International Journal of Advanced Robotic Systems

OPEN ACCESS ARTICLE

Curvature-continuous 3D Path-planning Using QPMI Method

Regular Paper

Seong-Ryong Chang[1] and Uk-Youl Huh[1]*

1 Inha University, In-cheon, Republic of Korea
*Corresponding author(s) E-mail: uyhuh@inha.ac.kr

Received 17 July 2014; Accepted 26 April 2015

DOI: 10.5772/60718

Abstract

It is impossible to achieve vertex movement and rapid velocity control in aerial robots and aerial vehicles because of momentum from the air. A continuous-curvature path ensures such robots and vehicles can fly with stable and continuous movements. General continuous path-planning methods use spline interpolation, for example B-spline and Bézier curves. However, these methods cannot be directly applied to continuous path planning in a 3D space. These methods use a subset of the waypoints to decide curvature and some waypoints are not included in the planned path. This paper proposes a method for constructing a curvature-continuous path in 3D space that includes every waypoint. The movements in each axis, x, y and z, are separated by the parameter u. Waypoint groups are formed, each with its own continuous path derived using quadratic polynomial interpolation. The membership function then combines each continuous path into one continuous path. The continuity of the path is verified and the curvature-continuous path is produced using the proposed method.

Keywords Aircraft navigation, Continuous path, Interpolation, Mobile robot, Path planning, Robot motion, Path-smoothing algorithm, Unmanned aerial vehicles

1. Introduction

Path searching is one of the main topics in path-planning research. The primary purpose of path planning is to construct a collision-free path. Commonly used algorithms for path planning include the D* algorithm [1], potential field algorithm [2], Probabilistic Roadmap Methods (PRM) algorithm [3], and Rapidly exploring Random Trees (RRT) [4]. These algorithms focus on path planning on a 2D plane. For path planning in a 3D space, researchers have proposed algorithms to create various optimal or sub-optimal paths [5] [6]. These algorithms provide a method of creating paths using waypoints for the 3D movement of aerial vehicles and aerial robots [7].

Many forces, such as gravitational, centrifugal, centripetal, and inertia, may obstruct stable movement in a real physical environment. These forces have a greater effect on movements in a 3D space compared to movements on a 2D plane. On a 2D plane, the movements obtain momentum from the ground. Therefore, as long as slippage does not occur or the path is continuous, the effects of these forces can be neglected. On the other hand, in a 3D space the movement of aerial robots or aerial vehicles is unable to obtain momentum from the ground. Instead, momentum is gained from the air; for this, a fixed-wing aircraft must have a larger velocity than stalling velocity [8]. If the velocity drops below the stalling velocity, the fixed-wing aircraft cannot obtain the lift force and will stall. Therefore, the fixed-wing aircraft must fly above the stalling velocity and cannot have any sharp angular movements or stop in the 3D space. A rotary-wing aircraft can hover in 3D space but cannot have any sharp angular movement or rapidly stop. These conditions imply that the movements of aircraft in 3D space should not have vertexes and that the velocity and acceleration should be continuous and maintained above the stalling velocity.

Paths with continuous velocity and acceleration, also referred to as smooth paths or continuous paths, are classified in terms of continuity, G^0, G^1, G^2, $\cdots$, G^n [9]. A G^0 continuous path is constructed by forming a straight line between the waypoints. PRM [3] produces the G^0 continuous path. A G^1 continuous path matches the first-order differential values at each of the waypoints. This path provides a continuous velocity and a common tangential direction at the waypoint. A G^2 continuous path has the same second-order differential values and also shares a common centre of curvature. This path has a continuous acceleration, and it is called a curvature- continuous path. A G^n continuous path is defined as a continuity of the n^{th} -order differential values. The path must have G^2 continuity to guarantee smooth movements.

Many studies focus on movements in 2D space and utilize spline-based methods to construct the continuous path. Magid et al. proposed a robot navigation algorithm based on spline [10]. Komoriya et al. proposed trajectory design and control of a wheel-type mobile robot using B-spline [11]. Piazzi et al. demonstrated the use of η^3 splines in smooth-path generation for wheeled mobile robots [12]. Berglund et al. explored an obstacle-avoiding minimum variation b-spline problem [13]. Anderson et al. studied real-time trajectory smoothing for unmanned air vehicles in a 2D plane [14]. Jung and Tsiotras applied b-spline to generate a path for small unmanned vehicles in a 2D plane [15].

Various algorithms have also been applied for constructing continuous paths in 3D space. Hasircioglu et al. showed 3D path planning for the navigation of UAV by applying evolutionary algorithms using B-spline [5]. Yang et al. demonstrated a 3D path-smoothing algorithm using cubic Bézier spiral curves [16]. Further, Babaei and Mortazavi described 3D curvature-constrained trajectory planning based on in-flight waypoints [6].

Most 2D and 3D continuous path-planning methods use spline interpolation, for example B-spline and Bézier spiral curves. However, such methods are not applicable in path-smoothing methods because some waypoints are not part of the smoothed path. Instead, control points should be used to decide curvature in the path.

This paper expands the quadratic polynomial interpolation (QPMI) path-smoothing method [17] [18], used for path smoothing in a 2D plane, to connect waypoints and form continuous curves in a 3D space.

The rest of the paper is organized as follows: section 2 explains the path-smoothing method using polynomial interpolation. In section 3, the QPMI path-smoothing method is expanded to apply to the 3D space. Section 4 proposes the membership function for the QPMI path-smoothing method in a 3D space. Section 5 discusses continuity and curvature in the proposed method. Section 6 describes a series of 3D path-smoothing simulations using the proposed method. Section 7 provides conclusions.

2. Path-smoothing method using polynomial interpolation

In the case of aerial robots or aerial vehicles polynomial interpolation cannot be applied directly, because in a 3D space the path involves a complex non-linear trajectory where each waypoint has a visit sequence. The planned path must visit every waypoint in sequential order. These conditions make it difficult to apply polynomial interpolation directly to 3D paths. Other spline-interpolation-based methods produce paths that do not visit all waypoints. Instead, control points are either used based on a subset of waypoints or determined using another method. In this case, the planned smooth path may be a shorter path, but it is not a unique path through the waypoints. Essentially, the path should depend on the waypoints only, and visit all waypoints.

General polynomial interpolation involves constructing a polynomial curve that includes every waypoint. Runge's phenomenon [9] occurs when using high-order polynomials. QPMI [17] avoids Runge's phenomenon by using the quadratic polynomial, which is the lowest-order polynomial connecting three waypoints with G^2 continuity. In addition, it has a unique solution, a shortest path, and can obtain variations in curvature. These features provide high reliability and require only low-performance hardware. However, in order to create a shortest path, the quadratic polynomial should have a rotated form. QPMI [17] can express a 3D path in the 3D space by improving the parameter u, which denotes the distance of the linear path between the neighbour waypoints. This parameter separates the x, y, and z values from the 3D axes. The values and the differential values of each axis can then be obtained. These values provide the variation of curvature and the continuity.

This paper expands the QPMI path-smoothing method [17] and demonstrates how the proposed method can create a smooth path by combining quadratic polynomials sequentially in the 3D space.

3. Continuous path creation using quadratic polynomials

3.1 Smooth path connecting three waypoints using quadratic polynomial

In QPMI [17], the path connecting three waypoints can be expressed using a quadratic polynomial. Equation (1) is the quadratic polynomial for presenting a path on the 2D plane.

$$\mathbf{P}:(x,y)\in R^2$$

$$y=ax^2+bx+c \tag{1}$$

The quadratic polynomial connecting the three waypoints (P_1:(x_1, y_1), P_2:(x_2, y_2), and P_3:(x_3, y_3)) can be obtained using the following method:

$$\begin{gathered}\begin{pmatrix}y_1\\y_2\\y_3\end{pmatrix}=\begin{pmatrix}x_1^2 & x_1 & 1\\x_2^2 & x_2 & 1\\x_3^2 & x_3 & 1\end{pmatrix}\cdot\begin{pmatrix}a\\b\\c\end{pmatrix}\\ \begin{pmatrix}a\\b\\c\end{pmatrix}=\begin{pmatrix}x_1^2 & x_1 & 1\\x_2^2 & x_2 & 1\\x_3^2 & x_3 & 1\end{pmatrix}^{-1}\cdot\begin{pmatrix}y_1\\y_2\\y_3\end{pmatrix}\end{gathered} \tag{2}$$

Equation (2) is an inverse matrix for obtaining the quadratic polynomial. This is the general quadratic polynomial interpolation for connecting three waypoints, which constructs the shortest G^2 continuous path.

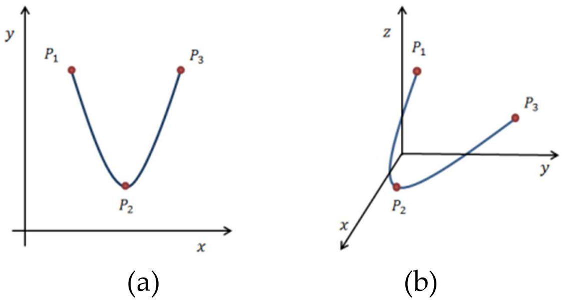


Figure 1. (a) Quadratic polynomial graph on the 2D plane. (b) Quadratic polynomial connecting three waypoints in the 3D space.

Figure 1 illustrates the quadratic polynomial graph on the 2D plane (a), and, similarly, in the 3D space (b). The waypoints in the 3D space are expressed by equation (3).

$$P_n:(x_n,y_n,z_n)\in R^3,\ n=1,2,\cdots,m \tag{3}$$

The aim of this paper is to construct and express smooth paths using quadratic polynomials in the 3D space. To apply this concept to the 3D space, a parametric method is used to express the 3D path.

3.2 Parametric method for 3D path

The parametric method provides a convenient description of the path. The parametric path in the 3D space can be expressed as

$$P(u):(x(u),y(u),z(u)) \tag{4}$$

The parameter u is the linear distance and can be calculated from the linear piecewise path. In the 3D space, there are three waypoints: P_1:(x_1, y_1, z_1), P_2:(x_2, y_2, z_2), and P_3:(x_3, y_3, z_3). The parameter u can then be obtained as:

$$\begin{gathered}u_1=0\\u_2=u_1+\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}\\u_3=u_2+\sqrt{(x_3-x_2)^2+(y_3-y_2)^2+(z_3-z_2)^2}\end{gathered} \tag{5}$$

The quadratic polynomials of the 3D path are as follows:

$$x(u)=a_xu^2+b_xu+c_x \tag{6}$$

$$y(u)=a_yu^2+b_yu+c_y \tag{7}$$

$$z(u)=a_zu^2+b_zu+c_z \tag{8}$$

From equations (6), (7), and (8), we obtain the following inverse matrix calculations:

$$\begin{pmatrix}a_x\\b_x\\c_x\end{pmatrix}=\begin{pmatrix}u_1^2 & u_1 & 1\\u_2^2 & u_2 & 1\\u_3^2 & u_3 & 1\end{pmatrix}^{-1}\cdot\begin{pmatrix}x(u_1)\\x(u_2)\\x(u_3)\end{pmatrix} \tag{9}$$

$$\begin{pmatrix}a_y\\b_y\\c_y\end{pmatrix}=\begin{pmatrix}u_1^2 & u_1 & 1\\u_2^2 & u_2 & 1\\u_3^2 & u_3 & 1\end{pmatrix}^{-1}\cdot\begin{pmatrix}y(u_1)\\y(u_2)\\y(u_3)\end{pmatrix} \tag{10}$$

$$\begin{pmatrix}a_z\\b_z\\c_z\end{pmatrix}=\begin{pmatrix}u_1^2 & u_1 & 1\\u_2^2 & u_2 & 1\\u_3^2 & u_3 & 1\end{pmatrix}^{-1}\cdot\begin{pmatrix}z(u_1)\\z(u_2)\\z(u_3)\end{pmatrix} \tag{11}$$

These equations express the G^2 continuous path in the 3D space using the parametric method. The path has a parabolic curve and the plane of this curve is defined as a triangle made by three waypoints in the 3D space. If variations of each axis are G^2 continuous, the combined path has G^2 continuity.

Equation (5) can be generalized as:

$$\begin{gathered} u_1 = 0 \\ u_n = \sum_{n=2}^{m} \sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2 + (z_n - z_{n-1})^2} \end{gathered} \tag{12}$$

In addition, equations (9), (10), and (11) can be described as follows:

$$\begin{pmatrix} a_{x_{n-1}} \\ b_{x_n} \\ c_{x_{n+1}} \end{pmatrix} = \begin{pmatrix} u_{n-1}{}^2 & u_{n-1} & 1 \\ u_n{}^2 & u_n & 1 \\ u_{n+1}{}^2 & u_{n+1} & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} x_n(u_{n-1}) \\ x_n(u_n) \\ x_n(u_{n+1}) \end{pmatrix} \tag{13}$$

$$\begin{pmatrix} a_{y_{n-1}} \\ b_{y_n} \\ c_{y_{n+1}} \end{pmatrix} = \begin{pmatrix} u_{n-1}{}^2 & u_{n-1} & 1 \\ u_n{}^2 & u_n & 1 \\ u_{n+1}{}^2 & u_{n+1} & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} y_n(u_{n-1}) \\ y_n(u_n) \\ y_n(u_{n+1}) \end{pmatrix} \tag{14}$$

$$\begin{pmatrix} a_{z_{n-1}} \\ b_{z_n} \\ c_{z_{n+1}} \end{pmatrix} = \begin{pmatrix} u_{n-1}{}^2 & u_{n-1} & 1 \\ u_n{}^2 & u_n & 1 \\ u_{n+1}{}^2 & u_{n+1} & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} z_n(u_{n-1}) \\ z_n(u_n) \\ z_n(u_{n+1}) \end{pmatrix} \tag{15}$$

$$(2 \le n \le m-1)$$

Equations (13), (14), and (15) are the generalized equations of (9), (10), and (11). Path (4) can be changed as follows:

$$\boldsymbol{P}(u) = \begin{cases} \mathrm{P}_2 : (x_2(u),\ y_2(u), z_2(u)),\ (u_1 \le u_2 \le u_3) \\ \mathrm{P}_3 : (x_3(u),\ y_3(u), z_3(u)),\ (u_2 \le u_3 \le u_4) \\ \vdots \\ \mathrm{P}_n : (x_{m-1}(u),\ y_{m-1}(u), z_{m-1}(u)),\ (u_{m-2} \le u_{m-1} \le u_m) \end{cases} \tag{16}$$

The m^{th} waypoints require $m-2$ polynomials at each axis. In the case of four or more waypoints, the general polynomial interpolation requires an increasing polynomial degree. For four waypoints, two quadratic polynomial paths are available. In this paper, the waypoints are grouped. The first group consists of P_1, P_2, and P_3. The first parametric quadratic polynomials $P_2 : (x_2(u),\ y_2(u),\ z_2(u))$ can be obtained by using equations (13), (14), and (15). This path is the G^2 continuous path connecting P_1, P_2, and P_3. The second group consists of P_2, P_3, and P_4 and the second parametric quadratic polynomials $P_3 : (x_3(u),\ y_3(u),\ z_3(u))$ are obtained similarly. These two quadratic polynomial groups overlap between P_2 and P_3. The two paths are then combined using a membership function at the overlapping region.

4. QPMI 3D path-smoothing method

The membership function applied uses a triangular function to distribute the degree of truth in the fuzzy control theory.

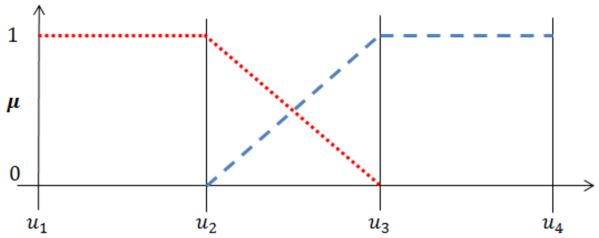


Figure 2. Triangular membership functions. Red dotted line is a variation of $P_2(u)$ degree. Blue dashed line is a variation of $P_3(u)$ degree.

Figure 2 shows variations of the membership functions in the case of four waypoints. The overlapped section P_2 to P_3 is expressed by the parameters u_2 and u_3. The degree of two waypoints varies by parameter u. The red dotted line decreases in the section from u_2 to u_3. In this section, the degree of $P_2(u)$ decreases. The blue dashed line increases in the section from u_2 to u_3. The degree of $P_3(u)$ is increasing in this section. The group of paths is defined as

$$\boldsymbol{P} : (X(u), Y(u), Z(u)) \tag{17}$$

$X(u)$, $Y(u)$ and $Z(u)$ are described as follows:

$$X(u) = \begin{cases} \mu_{r1}(u)x_2(u) + \mu_{f1}(u)x_1(u) : u_1 \le u \le u_2 \\ \mu_{r2}(u)x_3(u) + \mu_{f2}(u)x_2(u) : u_2 \le u \le u_3 \\ \mu_{r3}(u)x_4(u) + \mu_{f3}(u)x_3(u) : u_3 \le u \le u_4 \\ \vdots \\ \mu_{rm-2}(u)x_{m-1}(u) + \mu_{fm-2}(u)x_{m-2}(u) : u_{m-2} \le u \le u_{m-1} \\ \mu_{rm-1}(u)x_m(u) + \mu_{fm-1}(u)x_{m-1}(u) : u_{m-1} \le u \le u_m \end{cases} \tag{18}$$

$$Y(u) = \begin{cases} \mu_{r1}(u)y_2(u) + \mu_{f1}(u)y_1(u) : u_1 \le u \le u_2 \\ \mu_{r2}(u)y_3(u) + \mu_{f2}(u)y_2(u) : u_2 \le u \le u_3 \\ \mu_{r3}(u)y_4(u) + \mu_{f3}(u)y_3(u) : u_3 \le u \le u_4 \\ \vdots \\ \mu_{rm-2}(u)y_{m-1}(u) + \mu_{fm-2}(u)y_{m-2}(u) : u_{m-2} \le u \le u_{m-1} \\ \mu_{rm-1}(u)y_m(u) + \mu_{fm-1}(u)y_{m-1}(u) : u_{m-1} \le u \le u_m \end{cases} \tag{19}$$

$$Z(u) = \begin{cases} \mu_{r1}(u)z_2(u) + \mu_{f1}(u)z_1(u) : u_1 \le u \le u_2 \\ \mu_{r2}(u)z_3(u) + \mu_{f2}(u)z_2(u) : u_2 \le u \le u_3 \\ \mu_{r3}(u)z_4(u) + \mu_{f3}(u)z_3(u) : u_3 \le u \le u_4 \\ \vdots \\ \mu_{rm-2}(u)z_{m-1}(u) + \mu_{fm-2}(u)z_{m-2}(u) : u_{m-2} \le u \le u_{m-1} \\ \mu_{rm-1}(u)z_m(u) + \mu_{fm-1}(u)z_{m-1}(u) : u_{m-1} \le u \le u_m \end{cases} \tag{20}$$

$\mu_f(u)$ is a decreasing function and is depicted by the red dotted line in Figure 2. This function starts at one and decreases to zero. On the other hand, $\mu_r(u)$ is an increasing function. This function starts at zero and increases to one. The membership functions are as follows.

$$\mu_{rn}(u) = \frac{(u - u_n)}{(u_{n+1} - u_n)}, (1 \le n \le m-1) \tag{21}$$

$$\mu_{rn}(u) = \frac{(u_{n+1} - u)}{(u_{n+1} - u_n)}, (1 \le n \le m-1) \tag{22}$$

Since there is no overlap in the first and last sections, the polynomials are not affected by other polynomials and the membership function is not required. Therefore, $\mu_{r_1}(u)$ and $\mu_{f\ m-1}(u)$. are both one. The general functions can be described as:

$$X(u) = \begin{cases} x_2(u) : u_1 \le u \le u_2 \\ \mu_{rn}(u)x_{n+1}(u) + \mu_{fn}(u)x_n(u), (2 \le n < m-1) : u_2 \le u \le u_{m-1} \\ x_{m-1}(u) : u_{m-1} \le u \le u_m \end{cases} \tag{23}$$

$$Y(u) = \begin{cases} y_2(u) : u_1 \le u \le u_2 \\ \mu_{rn}(u)y_{n+1}(u) + \mu_{fn}(u)y_n(u), (2 \le n < m-1) : u_2 \le u \le u_{m-1} \\ y_{m-1}(u) : u_{m-1} \le u \le u_m \end{cases} \tag{24}$$

$$Z(u) = \begin{cases} z_2(u) : u_1 \le u \le u_2 \\ \mu_{rn}(u)z_{n+1}(u) + \mu_{fn}(u)z_n(u), (2 \le n < m-1) : u_2 \le u \le u_{m-1} \\ z_{m-1}(u) : u_{m-1} \le u \le u_m \end{cases} \tag{25}$$

The polynomial is obtained by finding the quadratic polynomial's solution using the 3 × 3 inverse matrix, which has a unique and simple solution without high-order polynomials and trigonometric functions. On the other hand, the quadratic polynomials have the cubic polynomial form at the overlapped sections using the membership functions.

5. Continuity and curvature

Path continuity is determined by differential values. If the first-order differential values connect each waypoint, the path is the G^1 continuous path. If the second-order differential values match at each waypoint, the path is the G^2 continuous path. The first-order differential values of $X(u)$, $Y(u)$, and $Z(u)$ can be obtained as:

$$\frac{dX(u)}{du} = \begin{cases} dx_2(u) : u_1 \le u \le u_2 \\ \mu_{rn}(u)dx_{n+1}(u) + \mu_{fn}(u)dx_n(u), (2 \le n < m-1) : u_2 \le u \le u_{m-1} \\ dx_{m-1}(u) : u_{m-1} \le u \le u_m \end{cases} \tag{26}$$

$$\frac{dY(u)}{du} = \begin{cases} dy_2(u) : u_1 \le u \le u_2 \\ \mu_{rn}(u)dy_{n+1}(u) + \mu_{fn}(u)dy_n(u), (2 \le n < m-1) : u_2 \le u \le u_{m-1} \\ dy_{m-1}(u) : u_{m-1} \le u \le u_m \end{cases} \tag{27}$$

$$\frac{dZ(u)}{du} = \begin{cases} dz_2(u) : u_1 \le u \le u_2 \\ \mu_{rn}(u)dz_{n+1}(u) + \mu_{fn}(u)dz_n(u), (2 \le n < m-1) : u_2 \le u \le u_{m-1} \\ dz_{m-1}(u) : u_{m-1} \le u \le u_m \end{cases} \tag{28}$$

The second-order differential values are described as:

$$\frac{d^2X(u)}{du} = \begin{cases} d^2x_2(u) : u_1 \le u \le u_2 \\ \mu_{rn}(u)d^2x_{n+1}(u) + \mu_{fn}(u)d^2x_n(u), (2 \le n < m-1) : u_2 \le u \le u_{m-1} \\ d^2x_{m-1}(u) : u_{m-1} \le u \le u_m \end{cases} \tag{29}$$

$$\frac{d^2Y(u)}{du^2} = \begin{cases} d^2y_2(u) : u_1 \le u \le u_2 \\ \mu_{rn}(u)d^2y_{n+1}(u) + \mu_{fn}(u)d^2y_n(u), (2 \le n < m-1) : u_2 \le u \le u_{m-1} \\ d^2y_{m-1}(u) : u_{m-1} \le u \le u_m \end{cases} \tag{30}$$

$$\frac{d^2Z(u)}{du^2} = \begin{cases} d^2z_2(u) : u_1 \le u \le u_2 \\ \mu_{rn}(u)d^2z_{n+1}(u) + \mu_{fn}(u)d^2z_n(u), (2 \le n < m-1) : u_2 \le u \le u_{m-1} \\ d^2z_{m-1}(u) : u_{m-1} \le u \le u_m \end{cases} \tag{31}$$

On the x axis, the $x_n(u)$ is the quadratic polynomial connecting the $(n-1)^{th}$, n^{th}, and $(n+1)^{th}$ waypoints. This polynomial has maximum degree at the n^{th} waypoint. At the n^{th} waypoint, $x_n(u)$ exists and the differential values of $x_n(u)$ are only affected at the n^{th} waypoint. Therefore, the first-order and second-order differential values at the n^{th} waypoint are decided by $x_n(u)$. This means the path is the G^2 continuous path. The $y_n(u)$ and $z_n(u)$ can be described as the case of $x_n(u)$. These values are as follows:

$$\frac{dx_n}{du} = 2a_{x_n}u + b_{x_n}, \frac{d^2x_n}{du^2} = 2a_{x_n} \tag{32}$$

$$\frac{dy_n}{du} = 2a_{y_n}u + b_{y_n}, \frac{d^2y_n}{du^2} = 2a_{y_n} \tag{33}$$

$$\frac{dz_n}{du} = 2a_{z_n}u + b_{z_n}, \frac{d^2z_n}{du^2} = 2a_{z_n} \tag{34}$$

Equations (32), (33), and (34) show the first-order and second-order differential values at the n^{th} waypoint.

The curvature is related to the differential values. The curvature of the parametric quadratic polynomial can be described by equation (35) using equations (26) to (31).

$$k = \frac{\sqrt{(z''y' - y''z')^2 + (x''z' - z''x')^2 + (y''x' - x''y')^2}}{(x'^2 + y'^2 + z'^2)^{\frac{3}{2}}} \tag{35}$$

Equation (35) consists of the differential values and is a function of u. The continuity of curvature is an important part in constructing the smooth path. The G^2 continuous path is called the curvature continuous path. The continuous curvature is necessary to verify G^2 continuity. If the curvature graph is continuous, this path is the G^2 continuous path.

6. Simulation

6.1 Movement of aerial robot in the 3D-space

This simulation assumes that the aerial robot requires a smooth 3D path. Table 1 shows ten waypoints. The proposed method is applied to construct the smooth 3D path from the waypoints and the continuity of the planned 3D path is then verified. In addition, variations of each axis and curvature are obtained.

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
x	0	20	50	80	90	80	50	20	20	0
y	0	10	20	20	50	80	80	60	40	0
z	0	15	20	15	20	15	20	15	10	0

Table 1. Waypoints for simulation

This path starts at the origin. The aerial robot flies around an area with variable elevation and returns to the origin. To apply the proposed method, the parameter u was obtained using equation (12).

u_1	u_2	u_3	u_4	u_5
0	26.93	58.94	89.35	121.37
u_6	u_7	u_8	u_9	u_{10}
153.39	183.80	220.20	240.82	286.64

Table 2. Parameter u for simulation

6.2 3D smooth path planning

The general linear path is shown in Figure 3. In this simulation, we change this linear piecewise path to the G^2 continuous path.

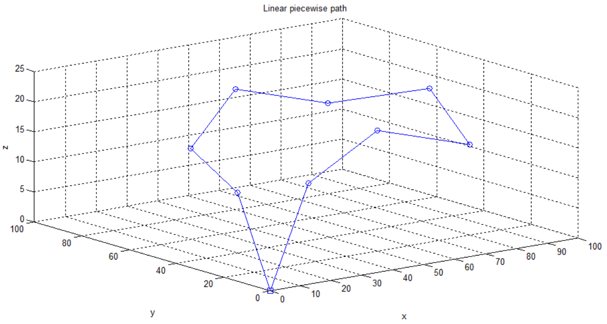
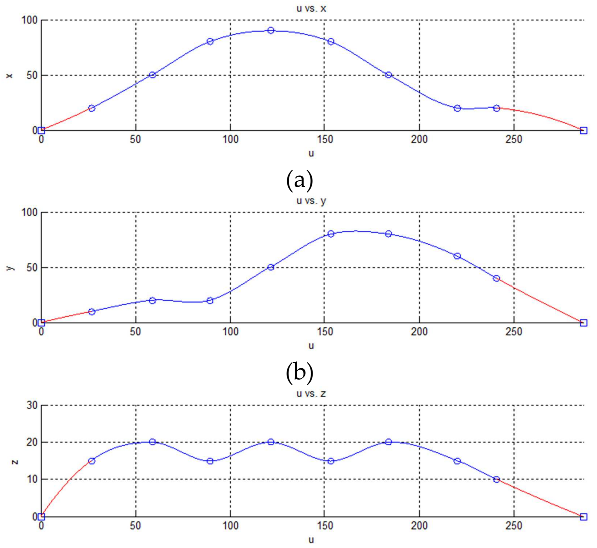


Figure 3. Linear piecewise path of aerial robot

The first step requires finding variations of x, y, and z using equations (23), (24), and (25). These variations are shown in Figure 4.



(a)

(b)

(c)

Figure 4. (a) Variation of the x axis. (b) Variation of the y axis. (c) Variation of elevation in the z axis.

Figure 5 shows the path on the 2D plane using the variations of x and y. The variation of z is in accordance with the altitude of the aerial robot.

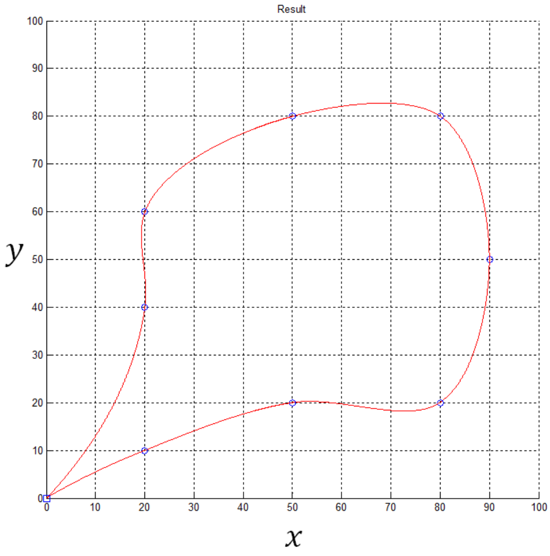


Figure 5. Aerial robot's movement on the 2D plane

The 3D path is formed by expressing the path as shown in Figure 5 along with the altitude in the 3D space. Figure 6 displays the 3D path using the proposed method.

Figure 6 has a continuous form, but determining whether it is a G^2 continuous path requires checking the continuity of the path.

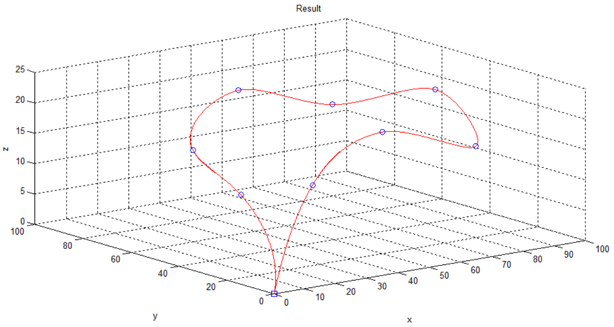
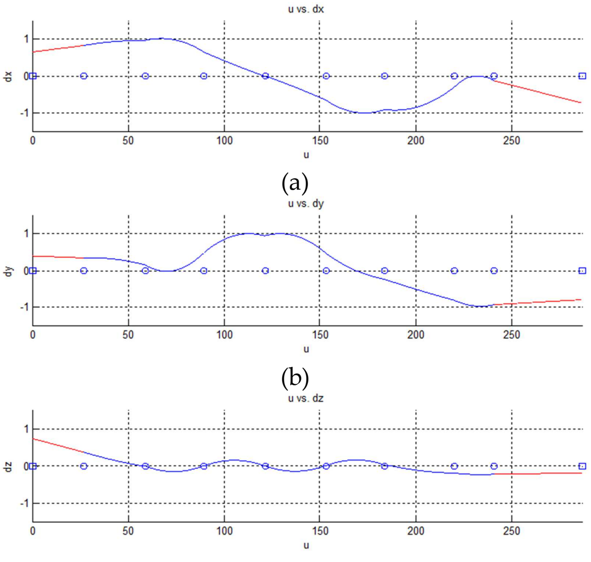


Figure 6. 3D smooth path using the proposed method

6.3 Checking continuity

Continuity can be determined by the first-order and second-order differential values. The graph of the first-order differential values is shown in Figure 7 and the graph of second-order differential values is shown in Figure 8.



(a)

(b)

(c)

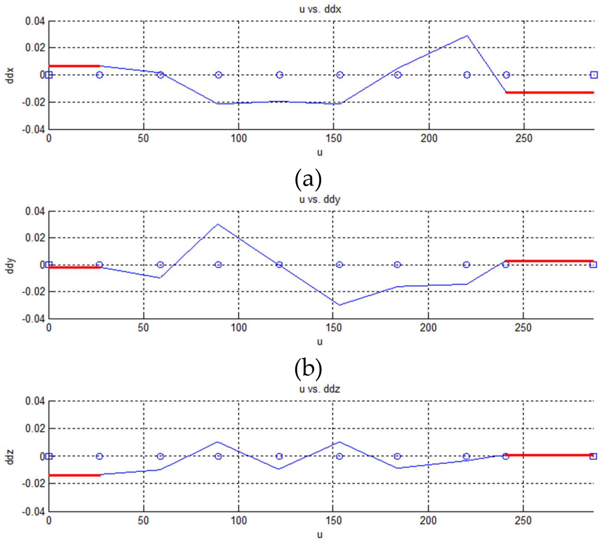
Figure 7. Graphs of the first-order differential values: (a) is the x, (b) is the y, and (c) is the z axis

Figures 7 and 8 show continuous graphs in all the axes. Since there are no disjointed sections visible in the graphs, the path has G^2 continuity.

These values can create the curvature by using equation (35). The graph of curvature is shown in Figure 9.

The G^2 continuous path has continuous curvature. The planned path in Figure 9 also has continuous curvature. The section in which the curvature changes in Figure 9 matches the variation shown in Figure 6. Therefore, the planned 3D smooth path is the G^2 continuous path.

The simulation showed that the proposed method constructs the 3D G^2 continuous path from the waypoints. The continuity is verified by checking the differential values. The proposed method does not use trigonometric and high-order functions.



(a)

(b)

(c)

Figure 8. Second-order differential values of the smoothed path: (a) is the x, (b) is the y, and (c) is the z axis

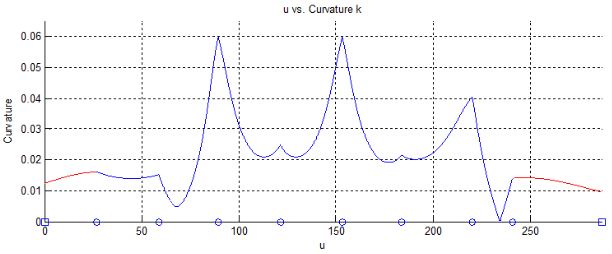


Figure 9. Graph of curvature

7. Conclusions

Many studies have focused on finding collision-free paths and solving mazes on the 2D plane. However, in real environments, movements are required to be stable. In order to achieve stable movement, paths should be continuous.

For aerial robots and aerial vehicles, the movement must follow a smooth path in the 3D space. They must maintain velocity and cannot change direction rapidly. General waypoint-planning methods form paths only using waypoints without considering the continuity of the path. If aerial robots and aerial vehicles follow a planned continuous path, the flight can be made more stable through optimal movements using the shortest path.

This paper has proposed a method for constructing the 3D smooth path from waypoints in a 3D space. The path must visit every waypoint and have G^2 continuity. The 3D smooth path is obtained without using complex calculations such as trigonometric functions or using high-order polynomial functions.

The proposed method expands the QPMI path-smoothing method and uses quadratic polynomial interpolation to create the unique, shortest, G^2 continuous path about three waypoints. This paper takes these advantages of quadratic polynomial interpolation for constructing the 3D smooth path. The proposed method has two steps. The first step is to make waypoint groups that consist in three waypoints. Each group has its own path obtained from the parametric quadratic polynomial interpolation. The 3D path is expressed using a parametric method and is the G^2 continuous path about each waypoint group having the unique, shortest path. The second step is to combine the planned waypoint group's paths. The membership function uses the parameter u and distributes the degree of each waypoint group's path. Each path is merged into one path to maintain continuity. The planned path is shown to be the G^2 continuous 3D path.

The proposed 3D path-smoothing method uses only quadratic polynomials and features a relatively simple computation. It produces a unique, shortest, G^2 continuous path that visits all the waypoints. The curvature of the path is verifiable. This method can also be combined with a waypoint planner or a path-searching method to construct a 3D smooth path. In addition, this method does not require storage for saving path data. These advantages enable the development of a simple system.

8. Acknowledgements

This research was supported by the Inha University.

9. References

[1] A. Stentz, "The focussed D* algorithm for real-time replanning," in *Artificial Intelligence. International Joint Conference on*, pp. 1652-1659, 1995.

[2] Y. K. Hwang and N. Ahuja, "A potential field approach to path planning," *Robotics and Automation. IEEE Transactions on,* vol. 8, pp. 23-32, 1992.

[3] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on,* vol. 12, pp. 566-580, 1996.

[4] J. J. Kuffner Jr and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Robotics and Automation (ICRA'00). Proceedings, IEEE International Conference on*, pp. 995-1001, 2000.

[5] I. Hasircioglu, H. R. Topcuoglu, and M. Ermis, "3-D path planning for the navigation of unmanned aerial vehicles by using evolutionary algorithms," in *Genetic and Evolutionary Computation. Proceedings, 10th Annual Conference on*, pp. 1499-1506, 2008.

[6] A. Babaei and M. Mortazavi, "Three-dimensional curvature-constrained trajectory planning based on in-flight waypoints," *Journal of Aircraft,* vol. 47, pp. 1391-1398, 2010.

[7] H. J. Kim and D. H. Shim, "A flight control system for aerial robots: algorithms and experiments," *Control Engineering Practice,* vol. 11, pp. 1389-1400, 2003.

[8] J. D. Anderson, *Fundamentals of Aerodynamics,* vol. 2. New York: McGraw-Hill, 2001.

[9] G. E. Farin, *Curves and surfaces for CAGD: a practical guide*. 5th ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.

[10] E. Magid, D. Keren, E. Rivlin, and I. Yavneh, "Spline-based robot navigation," in *Intelligent Robots and Systems. 2006 IEEE/RSJ International Conference on*, pp. 2296-2301, 2006.

[11] K. Komoriya and K. Tanie, "Trajectory design and control of a wheel-type mobile robot using B-spline curve," in *Intelligent Robots and Systems (IROS'89), The Autonomous Mobile Robots and Its Applications. Proceedings, IEEE/RSJ International Workshop on*, pp. 398-405, 1989.

[12] A. Piazzi, C. G. L. Bianco, and M. Romano, "Smooth path generation for wheeled mobile robots using η3-splines," *Motion Control,* Federico Casolo (Ed.), pp. 75-96, 2010.

[13] T. Berglund, H. Jonsson, and I. Soderkvist, "An obstacle-avoiding minimum variation b-spline problem," in *Geometric Modeling and Graphics. Proceedings, 2003 International Conference on*, pp. 156-161, 2003.

[14] E. P. Anderson, R. W. Beard, and T. W. McLain, "Real-time dynamic trajectory smoothing for unmanned air vehicles," *Control Systems Technology, IEEE Transactions on,* vol. 13, pp. 471-477, 2005.

[15] D. Jung and P. Tsiotras, "On-line path generation for small unmanned aerial vehicles using B-spline path templates," in *AIAA Guidance, Navigation and Control Conference, AIAA*, vol. 7135, 2008.

[16] K. Yang and S. Sukkarieh, "3D smooth path planning for a UAV in cluttered natural environments," in *Intelligent Robots and Systems (IROS 2008). IEEE/RSJ International Conference on*, pp. 794-800, 2008.

[17] U. Y. Huh and S. R. Chang, "A G^2 Continuous Path-smoothing Algorithm Using Modified Quadratic Polynomial Interpolation," *International Journal of Advanced Robotic Systems,* vol. 11, p. 25, 2014.

[18] S. R. Chang and U. Y. Huh, "A Collision-Free G^2 Continuous Path-Smoothing Algorithm Using Quadratic Polynomial Interpolation," *International Journal of Advanced Robotic* Systems, vol. 11, p. 194, 2014.