Proceedings of the 1996 IEEE
International Conference on Robotics and Automation
Minneapolis, Minnesota - April 1996

Local Optimization of Redundant Manipulator Kinematics within Constrained Workspaces

Joseph Wunderlich and Charles Boncelet
Department of Electrical Engineering, The University of Delaware, Newark, DE

***Abstract-* Finding optimal kinematics for manipulators dedicated to working within enclosed workspaces can be applied to assembly-line tasks such as welding, grinding, and spray-painting within automobile interiors. This paper presents a method of designing redundant and hyper-redundant manipulators for enclosed workspaces. A heuristic search is used to generate designs capable of maneuvering within a specified *target-workspace* in a given enclosure. A locally-optimal design is then selected for minimum degrees of freedom, or for performance values measured during the search. A variation on task-priority based redundancy control is also presented which allows many secondary-priority obstacle avoidance tasks to be satisfied simultaneously, therefore permitting easier maneuvering through complex enclosures.**

I. Introduction

A redundant manipulator has more *degrees of freedom* (DOF) than are needed to position its end-effector anywhere within the workspace defined by its joint limits. This allows the end-effector location to remain unchanged while the rest of the arm is free to move. These extra degrees of freedom increase the ability of the manipulator to avoid obstacles (or singularities) by broadening the range of joint-angle trajectories that satisfy a specified end-effector motion. As the complexity of the workspace is increased, higher degrees of freedom are warranted until the appropriate manipulator for the task can maneuver like an elephant trunk. This type of manipulator is referred to as *Hyper-redundant.*

Many researchers have explored the use of fixed link length, fixed DOF manipulators in constrained spaces [1]-[7]. Research has also been conducted on optimizing the link lengths of non-redundant manipulators in free space [8]-[10]. The research presented here involves optimizing the link lengths and DOF of redundant manipulators in constrained workspaces.

In this paper we study two-dimensional (planar) manipulators; not only as a precursor to designing three-dimensional manipulators, but as a means of finding useful planer designs.

A 2-DOF planar manipulator is shown in Fig. 1.

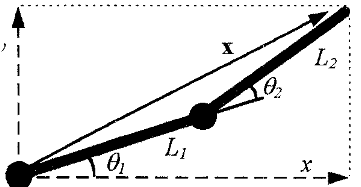


Fig. 1. 2-DOF planar manipulator.

The end-effector velocity $\dot{\mathbf{x}}$ is related to the joint velocity $\dot{\theta}$ by

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\theta} \qquad (1)$$

where **J** is the Jacobian matrix. **J** for a 2-DOF planar manipulator, if full rank, is represented by the linear mapping of Fig. 2(a). A 2-DOF manipulator becomes singular when fully extended (θ_2 = 180) or fully contracted ($\theta_2 = 0$); where **J** is reduced in rank and can no longer be inverted. Therefore, a least squares

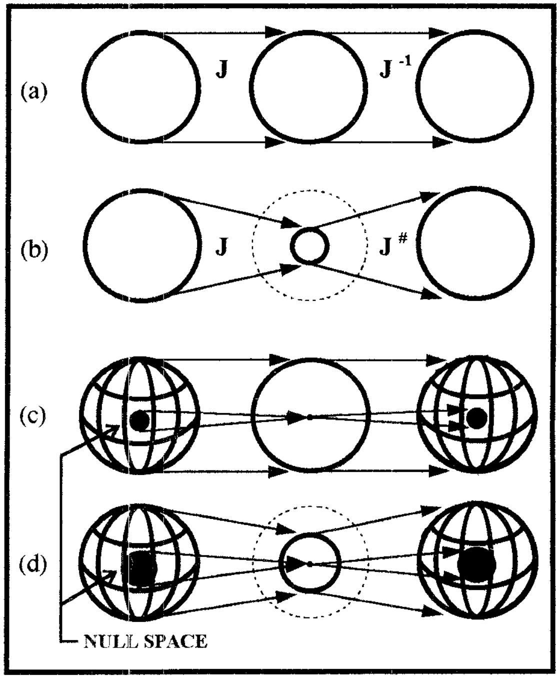


Fig. 2. $\dot{\theta}$ to $\dot{\mathbf{x}}$ to $\dot{\theta}$ Planar Mappings: (a)2-DOF away from singularities. (b)2-DOF at a singularity. (c)3-DOF away from singularities. (d)3-DOF at a singularity.

0-7803-2988-4/96 $4.00 © 1996 IEEE

approximate inverse, the *pseudoinverse* ($\mathbf{J}^{\#}$), is needed to map $\dot{\mathbf{x}}$ to $\dot{\theta}$ [11], [12]. Fig. 2(b) shows this mapping.

By adding one link to a 2-DOF planer manipulator, the manipulator becomes redundant. As links are added, the end-effector position $\mathbf{x}$ is related to joint-angles θ_1, θ_2,...θ_n and link lengths L_1, L_2,...L_n by

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} L_i * \left(\cos\left(\sum_{j=1}^{i} \theta_j\right)\right) \\ \sum_{i=1}^{n} L_i * \left(\sin\left(\sum_{j=1}^{i} \theta_j\right)\right) \end{bmatrix} \tag{2}$$

and (1) becomes:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \partial x/\partial\theta_1 & \partial x/\partial\theta_2 \ldots & \partial x/\partial\theta_n \\ \partial y/\partial\theta_1 & \partial y/\partial\theta_2 \ldots & \partial y/\partial\theta_n \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_n \end{bmatrix} \tag{3}$$

The general form of the least-squares solution to (3) is

$$\dot{\theta} = \mathbf{J}^{\#}\dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}^{\#}\mathbf{J})\dot{\Psi} \tag{4}$$

where $\mathbf{I}$ is an identity matrix, $\dot{\Psi}$ is an arbitrary joint velocity vector, and $(\mathbf{I} - \mathbf{J}^{\#}\mathbf{J})\dot{\Psi}$ is the projection of $\dot{\Psi}$ onto the null space of $\mathbf{J}$ [13]. Here, the Pseudoinverse is

$$\mathbf{J}^{\#} = \mathbf{J}^{T}(\mathbf{J}\mathbf{J}^{T})^{-1} \tag{5}$$

since (m < n) and $\mathbf{J}$ is assumed to be of rank m.

Equation (4) represents the least-squares solution which minimizes the error norm:

$$\min\left\|\dot{\mathbf{x}} - \mathbf{J}\dot{\theta}\right\| \tag{6}$$

where $\| \;\; \|$ denotes the Euclidean norm [14].

The first term of (4) represents the minimum norm solution among all the solutions provided by (4) by also satisfying

$$\min\left\|\dot{\theta}\right\| \tag{7}$$

The relation between $\dot{\theta}$ and $\dot{\mathbf{x}}$ for a 3-DOF planar manipulator is shown in Fig. 2(c) and (d). The null spaces shown correspond to the set of $\dot{\Psi}$ joint velocities which result in no change in $\dot{\mathbf{x}}$.

II. Simulation

In this paper we investigate the design of manipulators dedicated to performing tasks within enclosures. The enclosures are defined by combining *primitives* as shown in Fig. 3; the dotted lines indicate *repelling-areas* where the manipulator is repelled at either 0, 45, or 90 degrees from the walls. Repelling-areas are also located above and below the enclosure's opening to help guide it into the enclosure.

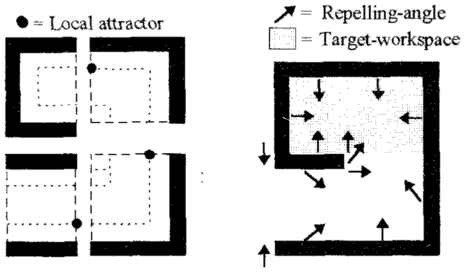


Fig. 3. Example enclosure, target-workspace, and repelling angles.

The end-effector is drawn toward a *local-attractor* in each primitive unless a task has been specified within it. Repelling-areas, repelling-angles, and local-attractor locations are defined when the primitives are assembled, and the *target-workspace* specified. Repelling angles can be changed to assist the manipulator in exiting the enclosure upon completion of target-workspace tasks. The repelling-velocity magnitudes must be specified, but can be easily found using repeated trial trajectories.

A task-priority based redundancy control technique is used here which is a variation of the following derived in [5] from equation (4):

$$\dot{\theta} = \mathbf{J}_{e}^{\#}\dot{\mathbf{x}}_{e} + [\mathbf{J}_{o}(\mathbf{I} - \mathbf{J}_{e}^{\#}\mathbf{J}_{e})]^{\#}(\dot{\mathbf{x}}_{o} - \mathbf{J}_{o}\mathbf{J}_{e}^{\#}\dot{\mathbf{x}}_{e}) \tag{8}$$

where (e) designates end-effector and (o) the point on the manipulator closest to an obstacle (*obstacle-avoidance-point*). If *enough* redundancy is available, a secondary-priority task of commanding the repelling velocity $\dot{\mathbf{x}}_{o}$ of an obstacle-avoidance-point can be accomplished while not disrupting the first-priority task of moving the end-effector along a fixed trajectory. Repelling is always directly away from an obstacle.

The research presented here allows many obstacle-avoidance-points, and governs joint-angle velocities by

$$\dot{\theta} = \mathbf{J}_{e}^{\#}\dot{\mathbf{x}}_{e} + \sum_{i=1}^{N}\left[[\mathbf{J}_{o_i}(\mathbf{I} - \mathbf{J}_{e}^{\#}\mathbf{J}_{e})]^{\#}(\dot{\mathbf{x}}_{o_i} - \mathbf{J}_{o_i}\mathbf{J}_{e}^{\#}\dot{\mathbf{x}}_{e})\right] \tag{9}$$

where N is the number of obstacle-avoidance-points; one fixed at each of the manipulator's joints (with the exception of the first two since they remain outside the enclosure). Up to eight obstacle-avoidance-points have been successfully simulated (i.e., on a 10-DOF manipulator). Here, there is no need to track the location of the obstacle-avoidance-point(s) on the manipulator since they are fixed.

In [5], the use of multiple secondary-priority tasks is suggested for avoiding multiple obstacles by splitting the second term of (8) and scaling the new terms by relative distances to obstacles. In (9), the summation terms are equally weighted which reduces the ability of the manipulator to perform a fixed end-effector trajectory when many obstacles are encountered simultaneously. However, this is offset by the efficient use of available redundancy in each summation term of (9) by repelling obstacle-avoidance-points as a function of the desired end-effector trajectory (i.e., by selection of $\dot{\mathbf{x}}_{o_i}$). The objective here is to find a compromise between accomplishing each obstacle avoidance task, and minimizing the norm of each $(\dot{\mathbf{x}}_{o_i} - \mathbf{J}_{o_i}\mathbf{J}_{\mathrm{e}}^{\#}\dot{\mathbf{x}}_{\mathrm{e}})$ term in (9), where $\mathbf{J}_{o_i}\mathbf{J}_{\mathrm{e}}^{\#}\dot{\mathbf{x}}_{\mathrm{e}}$ represents the motion at the obstacle-avoidance-point to satisfy the end-effector velocity constraint. Therefore, a joint-angle velocity solution closer to the $\mathbf{J}_{\mathrm{e}}^{\#}\dot{\mathbf{x}}_{\mathrm{e}}$ minimum norm solution is possible. This technique is also made feasible by repelling only at close proximity to the enclosure's walls and at a magnitude inversely related to the manipulator's distance from the wall (similar to the method in [5]).

Since the end-effector is guided by local-attractors, its trajectory is allowed to vary when the available redundancy becomes diminished. Approximate fixed-trajectory tasks are performed by disabling local-attractors, and substituting closely spaced *sub-goals* along the desired path.

Once the enclosure and target-workspace have been specified, an initial guess is made of manipulator kinematics, and an initial configuration is specified which allows the manipulator to easily feed into the enclosure during the initial simulation run and subsequent simulation runs where link lengths are changed. Fig. 4(a) shows an example initial guess. Here, the link-lengths and DOF have been selected to allow the manipulator to reach the furthest point within the enclosure. Fig. 4(b) shows a specified initial configuration.

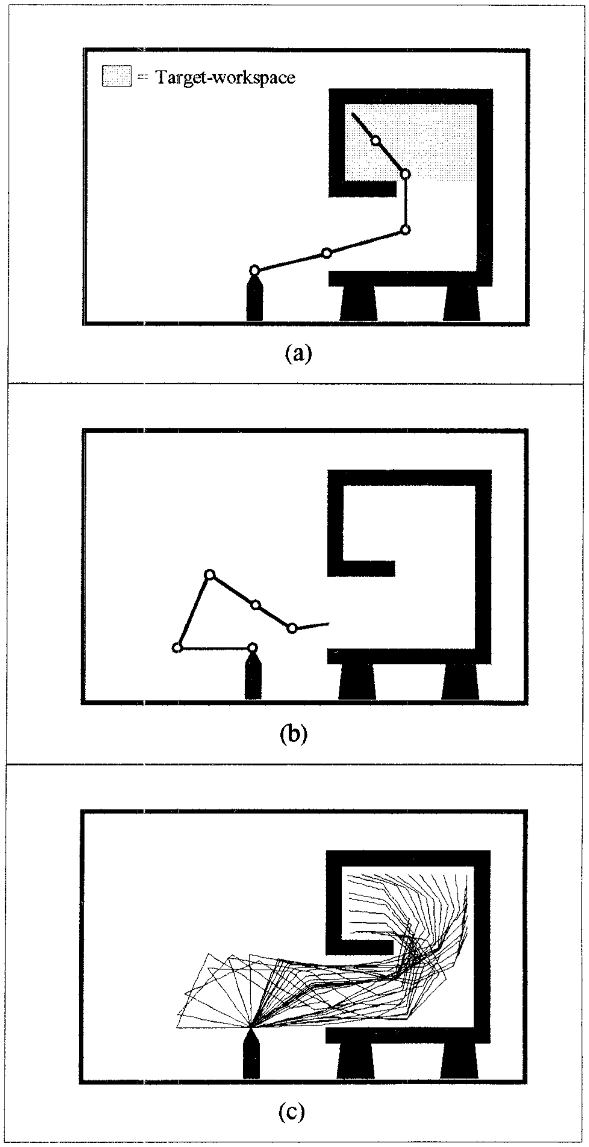
All designs are tested on their ability to maneuver within the target-workspace by commanding the manipulator to follow a path along the walls of the target-workspace after it has maneuvered through the enclosure to the target-workspace boundary. This is shown in Fig. 4(c). The repelling-velocity magnitudes have been found here using repeated trial trajectories.

A. Heuristic Search

Once an initial design is specified, a search is undertaken to find new designs by repeatedly changing link lengths and testing each manipulator within the target-workspace.

The search for combinations of link lengths can be computationally expensive if an exhaustive search is performed. For example, if the (100,110,75,60,50)cm link lengths of the design shown in Fig. 4 are each allowed to vary by 10cm from 0 to 150cm, 15^5=760,000 different designs must be tested. If the number of floating-point operations used by the (100,110,75,60,50)cm design (i.e., $5*10^6$ flops without graphics) is used as an average, the search could require $4*10^{12}$ flops. This search space has been reduced here by:

1) Only considering link length changes such that the total manipulator length remains constant.
2) Changing link lengths by discrete amounts
3) Implementing a heuristic search of *most-likely-successful* changes to the initial design.



(a)

(b)

(c)

Fig. 4. (a)Initial guess of link lengths and degrees of freedom [i.e.,100, 110, 75,60, and 50 cm]. (b)Specified initial configuration. (c)Design reaching all of the specified target-workspace.

At each level of the heuristic search the simulation adds and subtracts a fixed amount to the links such that the total manipulator length remains constant. For a 5-DOF initial design, example variations are: (L1+change, L2+change, L3-change, L4-change, L5) or (L1+change, L2-change, L3, L4, L5) where L(i) is one of the link lengths of the initial design. There are 50 of these variations for a 5-DOF design. Therefore, 50 variations are tested at the first level of the search; each tested on its ability to maneuver within the target-workspace. The resulting successful designs and *minor-hit* designs are then each subjected to the same heuristic to produce 50 more successors each. "Minor-hit" designs are those which can follow the walls of the target-workspace, but partially hit the enclosure between obstacle-avoidance-points. Although these designs are not considered successful, they often lead to successful designs in the search. Successors created which are identical to any previously generated design are eliminated (before testing). This process is continued until no more unique successors are generated.

The rationale behind this heuristic is that the most likely successful designs are those *close* to the initial design, which has been proven to work before executing the full search. Also, since the initial design does not need to be successful for the search to execute, the first level of the search can provide a means of finding an initial design.

The search space can be further reduced by only changing the lengths of two links each iteration (e.g., creating only 20 variations for a 5-DOF initial design instead of 50).

If no combination of link lengths is found to provide a successful trajectory, an extra link can be added, and the search performed again. In this manner a hyper-redundant manipulator may result. An excess of links can be determined by setting the minimum link length to zero in the search, therefore minimizing DOF.

Although the heuristic search does not guarantee an optimal design, it does find a set of designs proven to work within the target-workspace.

B. Performance criteria

Once successful designs have been generated, a final design can be selected for minimum DOF, or by comparing performance criteria measured as each design was tested. The following cost function contains these measures and can be used to select the final design:

$$COST = k_D\left(\frac{D}{D_{MAX}}\right) + k_S\left(\frac{S}{S_{MAX}}\right) + k_R\left(\frac{R}{R_{MAX}}\right) - k_w\left(\frac{\overline{w}}{\overline{w}_{MAX}}\right) \tag{10}$$

where D, S, R, and $\overline{w}$ are measures of DOF, simulation steps (simulated-time), joint-angle displacement, and dexterity; k_D, k_S, k_R, and k_w are constants which determine the relative weight of each measure. Joint-angle displacement is measured as

$$R = \int_{t_0}^{t_2}\left(\sum_{i=1}^{DOF}\left|\Delta\theta_i(t)\right|\right)dt \tag{11}$$

where $\Delta\theta_i(t)$ is the change in θ_i during a simulation step. R is measured over the entire trajectory (t_0 to t_2).

The dexterity measure used here is the *measure of manipulability* [15]:

$$w = \sqrt{\det(\mathbf{JJ}^T)} \tag{12}$$

which gives some indication of how far a manipulator configuration is from being singular. Manipulability is measured here over the time the manipulator enters the target-workspace (t_1) until the target-workspace task is completed (t_2) and is expressed here as the *average-manipulability*:

$$\overline{w} = \left[\frac{\int_{t_1}^{t_2}\sqrt{\det(\mathbf{JJ}^T)}dt}{(t_2 - t_1)}\right] \tag{13}$$

III. Simulation Results

In the example below, the "reduced" heuristic search (i.e., 20 variations per design) was applied to the (100,110,75,60,50)cm design of Fig. 4. The link lengths where allowed to vary by 10cm from 0 to infinity. For this example, 2,600 different designs were tested over the 10-level tree search shown in Fig. 5. This figure shows all of the 31 successful designs, and the 9 (of 174) minor-hit designs which lead to successful designs. Table 1. summarizes the results.

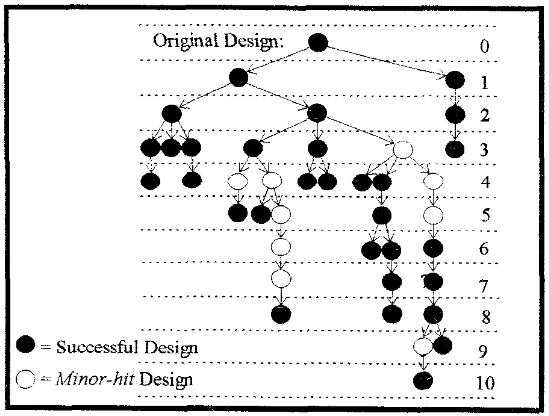


Fig. 5. Resulting tree from example heuristic search.

Table 1. Successful Designs

LINK LENGTHS(cm)									
L1	L1	L3	L4	L5	STEPS	*R*	$\overline{w}$	DOF	#
100	110	75	60	50	105	13.27	41920	5	1
100	110	85	50	50	104	13.50	40360	5	2
100	110	85	60	40	107	13.30	39990	5	3
110	110	85	50	40	107	13.85	38670	5	4
100	110	95	40	50	106	13.08	39800	5	5
100	120	85	60	30	108	14.05	37290	5	6
110	100	95	50	40	108	14.27	38190	5	7
110	110	95	40	40	109	14.02	37460	5	8
110	110	95	50	30	108	13.97	36680	5	9
90	120	95	40	50	104	12.10	39450	5	10
100	120	95	40	40	106	12.93	37920	5	11
90	120	95	60	30	106	13.42	38270	5	12
120	90	95	50	40	110	14.17	38400	5	13
110	110	95	60	20	108	14.24	36490	5	14
100	120	105	30	40	107	13.38	37080	5	15
100	120	105	40	30	106	13.42	35820	5	16
110	110	105	20	50	108	13.68	37930	5	17
100	120	105	20	50	108	13.02	38040	5	18
80	140	85	40	50	103	14.29	39180	5	19
90	130	105	20	50	105	12.37	38090	5	20
100	120	105	10	60	108	12.83	39400	5	21
100	130	105	**0**	60	125	14.22	38280	4	22
100	110	105	10	70	107	12.62	41620	5	23
120	100	105	30	40	109	13.50	36120	5	24
110	110	105	**0**	70	112	13.17	40830	4	25
120	100	95	30	50	112	14.45	38570	5	26
90	120	75	30	80	106	13.19	44800	5	27
120	110	95	**0**	70	123	14.66	38650	4	28
120	100	85	40	50	110	13.89	38990	5	29
120	100	75	40	60	125	15.55	40610	5	30
120	90	75	40	70	112	14.48	42300	5	31

The heuristic search succeeded in producing three 4-DOF designs from the original 5-DOF design:

(#22): (100,130,105,**0**,60)cm
(#25): (110,110,105,**0**,70)cm
(#28): (120,110,95,**0**,70)cm

which demonstrates the ability of the search to minimize DOF. The (120,110,95,**0**,70)cm design is shown in Fig. 6.

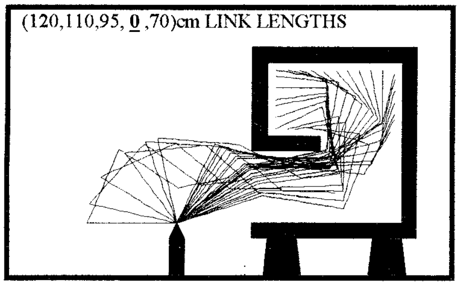


Fig. 6. A 4-DOF design resulting from the heuristic search on an initial 5-DOF design.

The (80,140,85,40,50)cm design (#19) shown in Fig. 7 was the fastest (103 simulation steps); the (90,120,95,40,50)cm design (#10) shown in Fig. 8 had the least joint-angle displacement (12.10 radians); and the (90,120,75,30,80)cm design (#27) maintained the highest average manipulability within the target-workspace (44800).

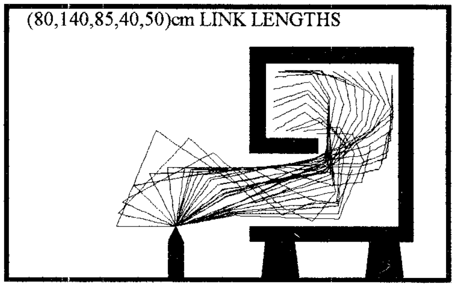


Fig. 7. Fastest design to reach all of the target-workspace.

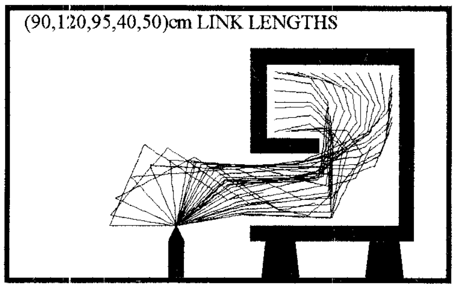


Fig. 8. Design using the least joint-angle displacement in reaching all of the target-workspace.

Using the equation (10) cost function, a final selection can be made based on the robot designer's bias towards each measure. This is accomplished through the selection of k_D, k_S, k_R, and k_w in (10). If for example each measure is considered equally significant (i.e., all $k's = 1$) in the above example, the (110,110,105,**0**,70)cm design (#25) has the smallest cost (1.63) and can be considered the overall locally-optimal design for this search. This design is shown in Fig. 9.

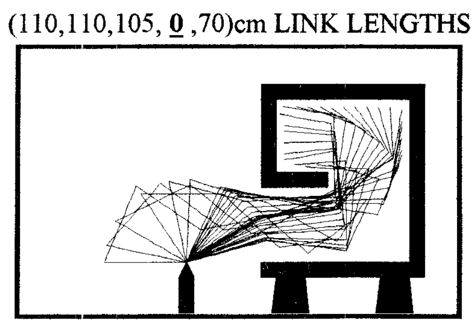


Fig. 9. **Overall locally-optimal design** if equal weighting given to DOF, simulated-time, joint-angle displacement, and average manipulability. This is a 4-DOF design.

IV. Hyper-redundant Design

The variation on task-priority based redundancy control can also be used to design hyper-redundant manipulators. An example 10-DOF hyper-redundant design is shown in Fig. 10.

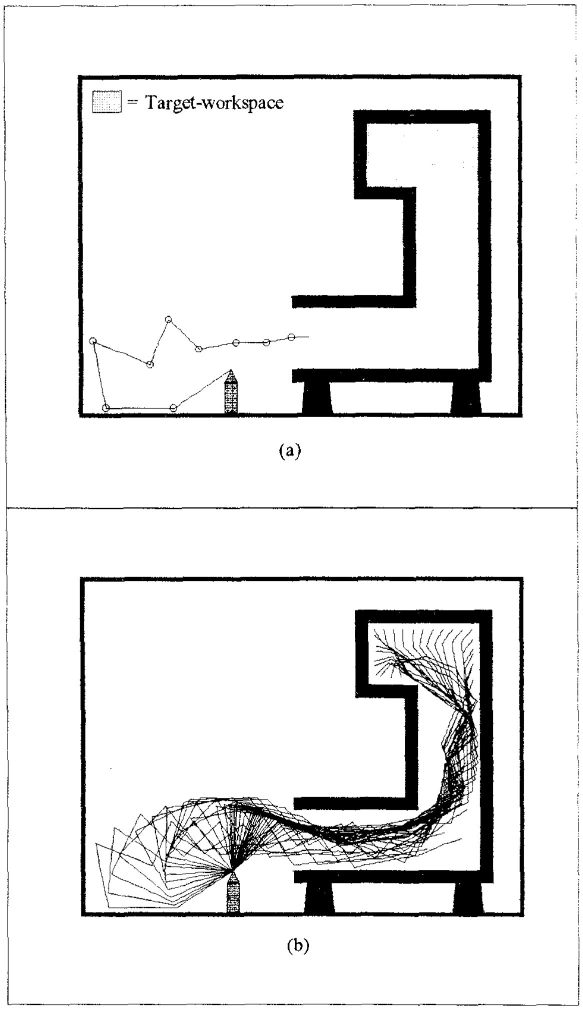


Fig. 10. A 10-DOF hyper-redundant manipulator design. (a)Initial configuration. (b)Manipulator reaching all of the specified target-workspace .

V. Conclusions

This paper presents a method of designing redundant and hyper-redundant manipulators for enclosed workspaces. A heuristic search is used to generate designs capable of maneuvering within a specified target-workspace in a given enclosure. A locally-optimal design is then selected for minimum degrees of freedom, or for performance values measured during the search. A variation on task-priority based redundancy control is also presented which allows many secondary-priority obstacle avoidance tasks to be satisfied simultaneously, therefore permitting easier maneuvering through complex enclosures. This is accomplished by compromising the desired end-effector trajectory between local attractors or sub-goals, and by repelling obstacle-avoidance-points in a direction which is a function of the desired end-effector trajectory. The accuracy in following a fixed trajectory can be improved by reducing the distance between sub-goals and by specifying a smaller simulation step size. If a high-precision fixed trajectory is required, the heuristic search can provide manipulator designs to be used with other controls schemes since all of the designs have been proven geometrically feasible of performing within the given enclosure.

References

[1] A. A. Maciejewski and J. J. Fox, "Path planning and the topology of configuration space ," *IEEE Trans. Robotics and Automation,* vol. 9, no. 4, pp. 444-456, 1993.

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

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *proc. IEEE Int. Conf. on Robotics and Automation,* 1985, pp. 500-505.

[4] G. S. Chirikjian, J. W. Burdick, "An obstacle avoidance algorithm for hyper-redundant manipulators," in *proc. IEEE Int. Conf. on Robotics and Automation,* 1990, pp. 625-631.

[5] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. Robotics Res.,* vol. 4, no. 3, pp. 109-117, 1985.

[6] R. Brooks, "Planning collision free motions for pick and place," *Int. Symp.of Robotics Res.,* Kyoto, 1984, pp. 5-37.

[7] H. Hanafusa, T. Yoshikawa, and Y. Nakamura, "Analysis and control of articulated robot arms with redundancy," in *proc. IFAC Control Science and Technology 8th Triennial World Congress,* 1981, pp. 1927-1931.

[8] C. J. J. Paredis and P. K. Khosla, "Kinematic design of serial link manipulators from task specifications ," *Int. J. Robotics Res.,* vol. 12, no. 3, pp. 274-287, 1993.

[9] Z. Shiller and S. Sundar, "Design of robotic manipulators for optimal dynamic performance," in *proc. IEEE Int. Conf. on Robotics and Automation,* 1991, pp. 344-349.

[10] R. V. Mayorga, B. Ressa, and A. K. C. Wong, "A kinematic criterion for the design optimization of robot manipulators," in *proc. IEEE Int. Conf. on Robotics and Automation,* 1991, pp. 578-583.

[11] R. Penrose, "On best approximate solutions of linear matrix equations," *Proc Cambridge Phil. Soc.,* vol. 52, pp. 17-19, 1956.

[12] H. Asada and J.-J. E. Slotine, *"Robot analysis and control."* New York: John Wiley and Sons, 1986, pp. 51-71.

[13] T. N. E. Greville, "The pseudoinverse of a rectangular or singular matrix and its applications to the solutions of systems of linear equations," *SIAM Review,* vol. 1, no. 1, pp. 38-43, 1959.

[14] Y. Nakamura, *"Advanced robotics: redundancy and automation."* Reading, MA: Addison-Wesley, 1991, pp.104-151.

[15] T. Yoshikawa, "Analysis and control of robot manipulators with redundancy," in *Robotics Research,* M. Brady and R. Paul, Eds., Cambridge, MA: MIT Press, 1984, pp. 735-747.