Application of Quadratic Algebraic Curve for 2D Collision-Free Path Planning and Path Space Construction

Ihn Namgung

Abstract: A new algorithm for planning a collision-free path based on an algebraic curve as well as the concept of path space is developed. Robot path planning has so far been concerned with generating a single collision-free path connecting two specified points in a given robot workspace with appropriate constraints. In this paper, a novel concept of path space (**PS**) is introduced. A **PS** is a set of points that represent a connection between two points in Euclidean metric space. A geometry mapping (**GM**) for the systematic construction of path space is also developed. A **GM** based on the 2nd order base curve, specifically Bézier curve of order two is investigated for the construction of **PS** and for collision-free path planning. The Bézier curve of order two consists of three vertices that are the start, **S**, the goal, **G**, and the middle vertex. The middle vertex is used to control the shape of the curve, and the origin of the local coordinate (ρ, θ) is set at the centre of **S** and **G**. The extreme locus of the base curve should cover the entire area of actual workspace (AWS). The area defined by the extreme locus of the path is defined as quadratic workspace (QWS). The interference of the path with obstacles creates images in the **PS**. The clear areas of the **PS** that are not mapped by obstacle images identify collision-free paths. Hence, the **PS** approach converts path planning in Euclidean space into a point selection problem in path space. This also makes it possible to impose additional constraints such as determining the shortest path or the safest path in the search of the collision-free path. The **QWS GM** algorithm is implemented on various computer systems. Simulations are carried out to measure performance of the algorithm and show the execution time in the range of 0.0008~0.0014 sec.

Keywords: Robot path planning, path space, collision-free path planning, geometry mapping, robot task planning.

1. INTRODUCTION

Determining a collision-free path or collision avoidance is the most fundamental area of research for the intelligent robot or autonomous robot. Path planning problems can be broadly categorized into two fields. One is sensor based collision avoidance and path generation and the other is an algorithmic approach for path planning, which assumes knowledge of robot surroundings by a vision system and other means. This paper belongs to the latter approach.

Numerous methods are proposed and demonstrated for the path planning problem. A rather extensive review of the algorithmic approach for path planning was reviewed in Namgung [1]. Up to now, the majority of methods have used some form of iteration; therefore their performance is unpredictable and they take a long time to complete. They are also unsuitable for real-time control systems and have an inability to easily control failure criteria. It is strongly desirable to find a method that does not employ an iterative method for finding a collision-free path.

Another aspect is that all path planning methods so far presented or published are concerned with producing only one path. However, theoretically, collision-free paths connecting two points consist of not just one, but infinite numbers. Hence from a mathematical point of view, a path connecting two points in Euclidean space belongs to any path space that can be constructed. All collision-free paths belong to a class of path space defined as a collision-free path space, which is a subset of the entire path space.

The introduction of the path space concept in robot path planning makes it possible to address additional constraints such as construction of the shortest path, the safest path or the minimum energy path from a particular class of path space. In order to formulate path space, a coordinate space concept must be used

Manuscript received May 20, 2003; revised December 15, 2003; accepted January 17, 2004. Recommended by Editorial Board member Wankyun Chung under the direction of Editor Keum-Shik Hong.

Ihn Namgung is with the Dae Jeon Intelligent Robot Center, 3F Jungkok, Daeduk College, 48 Jang-Dong, Yusong-gu, Daejon, 305-715, KOREA (e-mail: inamgung@dif.or.kr).

because it is the only way for the systematic construction of the path space. Namgung and Duffy [2] introduced the use of an algebraic curve for path planning and the construction of path space is a natural extension of this method. The next section reiterates the path space concepts from Namgung [1] to give the general idea of path space. This paper details how path space can be constructed with a simple curve and then used to find a collision-free path.

2. THEORY

2.1. Concepts of path space

Let **W** be the robot workspace in Euclidean space, and **S** and **G** be the start and goal points (i.e. **S**, **G** $\in$ **W**). **L** represents a path based on a class of algebraic curves that connect points **S** and **G**. Let $\mathbf{L_{free}}$ be a collision-free path and $\mathbf{L_{int}}$ be an obstacle interfering path. Now define an n-dimensional path space ($\mathbf{PS_n}$) that contains all possible paths based on the same class of algebraic curves as follows:

$$PS_n = \sum_{i=0}^{\infty} L_i^n = \sum_{i=0}^{\infty} L_{free\_i}^n + \sum_{i=0}^{\infty} L_{\text{int}\_i}^n, n = 1, \cdots, \infty \quad (1)$$

where $\mathbf{PS_n}$ is n-dimensional path space and n is the number of parameters that are used to define path L_i^n. For example, $\mathbf{PS_1} = \mathbf{L_{free\_0}} + \mathbf{L_{int\_0}}$ is a special case that consists of only one path, the direct path from **S** to **G**. $\mathbf{PS_2}$ is a 2-dimensional path space based on two independent parameters, which may be based on two linear curves or a single quadratic curve in 2-dimensions. $\mathbf{PS_3}$ is a 3-dimensional path space based on three independent parameters, which may be a cubic polynomial curve or a combination of a linear and a quadratic algebraic curve, etc.

Clearly, an n-dimensional path space ($\mathbf{PS_n}$) consists of collision-free path space ($\mathbf{PS_{n\text{-}free}}$) and obstacle interfering path space ($\mathbf{PS_{n\text{-}int}}$).

$$PS_n = PS_{n-free} + PS_{n-\text{int}} \quad (2)$$

The complete **PS**, denoted as **CPS**, is the entire path space that consists of all path space classes.

$$CPS = \sum_{n=1}^{\infty} PS_n \quad (3)$$

The relationships are graphically shown in Fig. 1.

The methods for object modeling or configuration space can be used in conjunction with path space construction [3-6]. With object modeling or configuration space, a robot can be modeled as a point, and a collision-free path connecting the start, **S**, to the goal, **G**, position can be generated using the method developed here. The advantage of path space concept is that it allows additional constraints such as the shortest path, the minimum time search path or the minimum energy path, etc. to be imposed in the search of a collision-free path. The additional constraints create a subset of the collision-free path, see Fig. 1.

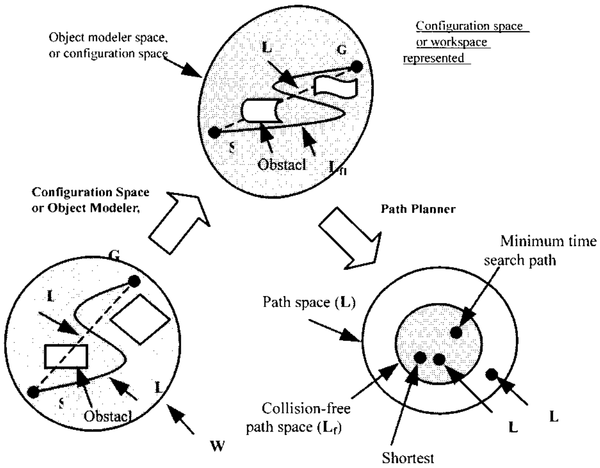


Fig. 1. Concept of PS and relationship with the robot workspace.

In this paper, one-dimensional path space, $\mathbf{PS_1}$, and two-dimensional path space based on linear algebraic curve, $\mathbf{PS_2}$, are investigated. Note that the collision-free path algorithm can be applied to mobile robots or the robot manipulators with appropriate configuration space or object modeling. In order to apply the algorithm to the robot manipulators, configuration space must be obtained first, and then the collision-free path algorithm can be applied to the configuration space. For the mobile robot, it is not necessary to construct the configuration space and one may practically proceed with the appropriate object modeling technique.

2.2. Algebraic curve for path planning

The problem of finding a collision-free path is here viewed as a geometric one requiring the solution to consist of a coordinated curve. The solution could be a connection of straight paths, a connection of straight and curved paths, a connection of curved paths, or a high-order polynomial curve. The collision-free path, therefore, can be expressed as a connection of algebraic curves in the Euclidean space. An algebraic curve in Euclidean space can be represented in implicit form (F(x, y) = 0), or in explicit form (y=f(x)), or in parametric form (x=f(s), y=g(t)). For some curves, for example closed curve or multi-valued curve, it is impossible to express the curve in explicit form. The explicit or implicit form of the curve may also suffer from having an infinite slope. On the other hand, parametric representation has many advantages over other forms, notably the inherent directional property it contains. The directional property simplifies the complexity of computation, and this is one reason that parametric curves/surfaces are widely used for

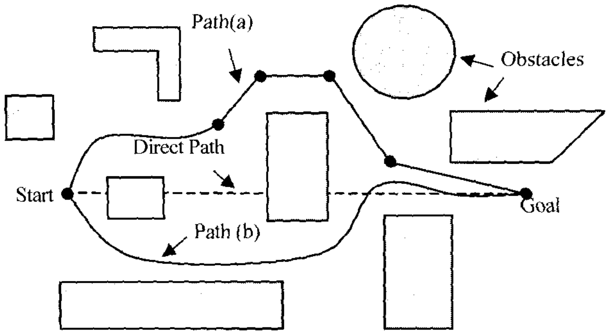


Fig. 2. Collision-free path in a 2D robot workspace environment.

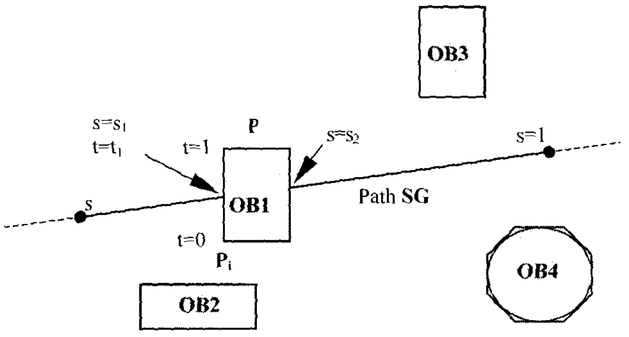


Fig. 3. Direct path and obstacle interference.

Computer Graphics and CAD/CAM [7, 8]. Namgung and Duffy [2, 9] introduced the use of parametric curves for robot path planning.

A collision-free path can be a series of connected line segments shown as path (a) in Fig. 2, or a high order polynomial curve shown as path (b) in Fig. 2. Whether the path is a connected line segment or a single polynomial curve, the shape of the curve must be controllable and the interference with any of the obstacles should be determined easily. It is clear that a collision-free path can be constructed by using algebraic curves.

For example, in Fig. 2 path (a), the intermediate connection point controls the shape of path (a) by changing their location. In addition to controlling the shape of the path, the defining parameters of connection points are used to create images of the obstacle by plotting the obstacle interference. The region of parameter space causing obstacle interference is the mapped image of the obstacle, and the empty space where the obstacle image is not mapped corresponds to the collision-free paths. The collision-free path defined here is a subset of **PS** that is defined by the specific base curves. The process of creating an obstacle image in parameter space is described in the following sections.

Another attractive feature concerning the use of an algebraic curve for path planning is that the complexity of a path can be controlled by limiting the number of control points or the order of the base curve. In other words, **PS** can be constructed with base curves of increasing complexity. Once a class of **PS** is filled with object images, it indicates that the refinement of **PS** or a search with additional connection points or higher order base curve is required.

3. PATH SPACE AND QUADRATIC WORK-SPACE DEFINITION

3.1. Direct path, $\mathbf{PS}_1$, and linear parametric curve

Here a direct path that belongs to a special class of path space, $\mathbf{PS}_1$, connecting two points, **S** (start point) and **G** (goal point) is represented by a parametric curve. The path in Fig. 3 is expressed as follows:

$$r(s) = (G - S)s + S\,. \tag{4}$$

S and **G** represent the start and the goal points. As the value of s increases from s= 0 to s=1, a point **r** moves from point **S** (=$\mathbf{r}(s)_{s=0}$), to point **G** (=$\mathbf{r}(s)_{s=1}$). It is noted that the directional property of parametric curve allows one to express the intersection of two line segments in simple terms. If they intersect, the resulting parameter values are within the defining range of two line segments. An intersection check between the polygonal edge of an obstacle and line segment **SG** should be carried out for all edges of the obstacle to determine if line segment **SG** intersects the object. In Fig. 3, two edges of obstacle OB1 intersect with line segment **SG** resulting in an interference range of s to be $s_1 < s < s_2$.

Calculation of intersection between the line segment **SG** and an edge $\mathbf{P}_i\mathbf{P}_j$ of an obstacle can be performed by equating the parametric equations for line **SG** and line $\mathbf{P}_i\mathbf{P}_j$, and solving for parameter s. Let **r**(s) represent line **SG** and **q**(t) represent an obstacle edge $\mathbf{P}_i\mathbf{P}_j$, then the intersection can be determined by the following equation.

$$(G - S)s + S = (P_j - P_i)t + P_i \tag{5}$$

From (5), the parameters s and t can be obtained by eliminating the other parameter. After substituting coordinate values for each point, the following equations for s and t are obtained.

$$s = \frac{(P_{ix} - S_x)(P_{jy} - P_{iy}) - (P_{iy} - S_y)(P_{jx} - P_{ix})}{(G_x - S_x)(P_{jy} - P_{iy}) - (G_y - S_y)(P_{jx} - P_{ix})} \tag{6}$$

$$t = \frac{(G_x - S_x)(S_y - P_{iy}) - (G_y - S_y)(S_x - P_{ix})}{(G_x - S_x)(P_{jy} - P_{iy}) - (G_y - S_y)(P_{jx} - P_{ix})} \tag{7}$$

The line segment **SG** intersects the edge $\mathbf{P}_i\mathbf{P}_j$ if $0 \le s \le 1$ and $0 \le t \le 1$, otherwise no intersections occur. There is an exception when the denominator vanishes

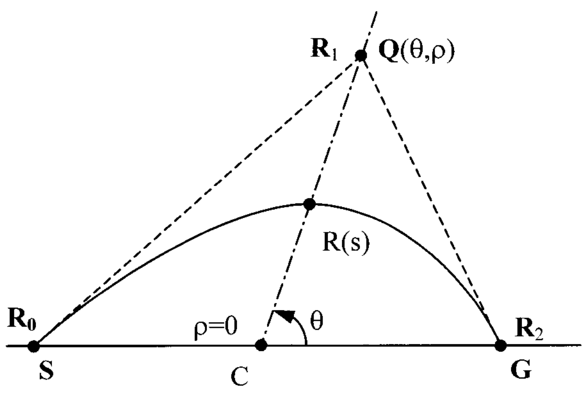


Fig. 4. Construction of quadratic curve (Bézier curve of order two).

from (6) and (7). This condition occurs when line segments **SG** and **P_iP_j** are in parallel. It can be proven that by setting the denominator of the above (6) or (7) as nil and rearranging it, the following equation can be obtained.

$$\frac{(G_x - S_x)}{(G_y - S_y)} = \frac{(P_{jx} - P_{ix})}{(P_{jy} - P_{iy})} \tag{8}$$

This is just the slope of line **SG** and the edge **P_iP_j**. Hence, calculation should proceed first with checking the ratio of denominator to numerator of (6) and (7) to prevent overflow. If both ratios are greater than one, i.e. numerator > denominator, calculation for s and t is not necessary since intersection between **SG** and **P_iP_j** does not occur. The calculation for the remaining edges of the obstacle can stop as soon as an intersection of an edge of an obstacle with **SG** is confirmed. This process of interference check should be carried out for all obstacles including actual workspace boundary.

Obstacles in the workspace can be categorized into 3 groups, obstacles intersecting line **SG**, obstacles above line **SG**, and obstacles below line **SG**. In case where there is no intersection between the line **SG** and an obstacle, as obstacle OB2 in Fig. 3, an additional calculation is necessary to determine on which side it is located. This does not require a precise value but requires a relative value for comparison purposes. One of the simplest yet fastest methods without involving trigonometric functions is to calculate the determinant given by (9).

$$A = \begin{vmatrix} S_x & G_x & P_{ix} \\ S_y & G_y & P_{iy} \\ 1 & 1 & 1 \end{vmatrix} \tag{9}$$

Note in fact that (9) gives twice the area of triangle **SGP$_i$**, and the order of vertex determines the sign of **A**. (9) indicates whether a vertex point, P$_i$, of an obstacle is located above, below, or on the line **SG**. For instance, all edges of an obstacle **OB2** from Fig. 3 do not interfere with line segment **SG**, and the location of **OB2** can be determined by calculating the sign of (9) after substituting any one of its vertex points for **P$_i$**. There are three cases that arise:

(i) **A** > 0, when **S**, **G**, and **P$_i$** are arranged in a anti-clock wise direction (**OB3** of Fig. 3),

(ii) **A** < 0, when **S**, **G**, and **P$_i$** are arranged in a clockwise direction (**OB2** of Fig. 3),

(iii) **A** = 0, when **S**, **G**, and **P$_i$** are collinear.

This calculation should be done for obstacles that do not interfere with the line **SG**. This categorization reduces calculation, in case when the direct path interferes with obstacles, which will become apparent in the next section.

3.2. Application of Bernstein-Bézier curve for path planning

In this paper, a quadratic curve that belongs to the Bernstein-Bézier curve family is used as a base curve of the path. Fig. 4 shows the Bernstein-Bézier curve of order 2, in which the start point, **S**, and the goal point, **G**, correspond to the end vertices of the curve. The remaining point controls the shape of the path that can be used to map the obstacle interference in **PS**. Path planning using a higher order curve requires a multiple number of control points, which lead to a higher dimensional space of **PS**.

A typical quadratic parametric curve defined by three vertices, one for the start point, **S**, another for the goal point, **G**, and the other for control point, **Q**, is the Bézier curve of order two. A general Bézier curve of order n, defined by n+1 vertices, can be expressed as follows:

$$\mathrm{R(s)} = \sum_{\mathrm{k=0}}^{\mathrm{n}} \frac{n!}{k!(n-k)!} s^k (1-s)^{n-k} R_k , \tag{10}$$

where **R$_k$** denotes the k-th vertex. Notice that the vertex numbering starts from 0 and ends with n. Bézier extended the idea of the approximation of a function to the approximation of a polygon, in which n+1 vertices of a polygon are approximated via the Bernstein basis. Hence it is also called a Bernstein-Bézier polynomial curve. When n=2, the Bézier curve is a parabola and it approximates a triangle formed by three vertices, see Fig. 4.

$$\begin{aligned} \mathrm{R(s)} &= (1\text{-s})^2 R_0 + 2s(1\text{-s})R_1 + s^2 R_2 \\ &= (R_0 - 2R_1 + R_2)s^2 + 2(R_1 - R_0)s + R_0 \end{aligned} \tag{11}$$

(11) can be derived geometrically. The valid interval of the parameter s is 0≤s≤1 where s=0 corresponds to point **R$_0$** and s=1 corresponds to point **R$_2$**. One should note that in Fig. 1 the lines **R$_0$R$_1$** and **R$_1$R$_2$** are tangents to the curve at points **R$_0$** and **R$_2$** respectively. The location of vertex **R$_1$** determines the

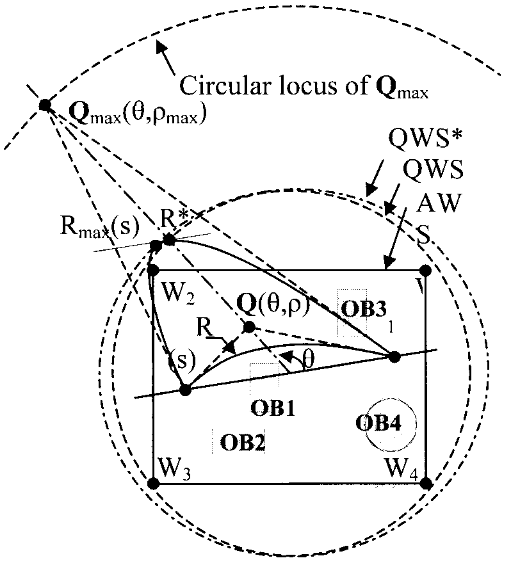


Fig. 5. Construction of quadratic workspace and bundle of parametric parabolas.

shape of the parabola.

3.3. Workspace determination for quadratic path

In this section, the formulation of the quadratic curve including the determination of parameter ranges is described. Fig. 5 shows a hypothetical workspace specified by a circular quadratic workspace (**QWS**) a centre, **C**, a start point, **S**, and a goal point, **G**.

$$R(s) = (S\text{-}2Q\text{+}G)s^2 + 2(Q-S)s + S \tag{12}$$

The control point **Q** is set in polar coordinates defined by two parameters, θ and ρ. The parameter ρ defines the radial distance of the control point from the centre and exhibits a bundle of parabolas along the line **CQ$_{max}$**. The parameter θ, which defines families of parabolas along a circumferential direction, sets the rotation angle of **CQ$_{max}$**. The curve **R(s)*** is a Bézier curve defined by triangle **SQ$_{max}$G** and indicates the extreme boundary of the work space **QWS**. The range of θ spans from 0° to 360°, and the range of ρ spans from 0 to ρ$_{max}$, where ρ$_{max}$ corresponds to the extreme control point, **Q$_{max}$**. Thus, a circular workspace in Euclidean space is mapped into a rectangular area in parameter space (θ, ρ). Because there are two independent parameters involved in the creation of **PS**, the resulting **PS** belongs to **PS$_2$**.

The extreme control point **Q$_{max}$** defines the boundary of the workspace **QWS**. When control point, **Q**, coincides with the centre point, **C**, the parabola degenerates to the line **SG**, and accordingly ρ is set to zero. The **QWS** shown by the circle in Fig. 5 should enclose the **AWS** (Actual Work Space). The **QWS** can be a shape other than a circle, however for the simplicity of calculation, a circular shape is chosen here. Any defining vertex of the **AWS** boundary

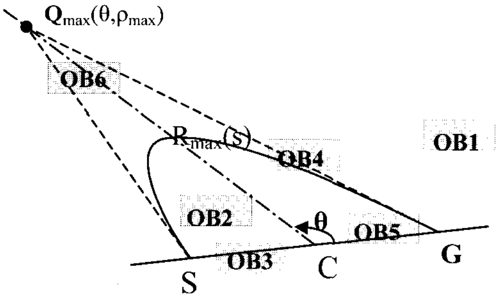


Fig. 6. Categorization of obstacles.

should be inside of the **QWS**. Calculation of ρ$_{max}$ of parabola, **Q$_{max}$**, is rather complicated since it requires calculating the contact of both the parabola and the circle (both curves are in second order). Here an approximation for circular **QWS** is used instead. Note from (12), when s=0.5 the parabola passes through bisection point of line segment **CQ$_{max}$** denoted by **R***. In other words, an approximation to the **Q$_{max}$** can be obtained from **R*** of **QWS**. This approximation creates the **QWS***, and that is slightly larger than that for the circular **QWS**, see Fig. 5.

The circular boundary **QWS*** defined by **R*** should contain the **AWS** boundary vertices. This is in fact finding the vertex of the **AWS** boundary that is farthest away from the centre, for example **W$_3$** of Fig. 5. **Q$_{max}$** can be obtained using (12) with s=0.5 and **R=R***.

$$Q_{\max} - C = 2\left(R^* - C\right), \tag{13}$$

where **C**=(**S**+**G**)/2. (13) indicates that the radius of the circle traced by **Q$_{max}$** is twice the radius of the circle generated by **R***.

So far the extreme radius of **QWS*** is discussed. Having calculated |**Q$_{max}$** -**C**|, the origin of the local coordinate system can be set at **C**, and the location of **Q$_{max}$** depending on the value of θ can be computed as follows:

$$\begin{aligned} Q_{\max,x} &= \frac{2\left|R^* - C\right|}{\left|G-S\right|}\left\{ \begin{matrix} (G_x - S_x)\cos\theta \\ -\left(G_y - S_y\right)\sin\theta + C_x \end{matrix} \right\} \\ Q_{\max,y} &= \frac{2\left|R^* - C\right|}{\left|G-S\right|}\left\{ \begin{matrix} \left(G_y - S_y\right)\cos\theta \\ -(G_x - S_x)\sin\theta + C_y \end{matrix} \right\} \end{aligned} \tag{14}$$

Now that the location of **Q$_{max}$** can be determined for a given value of θ, the bundle of parabolas can be set along the line **CQ$_{max}$**.

$$Q = (Q_{\max} - C)\rho + C \tag{15}$$

(15) can be inserted into (12) to get a parabolic locus that is defined by parameter s.

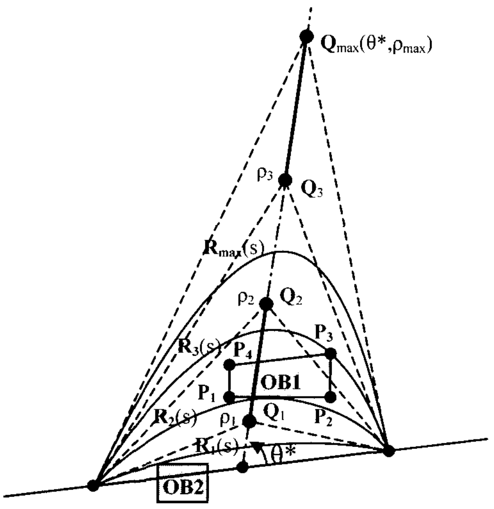


Fig. 7. Interference between a bundle of parabolas and obstacles.

4. CALCULATION OF INTERFERENCE BETWEEN PARABOL AND OBSTACLES

4.1. Categorization of obstacles

Actual calculation of obstacle interference can be delayed until categorization of the obstacle is done. Fig. 6 shows classification of obstacles relative to the triangle $\mathbf{SQ_{max}G}$. The reason for doing this is to reduce the calculations involved in the interference check.

From Fig. 6, the number of cases can be observed as follows.

i) Case 1, obstacle is located completely outside of triangle $\mathbf{SQ_{max}G}$, so no interference check is necessary.

ii) Case 2, obstacle is located completely inside of triangle $\mathbf{SQ_{max}G}$, which may result in two clear ranges of ρ.

iii) Case 3, obstacle intersects line segment **SG**, lower bound of clear range of ρ is obtained.

iv) Case 4, obstacle intersects line segment $\mathbf{SQ_{max}}$ or $\mathbf{Q_{max}G}$, upper bound of clear range of ρ is obtained.

v) Case 5, obstacle intersects both line segment **SG** and $\mathbf{SQ_{max}}$ or $\mathbf{Q_{max}G}$, therefore no interference check is necessary. In this case, a bundle of parabolas is completely blocked by the obstacle.

vi) Case 6, obstacle intersects both line segments $\mathbf{SQ_{max}}$ and $\mathbf{Q_{max}G}$, and the obstacle may be completely located outside of the parabola. There is still a chance that the obstacle may intersect the parabola, and the upper bound of the clear range of ρ is obtained.

The categorization can be done using line intersection between triangle $\mathbf{SQ_{max}G}$ and obstacle edges. Calculation of the intersection between the line segment **SG** and an edge $\mathbf{P_iP_j}$ of an obstacle can be done by equating the parametric equations for line **SG** and line $\mathbf{P_iP_j}$, and solving for parameters. The calculation is identical to the one presented in Section 3.1 and it can be used with (5), (6) and (7). The line segment **SG** intersects the edge $\mathbf{P_iP_j}$ if both parameter s and t are within $0 \le s \le 1$ and $0 \le t \le 1$, otherwise no intersection occurs. There is an exception when the denominator vanishes from (6) and (7). This condition occurs when line segments **SG** and $\mathbf{P_iP_j}$ are in parallel. (8) shows that the slope of the line **SG** and the edge $\mathbf{P_iP_j}$ are the same. Hence calculation should proceed with checking the ratio of the denominator to numerators of (6) and (7). If both the ratio are greater than one, i.e. numerator > denominator, calculations for s and t are not necessary since an intersection between **SG** and $\mathbf{P_iP_j}$ is impossible. The calculation for the remaining edges of the obstacle may stop as soon as the intersection of an edge of an obstacle with **SG** occurs. This interference checking process should be performed for all obstacles, and for line segments $\mathbf{SQ_{max}}$ and $\mathbf{Q_{max}G}$ to obtain a complete categorization of an obstacle.

4.2. Interference of obstacles and parabola

Fig. 7 shows an obstacle and a bundle of parabolas. The expression for a bundle of parabolas can be obtained by substituting (15) into (12).

$$R(\rho,s) = 2(C - Q_{\max})\rho s^2 + 2(Q_{\max} - C)\rho s \\ + (G - S)s + S \qquad (16)$$

It is clear that obstacles limit the valid range of parameter ρ, see Fig. 7. It is noted that two basic operations are necessary for the calculation of obstacle interference. One is a contact with a vertex of the obstacle and the other is a contact with an edge of the obstacle. With these basic operations, the interference range of parameter ρ can be obtained. For instance, in Fig. 7, vertex $\mathbf{P_3}$ of **OB1** is in contact with parabola $\mathbf{R_3(s)}$, and edge $\mathbf{P_1P_2}$ is in contact with parabola $\mathbf{R_2(s)}$. Clearly, the parabolas in the range of $\rho_2 \le \rho \le \rho_3$ intersect with obstacle **OB1**. Likewise, obstacle **OB2** creates an interference range of parameter $0 \le \rho \le \rho_1$. It is important to note that ρ_3 is the largest parameter value among the vertices and edge contact of **OB2**, and ρ_2 is the largest parameter value among the vertices and edge contact of **OB1**. Additionally, it must be decided from among which of the intervals $0 \le \rho \le \rho_1$, $\rho_1 \le \rho \le \rho_2$ and $\rho_2 \le \rho \le 1$, the parabolas intersect the obstacle. The categorization of obstacles treated above is used to clear the ambiguity of interference range.

The contact between the vertex of an obstacle and a parabola can be computed by substituting vertex point, $(\mathbf{P_x}, \mathbf{P_y})$, into the left-hand side of (16). From the two simultaneous equations, the parameter ρ can be eliminated and the remaining unknown parameter s

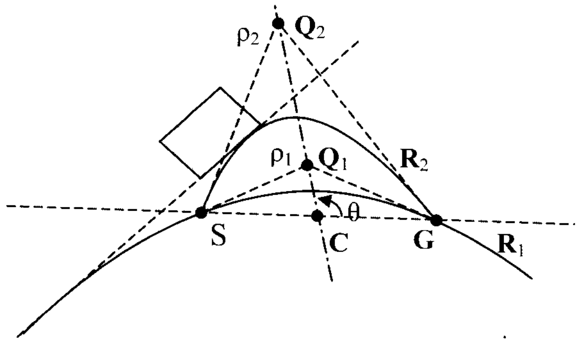


Fig. 8. Two possible cases of contact between a line and parabolas passing through S and G.

can be obtained as follows.

$$s = \frac{\left(C_x - Q_{\max,x}\right)\left(P_y - S_y\right) - \left(C_y - Q_{\max,y}\right)\left(P_x - S_x\right)}{\left(C_x - Q_{\max,x}\right)\left(G_y - S_y\right) - \left(C_y - Q_{\max,y}\right)\left(G_x - S_x\right)} \quad (17)$$

Parameter s represents a point along the parabola. The other parameter ρ can now be obtained by substituting (17) into (16) provided that the denominator of (17) does not vanish.

$$\rho = \frac{(P - S) + (S - G)s}{2s(1 - s)(Q_{\max} - C)} \quad (18)$$

The parameter ρ can be calculated by substituting either x or y values and the parameter s from (17). When the parameter s is either zero or one, an exception occurs from (18) indicating either indeterminate (s=0) or infinite (s=1). If s=0, the vertex coincides with the start point, **S**, and if s=1, the vertex coincides with the goal point, **G**. Both cases are invalid, and could not occur in reality. From (17) and (18), if both the solutions of s and ρ are within the range of 0<s<1 and 0≤ρ≤1, contact between vertex **P** and parabola **R(s)** occurs.

Fig. 8 shows the contact between a line and a bundle of parabolas where two parabolas can contact with the given line and at the same time pass through the start point and the goal point. This contact condition between an obstacle edge and a parabola is used to obtain the parametric values of t, for the edge, and ρ and s for the parabola from (19).

$$\begin{aligned}\left(P_j - P_i\right)t - P_i &= 2\left(C - Q_{\max}\right)\rho s^2 \\ &\quad + 2\left(Q_{\max} - C\right)\rho s + (G - S)s + S\end{aligned} \quad (19)$$

The parameter t describing the obstacle edge can be eliminated first from (19) to obtain the following (20).

$$A\rho s^2 + \left(-A\rho + B\right)s + C = 0, \quad (20)$$

where

$\mathrm{A} = 2(\mathbf{C}_x\text{-}\mathbf{Q}_{\max.x})(\mathbf{P}_{j,y}\text{-}\mathbf{P}_{i,y}) - 2(\mathbf{C}_y\text{-}\mathbf{Q}_{\max,y})(\mathbf{P}_{j,x}\text{-}\mathrm{P}_{i,x})$,
$\mathrm{B} = (\mathbf{G}_x\text{-}\mathbf{S}_x)(\mathbf{P}_{j,y}\text{-}\mathbf{P}_{i,y}) - (\mathbf{G}_y\text{-}\mathbf{S}_y)(\mathbf{P}_{j,x}\text{-}\mathbf{P}_{i,x})$, and
$\mathrm{C} = (\mathbf{S}_x+\mathbf{P}_{i,x})(\mathbf{P}_{j,y}\text{-}\mathbf{P}_{i,y}) - (\mathbf{S}_y+\mathbf{P}_{i,y})(\mathbf{P}_{j,x}\text{-}\mathbf{P}_{i,x})$.

There are two different roots of ρ that satisfy (20) and they are also shown graphically in Fig. 5. It is not readily recognizable from the equation; however it can be proved by using the fact that the parabola is just in contact with the edge of the obstacle defined by $\mathbf{P}_i$ and $\mathbf{P}_j$. Thus the discriminant of (20) should vanish.

$$A^2\rho^2 - 2A(B + 2C)\rho + B^2 = 0 \quad (21)$$

(21) may have two different solutions for ρ as shown in Fig. 5 in which $\mathbf{R}_1$ and $\mathbf{R}_2$ are the corresponding curves.

$$\begin{aligned}\rho_1 &= \frac{1}{A}\left\{B + 2C - 2\sqrt{(B + C)C}\right\} \\ \rho_2 &= \frac{1}{A}\left\{B + 2C + 2\sqrt{(B + C)C}\right\}\end{aligned} \quad (22)$$

Fig. 5 reveals that the root ρ_1 is always smaller than the root ρ_2. The root ρ_2 is the possible valid solution due to the fact that the parameter value s at contact point for ρ_1 is always outside the valid range, i.e. either s<0 or s>1.

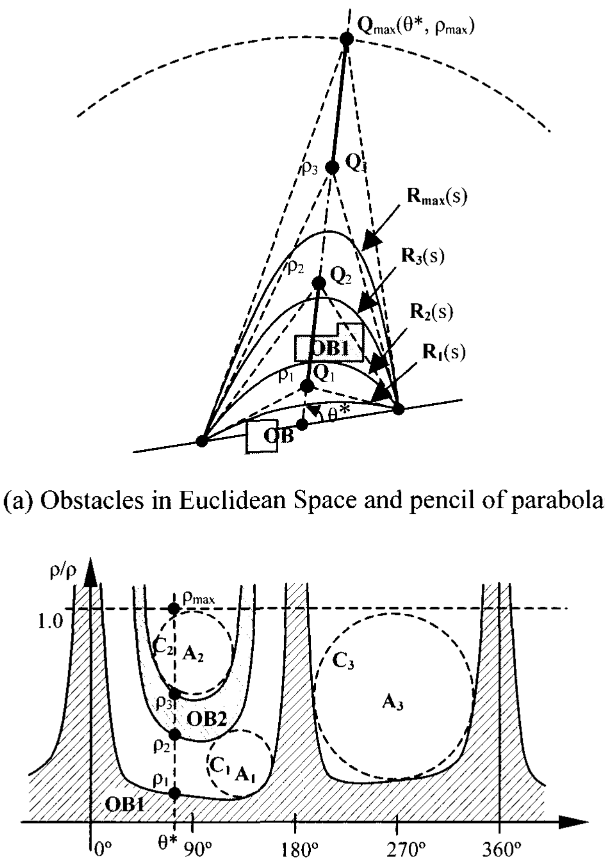
A number of exceptions can occur during the evaluation of ρ. From (22), if A=0 an exception occurs and it is the instance when the edge is in parallel with the line $\mathbf{CQ}_{\mathbf{max}}$. A simple and general method to avoid this exception is to compare the denominator with the numerator of (22) before evaluating the division. If the absolute value of the numerator is smaller than the absolute value of the denominator, ρ is in the range of -1≤ρ≤1 and the division should be evaluated, otherwise the evaluation can be omitted. Another exception occurs when the discriminant of (21) is negative, in which an imaginary solution transpires. This exception, (B+C)C<0, corresponds to either (B+C)<0 and C>0 or (B+C)>0 and C<0. This case arises when the line describing the edge under consideration intersects the line segment **SG**. Such cases can be avoided by computing the discriminant of (21). A negative discriminant value indicates no further evaluation is necessary and the calculation of the vertex contact given by (17) and (18) is necessary.

Substituting the valid result, ρ_2, from (22) into (21), the unknown parameter s is obtained as follows.

$$s = \frac{C + \sqrt{(B + C)C}}{B + 2C + 2\sqrt{(B + C)C}}, \quad (23)$$

where A,B and C are defined in (20). The parameter t can now be calculated from (19) if the denominator does not vanish. If the values of ρ, s and t are all within the valid range, i.e. 0≤ρ≤1, 0≤s≤1, and 0≤t≤1, a contact occurs.

The exception to (23) takes place when the edge is on the line **SG**, in which the coefficient B and C vanish and (23) becomes indeterminate. This exception results in either ρ_1=0 or ρ_2=0. For such



(a) Obstacles in Euclidean Space and pencil of parabolas.

(b) Images of obstacle in PS.

Fig. 9. QWS GM and construction of PS.

cases, the contact between vertices of the edge and the line segment **SG** is used to obtain the intersection point.

Obstacles in the workspace limit the range of parameter ρ. Since the calculation of the intersection between an obstacle and bundle of parabolas blocks the range of parameter ρ, the final blocked range of ρ is the union of all blocked ranges of ρ caused by all obstacles.

5. SIMULATION OF QWS GM ALGORITHM AND COMPARISON WITH OTHER GM

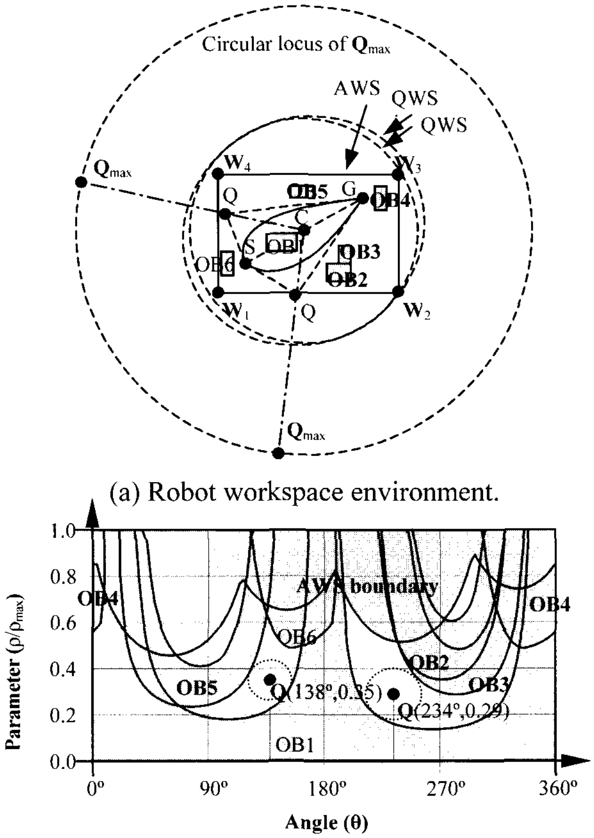
Fig. 9 shows a typical case of **QWS GM** in which the two obstacles **OB1** and **OB2** are mapped into **PS** images with different hatching patterns. The **QWS GM** described in the previous section does not impose any restrictions on the shape of an obstacle. For example, a concave obstacle such as **OB2** in Fig. 9 is a valid obstacle of geometry. The bundle of parabolas, **R_1(s)** through **R*(s)**, shown in Fig. 6(a) corresponds to points ρ_1 through ρ_{max}, respectively, on the dotted line indicated by θ*. The resulting interference ranges of ρ for obstacles **OB1** and **OB2** are $0 \leq \rho \leq \rho_1$ and $\rho_2 \leq \rho \leq \rho_3$, respectively. Thus, the clear ranges of ρ are $\rho_1 \leq \rho \leq \rho_2$ and $\rho_3 \leq \rho \leq 1$. The obstacle **OB2** maps into a horseshoe shaped image, while **OB1** maps into an image occupying the bottom part around θ=0° and the 360° area.

The number of obstacles that intersect with parabolas varies as the angular parameter of the control point of the parabola changes. In the implementation of the **QWS GM** algorithm, the routine for finding the valid range of ρ should cope with any number of obstacles. Such a process creates images in the path space as shown in Fig. 9(b). In the Appendix, a procedural description of the **QWS GM** algorithm in which the details of categorization of the obstacle, the interference computation and the process flow can be observed.

With some additional constraints, the entire geometric mapping for $0 \leq \theta \leq 360$ may be unnecessary. For instance, the values of θ near 0° and 180° produce sharp parabolas passing nearby line segment **SG**, and the range of angular parameters can be set up so that those intervals can be skipped. One of the most important benefits of constructing **PS** is that it allows finding the safest path. In Fig. 9(b), there are three areas, A_1, A_2, andA_3, that are not occupied by obstacle images. The centre of the inscribing circles, C_1, C_2, and C_3, produces a safer path than other points from that area. Because circle C_3 is the largest, the centre point of C_3 is the safest path. Since the path space is not a metric space, the safest path considered so far may not correspond to the true safest path. However, it would give an approximation to the true safest path. A rather simple way of finding the pseudo safest path is as follows. The clear ranges of ρ for each different value of θ are compared to find the one that has the widest range. The mid-point of that parameter range is selected as the pseudo safest path. This is only an approximation to the safest path, since the comparison is made in the θ -direction only.

A number of **GM** simulations were performed and shown in Figs. 10~15. Adding an obstacle increases run time, and the run time varies as the location of the start or the goal point changes because the obstacles interference calculation demands a varying degree of computational load. The simulations are carried out on three different systems; i) Windows 2000 on PentiumlV/2.53GHz with 1GB DDR-SDRAM system, ii) Windows ME on PentiumlII (Celeron)/533MHz with 256SDRAM system, and iii) Win 95 DOS Mode on Pentium/90MHz with 16MB SDRAM system with the **GM** resolution of Δθ = 3° and 15°, and results of Δθ = 3° are presented in Figs. 10~15. The measure of pure computational time for Figs. 10~15 are given in Table 1. The computational time for Δθ = 3° is not exactly five times that of Δθ = 15°because obstacle categorization and corresponding interference calculations do not increase proportionally.

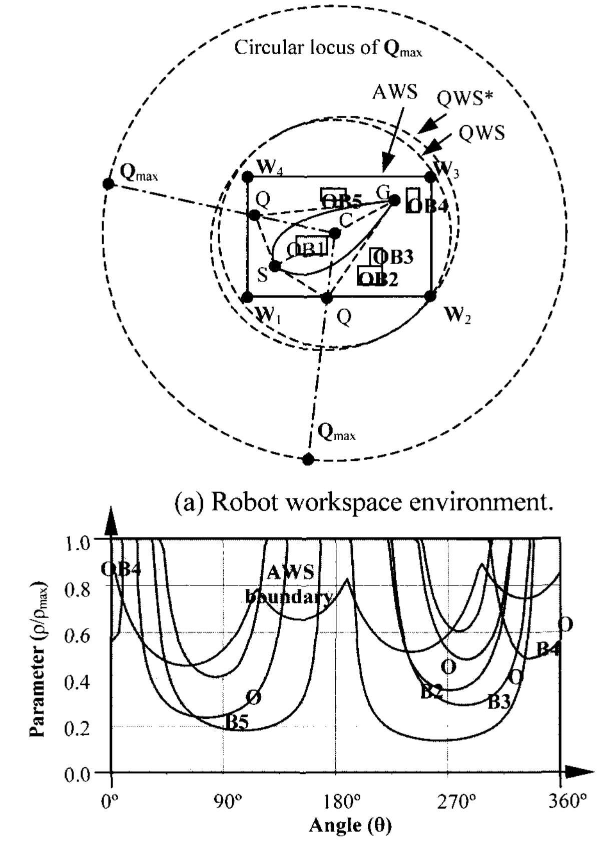
Case 1 is shown in Fig. 10 and contains 6 obstacles, where the upper path and lower path are selected from



(a) Robot workspace environment.

(b) QWS geometry mapping of obstacles.

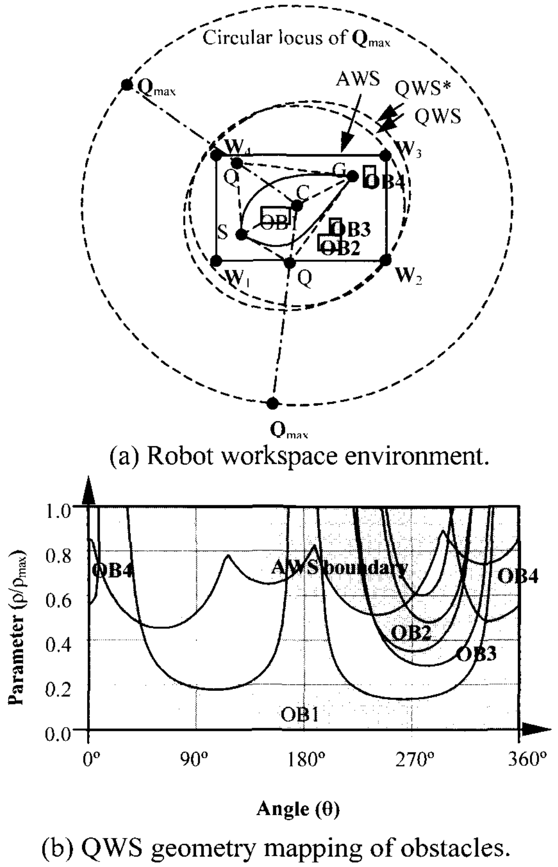
Fig. 10. Simulation of GM - case 1.



(a) Robot workspace environment.

(b) QWS geometry mapping of obstacles.

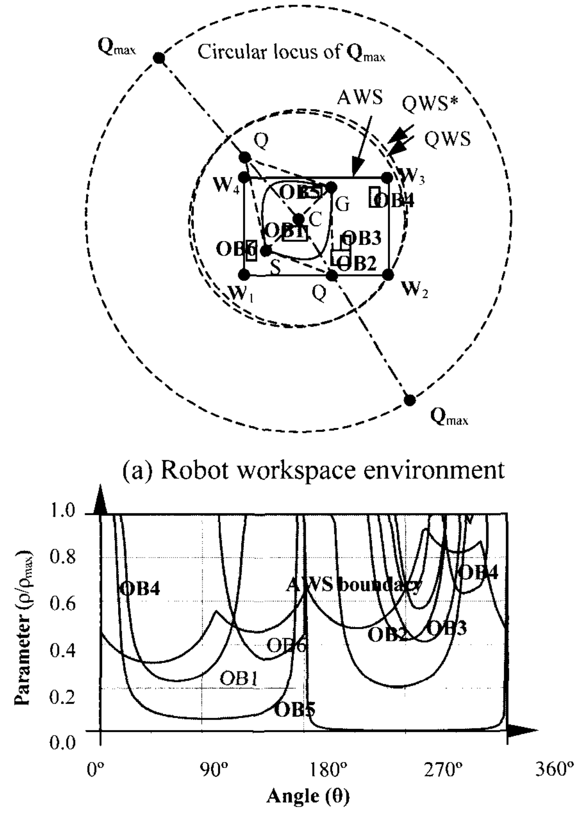
Fig. 12. Simulation of GM - case 3.



(a) Robot workspace environment.

(b) QWS geometry mapping of obstacles.

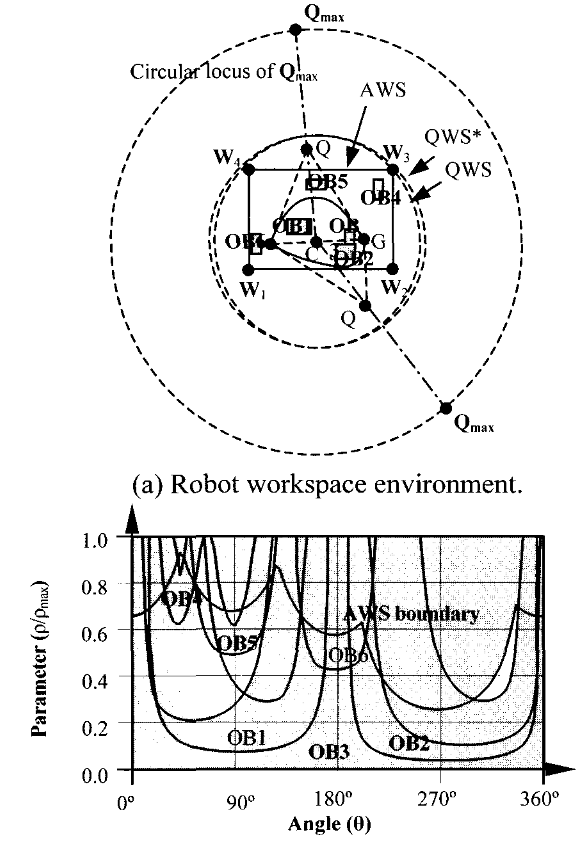
Fig. 11. Simulation of GM - case 2.



(a) Robot workspace environment

(b) QWS geometry mapping of obstacles.

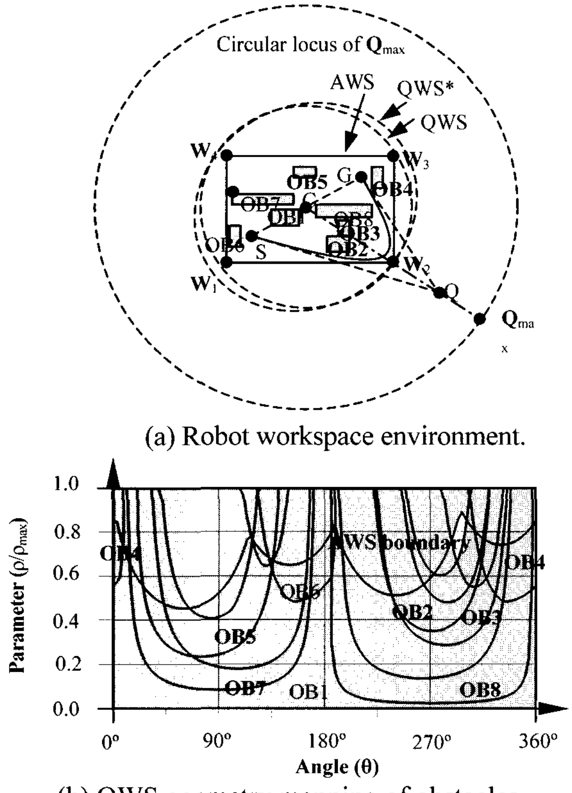
Fig. 13. Simulation of GM - case 4.



(a) Robot workspace environment.

(b) QWS geometry mapping of obstacles.

Fig. 14. Simulation of GM - case 5.



(a) Robot workspace environment.

(b) QWS geometry mapping of obstacles.

Fig. 15. Simulation of GM - case 6.

Table 1. Summary of QWS GM simulation results.

(unit: sec.)

Ref. Fig.	PentiumIV/2.53GHz system		PentiumIII (Celeron)/533 MHz system		Pentium/90MHz system	
	Δθ = 3°	Δθ= 15°	Δθ= 3°	Δθ= 15°	Δθ = 3°	Δθ= 15°
Fig. 10 (case 1)	0.001156	0.000250	0.00412	0.00088	0.120879	0.021978
Fig. 11 (case 2)	0.000812	0.000187	0.00297	0.00060	0.087363	0.016484
Fig. 12 (case 3)	0.000968	0.000218	0.00346	0.00077	0.102143	0.020879
Fig. 13 (case 4)	0.001234	0.000266	0.00440	0.00094	0.131868	0.026923
Fig. 14 (case 5)	0.001125	0.000250	0.00401	0.00087	0.119011	0.021978
Fig. 15 (case 6)	0.001485	0.000312	0.00527	0.00110	0.157033	0.032363

the centre point of the circle in **FS**. Case 2 contains 4 obstacles, and case 3 contains 5 obstacles in which **OB5** is added to case 2. Cases 4 and 5 contain 6 obstacles in which **OB6** is added to case 3, and are exactly the same as that of case 1 except that the goal point location is moved to another place. Case 6 contains 8 obstacles in which **OB7** and **OB8** are added to case 1.

6. CONCLUSION AND DISCUSSIONS

The path planning method introduced in this paper is an attempt to develop an alternative path planning strategy to overcome the shortcomings of many of the current path planning methods. The development of the path space concept and geometry mapping in this paper is new in the field of robot path planning research. There are an infinite number of paths connecting the start to the goal, and they are defined to be a path space. The path space consists of path space that does not interfere with obstacles and path space that does interfere with obstacles. The generalized path space is the summation of all path space in which each path space is based on a different base curve. Hence, the generalized path space is a hierarchical structure that consists of path spaces based on a simple curve to a complicated curve. In this series of papers, two-dimensional path space, **PS**$_2$, was investigated.

In this paper, the concept of **PS** is established and a quadratic workspace geometry mapping (**QWS GM**) was developed. The **QWS GM** was implemented for the simulation of the same robot workspace environment. The categorization of obstacles and the interference calculations are all expressed in closed form equations and no iterative calculations are involved. The simulation confirmed the effectiveness of the method.

Although the method introduced in this paper is efficient for the scattered obstacle environment, it is not good at solving the path-planning problem for the maze-like environment. For such environment, a higher-order based curve is required in the form of a single high order algebraic curve or iterative application of low order curve. Application of single high-order based curve for geometry mapping may not be desirable, because it will need an iterative solution for the contact and intersection calculation. Although those curves will give higher continuity in the resulting path, C^2 continuity may be enough for most applications. Hence, iteratively applying low order curve for the given interval will be an effective and simple method for the problem. In this regard, the **GMs** developed in this paper are fundamental in that they can be used in sub-intervals of the original interval.

The path space concept converts the problem of collision-free path planning in Euclidean space into the selection of a point in path space. Selecting a point from a given space is an inherently simpler problem than creating a path in a space. The path space also provides information on free space. It can be used for the safer path generation and can also be used for post processing of the final path to give proper continuity, etc.

Further investigation and development in the generalization of the method for higher order path space is the next research step. Especially in the case of the higher dimensional path space, the post processing of final path for proper continuity is yet another area of research that requires attention. And the integration of the **GM** algorithm with the configuration space or the object modeling is another area of research needed to be performed for the generalization of the method. Once these fundamental aspects are established, the use of the **GM** method for path planning and obstacle avoidance in a dynamic environment may require further research.

REFERENCES

[1] I. Namgung, "Path space approach for planning 2d shortest path based on elliptic workspace geometry mapping," *KSME international Journal*, vol. 18, no. 1, pp.92-105, 2004.

[2] I. Namgung and J. Duffy, "Two dimensional collision-free path planning using linear parametric curve," *Journal of Robotic Systems*, vol. 14, no. 9, pp. 659-673, 1997.

[3] E. J. Bernabeu and J. Tornero, "Optimal geometric modeler for robot motion planning," *Journal of Robotic Systems*, vol. 17, no. 11, pp. 593-608, 2000.

[4] P. Moutarlier, B. Mirtich, and J. Canny, "Shortest paths for a car-like robot to manifolds in configuration space," *International Journal of Robotics Research*, vol. 15, no. 1, pp. 36-60, 1996.

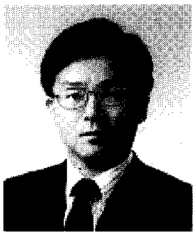
[5] K. M. Lynch, N. Shiroma, H. Arai, and K. Tanie, "Collision-free trajectory planning for a 3-dof robot with a passive joint," *International Journal of Robotics Research*, vol. 19, no. 12, pp. 1171-1184, 2000,

[6] T. Lozano-Pérez, "Automatic planning of manipulator transfer movements," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-11, no. 10, pp. 681-698, 1981.

[7] P. Bèzier, *Numerical Control-Mathematics and Application*, Translated by A.R. Forrest and A.F. Pankhurst, John Wiley & Sons, New York, 1972.

[8] D. F. Rogers and J. A. Adams, *Mathematical Elements for Computer Graphics, 2nd Ed.*, McGraw-Hill, New York, 1990.

[9] I. Namgung, "Planning collision-free paths with applications to robot manipulators," (Ph.D. Dissertation), University of Florida, 1989.



Ihn Namgung received the Ph.D. degree in Mechanical Engineering from the University of Florida in 1989. He is currently with the Daejeon Hightech Industry Promotion Foundation as the Director of the Intelligent Robot Center. His research interests include autonomous robot control, robot path planning, obstacle avoidance and hazardous environment robotics.