Potential-Based Path Planning for Robot Manipulators

Chien-Chou Lin,* Lo-Wei Kuo, and Jen-Hui Chuang
Department of Computer and Information Science
National Chiao Tung University
Hsinchu, 30010, Taiwan
Republic of China
e-mail: jeremylin@cis.nctu.edu.tw, jchuang@cis.nctu.edu.tw

Received 14 April 2003; accepted 14 January 2005

In this paper, a potential-based path-planning algorithm for a high DOF robot manipulator is proposed. Unlike some c-space-based approaches, which often require expensive preprocessing for the construction of the c-space, the proposed approach uses the workspace information directly. The approach computes, similar to that done in electrostatics, repulsive force and torque between objects in the workspace. A collision-free path of a manipulator will then be obtained by locally adjusting the manipulator configuration to search for minimum potential configurations using that force and torque. The proposed approach is efficient because these potential gradients are analytically tractable. Simulation results show that the proposed algorithm works well, in terms of computation time and collision avoidance, for manipulators up to 9 degrees of freedom (DOF).

1. INTRODUCTION

Path planning of a manipulator is to determine a collision-free trajectory from its original location and orientation (called starting configuration) to goal configuration.[1] Some planners adopt the configuration space (c-space)-based approach,[2–6] which considers both the manipulator and obstacles at the same time by identifying manipulator configurations intersecting the obstacles. A point in a c-space indicates a configuration of manipulators. A configuration is usually encoded by a set of manipulator's parameters; i.e., angles of links of manipulators. The forbidden regions in the c-space are the points which imply manipulator configurations intersecting the obstacles. Thus, path planning is reduced to the problem of planning a path from a start point to goal in free space.

*To whom all correspondence should be addressed. Present address: Department of Computer Science and Information Engineering, Shu-Te University, Kaohsiung County, 82445, Taiwan, Republic of China.

Unlike c-space-based approaches, geometric algorithms directly use spatial occupancy information of the workspace (w-space) to solve path planning problem.[7–11] Workspace-based algorithms usually extract the relevant information about the free space and use that together with the manipulator geometry to find a path. However, most of the chosen paths are not a safer one. One way to obtain the best match and minimize the risk of collision is to define a repulsive potential field between the manipulator and the obstacles.[7,10,11]

In this paper, an artificial potential field, whose magnitude is unbounded near the obstacle boundary and decreases with range,[12] is applied to model the workspace for the path planning of robot manipulators. The approach computes, similar to that done in electrostatics, repulsive force and torque between objects in the workspace. A collision-free path of a manipulator will then be obtained by locally adjusting the manipulator configuration to search for minimum potential configurations using the force and torque. The proposed approach uses one or more guide lines (GLs) as final or intermediate goals in the workspace. The GLs are line segments among obstacles in the free space, providing the manipulator a general direction to move forward. As a GL is an intermediate goal for a manipulator to reach, it also helps to establish certain motion constraints for adjusting manipulator configuration during path planning, as discussed in Section 3. In Section 4, simulation results are presented for path planning performed on a SUN workstation for manipulators in a 2-D environment. Section 5 gives some conclusions of this work.

2. REPULSION DUE TO CHARGED POLYGONAL REGION BORDERS

Polygonal description of obstacles is often used in path planning because of its simplicity. It is shown in ref. 7 that the repulsion, in forms of repulsive force and torque, between polygonal regions whose borders are uniformly charged can be derived in closed form. Planning a path among obstacles thus modeled will guarantee collision avoidance as the potential will increase indefinitely as an object approaching an obstacle. In ref. 12, it is shown that the repulsion between the charged object and obstacle boundaries due to linear and quadratic source distributions is also analytically tractable, as reviewed in this section.

For a polygonal object and obstacles in the 2-D space, line segments of their boundaries can be used as basic elements in the calculation of the repulsive force and torque needed in path planning. The repulsive force and torque between these polygonal regions can then be derived by superposing the repulsion between pairs of border segments, each containing one line segment from the moving object and the other from one of the obstacles.

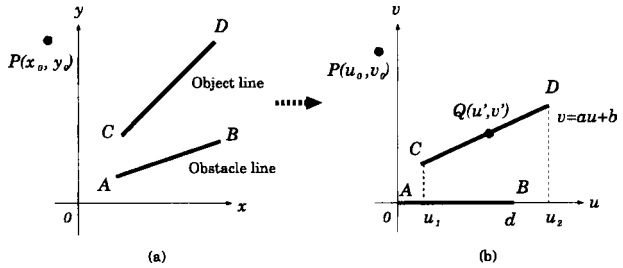

Figure 1. Coordinate transformation. (a) The original coordinate system (xy plane). (b) The new coordinate system (uv plane) after the transformation.

In general, each pair of the repelling line segments can have an arbitrary configuration in the work space as shown in Figure 1(a). To simplify the expressions of the repulsion between them in the following subsections, a coordinate system is chosen so that the obstacle line segment, $\overline{AB}$, lies on the base line, as shown in Figure 1(b). In the new coordinate system (uv plane), the coordinates of the endpoints of $\overline{AB}$ are assumed to be (0,0) and ($d>0$,0), respectively, and the line containing the object line segment, $\overline{CD}$, can be represented as $v=au+b$, $u_1 \leq u \leq u_2$.

2.1. The Integral Equations for Forces

Consider the electric field at point $Q=(u',v')$ of $\overline{CD}$ due to a point $(u,0)$ on $\overline{AB}$ shown in Figure 1(b). We have

$$\vec{E} \triangleq (E_u, E_v) = -\nabla\left[\frac{1}{r}\right] = -\frac{1}{r^2}\hat{r} \tag{1}$$

where $\vec{r}=(u'-u,v')$, $r=|\vec{r}|=\sqrt{(u'-u)^2+v'^2}$, and $\hat{r}=\vec{r}/r$.

Thus, the total force at point Q due to $\overline{AB}$ can be decomposed into two parts, i.e.,

$$F_u(u',v') = \int_{q_1}^{q_2} E_u\, dq = \int_0^d \frac{u'-u}{r^3}\rho(u)du, \tag{2}$$

$$F_v(u',v') = \int_{q_1}^{q_2} E_v \, dq = \int_0^d \frac{au'+b}{r^3} \rho(u) du, \quad (3)$$

where $\rho(u)$ is the charge density along $\overline{AB}$.

For the total force on the object line segment $\overline{CD}$, we have[1]

$$F_u = \int_{q_1}^{q_2} F_u(u',v') dq = \int_{s_1}^{s_2} F_u(u',v') \rho(s) ds, \quad (4)$$

where $s = \sqrt{1+a^2}u'$, $\rho(s)$ is the charge density along $\overline{CD}$, and $dq = \rho(s)ds = \sqrt{1+a^2}\rho(u')du'$. Thus, the above integral equations can be formulated as

$$F_u = \sqrt{1+a^2} \int_{u_1}^{u_2} \int_0^d \frac{u'-u}{r^3} \rho(u)\rho(u') du \, du'. \quad (5)$$

2.2. The Integral Equations for Torques

Given any collision-free orientation of an object with nonzero torque with respect to its rotation center, the direction of the torque directly gives the direction in which the object can rotate to reach a configuration of smaller potential. In this subsection, we will consider the formulation of the repulsive torque between two charged line segments. Consider $\overline{AB}$ and $\overline{CD}$ shown in Figure 1(b) and let $P = (u_0, v_0)$ be a reference point, e.g., the rotation center of the object. The torque with respect to P, due to the repulsive force from $\overline{AB}$ on point Q, is equal to

$$\begin{aligned} \tau_P(u',v')\vec{i}_z &= \vec{l}(u',v') \times \vec{F}(u',v') \\ &= (u'-u_0, v'-v_0) \\ &\quad \times (F_u(u',v'), F_v(u',v')), \end{aligned} \quad (6)$$

where $\vec{i}_z = \vec{i}_u \times \vec{i}_v$, $\vec{l}(u',v') = \overrightarrow{PQ} = (u'-u_0, v'-v_0)$ and $\vec{F}(u',v') = (F_u(u',v'), F_v(u',v'))$. Thus, the total torque with respect to P due to the repulsion between the two line segments becomes

$$\begin{aligned} \tau_P &= \int_{s_1}^{s_2} \tau_P(u',v') \rho(s) ds \\ &= \sqrt{1+a^2} \left(\int_{u_1}^{u_2} \int_0^d (u'-u_0) \frac{au'+b}{r^3} \right. \\ &\quad \times \rho(u)\rho(u') du \, du' \\ &\quad \left. - \int_{u_1}^{u_2} \int_0^d (au'+b-v_0) \frac{u'-u}{r^3} \rho(u)\rho(u') du \, du' \right). \end{aligned} \quad (7)$$

We will now show that with the above integral equations, the repulsive forces and torques between a rigid object and obstacles can be evaluated analytically for linear and quadratic charge distributions along their boundaries.

2.3. Repulsion Due to Linear and Quadratic Charge Distributions

For the application of the proposed potential model in achieving collision avoidance in path planning, the optimal object configurations along a path can be found more efficiently if the above integral equations can be evaluated analytically instead of numerically. The main advantage of the proposed model is that such analytic expressions can be found for linear and quadratic charge densities considered in this subsection.

Assume the charge density $\rho(u)$ is equal to 1, u, or u^2 for an obstacle line, and $\rho(s) = 1$, s, or s^2 for an object line. Nine different combinations of charge distributions will need to be considered in evaluating the repulsion between the two line segments. For example, from (5), the repulsive force along the u axis for these nine combinations can be obtained from

$$\begin{aligned} F_u^{ij} &= F_u^{ij}(u_2) - F_u^{ij}(u_1) \\ &= (1+a^2)^{(j+1)/2} \int_{u_1}^{u_2} \int_0^d \frac{u'-u}{r^3} u^i \, du (u')^j \, du', \end{aligned} \quad (8)$$

where i is equal to the order of the charge density of the obstacle line, and j is equal to that of the object line. It can be shown that analytic expressions exist for all these integral equations. For example, we have

[1]For simplicity, only the u component is considered for the rest of the paper.

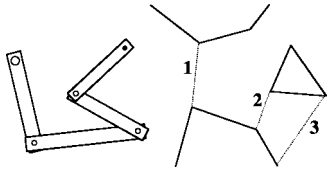

Figure 2. A manipulator is moved toward the goal (not shown) by sequentially traversing a sequence of GLs.

$$F_u^{00}(u') = \log \frac{f_1'(u')/2 + \sqrt{1+a^2} f_1^{1/2}(u')}{f_2'(u')/2 + \sqrt{1+a^2} f_1^{1/2}(u')}, \qquad (9)$$

where $f_1(u') = (au' + b)^2 + (u - d)^2$ and $f_2(u') = (au' + b)^2 + u^2$.

In general, $\rho(u)$ and $\rho(s)$ can be any quadratic functions, or

$$\rho(u) = \alpha_1 u^2 + \beta_1 u + \gamma_1, \qquad (10)$$

$$\rho(s) = \alpha_2 s^2 + \beta_2 s + \gamma_2, \qquad (11)$$

where coefficients α_1, β_1, γ_1, α_2, β_2, and γ_2 are some real numbers. The repulsive force along the u axis can still be evaluated analytically as

$$F_u = \alpha_1\alpha_2 F_u^{22} + \alpha_1\beta_2 F_u^{21} + \alpha_1\gamma_2 F_u^{20} + \beta_1\alpha_2 F_u^{12} + \beta_1\beta_2 F_u^{11} + \beta_1\gamma_2 F_u^{10} + \gamma_1\alpha_2 F_u^{02} + \gamma_1\beta_2 F_u^{01} + \gamma_1\gamma_2 F_u^{00}. \qquad (12)$$

Similar results can be obtained for F_v and τ_p. Thus, for any $\rho(u)$ and $\rho(s)$, we can first evaluate the coefficients of the charge density functions and then use the nine sets of expressions of the repulsion, as in (12), to evaluate the total repulsion.

3. PATH PLANNING

The application of the potential model reviewed in the previous section for path planning of manipulators will be discussed in this section. Unlike some c-space-based approaches, which often require expensive preprocessing to construct the c-space, the proposed approach uses the workspace information directly. The approach computes repulsive force and torque experienced by each rigid component, e.g., a link, of a manipulator. A collision-free path of the manipulator will then be obtained by locally adjusting its configuration along the path for minimum potential using these force and torque.

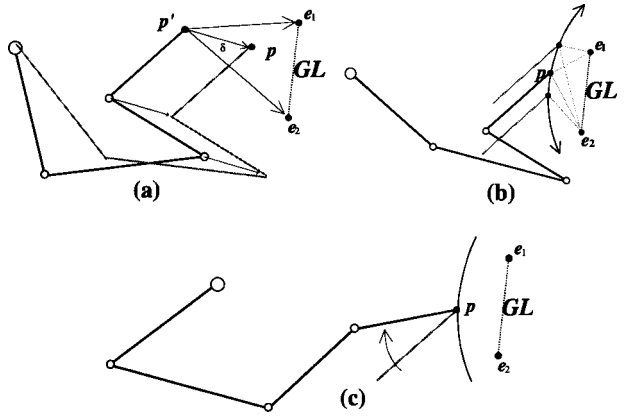

Figure 3. Basic path planning procedure for a given GL (see text).

For a rough description of object path, the proposed approach uses one or more guide lines (GLs) as final or intermediate goals in the workspace. The GLs are line segments among obstacles in the free space, providing the manipulator a general direction to move forward (see Figure 2). A collision-free traversal of a given sequence of GLs by the end-effector is regarded as a global solution of the path planning problem of a manipulator. Although free space bottlenecks defined by minimal distance links[2] (MDLs) among the obstacles (see ref. 13) are good candidates as GLs for path planning, other links among the obstacles can also be used, e.g., GL_3 shown in Figure 2. As for the amount (or density) of the GLs along a path, as long as two consecutive GLs are visible to each other, additional GLs can be inserted (interpolated) between them in a very straightforward manner (see ref. 14). It can be shown easily that the MDLs described in ref. 13 satisfy such a visibility condition.

3.1. Basic Procedure of Path Planning

In this paper, the proposed path planning approach derives a series of minimum potential configurations

[2]In this paper, it is assumed that a sequence of GLs is given in advance. Usually, the GLs can be chosen as the minimal distance links (MDLs). The MDLs can be determined easily as to connect (convex) obstacle nodes in the obstacle neighborhood graph in ref. 13. An ordered subset of those links can be used as the sequence of GLs. In ref. 13, a path containing a sequence of GLs is obtained efficiently by A* search algorithms. Therefore, the computing time of determining a sequence of GLs includes the time of connectivity graph construction and the time of searching the sequence of GLs. The graph construction algorithm requires time $O(nlogn)$, where n is the total number of vertices of obstacles. The A* search algorithm will be bounded in time $O(n^2)$, where n is the number of nodes of a graph.

along the path of a manipulator by locally adjusting its configuration for minimum potential using the results given in Section 2. Assuming that a guide line $\overline{e_1e_2}$ is given as an intermediate goal, the basic path planning procedure for moving the end-effector p of a manipulator onto $\overline{e_1e_2}$ include (see Figure 3) the following:

(i) Translate the distal links of the manipulator to move p toward the $\overline{e_1e_2}$ [Figure 3(a)].

(ii) Search for the minimum potential configuration of the manipulator for constant $|pe_1| + |pe_2|$ by repeatedly execute:

(a) Search for the minimum potential configuration of the manipulator with the distal link fixed in orientation [Figure 3(b)].
(b) Search for the minimum potential configuration of the manipulator with the end-effector p fixed in position [Figure 3(c)].

(iii) Repeat (i) and (ii) until the end-effector reaches the line segment $\overline{e_1e_2}$.

In general, there are different ways to change the manipulator configuration to move p toward $\overline{e_1e_2}$. A simple translation of distal links[3] is adopted in (i) as a preliminary implementation of our algorithm. As shown in Figure 3(a), the translation of a predetermined distance δ of the distal link is carried out to move the end-effector from p' to p. The direction of the translation is determined with respect to the two end points of the GL, e_1 and e_2, such that $\overrightarrow{p'p}$ bisects $\angle e_1p'e_2$.[4] A collision check is performed for such a translation.[5] If there is a collision, each time the distance of the translation is reduced by 50% until it is collision-free. No configuration improvement using the repulsion experienced by the manipulator is considered at this stage.

As for searching for the minimum potential configuration of the manipulator in (ii), links of the manipulator are adjusted from the distal link to the base link using repulsion. The distal link has two DOFs, i.e., its location which is constrained by $|pe_1|+|pe_2| = \text{const}$ and its orientation, while each of other distal links has one DOF, its orientation. The two base links, together, have at most two possible configurations with no freedom to fine-tune their configurations.

In (ii), the associated constrained optimization problem is divided into two iterative univariant optimization procedures, as in (ii-a) and (ii-b). In (ii-a), the distal link is fixed in its orientation [see Figure 3(b)] as p slides along the elliptic trajectory to search for the minimal potential configuration and other distal links are sequentially adjusted in orientation, staring from the link connected to the distal link. In (ii-b), the distal link is adjusted in orientation while fixed in position [see Figure 3(c)] and the procedure for adjusting the rest links is similar to that in (ii-a). Detailed implementation is presented in the next section.

For each elliptical trajectory, (ii-a) and (ii-b) are repeatedly performed until negligible changes in the manipulator configuration are obtained. Then another smaller elliptical trajectory is obtained with (i) and the process repeats. The path planning algorithm, as summarized below, ends as the end-effector reaches the given GL (and the ellipse degenerates into a line segment, the GL).

Algorithm End_Effector_to_GL

Step 0 Initialize $\delta = \delta_0$, where δ_0 is arbitrarily chosen.

Step 1 Translate the manipulator with distance δ along the direction of the bisector vector of $\overrightarrow{p'e_1}$ and $\overrightarrow{p'e_2}$. If collision occurs or if the two base links become unconnected, $\delta \leftarrow \delta/2$ and go to step 1.

Step 2 Translate the distal link to move the end-effector p along the elliptical trajectory, $|pe_1|+|pe_2| = \text{const}$, to minimize the potential.

Step 3 Joint angle adjustment for the minimum potential configuration with p fixed in position.

Step 4 Go to step 2 if the translation in step 2 or the joint angle adjustment in step 3 is not negligible.

[3]In step (i), an intermediate simple solution of the inverse kinematics problem is obtained by the translation of all manipulator links except for the two base links. For each translation, the two base links and thus the manipulator can have at most two feasible configurations for each translation. In step (ii), the problem is solved by using the potential gradient to compute a sequence of suboptimal solutions with monotonically decreasing potential until the minimal potential solution is found.

[4]Such a direction is actually the inward normal direction of the ellipse, $|pe_1|+|pe_2| = \text{const}$, at p.

[5]Whenever a link of the manipulator changes its location and/or orientation, a simplified, quadrilateral swept region is obtained for each edge of the link by using two line segments to connect the two ends of the edge, respectively. The detection of collision is then performed for that edge by checking for any intersections between the swept region and all obstacle borders.

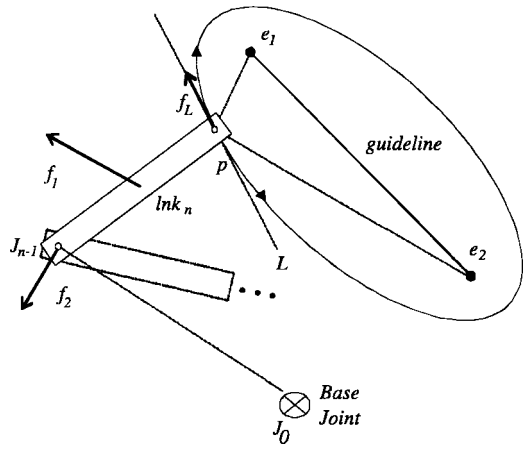

Figure 4. Sliding p along ellipse E_1, by translating lnk_n to reduce the repulsive potential.

Step 5 If p reaches GL, the planning is completed. Otherwise, go to step 1 with $\delta = \delta_0$.

For path planning involving multiple GLs, the above algorithm will be executed for each of them sequentially. It is assumed that the planning for a GL starts as the planning of the previous GL is accomplished. The path planning ends as the end-effector reaches the goal, which is usually a (goal) GL in the path planning problems considered in this paper.

3.2. Implementation Details

3.2.1. Step 1 of Algorithm End_effector_to_GL

Consider the manipulator shown in Figure 3(a). Its end-effector is initially located at point p' and is moved toward an intermediate goal GL, such that it is moved with distance δ along the direction of the bisector vector of $\overrightarrow{p'e_1}$ and $\overrightarrow{p'e_2}$, from p' to p. Such a movement is achieved in the implementation by translating every link except the two base links. It is not hard to see that the two base links have at most two possible configurations; i.e., the two base links together may have two, one, or no feasible configurations depending on the amount of translations of the distal links. In the computer implement, if there are two feasible configurations for the two base links, the one that requires less adjustments in joint angles of the two links is selected.

3.2.2. Adjusting End-Effector Position in Step 2

After the end-effector reaches the p, as shown in Figure 3(b), one of the univariant procedures, which allows the distal link to adjust its location but not its orientation in minimizing the repulsive potential, is performed in step 2 of *End_effector_to_GL*. Since the minimization is constrained by $|pe_1|+|pe_2|=\text{const}$, only the resultant force experienced by the distal link along the tangential direction of the ellipse is taken into account.

Consider the forces exerted on the distal link lnk_n, as shown in more detail in Figure 4. Let f_1 be the repulsive force exerted on lnk_n due to the repulsion between lnk_n and the obstacles, and f_2 be the force exerted on J_{n-1} due to the repulsive torque between other manipulator links and obstacles. For a univariant minimization approach, only one variable is adjusted at a time. To determine the minimum potential location of lnk_n under the elliptical constraint, all of the joint angles of the manipulator, except the base joint J_0, are assumed to be fixed. Therefore, the above repulsive torque, denoted as τ_0, is calculated for a single rigid composite link formed by all the other manipulator links with respect to J_0. Thus, we have

$$f_2 = \frac{\tau_0}{l_0}, \tag{13}$$

where l_0 is the length of $\overline{J_0 J_{n-1}}$. To determine the direction in which p should slide along the ellipse E_1, and thus lnk_n should translate, to reduce the repulsive potential, the direction of the projection of the resultant force exerted on lnk_n along the tangent of E_1 at p, L,

$$f_L = f_{1L} + f_{2L}, \tag{14}$$

is calculated. A gradient-based binary search for the minimum potential location of p along E_1 can thus be performed using (14). The search ends when the movement of lnk_n or the magnitude of f_L is negligible.

Each time the position p, and lnk_n, is changed in the above search process, the orientation of rest links, i.e., joint angles at J_0 through J_{n-1}, need to be adjusted for connectivity and for minimum potential of the manipulator. Such a procedure is very similar to the joint angle adjustment performed in step 3 of *End_effector_to_GL*, as discussed next.

3.2.3. Adjusting Joint Angle in Step 3

Once the minimum potential position of the distal link is determined with step 2 of *End_effector_to_GL*, another univariant procedure, which allows the dis-

tal link to adjust its orientation but not its location, is performed to reduce the potential further, as shown in Figure 3(c). Under the constraint that end-effector is fixed in location p, the distal link can rotate with respect to p to reduce the repulsive potential. The direction in which the distal link should rotate is determined by the repulsive torque experienced by the distal link with respect to p.

Consider the repulsion experienced by the distal link lnk_n with end-effector p fixed in position. Let τ_n be the repulsive torque experienced by lnk_n with respect to p due to the repulsion between lnk_n and let f_2, as described in the previous subsection, be the force exerted on J_{n-1} due to the repulsive torque between other manipulator links and obstacles. The resultant torque experienced by lnk_n with respect to p is equal to

$$\tau_n^* = \tau_n + f_{2\perp} \cdot l_n, \tag{15}$$

where l_n is the length of lnk_n and $f_{2\perp}$ is the projection of f_2 along the direction perpendicular to lnk_n. A gradient-based binary search for the minimum potential orientation of lnk_n for p fixed in position can thus be performed using the direction of τ_n^*. Each time the orientation of lnk_n is changed, the orientation of the rest links are adjusted iteratively for connectivity and for minimum potential using τ_{n-1}^*, τ_{n-2}^*,..., etc. The search ends when the movement of lnk_n or the torque τ_n^* is negligible.

As for the computation complexity, if every link needs k binary searches on the average to find a best orientation, the total number of binary searches needed for the derivation of the minimum potential configuration of an n-link manipulator is equal to

$$k^1 + k^2 + \cdots + k^{n-2} = (k^{n-1} - k)/(k-1), \tag{16}$$

which appears to have a fairly large value. Fortunately, much lower computation complexity is often observed, as in the examples considered in next section. Nonetheless, since path planning problems vary significantly in nature, further investigation into a more practical estimation of the number of binary searches required in the step of joint angle optimization is needed.

4. SIMULATION RESULTS

In the section, simulation results are presented for path planning performed on SUN Ultra-1 for manipulators in a 2-D environment. In order to show the path of a manipulator more clearly, every configuration of a path obtained by *End_Effector_to_GL* is shown in a different gray level; i.e., the initial configuration is shown in black and final configuration is shown in white. Moreover, all manipulator and obstacle boundaries are assumed to be uniformly charged, except for those marked out explicitly for nonuniform charge distributions. The precision for optimal configurations obtained in steps 2 and 3 is equal to 0.3 units and 0.625°, respectively, for link locations and orientations. A feasible initial rotating degree, 5°, and a precision degree, 0.625°, are used in our simulations. Using the initial rotating degree, a binary search algorithm will take four time checks at most to determine a rotation. It is obvious that the orientation of a link may change insignificantly, when it moves to next configuration. In other words, the rotating degree of a link is small and the planned paths are smooth. Therefore, the initial degree, 5°, for each rotation is a feasible choice.

Figure 5. A path planning example derived from ref. 15.

Figure 5 shows the path planning result similar to an example shown in ref. 15. In our simulation, there are two GLs predetermined in the workspace of size 30 by 20 units. The initial distance of translation, δ_0,

Table I. Computation time for manipulator configurations shown in Figure 5.

Computation times (s)	
config.1: 1.6000	config.7: 1.2900
config.2: 0.4200	config.8: 0.2900
config.3: 0.2300	config.9: 0.1200
config.4: 0.1500	config.10: 0.1600
config.5: 0.0330	config.11: 0.8900
config.6: 0.1400	**total time: 5.6600**

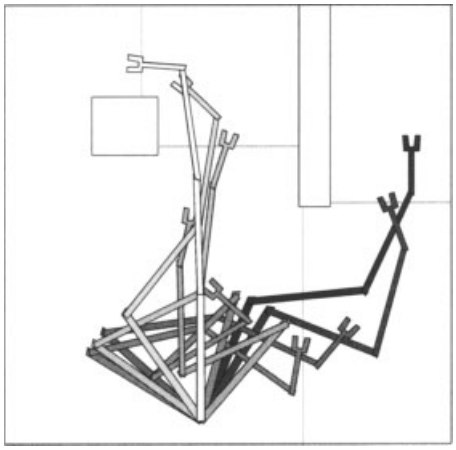

Figure 6. A path planning example derived from ref. 16.

is arbitrarily chosen as 2.2 units. The simulation result takes a total of 5.66 s to plan the 11-configuration collision-free path and the computation time for each configuration is shown in Table 1. The path planning spends most CPU times for the first and seventh configurations. The planned manipulator path is observed to achieve better results in terms of collision avoidance compared with the path obtained in ref. 15. Figure 6 shows the path planning result similar to an example shown in ref. 16. In our simulation, there are four GLs predetermined in the workspace of size 30 by 30 units. The initial distance of translation, δ_0, is chosen as 4.0 units. While it takes 5 s for a SUN SPARC to generate a four-configuration path in ref. 16, the eight-configuration path is obtained by the proposed algorithm in 0.87 s for a SUN Ultra-1. In this simulation, the shortest computation time is obtained when the δ_0 is equal to 4 units.

In general, the number of configurations and the computation time depend on the size of the initial distance of translation, δ. For larger δ, fewer configurations will be generated along the path if there is no collision. On the other hand, if δ is too large, the computation time may increase because collisions may occur frequently. Figure 7 shows the path planning result similar to an example shown in ref. 6. In our simulation, there are seven GLs predetermined in the workspace of size 30 by 20 units. The initial distance of translation, δ_0, is chosen as 2.2 units. The simulation result takes a total of 6.5934 s to plan the 16-configuration collision-free path. Table 2 shows the total computation time and the number of configurations of the planned manipulator path, for different values of δ_0.[6] For the PRM presented in ref. 6 using a DEC Alpha, a computation time of 5 s only corresponds to success rates of 50%. For a success rate of more than 97%, the total computation time will exceed 15 s.[7]

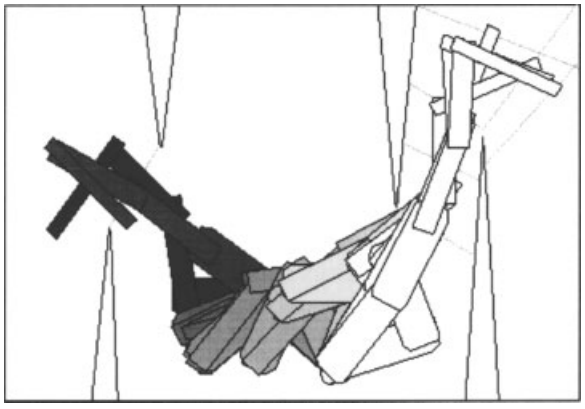

Figure 7. A path planning example derived from ref. 6.

Table II. Computation time of various δ.

δ_0	Total time	No. of configurations
2.4	6.2637	15
2.2	6.5934	16
1.5	7.5275	20
1.1	9.8901	28
0.5	16.4835	63

The proposed algorithm also performed well for high DOF manipulators as shown in Figure 8. There are three GLs predetermined in the workspace of size 30 by 30 units. The initial distance of translation, δ_0, is chosen as 1.2 units. The computation time of the 11-configuration path is equal to 8.13 s. Figure 9 shows another high DOF example in which a nine-link manipulator need to snake into the cave with a

[6]One can see from Table 2 that the products of δ_0 and the number of configurations are about the same, about 30, except for the two largest δ_0's wherein collision will occur more frequently. As for the selection of an initial δ_0, it seems that a value of about 10% of the size of the workspace is a reasonable selection for an initial δ_0, which corresponds to a maximum of 10 steps for the manipulator to reach any goal in a straightforward path planning problem. For the path planning problems considered in this paper, the initial δ_0's are chosen between 4% and 15% of the workspace size such that enough configurations can be generated to show the spatial smoothness of the planned path.

[7]In general, a direct comparison of the proposed approach with the PRM method is not trivial. While a sequence of GLs is assumed given in the former, preprocessing of the construction of c-space is needed in the latter.

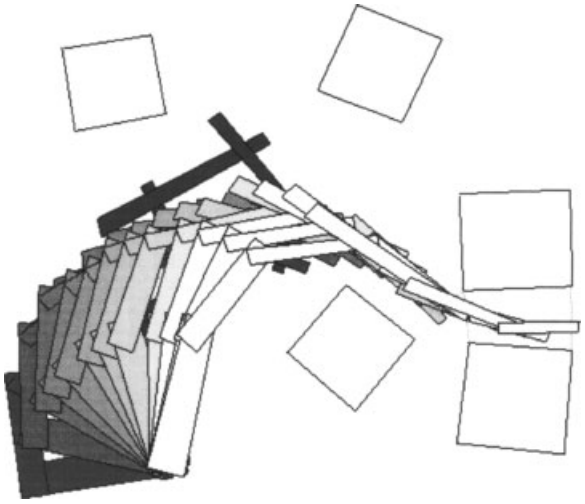

Figure 8. A six-link manipulator example.

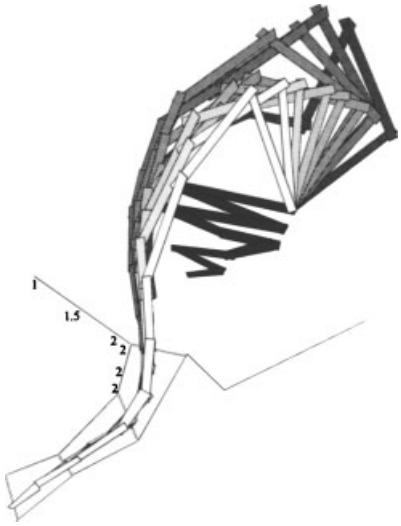

Figure 10. A nine-link manipulator example with nonuniform charge distributions.

collision-free path. There are four GLs predetermined in the workspace of size 30 by 30 units. The initial distance of translation, δ_0, is chosen as 2.2 units. The computation time of the 11-configuration path takes a total of 9.26 s.

In Figure 9, the manipulator may move very close to an obstacle's corner (as indicated with a pointer). In order to keep a safe distance away from the corner, the charge distributions of the left side of the cave is modified to be nonuniform and a safer manipulator path is obtained, as shown in Figure 10. In Figure 10, the charge density is doubled for the segment marked with three 2's and is increased linearly from 1 to 2 for the other segment marked with three different numbers. The computation time is equal to 12.4 s for the path shown in Figure 10 as the calculation of potential with new charge distribution is more complex. In general, for similar robot/obstacle shapes and charge distributions, the computation time will be proportional to (i) the number of line segments of robot, (ii) the number of line segments of obstacles, and (iii) the number of robot configurations calculated.

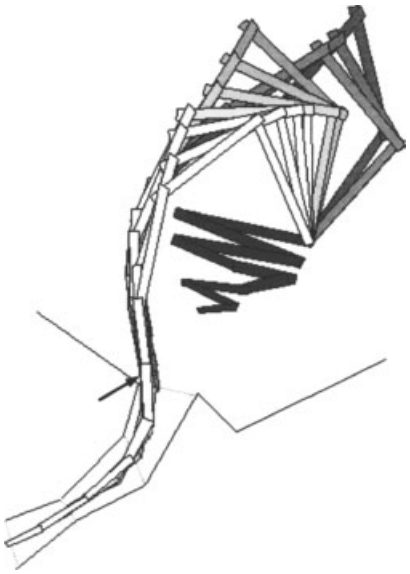

Figure 9. A nine-link manipulator example with uniform charge.

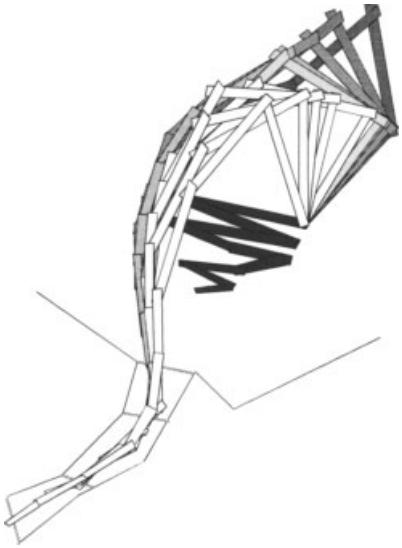

Figure 11. A path obtained by changing a GL in Figure 10.

Finally, if the GLs are changed in position, the change of the planned path is usually not significant. For example, Figure 11 shows a path obtained by changing the position of a GL in Figure 10. While the GL connects two vertices, one from each side, of the polygonal passage in Figure 10, the modified GL shown in Figure 11 corresponds to an MDL. One can see easily that the two paths are very similar.

5. CONCLUSION AND FUTURE WORK

In this paper, a potential-based algorithm for the path planning of a robot manipulator is proposed. The proposed algorithm uses an artificial potential field to model the workspace wherein object boundaries are assumed to be charged with various source distributions. The repulsive force and torque between objects thus modeled are analytically tractable, which makes the algorithm efficient. To give a general direction of the path, a sequence of guide lines to be reached by the manipulator is assumed to be given in advance in the workspace. As a GL is an intermediate or final goal for a manipulator to reach, it also helps to establish certain motion constraints for adjusting manipulator configuration during path planning. According to such constraints, the proposed approach derives the path for a manipulator by adjusting its configurations at different locations along the path to minimize the potential using the above force and torque. Simulation results show that a path thus derived is always spatially smooth while its safety can be adjusted easily by selecting proper charge distributions for the potential-based workspace model. Since the proposed approach uses workspace information directly, it is applicable to manipulators of high DOFs. In our experiments, manipulators with DOFs up to 9 have been tested with satisfactory results.

The proposed approach can be extended to cases involving various types of manipulators, and to workspace of higher dimensions with some modifications. In 3D environment, the potential model proposed in ref. 17 can be used to model the 3D workspace. Accordingly, the minimization of potential in 3D workspace is extended to a search on a "guide plane" for the configuration adjustment (see ref. 14). On the other hand, since the proposed approach performs nicely in a narrow passage in terms of collision avoidance, it may very well be used as a local planner in a PRM context if the PRM can provide the needed initial conditions to the local planner.

REFERENCES

1. J.C. Latombe, Motion planning: A journey of robots, molecules, digital actors, and other artifacts, Int J Robot Res 18:(11) (1999), 1119–1128.
2. T. Lozano-Perez, Spatial planning: a configuration space approach, IEEE Trans Comput C-32:(2) (1983), 108–120.
3. R. Brooks and T. Lozano-Preez, A subdivision algorithm in configuration space for find path with rotation, IEEE Trans Syste Man Cyberne 15:(2) (1985), 224–233.
4. J. Barraquand and J.C. Latombe, Robot motion planning: a distributed representation approach, Inte J Roboti Res 10:(6) (1991), 628–649.
5. L. Kavraki and J.C. Latombe, Randomized preprocessing of configuration space for fast path planning, Proc IEEE Int Conf on Robotics and Automation, 1994, pp. 2138–2139.
6. L.E. Kavraki, P. Svestka et al., Probabilistic roadmaps for path planning in high-dimensional configuration spaces, IEEE Trans Robot Automa 12:(4) (1996), 566–580.
7. J.-H. Chuang and N. Ahuja, An analytically tractable potential field model of free space and its application in obstacle avoidance, IEEE Trans Syst Man Cybern Part B: Cybern 28:(5) (1998), 729–736.
8. J.-H. Chuang, Potential-based modeling of three-dimensional workspace of the obstacle avoidance, IEEE Trans Robot Automa 14:(5) (1998), 778–785.
9. A. Thanailakis et al., Collision-free path planning for a diamond-shaped robot using two-dimensional cellular automata, IEEE Trans Robot Automa 13 (1997), 237–250.
10. Y.K. Hwang and N. Ahuja, Gross motion planning a survey, ACM Comput Survey 24:(3) (1992), 219–291.
11. P. Khosla and R. Volpe, Superquadric artificial potentials for obstacle avoidance and approach, Proc IEEE Int Conf on Robotics and Automation, 1988, pp. 1778–1784.
12. J.-H. Chuang, C.-H. Tsai, W.-H. Tsai, and C.-Y. Yang, Potential-based modeling of 2-d regions using nonuniform source distribution, IEEE Trans. Syst Man Cybern Part A: Syst Hum 3:(2) (2000), 197–202.
13. D.T. Kuan, J.C. Zamiska, and R.A. Brooks, Natural decomposition of free space planning, Proc IEEE Int Conf on Robotics and Automation, 1985, pp. 168–173.
14. C.-C. Lin, C.-C. Pan, and J.-H. Chuang, A novel potential-based path planning of 3-D articulated robots with moving bases, Proc IEEE Int Conf on Robotics and Automation, September 2003, pp. 3365–3370.
15. T. Lozano-Perez and M.A. Wesley, An algorithm for planning collision-free paths among polyhedral obstacles, Commun ACM, 22:(10) (1979), 560–570.
16. T. Laliberte and C. Gosselin, Efficient algorithms for the trajectory planning for redundant manipulators with obstacle avoidance, Proc IEEE Int Conf on Robotics and Automation, 1994, pp. 2044–2049.
17. C.-H. Tsai, J.-S. Lee, and J.-H. Chuang, Path planning of 3-D objects using a new workspace model, IEEE Trans Syst Man Cybern Part C: Appl Rev 31:(3) (2001), 405–410.