

# Smooth Path Generation Based on Bézier Curves for Autonomous Vehicles

Ji-wung Choi <sup>\*</sup>, Renwick E. Curry <sup>†</sup>, Gabriel Hugh Elkaim <sup>‡</sup>

**Abstract**—In this paper we present two path planning algorithms based on Bézier curves for autonomous vehicles with waypoints and corridor constraints. Bézier curves have useful properties for the path generation problem. This paper describes how the algorithms apply these properties to generate the reference trajectory for vehicles to satisfy the path constraints. Both algorithms join a set of low-degree Bézier curve segments smoothly to generate the path. Additionally, we discuss the constrained optimization problem that optimizes the resulting path for a user-defined cost function. The simulation demonstrates the improvement of trajectory generation in terms of smoother steering control and smaller cross track error compared to previous work.

**Keywords:** *Bézier, Path Planning, Optimization, Autonomous Vehicle, Feedback Control.*

## 1. Introduction

Bézier Curves were invented in 1962 by the French engineer Pierre Bézier for designing automobile bodies. Today Bézier Curves are widely used in computer graphics and animation [4]. The Bézier curves have useful properties for the path generation problem.

Choi has presented two path planning algorithms based on Bézier curves for autonomous vehicles with waypoints and corridor constraints [2]. Both algorithms join cubic Bézier curve segments smoothly to generate the reference trajectory for vehicles to satisfy the path constraints. Also, both algorithms are constrained in that the path must cross over a bisector line of corner area such that the tangent at the crossing point is normal to the bisector. Additionally, that paper discuss the constrained optimization problem that optimizes the resulting path for user-defined cost function. Since the Bézier curve is uniquely defined by its control points, the optimization problem is parameterized by the location of control points. Even though the simulation provided in that paper has shown the generation of

smooth routes, discontinuities of the yaw angular rate have appeared at junction nodes between curve segments. This is because the curve segments are constrained to connect each other by only  $C^1$  continuity, so the curvature of the path is discontinuous at the nodes. (Section 2 describes this more detail.)

To resolve this problem, the current paper proposes new path planning algorithms. The algorithms impose constraints such that curve segments are  $C^2$  continuous in order to have curvature continuous for every point on the path. In addition, they give the reference path more freedom by getting rid of redundant constraints used in [2], such as the tangent being normal to the bisector, the initial/final heading, and symmetry of curve segments on corner area. The degree of each Bézier curve segments are determined by the minimum number of control points to satisfy imposed constraints while cubic Bézier curves are used for every segments in [2]. The optimized resulting path is obtained by computing the constrained optimization problem for the same cost function as the one in [2]. The numerical simulation results provided in this paper demonstrate the improvement of trajectory generation in terms of smoother steering control and smaller cross track error.

The paper is organized as follows: Section 2 begins by describing the definition of the Bézier curve and its useful properties for path planning. Section 3 discusses the control problem for autonomous vehicles, the vehicle dynamics, and vehicle control algorithms. Section 4 proposes two path planning methods based on Bézier curves, and discusses the constrained optimization problem of these methods. In Section 5, simulation results of control problem for autonomous vehicles are given. Finally, Section 6 provides conclusions.

## 2. Bézier Curve

A Bézier Curve of degree  $n$  can be represented as

$$P(t) = \sum_{i=0}^n B_i^n(t) P_i, \quad t \in [0, 1]$$

<sup>\*</sup>Ph.D. Candidate, Autonomous Systems Lab, Computer Engineering Department, University of California, Santa Cruz, 95064, Tel: 831-428-2146, Email: jwchoi@soe.ucsc.edu

<sup>†</sup>Adjunct Professor, Computer Engineering Department, University of California, Santa Cruz, 95064, Email: rcurry@ucsc.edu

<sup>‡</sup>Associate Professor, Autonomous Systems Lab, Computer Engineering Department, University of California, Santa Cruz, 95064, Tel: 831-459-3054, Email: elkaim@soe.ucsc.edu

Where  $P_i$  are control points such that  $P(0) = P_0$  and  $P(1) = P_n$ ,  $B_i^n(t)$  is a Bernstein polynomial given by

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i, \quad i \in \{0, 1, \dots, n\}$$

Bézier Curves have useful properties for path planning:

- They always passes through  $P_0$  and  $P_n$ .
- They are always tangent to the lines connecting  $P_0 \rightarrow P_1$  and  $P_n \rightarrow P_{n-1}$  at  $P_0$  and  $P_n$  respectively.
- They always lie within the convex hull consisting of their control points.

## 2.1. The de Casteljau Algorithm

The de Casteljau algorithm describes a recursive process to subdivide a Bézier curve  $P(t)$  into two segments. The subdivided segments are also Bézier curves. Let  $\{P_0^0, P_1^0, \dots, P_n^0\}$  denote the control points of  $P(t)$ . The control points of the segments can be computed by

$$P_i^j = (1-\tau)P_i^{j-1} + \tau P_{i+1}^{j-1}, \quad (1)$$

$$j \in \{1, \dots, n\}, \quad i \in \{0, \dots, n-j\}$$

where  $\tau \in (0, 1)$ . Then,  $\{P_0^0, P_1^0, \dots, P_n^0\}$  are the control points of one segment and  $\{P_0^n, P_1^n - 1, \dots, P_n^n\}$  are the another. This leads to the following property [2]:

**Remark 1.** A Bézier curve  $P(t)$  constructed by control points  $\{P_0^0, P_1^0, \dots, P_n^0\}$  always passes through the point  $P_0^n$  computed by applying the de Casteljau algorithm and using (1). Also, it is always tangent to  $\overline{P_0^{n-1}P_1^{n-1}}$  at  $P_0^n$ .

## 2.2. Derivatives, Continuity and Curvature

The derivatives of a Bézier curve can be determined by its control points [4]. For a Bézier curve  $P(t) = \sum_{i=0}^n B_i^n(t)P_i$ , the first derivative can be represented as:

$$\dot{P}(t) = \sum_{i=0}^{n-1} B_i^{n-1}(t)D_i \quad (2)$$

Where  $D_i$ , control points of  $\dot{P}(t)$  is

$$D_i = n(P_{i+1} - P_i)$$

The higher order derivative of a Bézier curve can be obtained by using the relationship of (2), iteratively.

Two Bézier curves  $P(t)$  and  $Q(t)$  are said to be  $C^k$  at  $t_0$  continuous [4] if

$$P(t_0) = Q(t_0), \dot{P}(t_0) = \dot{Q}(t_0), \dots, P^{(k)}(t_0) = Q^{(k)}(t_0) \quad (3)$$

The curvature of a Bézier curve  $P(t) = (x(t), y(t))$  at  $t$  is given by [4]

$$\kappa(t) = \frac{|\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)|}{(\dot{x}^2(t) + \dot{y}^2(t))^{\frac{3}{2}}}. \quad (4)$$

We can come up with the following property:

**Lemma 1.** For the path constructed by two Bézier curve segments  $P(t)|_{t \in [t_0, t_1]}$  and  $Q(t)|_{t \in [t_1, t_2]}$ , if  $P(t)$  and  $Q(t)$  are at least  $C^2$  continuous at  $t_1$  then the path has continuous curvature for every point on it.

*Proof.* The curvature is expressed in terms of the first and the second derivative of a curve in (4). Since the Bézier curves are defined as polynomial functions of  $t$ , their  $k$ -th derivative for all  $k = 1, 2, \dots$  are continuous. Hence, they always have continuous curvature for all  $t$ . For two different Bézier curves  $P(t)$  and  $Q(t)$ , it is sufficient that  $\kappa(t_1)$ , the curvature at the junction node is continuous if  $\dot{P}(t) = \dot{Q}(t)$  and  $\ddot{P}(t) = \ddot{Q}(t)$  are continuous at  $t_1$ . Therefore, if  $P(t)$  and  $Q(t)$  are at least  $C^2$  continuous at  $t_1$  then the path have the curvature continuous for every point on it.  $\square$

## 3. Problem Statement

Consider the control problem of a ground vehicle with a mission defined by waypoints and corridor constraints in a two-dimensional free-space. Our goal is to develop and implement an algorithm for navigation that satisfies these constraints. Let us denote each waypoint  $W_i \in \mathbb{R}^2$  for  $i \in \{1, 2, \dots, N\}$ , where  $N \in \mathbb{R}$  is the total number of waypoints. Corridor width is denoted as  $w_j$ ,  $j$ -th widths of each segment between two waypoints,  $j \in \{1, \dots, N-1\}$ .

### 3.1. Dynamic Model of Vehicle Motion

This section describes a dynamic model for motion of a vehicle that is used in the simulation in Section 5. For the dynamics of the vehicle, the state and the control vector are denoted  $\mathbf{q}(t) = (x_c(t), y_c(t), \psi(t))^T$  and  $\mathbf{u}(t) = (v(t), \omega(t))^T$  respectively. Where  $(x_c, y_c)$  represents the position of the center of gravity of the vehicle. The vehicle yaw angle  $\psi$  is defined to the angle from the  $X$  axis.  $v$  is the longitudinal velocity of the vehicle at the center of gravity.  $\omega = \dot{\psi}$  is the yaw angular rate. It follows that

$$\dot{\mathbf{q}}(t) = \begin{pmatrix} \cos \psi(t) & 0 \\ \sin \psi(t) & 0 \\ 0 & 1 \end{pmatrix} \mathbf{u}(t)$$

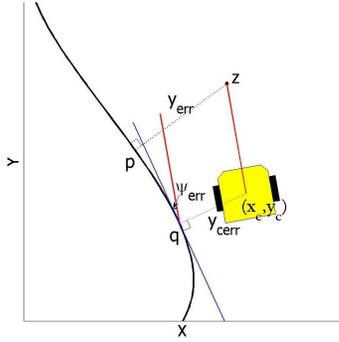


Figure 1: The position error is measured from a point  $z$ , projected in front of the vehicle, and unto the desired curve to point  $p$ .

### 3.2. Controls

The vehicle uses feed forward path planning with feedback corrections as illustrated in Figure 1 [3]. A position and orientation error is computed every 50 *ms*. The cross track error  $y_{cerr}$  is defined by the shortest distance between the reference trajectory and the position of the center of gravity of the vehicle  $(x_c, y_c)$ . A point  $z$  is computed with the current longitudinal velocity and heading of the vehicle from the current position.  $z$  is projected onto the reference trajectory at point  $p$  such that  $\overline{zp}$  is normal to the tangent at  $p$ . The cross track error  $y_{err}$  is defined by the distance between  $z$  and  $p$ . The steering control  $\omega$  uses PID controller with respect to cross track error  $y_{err}$ .

$$\delta\omega = k_p y_{err} + k_d \frac{dy_{err}}{dt} + k_i \int y_{err} dt$$

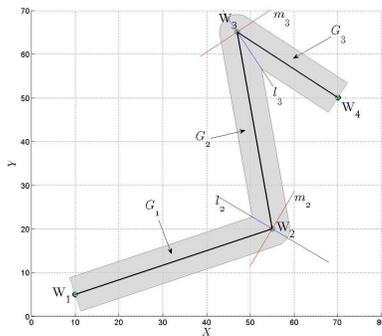


Figure 2: The course with four waypoints. Gray area is the permitted area for vehicles under a corridor constraint.

## 4. Path Planning Algorithm

In this section, two path planning methods based on Bézier curves are proposed. To describe the algorithms, let us

denote  $l_j$  as the bisector vector of  $\angle W_{j-1}W_jW_{j+1}$  for  $j \in \{2, \dots, N-1\}$  and  $m_j$  as the normal line to  $l_j$  at the intersection of the curves and  $l_j$  as illustrated in Figure 2. The planned path must cross over the bisectors under the waypoints and the corridor constraints. The location of the crossing point is represented as  $d_j \cdot l_j$ , where  $d_j \in \mathbb{R}$  is a scalar value. The course is divided into segments  $G_i$  by  $l_j$ .  $G_i$  indicates the permitted area for vehicles under corridor constraint  $w_i$ , from  $W_i$  to  $W_{i+1}$ .

Bézier curves constructed by large numbers of control points are numerically unstable. For this reason, it is desirable to join low-degree Bézier curves together in a smooth way for path planning [5]. Thus both methods use a set of low-degree Bézier curves such that the neighboring curves are  $C^2$  continuous at their end nodes. This will lead to continuous curvature on the resulting path by Lemma 1.

The Bézier curves used for the path planning are denoted as  ${}^iP(t) = \sum_{k=0}^{n_i} B_k^{n_i}(t) \cdot {}^iP_k$  for  $i \in \{1, \dots, M\}$ ,  $t \in [0, 1]$  where  $M$  is the total number of the Bézier curves and  $n_i$  is the degree of  ${}^iP$ . The planned path denoted as  $P$  is a concatenation of all  ${}^iP$ .

### 4.1 Path Planning Placing Bézier Curves within Segments

In this path planning method, the Bézier curve  ${}^iP$  for  $i \in \{1, \dots, N-1\}$  are used within each segment  $G_i$ . The planned path  $P$  are designed such that it begins from  $W_1$  and ends to  $W_N$ . Furthermore, the corridor and the  $C^2$  continuity constraints are satisfied.

The control points of  ${}^iP$ ,  ${}^iP_k$  for  $k = \{0, \dots, n_i\}$  are determined to maintain these conditions.

- The beginning and the end point are  $W_1$  and  $W_N$ .

$${}^1P_0 = W_1, \quad {}^{N-1}P_{n_{N-1}} = W_N \quad (5)$$

- The adjacent curves,  ${}^{j-1}P$  and  ${}^jP$  are  $C^2$  continuous at the crossing point,  $d_j \cdot l_j$  for  $j \in \{2, \dots, N-1\}$ .

$$\begin{aligned} {}^{j-1}P_{n_{j-1}} &= {}^jP_0 = d_j \cdot l_j \\ n_{j-1}({}^{j-1}P_{n_{j-1}} - {}^{j-1}P_{n_{j-1}-1}) &= n_j({}^jP_1 - {}^jP_0) \\ n_{j-1}(n_{j-1}-1)({}^{j-1}P_{n_{j-1}} - 2 \cdot {}^{j-1}P_{n_{j-1}-1} + {}^{j-1}P_{n_{j-1}-2}) &= n_j(n_j-1)({}^jP_2 - 2 \cdot {}^jP_1 + {}^jP_0) \end{aligned} \quad (6)$$

- The crossing points are bounded within the corridor.

$$d < \frac{1}{2} \min(w_{j-1}, w_j) \quad (7)$$

- ${}^iP_k$  always lie within the area of  $G_i$ .

$${}^iP_1 \in G_i, \dots, {}^iP_{n_i-1} \in G_i \quad (8)$$



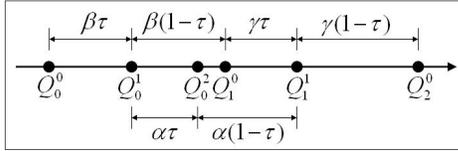


Figure 4: The geometry of  ${}^jQ(t)$  in  ${}^jT$  when  $y_0 = y_2 = 0$ .

control points as shown in Figure 4, we obtain

$$\begin{aligned} x_0 &= -(\alpha + \beta)\tau \\ x_2 &= (\alpha + \gamma)(1 - \tau) \\ \alpha &= \beta(1 - \tau) + \gamma\tau \end{aligned} \quad (15)$$

where  $\alpha > 0$ ,  $\beta > 0$ ,  $\gamma > 0$  are constants. Using (15),  $Q_1^0 = (x_1, 0)$  is represented in terms of arbitrary  $\tau \in (0, 1)$ :

$$x_1 = -\frac{1}{2} \left( \frac{1 - \tau}{\tau} x_0 + \frac{\tau}{1 - \tau} x_2 \right) \quad (16)$$

Then Bézier curves  ${}^iP(t)$  for  $i \in \{1, 2, \dots, N - 1\}$  are used within each segment  $G_i$  so that  ${}^{j-1}P$  and  ${}^jQ$  are  $C^2$  continuous at  ${}^jQ_0$ ,  ${}^jP$  and  ${}^jQ$  are  $C^2$  continuous at  ${}^jQ_2$ . The degree of  ${}^iP(t)$ ,  $n_i$  is determined by the minimum number of control points to satisfy the constraint:

$$\begin{cases} n_i = 3, & i \in \{1, N - 1\} \\ n_i = 5, & i \in \{2, \dots, N - 2\} \end{cases} \quad (17)$$

The constraints imposed on the planned path are as follows:

- The beginning and end point of  $P$  is  $W_1$  and  $W_N$ .

$${}^1P_0 = W_1, \quad {}^{N-1}P_{n_{N-1}} = W_N \quad (18)$$

- ${}^{j-1}P(t)$  and  ${}^jQ(t)$  are  $C^2$  continuous at  ${}^jQ_0$ .

$$\begin{aligned} {}^{j-1}P_{n_{j-1}}^0 &= {}^jQ_0^0 \\ n_{j-1}({}^{j-1}P_{n_{j-1}}^0 - {}^{j-1}P_{n_{j-1}-1}^0) &= 2({}^jQ_1^0 - {}^jQ_0^0) \\ n_{j-1}(n_{j-1} - 1)({}^{j-1}P_{n_{j-1}}^0 - 2 \cdot {}^{j-1}P_{n_{j-1}-1}^0 + {}^{j-1}P_{n_{j-1}-2}^0) &= 2 \cdot 1 \cdot ({}^jQ_2^0 - 2{}^jQ_1^0 + {}^jQ_0^0) \end{aligned} \quad (19)$$

- ${}^jP(t)$  and  ${}^jQ(t)$  are  $C^2$  continuous at  ${}^jQ_2$ .

$$\begin{aligned} {}^jP_0^0 &= {}^jQ_2^0 \\ n_j({}^jP_1^0 - {}^jP_0^0) &= 2({}^jQ_2^0 - {}^jQ_1^0) \\ n_j(n_j - 1)({}^jP_2^0 - 2 \cdot {}^jP_1^0 + {}^jP_0^0) &= 2 \cdot 1 \cdot ({}^jQ_2^0 - 2{}^jQ_1^0 + {}^jQ_0^0) \end{aligned} \quad (20)$$

- The crossing points are bounded within the corridor.

$$|d_j| < \frac{1}{2} \min(w_{j-1}, w_j) \quad (21)$$

- The slope of the tangent does not exceed  $l_j$ .

$$|\theta_j| < \pi/2 \quad (22)$$

- ${}^jQ_0^0$  and  ${}^jQ_2^0$  with respect to  ${}^jT$  satisfies Lemma 2.

$$\{^j\}y_0 \{^j\}y_2 \geq 0 \quad (23)$$

Where  $\{^j\}y_k$  denotes the coordinate with respect to  ${}^jT$ .

- ${}^jQ_0^0$  and  ${}^jQ_1^0$  lie within  $G_{j-1}$ .  ${}^jQ_2^0$  and  ${}^jQ_1^1$  lie within  $G_j$ .

$${}^jQ_0^0 \in G_{j-1}, \quad {}^jQ_1^0 \in G_{j-1}, \quad {}^jQ_2^0 \in G_j, \quad {}^jQ_1^1 \in G_j \quad (24)$$

- $\{^iP_1, \dots, {}^iP_{n_i-1}\}$  always lie within the area of  $G_i$ .

$${}^iP_1 \in G_i, \quad \dots, \quad {}^iP_{n_i-1} \in G_i \quad (25)$$

Then  $6(N - 2)$  free variables  $\mathbf{Q} = \{^jQ_0\}$ ,  $\mathbf{d} = \{d_j\}$  and  $\theta = \{\theta_j\}$  for  $j \in \{2, \dots, N - 1\}$  are computed by minimizing the constrained optimization problem:

$$\min_{\mathbf{Q}, \mathbf{d}, \theta} J = \sum_{i=1}^{N-1} J_i \quad (26)$$

subject to (21), (22), (23), (24), and (25).

Notice that the convex hull property is tested for  ${}^jQ_0^1$  and  ${}^jQ_1^1$  of the divided curves instead of  ${}^jQ_1^0$ . As the result, it comes up with more tight condition for curves against the corridor constraint.

## 5. Simulation Results

Simulation performed in this paper uses the course with waypoints  $\mathbf{W} = \{W_k\}$ ,  $k \in \{1, \dots, N\}$  and corridor width  $w_i = 8$ ,  $i \in \{1, \dots, N - 1\}$  for  $N = 4$  as illustrated in Figure 2. The location of waypoints are given by two-dimensional world coordinates  $(X, Y)$  in meter scale:  $W_1 = (10, 5)$ ,  $W_2 = (55, 20)$ ,  $W_3 = (47, 65)$ ,  $W_4 = (70, 50)$ . Initial position is assumed to fit to the first waypoint of the reference path respectively. The constant longitudinal velocity  $v(t) = 10$  m/s is used. The magnitude of  $\omega$  is bounded within  $|\omega|_{max} = 25$  rpm. The PID gains are given by:  $k_p = 2$ ,  $k_d = 1$ , and  $k_i = 0.1$ .

Path planning methods based on Section 4.1, and 4.2 are denoted as *Bézier1* and *Bézier2* respectively. Figure 5(a) and 5(b) are the path planned by *Bézier1* of [2] and of this paper, respectively. Figure 6(a) and 6(b) are ones by *Bézier2* of two methods. Circles indicate the location of control points of Bézier curve segments,  ${}^iP$ . In Figure 6(a) and

6(b), control points of  $^jQ$  are marked by stars. All of them are obtained by solving Equation (10) or (26) with

$$J_i = \int_0^1 \left[ (a_i |\kappa(t)|^2 + b_i |\dot{\kappa}(t)|^2) dt \right]$$

Where  $a_i = b_i = 1$ . The cost function leads to resulting paths with larger radii of curvature for Bézier curves. Comparing to the paths generated by [2], the proposed algorithm generated smoother paths in turning area.

In Figure 7(a) and 8(a), we can see that path planning by the proposed algorithms has smoother steering compared to the ones obtained by [2]. The discontinuity of  $\omega$  by that method imposes large forces and large changes in forces on the vehicle in the lateral direction. Moreover, the proposed algorithms result in smaller cross track error in Figure 7(b) over the one by [2].

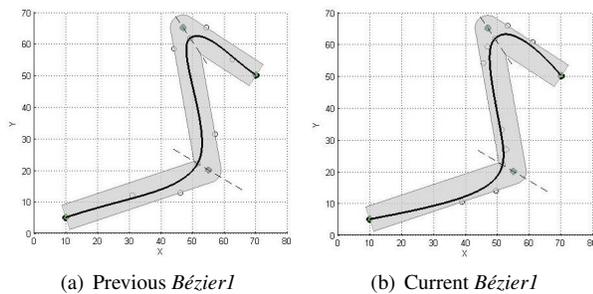


Figure 5: The planned path by previous *Bézier1* method of [2] (left) and by current method (right).

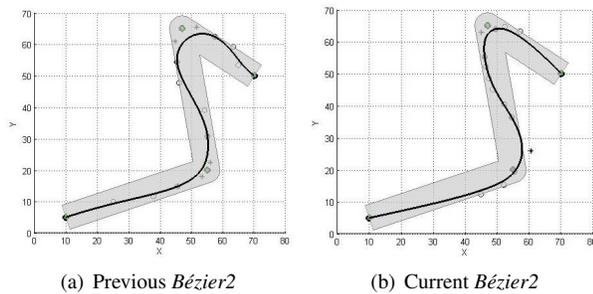


Figure 6: The planned path by previous *Bézier2* method of [2] (left) and by current method (right).

## 6. Conclusions

This paper presents two path planning algorithms based on Bézier curves for autonomous vehicles with waypoints and corridor constraints. Bézier curves provide an efficient way to generate the optimized path and satisfy the constraints at

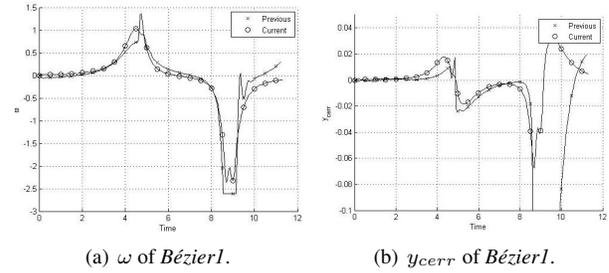


Figure 7: The steering control  $\omega$  and the cross track error  $y_{cerr}$  by previous *Bézier1* method of [2] (x) and by current method (o).

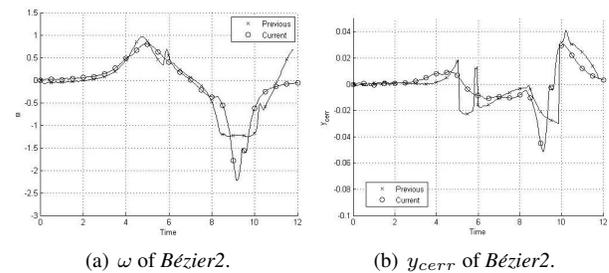


Figure 8: The steering control  $\omega$  and the cross track error  $y_{cerr}$  by previous *Bézier2* method of [2] (x) and by current method (o).

the same time. The simulation results also show that the trajectory of the vehicle follows the planned path within the constraints.

## References

- [1] J. Choi, G. Elkaim, "Bezier Curves for Trajectory Guidance," *World Congress on Engineering and Computer Science, WCECS 2008*, San Francisco, CA, Oct. 22-24, 2008
- [2] J. Choi, R. Curry, G. Elkaim, "Path Planning based on Bezier Curve for Autonomous Ground Vehicles," in *Proceedings of the Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008 (WCECS 2008)*
- [3] G. Elkaim, J. Connors, and J. Nagel, "The Overbot: An off-road autonomous ground vehicle testbed," *ION Global Navigation Satellite Systems Conference (ION-GNSS 2006)*, 1, Sept. p.22-24, 2006.
- [4] T. W. Sederber, "Computer aided geometric design," *CAGD Course Notes*, Brigham Young University, Provo, UT, 84602, April 2007.
- [5] I. Skrjanc, G. Klančar "Cooperative Collision Avoidance between Multiple Robots Based on Bzier Curves," *Information Technology Interfaces, 2007 (ITI 2007)*, p.451-456, June 25-28, 2007.