Path Planning of 3-D Objects Using a New Workspace Model

Chi-Hao Tsai, Jou-Sin Lee, and Jen-Hui Chuang

***Abstract*—This paper proposes a collision avoidance algorithm to solve the problem of (local) path planning for a three-dimensional (3-D) object moving among polyhedral obstacles. The algorithm is based on a generalized potential model of workspace [1] which assumes that the boundary of every 3-D object is uniformly charged. According to the proposed approach, the repulsive force and torque between the moving object and the obstacles due to the above model is used to adjust the position and orientation of the object so as to keep it away from the obstacles while passing through a bottleneck in the free space. Simulation results demonstrate that the path of a 3-D object thus obtained is indeed safe and spatially smooth. The adopted potential field is analytically tractable which makes the path planning efficient.**

***Index Terms*—Collision avoidance, generalized potential model, path planning.**

I. Introduction

In planning a path of a robot, a repulsive potential function is usually used to keep a safe distance between the robot and obstacles. A collision-free path of a robot can be obtained by adjusting its configuration to minimize the potential experienced by the robot. In general, a potential function used to model the workspace can be a scalar function of the distances between the boundary points of the robot and those of obstacles. The gradient of such a scalar function can be used as a repulsive force between the robot and obstacles, making potential-based methods simple. (For a survey of related works see [2] and [3].) An artificial repulsive potential whose value is determined by the Yukawa function [4] and whose isopotential contours are modified n-ellipses is used in [5] for local planning of linked line segments. The potential from an obstacle is given by

$$U(K) = A\frac{e^{-\alpha K}}{K} \tag{1}$$

where the pseudodistance K is made to change linearly along the x-axis and is used to specify each contour. Ideally, as mentioned in [5], a potential field should have the following attributes.

1) The magnitude of potential should be unbounded near obstacle boundaries and should decrease with range. (This property captures the basic requirement of collision avoidance.)
2) The potential should have a spherical symmetry far away from the obstacle.
3) The equipotential surface near an obstacle should have a shape similar to that of the obstacle surface.
4) The potential, its gradient, and their effects on paths must be spatially continuous.

A potential function which is a cubic function of the distance between a point object and the obstacles is used in [6] for moving a point object in the two–dimensional (2-D) space. The potential function ranges from zero at some maximum distance to a maximum value ($<\infty$) at zero distance. In [7], local planning similar to that discussed in [5] is done using an artificial potential function which is a function of the shortest distance between the moving object and the obstacles. The potential function is described by

$$U(r) = A\left(\frac{1}{r} - \frac{1}{r_0}\right)^2 \tag{2}$$

where r is the shortest distance and r_0 is the effective range. Clearly, the cubic function mentioned in [6] does not have the first (unbounded) attribute. Furthermore, at the locations where the shortest distance corresponds to multiple obstacle points, the gradient of the potential function will be undefined (the same problem exists for the potential function used in [7]).

Harmonic functions which do not exhibit local minimum in the free space are used in [8] to find object trajectories in the configuration space. Since the potential along an obstacle of nonzero extent is finite, the only obstacle structure for which collision avoidance can be guaranteed is a point itself (see [1] for a more detailed discussion). For each given source/goal pair in the configuration space, an iterative method is used to generate a discrete regular sampling of a potential field on a grid numerically such that following the gradient from the start point will move the robot to the goal safely. A potential function, called a navigation function, is constructed in [9] for a point object to navigate among disk obstacles toward the goal position. By adjusting a parameter of the potential function, all local minima can be removed. This algorithm is later generalized to star-shaped sets [10]. Harmonic potential due to an electric charge in the 2-D space

$$U(r) = q\ln\frac{1}{r} \tag{3}$$

is used for obstacle avoidance in [11]. Similar potential function is used in a sensor-based 2-D potential panel method for robot motion planning in both a static and dynamic environment [12]. The approach in [11] is later generalized in [13] to the 3-D space by considering the 2-D plane determined by the source and goal points as well as the point charge representing the closest obstacle.

Boundary equations of polytopes are used in [14] to create an artificial potential function. Let

$$g(x) \le 0 \quad x \in R^n \tag{4}$$

be the set of linear inequalities describing a convex region. Assuming there are N boundary polytopes, the potential function is defined as

$$p(x) = \frac{1}{\delta + f(x)} \tag{5}$$

where δ is a small number and the scalar function

$$f(x) = \sum_{i=1}^{N} g_i(x) + |g_i(x)| \tag{6}$$

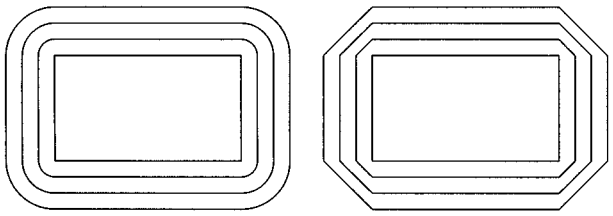
is zero inside the region and grows linearly as the distance from the region increases. In [15], an efficient and simple method for finding a collision-free object path in a dynamically observable 3-D environment is developed by combining the above potential function and the octree representation [16].

It is easy to see that the potential functions described in (2) and (5) do not have the attribute of spherical symmetry. For example, the equipotential contour of (2) due to a rectangle never converges to a circular shape in far field in the sense that the difference between the maximum and minimum distances from points on any contour to the centroid of the rectangle is always equal to the length difference between two neighboring edges of the rectangle [see Fig. 1(a)]. Similarly, the

Manuscript received April 16, 2000; revised August 30, 2001. This work was supported by the National Science Council of Taiwan, R.O.C., under Grant NSC 89 2213 E009 103. This paper was recommended by Associate Editor R. A. Hess.

The authors are with the Department of Computer and Information Science, National Chiao Tung University, Hsinchu 30010, Taiwan, R.O.C.

Publisher Item Identifier S 1094-6977(01)10034-9.



(a) (b)

Fig. 1. Equipotential contours of the potential functions specified by (a) (2) and (b) (5) due to a rectangle.

equipotential contours of (5) due to a rectangle always consist of sets of parallel line segments [see Fig. 1(b)].

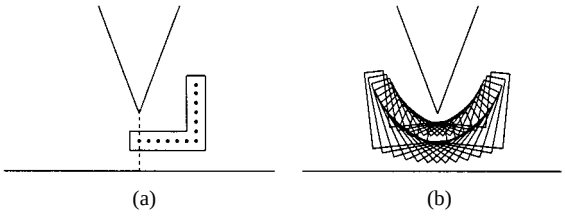
The Newtonian potential which is inversely proportional to the distance between two point-charges is used in [17] for the path planning in the 2-D space. By assuming that the polygonal obstacle boundaries are uniformly charged, it is shown that the resultant potential field have the above attributes. Moreover, such a workspace model is unique in the following ways.

1) The resultant potential field is obtained by superposing the potential due to individual boundary point directly, instead of intermediate representation of the obstacle boundaries such as boundary equations. Furthermore, each boundary point contributes to the potential field in an independent and identical fashion.
2) An analytic expression of the potential function due to a line segment enables the computation of the exact potential for the obstacles, avoiding the need to discretize the obstacle boundary into a set of points.
3) The potential field and its gradient are analytically computable throughout the free space. Therefore, establishing a database of the potential function, e.g., a distance map, upon a discrete representation of the free space is not necessary.

It is not hard to see that the potential field established in [5] is not obtained by superposing the potential due to individual boundary point directly. Thus, whether each boundary point is contributing to the potential field in an independent and identical fashion is out of the question. In fact, since the definition of the pseudodistance depends on the orientation of the coordinate system, the potential field is not rotational invariant except for circularly symmetric obstacles, e.g., disks, in the 2-D space.

In [17], an algorithm is developed to compute a safe and smooth object path by minimizing the potential function locally for obstacle avoidance, while the gross robot movement is subject to the constraints derived from the topology of the path given *a priori*. Since the potential gradient is not used directly as the direction of object path, like many potential-based approaches, the potential minima, in general, will not cause an object to get stuck during the path planning.

A 3-D extension of the above potential-based path planning approach is proposed in this paper. It is shown that with the analytically tractable, potential-based free space model proposed in [1], collision avoidance can be achieved effectively and efficiently. Although this paper mainly considers path planning for a single rigid object among stationary and rigid obstacles in the 3-D space, the concept of obstacle avoidance using potential fields can be extended easily to more general path planning problems. The remainder of this paper is organized as follows. The above generalized potential model in the 3-D space is briefly reviewed in the next section. A path planning algorithm using the closed-form expressions of repulsive force and torque between the moving object and the obstacles due to the generalized potential model to ensure collision avoidance is proposed in Section III. In Section IV, simulation results are shown for several 3-D objects moving among polyhedral obstacles. Section V summarizes the paper.



(a) (b)

Fig. 2. Two-dimensional (2-D) path planning example. (a) Initial conditions. (b) Resulting object path.

II. Review of Generalized Potential Fields in the 3-D Space

In [1], a potential-based modeling of 3-D workspace for collision avoidance is proposed. It is shown that the Newtonian potential, being harmonic in the 3-D space, cannot prevent a point charge from running into an object surface which is uniformly charged. This is because the value of such a potential function is finite at the continuously charged surface. Subsequently, generalized potential models are developed to assure collision avoidance between 3-D objects. The potential function is inversely proportional to the distance between two point charges to the power of an integer (m) and the potential and thus its gradient due to a 3-D polygon can be calculated analytically.

In particular, it is shown that the repulsive force exerted on a point charge due to polyhedral surfaces can be obtained analytically by evaluating the gradient of the following function

$$\Phi(x,\, y,\, z) = \frac{1}{z}\tan^{-1}\frac{xz}{y\sqrt{x^2+y^2+z^2}} \tag{7}$$

at some $(x,\, y,\, z)$s for $m = 3$.

In general, the force exerted on a point due to polyhedral object surfaces, which will be used in the proposed 3-D path planning algorithm discussed next, can be obtained by summing the forces due to individual polygonal object faces. Finally, it is also shown in [1] that the generalized potential will diverge for a point charge located on the surface of a polyhedral object, i.e., the basic requirement for the proposed path planning algorithm is satisfied.

III. Local Planning Algorithm

In the previous section, it is shown that the repulsive force exerted on a point due to polyhedral surfaces can be obtained by evaluating the gradient of (7). In this section, the above results will be used to achieve collision avoidance for path planning in the 3-D space. To that end, the obstacles are represented as polyhedra, while a moving object is represented by a set of sampling points obtained from its surfaces.[1] The repulsive force and torque between the moving object and the obstacles can thus be obtained by superposing the repulsive force and torque between individual sampling points and obstacle surfaces.

For a path planning problem, the places where the moving object is more likely to collide with obstacles are bottlenecks in the free space. In [20], free space bottlenecks in the 2-D space are represented by the minimal distance links (MDLs) among obstacles, which also connect (convex) obstacle nodes in the obstacle neighborhood graph. In [17], a local planner is developed to obtain a local path around an MDL, e.g., the dashed line segment shown in Fig. 2(a). With the leading skeleton point of the object initially located on the MDL, the local planner determines the optimal location and orientation of the object, as successive

[1]Unless otherwise specified, it is assumed that the point samples are fairly uniformly distributed over the object surfaces.

skeleton points are moved onto the MDL, such that the potential experienced by the object is minimized. Fig. 2(b) shows the path planning results thus obtained. The effectiveness of the local planner in generating safe and smooth object path is readily observable.

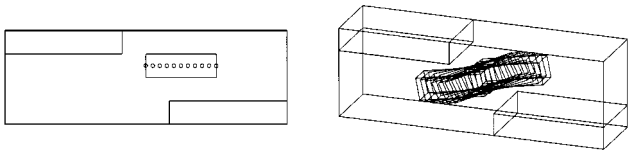
In this section, the above local planner is generalized to the 3-D space using the potential-based workspace model reviewed in the previous section. For simplicity, each bottleneck region in the 3-D space is represented by a polygon. The topology of a local path, given as input to the local planner, is described in terms of the associated bottleneck and the object skeleton. The description is of a very concise form which only specifies the sequence in which the skeleton points should cross the bottleneck. If such a description corresponds to a feasible object path, the local planner will generate a sequence of object configurations along the path, each of minimum potential; otherwise, a failure will be reported. In the latter case, the failure may also result from a violation of a predetermined safe margin.

Let s_i, $1 \leq i \leq N$ denote the sequence of N selected skeleton points to cross the bottleneck P. The local path begins when s_1 reaches P and ends when s_N leaves P. The following algorithm developed for the local planner performs the path planning by sequentially ensuring that as each skeleton point moves onto P, it stays on P while the location and orientation of the object are adjusted to minimize the potential function using repulsive force ($\mathbf{F}$) and torque ($\mathbf{T}$), respectively, experienced by the object. Additional skeleton points may be added (see algorithm) to reduce the step size along the path, allowing for finer adjustments in the object configuration to avoid collision. The total number of skeleton points used directly determines the number of optimal object configurations computed along the path and, thus, the computation time; therefore, it is desirable to use as few skeleton points as possible. To restrict the total amount of computation, a limit is placed upon the minimum spacing $s_{\min}$ between adjacent skeleton points used in the simulation, which effectively serves as a feasibility test of the local plan.

ALGORITHM *LOCAL_PLAN*

1. (Begin with the first skeleton point)
   Initialize $i = 1$.
2. (Find the minimal potential object configuration)
   Translate the object on P and rotate it with respect to s_i until $\mathbf{T}$ and the projection of $\mathbf{F}$ on P are both zero.
3. (End when done with the last skeleton point)
   If $i = N$, the local planning is completed.
4. (Try for the next skeleton point)
   Translate the object such that s_{i+1} is shifted to its projection on P. If there is no collision during the translation, then let $i \leftarrow i+1$, and go to Step 2.
5. (Use an intermediate skeleton point)
   Find the smallest $n \geq 1$ such that $s'_i = s_i + (s_{i+1} - s_i)/2^n$ can be shifted to its projection on P without collision between the object and obstacles. If $|s'_i - s_i| > s_{\min}$, then let $s_i \leftarrow s'_i$, and go to Step 2.
6. (End abnormally)
   Exit with failure due to the need for less than allowed spacing of skeleton points.

An object configuration obtained in Step 2 is not only collision-free but also the safest with respect to the generalized potential under the constraint that the corresponding skeleton point stays on P. For the implementation of Step 2, the minimal potential configuration is identified efficiently by performing two gradient-based searches: one for the object location using $\mathbf{F}$ and the other for the object orientation using $\mathbf{T}$. Different from the 2-D local planner developed in [17], the translation and rotation are both performed in three dimensions instead of one and two dimensions, respectively, for the 2-D case. The precisions required for specifying the final object location and orientation determine the number of iterations needed for solving the corresponding constrained optimization problems. For the simulation results presented in the next section, the minimal potential configurations are specified to within 1/160 of the length of the maximal distance between the vertices of P in location and within 1° in orientation.



(a) (b)

Fig. 3. Path planning example. (a) Side view of the initial conditions. (b) Resulting object path.

IV. Simulation Results

In this section, simulation results are presented for path planning of objects moving around bottleneck regions in the free space. The algorithm is written in the C programming language and all examples are tested on a Sun Sparc Ultra-1 workstation. In order to make the observation easy, a local path obtained with *LOCAL_PLAN* may be shown in different perspectives.

Fig. 3 shows a rectangular solid, represented by a set of 58 sampling points uniformly distributed over its surfaces, moving through a stairlike region with a rectangular bottleneck. The safe and smooth object path is represented by 11 object configurations which correspond to 11 skeleton points equally spaced on a linear skeleton, as shown in Fig. 3(a). No additional skeleton point is added while running the algorithm, i.e., Step 5 of *LOCAL_PLAN* is not necessary because sufficient skeleton points are used.

Fig. 4 shows an L-shaped solid with 48 sampling points moving through a V-shaped bottleneck region. While the two L-shaped faces of the object lie in $z = \pm 0.5$ planes, respectively, the two largest obstacle faces lie in $z = \pm 1$ planes, respectively, and points A, B, C, D, and E lie in the $z = 0$ plane, as shown in Fig. 4(a). For points A, B, and C chosen as skeleton points of the object, the local path derived by the *LOCAL_PLAN* is shown in Fig. 4(b). Five intermediate skeleton points, and thus five additional object configurations, are required to accomplish the local planning. The robustness of the proposed approach with respect to the selected object skeleton is illustrated with Fig. 4(c) and (d) wherein points D and E, respectively, are used in place of B. It is not hard to see that the two object paths are very similar to the one shown in Fig. 4(b) except for different numbers and locations of intermediate samples of object configurations taken along the underlying continuous object path.

In each of the above two examples, because there is a plane of symmetry for both the moving object and the obstacles, and the plane also contains the skeleton points of the object, the corresponding path planning problems are essentially two-dimensional. In Fig. 5, a truly 3-D problem is established by tilting outward part of two obstacle faces shown in Fig. 4. The object path, shown in four different perspectives, is derived by *LOCAL_PLAN* for seven equally spaced point samples obtained along the skeleton determined by points A, B, and C. A twisting movement of the object in accordance with the tilts to reduce the likelihood of collision can be seen clearly.

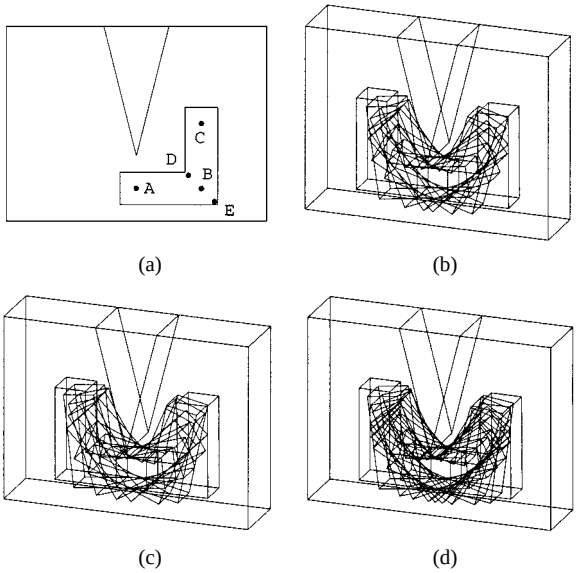


Fig. 4. Single-bottleneck path planning examples. (a) Side view of initial conditions and different sets of skeleton points. (b) Local path obtained for A, B, C. (c) Local path obtained for A, D, C. (d) Local path obtained for A, E, C.

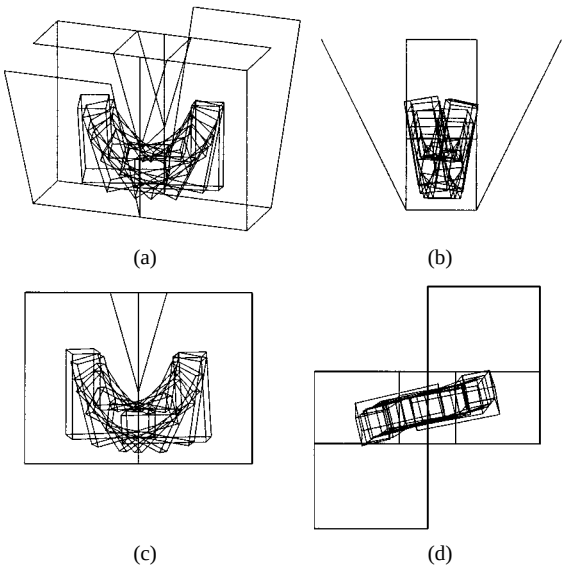


Fig. 5. Single-bottleneck path planning example shown in four different perspectives.

In general, the computational complexity of the path planning is determined by the shape of the object and obstacles, the number of skeleton points used, the number of sampling points of the object, the number of obstacle faces, and the precisions in specifying final object configurations. For example, it takes 47.78, 21.72, and 27.94 s to generate object paths shown in Figs. 3, 4(b), and 5, respectively. Fig. 6 shows a spiral with an angular range of 720° with respect to its axis going through a small rectangular hole on the right side of a box. The skeleton, which is the spiral itself, has 64 equally spaced sampling points. The complexity of this example is much higher than the previous three examples that it takes 1943.48 s to generate the local path of the object.

Figs. 7 and 8 show path planning examples involving multiple local paths. In each example, it is assumed that the order in which the local paths should be connected to form a global path is given in advance. Furthermore, it is assumed that each local path is relatively close to its neighbors that the path planning for connecting adjacent local paths into a global one is straightforward.

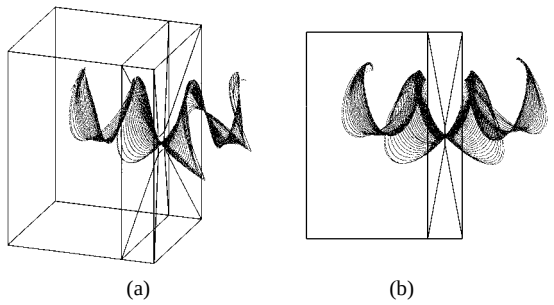


Fig. 6. Single-bottleneck path planning example shown in two different perspectives.

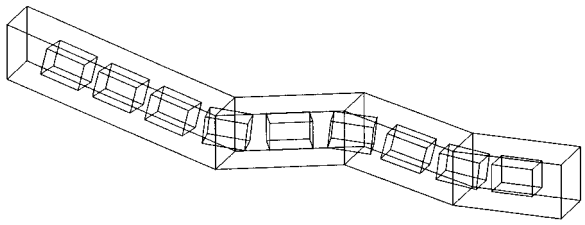


Fig. 7. Multibottleneck path planning example.

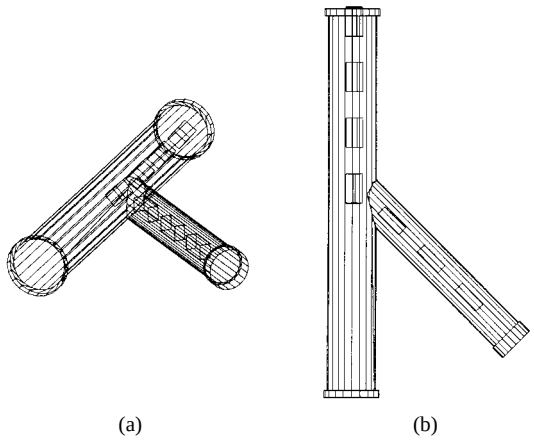


Fig. 8. Multibottleneck path planning example.

Fig. 7 shows a rectangular solid with eight sampling points (its vertices) moving through a piecewise linear passage. The passage corresponds to the swept volume obtained by translating a square cross-section, from right to left, along the horizontal $(-1, 0, 0)$ direction, the $(-4, 1, 0)$ direction, the $(-2, 0, 1)$ direction, and finally the $(-8, 2, -1)$ direction. The shape of the object is represented by a single skeleton point and its centroid; each local path, consisting of a single object configuration, is obtained for the skeleton point constrained on the square cross-section at one of the nine selected locations. (Unlike a bottleneck, these square cross-sections do not really reflect local narrowness of the free space along such a passage. Nonetheless, as the bottlenecks considered so far, each of these cross-sections does provide similar constraint for *LOCAL_PLAN* to derive an object configuration of minimum potential.) It is easy to see that the object stays pretty close to the center line of the passage while keeping away from its boundaries.

In Fig. 8, a similar rectangular solid is moved inside a Y-shaped pipe consisting of 170 polygons. The object is moved from the lower end of the slim pipe to the upper end of the wider one, as shown in Fig. 8(b), through the identification of local paths obtained for seven

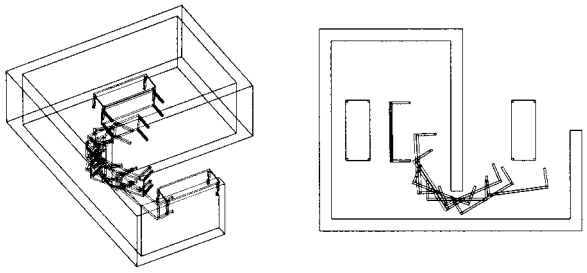


Fig. 9. More realistic example.

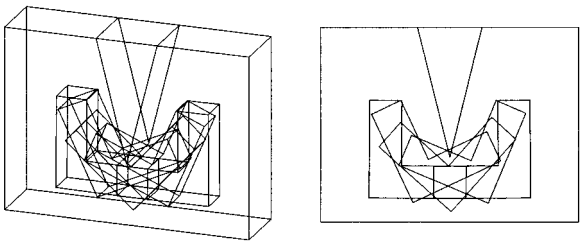


Fig. 10. Problematic path planning example due to insufficient sampling points.

cross-sections. There is essentially no change in object orientation, due to the local symmetry in the shape of a pipe, except near the junction of the two pipes where the object undergoes a composition of 1) a turning movement due to the change in the passage direction and 2) a twisting movement, i.e., a rotation of about 45°, due to the wideness of the free space. Fig. 9 shows a more realistic example of moving a desk out of a room. Intelligent maneuvering of the desk at the doorway (bottleneck region) is necessary to generate a collision-free path.

A. Computational Complexity

The proposed approach is not only easy to implement, but the associated algorithm is also of low time complexity because the generalized potential field is analytically tractable. Suppose the obstacles are represented by totally n polygonal surfaces and the moving object is represented by m sampling points. The time complexity for calculating the repulsive force and torque exerted on each sampling point of the object is $O(n)$. Thus, the time complexity for obtaining $\mathbf{F}$ and $\mathbf{T}$ exerted on the moving object in Step 2 of *LOCAL_PLAN* is $O(nm)$. However, the total number of translations (and rotations) required in adjusting the position (and orientation) of the object for minimum potential, and thus the total number of calculations made for $\mathbf{F}$ (and $\mathbf{T}$), is, in general, dependant on the shape of the object and that of the free space, and cannot be expressed analytically. For example, consider the local paths shown in Figs. 4(b) and 5 which have the same m and n. While an additional object configuration is required to derive the local path shown in Fig. 4(b) compared with that in Fig. 5, because the latter has a more open free space, it takes less time on the average to compute an object configuration in the former wherein the free space has a simpler shape.

As for a rough comparison of overall performance in terms of efficiency, most of the examples presented so far in this paper are computed in less than a minute, except for the curved object/obstacle shown in Figs. 6 and 8. Such results compare favorably to those obtained with the potential-based approach presented in [14] for 3-D problems of similar complexity. In [14], without giving the computation time for individual examples, the times spent on path planning are reported to range from 5 to 30 min on a Sun 3/260 computer.

B. Unsuccessful Path Planning Results

While satisfactory path planning results are obtained in the above examples, there are certain situations in which the path planning performed by the proposed local planner can be problematic. Fig. 10 shows a path planning example similar to that shown in Fig. 4 but uses less sampling points, the 12 vertices, to represent the L-shaped object. Although these points are free from collision along the derived object path, it is not the case for this polyhedral object as a whole since the distribution of them over the object surfaces is too sparse.

On the other hand, the way the simple topological description of an object path is used by the local planner does not always yield meaningful path planning results. Fig. 11(a) shows the side view of a path

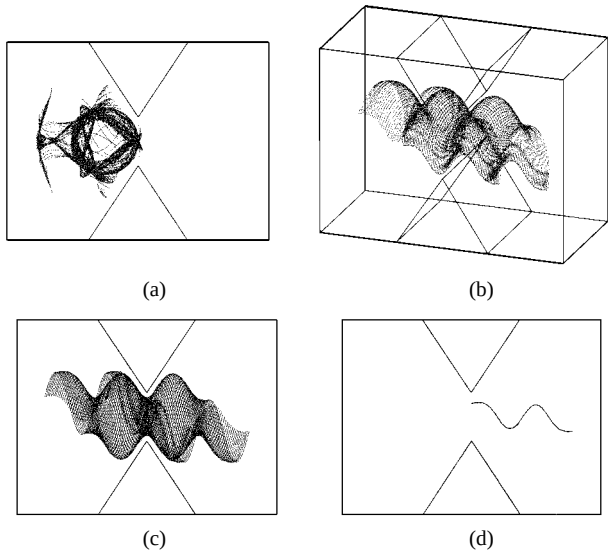


Fig. 11. (a) Side view of another problematic path planning example. (b) Path planning results obtained with a simplified version of *LOCAL_PLAN* (see text). (c) Side view of (b). (d) Path of the first skeleton point obtained from (c).

obtained for the same spiral used in Fig. 6 but with different obstacles. With object configurations of minimum likelihood of collision obtained for the specified sequence of skeleton points, the object ends up on the same side of the bottleneck. This is because no restriction is imposed upon the motion of a skeleton point that, once it reaches the bottleneck, it should move forward and stay on the other side of the bottleneck. To further examine the influence of different potential minimization approaches on path planning, on an empirical basis, Fig. 11(b) shows successful path planning results obtained with a simplified version of *LOCAL_PLAN* which only allows object translation. One can see easily from the side view of the path, as shown in Fig. 11(c), that the object path is safe and smooth. The object, in fact, undergoes a 2-D sinusoidal movement, as illustrated by the path of the first skeleton point shown in Fig. 11(d).

V. Conclusion and Future Works

A local path planning algorithm based on a new workspace model is presented in this paper. The effectiveness of such a model for ensuring collision avoidance in path planning problems is demonstrated by considering the local path going through a free space bottleneck in which careful maneuvering of the object is required. A local planner is developed to identify the optimal (minimum potential) object configuration along the object path around a free space bottleneck. Unlike approaches that only work for a point or spherical object, possibly in the C-space, the object considered in the proposed approach, which is represented by point samples from its surface, is allowed to have arbitrary shape. The optimal object configurations can be obtained with efficient search

methods because the repulsion between the object and obstacle is available in closed-form. According to the simulation results, not only can an object configuration obtained with the proposed approach avoid obstacles with satisfactory (optimal) margins, a sequence of object configurations thus obtained also connect naturally into a spatially smooth object path. Preliminary results of connecting local paths obtained with the proposed local planner into a global one are also included.

Despite the aforementioned success in applying the proposed algorithm in 3-D path planning, several related issues are yet to be addressed. For example, the sampling of the object surface is not a trivial problem. There is certainly a tradeoff between the computation efficiency and the correctness in the resultant object path. Other issues include the developments of a systematic way of identifying free space bottlenecks of more complex geometry, suitable global planning strategies to connect the local paths, and other local planning algorithms. On the other hand, it is possible to combine the proposed algorithm with some other global path planning approaches, e.g., a probabilistic roadmaps method presented in [21]. Extensions of the proposed approach to more general problems, other than the one involvinga single rigid object among stationary obstacles, are also under investigation.

REFERENCES

[1] J.-H. Chuang, "Potential-based modeling of three dimensional workspace for obstacle avoidance," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 778–785, Oct. 1998.

[2] Y. K. Hwang and N. Ahuja, "Gross motion planning: A survey," *ACM Comput. Surv.*, vol. 24, no. 3, pp. 219–291, 1992.

[3] J. C. Latome, *Robot Motion Planning*. Norwell, MA: Kluwer, 1991.

[4] B. Cohen-Tannoudji, C. Diu, and F. Laloe, *Quantum Mechanics*. New York: Wiley, 1977, vol. 2.

[5] P. Khosla and R. Volpe, "Superquadric artificial potentials for obstacle avoidance and approach," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1988, pp. 1778–1784.

[6] C. E. Thorpe, "Path planning for a mobile robot," in *Proc. AAAI*, 1984, pp. 318–321.

[7] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1985, pp. 500–505.

[8] C. I. Connolly, J. B. Burns, and R. Weiss, "Path planning using Laplace's equation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1990, pp. 2102–2106.

[9] D. E. Koditschek, "Exact robot navigation by means of potential functions: Some topological considerations," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1987, pp. 1–6.

[10] E. Rimon and D. E. Koditschek, "The construction of analytic diffeomorphism for exact robot navigation on star worlds," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1989, pp. 21–26.

[11] J. Guldner and V. I. Utkin, "Sliding mode control for gradient tracking and robot navigation using artificial potential fields," *IEEE Trans. Robot. Automat.*, vol. 11, pp. 247–254, Apr. 1995.

[12] Y. Zhang and K. P. Valavanis, "Sensor-based 2-D potential panel method for robot motion planning," *Robotica*, vol. 14, pp. 81–89, 1996.

[13] J. Guldner *et al.*, "Obstacle avoidance in R^n based on artificial harmonic potential fields," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1995, pp. 3051–3056.

[14] Y. K. Hwang and N. Ahuja, "A potential field approach to path planning," *IEEE Trans. Robot. Automat.*, vol. 8, pp. 23–32, Feb. 1992.

[15] Y. Kitamura *et al.*, "3-D path planning in a dynamic environment using an octree and an artificial potential field," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, vol. 2, 1995, pp. 474–481.

[16] M. Herman, "Fast, three-dimensional, collision-free motion planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1986, pp. 1056–1063.

[17] 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. B*, vol. 28, pp. 729–736, Oct. 1998.

[18] D. R. Wilton *et al.*, "Potential integrals for uniform and linear source distributions on polygonal and polyhedral domains," *IEEE Trans. Antennas Propagat.*, vol. AP-32, pp. 276–281, Mar. 1984.

[19] L. Bers and F. Karal, *Calculus*. New York: Holt, Rinehart, and Winston, 1976.

[20] D. T. Kuan, J. C. Zamiska, and R. A. Brooks, "Natural decomposition of free space for path planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1985, pp. 168–173.

[21] L. E. Kavraki *et al.*, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 566–580, Aug. 1996.

Adaptive Control of a Class of Nonlinear Systems With a First-Order Parameterized Sugeno Fuzzy Approximator

Mohanad Alata, Chun-Yi Su, and Kudret Demirli

***Abstract*—In this paper, an adaptive fuzzy control scheme for tracking of a class of continuous-time plants is presented. A parameterized Sugeno fuzzy approximator is used to adaptively compensate for the plant nonlinearities. All parameters in the fuzzy approximator are tuned using a Luapunov-based design. In the fuzzy approximator, first-order Sugeno consequents are used in the IF–THEN rules of the fuzzy system, which has a better approximation capability than those using constant consequents. Global boundedness of the adaptive system is established. Finally, a simulation is used to demonstrate the effectiveness of the proposed controller.**

***Index Terms*—Adaptive control, fuzzy approximator, nonlinear systems, robustness, stability, Sugeno fuzzy systems.**

I. INTRODUCTION

The weakness of traditional quantitative techniques to adequately describe and control complex and ill-defined phenomena was summarized in the well known principle of incompatibility formulated by Zadeh [1]. This principle states that "as the complexity of a system increases, our ability to make precise and yet significant statements about its behaviors diminishes." The idea of fuzzy modeling first emerged in Zadeh [1], and has subsequently been pursued by many others. Although fuzzy modeling and control is thought of as an alternative approach compared with traditional control methods, its effectiveness is now well proven. Over the past two decades, engineers have applied fuzzy modeling and control methods very successfully [2]–[7].

Recently, in [11], [12], and [18]–[20] fuzzy controllers have been justified by universal approximation theorems. In other words, these fuzzy controllers are general enough to perform any nonlinear control action. Therefore, by carefully choosing the parameters of the fuzzy controller, it is always possible to design a fuzzy controller that is suitable for the nonlinear system under consideration. Based on this fact, a global stable adaptive fuzzy controller is firstly synthesized from a collection of fuzzy IF–THEN rules [10]. The fuzzy system, used to approximate an optimal controller, is adjusted by an adaptive law based on Luapunov synthesis approach. An adaptive tracking control architecture is proposed in [8] for a class of continuous time nonlinear dynamic systems, where an explicit linear parameterization of the uncertainty in the dynamics is not possible. The architecture employs fuzzy

Manuscript received May 28, 2000; revised August 30, 2001. This work was supported by the Natural Science and Engineering Research Council of Canada (NSERC) and the Canadian International development agency (CIDA). This paper was recommended by Associate Editor T. Sudkamp.

M. Alata is with the Mechanical Engineering Department, Jordan University of Science and Technology, Irbid, Jordan.

C.-Y. Su and K. Demirli are with the Department of Mechanical Engineering, Concordia University, Montreal, QC, H3G 1M8 Canada.

Publisher Item Identifier S 1094-6977(01)10035-0.