Path Planning Based on Bézier Curve for Autonomous Ground Vehicles

Ji-wung Choi
Computer Engineering Department
University of California Santa Cruz
Santa Cruz, CA95064, US
jwchoi@soe.ucsc.edu

Renwick Curry
Computer Engineering Department
University of California Santa Cruz
Santa Cruz, CA95064, US
rcurry@ucsc.edu

Gabriel Elkaim
Computer Engineering Department
University of California Santa Cruz
Santa Cruz, CA95064, US
elkaim@soe.ucsc.edu

***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. The paper describes how the algorithms apply these properties to generate the reference trajectory for vehicles to satisfy the path constraints. Both algorithms join cubic 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 shows the generation of successful routes for autonomous vehicles using these algorithms as well as control results for a simple kinematic vehicle. Extensions of these algorithms towards navigating through an unstructured environment with limited sensor range are discussed.**

Keywords–Bézier curve; path planning; optimization; autonomous vehicle; feedback control

I. Introduction

Exploiting the versatility of autonomous vehicles for academic, industrial, and military applications will have a profound effect on future applications. Current research on control systems for autonomous vehicles demonstrates that trajectory generation is hardly a "solved" problem. For vehicle viability, it is imperative to be able to generate safe paths in real time.

Many path planning techniques for autonomous vehicles have been discussed in the literature. Cornell University Team for 2005 DARPA Grand Challenge [9] used a path planner based on Bézier curves of degree 3 in a sensing/action feedback loop to generate smooth paths that are consistent with vehicle dynamics. Skrjanc [8] proposed a new cooperative collision avoidance method for multiple robots with constraints and known start and goal velocities based on Bézier curves of degree 4. In this method, four control points out of five are placed such that desired positions and velocities of the start and the goal point are satisfied. The fifth point is obtained by minimizing penalty functions. Lizarraga [5] used Bézier curves for generating spatially deconflicted paths for multiple UAVs.

Connors and Elkaim [1] previously presented a method for developing feasible paths through complicated environments using a kernel function based on cubic splines. This method iteratively refines the path to compute a feasible path and thus find a collision free path in real time through an unstructured environment. This method, when implemented in a receding horizon fashion, becomes the basis for high level control. This previous method, however, result in an incomplete path planning algorithm to satisfy the computational requirements in a complicated environment.

This paper presents a new approach based on Bézier curves as the seed function for the path planning algorithm as an alternative to cubic splines. The resulting path is manipulated by the control points of the bounding polygon. Though the optimization function for collision avoidance is non-linear, it can be solved quickly and efficiently. The results of the new algorithm demonstrate the generation of higher performance, more efficient, and successful routes for autonomous vehicles. Feedback control is used to track the planned path. The new algorithm is validated using simulations, and demonstrates suitability for successful tracking.

The paper is organized as follows: Section II begins by describing the definition of the Bézier curve and its useful properties for path planning. Section III discusses the control problem for autonomous vehicles, the vehicle dynamics, and vehicle control algorithms. Section IV proposes four path planning methods of which two are based on Bézier curves, and discusses the constrained optimization problem of these methods. In Section V, simulation results of control problem for autonomous vehicles are given. Finally, Section VI provides conclusions and future work.

II. Bézier Curve

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 [7]. A Bézier Curve of degree n can be represented as

$$P_{[t_0,t_1]}(t) = \sum_{i=0}^{n} B_i^n(t) P_i \tag{1}$$

where the P_i are control points such that $P(t_0) = P_0$ and $P(t_1) = P_n$, $B_i^n(t)$ is a Bernstein polynomial given by

$$B_i^n(t) = \binom{n}{i} (\frac{t_1 - t}{t_1 - t_0})^{n-i} (\frac{t - t_0}{t_1 - t_0})^i, \quad i \in \{0, 1, \ldots, n\} \tag{2}$$

Bézier Curves have useful properties for path planning:

978-0-7695-3555-5/08 $25.00 © 2008 IEEE
DOI 10.1109/WCECS.2008.27

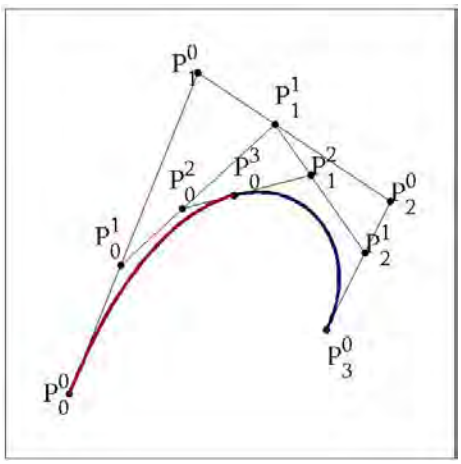


Figure 1. Subdividing a cubic Bézier curve with $\tau = .4$ by the de Casteljau Algorithm.

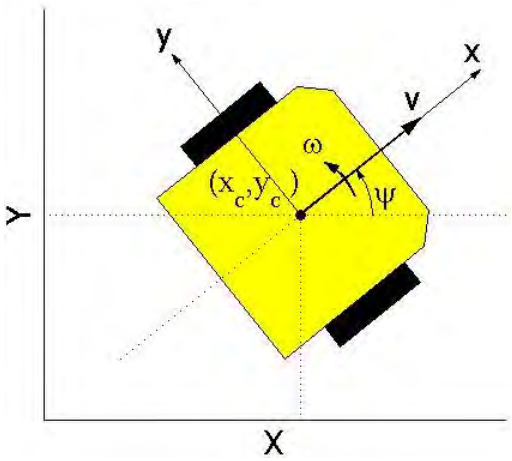


Figure 2. Schematic drawing of dynamic model of vehicle motion.

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

A. The de Casteljau Algorithm

The de Casteljau Algorithm is named after the French mathematician Paul de Casteljau, who developed the algorithm in 1959. The de Casteljau algorithm describes a recursive process to subdivide a Bézier curve $P_{[t_0,t_2]}(t)$ into two segments $P_{[t_0,t_1]}(t)$ and $P_{[t_1,t_2]}(t)$ [7]. Let $\{P_0^0, P_1^0, \ldots, P_n^0\}$ denote the control points of $P_{[t_0,t_2]}(t)$. The control points of $P_{[t_0,t_1]}(t)$ and $P_{[t_1,t_2]}(t)$ can be computed by

$$\begin{aligned} P_i^j =& (1-\tau)P_i^{j-1} + \tau P_{i+1}^{j-1}, \\ & j \in \{1,\ldots,n\}, \quad i \in \{0,\ldots,n-j\} \end{aligned} \tag{3}$$

where $\tau = \frac{t_1 - t_0}{t_2 - t_0}$. Then, $\{P_0^0, P_0^1, \ldots, P_0^n\}$ are the control points of $P_{[t_0,t_1]}$ and $\{P_0^n, P_1^{n-1}, \ldots, P_n^0\}$ are the control points of $P_{[t_1,t_2]}$. Figure 1 shows an example of the subdividing process for a cubic Bézier curve of which control points are $\{P_0^0, P_1^0, P_2^0, P_3^0\}$ by applying the de Casteljau algorithm with $\tau = .4$.

Remark 1. A Bézier curve $P_{[t_0,t_2]}$ always passes through the point $P(t_1) = P_0^n$ computed by applying the de Casteljau algorithm to subdivide itself into $P_{[t_0,t_1]}$ and $P_{[t_1,t_2]}$. Also, it is always tangent to $\overline{P_0^{n-1} P_1^{n-1}}$ at $P(t_1)$.

The path planning method introduced in the Section IV-C2 is motivated by this property.

B. Derivatives

The derivatives of a Bézier curve, referred to as the hodograph, can be determined by its control points [7]. For a Bézier curve $P_{[t_0,t_1]}(t) = \sum_{i=0}^{n} B_i^n(t) P_i$, the first derivative can be represented as:

$$\dot{P}_{[t_0,t_1]}(t) = \sum_{i=0}^{n-1} B_i^{n-1}(t) D_i \tag{4}$$

where D_i, the control points of $\dot{P}_{[t_0,t_1]}(t)$, are

$$D_i = \frac{n}{t_1 - t_0}(P_{i+1} - Pi) \tag{5}$$

Thus the first derivative of the Bézier curve $P_{[t_0,t_1]}(t)$ at t_0 and t_1 are

$$\dot{P}_{[t_0,t_1]}(t_0) = D_0 = \frac{n}{t_1 - t_0}(P_1 - P_0) \tag{6}$$

$$\dot{P}_{[t_0,t_1]}(t_1) = D_{n-1} = \frac{n}{t_1 - t_0}(P_n - P_{n-1}) \tag{7}$$

The higher order derivative of a Bézier curve can be obtained by using the relationship of (4).

C. Curvature

The curvature of a Bézier curve $P_{[t_0,t_1]}(t) = \big(x(t), y(t)\big)$ at any point with respect to t is given by

$$\kappa(t) = \frac{1}{R(t)} = \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{(\dot{x}^2(t) + \dot{y}^2(t))^{3/2}} \tag{8}$$

where $R(t)$ is the radius of curvature at the point.

III. 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 generating a reference path that satisfies these constraints. Let us denote each waypoint $W_i \in \mathbb{R}^2$ for $i \in \{1, 2, \ldots, N\}$, where $N \in \mathbb{N}$ is the total number of waypoints. Corridor width is denoted as $w_j \in \mathbb{R}$, j-th widths of each segment between two waypoints, $j \in \{1, \ldots, N-1\}$.

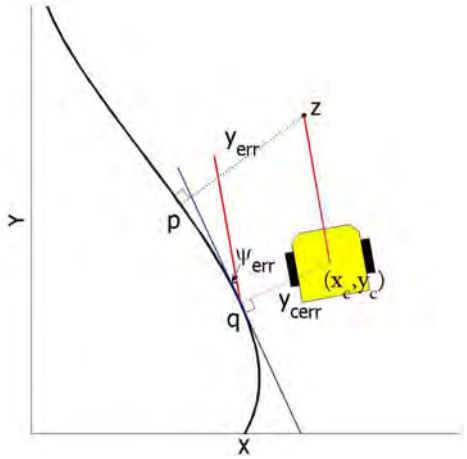


Figure 3. The position error of the vehicle is measured from a point z, projected in front of the vehicle, and unto the ideal curve to point p.

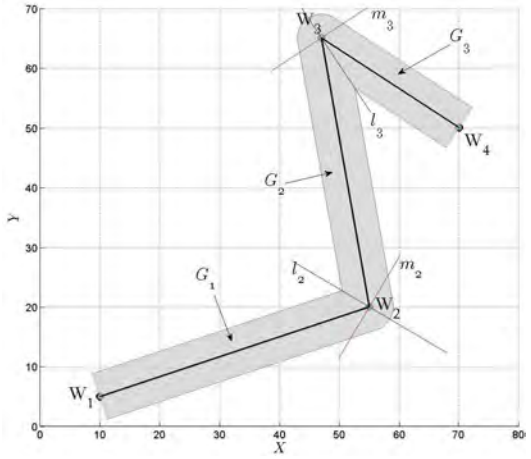


Figure 4. The course with four waypoints. Gray area is the permitted area for vehicles under a corridor constraint.

A. Kinematic Model of Vehicle Motion

This section describes a dynamic model for motion of a vehicle that is used in the simulation in Section V. Figure 2 shows the schematic drawing of dynamic model of the vehicle that is used in the simulation.

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 ψ is defined to the angle from the X axis. v is the constant longitudinal velocity of the vehicle at the center of gravity. $\omega = \dot{\psi}$ is the yaw angular velocity. It follows that

$$\dot{\mathbf{q}}(t) = \begin{pmatrix} \cos\psi(t) & 0 \\ \sin\psi(t) & 0 \\ 0 & 1 \end{pmatrix} \mathbf{u}(t) \tag{9}$$

The acceleration of the vehicle consists of two components: tangential acceleration and radial acceleration. The tangential acceleration is computed as the derivative of longitudinal velocity with respect to time. Since this paper considers constant longitudinal velocity, the tangential acceleration is zero.

$$a_t = \frac{dv}{dt} = 0 \tag{10}$$

The radial acceleration arises from the change in direction of the velocity vector and is given by

$$a_r = \frac{v^2}{R} = \kappa v^2 \tag{11}$$

We can see that the acceleration of the vehicle moving along the reference path is proportional to the curvature of the path at the point where the vehicle resides.

B. Control

The vehicle uses feed forward path planning with feedback corrections as illustrated in Figure 3 [4]. 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 one sample ahead 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 ω uses PID controller with respect to cross track error y_{err}.

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

IV. Path Planning Algorithms

In this section, four path planning methods are proposed. To describe the methods, we consider the course with waypoints $\mathbf{W} = \{W_i\}$, $i \in \{1, \ldots, N\}$ and corridor width w_k, $k \in \{1, \ldots, N-1\}$ for $N = 4$ as illustrated in Figure 4. The location of waypoints are given by two-dimensional world coordinates (X, Y) in $meter$ scale.

To describe the algorithms, let us denote l_j as the bisector of $\angle W_{j-1} W_j W_{j+1}$ for $j \in \{2, \ldots, N-1\}$ and m_j as the normal line to l_j at the intersect of the planned path and l_j. The course is divided into segments G_k by l_j. G_k indicates the permitted area for vehicles under corridor constraint w_k, from W_k to W_{k+1}.

A. Path Planning Based on Center Lines

The simplest path planning method is to follow a series of center lines that connect neighboring waypoints, plotted as a black line in Figure 4.

Although this method has advantages of simplicity and optimality in terms of the shortest path, the sharp turns at waypoints are not feasible for vehicles in practice. It will result in a position error varying as a function of the longitudinal velocity and the maximum steering angle rate.

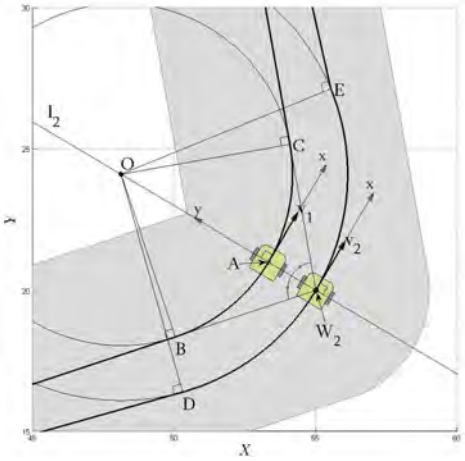


Figure 5. An enlarged path based on circular arc for a turn at W_2.

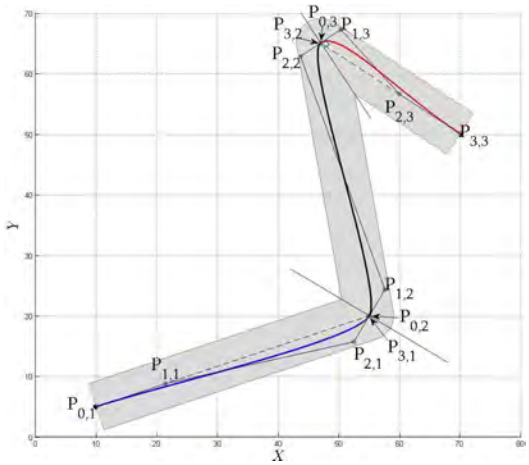


Figure 6. An example of the planned path based on cubic Bézier curves used within segments when $T_i = W_i$. The Bézier curves are plotted with different colors: blue, black, and red.

B. Path Planning Using Circular Arcs for Turns

To reduce position error at segment transitions with sharp turns, another path planning method is proposed with circular arcs. Under the assumption that the slip angle of the vehicle is zero in lateral motion, the ideal shape of a path with a turn is a circular arc as it has constant radius of curvature.

On a circular path of radius R, the yaw angular velocity ω that the vehicle with the longitudinal velocity v experiences is

$$\omega = \frac{v}{R} \tag{12}$$

Since R is a constant, if v is constant then ω is also constant.

Figure 5 shows the potential trajectories consisting of a circular arc and its tangents at W_2. The circular arc $\overset{\frown}{BAC}$ having its center at O is tangent to the center lines $\overline{W_1W_2}$ and $\overline{W_2W_3}$ at B and C respectively. For this case, the planned path is a concatenation of $\overline{W_1B}$, $\overset{\frown}{BAC}$, and $\overline{CW_2}$.

It is interesting to note two properties from the geometry. The one is that O is on l_2 because $\triangle OW_2B$ and $\triangle OW_2C$ are congruent. The other is that the tangent of the circular arc at the bisector l_2 is normal to l_2. It can be generalized as follows:

Remark 2. The tangent of the path at the bisector l_j is normal to l_j for $j \in \{2, \ldots, N-1\}$.

This property will be used as a constraint of the path planning methods introduced in the Section IV-C1 and IV-C2.

Because of the constraints, O moves along the bisector as the radius of the arc is extended or shrunk. For example, to satisfy the constraint of passing through waypoints, the radius should be extended from $\overline{OA}$ to $\overline{OW_2}$. For this case, the tangents are no longer center lines. However, the two geometrical properties are still preserved. In other words, the longitudinal velocity of the vehicle v on the bisector l_j for $j \in \{2, \ldots, N-1\}$ is normal to the bisector given that the vehicle follows the planned path perfectly.

C. Path Planning Based on Bézier Curves

In this section, two methods of path planning based on Bézier curves are proposed. Both methods are motivated by Remark 2 and are developed to preserve it.

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 [8]. The basic requirement of path planning is to pass through the beginning point and the end point with specified directions. The least degree of the Bézier curve that can satisfy this requirement is three (See properties of Bézier curves in Section II). Both methods use a set of cubic Bézier curves.

The cubic Bézier curves used for the path planning are denoted as $P_{[t_{i-1},t_i]}(t) = \sum_{k=0}^{3} B_k^3(t) P_{k,i}$ for $i \in \{1, \ldots, M\}$ where M is the total number of the Bézier curves. The planned path $P(t)$ for $t \in [t_0, t_M]$ is represented as

$$P(t) = \{P_{[t_{i-1},t_i]}(t)\}, \quad i \in \{1, \ldots, M\} \tag{13}$$

1) Path Planning Placing Bézier Curves within Segments: In this path planning method, one Bézier curve is used within each segment G_i as shown in Figure 6. The $N-1$ cubic Bézier curves are denoted $P_{[t_{i-1},t_i]}(t) = \sum_{k=0}^{3} B_k^3(t) P_{k,i}$ for $i \in \{1, \ldots, N-1\}$. The planned path $P(t)$ is designed such that it begins from W_1 with the heading of $\overrightarrow{W_1W_2}$ and ends to W_N with the heading of $\overrightarrow{W_{N-1}W_N}$. Furthermore, the corridor constraint and Remark 2 are satisfied.

The control points of $P_{[t_{i-1},t_i]}(t)$, $P_{k,i}$ are determined to maintain these conditions. The beginning point of $P_{[t_0,t_1]}(t)$, $P_{0,1}$ is W_1. The end point of $P_{[t_{N-2},t_{N-1}]}(t)$, $P_{3,N-1}$ is W_N. Other beginning/end points denoted as T_j are chosen on the bisectors l_j for $j \in \{2, \ldots, N-1\}$ and bounded within the corridor. These are $P_{3,j-1}$ and $P_{0,j}$. Also, $\{P_{1,i}, P_{2,i}\}$ are chosen such that the derivatives of $P_{[t_{j-2},t_{j-1}]}(t)$ and $P_{[t_{j-1},t_j]}(t)$ at T_j are continuous and are normal to l_j. Since $\{P_{0,i}, P_{1,i}, P_{2,i}, P_{3,i}\}$ all lie within the area of G_i, the result-

ing Bézier curve satisfies the corridor constraint by the convex hull property.

These can be formulated as the following constraints $\forall i, j$:

- The initial and the final constraints for the positions and the headings.

$$P_{[t_0,t_1]}(t_0) = P_{0,1} = W_1 \tag{14}$$
$$P_{[t_{N-2},t_{N-1}]}(t_{N-1}) = P_{3,N-1} = W_N \tag{15}$$
$$\dot{P}_{[t_0,t_1]}(t_0) = \nu_0 \overrightarrow{W_1 W_2} \tag{16}$$
$$\dot{P}_{[t_{N-2},t_{N-1}]}(t_{N-1}) = \nu_f \overrightarrow{W_{N-1} W_N} \tag{17}$$

  where $\nu_0, \nu_f \in \mathbb{R}^+$. Equation (16) and (17) can be rewritten by applying (4) and using (14) and (15):

$$\frac{3}{t_1 - t_0}(P_{1,1} - W_1) = \nu_0(W_2 - W_1) \tag{18}$$
$$\frac{3}{t_{N-1} - t_{N-2}}(W_N - P_{2,N-1}) = \nu_f(W_N - W_{N-1}) \tag{19}$$

- Continuity constraints of the positions and the velocities at T_j.

$$P_{[t_{j-2},t_{j-1}]}(t_{j-1}) = P_{[t_{j-1},t_j]}(t_{j-1}) \tag{20}$$
$$\dot{P}_{[t_{j-2},t_{j-1}]}(t_{j-1}) = \dot{P}_{[t_{j-1},t_j]}(t_{j-1}) \tag{21}$$

  Equation (20) and (21) can be rewritten by applying (1) and (4), respectively:

$$T_j = P_{3,j-1} = P_{0,j} \tag{22}$$
$$\frac{3}{t_{j-1} - t_{j-2}}(P_{3,j-1} - P_{2,j-1}) = \frac{3}{t_j - t_{j-1}}(P_{1,j} - P_{0,j}) \tag{23}$$

- Constraints of T_j on l_j and within G_j.

$$T_j \in l_j \tag{24}$$
$$|T_j - W_j| < \frac{1}{2}\min(w_{j-1}, w_j) \tag{25}$$

- Boundary constraints of $P_{1,i}$ and $P_{2,i}$ within G_i.

$$\left.\begin{array}{ll} P_{1,i} & \in G_i \\ P_{2,i} & \in G_i \end{array}\right\} \tag{26}$$

- Normality constraint of the velocities at T_j to l_j.

$$\dot{P}_{[t_{j-1},t_j]}(t_{j-1}) \cdot l_j = \frac{3}{t_j - t_{j-1}}(P_{1,j} - P_{0,j}) \cdot l_j = 0 \tag{27}$$

where $[t_{i-1}, t_i]$ are assumed to be given. We can see that (14)-(27) are parameterized by control points of Bézier curves, $\{P_{0,i}, P_{1,i}, P_{2,i}, P_{3,i}\}$. Since the number of all i is $N-1$ and each control point is in two dimensional space, the total number of parameters is $8(N-1)$. The degrees of freedom is computed as the number of parameters minus the number of equality constraint equations. Among the equality constraint equations, the total number of equations (18), (19), (24), and (27) is $2(N-1)$. Each of equations (14), (15), (22), and (23) is paired for x and y coordinates. The total number of the paired equations are $4(N-1)$. Therefore, the total number of equality constraint equations is $6(N-1)$ and the degrees of freedom is $2(N-1)$. The parameters are denoted $\mathbf{Z} = \{P_{0,i}, P_{1,i}, P_{2,i}, P_{3,i}\}, \forall i$ and computed by minimizing the constrained optimization problem:

$$\min_{\mathbf{Z}} J = \sum_{i=1}^{N-1} J_i \tag{28}$$

$$\begin{aligned} J_i = &\int_{t_{i-1}}^{t_i} \left[a_i \left|\kappa_i(t)\right|^2 + b_i \left|\dot{\kappa}_i(t)\right|^2\right] dt \\ &+ c_i \left|\kappa_i(t_i) - \kappa_{i+1}(t_i)\right|^2, \quad i \in \{1, \ldots, N-2\} \\ J_i = &\int_{t_{i-1}}^{t_i} \left[a_i \left|\kappa_i(t)\right|^2 + b_i \left|\dot{\kappa}_i(t)\right|^2\right] dt, \quad i = N-1 \end{aligned} \tag{29}$$

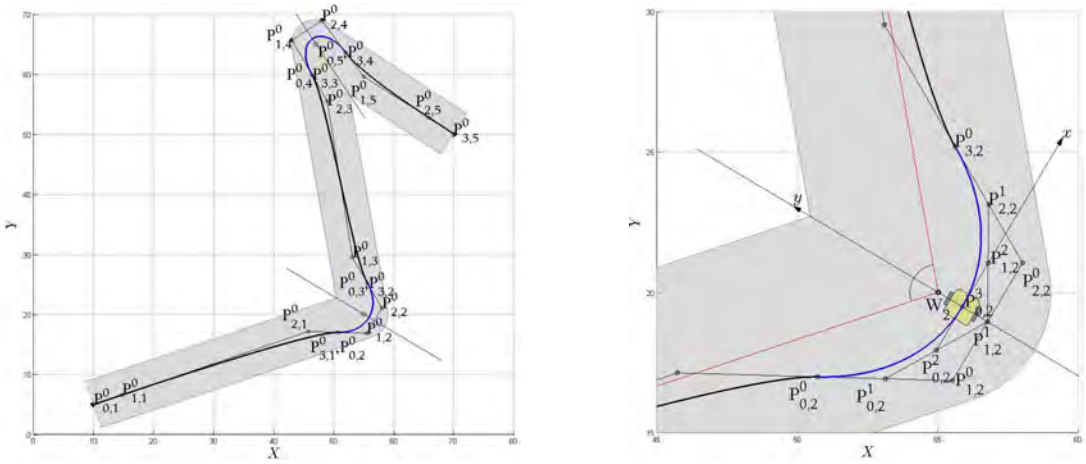
where $\kappa_i(t)$ and $\dot{\kappa}_i(t)$ are the curvature and the derivative of curvature at the point of $P_{[t_{i-1},t_i]}(t)$, respectively. The $a_i, b_i, c_i \in \mathbb{R}$ are weighting factors. J_i penalizes $|\kappa_i(t)|^2$ and $|\dot{\kappa}_i(t)|^2$ to give smoothness to each Bézier curve. It also penalizes the curvature difference at the point where neighboring Bézier curves meet for the vehicle moving along the resulting path to avoid a sudden change in radial acceleration. The optimization is subject to (14) (15), (18), (19), (22), (23), (24), (25), (26), and (27). As the result, the planned trajectory passes through $\{W_1, T_2, \ldots, T_{N-1}, W_N\}$ with Remark 2 preserved.

2) Path Planning Placing Mid-points of Bézier Curves on Bisectors of Turns: In the section IV-C1, a Bézier curve is used within each segment. Another path planning method places the mid-points of $N-2$ Bézier curves on the bisectors l_j, $j \in \{2, \ldots, N-1\}$ using the de Casteljau algorithm, with $N-1$ curves adjoining them, for a total of $2N-3$ cubic Bézier curves $P_{[t_{i-1},t_i]}(t) = \sum_{k=0}^{3} B_k^3(t) P_{k,i}^0$ for $i \in \{1, \ldots, 2N-3\}$.

The local area of the course around W_j can be seen as symmetric with respect to l_j. Thus the cubic Bézier curves $P_{[t_{i'-1},t_{i'}]}(t)$, $i' \in \{2, 4, \ldots, 2N-4\}$ used for this area will also be symmetric with respect to $l_{i'/2+1}$. In other words, $P_{0,i'}^0$ and $P_{3,i'}^0$ are symmetric with respect to $l_{i'/2+1}$. So are $P_{1,i'}^0$ and $P_{2,i'}^0$. After applying the de Casteljau algorithm with $\tau = 0.5$ to the curve, the mid-point $P_{0,i'}^3$ is on $l_{i'/2+1}$. Then Bézier curves $P_{[t_{i''-1},t_{i''}]}(t)$ for $i'' \in \{1, 3, \ldots, 2N-3\}$ are used for the rest part of the path. Figure 7a shows an example of the planned path by applying the method for $N = 4$.

The constraints imposed on the planned path are as follows $\forall i', i''$:

- The initial and the final constraints for the positions and



(a) Overall path (b) Geometry of turning area

Figure 7. The path based on cubic Bézier curves placed such that their mid-point lies on the bisectors. The blue curves with control points $\{P^0_{k,2}, P^0_{k,4}\}$ for $k \in \{0,1,2,3\}$ are used for turning area. The rest of the path plotted by black lines are also cubic Bézier curves constructed by $\{P^0_{k,1}, P^0_{k,3}, P^0_{k,5}\}$.

the headings.

$$P^0_{0,1} = W_1 \tag{30}$$

$$P^0_{3,2N-3} = W_N \tag{31}$$

$$\frac{3}{t_1 - t_0}(P^0_{1,1} - W_1) = \nu_0(W_2 - W_1) \tag{32}$$

$$\frac{3}{t_{2N-3} - t_{2N-4}}(W_N - P^0_{2,2N-3}) = \nu_f(W_N - W_{N-1}) \tag{33}$$

where $\nu_0, \nu_f \in \mathbb{R}^+$.

- Continuity constraints of the positions and the velocities at $P^0_{0,i'}$ and $P^0_{3,i'}$.

$$\left.\begin{array}{ll} P^0_{3,i'-1} & = P^0_{0,i'} \\ P^0_{3,i'} & = P^0_{0,i'+1} \end{array}\right\} \tag{34}$$

$$\left.\begin{array}{l} \frac{3}{t_{i'-1}-t_{i'-2}}(P^0_{3,i'-1} - P^0_{2,i'-1}) = \frac{3}{t_{i'}-t_{i'-1}}(P^0_{1,i'} - P^0_{0,i'}) \\ \frac{3}{t_{i'}-t_{i'-1}}(P^0_{3,i'} - P^0_{2,i'}) = \frac{3}{t_{i'+1}-t_{i'}}(P^0_{1,i'+1} - P^0_{0,i'+1}) \end{array}\right\} \tag{35}$$

- Symmetry constraint of $P_{[t_{i'-1},t_{i'}]}(t)$ with respect to $l_{i'/2+1}$.

$$\left.\begin{array}{ll} (P^0_{0,i'} + P^0_{3,i'})/2 & \in l_{i'/2+1} \\ (P^0_{1,i'} + P^0_{2,i'})/2 & \in l_{i'/2+1} \end{array}\right\} \tag{36}$$

$$\left.\begin{array}{ll} \overrightarrow{P^0_{3,i'}P^0_{0,i'}} & \perp l_{i'/2+1} \\ \overrightarrow{P^0_{2,i'}P^0_{1,i'}} & \perp l_{i'/2+1} \end{array}\right\} \tag{37}$$

- Boundary condition of the subdivided Bézier curves with control points $\{P^0_{0,i'}, P^1_{0,i'}, P^2_{0,i'}, P^3_{0,i'}\}$ lying inside of $G_{i'/2}$.

$$\left.\begin{array}{ll} P^0_{0,i'} & \in G_{i'/2} \\ P^1_{0,i'} & \in G_{i'/2} \\ P^2_{0,i'} & \in G_{i'/2} \end{array}\right\} \tag{38}$$

Equation (38) can be rewritten by applying (3):

$$\left.\begin{array}{ll} P^0_{0,i'} & \in G_{i'/2} \\ \frac{1}{2}P^0_{0,i'} + \frac{1}{2}P^0_{1,i'} & \in G_{i'/2} \\ \frac{1}{4}P^0_{0,i'} + \frac{1}{2}P^0_{1,i'} + \frac{1}{4}P^0_{2,i'} & \in G_{i'/2} \end{array}\right\} \tag{39}$$

- Corridor constraints for the positions of $P^3_{0,i'}$.

$$|P^3_{0,i'} - W_{i'/2+1}| < \frac{1}{2}\min(w_{i'/2}, w_{i'/2+1}) \tag{40}$$

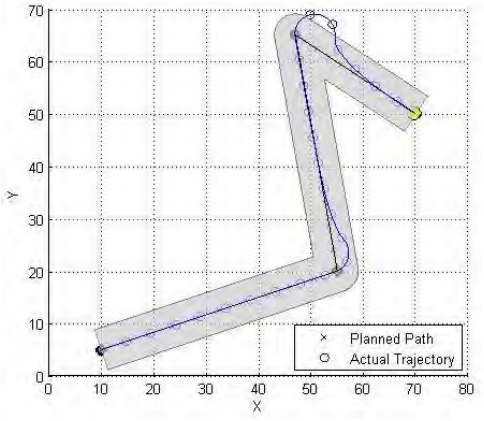
Equation (40) can be rewritten by applying (3):

$$\begin{array}{l} \left|\frac{1}{8}P^0_{0,i'} + \frac{3}{8}P^0_{1,i'} + \frac{3}{8}P^0_{2,i'} + \frac{1}{8}P^0_{3,i'} - W_{i'/2+1}\right| \\ \quad < \frac{1}{2}\min(w_{i'/2}, w_{i'/2+1}) \end{array} \tag{41}$$

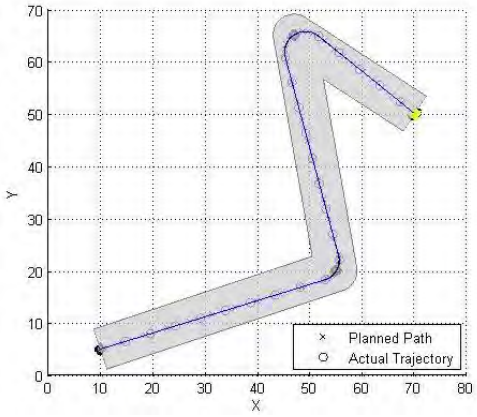
- Boundary constraints of $P^0_{1,i''}$ and $P^0_{2,i''}$ lying inside of $G_{(i''+1)/2}$.

$$\left.\begin{array}{ll} P^0_{1,i''} & \in G_{(i''+1)/2} \\ P^0_{2,i''} & \in G_{(i''+1)/2} \end{array}\right\} \tag{42}$$

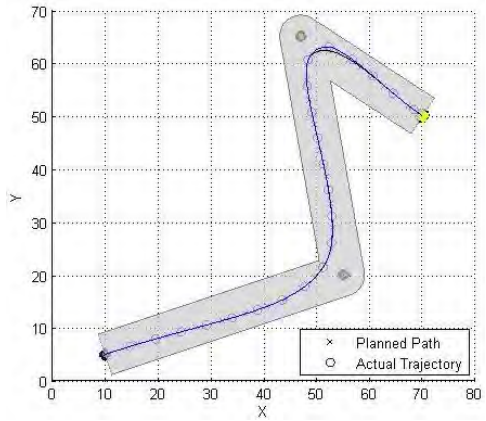
where $[t_{i-1}, t_i]$ are assumed to be given. We can see that (30)-(42) are parameterized by control points of Bézier curves. The control points are categorized into two types of sets: $\{P^0_{0,i'}, P^0_{1,i'}, P^0_{2,i'}, P^0_{3,i'}\}$ and $\{P^0_{0,i''}, P^0_{1,i''}, P^0_{2,i''}, P^0_{3,i''}\}$. Since the number of all i' is $N-2$ and the number of all i'' is $N-1$, the total number of parameters are $8(2N-3)$. Among equality constraint equations, the total number of equations (32), (33), (36), and (37) are $2(2N-3)$. Equations (30), (31), (34) and (35) are paired for x and y coordinates. The total number of the paired equations are $4(2N-3)$. So the total number of equality constraint equations are



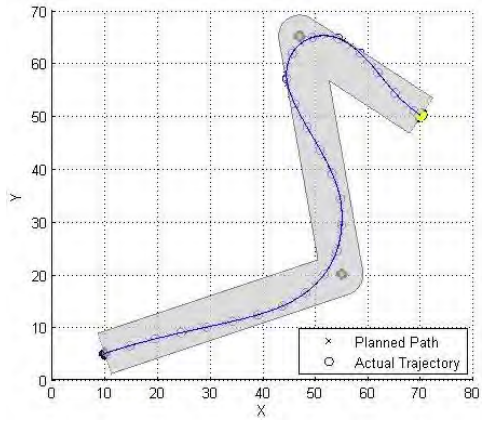
(a) Line



(b) Arc

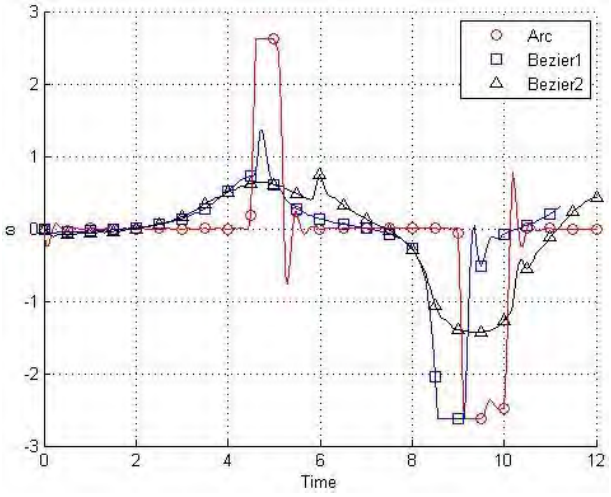


(c) Bézier1

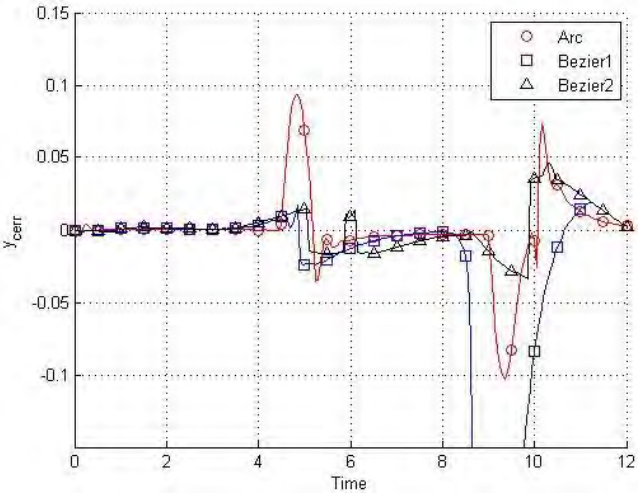


(d) Bézier2

Figure 8. Tracking simulation results (blue) over the planned paths (black).



(a) The steering control ω



(b) The cross track error

Figure 9. The steering control ω and the cross track error y_{cerr}.

$6(2N-3)$ and the degrees of freedom is $8(2N-3) - 6(2N-3) = 2(2N-3)$. The parameters are denoted $\mathbf{Z} = \{P^0_{0,i'}, P^0_{1,i'}, P^0_{2,i'}, P^0_{3,i'}, P^0_{0,i''}, P^0_{1,i''}, P^0_{2,i''}, P^0_{3,i''}\}, \forall i', i''$ and computed by minimizing the constrained optimization problem:

$$\min_{\mathbf{Z}} J = \sum_{i=1}^{2N-3} J_i \tag{43}$$

$$\begin{aligned} J_i &= \int_{t_{i-1}}^{t_i} \Big[a_i \big|\kappa_i(t)\big|^2 + b_i \big|\dot{\kappa}_i(t)\big|^2\Big] dt \\ &\quad + c_i \big|\kappa_i(t_i) - \kappa_{i+1}(t_i)\big|^2, \quad i \in \{1, 2, \ldots, 2N-4\} \\ J_i &= \int_{t_{i-1}}^{t_i} \Big[a_i \big|\kappa_i(t)\big|^2 + b_i \big|\dot{\kappa}_i(t)\big|^2\Big] dt, \quad i = 2N-3 \end{aligned} \tag{44}$$

where $a_i, b_i, c_i \in \mathbb{R}$. Note that the form of J_i is the same as that in (29). The cost function leads to resulting paths with larger radii of curvature for Bézier curves used in turns and paths close to a straight line for the other Bézier curves. The optimization is subject to the constraints equations of (30), (31), (32), (33), (34), (35), (36), (37), (39), (41), and (42).

A remarkable feature of this method is the fact that it imposes desirable position and direction constraints of the mid-point of a Bézier curve as well as those at the beginning and the end points. Furthermore, the convex hull property is tested for $\{P^0_{0,i'}, P^1_{0,i'}, P^2_{0,i'}, P^3_{0,i'}\}$ of the divided curves instead of $\{P^0_{0,i'}, P^0_{1,i'}, P^0_{2,i'}, P^0_{3,i'}\}$. Since the curve and the course are locally symmetrical, we do not need to check $\{P^3_{0,i'}, P^2_{1,i'}, P^1_{2,i'}, P^0_{3,i'}\}$. As the result, it comes up with tighter conditions for curves against the corridor constraint than checking $\{P^0_{0,i'}, P^0_{1,i'}, P^0_{2,i'}, P^0_{3,i'}\}$ by performing the same number of tests.

V. Simulation Results

Simulations performed in this paper use the course shown in Figure 4 with $N = 4$, $w_k = 8$ and the coordinates of waypoints: $W_1 = (10, 5)$, $W_2 = (55, 20)$, $W_3 = (47, 65)$ and $W_4 = (70, 50)$. Initial position and heading are assumed to fit to the first waypoint and the direction to the second waypoint respectively. A constant longitudinal velocity $v(t) = 10\, m/s$ is used. The magnitude of ω is bounded within $|\omega|_{max} = 2.618$ rad/s. The PID gains are given by: $k_p = 2$, $k_d = 1$, and $k_i = 0.1$.

Path planning methods based on Section IV-A, IV-B, IV-C1, and IV-C2 are denoted as *Line*, *Arc*, *Bézier1*, and *Bézier2* respectively. The simulated trajectory of a vehicle that tracks the planned path by *Line* is shown in Figure 8a. Although the vehicle tracks the straight parts of the planned path accurately, it overshoots in turns resulting in a large position error due to maximum steer angle limitation. Decreasing speed as the vehicle approaches to the turning area can reduce the position error. However, in this simulation, the longitudinal velocity v was kept constant which magnifies the tracking errors with this method of path planning.

Figure 8b shows the tracking result of *Arc* where a uniform radius $R = 5$ is used. It satisfies the condition of $R \geq \frac{v}{|\omega|_{max}} = 3.8197$ for vehicles to converge to the planned circular arc path. The actual trajectory of the vehicle fits to the planned path. In Figure 9a, ω is zero except for turning areas. At the areas, $|\omega| \approx 2$ agrees with Equation (12).

Figure 8c is the tracking result for the path planned by *Bézier1*. This path is obtained by solving the constrained optimization of Equation (28) with $t_i = i$, $a_i = 1$, $b_i = 1$ and $c_i = 1$.

Figure 8d is the tracking result for the path planned by *Bézier2*. This path is obtained by solving the constrained optimization of Equation (43) with $t_i = i$, $a_i = 1$, $b_i = 1$ and $c_i = 1$.

Figure 9 shows steering control signals and cross track errors of the vehicle for the different algorithms. In Fig. 9a, we can see that path planning based on Bézier curves has smoother steering compared to the circular arc method. The discontinuity of ω by that method imposes large forces and large changes in forces on the vehicle in the lateral direction. Examination of Fig. 9b shows that the *Bézier2* method has the smallest cross track error.

More tracking results over arbitrary and longer courses are shown in Fig. 10. The results are obtained by applying *Bézier2* method with the same parameters used above. These simulation results demonstrate generation of successful routes for vehicles using the algorithm as well as control results.

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

These path planning algorithms will be implemented on the Overbot [4], the autonomous ground vehicle at Autonomous Systems Lab at UCSC.

In this work, proposed algorithms only generate a nominal path. Enabling autonomous vehicles to detect unknown obstacles and safely avoid them is essential to future operations. Future work will employ receding horizon control methods to generate real-time Bézier-based optimal trajectories while avoiding obstacles.

References

[1] J. Connors and G. Elkaim, *Analysis of a Spline Based, Obstacle Avoiding Path Planning Algorithm*. IEEE Vehicle Technology Conference, IEEE VTC 2007, Dublin, Ireland, Apr. 22-25, 2007.

[2] J. Connors and G. Elkaim, *Experimental Results for Spline Based Obstacle Avoidance of an Off-Road Ground Vehicle*. ION Global Navigation Satellite Systems Conference, ION GNSS 2007, Fort Worth, TX, Sept. 25-28, 2007.

[3] J. Connors and G. Elkaim, *Manipulating B-Spline Based Paths for Obstacle Avoidance in Autonomous Ground Vehicles*. ION National Technical Meeting, ION NTM 2007, San Diego, CA, Jan. 22-24, 2007.

[4] 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.

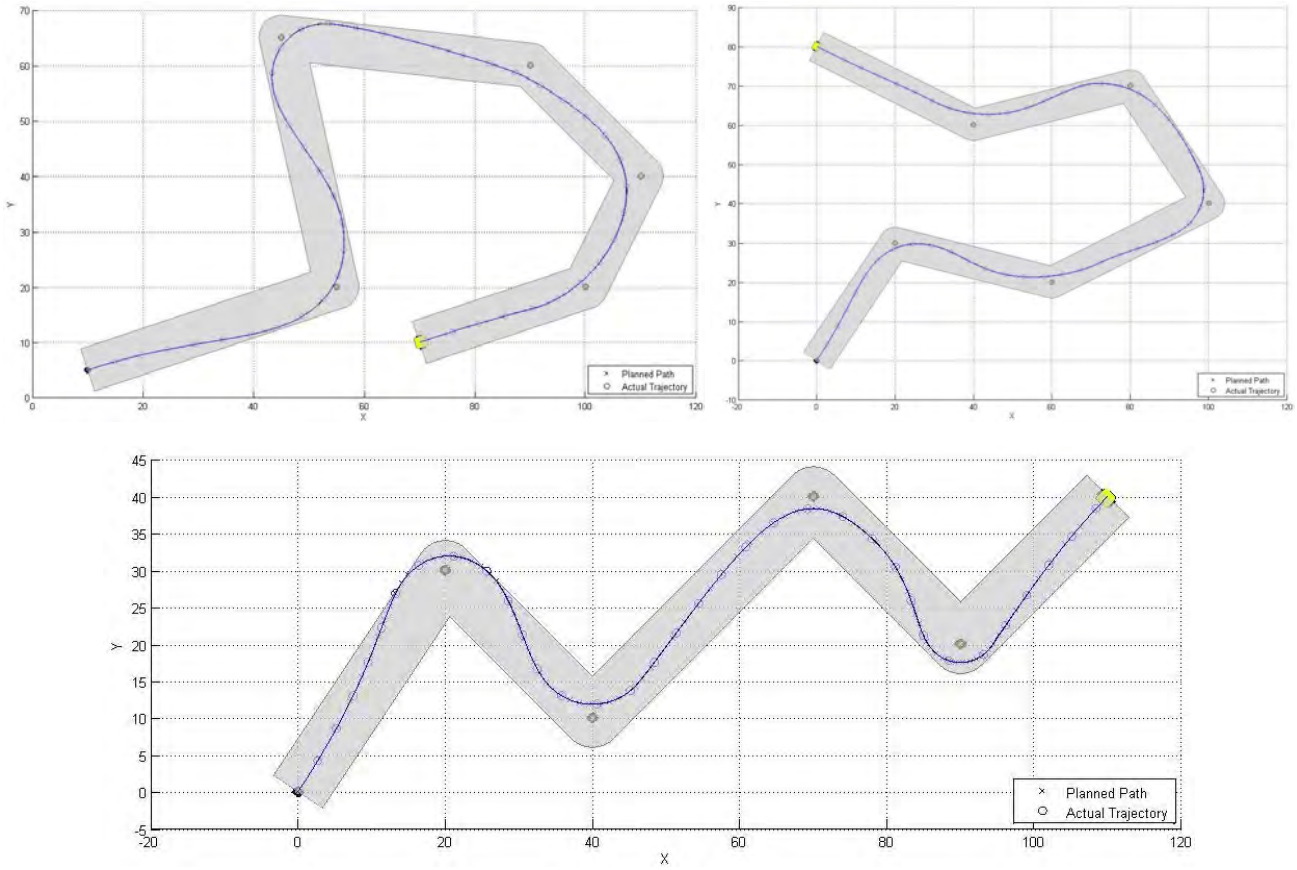


Figure 10. Tracking simulation results over arbitrary courses using *Bézier2* method.

[5] M. Lizarraga and G. Elkaim, *Spatially Deconflicted Path Generation for Multiple UAVs in a Bounded Airspace*. ION/IEEE Position, Location, and Navigation Symposium, ION/IEEE PLANS 2008, Monterey, CA, May 5-8, 2008.

[6] R. Rajamani, *Vehicle Dynamics and Control*. Springer, 2006.

[7] T. W. Sederberg, *Computer aided geometric design*. CAGD Course Notes, Brigham Young University, Provo, UT, 84602, April 2007.

[8] I. Skrjanc and G. Klancar, *Cooperative Collision Avoidance between Multiple Robots Based on Bézier Curves*. Information Technology Interfaces, 2007 (ITI 2007), p.451-456, June 25-28, 2007.

[9] *The 2005 DARPA Grand Challenge*, vol. 36/2007. p.363-405, Springer Berlin / Heidelberg, 2007.