actuated module is obtained by dividing the workspace into small elements and recording the number of end-effector frames in each element. The generation of the density function is irrespective of the module structure or actuation method; modules may be composed of parallel platforms, single actuator revolute or prismatic joints, or any combination of the above. The density of a serial manipulator of any length can be obtained by the sequential convolution of the density functions of each of its modules [21].

Referring to Figure 1, F_A and F_B are moving frames attached to the distal end of two modules described by the homogeneous transformations H_A and H_B respectively. The transformation H_A describes the position and orientation of F_A with respect to the fixed reference frame and H_B describes the position and orientation of F_B with respect to F_A. The coordinates of F_B with respect to the fixed reference frame can then be described by $H_t = H_A H_B$. The homogeneous transformation H_B can then also be written as:

$$H_B = H_A^{-1} H_t \tag{1}$$

The density function $\rho(H_t)$ of F_B with respect to the fixed frame can be obtained from the convolution of the density functions $\rho_A(H_A)$ and $\rho_B(H_B)$ of the two modules [18] [21].

$$\rho(H_t) = (\rho_A * \rho_B)(H_t) = \int_{SE(D)} \rho_A(H_A)\rho_B(H_A^{-1}H_t)d\mu(H_A) \tag{2}$$

where $SE(D)$ denotes the group of rigid body motions of N-dimensional space and $d\mu(H_A)$ is the integration measure on the motion group. In the case of a planar manipulator $N = 2$, and $d\mu = dxdyd\theta$. Given the transformations H_A and H_t

$$H_A = \begin{pmatrix} \cos\alpha & \sin\alpha & \xi \\ -\sin\alpha & \cos\alpha & \eta \\ 0 & 0 & 1 \end{pmatrix} \text{ and } H_t = \begin{pmatrix} \cos\theta & \sin\theta & x \\ -\sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

the convolution (2) can be written as

$$(\rho_A * \rho_B)(x, y, \theta) = \\ \iiint_{x,y,\theta} \rho_A(\xi, \eta, \alpha)\rho_B \begin{pmatrix} (x-\xi)\cos\alpha + (y-\eta)\sin\alpha, \\ -(x-\xi)\sin\alpha + (y-\eta)\cos\alpha, \\ \theta - \alpha \end{pmatrix} dxdyd\theta \tag{4}$$

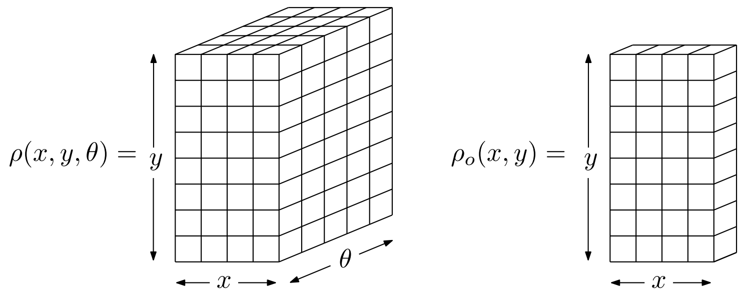
In the numerical simulations, the workspace density function are approximated by a piece-wise constant histograms in the form of Figure 2a, and the convolution (4) can be approximated by a Riemann-Stieltjes sum [18].

$$(\rho_A * \rho_B)(x, y, \theta) \simeq \\ \sum_{l=1}^{L}\sum_{m=1}^{M}\sum_{n=1}^{N} \rho_A(\xi_l, \eta_m, \alpha_n)\rho_B \begin{pmatrix} (x-\xi_l)\cos\alpha + (y-\eta_m)\sin\alpha, \\ -(x-\xi_l)\sin\alpha + (y-\eta_m)\cos\alpha, \\ (\theta - \alpha_n)\text{mod}2\pi \end{pmatrix} \Delta x\Delta y\Delta\theta \tag{5}$$

where L, M, and N represent the size of the workspace density function histogram. The workspace density function of a manipulator with P modules can be obtained from the sequential convolution of the density functions of each module in the manipulator, from the base module $\rho_{0,1}$ to the end-effector module $\rho_{P-1,P}$.

$$\rho_{0,P}(x, y, \theta) = (\rho_{0,1} * \rho_{1,2} * ... * \rho_{P-1,P})(x, y, \theta) \tag{6}$$

The convolution (6) becomes computationally unfeasible for manipulators with many modules since the density function size increases with the number of modules and the size of the workspace. Several methods have been developed to overcome this problem such as the conversion of the density function to a Fourier transform [21] and the use of a diffusion-type equation [22].



a) Manipulator workspace density b) Obstacle density map

Figure 2: Workspace density functions

3 Breadth-first search inverse kinematics

The breadth-first search inverse kinematics algorithm involves determining and fixing the state of each module sequentially, starting at the base module of the manipulator, while maximizing the density of the remaining modules at the desired coordinates for each step [18]. Given the desired target coordinate $H_d = f(x_d, y_d, \theta_d)$, the density

$$\rho_{1,P}((H_1^S)^{-1} H_d) \tag{7}$$

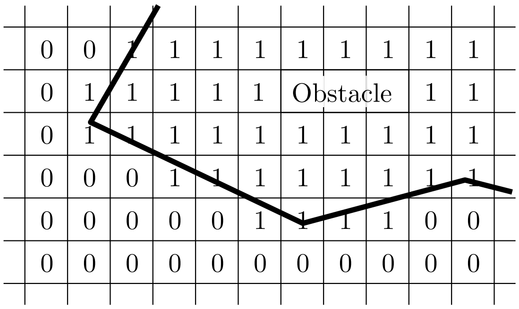
is computed for each state S of the first module, where H_1^S is the transformation of one state of the first module. The configuration $H_1^C = H_1^S$ which maximizes $\rho_{1,P}$ is selected and the state of the second module is found by computing

$$\rho_{2,P}((H_1^C H_2^S)^{-1} H_d) \tag{8}$$

for each state S of the second module, where the transformation H_1^C is the optimal state of the first module and H_2^S is the transformation of one state of the second module. The configuration $H_2^C = H_2^S$ which results in the highest density $\rho_{2,P}$ is selected and the process is repeated until the end-effector module is reached. Using this method, the computation time of the inverse kinematics is proportional to the number of modules in the manipulator.

4 Obstacle avoidance using workspace density functions

Obstacles in the workspace can easily be represented by probability distribution maps based on data is generated from range sensors such as the ultrasonic or infrared sensors used by mobile robots. Each cell in the obstacle density map quantifies the confidence in the existence of an obstacle at that location. In the case of a planar manipulator or mobile robot operating on a plane surface, the obstacle density map ρ^O would take the form of Figure 2b. There several different methods for producing these maps in the literature such as the histogram grid or certainty grid algorithms [7] [26]. In the case of the present analysis, simulated obstacle density maps are used. Shown in Figure 3 is an example of a simulated obstacle. The cells containing an obstacle have a value of one and cells in free areas have a value of zero.



0	0	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	Obstacle			1	1
0	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0

Figure 3: Obstacle density map

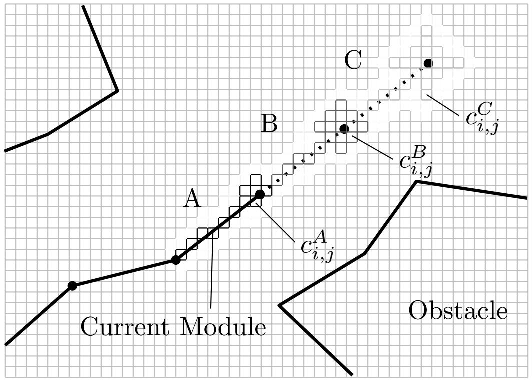


Figure 4: Active cells of the obstacle density map for one module state

At each iteration, the density of the remaining number of module states at the target coordinates is computed using the method described above. The density at each state is then scaled by an obstacle density factor representing the density of obstacles located around the current module state and a region up to a distance of two additional modules away. For one module state, the searched area is shown in Figure 4. Three sectors denoted by the letters A, B, and C represent three areas of high, medium, and low concern respectively. The area A of high concern is located at the current module state whereas the areas of low concern B, and C are located at along the current module axis at a distance of one and two additional modules.

Ten points along the length of each module are selected in the obstacle avoidance algorithm. The number of points selected reflects ratio between the length of the module (the width is assumed to be negligeable) and the size of the discretization of the workspace. The location of these points on the obstacle density map determines the values $c_{i,j}$ of the cells in obstacle density map. The obstacle strength k_o^S at the current module state is obtained from the sum of the cells $c_{i,j}$ of each sector.

$$k_o^S = p_1 \sum_A c_{i,j}^A + p_2 \sum_B c_{i,j}^B + p_3 \sum_C c_{i,j}^C \tag{9}$$

The values of the weights p_i are determined such that effect of obstacles diminish as the distance from the current module increases. A large magnitude of p_1 ensures that the inverse of the obstacle strength, given by the obstacle density factor K_o^S, is reduced to zero in the presence of obstacles near the current module state region A (as seen in Figure 4).

$$K_o^S = \begin{cases} 1/k_o^S & \text{if } k_o^S \neq 0 \\ 1 & \text{if } k_o^S = 0 \end{cases} \tag{10}$$

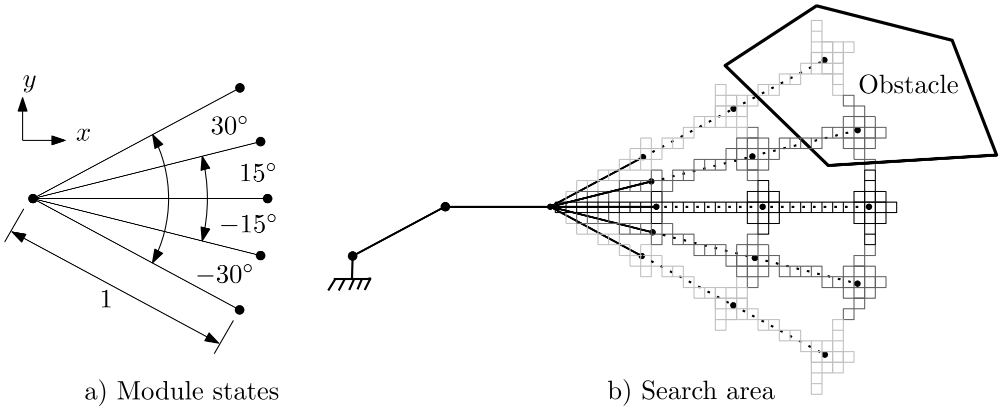
The density value obtained from the breadth-first search inverse kinematics algorithm is then scaled by the obstacle density factor. For each state of the current module, and the configuration of the ith module is selected from

$$\max\{K_o^S \cdot (\rho_{i,P}((H_1^C H_2^C \ldots H_i^S)^{-1} H_d)\} \tag{11}$$

This process is repeated for each module in the manipulator until the end-effector module is reached.

5 Numerical simulations

Numerical simulations were performed on a planar hyper-redundant manipulator with seventeen unit-length modules. Each discretely-actuated module is capable of reaching five equally spaced states, as shown in Figure 5a. In Figure 5b, the search patterns corresponding to each of the five states are superimposed to illustrate the searched area of one module. The discretization of the manipulator workspace density functions and the obstacle density map were chosen to be identical and equal to $\Delta x = 0.1$ and $\Delta y = 0.1$ of the module length to facilitate the identification of the active cells in the obstacle density map, and $\Delta\theta = 6°$ was used to obtain a good angular resolution for the module states. It is assumed that the range sensors are strategically located on the manipulator structure such that the obstacle density map can be generated with reasonable



a) Module states

b) Search area

Figure 5: Active cell search of one module

certainty. In this paper, the obstacle density map is generated by a Matlab and used directly with the inverse kinematics algorithm. The obstacles consist of uniformly-spaced circles with a radius equal to the length of one module, as shown in Figure 6. The performance of the proposed method was evaluated by computing the inverse kinematics of over eight thousand target coordinates, bounded by

$$0 \leq x \leq 18 \quad 0 \leq y \leq 12 \quad -\pi \leq \theta \leq \pi$$

Since the states of the discretely-actuated revolute joint of Figure 5 are symmetric about the x-axis, the manipulator forward kinematics are also symmetric about the x-axis. As a result, the target coordinates that lie on the negative y-axis were eliminated to limit the redundancy of the data. The computation time, mean end-effector error, and percent convergence were recorded for each simulation. The mean end-effector error is given by:

$$e = \sqrt{(x_d - x)^2 + (y_d - y)^2 + L^2(\theta_d - \theta)^2} \tag{12}$$

where $[x_d, y_d, \theta_d]$ is the target end-effector coordinate and L is a length scale used to homogenize the positional and orientational units [23]. In the actual implementation, the value $L^2 = 0.1$ was used. Other quantities of L may be used but this value gave a good balance between position and orientation errors. The magnitude of the end-effector error is given in terms of the unit length of one modules. The values of the weights in (9) were set to $(p_1, p_2, p_3) = (10^{10}, 10^1, 10^0)$. The high

Table 1: Numerical simulation results

Algorithm	Convergence [%]	Mean computation time [s]	Mean end-effector error
ABC	32.1	0.0936	0.167
AB	32.8	0.0933	0.168
A	33.8	0.0928	0.182
Breadth-first search in free-space	39.0	0.0823	0.088

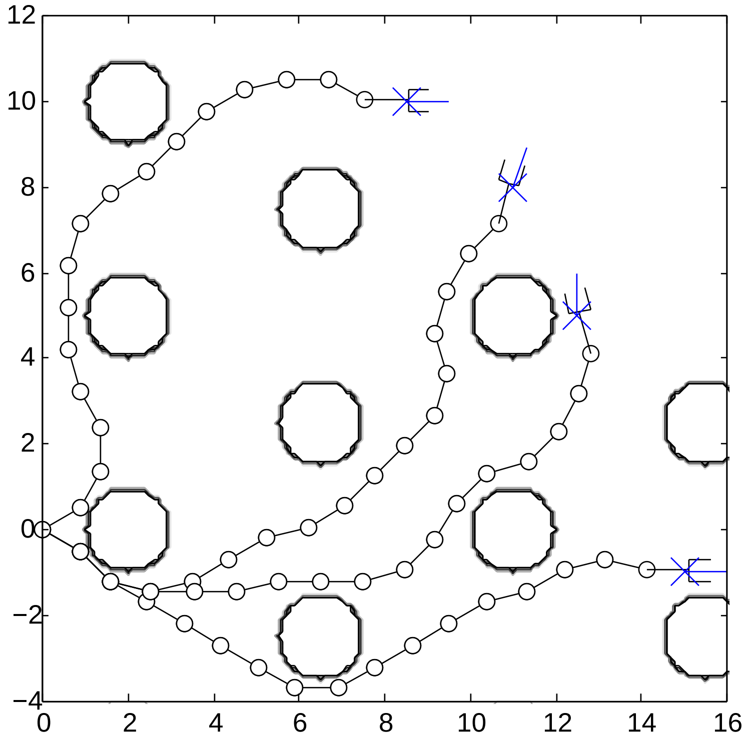


Figure 6: Manipulator configurations in an obstacle field

value of p_1 ensures that any obstacles at the current module state location would reduce the total density (11) to zero.

Three variations on the above algorithm were tested using Matlab on a Pentium IV 3.00GHz PC, and the results are given in Table 1. The algorithms A, AB, and ABC represent the active cells used to perform the obstacle avoidance. For example, in algorithm A, only the cells of region A in Figure 4 are searched for possible obstacles. As a comparison, the last row of Table 1 represents the results obtained from the breadth-first search algorithm operating in an obstacle-free workspace using the same target end-effector coordinates. It is to be noted that, although the convergence of the three obstacle avoidance algorithms is less than breadth-first search method, many end-effector target coordinates located in the vicinity of obstacles were unreachable due to the required target orientations. The additional computation time required to evaluate the active cells and the obstacle density factor is approximately 10ms, and although the mean end-effector error is higher for the obstacle avoidance, it remains a fraction of the manipulator length. There is a slight difference between the capabilities of the three obstacle avoidance algorithm variations. Although the cause of these variations remains to be determined, the actual difference in performance may become more evident as a greater number of obstacle scenarios are tested. Figure 6 illustrates four manipulator configurations within the obstacle field used in the simulations.

6 CONCLUSIONS

This paper presents an obstacle avoidance strategy for hyper-redundant manipulators using workspace density functions. Simulations on an obstacle-laden workspace demonstrated that the proposed method could achieve results similar to the original breadth-first search method. It is important to note that the sequential nature of the proposed algorithm could potentially lead to problems in certain types of scenarios. These problems would arise from the fact that the base sections must commit to particular configurations without any knowledge on the nature of the obstacles in the end-effector region. However, the large redundancy of these types of devices could be used suggest alternate module configurations in the event of a collision.

REFERENCES

[1] Chiaverini, S., Siciliano, B., and Egeland, O., Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator, IEEE Transactions on Control Systems Technology, **2**(2):123-134, 1994.

[2] Baker, D. R., and Wampler II ,C. W., On the inverse kinematics of redundant manipulators, International Journal of Robotic Research, **7**(3):2-21, 1988.

[3] Deo, S., and Walker, I. D., Minimum effort inverse kinematics for redundant manipulators, IEEE Transactions on Robotics and Automation, **13**(5):767-775, 1995.

[4] Nakamura, Y., Hanafusa, H., and Yoshikawa, T., Task-priority based redundancy control of robot manipulators, The International Journal of Robotics Research, **6**(2):3-9, 1987.

[5] Maciekewski, A. A., and Klein, C. A., Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments, The International Journal of Robotics Research, **4**:109-117, 1985.

[6] Sciavicco, L., and Siciliano, B., A solution algorithm to the inverse kinematic problem for redundant manipulators, IEEE Journal of Robotics and Automation, **4**:403-410, 1988.

[7] Borenstein, J., and Koren, Y., Real-time obstacle avoidance for fast mobile robots in cluttered environments, IEEE Journal of Robotics and Automation, **7**(3):278-588, 1991.

[8] Ge, S. S., and Cui, Y. J., New potential functions for mobile robot path planning, IEEE Transactions on Robotics and Automation, **16**:615-620, 2000.

[9] Khatib, O. Y., Real-time obstacle avoidance for manipulators and mobile robots, The International Journal of Robotics Research, **5**(1):90-98, 1986.

[10] Wang, C.-C., Kumar, V., and Chiu, G.-M., A motion control and obstacle avoidance algorithm for hyper-redundant manipulators, Proceedings of the 1998 International Symposium on Underwater Technology, 466-471, 1998.

[11] Agirrebietia, J., Avilés, R., de Bustos, I.F., and Ajuria, G., A method for the study of position in highly redundant multibody systems in environments with obstacles, IEEE Transactions on Robotics and Automation, **18**(2):257-262, 2002.

[12] Chirikjian, G. S., and Burdick, J. W., A modal approach to hyper-redundant manipulator kinematics, IEEE Transactions on Robotics and Automation, **10**(3):343-352, 1994.

[13] Zanganeh, K. E., and Angeles, J., The inverse kinematics of hyper-redundant manipulators using splines, IEEE International Conference on Robotics and Automation, **3**:2797-2802, 1995.

[14] Chirikjian, G. S., and Burdick J. W., An obstacle avoidance algorithm for hyper-redundant manipulators, IEEE International Conference on Robotics and Automation, **1**:625-631, 1990.

[15] Mao, Z., and Hsia, T.C., Obstacle avoidance inverse kinematics solution of redundant robots by neural networks, Robotica, **15**:3-10, 1997.

[16] Cheng, F.-T., Lu Y.-T., and Sun Y.-Y., Window-shaped obstacle avoidance for a redundant manipulator, IEEE Transactions on Systems, Man, and Cybernetucs, Part B, **28**:806-815, 1998.

[17] Glass, K., Colbaugh, R., Lim, D., and Seraji, H., Real-time collision avoidance for redundant manipulators, IEEE Transactions on Robotics and Automation, **11**:448-457, 1995.

[18] Chirikjian, G. S., and Ebert-Uphoff, I., Numerical convolution on the Euclidian group with applications to workspace generation, IEEE Transactions on Robotics and Automation, **14**(1):123-136, 1998.

[19] Ebert-Uphoff, I., and Chirikjian, G. S., Inverse kinematics of discretely actuated hyper-redundant manipulators using workspace densities, IEEE International Conference on Robotics and Automation, 139-145, 1996.

[20] Kyatkin, A. B., and Chirikjian, G. S., Synthesis of binary manipulators using the Fourier transform on the Euclidean Group, Journal of Mechanical Design, **121**(1):9-14, 1999.

[21] Chirikjian, G. S., and Kyatkin, A. B., Engineering applications of noncommutative harmonic analysis, CRC Press, ch. 12, 2000.

[22] Wang, Y., and Chirikjiam, G. S., Workspace generation of hyper-redundant manipulators as a diffusion process on SE(N), IEEE Transactions on Robotics and Automation, **20**(3):399-408, 2004.

[23] Suthakorn, J., and Chirikjian, G. S., A new inverse kinematics algorithm for binary manipulators with many actuators, Advanced Robotics, **5**(2):225-244, 2001.

[24] Borenstein, J., and Koren, Y., The vector field histogram-fast obstacle avoidance for mobile robots, IEEE Transactions on Robotics and Automation, **7**(3):278-288, 1991.

[25] Ulrich, I., and Borenstein, J., Y., VFH+: reliable obstacle avoidance for fast mobile robots, IEEE International Conference on Robotics and Automation, **2**:1572-1577, 1998.

[26] Moravec, H., and Elfes, A. , High resolution maps from wide angle sonar, IEEE International Conference on Robotics and Automation, **2**:116-121, 1985.