A $G^2$ Continuous Path-smoothing Algorithm Using Modified Quadratic Polynomial Interpolation

Regular Paper

Uk-Youl Huh\textsuperscript{1,*} and Seong-Ryong Chang\textsuperscript{1}

\textsuperscript{1} Inha University, In-cheon, Republic of Korea
* Corresponding author E-mail: uyhuh@inha.ac.kr

Received: Sep 05, 2013; Accepted: Oct 16, 2013

DOI: 10.5772/57340

© 2014 The Author(s). Licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract Path searching algorithm is one of the main topics in studies on path planning. These algorithms are used to avoiding obstacles and find paths from starting point to target point. There are dynamic problems that must be addressed when these paths are applied in real environments. In order to be applicable in actual situations, the path must be a smooth path. A smooth path is a path that maintains continuity. Continuity is decided by the differential values of the path. In order to be $G^2$ continuous, the secondary differential values of the path must be connected throughout the path. In this paper, the interpolation method is used to construct continuous paths. The quadratic polynomial interpolation is a simple method for obtaining continuous paths about three points. The proposed algorithm makes a connection of three points with curves and the proposed path is rotated using the parametric method in order to make the path optimal and smooth. The polynomials expand to the next three points and they merge into the entire path using the membership functions with $G^2$ continuity.

Keywords Continuous Path, Curve Fitting, Function Approximation, Interpolation, Path Planning, Robot Motion, Smoothing Algorithm, Smooth Path, Vehicle Navigation

1. Introduction

Path planning is an important topic in robotics studies. One of the general purposes of path planning is to construct collision free paths. Typical approaches include the D* algorithm [1], the Probabilistic Roadmap Method (PRM) algorithm [2], the Rapidly-exploring Random Tree algorithm (RRT) [3], and the potential field algorithm [4]. These algorithms can construct collision free paths, but they cannot guarantee that the resulting paths will be able to apply driving with stability and optimality in real environments.

Collision free paths are designed for obstacle avoidance. However, a continuous path is also suitable for the actual movements of robots. The actual movements of robots are affected by various forces (i.e., gravity, centrifugal force and centripetal force). These forces contribute to dynamic problems such as the slip and over-actuation of wheels,
limitations of velocity and acceleration, and the effects of inertia. The forces that affect real movements need to be taken into consideration when planning and improving paths.

Considering these forces, paths should be continuous paths. There are paths connecting points on a two dimensional plane. Continuity is defined using geometry [5]. A $G^0$ continuous path is simply a path that is connected for entire sections. These paths can be constructed by the path searching algorithms mentioned above. A $G^2$ continuous path matches the primary differential values at each point. This path shares a common tangent direction. A $G^2$ continuous path has the same secondary differential values at each point. This path also shares a common centre of curvature. Hence, a $G^n$ continuous path signifies the equality of the $n^{th}$ differential value at each point.

The primary differential values on the paths indicate velocity. In addition, the secondary difference values indicate acceleration. Thus, the $G^2$ continuous paths have continuous velocities. The $G^2$ continuous paths are guaranteed to have continuous accelerations. These can reduce dynamic problems.

In order to apply the mobile robot or vehicle, Suzuki et al. demonstrated smooth path planning with pedestrian avoidance for a wheeled robot [6]. Vaillagra et al. proposed smooth path and speed planning for smooth autonomous navigation [7]. Yang et al. reported a continuous curvature path-smoothing algorithm using cubic B ezier spiral curves for non-holonomic robots [8].

A continuous path is also called a smooth path and the construction of continuous path algorithms is defined as path smoothing algorithms. There are two kinds of path smoothing algorithms. The first type is a graphical method and the second is a functional method. The graphical method uses lines, arcs, circles and clothoids to construct smooth paths. It is possible to analyse the results from graphical methods. The Dubins method [9] is a typical method for doing this. The Dubins method finds the shortest curve that connects two points in the two dimensional plane using lines and arcs. This method can be expanded into another graphical method. However, methods such as these have problems with limitations on path expressions and discontinuities at junction points. In order to solve these problems, studies have focused on functional methods.

Functional methods use mathematical functions to express the path. These methods can be used to construct various types of paths. The spline interpolation method is widely used. This method finds functions for connections between two points using differential values at each point [10]. Komoriya et al. suggested the B-spline method for designing trajectory and controlling wheel-type mobile robots [11]. Magid et al. proposed a spline-based method for robot navigation [12] and Piazzi et al. demonstrated smooth path generation using $η^3$ splines for wheeled mobile robots [13]. The $η^3$ spline method uses trigonometric functions and high order functions. Therefore, the complexities of the calculations increase and the numerical analysis can be complicated.

The polynomial interpolation method [5] is a simple functional method. This method finds a polynomial that includes all of the points. The number of points that are needed to construct a $(n-1)^{th}$ order polynomial is $n$. Polynomials can be used to find simple methods for constructing paths. However, high order polynomials lead to the Runge’s phenomenon [14] and a sharp increase in computation.

The essential requirements of a path smoothing algorithm are as follows: it must be possible to obtain the path using simple methods and calculations; the functions should be expressed in differential form; $G^2$ continuity must be maintained throughout the proposed path; it must be possible to express the vertical and horizontal path; it must be able to obtain the heading angle and curvature in order to enable analysis of the planned path.

To satisfy these requirements, this paper used a functional method. The spline interpolation method is used to form connections between points. These connections were suitable for the path smoothing algorithm using the start point (SP), the way points (WP) and the target point (TP) (these points are the position that robots or vehicles must visit on the map). They are also suitable for converting paths from line connecting paths to smooth paths. However, the trigonometric functions and high order functions can be used with the advanced spline interpolation method [13]. These functions lead to a high level of complexity. It is possible to use the polynomial interpolation method for the smoothing algorithm along with the necessary requirements. However, the polynomial method does not give an optimal path for random points. Also this method cannot be used many WPs, because Runge’s phenomenon can occur. Additionally, high order polynomials increase the calculation load.

Spline interpolation methods and polynomial interpolation methods can be used to interpolate a graph of the geometry. These methods should be used with the variations of $x$ or $y$. The $x$ or $y$ values are always increased or decreased in order to obtain the functions by general polynomial interpolation methods (Newton’s polynomial interpolation and Lagrange’s polynomial interpolation) [5]. Therefore, general polynomial interpolation methods cannot express vertical or horizontal movements like those of robots and vehicles.
In this paper, the polynomial interpolation method is used. If the $x$ or $y$ values of points are located at equal distances, the quadratic polynomial is the minimum order polynomial that includes three points with $G^2$ continuity. However, another method is required in order to express a path of irregular points or a path of points that has the same value on axis. It is possible for the rotated polynomials to give the shortest and continuous path for three points; however, it is difficult to obtain the rotated polynomials using normal methods. The general rotation method uses a rotation matrix. This matrix relies on trigonometric functions and can therefore lead to complex calculations. In this paper, the parametric method is used to obtain the rotated polynomials.

To apply for three or more points, the quadratic polynomial is extended to the subsequent three points and they are combined with the previous polynomial. Therefore, the proposed method should be able to construct a $G^2$ continuous polynomial that includes the next three points using the quadratic polynomial. In this paper, the membership function is used to combine polynomials. The membership function is part of a fuzzy algorithm. This function is used to regulate the certainty of a fuzzy set. The proposed method uses the membership function for distributing the degree of the polynomials in the overlapping sections. The fuzzy set assumes the polynomials and the membership function is changed based on the accumulation value of the linear distance. This function controls the degree of each polynomial. Thus, the $n^{th}$ polynomial has the maximum degree at the $n^{th}$ WP, but this degree decreases closer to the $(n + 1)^{th}$ WP. The $(n + 1)^{th}$ polynomial has the minimum degree at the $n^{th}$ WP. This degree increases closer to the $(n + 1)^{th}$ WP. Hence, the polynomials are combined by the membership function controlling degree at each WP.

This paper is organized as follows: section 2 explains the characteristics of quadratic polynomials and a conception of path smoothing using the quadratic polynomials. In section 3, the parametric method is proposed for constructing the rotated quadratic polynomials. Section 4 analyses the rotated quadratic polynomials. Section 5 describes how the membership function can be used to merge the polynomials. Section 6 reports the simulation results for the proposed path. Section 7 documents the conclusions.

2. Quadratic polynomial interpolation

2.1 Characteristics of quadratic polynomial

The quadratic polynomial is a second order function for connecting three points. If this function is used for the path, then the three points include the SP, WP and TP. If the WP is closer to the vertex, the path can be more appropriate for the shortest and continuous path.
The quadratic polynomial can be obtained from (1) depending on the location of the point. In the case of #1 (green curve) in Figure 2, \( P_1 \) is the vertex between \( P_2 \) and \( P_3 \). However, in the case of #2 (blue curve) in Figure 2, \( P_1 \), \( P_2 \) and \( P_3 \) are not sequential. This path can be a distorted path. Therefore, constraints (3) and (4) must be satisfied to obtain the shortest and continuous path with \( C^2 \) continuity in the general quadratic polynomial.

If constraints (3) and (4) do not hold or points \( P_1 \), \( P_2 \) and \( P_3 \) are not sequential (5), the quadratic polynomial cannot be obtained for the following condition: \( (x_n = x_{n+1}) \). The values of \( x \) must have variation. These constraints mean that general polynomial interpolation cannot apply to the path smoothing algorithm.

\[
x_1 > x_2 > x_3 \text{ or } x_1 < x_2 < x_3
\]  
(5)

2.2 Shortest and continuous path using quadratic polynomial

If the points do not satisfy constraint (5), then the values of \( x \) are not sequential or the values of \( y \) have not changed. Figure 3 shows the graph of the polynomial for an irregular \( x \). Points \( P_1 \), \( P_2 \) and \( P_3 \) are located irregularly on the curve.

The simple method (2) can be used to construct the quadratic polynomial that connects \( P_1 \), \( P_2 \) and \( P_3 \). However, it is impossible to use this method to produce a sequential path that connects \( P_1 \), \( P_2 \) and \( P_3 \). Furthermore, the other general methods (Lagrangian interpolation polynomials and Newton's interpolation polynomials) [5] require the variation of \( x \) or \( y \). The path cannot be constructed using these methods for the following conditions: \( x_1 = x_2, \ x_2 = x_3, \ y_1 = y_2 \) and \( y_2 = y_3 \). These conditions occurred for vertical and horizontal movements on the 2D-plane.

The shortest and continuous path for irregular values of \( x \) is shown in Figure 4.

The red curve is the shortest curve centred on \( P_2 \) (middle point) and includes \( P_1 \), \( P_2 \) and \( P_3 \). This curve is constructed using the quadratic polynomial. In this paper, this red curve is used to construct the smooth path with continuity.

3. Modified quadratic polynomial interpolation using the parametric method

In section 2, the quadratic polynomial graph is centred on \( P_2 \) and is rotated. However, (1) is changed by the rotation matrix, which is based on the trigonometric functions that are used to construct the rotated quadratic polynomial [15]. The trigonometric functions are occurred complex calculations and open sections. Vertical and horizontal paths cannot be constructed using the trigonometric functions. The key characteristics of trigonometric functions are that \( \tan \theta \) has an unlimited value at \( \theta = \frac{\pi}{2} \) and \( \theta = \frac{3\pi}{2} \).

In this paper, a parameter \( u \) is used. The parameter \( u \) is the linear distance between each point. \( u_1 \) is always zero. \( u_2 \) is the distance between \( P_1 \) and \( P_2 \). \( u_3 \) can be obtained by calculating \( u_2 \) and the distance between \( P_2 \) and \( P_3 \). The functions are as follows:

\[
\begin{align*}
u_1 &= 0 \\
u_2 &= u_1 + \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \\
u_3 &= u_2 + \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}
\end{align*}
\]  
(6)

(1) can be transformed into (8) and (9) using \( u \).

\[
\begin{align*}
P(u) &= (x(u), y(u)) \quad (7) \\
x(u) &= a_xu^2 + b_xu + c_x \quad (8) \\
y(u) &= a_yu^2 + b_yu + c_y \quad (9)
\end{align*}
\]

Equations (8) and (9) are the quadratic polynomials of \( u \). The parameter \( u \) always has a positive value and connects each point in the serial order with irregular \( x \) and \( y \) values. The function of \( x \) is changed to the function of \( u \). The function of \( u \) is separate from that of \( x \) and \( y \). Each of the functions has the same characteristics and the continuity that is derived from the quadratic polynomial.
Equation (2) can be transformed into (10) and (11) using the \( \alpha \) parameter.

\[
\begin{align*}
(a_x) &= \begin{pmatrix} a_{x1} \\ b_x \\ c_x \end{pmatrix} = \begin{pmatrix} u_1^2 & u_4 & 1 \\ u_2^2 & u_2 & 1 \\ u_3^2 & u_3 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x(u_1) \\ x(u_2) \\ x(u_3) \end{pmatrix} \\
(b_y) &= \begin{pmatrix} u_1^2 & u_4 & 1 \\ u_2^2 & u_2 & 1 \\ u_3^2 & u_3 & 1 \end{pmatrix}^{-1} \begin{pmatrix} y(u_1) \\ y(u_2) \\ y(u_3) \end{pmatrix} \\
(c_y) &= \begin{pmatrix} u_1^2 & u_4 & 1 \\ u_2^2 & u_2 & 1 \\ u_3^2 & u_3 & 1 \end{pmatrix}^{-1} \begin{pmatrix} z(u_1) \\ z(u_2) \\ z(u_3) \end{pmatrix}
\end{align*}
\]

(10)

Equation (12) is a generalized form of equation (6). The \( m \) parameter represents the number of points.

\[
u_n = \sum_{n=2}^{m} \sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2}
\]

(12)

Equations (10) and (11) can be generalized to:

\[
\begin{align*}
(a_{x_n}) &= \begin{pmatrix} u_{n-1}^2 & u_{n-1} & 1 \\ u_n^2 & u_n & 1 \\ u_{n+1}^2 & u_{n+1} & 1 \end{pmatrix}^{-1} \begin{pmatrix} x_n(u_{n-1}) \\ x_n(u_n) \\ x_n(u_{n+1}) \end{pmatrix} \\
(b_{y_n}) &= \begin{pmatrix} u_{n-1}^2 & u_{n-1} & 1 \\ u_n^2 & u_n & 1 \\ u_{n+1}^2 & u_{n+1} & 1 \end{pmatrix}^{-1} \begin{pmatrix} y_n(u_{n-1}) \\ y_n(u_n) \\ y_n(u_{n+1}) \end{pmatrix} \\
(c_{y_n}) &= \begin{pmatrix} u_{n-1}^2 & u_{n-1} & 1 \\ u_n^2 & u_n & 1 \\ u_{n+1}^2 & u_{n+1} & 1 \end{pmatrix}^{-1} \begin{pmatrix} z_n(u_{n-1}) \\ z_n(u_n) \\ z_n(u_{n+1}) \end{pmatrix}
\end{align*}
\]

(13)

(14)

Equations (8) and (9) are described as follows:

\[
\begin{align*}
x_n(u) &= a_x u^2 + b_x u + c_x, \quad 2 \leq n \leq m - 1 \\
y_n(u) &= a_y u^2 + b_y u + c_y, \quad 2 \leq n \leq m - 1
\end{align*}
\]

(15)

(16)

Equation (7) is defined based on (17).

\[
P(u) = \begin{cases} 
(x(u), y(u)) : (u_i \leq u \leq u_j), \\
(x_{n-1}(u), y_{n-1}(u)), & (u_{n-1} \leq u \leq u_n) \\
(x_{n+1}(u), y_{n+1}(u)) \end{cases}
\]

(17)

Equations (8) and (9) can be differentiated using \( u \).

\[
\begin{align*}
dx &= 2a_x u + b_x, \quad \frac{dx}{du} = 2a_x \\
dy &= 2a_y u + b_y, \quad \frac{dy}{du} = 2a_y
\end{align*}
\]

(18)

(19)

Equations (8) and (9) can be differentiated using \( u \).

\[
k = \frac{y''}{(1 + y'^2)^{3/2}} = \frac{2a}{((2ax + b)^2 + 1)^{3/2}}
\]

(20)

The primary differential function is a function of \( u \). The secondary differential function takes on a constant value. The general quadratic polynomial (that connects only three points) has the same primary and secondary differential values at each point. Therefore, each polynomial has \( G^2 \) continuity as those in (21) and (22).

The heading angles of mobile robots and vehicles are defined as follows:

\[
\theta = \tan^{-1} \left( \frac{y'}{x'} \right) = \tan^{-1} \left( \frac{2a_x u + b_x}{2a_y u + b_y} \right)
\]

(24)

The heading angles are important data for path analysis and driving. The primary differential values of \( x \) and \( y \) can be used to obtain these values (24).
However, the method for finding continuity and heading angles cannot be applied to the overlapping sections. Section 5 describes the method for finding these values in the overlapping sections.

5. Modified quadratic polynomial interpolation using the parametric method and membership function

5.1 Membership function

The membership function is used to control the degree of truth in the fuzzy control theory.

Figure 6. Membership function of a fuzzy set (triangle)

The membership function $\mu(x)$ increases from 0 to 1 as $x$ increases. $\mu(x)$ starts at 0 and increases to 1 until the centre of $x$. This function decreases to 0 again as $x$ increases. In this paper, the membership function is used to control the degree of each polynomial’s effect on the entire path. In Figure 6, $x$ is changed to $u$ and the membership function is extended to the $m^{th}$ membership functions, as shown in Figure 7.

Figure 7. Graph of membership functions vs. $u$

The membership function of $n^{th}$ has two types, a rising function and a falling function. The dotted lines in Figure 7 represent the rising function. This function has increased from 0 to 1 for the $u_{m-1}$ to $u_m$ section. The falling function is represented by the dashed lines in Figure 7. This function has decreased 1 to 0 for the $u_n$ to $u_{n+1}$ section. General function can be described as follows:

\[
\mu_{r_n}(u) = \frac{(u-u_{n-1})}{(u_{n+1}-u_{n-1})}, \quad (1 \leq n \leq m-1) \tag{25}
\]

\[
\mu_{f_n}(u) = \frac{(u_{n+1}-u)}{(u_{n+1}-u_{n})}, \quad (1 \leq n \leq m-1) \tag{26}
\]

5.2 Polynomials and membership functions

P: $(X(u), Y(u))$ defines the group of the paths. These paths are constructed by combinations of the polynomials and membership functions. $X(u)$ and $Y(u)$ are described in (27) and (28) using (25) and (26).

\[
X(u) = \begin{cases} 
\mu_{r_1}(u)x_1(u) + \mu_{f_1}(u)x_2(u) : u_1 \leq u \leq u_2 \\
\mu_{r_2}(u)x_3(u) + \mu_{f_2}(u)x_4(u) : u_2 \leq u \leq u_3 \\
\mu_{r_3}(u)x_5(u) + \mu_{f_3}(u)x_6(u) : u_3 \leq u \leq u_4 \\
\vdots \\
\mu_{r_{m-2}}(u)x_{m-3}(u) + \mu_{f_{m-2}}(u)x_{m-2}(u) : u_{m-2} \leq u \leq u_{m-1} \\
\mu_{r_{m-1}}(u)x_{m-1}(u) + \mu_{f_{m-1}}(u)x_{m}(u) : u_{m-1} \leq u \leq u_m 
\end{cases} \tag{27}
\]

\[
Y(u) = \begin{cases} 
\mu_{r_1}(u)y_1(u) + \mu_{f_1}(u)y_2(u) : u_1 \leq u \leq u_2 \\
\mu_{r_2}(u)y_3(u) + \mu_{f_2}(u)y_4(u) : u_2 \leq u \leq u_3 \\
\mu_{r_3}(u)y_5(u) + \mu_{f_3}(u)y_6(u) : u_3 \leq u \leq u_4 \\
\vdots \\
\mu_{r_{m-2}}(u)y_{m-3}(u) + \mu_{f_{m-2}}(u)y_{m-2}(u) : u_{m-2} \leq u \leq u_{m-1} \\
\mu_{r_{m-1}}(u)y_{m-1}(u) + \mu_{f_{m-1}}(u)y_{m}(u) : u_{m-1} \leq u \leq u_m 
\end{cases} \tag{28}
\]

In the first section $[P_1; P_2]$, the entire path consists of only $(x_2(u), y_2(u))$. In the last section $[P_{m-1}; P_m]$, the entire path is constructed using only $(x_{m-1}(u), y_{m-1}(u))$. Therefore, $\mu_{r_1}(u)$ and $\mu_{f_{m-1}}(u)$ are always equal to 1.

\[
x_1(u) = x_m(u) = y_1(u) = y_m(u) = 0 \tag{29}
\]

\[
\mu_{r_1}(u) = \mu_{f_{m-1}}(u) = 1 \tag{30}
\]

The general functions can be obtained by applying (29) and (30) to (27) and (28).

Hence, the quadratic polynomial will be the cubic polynomial in each of the sections $[P_2; P_3], [P_3; P_4], \ldots, [P_{m-2}; P_{m-1}]$. In this paper, the proposed algorithm is concerned with finding the quadratic polynomial using irregular $x$ and $y$. However, these polynomials are changed into the cubic functions by combining the membership functions. The graph of the overlapping sections has a curve representing the degree distribution based on the membership functions. The proposed method changes the path in Figure 5 into the path in Figure 8.

Figure 8. Graph of the proposed method (dashed line)
5.3 Analysis of proposed path

To analyse the proposed path, the differential values should be obtained and checked. The primary differential value can be defined using (21), (22) and the membership functions. The primary differential values are described in (33) and (34).

\[
dX(u) = \begin{cases} 
\mu_x(u)d_1x_{n-1}(u) + \mu_{-1}(u)d_2x_{n}(u), & (2 \leq n < m-1) : u_2 \leq u \leq u_{n-1} \\
\mu_x(u)d_1x_{n}(u), & u_1 \leq u \leq u_2 
\end{cases} (33)
\]

\[
dY(u) = \begin{cases} 
\mu_y(u)d_1y_{n-1}(u) + \mu_{-1}(u)d_2y_{n}(u), & (2 \leq n < m-1) : u_2 \leq u \leq u_{n-1} \\
\mu_y(u)d_1y_{n}(u), & u_1 \leq u \leq u_2 
\end{cases} (34)
\]

The secondary differential values are described in (35) and (36).

\[
dX(u) = \begin{cases} 
\mu_{x_2}(u)d_1x_{n-1}(u) + \mu_{-1}(u)d_2x_{n}(u), & (2 \leq n < m-1) : u_2 \leq u \leq u_{n-1} \\
\mu_{x_2}(u)d_1x_{n}(u), & u_1 \leq u \leq u_2 
\end{cases} (35)
\]

\[
dY(u) = \begin{cases} 
\mu_{y_2}(u)d_1y_{n-1}(u) + \mu_{-1}(u)d_2y_{n}(u), & (2 \leq n < m-1) : u_2 \leq u \leq u_{n-1} \\
\mu_{y_2}(u)d_1y_{n}(u), & u_1 \leq u \leq u_2 
\end{cases} (36)
\]

\(G^1\) continuity is related to the matching of the primary differential value at each point and \(G^2\) continuity is related to the agreement of the secondary differential value at each point. Each polynomial has the maximum degree at each point. Thus, \(x_n(u)\) and \(y_n(u)\) have the maximum degree at the \(n^{th}\) point. Therefore, the continuity at each point is decided by the differential values of \(x_n(u)\) and \(y_n(u)\). The values of \(x_n(u)\) and \(y_n(u)\) have \(G^2\) continuity at the \((n-1)^{th}\), \(n^{th}\) and \((n+1)^{th}\) points. The next set of polynomials, \(x_{n+1}(u)\) and \(y_{n+1}(u)\) are \(G^2\) continuous at the \(n^{th}\), \((n+1)^{th}\) and \((n+2)^{th}\) points. Hence, each polynomial has \(G^2\) continuity. If these polynomials are combined using the membership functions, then \(G^2\) continuity can be maintained throughout the entire path, because the membership functions distribute the degree of the polynomials and the differentials of the polynomials.

The heading angle can be obtained by modifying (24) using the primary differential values from (33) and (34).

\[
\theta = \tan^{-1}\left(\frac{dy(u)}{dx(u)}\right), (2 \leq n < m-1) (37)
\]

6. Simulations

6.1 Smooth path planning of random points

There are random points on the 2D-plane.

| P0(SP) | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9(TP) |
|-------|----|----|----|----|----|----|----|----|--------|
| x     | 8  | 7  | 2  | 11 | 11 | 45 | 45 | 25 | 12     |
| y     | 5  | 15 | 10 | 10 | 14 | 30 | 12 | 45 | 45     |

Table 1. Random points for simulation

The points of \(x\) and \(y\) have various values. \(P_4\) and \(P_5\) have the same \(x\) value. Also, \(P_9\) and \(P_{10}\) have the same \(y\) value. This means that the movement from \(P_4\) to \(P_5\) is a vertical movement and the movement from \(P_9\) to \(P_{10}\) is a horizontal movement.

The linear distance between the points can be obtained using (12). Table 2 displays a set of \(u\) values.

| \(u_1\) | \(u_2\) | \(u_3\) | \(u_4\) | \(u_5\) |
|--------|--------|--------|--------|--------|
| 0      | 10.05  | 17.12  | 26.12  | 30.12  |
| 67.69  | 85.69  | 124.28 | 137.28 | 161.13 |

Table 2. Set of \(u\) values

The variations of \(x\) and \(y\) can be changed into a function of \(u\) using (31) and (32). Figures 9 and 10 show the graphs of \(x\) and \(y\) vs. \(u\).

Figure 9. Graph of \(u\) vs. \(x\)

Figure 10. Graph of \(u\) vs. \(y\)

The entire path that is shown in Figure 11 is based on (31), (32), Figure 9 and Fig 10.

Figure 11. Smooth path of random points based on the proposed method
The proposed path is a continuous curve. To verify the continuity, the primary differential values should be checked. The primary differential values are obtained using (33) and (34). Figures 12 and 13 display the graphs of the primary differential values.

![Figure 12](image1.png)

**Figure 12.** Graph of \( u \) vs. \( dx \)

The \( dx \) and \( dy \) values are connected throughout each of the graphs. The connectedness of the primary differential values signifies that the proposed path is \( G^1 \) continuous.

The status for \( G^2 \) continuity can be decided by checking the secondary differential values. These values can be obtained using (35) and (36). The graphs are as follows:

Figures 14 and 15 display the secondary differential values of proposed path. The values are also connected throughout each of these graphs. These graphs demonstrate that the path has \( G^2 \) continuity.

![Figure 13](image2.png)

**Figure 13.** Graph of \( u \) vs. \( dy \)

The curvature graph is shown in Figure 16. The curvature can be obtained using (23).

![Figure 16](image3.png)

**Figure 16.** Graph of \( u \) vs. curvature

The heading angle graph that is shown in Figure 17 is based on (37).

![Figure 17](image4.png)

**Figure 17.** Graph of \( u \) vs. heading angle

The heading angle and curvature can be applied to the analysis of the planned path.

This simulation shows that the proposed method can be used to construct \( G^2 \) continuous paths. This method can also use for the analysis of paths with random points.

### 6.2 Path smoothing of PRM planner planned path

Figure 18 shows a path from the PRM planner [16]. This planner is used to construct collision free \( G^0 \) continuous paths. In this simulation, the proposed method will be applied to the planning process for \( G^0 \) continuous paths in order to improve continuity.

![Figure 18](image5.png)

**Figure 18.** Path from the PRM algorithm (bold line)

The variation of \( x \) and \( y \) are shown in Figures 19 and 20.

The functions of \( x \) and \( y \) are the increased functions with respect to \( u \). To show the entire path, Figures 19 and 20 are combined to produce Figure 21.

![Figure 20](image6.png)

**Figure 20.** Graph of \( u \) vs. \( dy \)
Figure 19. Graph of $u$ vs. $x$

Figure 20. Graph of $u$ vs. $y$

Figure 21. Path with improved continuity based on the proposed method
Each of the functions is $G^2$ continuous. It is necessary to check the differential values in order to verify continuity. Figures 22-25 are the graphs of the differential values.

Figures 22 and 23 prove that the differential values are connected.

Figure 22. Graph of $u$ vs. $dx$

Figure 23. Graph of $u$ vs. $dy$

Figure 24. Graph of $u$ vs. $d^2x$

Figure 25. Graph of $u$ vs. $d^2y$

The secondary differential values do not have any open sections. These figures indicate that the proposed path is also $G^2$ continuous.

The curvature and heading angle are shown in the following graphs:

Figure 26. Graph of $u$ vs. curvature

Figure 27. Graph of $u$ vs. heading angle

Figures 26 and 27 demonstrate that it is possible to analyse the proposed path.

In this simulation, the proposed method was used to improve a $G^0$ continuous path so that it was also $G^2$ continuous. The differential values, curvature and heading angle were also obtained. These values show that the proposed method can be used to construct continuous and analysable paths.
7. Conclusions

Continuous paths are important for driving mobile robots and vehicles in real environments. The spline interpolation method and polynomial interpolation methods have problems with tasks involving expressing and analysing paths. In this paper, the quadratic polynomial interpolation method was used to construct a $G^2$ continuous path. The quadratic polynomial can easily be obtained using a simple matrix. The proposed method is based on the rotated quadratic polynomial. This polynomial gives the shortest path with $G^2$ continuity and also the minimum order. To express the rotated quadratic polynomial, the parametric method was used. This method uses an accumulated linear distance value $u$. The variations of $x$ and $y$ values are transformed into functions of $u$. The polynomials of $x$ and $y$ can be expressed using the polynomial interpolation method. These functions are always increasing and make it possible to express vertical and horizontal movements. However, it is possible to use the proposed method only if there are three points. To apply the proposed method for three or more points, the membership functions are used to distribute the degrees of polynomials in overlapping sections. The quadratic polynomials are combined with the membership functions, which affect the whole path because of variations in the membership functions.

The differential values are obtained using the differential functions of $x$ and $y$. These values can be used to prove the continuity of path and can also serve as the basis for the curvature and heading angle.

Simulation results show that proposed methods can construct a $G^2$ continuous path with WPs. The first simulation constructed a $G^2$ continuous path with random points. The second simulation demonstrated that the proposed algorithm can be applied directly to the searched path from the PRM algorithm.

In order to use a real robot navigation system, the searched paths have WPs, or they are $G^0$ continuous paths. PRM algorithm planned path can change directly, as shown in the simulation. However, the $D^n$ algorithm and potential field algorithm requires another algorithm to decide WPs.

If there are WPs and a sequential rule of visiting WPs, the proposed method can construct a $G^2$ continuous path using only a second order polynomial. In addition, this method is a new interpolation method. It is not a required database for saving entire path. Additionally, the proposed method can use local path planning. In local path planning, if mobile robots or vehicles know the following two WPs, the proposed method can construct a $G^2$ continuous path of three points. In addition, if robots or vehicles reach the next point, the proposed method can update the paths using the reached point and the next two points.

This study suggests simple methods for obtaining $G^2$ continuous paths using analysable and functional forms. This research can be applied to the path smoothing algorithm and the curve fitting algorithm for $G^2$ continuity. If the proposed method is extended to 3D-space, it can be applied to flying robots and aircraft.

8. Acknowledgments

This study was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No.2012-0005564).

9. References

[1] A. Stentz (1995) The focussed D* algorithm for real-time replanning. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Lawrence Erlbaum Associates Ltd, pp. 1652-1659.
[2] L.E. Kavraki, P. Svestka, J.-C. Latombe, M.H. Overmars (1996) Probabilistic roadmaps for path planning in high-dimensional configuration spaces. Robotics and Automation, IEEE Transactions on Robotics and Automation, vol. 12, pp. 566-580.
[3] S.M. LaValle, J.J. Kuffner Jr (2001) Rapidly-exploring random-trees: Progress and prospects, Algorithmic and computational Robotics: New Directions, B. R. Donald, K. M. Lynch, and D. Rus, editors. A. K. Peters, Wellesley, MA, pp. 293-308.
[4] Y.K. Hwang, N. Ahuja (1992) A potential field approach to path planning, IEEE Transactions on Robotics and Automation, vol. 8, pp. 23-32.
[5] R.J.Y McLeod, M.L Baart (1998) Geometry and interpolation of curves and surfaces, pp.1-180, UK: Cambridge University Press.
[6] Y. Suzuki, S. Thompson, S. Kagami (2009) Smooth path planning with pedestrian avoidance for wheeled robots: Implementation and evaluation. Proceedings of the 4th International Conference on Autonomous Robots and Agents, pp. 657-662.
[7] J. Villagra, V. Milanés, J.P. Rastelli, J. Godoy, E. Onieva (2012) Path and speed planning for smooth autonomous navigation, IEEE Intelligent Vehicles Symposium.
[8] K. Yang, D. Jung, S. Sukkarieh (2013) Continuous curvature path-smoothing algorithm using cubic B-spline spiral curves for non-holonomic robots, Advanced Robotics, vol. 27, pp. 247-258.
[9] L.E. Dubins (1957) On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents,
American Journal of Mathematics, vol. 79, pp. 497-516.

[10] R.A. Usmani (1987) On quadratic spline interpolation, BIT Numerical Mathematics, vol. 27, pp. 615-622.

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

[12] E. Magid, D. Keren, E. Rivlin, I. Yavneh (2006) Spline-based robot navigation. Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference, pp. 2296-2301.

[13] A. Piazzi, C. Guarino Lo. Bianco, M. Romano (2007) η3 Splines for the Smooth Path Generation of Wheeled Mobile Robots. IEEE Transactions on Robotics, vol. 23, pp. 1089-1095.

[14] J.F. Epperson (1987) On the Runge example. Amer. Math. Monthly, vol. 94, pp. 329-341.

[15] K. Shoemake (1994) Graphics Gems IV, Ch. Euler angle conversionm, pp.222-229, Boston: Academic Press Professional.

[16] H. Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki, S. Thrun (2005) Principles of robot motion: theory, algorithms, and implementations, pp.107-262, Cambridge, MA: MIT Press.