

# $G^2$ Continuity Smooth Path Planning using Cubic Polynomial Interpolation with Membership Function

Seong-Ryong Chang\* and Uk-Youl Huh<sup>†</sup>

**Abstract** – Path planning algorithms are used to allow mobile robots to avoid obstacles and find ways from a start point to a target point. The general path planning algorithm focused on constructing of collision free path. However, a high continuous path can make smooth and efficiently movements. To improve the continuity of the path, the searched waypoints are connected by the proposed polynomial interpolation. The existing polynomial interpolation methods connect two points. In this paper, point groups are created with three points. The point groups have each polynomial. Polynomials are made by matching the differential values and simple matrix calculation. Membership functions are used to distribute the weight of each polynomial at overlapped sections. As a result, the path has  $G^2$  continuity. In addition, the proposed method can analyze path numerically to obtain curvature and heading angle. Moreover, it does not require complex calculation and databases to save the created path.

**Keywords:** Continuous-curvature path, Geometric continuity, Interpolation, Path planning, Cubic polynomial, Mobile robot, Robot motion, Smooth path, Spline interpolation, Vehicles navigation

## 1. Introduction

The aim of path planning for mobile robots and vehicles is finding a path from the start to the target point with avoiding obstacles. The potential field method [1-3], ‘A\* and D\* algorithm’ [4-6] and ‘RRT algorithm’ [7, 8] are proposed for searching path.

The purposes of global path planning are the avoidance of stationary obstacles and solving mazes. On the other hand, considering the continuity, the path can be improved to use real movement, stable motion, smooth movement and optimal path. In addition, it can be used to analyze a planned path before driving.

The continuity of the path is defined as  $G^0, G^1, G^2 \dots G^n$  in geometry [9].  $G^0$  continuity is just contacted at each point by a straight line. The  $G^1$  continuity path is equal to the primary differential value at each point. The path will curve from  $G^1$  continuous. The  $G^2$  continuity path has an identical second-order differential value at each point. This means the same curvature at each point. In particular, the  $G^n$  continuity means continuity of the  $n^{\text{th}}$  order differential value at each point [10].

In order to use smooth path in the real environment, the continuity should be moreover  $G^2$  continuity. It means the  $G^2$  continuous path has a continuous velocity functions and acceleration functions. So this path has continuous-curvature. As a result, if the mobile robot and vehicles

follow the smooth path, the velocity is not changed abruptly in the path. Thus, they can be moved with the continuous velocity and acceleration without stop motion. This movement is efficient with time, energy and dynamics.

Kanayama and Hartman documented a definition of path curvature and the derivative path curvature [11]. Fraidchard and Ahuactzin suggested the continuous-curvature path planning for cars [12]. Lamiroux and Lammond proposed steering method for a car-like vehicle providing smooth paths [13]. Kito et al. described the path planning with maximum curvature, curvature derivative and curvature continuity [14]. Yang and Sukkarieh reported the continuous-curvature path-smoothing algorithm with efficient and analytical [15]. Villagra et al. indicated the efficiency and comfort of the smooth path and speed planning for the vehicle transportation system [16].

The obtained path from ‘RRT connect algorithm’ [7, 8] consists of ‘ $q_{init}$  (the initial configuration)’, waypoints, ‘ $q_{goal}$  (the goal configuration)’ and a line connecting the points in consecutive order [7, 17, 18]. A line connecting path is  $G^0$  continuity. ‘RRT connect algorithm’ [7, 8] can only solve mazes and stationary obstacle avoidance problems. The paths required to improve the  $G^2$  continuity to apply at the real environment.

Generally, in order to improve the continuity of path, the path should be reconstructed using an interpolation method considering the continuity. The interpolation method estimates  $f(x_{j+m})$  at  $x_{j+m}$  with only given  $x$  and  $f(x_i)$  [19, 20]. Polynomial interpolation and spline interpolation are used in general. Polynomial interpolation aims to find a polynomial including whole points [19]. The order of the polynomial is increased by the number of points. A high

<sup>†</sup> Corresponding Author: Dept. of Electrical Engineering, Inha University, Korea. (uyhuh@inha.ac.kr)

\* Dept. of Electrical Engineering, Inha University, Korea. (themir@naver.com)

Received: July 4, 2013; Accepted: October 27, 2014

order polynomial occurs ‘Runge’s phenomenon’ [21]. This type of polynomial is unsuitable for applying a large number of points. Spline interpolation produces a polynomial between points by the differential value at the points [19]. Cubic spline interpolation is widely used and satisfies the  $C^2$  continuous [10].

Polynomial interpolation and spline interpolation involve numerical analysis to determine the function of connecting characteristic points that satisfy the ‘ $x_0 < x_1 < x_2 < \dots < x_n$ ’ condition. In other words, they can be used under the condition of an always increasing  $x$  [19]. Therefore, normally interpolation methods are unsuitable for mobile robots and vehicles with complex movement. Therefore modified interpolation method has applied a mobile robots and vehicles. Piazzzi et al. attempted to smooth path planning using the quintic  $G^2$ -splines [22], the  $\eta^3$ -splines [23, 24] and the  $G^3$ -splines [25]. However, methods of Piazzzi have complex calculations such as high order polynomial and trigonometric functions which require the high performance hardware. Yang et al. proposed a continuous-curvature path-smoothing algorithm using cubic Bézier curves [26]. Bézier curve is a popular method to create curves for computer graphics [9]. Many smooth path studies have used this method, but the planned path is not visited entire waypoints because this method requires control points to decide the curvature. So some waypoints are used to the control points or require an algorithm to decide the control points. In the path planning method, the planned path must visit every waypoint. Therefore, a method using Bézier curves is not suitable to apply the smooth path planning.

On the movement of 2D plane,  $x$  is not changed in horizontal movement and  $y$  shows no variations in vertical movement on the 2D map. The parametric method is used in the present paper. The parametric method expresses  $x$  and  $y$  by  $u$ . The parameter  $u$  is the accumulation value of the distance between waypoints. This parameter is always a positive, increasing function and does not require the ‘ $x_0 < x_1 < x_2 < \dots < x_n$ ’ condition. The polynomial of  $x$  and  $y$  can be obtained using parameter  $u$  in any movement.

Spline interpolation makes a polynomial containing two points using point values and differential values. In addition, it makes another polynomial of the next two points in the same method. In this paper, the point group is made by three points. The quadratic polynomial and cubic polynomial include three points obtained by point values and differential values. The next group is made by two previous points and one new point. The quadratic polynomial and cubic polynomial about the new group is derived and extended to the next groups. The neighboring polynomial occurs in the overlapped section. Membership functions distribute the weight of each polynomial to make an improved continuity path in an overlapped section by parameter  $u$ . In addition, the planned smooth path could visit every waypoint unlike any other smooth path planning

methods.

The proposed method provides the difference values. The curvature and heading angle can be calculated using the differential values of  $x$  and  $y$ . These data can be applied to an analysis of the planned path and real movement. In addition, a controller and control algorithm use these data. The curvature is concerned with the dynamics. In other words, the robots and vehicles have the maximum velocity about curvature. If the controller and control algorithm use this data, the user can check a driving time with stable movement and the maximum velocity before real driving.

This paper is organized as follows: section 2 describes a method for obtaining a quadratic polynomial and cubic polynomial with three points. The difference between the proposed method and the general spline interpolation method is explained. Section 3 proposes a method for making a rotated polynomial using the parametric method. In section 4, the distribution of weight is explained in a polynomial overlapped section. In section 5, the continuity, curvature and heading angle functions are described. Section 6, the proposed method is simulated to find the alternation of the primary differential values, second-order difference values. These values prove that the proposed method can make a  $G^2$  continuity path. In addition, the curvature and heading angle are obtained by the proposed method. These provide a data for real driving and path analysis

## 2. Spline Interpolation and Polynomial Interpolation

### 2.1 Spline interpolation

Spline interpolation connects each point by a polynomial. A cubic polynomial interpolation is used in general. The method for obtaining the polynomial involves matching the primary differential value and second-order differential value at each point [19, 20]. The planned path is the matched curvature at each point that satisfies the  $G^2$  continuity.

The  $G^2$  continuity requires the matching of primary and second-order differential values at each point. Matching of the differential value is difficult in a quadratic polynomial because the second-order differential value is constant in a quadratic polynomial. Therefore, a quadratic polynomial can only use the  $G^1$  continuity graph. Thus, the minimum order for satisfying over  $G^1$  continuity is three, which means a cubic polynomial. This is why a cubic polynomial used generally.

The widely used spline interpolation requires a change in  $x$ .  $x$  should be increased.

$$x_0 < x_1 < x_2 < \dots < x_n \quad (1)$$

On the other hand, the paths of the robots or vehicles are not satisfied (1). They have a variety of movements.  $x$  or  $y$  is not variable in vertical or horizontal movement. Therefore, general spline interpolation is unsuitable for expressing the path trajectory.

**2.2 Quadratic polynomial interpolation**

The quadratic polynomial interpolation is a method to connect three points continuously. There are three points like as  $P_1, P_2$  and  $P_3$ . The polynomial connecting these three points can be expressed as equation (2).

$$P_n : (x_n, y_n) \in R^2$$

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

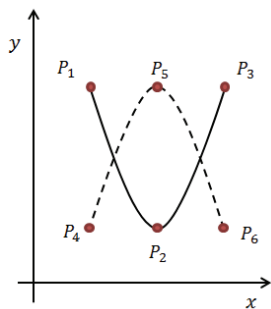
The primary and second-order differential values of quadratic polynomial are (3) and (4).

$$\frac{dy}{dx} = 2ax + b \tag{3}$$

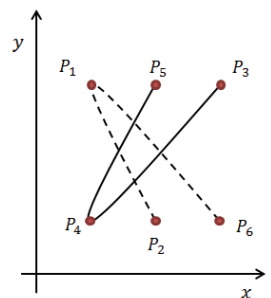
$$\frac{d^2y}{dx^2} = 2a \tag{4}$$

The curvature  $k$  is obtained using equation (5) by (3) and (4).

$$k = \frac{2a}{(1 + (2ax + b)^2)^{\frac{3}{2}}} \tag{5}$$



**Fig. 1.** Quadratic polynomial with 3 points ( $P_1 \rightarrow P_2 \rightarrow P_3$  and  $P_4 \rightarrow P_5 \rightarrow P_6$ )



**Fig. 2.** Graph of the rotated quadratic polynomial ( $P_2 \rightarrow P_1 \rightarrow P_6$  and  $P_5 \rightarrow P_4 \rightarrow P_3$ )

The graph of the quadratic polynomial has a concave or convex shape (Fig. 1). To use the quadratic polynomial interpolation, (1) should be satisfied. If (1) is not satisfied, the graph can be expressed, as shown in Fig. 2. In Fig. 2, the line is a rotation polynomial graph of connecting  $P_5, P_4$  and  $P_3$  in serial order, and the dot line is the rotation polynomial graph with  $P_2, P_1$  and  $P_6$  in consecutive order.

**2.3 Cubic polynomial interpolation**

The cubic polynomial is the third-order function. The basic form can be expressed as (6).

$$P_n : (x_n, y_n) \in R^2$$

$$y = ax^3 + bx^2 + cx + d \tag{6}$$

The second-order differential function is needed to check the  $G^2$  continuity. The primary and second-order differential functions are (7) and (8).

$$\frac{dy}{dx} = 3ax^2 + 2bx + c \tag{7}$$

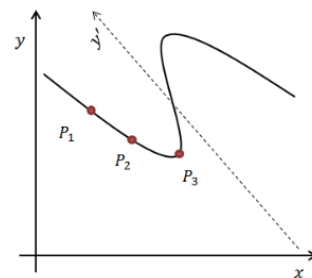
$$\frac{d^2y}{dx^2} = 6ax + 2b \tag{8}$$

The curvature  $k$  (9) can be calculated by (7) and (8).

$$k = \frac{6ax + 2b}{\left(1 + (3ax^2 + 2bx + c)^2\right)^{\frac{3}{2}}} \tag{9}$$

Fig. 3 shows the basic graph of cubic polynomial.

More than four conditions are needed to use a cubic polynomial interpolation by (6). On the other hand, a normal polynomial graph is unconformable to apply the path of a mobile robots or vehicles. The path should be shortest and realizable. In Fig. 3, there are convex and detour paths when connecting  $P_1$  and  $P_2$ . In particular, the path should be as short and realizable as possible with continuity. Therefore, if done correctly, the general cubic



**Fig. 3.** Graph of rotated cubic polynomial

polynomial interpolation is unsuitable for application to mobile robots or vehicles.

The proposed method is the rotated form of the cubic polynomial. Fig. 4 is the rotated graph of cubic polynomial. The graph can be expressed unsatisfied (1).

### 2.4 Rotated polynomial graph

The center point is set to obtain a polynomial of three points connected in serial order. Three points exist  $P_{n-1}$ ,  $P_n$  and  $P_{n+1}$ .  $P_n$  is the center point. If the three points connecting the graph are rotated centered on  $P_n$ , two cases exist. The first case is a straight line on  $P_{n-1}$  and  $P_{n+1}$ . In this case, the curvature is unlimited and is the simplest way of movement. The second case is the curve on  $P_{n-1}$  and  $P_{n+1}$ . In this case, the graph will be convex or concave. The optimal smooth path is a convex or a concave graph by maintaining  $G^2$  continuity.

A quadratic polynomial is a polynomial including three points with continuity. If it is rotated, the center point will approach the vertex.

(10) are the general rotation functions.  $\theta$  is the rotation angle.

$$\begin{aligned} x &= x' \cos\theta - y' \sin\theta \\ y &= x' \sin\theta + y' \cos\theta \end{aligned} \tag{10}$$

(10) are impossible to obtain a graph that connects each point in consecutive order. In addition, it is difficult to express with a quadratic polynomial or cubic polynomial, such as (11) and (12). (11) and (12) should be used to find  $a$ ,  $b$ ,  $c$ ,  $d$  and  $\theta$ . On the other hand, trigonometric functions hinder solutions from being found.

$$x' \sin\theta + y' \cos\theta = a(x' \cos\theta - y' \sin\theta)^2 + b(x' \cos\theta - y' \sin\theta) + c \tag{11}$$

$$x' \sin\theta + y' \cos\theta = a(x' \cos\theta - y' \sin\theta)^3 + b(x' \cos\theta - y' \sin\theta)^2 + c(x' \cos\theta - y' \sin\theta) + d \tag{12}$$

Fig. 5 shows three cases, where  $P_1$ ,  $P_2$  and  $P_3$  are connected. The sequential path cannot be derived by (11) or (12). Therefore, another method is needed to connect the points sequentially and rotated graph. This method should be able to apply a quadratic polynomial and cubic

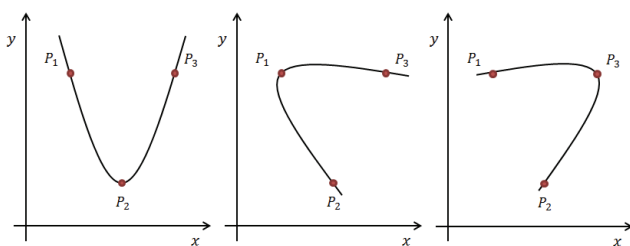


Fig. 4. Cases of three points connected quadratic polynomial graph

polynomial.

## 3. Parametric Quadratic Polynomial

### 3.1 Quadratic polynomial with parametric method

In this section, the quadratic polynomial and cubic polynomial are expressed by parameter  $u$ . If the three points are existed, the center point should be approached the vertex of a graph to rotate the polynomial and the points should be connected in serial order. To rotate the polynomial,  $x$  and  $y$  are separated by parameter  $u$ . The parameter  $u$  is the accumulation value of the straight line distance between points. As a result, parameter  $u$  is always a positive and increasing value and the graph has rotated shape.  $u$  can express to (13) with three points.

$$\begin{aligned} u_1 &= 0 \\ u_2 &= u_1 + \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \\ u_3 &= u_2 + \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2} \end{aligned} \tag{13}$$

(2) can change to (14) and (15) using parameter  $u$ .

$$x(u) = a_x u^2 + b_x u + c_x \tag{14}$$

$$y(u) = a_y u^2 + b_y u + c_y \tag{15}$$

$a_x$ ,  $b_x$ ,  $c_x$ ,  $a_y$ ,  $b_y$  and  $c_y$  can be calculated by (16) and (17).

$$\begin{pmatrix} a_x \\ b_x \\ c_x \end{pmatrix} = \begin{pmatrix} u_1^2 & u_1 & 1 \\ u_2^2 & u_2 & 1 \\ u_3^2 & u_3 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} x(u_1) \\ x(u_2) \\ x(u_3) \end{pmatrix} \tag{16}$$

$$\begin{pmatrix} a_y \\ b_y \\ c_y \end{pmatrix} = \begin{pmatrix} u_1^2 & u_1 & 1 \\ u_2^2 & u_2 & 1 \\ u_3^2 & u_3 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} y(u_1) \\ y(u_2) \\ y(u_3) \end{pmatrix} \tag{17}$$

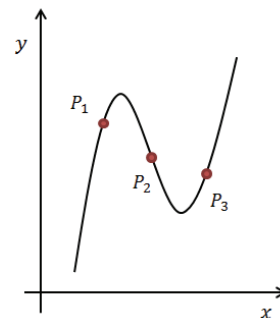


Fig. 5. Graph of the cubic polynomial with three points

Primary and second-order differential value of (16) and (17) are expressed as (18) and (19).

$$\frac{dx}{du} = 2a_x u + b_x, \quad \frac{d^2x}{du^2} = 2a_x \quad (18)$$

$$\frac{dy}{du} = 2a_y u + b_y, \quad \frac{d^2y}{du^2} = 2a_y \quad (19)$$

(16) and (17) make a polynomial with three points. The three points are then applied to (16) and (17) in more than three points path.  $P_{n-1}$ ,  $P_n$  and  $P_{n+1}$  are used to obtain  $n^{\text{th}}$  polynomial.  $(n + 1)^{\text{th}}$  polynomial is made by  $P_n$ ,  $P_{n+1}$  and  $P_{n+2}$ . (13) is changed to the general function as (20). (16) and (17) can express to (21) and (22) with  $(n + 1)^{\text{th}}$  points.

$$u_1 = 0$$

$$u_n = \sum_{n=2}^m \sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2}, \quad (2 \leq n) \quad (20)$$

$$\begin{pmatrix} a_{x_{n-1}} \\ b_{x_n} \\ c_{x_{n+1}} \end{pmatrix} = \begin{pmatrix} u_{n-1}^2 & u_{n-1} & 1 \\ u_n^2 & u_n & 1 \\ u_{n+1}^2 & u_{n+1} & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} x(u_{n-1}) \\ x(u_n) \\ x(u_{n+1}) \end{pmatrix} \quad (21)$$

$$\begin{pmatrix} a_{y_{n-1}} \\ b_{y_n} \\ c_{y_{n+1}} \end{pmatrix} = \begin{pmatrix} u_{n-1}^2 & u_{n-1} & 1 \\ u_n^2 & u_n & 1 \\ u_{n+1}^2 & u_{n+1} & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} y(u_{n-1}) \\ y(u_n) \\ y(u_{n+1}) \end{pmatrix} \quad (22)$$

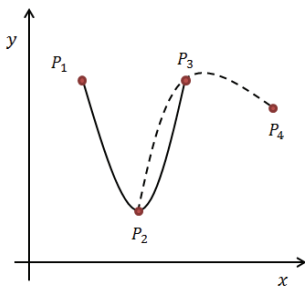
$$(2 \leq n \leq m - 1)$$

$n^{\text{th}}$  points are needed in a  $(n - 2)^{\text{th}}$  quadratic polynomial. Every polynomial contacts the whole points and each polynomial had  $G^2$  continuity including three points  $P_{n-1}$ ,  $P_n$  and  $P_{n+1}$ .

For example, two polynomials are the existence with four points. Each polynomial has the overlapped section (Fig. 6).

### 3.2 Cubic polynomial with parametric method

Parameter  $u$  consists of quadratic polynomial obtained



**Fig. 6.** Two quadratic polynomial graph including four points

by (16) and (17). The quadratic polynomial can be expressed using parameter  $u$ . The polynomial is in contact with three points in serial order and has a rotated graph form. Each polynomial has  $G^2$  continuity. On the other hand, it is unsure in more than three points. To make  $G^2$  continuity, a second-order differential value is matched at each point. Therefore a higher order polynomial is needed.

(14) and (15) can be expanded to a cubic polynomial. (23) and (24) are the basic form of cubic polynomial using parameter  $u$ .

$$x(u) = a_x u^3 + b_x u^2 + c_x u + d_x \quad (23)$$

$$y(u) = a_y u^3 + b_y u^2 + c_y u + d_y \quad (24)$$

(23) and (24) will be (25) and (26) in more than three points case.  $P_{x_n}(u)$  is  $n^{\text{th}}$   $x$ -axis polynomial including  $x_{n-1}$ ,  $x_n$  and  $x_{n+1}$ .  $P_{y_n}(u)$  is  $n^{\text{th}}$   $y$ -axis polynomial including  $y_{n-1}$ ,  $y_n$  and  $y_{n+1}$ .

$$P_{x_n}(u) = a_{x_n} u^3 + b_{x_n} u^2 + c_{x_n} u + d_{x_n} \quad (25)$$

$$P_{y_n}(u) = a_{y_n} u^3 + b_{y_n} u^2 + c_{y_n} u + d_{y_n} \quad (26)$$

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

(25) and (26) are cubic polynomials including  $P_{n-1}$ ,  $P_n$ ,  $P_{n+1}$  in consecutive order.

$a$ ,  $b$ ,  $c$  and  $d$  of (25) and (26) cannot be obtained by three points because another condition is needed to match the second-order differential value at each point. (27) and (28) are added to the conditions.

$$\frac{d^2 P_{x_{n-1}}(u)}{du^2} = \frac{d^2 P_{x_n}(u)}{du^2} \quad (27)$$

$$\frac{d^2 P_{y_{n-1}}(u)}{du^2} = \frac{d^2 P_{y_n}(u)}{du^2} \quad (28)$$

(27) and (28) apply to (25) and (26). (29) and (30) are then obtained.

$$6a_{x_{n-1}} u_{n-1} + b_{x_{n-1}} = 6a_{x_n} u_{n-1} + 2b_{x_n} \quad (29)$$

$$6a_{y_{n-1}} u_{n-1} + b_{y_{n-1}} = 6a_{y_n} u_{n-1} + 2b_{y_n} \quad (30)$$

(25), (26), (27) and (28) can be combined into matrix (31).

$$\begin{pmatrix} x_{n-1} \\ x_n \\ x_{n+1} \\ 6a_{x_{n-1}} u_{n-1} + 2b_{x_{n-1}} \end{pmatrix} = \begin{pmatrix} u_{n-1}^3 & u_{n-1}^2 & u_{n-1} & 1 \\ u_n^3 & u_n^2 & u_n & 1 \\ u_{n+1}^3 & u_{n+1}^2 & u_{n+1} & 1 \\ 6u_{n-1} & 2 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} a_{x_n} \\ b_{x_n} \\ c_{x_n} \\ d_{x_n} \end{pmatrix}$$

$$\begin{pmatrix} y_{n-1} \\ y_n \\ y_{n+1} \\ 6a_{y_{n-1}} u_{n-1} + 2b_{y_{n-1}} \end{pmatrix} = \begin{pmatrix} u_{n-1}^3 & u_{n-1}^2 & u_{n-1} & 1 \\ u_n^3 & u_n^2 & u_n & 1 \\ u_{n+1}^3 & u_{n+1}^2 & u_{n+1} & 1 \\ 6u_{n-1} & 2 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} a_{y_n} \\ b_{y_n} \\ c_{y_n} \\ d_{y_n} \end{pmatrix}$$

$$(2 \leq n \leq m - 1) \quad (31)$$

(31) can change to (32) via an inverse matrix.

$$\begin{pmatrix} a_{x_n} \\ b_{x_n} \\ c_{x_n} \\ d_{x_n} \end{pmatrix} = \begin{pmatrix} u_{n-1}^3 & u_{n-1}^2 & u_{n-1} & 1 \\ u_n^3 & u_n^2 & u_n & 1 \\ u_{n+1}^3 & u_{n+1}^2 & u_{n+1} & 1 \\ 6u_{n-1} & 2 & 0 & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} x_{n-1} \\ x_n \\ x_{n+1} \\ 6a_{x_{n-1}}u_{n-1} + 2b_{x_{n-1}} \end{pmatrix}$$

$$\begin{pmatrix} a_{y_n} \\ b_{y_n} \\ c_{y_n} \\ d_{y_n} \end{pmatrix} = \begin{pmatrix} u_{n-1}^3 & u_{n-1}^2 & u_{n-1} & 1 \\ u_n^3 & u_n^2 & u_n & 1 \\ u_{n+1}^3 & u_{n+1}^2 & u_{n+1} & 1 \\ 6u_{n-1} & 2 & 0 & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} y_{n-1} \\ y_n \\ y_{n+1} \\ 6a_{y_{n-1}}u_{n-1} + 2b_{y_{n-1}} \end{pmatrix}$$

$(2 \leq n \leq m - 1)$  (32)

$a_{x_{n-1}}, b_{x_{n-1}}, a_{y_{n-1}}, b_{y_{n-1}}$  and  $u_{n-1}$  can be obtained by  $P_n$ . On the other hand, they do not exist if  $(n = 2)$ . Therefore, polynomial of  $P_2$  must use a quadratic polynomial by (21) and (22). The range of  $n$  values is changed, and (32) can be expressed as (33).

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

$$\begin{pmatrix} a_{y_n} \\ b_{y_n} \\ c_{y_n} \\ d_{y_n} \end{pmatrix} = \begin{pmatrix} u_{n-1}^3 & u_{n-1}^2 & u_{n-1} & 1 \\ u_n^3 & u_n^2 & u_n & 1 \\ u_{n+1}^3 & u_{n+1}^2 & u_{n+1} & 1 \\ 6u_{n-1} & 2 & 0 & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} y_{n-1} \\ y_n \\ y_{n+1} \\ 6a_{y_{n-1}}u_{n-1} + 2b_{y_{n-1}} \end{pmatrix}$$

$(2 < n \leq m - 1)$  (33)

As a result, the  $P_{x_2}$  and  $P_{y_2}$  polynomials are connecting  $P_1, P_2$  and  $P_3$  by a quadratic polynomial. In addition,  $n$  is expanded to  $n + 1, n + 2, \dots, m - 1$  to obtain a cubic polynomial with matching second-order differential value at each point.

The parametric method provides a variation of  $x$  and  $y$ . They are connected with parameter  $u$ . This method is useful to express complex movements.

#### 4. Polynomial Interpolation with Membership Function

Polynomials of  $m^{\text{th}}$  points are consist of quadratic polynomials and cubic polynomials with a matching second-order differential value. On the other hand, there are overlapped sections between  $n^{\text{th}}$  polynomial and  $(n + 1)^{\text{th}}$  polynomial. Membership functions distribute the weight of the polynomial in overlapped sections. They are functions of the parameter  $u$ .

$X(u)$  is a group of  $P_{x_n}$  and  $Y(u)$  is a group of  $P_{y_n}$ . The smooth path is denoted as (34),

$$P(X(u), Y(u)) \tag{34}$$

The weight of the  $n^{\text{th}}$  polynomial and  $(n + 1)^{\text{th}}$

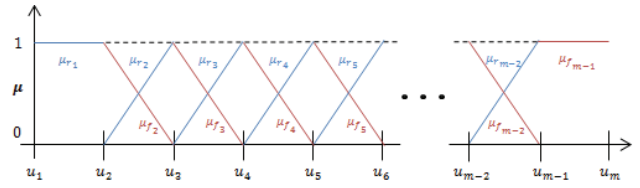


Fig. 7. Graph of the membership functions

polynomial are adjusted by the distance between the points in overlapped sections. In particular, the  $n^{\text{th}}$  polynomial has the largest weight at the  $n^{\text{th}}$  point and decreases as it approaches the  $(n + 1)^{\text{th}}$  point. The weight of the  $(n + 1)^{\text{th}}$  polynomial is zero at  $n^{\text{th}}$  point, but it increases by approaching the  $(n + 1)^{\text{th}}$  point. (35) and (36) are the membership functions expressing parameter  $u$ .

$$\mu_{r_n}(u) = \frac{(u-u_n)}{(u_{n+1}-u_n)}, (1 \leq n \leq m - 1) \tag{35}$$

$$\mu_{f_n}(u) = \frac{(u_{n+1}-u)}{(u_{n+1}-u_n)}, (1 \leq n \leq m - 1) \tag{36}$$

In Fig. 7,  $\mu_r$  is zero at  $P_n$  and 1 at  $P_{n+1}$ . At the same time,  $\mu_f$  is 1 at  $P_n$  and zero at  $P_{n+1}$ . (37) is (34) combined (35) and (36).

$$X(u) = \begin{cases} \mu_{r_1}(u)Px_2(u) + \mu_{f_1}(u)Px_1(u) : u_1 \leq u \leq u_2 \\ \mu_{r_2}(u)Px_3(u) + \mu_{f_2}(u)Px_2(u) : u_2 \leq u \leq u_3 \\ \mu_{r_3}(u)Px_4(u) + \mu_{f_3}(u)Px_3(u) : u_3 \leq u \leq u_4 \\ \vdots \\ \mu_{r_{m-2}}(u)Px_{m-1}(u) + \mu_{f_{m-2}}(u)Px_{m-2}(u) : u_{m-2} \leq u \leq u_{m-1} \\ \mu_{r_{m-1}}(u)Px_m(u) + \mu_{f_{m-1}}(u)Px_{m-1}(u) : u_{m-1} \leq u \leq u_m \end{cases}$$

$$Y(u) = \begin{cases} \mu_{r_1}(u)Py_2(u) + \mu_{f_1}(u)Py_1(u) : u_1 \leq u \leq u_2 \\ \mu_{r_2}(u)Py_3(u) + \mu_{f_2}(u)Py_2(u) : u_2 \leq u \leq u_3 \\ \mu_{r_3}(u)Py_4(u) + \mu_{f_3}(u)Py_3(u) : u_3 \leq u \leq u_4 \\ \vdots \\ \mu_{r_{m-2}}(u)Py_{m-1}(u) + \mu_{f_{m-2}}(u)Py_{m-2}(u) : u_{m-2} \leq u \leq u_{m-1} \\ \mu_{r_{m-1}}(u)Py_m(u) + \mu_{f_{m-1}}(u)Py_{m-1}(u) : u_{m-1} \leq u \leq u_m \end{cases}$$

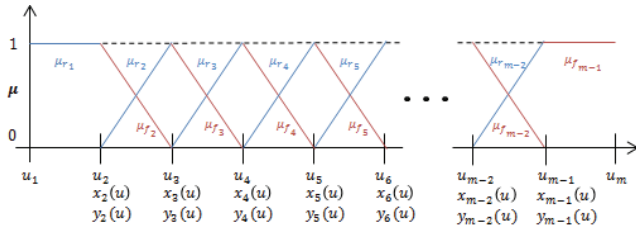
$$\begin{aligned} x_1(u) = x_m(u) = y_1(u) = y_m(u) = 0 \\ \mu_{r_1}(u) = \mu_{f_{m-1}}(u) = 1 \end{aligned} \tag{37}$$

The general form can be expressed as (38).

$$X(u) = \begin{cases} Px_2(u) : u_1 \leq u \leq u_2 \\ \mu_{r_n}(u)Px_{n+1}(u) + \mu_{f_n}(u)Px_n(u), (2 \leq n < m - 1) : \\ \quad u_2 \leq u \leq u_{m-1} \\ Px_{m-1}(u) : u_{m-1} \leq u \leq u_m \end{cases}$$

$$Y(u) = \begin{cases} Py_2(u) : u_1 \leq u \leq u_2 \\ \mu_{r_n}(u)Py_{n+1}(u) + \mu_{f_n}(u)Py_n(u), (2 \leq n < m - 1) : \\ \quad u_2 \leq u \leq u_{m-1} \\ Py_{m-1}(u) : u_{m-1} \leq u \leq u_m \end{cases} \tag{38}$$

$P_{x_2}(u)$  and  $P_{y_2}(u)$  effect to the first section ( $P_1$  to  $P_2$ )



**Fig. 8.** Membership functions and polynomials

and  $P_{x_{m-1}}(u)$  and  $P_{y_{m-1}}(u)$  make the path in the last section ( $P_{m-1}$  to  $P_m$ ). On the other hand, memberships function and polynomials are combined in the other sections.

Fig. 8 is the weight variation of the polynomial by parameter  $u$ .

The membership functions change the weight of each polynomial by a parameter  $u$  in the overlapped section. As a result, separated smooth paths are combined to one smooth path that maintain the continuity.

### 5. Continuity, Curvature and Heading Angle

The proposed method can calculate the curvature and heading angle. This method can be used to analyze the planned path and background data of real movements.

The primary differential value and second-order differential value are needed to obtain the curvature. They can be expressed as (18) and (19) in the quadratic polynomial. (39) are differential values of  $P(X(u), Y(u))$  using (18) and (19) in case of the quadratic polynomial.

$$\begin{aligned} \frac{dP_{x_n}(u)}{du} &= 2a_{x_n}u + b_{x_n} \\ \frac{dP_{y_n}(u)}{du} &= 2a_{y_n}u + b_{y_n} \\ \frac{d^2P_{x_n}(u)}{du^2} &= 2a_{x_n} \\ \frac{d^2P_{y_n}(u)}{du^2} &= 2a_{y_n} \end{aligned} \quad (39)$$

(40) are the primary and second-order differential values of the cubic polynomial by (29) and (30).

$$\begin{aligned} \frac{dP_{x_n}(u)}{du} &= 3a_{x_n}u^2 + 2b_{x_n}u + c_{x_n} \\ \frac{dP_{y_n}(u)}{du} &= 3a_{y_n}u^2 + 2b_{y_n}u + c_{y_n} \\ \frac{d^2P_{x_n}(u)}{du^2} &= 6a_{x_n}u + 2b_{x_n} \\ \frac{d^2P_{y_n}(u)}{du^2} &= 6a_{y_n}u + 2b_{y_n} \end{aligned} \quad (40)$$

(38) can change to (41) and (42) to find differential values.

$$\begin{aligned} \frac{d}{du}X(u) &= \begin{cases} \frac{dP_{x_2}(u)}{du} : u_1 \leq u \leq u_2 \\ \mu_{r_n}(u) \frac{dP_{x_{n+1}}(u)}{du} + \mu_{f_n}(u) \frac{dP_{x_n}(u)}{du}, (2 \leq n < m-1) : \\ u_2 \leq u \leq u_{m-1} \\ \frac{dP_{x_{m-1}}(u)}{du} : u_{m-1} \leq u \leq u_m \end{cases} \\ \frac{d}{du}Y(u) &= \begin{cases} \frac{dP_{y_2}(u)}{du} : u_1 \leq u \leq u_2 \\ \mu_{r_n}(u) \frac{dP_{y_{n+1}}(u)}{du} + \mu_{f_n}(u) \frac{dP_{y_n}(u)}{du}, (2 \leq n < m-1) : \\ u_2 \leq u \leq u_{m-1} \\ \frac{dP_{y_{m-1}}(u)}{du} : u_{m-1} \leq u \leq u_m \end{cases} \quad (41) \end{aligned}$$

$$\begin{aligned} \frac{d^2}{du^2}X(u) &= \begin{cases} \frac{d^2P_{x_2}(u)}{du^2} : u_1 \leq u \leq u_2 \\ \mu_{r_n}(u) \frac{d^2P_{x_{n+1}}(u)}{du^2} + \mu_{f_n}(u) \frac{d^2P_{x_n}(u)}{du^2}, (2 \leq n < m-1) : \\ u_2 \leq u \leq u_{m-1} \\ \frac{d^2P_{x_{m-1}}(u)}{du^2} : u_{m-1} \leq u \leq u_m \end{cases} \\ \frac{d^2}{du^2}Y(u) &= \begin{cases} \frac{d^2P_{y_2}(u)}{du^2} : u_1 \leq u \leq u_2 \\ \mu_{r_n}(u) \frac{d^2P_{y_{n+1}}(u)}{du^2} + \mu_{f_n}(u) \frac{d^2P_{y_n}(u)}{du^2}, (2 \leq n < m-1) : \\ u_2 \leq u \leq u_{m-1} \\ \frac{d^2P_{y_{m-1}}(u)}{du^2} : u_{m-1} \leq u \leq u_m \end{cases} \quad (42) \end{aligned}$$

$P_{x_2}(u)$  and  $P_{y_2}(u)$  are expressed as the quadratic polynomial connecting  $P_1$ ,  $P_2$  and  $P_3$ . The cubic polynomials are shown in other sections. This can maintain the  $G^2$  continuity by (31) and (32).

The curvature  $k$  can be obtained using (41) and (42).

$$k = \frac{|X'Y'' - Y'X''|}{(X'^2 + Y'^2)^{\frac{3}{2}}} \quad (43)$$

The heading angle can be calculated using (44).

$$\theta = \tan^{-1} \left( \frac{y_n - y_{n-1}}{x_n - x_{n-1}} \right) = \tan^{-1} \left( \frac{dY(u)}{dX(u)} \right) \quad (44)$$

The differential values are used to analysis of the planned path for the velocity and acceleration control. The

curvature concerns the maximum velocity. In addition, the heading angle indicates the direction of robots and vehicles, and they use to obtain an angular speed.

### 6. Simulation

In this section, two simulations are shown. The first simulation is a comparison with a ‘Quintic  $G^2$ -splines’ [22]. This spline method connects each waypoint using the fifth-order functions and trigonometric functions. The proposed method created a smooth path using given waypoints. The second simulation shows the smooth path from an ‘RRT algorithm’ searched path. This path has  $G^0$  continuity. The proposed method improved the continuity to  $G^2$  continuity.

#### 6.1 Case of [22] ‘Quintic $G^2$ -splines for the iterative steering of vision-based autonomous vehicles’

[22] reported motion planning using the quintic  $G^2$ -spline. The  $G^2$  continuity path is shown with the five points using the proposed method. Table 1 is the given waypoints of [22].

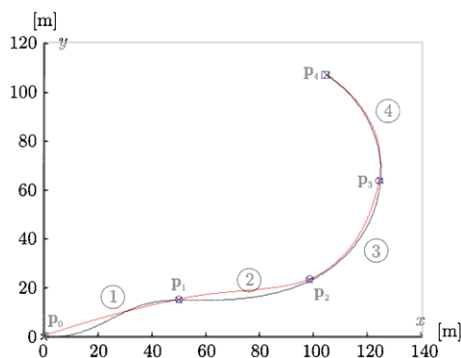
**Table 1.** Points of [22]

	$P_0$	$P_1$	$P_2$	$P_3$	$P_4$
$x$	0	50	98.75	124.67	104.72
$y$	0	15	23.19	63.53	107.12

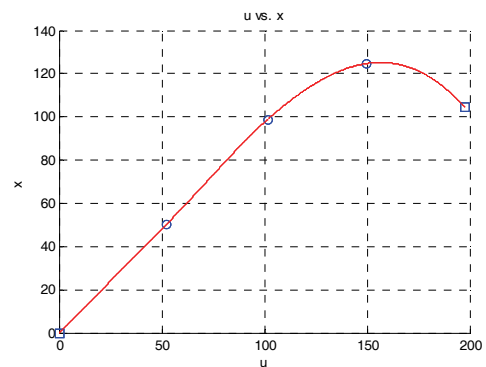
The proposed method can be applied to [22]. Fig. 9 shows the results by (38). In [22], paths between each waypoint consist of quintic function and trigonometric functions. However, the proposed method can create the path by using the cubic polynomial.

Figs. 10 and 11 show the variation of  $x$  and  $y$  by  $u$ . These are continuous shape. To check continuity, the differential values should be observed.

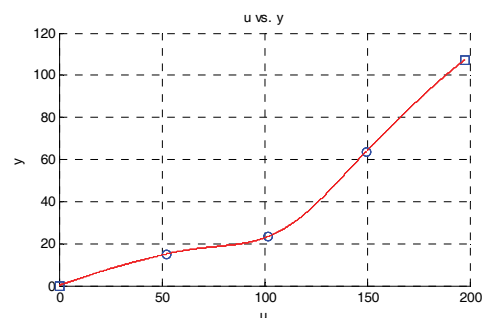
Figs. 12 and 13 show the primary differential values of  $X(u)$  and  $Y(u)$  by (41). The graph has continuous shape. This means that the path has  $G^1$  continuity at least.



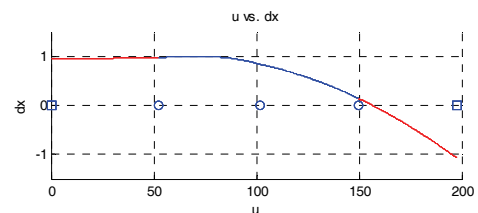
**Fig. 9.** [22] (black line) and the cubic polynomial interpolation with a membership function (red line)



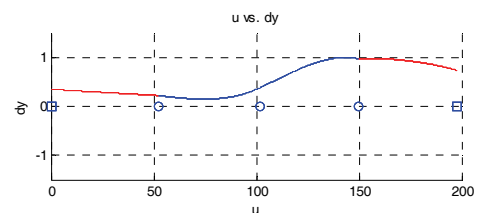
**Fig. 10.**  $u$  vs.  $X(u)$



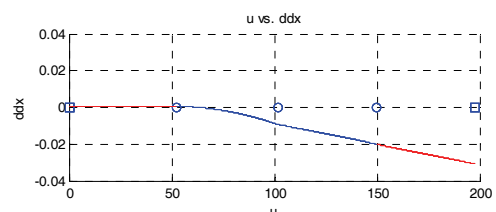
**Fig. 11.**  $u$  vs.  $Y(u)$



**Fig. 12.**  $u$  vs.  $dX(u)$



**Fig. 13.**  $u$  vs.  $dY(u)$



**Fig. 14.**  $u$  vs.  $d^2X(u)$

Therefore, the planned path is continuous on the velocity.

To confirm  $G^2$  continuity, the second-order differential values are checked. Figs. 14 and 15 show the second-order differential values by (42).



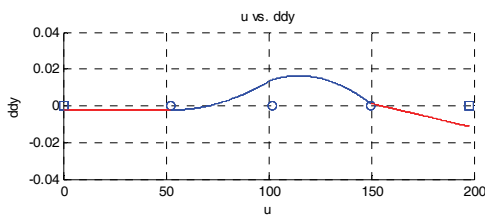


Fig. 15.  $u$  vs.  $d^2Y(u)$

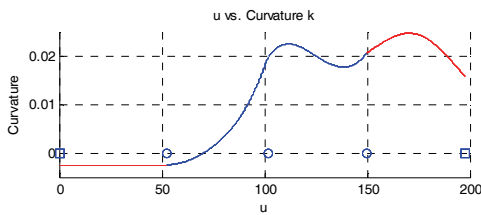


Fig. 16.  $u$  vs. curvature  $k$

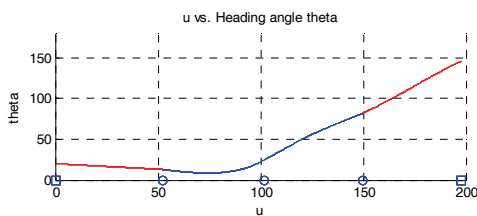


Fig. 17.  $u$  vs. heading angle  $\theta$

Figs. 14 and 15 have not open sections with any points. These are show that path is  $G^2$  continuity. These graphs indicate that the planned path has acceleration continuity.

Fig. 16 shows the variation of curvature  $k$  by (43). This graph is continuous and closed in whole sections. These are features of the  $G^2$  continuity path. This data can be applied to control of the velocity and acceleration for real driving.

Fig. 17 shows the variation of heading angle  $\theta$  using (41) and (44).  $\theta$  can be important data of the driving simulation, real driving and analyze of planned path.

Figs. 10-15 show that the path is the  $G^2$  continuity. Fig. 9 compares path of [22] with the proposed path. The figure shows some difference between the path of [22] and the proposed path in  $P_1$  to  $P_2$  section. This is because path of [22] has an internal heading angle at  $P_1$ , but the proposed path has no internal heading angle. The internal heading angle is the constraint of simulation. On the other hands, it is not needed in real driving.

The curvature graph (Fig. 16) is continuous, and Fig. 14 and Fig. 15 are not open in whole sections. In addition, the variation of heading angle can be obtained.

As a result, proposed path has  $G^2$  continuity with checking variation of differential values and curvature.

## 6.2 Case of the roadmap

This simulation is an improving continuity. The proposed method uses only waypoints. If searching algorithm obtains

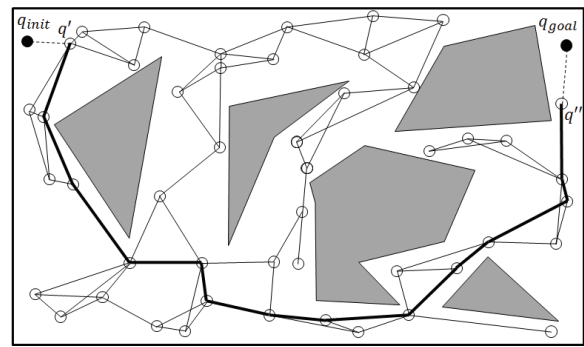


Fig. 18. The path from ‘RRT algorithm’ [18], an example of how to solve a query with the roadmap

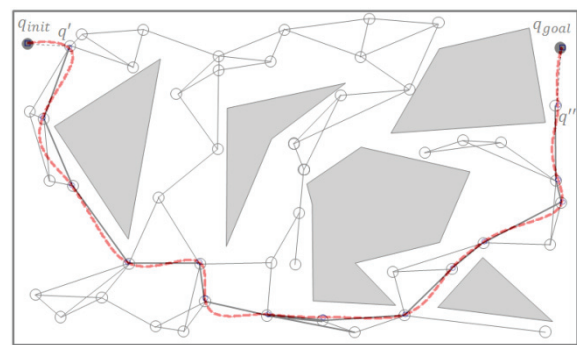


Fig. 19. The path from ‘RRT algorithm’ [18] vs.  $G^2$  continuity path using proposed method

a collision-free linear path and waypoints, the proposed method can improve to the  $G^2$  continuity using searched waypoints.

Fig. 18 shows a path from ‘RRT algorithm’ [7, 8]. This is a simply connected path at each point with  $G^0$  continuity. The proposed method was applied to Figs. 16 and 18 can be express as Fig. 19. This figure shows the  $G^2$  continuity smooth path including every waypoint. In order to check the continuity of the path, the differential values were obtained.

Figs. 20 and 21 show the variation in  $X(u)$  and  $Y(u)$  by (38).

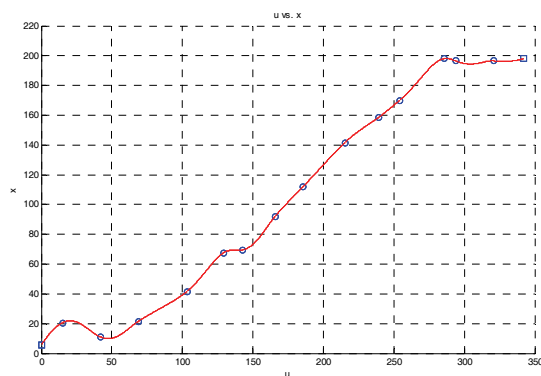


Fig. 20.  $u$  vs.  $X(u)$

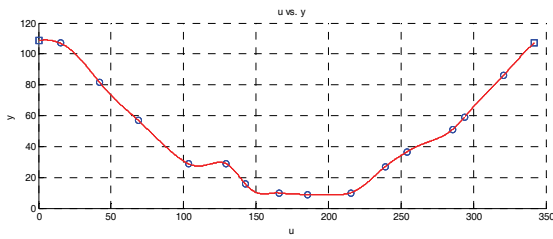


Fig. 21.  $u$  vs.  $Y(u)$

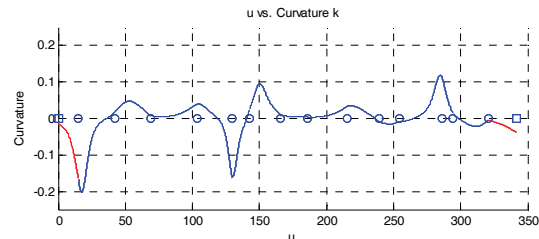


Fig. 26.  $u$  vs. curvature  $k$

Fig. 27 shows the variation of the heading angle by (44).

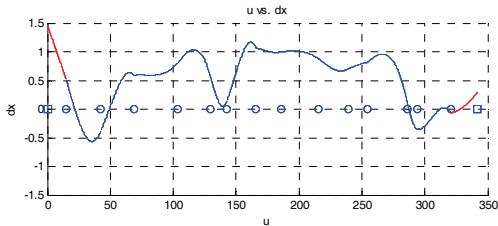


Fig. 22.  $u$  vs.  $dX(u)$

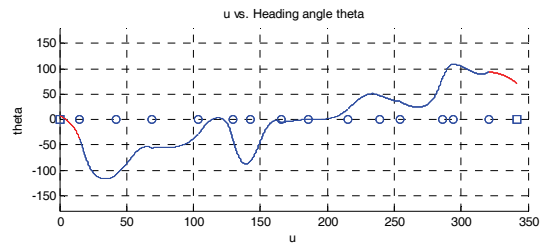


Fig. 27.  $u$  vs. heading angle  $\theta$

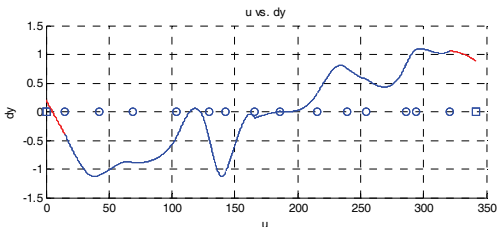


Fig. 23.  $u$  vs.  $dY(u)$

The  $G^0$  continuity path using the ‘RRT algorithm’ [7, 8] change to the  $G^2$  continuity path by the proposed method. The proposed method could obtain the change in curvature and heading angle data. This simulation indicates the combination with the searching algorithm. The searching algorithm cannot create the smooth path. However, it can provide waypoints. The proposed method uses these waypoints and the process of finding a smooth path is simple.

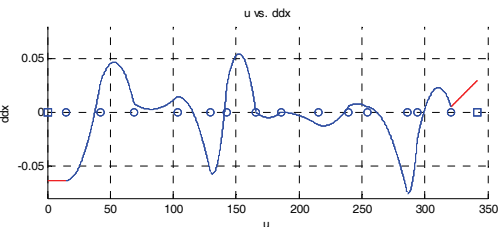


Fig. 24.  $u$  vs.  $d^2X(u)$

## 7. Conclusion

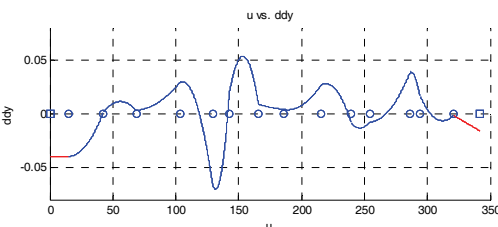


Fig. 25.  $u$  vs.  $d^2Y(u)$

Most searching algorithms do not consider the continuity of a path. The proposed method aimed to improve the continuity. Polynomials derived using point groups consisted of three points. They were extended to other groups and used to create polynomials. The point groups considered the differential values at the points. Differential value matching improved the continuity, and a parametric method was used to obtain a polynomial with complex movement. The parameter functions consisted of a linear distance variable  $u$ . Membership functions were used to distribute the weight of each polynomial’s influence. As a result, the  $G^2$  continuity path was obtained in the whole sections. The proposed method could analyze the path numerically. This shows that the curvature and heading angle can be found using differential values. The curvature and heading angle are useful data to design a control algorithm considering dynamics.

The proposed method was verified by two simulations. The first simulation compared with the quintic  $G^2$ -spline. A similar result was obtained and could analyze variation

Figs. 22-25 show the variation in the primary differential values and second-order differential values by (41) and (42). They have not open sections in every point. As a result, the proposed path is  $G^2$  continuity that just checking these graphs.

Fig. 26 shows the change in curvature by (43).

of  $x$ ,  $y$ , the curvature and the heading angle. The second simulation produced a  $G^2$  continuity path from 'RRT algorithm' path. The 'RRT algorithm' could obtain only a  $G^0$  continuity path. On the other hand, with the 'RRT algorithm', a path could change to the  $G^2$  continuity path using the proposed method.

When there were waypoints by the searching algorithms, the continuous path between waypoints could be obtained calculations using the proposed method. An increase in the volume of data is not necessary and the path can be updated upon a change in environment. This is because the proposed method uses a simple matrix form and does not use trigonometric functions or high order functions which require high performance hardware and complex calculations compare with other smooth path planning algorithms. As a result, the proposed method can apply to simple system. In addition, the control algorithm could be stable and simple.

The proposed method can be applied to mobile robots, vehicles, game algorithms and computer graphics. The method can be used in flying robots and aircraft if it can be expanded to 3D space. The method of the expanding points group can be used to improve the continuity of the path in local path planning.

### Acknowledgements

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

### References

- [1] 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.
- [2] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *Robotics and Automation, IEEE Transactions on*, vol. 16, pp. 615-620, 2000.
- [3] Y. Wang and G. S. Chirikjian, "A new potential field method for robot path planning," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 2000, pp. 977-982.
- [4] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, pp. 100-107, 1968.
- [5] A. Stentz, "The focussed D\* algorithm for real-time replanning," in *International Joint Conference on Artificial Intelligence*, 1995, pp. 1652-1659.
- [6] S. Koenig and M. Likhachev, "D\* Lite," in *Proceedings of the national conference on artificial intelligence*, 2002, pp. 476-483.
- [7] J. J. Kuffner Jr and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 2000, pp. 995-1001.
- [8] S. M. LaValle, *Planning algorithms*: Cambridge university press, 2006.
- [9] G. E. Farin, *Curves and surfaces for CAGD [electronic resource]: a practical guide*: Morgan Kaufmann, 2002.
- [10] B. A. Barsky and A. D. DeRose, "Geometric continuity of parametric curves," 1984.
- [11] Y. Kanayama and B. I. Hartman, "Smooth local path planning for autonomous vehicles," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, 1989, pp. 1265-1270.
- [12] T. Fraichard and J.-M. Ahuactzin, "Smooth path planning for cars," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 2001, pp. 3722-3727.
- [13] F. Lamiroux and J.-P. Lammond, "Smooth motion planning for car-like vehicles," *Robotics and Automation, IEEE Transactions on*, vol. 17, pp. 498-501, 2001.
- [14] T. Kito, J. Ota, R. Katsuki, T. Mizuta, T. Arai, T. Ueyama, *et al.*, "Smooth path planning by using visibility graph-like method," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, 2003, pp. 3770-3775.
- [15] K. Yang and S. Sukkarieh, "An analytical continuous-curvature path-smoothing algorithm," *Robotics, IEEE Transactions on*, vol. 26, pp. 561-568, 2010.
- [16] J. Villagra, V. Milanés, J. Pérez, and J. Godoy, "Smooth path and speed planning for an automated public transport vehicle," *Robotics and Autonomous Systems*, vol. 60, pp. 252-265, 2012.
- [17] N. A. Melchior and R. Simmons, "Particle RRT for path planning with uncertainty," in *Robotics and Automation, 2007 IEEE International Conference on*, 2007, pp. 1617-1624.
- [18] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, *et al.*, *Principles of robot motion: theory, algorithms, and implementations*: MIT press, 2005.
- [19] J. Stoer, R. Bulirsch, R. Bartels, W. Gautschi, and C. Witzgall, *Introduction to numerical analysis* vol. 2: Springer New York, 1993.
- [20] A. Gilat and V. Subramaniam, *Numerical methods for engineers and scientists*: Wiley, 2007.
- [21] J. F. Epperson, "On the Runge example," *Amer. Math. Monthly*, vol. 94, pp. 329-341, 1987.
- [22] A. Piazza, C. Lo Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, "Quintic  $G^2$ -splines for the iterative steering of vision-based autonomous vehicles," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 3, pp. 27-36, 2002.

- [23] A. Piazzzi, C. Guarino Lo Bianco, and M. Romano, “ $\eta^3$  Splines for the Smooth Path Generation of Wheeled Mobile Robots,” *IEEE TRANSACTIONS ON ROBOTICS*, vol. 23, pp. 1089-1095, 2007.
- [24] A. Piazzzi, C. G. L. Bianco, and M. Romano, “Smooth path generation for wheeled mobile robots using  $\eta^3$ -splines,” *Motion Control*, pp. 75-96, 2010.
- [25] A. Piazzzi, M. Romano, and C. G. L. Bianco, “G3-splines for the path planning of wheeled mobile robots,” in *European Control Conference*, 2003.
- [26] K. Yang, D. Jung, and S. Sukkarieh, “Continuous curvature path-smoothing algorithm using cubic B zier spiral curves for non-holonomic robots,” *Advanced Robotics*, vol. 27, pp. 247-258, 2013.



**Seong-Ryong Chang** received his B.S and M.S degree in Electronic Engineering from Dankook University in 1999 and 2001, and a Ph.D. degree in Electrical Engineering from Inha University of Incheon, Korea in 2013. His research interests are path planning, robot motion, mobile robot and embedded system.



**Uk-Youl Huh** received his B.S, M.S and Ph.D degrees in Electrical Engineering from Seoul University in 1974, 1978 and 1982. He is currently a professor with the Department of Electrical Engineering, Inha University of Incheon, Korea. His research interests are in the areas of intelligent control, design of fuzzy controller design and mobile robots.