



# Generation of Curvature Continuous Trajectories for Transport Aircraft using Bezier Curves

Héctor Escamilla Núñez, Felix Mora-Camino

## ► To cite this version:

Héctor Escamilla Núñez, Felix Mora-Camino. Generation of Curvature Continuous Trajectories for Transport Aircraft using Bezier Curves. NTCA 2017, 19th International Conference on New Trends in Civil Aviation, Dec 2017, Prague, Czech Republic. pp.7 / ISBN: 9780815376026, 10.1201/9781351238649 . hal-01631386

**HAL Id: hal-01631386**

**<https://enac.hal.science/hal-01631386>**

Submitted on 9 Nov 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Generation of Curvature Continuous Trajectories for Transport Aircraft using Bezier Curves

H. Escamilla Núñez & F. Mora Camino

*Laboratoire ENAC-Optim*

*École Nationale de l'Aviation Civile, France.*

**ABSTRACT:** With the increase in air traffic, better management and organization of it is crucial for increasing traffic safety and airspace capacity. Hence the need of ad-hoc descriptions of more complex and flexible aircraft trajectories, allowing high traffic densities and limiting environmental impact. The approach consists of a smooth 4D path generation from pre-existing control points by stitching several Bezier curves while assuring  $G^2$  continuity at the joints. Also, as the Euclidian distance between control points and the proposed trajectory is controlled by an optimal reshaping of the trajectory, the tradeoff between curvature-speed-load factor of the trajectory is considered. The generated trajectories are meant to be a complement to regular flight plans, helping to solve air traffic conflicts and enhance air capacity by better timing. Matlab simulations corroborate the feasibility of the approach, showing promising results when a distance range is defined for the reshaping algorithm.

## 1 INTRODUCTION

Nowadays, even aircraft with operating ACAS/TCAS are exposed to near mid air collisions, or in worst case scenarios, to events like the one occurred on July 1 over Überlingen, Germany in 2002, where two transport-category aircraft collided after TCAS instructed one pilot to climb, but the pilot descent in compliance to air traffic controller instructions. Moreover, even if only 1 in 10 Traffic Advisory (TAs) provided by TCAS result in a Resolution Advisory (RAs), this event and others <sup>1</sup>, are clear examples of human error and lack of homogeneity in TCAS.

As both SESAR and NextGen projects plan to implement new operational concepts which will reduce the spacing between aircraft. TCAS in its current form is not compatible with such concepts and would alert too frequently to be useful. Consequently, in order to allow aircraft to fly closer, quick and efficient 4D trajectory generation when needed is crucial. In this manner, air traffic conflicts can be avoided or addressed if present, taking into account load factor limits to look after passengers comfort.

The generation of flyable and efficient trajectories has been considered by several authors (Bakolas et al. 2011, Anderson et al. 2005, Judd & McLain 2001, Yang et al. 2015, Yang & Sukkarieh 2010, Delahaye et al. 2014). From a general view, reference trajectories are essential for flight plans which meet a large set of overfly or profile constraints, which vary in general from flight to flight. On the other hand, the generation of reference trajectories at short term should allow the implementation of procedures in the case of potential conflicts. Current path generation for transport aircraft is based on a sequence of objective points in a 2D/3D space given to connect two geographical locations (Walter 2015). Then, using diagrams like Voronoi, or Delaunay triangulation, or any other method, a piecewise path is constructed using straight lines denominated *legs*. This path may

---

<sup>1</sup>[https://www.skybrary.aero/index.php/Accident\\_and\\_Serious\\_Incident\\_Reports:\\_LOS](https://www.skybrary.aero/index.php/Accident_and_Serious_Incident_Reports:_LOS)

be offered by a high-level path planner from techniques such as Dijkstra's, A\*, probabilistic roadmaps, genetic algorithms, or Rapidly exploring random tree star (RRT\*), just to mention a few. However, as path differential requirements have to be satisfied, the initial trajectory needs to be reshaped in order to provide a flyable trajectory for transport aircraft. The problem of designing a flyable path over or close to the objective points while satisfying constraints such as maximum curvature and/or  $G^2$  continuity (curvature continuity), has been addressed using different approaches.

As a pioneer, Dubins assured that the shortest path between two points involves circles and straight line path segments (Dubins 1957). However, curvature continuity at the joints of lines and circular arcs is not satisfied. Techniques to solve this track transition problem are addressed using Clothoids (Scheuer & Fraichard 1997), but as they do not have closed-form expressions, the computation complexity is increased. Circular arcs as transitions tracks have been also proposed (Anderson et al. 2005), where a 2D real-time trajectory is generated satisfying curvature and velocity constraints. Also, the deviation between the generated circular path and the associated control points is minimized using a parameter  $\kappa \in [0, 1]$ . If  $\kappa = 0$ , the distance is zero, so the plane will fly exactly over the waypoint, and if  $\kappa = 1$ , minimum-time transitions between control points are achieved. However, the election of this parameter  $\kappa$  becomes an issue when a range of distance wants to be elected as a permitted deviation from the generated path. Dubins' approaches generate paths limited to straight lines and arcs of circles without parametrization. Thus, the method shows disadvantages being able to generate several arcs without curvature continuity or with high computation complexity.

Another tool to generate smooth flyable paths are splines, defined by series of low order polynomials. In (Judd & McLain 2001), after finding the best path from a UAV position location to a target location using a Voronoi diagram and a Dijkstra's algorithm, the 2D path is smoothed using cubic splines. It is worth to say that the optimal locations of the middle knots of a spline are crucial for the shape of the segments. However, the optimal computation of these points is very time-consuming unless a set of cases are defined a priori.

As an attractive approach, Bezier curves are able to generate continuous-curvature paths, having the advantage of passing through initial and final points while the whole curve always remains within the convex hull that is built by the control points. Also, the initial and final piecewise straight lines of the control polygon are always tangent to the Bezier curve at the starting and ending control points. An example is given in (Lee et al. 2016), where the authors present the generation of a 3D path obtained from a combination of Rapidly exploring random tree star (RRT\*) using a  $C^2$  class Bezier curve, and Dynamic Movement Primitives (DMP's), that allow cooperative aerial manipulators to avoid known and unknown obstacles. As the final Bezier curve is a barycentric combination of the polygon vertices formed by the control points, it may be thought that a trajectory could not be accurately generated if the control points positions are zig-zags like. However, this aspect can be handled by stitching several Bezier curves to form a bigger path while assuring  $G^2$  continuity at the joints. In (Yang et al. 2015), after a Guiding Attraction based Random Tree (GART) is used to obtain control points in a 2D plane, a kinematic smoother based on sixth order Bezier curves to achieve second derivative continuity (curvature continuity) is proposed. Finally, a local optimal reshaping of the path, minimizing length and curvature cost is performed. In (Yang & Sukkarieh 2010), it is presented an algorithm based on cubic Bezier curves for 3D path smoothing, satisfying  $G^2$  continuity and maximum curvature constraints, where instead of addressing a direct solution for the 3D path smoothing, a 2D path smoothing for consecutive triplets of control points is applied, considering each triplet as a 2D plane; Nevertheless, deviation from the resulting trajectory w.r.t. the control points is not controlled.

In this paper, assuming velocity/time constraints, a time-parametrized smooth trajectory is developed by stitching several Bezier curves while assuring  $G^2$  continuity at the joints, also, the Euclidian distance between the control points and the proposed trajectory is controlled, yielding an optimal 4D trajectory valid for 4D guidance. Finally, since flyable trajectories depend heavily on the load factor, the tradeoff between curvature-speed-load factor is analysed.

## 2 TRAJECTORY GENERATION

### 2.1 Bezier Curves and $G^2$ Continuity

A Bezier curve  $P(s)$  of degree  $n$ , obtained from  $n + 1$  control points  $(P_0, \dots, P_n)$ , is described with its derivatives by

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

$$P'(s) = \sum_{i=0}^{n-1} B_i^{n-1}(s) Q_i \quad (2)$$

$$P''(s) = \sum_{i=0}^{n-2} B_i^{n-2}(s) (n(n-1)(P_{i+2} - 2P_{i+1} + P_i)) \quad (3)$$

where  $Q_i = n(P_{i+1} - P_i)$  and  $B_i^n(s)$  is the  $i^{th}$  Bernstein polynomial of degree  $n$ , given by

$$B_i^n(s) = \binom{n}{i} s^i (1-s)^{n-i} \quad i \in \{0, 1, \dots, n\} \quad (4)$$

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \quad (5)$$

Note that

$$P(0) = P_0 \quad P(1) = P_n \quad (6)$$

$$P'(0) = n(P_1 - P_0) \quad P'(1) = n(P_n - P_{n-1}) \quad (7)$$

$$P''(0) = n(n-1)(P_2 - 2P_1 + P_0) \quad (8)$$

$$P''(1) = n(n-1)(P_n - 2P_{n-1} + P_{n-2}) \quad (9)$$

According to (Barsky & DeRose 1984),  $G^2$  continuity is the second-order geometric continuity, implying the second derivative continuity of two curves at the joint. Moreover, if  $P(s) = (x(s), y(s), z(s))$  is a regular parametrization of a curve in an interval  $I = [a, b] \rightarrow R^3$ . An expression for the curvature is obtained by:  $\kappa(s) = \frac{\|P'(s) \times P''(s)\|}{\|P'(s)\|^3} \geq 0$ . If  $\kappa(s) = 0$ ,  $P(s)$  is a straight line. Otherwise, the curve will have a curvature radius of  $1/\kappa(s)$ . Thus, let a Bezier curve  $C(s)$  with  $m+1$  control points  $(C_0, \dots, C_m)$ , and a second Bezier curve  $D(s)$  with  $n+1$  control points  $(D_0, \dots, D_n)$  be joined.  $C^0$  continuity is guaranteed if:

$$C_m = D_0 \quad (10)$$

A smooth transition is assured if (10) is satisfied and  $C_m, D_0, C_{m-1}, D_1$  are on the same line. Furthermore,  $C^1$  continuity is guaranteed if the tangent vector of the first curve at  $s = 1$  is identical to the tangent vector of the second curve at  $s = 0$ , meaning that:

$$C'(1) = m(C_m - C_{m-1}) = n(D_1 - D_0) = D'(0) \quad (11)$$

This states that the ratio  $\left(\frac{C_m - C_{m-1}}{D_1 - D_0}\right)$  must be  $\frac{n}{m}$ . Since  $n$  and  $m$  are fixed numbers, the positions of  $C_{m-1}$  and  $D_1$  can be rearranged to be not only at the same line, but also at the proper distance to assure  $C^1$  continuity. In the same tenor,  $C^2$  and  $G^2$  continuity are guaranteed at the joint if it is verified that

$$\begin{aligned} C''(1) &= m(m-1)(C_m - 2C_{m-1} + C_{m-2}) \\ &= n(n-1)(D_2 - 2D_1 + D_0) = D''(0) \end{aligned} \quad (12)$$

Hence, to assure  $G^2$  continuity, the locations of  $C_m, C_{m-1}, C_{m-2}, D_0, D_1, D_2$ , where (10)-(12) are satisfied, need to be proposed.

## 2.2 Path Generation and Time-parametrization

In this paper, the 3D path is decomposed into several 2D planar trajectories laying on the tangent/normal plane of the frenet frame, similar to (Yang & Sukkarieh 2010). Since three control points are required to form a plane, at least three control points are defined, and then divided into triplets. Several quintic Bezier curves, one for each triplet of control points, are stitched together to form a bigger path while assuring  $G^2$  continuity at the joints. For a triplet of control points  $(P_1, P_2, P_3)$ ,  $G^1$  continuity can be achieved by interpolating four points  $(Q_0, Q_1, Q_4, Q_5)$ , and for  $G^2$  continuity, six points are interpolated  $(Q_i; i \in \{0, \dots, 5\})$ . The points are interpolated as follows:  $Q_0$  and  $Q_5$  are defined to be at the middle point of  $(P_1, P_2)$  and  $(P_2, P_3)$  respectively, forcing a past and following Bezier curve formed by the past and next triplet of control points, to finish and start at  $Q_0$  and  $Q_5$ . In other words, a next triplet of points conformed by  $(P_2, P_3, P_4)$  will have a Bezier curve starting at  $Q_5$ , so (10) is fulfilled. For  $Q_1$  and  $Q_4$ , they are computed to be also in the same line of  $(P_1, P_2)$  and

$(P_2P_3)$  respectively, but separated by a  $\delta_1$  distance from  $Q_0$  in the case of  $Q_1$ , and a distance  $\delta_2$  from  $Q_5$  in the case of  $Q_4$ . Note that the number of auxiliary interpolated points to compute the Bezier curves are the same for all the triplet of points, so  $m$  and  $n$  from (11) are equal. Hence, to fulfill  $G^1$  continuity, only remains to guarantee that the distance  $\delta_2$  of a certain triplet of control points is equal to the distance  $\delta_1$  of the next triplet of control points. Finally, if the positions of  $Q_2$  and  $Q_3$  are computed also in the same line of  $(P_1P_2)$  and  $(P_2P_3)$  respectively, separated by the same distances  $\delta_1$  from  $Q_1$ , and  $\delta_2$  from  $Q_4$ , if it is proposed:  $\delta_1 = \frac{P_1P_2}{4}$ , and  $\delta_2 = \frac{P_2P_3}{4}$ , equations (11), (12) are satisfied. Besides, knowing that the first and last Bezier curves conforming the total path will start and end at the middle point of their corresponding control points, as straight lines have zero curvature, the path can be completed with straight lines without affecting the  $G^2$  continuity. After the interpolation of these auxiliary control points, a quintic Bezier curve is adjusted to them in the case of the  $G^2$  continuity path.

As several Bezier curves conform the total path, and each Bezier curve is parametrized by  $s \in [0, 1]$ , a time-parametrization needs to be done for the creation of a flyable path for transport aircraft. This re-parametrization consists in perform the operation:  $s = \frac{t-t_i}{t_{i+1}-t_i}$  where  $i \in [0, 1, \dots, B_n]$ , considering  $B_n$  as the number of existent Bezier curves, such that the multiple Bezier curves parametrized by  $s \in [0, 1]$ , can be used at their proper time intervals  $[t_i, t_{i+1}]$ . Note that the times in which each Bezier curve is used need to be known to paramterize the curve, therefore, the velocity for the aircraft to follow the path, or as an alternative way, the timestamps in which the aircraft is supposed to fly over the joints of the Bezier curves, are defined in advance, thus, the computation of the arc length of each Bezier curve is inherent.

### 3 OPTIMAL RESHAPING AND LOAD FACTOR OF THE TRAJECTORY

Depending on the application, some aircraft may need to fly directly over the control points or at least near a defined range of distance from the control points. In order to achieve this demand, the Euclidian distance from the  $G^2$  continuity path w.r.t. the nearest control point is controlled. For this special case, an extra auxiliary point is computed such that an optimum path based on sixth order Bezier curves is obtained. The other auxiliary control points are computed as before in order to do not affect curvature continuity, but the position of the heptic auxiliary point is moved gradually until the generated path passes within a distance range defined by the user. Considering that  $(Q_0, Q_1, Q_2, Q_4, Q_5, Q_6)$  are the necessary points to achieve  $G^2$  continuity, the auxiliary point ( $Q_3$ ) is moved away in the direction of:

$$aux = \frac{(Q_2 - Q_1) + (Q_4 - Q_5)}{\|(Q_2 - Q_1) + (Q_4 - Q_5)\|} \quad (13)$$

As transport aircraft are designed to flight smooth trajectories, some limitations on the load factor have to be respected for the well being and comfort of passengers. According to (FAA 2017), on a typical flight, the load factor is limited to +2.5g and -1g for regular maneuvers. The expressions relating the load factor and speed of a turning/pitching aircraft, with the horizontal path of radius  $R$ , and radius of vertical curvature  $R'$  are:

$$R = \frac{V_a^2}{g\sqrt{n_z^2 - 1}} \quad q = \frac{(n_z - 1)g}{V_a} = \frac{V_a}{R'} \quad (14)$$

As the path has a curvature radius of  $1/\kappa$ , the curvature  $\kappa$  of the path is directly related to (14). In this manner, for independent lateral or vertical maneuvers, the path that an aircraft can follow without infringing load factor limits can be computed, taking into account the speed of the aircraft and the curvature of the path generated by the position of the control points. Then, path constraints regarding to maximum curvature can be established for different flyability requirements based on load factor limits. According to (14), maximum radius of curvature for circular motions can be computed for different speeds after defining load factor limits. For example, a 2.5g load factor at 200m/s is generated by a circular trajectory of a radius of 1,779.55m, equivalent to a roll angle of  $66.5^\circ$ , which is also the maximum roll angle permitted for transport aircraft.

### 4 SIMULATION RESULTS

In Figure 1, a  $G^1/G^2$  continuity path for a triplet of points  $(P_1, P_2, P_3)$  is shown. The locations of the auxiliary points is according to Subsection 2.2, as well as the straight lines in the extremes to complete the path without affecting the curvature continuity. Consequently, consider that some control points emulating the avoidance of potential threats are given in Table 1. They generate a  $G^1$  and  $G^2$  path depicted in Figure 2. Regarding to the

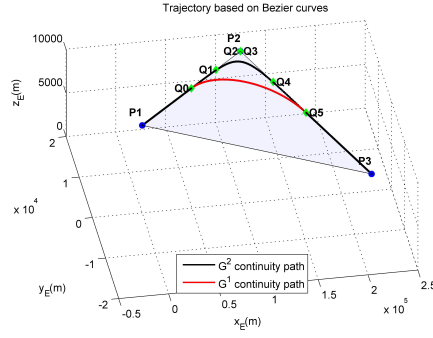


Figure 1: Bezier curve completed with straight lines.

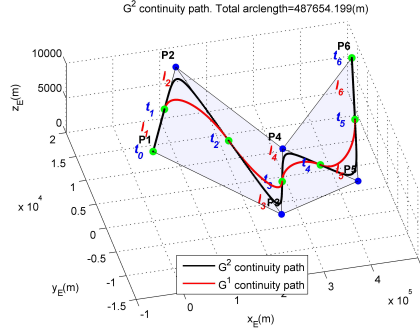


Figure 2: Arc lengths and joints of Bezier curves.

joints of the  $G^2$  Bezier curves, and assuming that the path is intended to be followed at  $200m/s$  velocity, Figure 2 shows the arc lengths ( $l_i; i \in [1, 6]$ ) and their times ( $t_j; j \in [0, 6]$ ) assigned. The values of  $l_i$  and  $t_j$  are given in Table 2. Then, comparing the curvatures of the  $G^1$  and  $G^2$  continuity paths (Fig. 3), it is clearly seen how the  $G^2$  path has continuity in the curvature at the joints of the different Bezier curves, and that these joints occur at the assigned time, represented by asterisks in the correspondent axis. Concerning to the optimal reshaping of the path, in Figure 4, a  $100m$  maximum deviation for a triplet of control points is commanded, meaning that the optimized path will be generated such that it will pass no further than  $100m$  away the control point  $P_2$ . In this case, the optimizing algorithm stopped at a distance of  $92.48m$ . The distance at which the algorithm stops depends on the step size in which the extra auxiliary control point ( $Q_3$ ) is moved away in the direction of  $a\vec{u}x$ . For a small step size, the path will be closer to the distance defined by the user, in this case  $100m$ , but the computation time will be increased. Now, for a complete trajectory, using the control points of Table 1, a  $100m$  maximum deviation is commanded, the resultant path is shown in Figure 5. The Euclidian distance at which the trajectory is generated from the closest control points is given by Table 3. Assuring that the maximum deviation distance is achieved. To better exemplify the relation of the load factor w.r.t. the velocity, let's propose a didactic scenario of the trajectory of an aircraft changing airways by changing its heading. Three control points located at the same altitude and separated by 50 nautical miles (nm) from each other are used (Table 4). The control points are shown in Figure 6, along with the computed Bezier curves of  $G^2$  continuity. Also, the arc lengths of each curve and the points where the Bezier curves are joined, denoted by their timestamp, are

Table 1: Control Points.

	X(m)	Y(m)	Z(m)
1	0	0	10,000
2	120,843	16,983	9,300
3	210,332	-14,779	9,000
4	272,744	-759	8,200
5	388,920	-11,130	9,500
6	478,501	12,964	9,800

Table 2: Time and Arc length.

	Arc length(m)	Time(s)
		$t_0$
$l_1$	61,016	$t_1$
$l_2$	107,536	$t_2$
$l_3$	78,523	$t_3$
$l_4$	89,990	$t_4$
$l_5$	104,206	$t_5$
$l_6$	46,383	$t_6$

Table 3: Distance of trajectory from control points.

From	$P_2$	$P_3$	$P_4$	$P_5$
Initial distance (m)	2800.77	2351.04	1449.10	2010.67
Optimized distance (m)	92.48	99.72	96.09	79.53

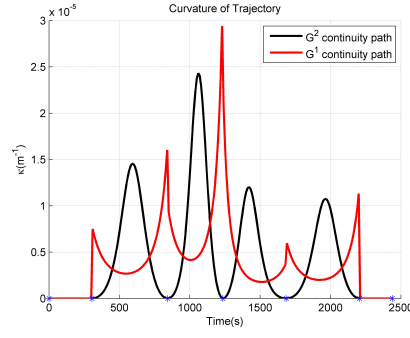


Figure 3: Curvature of the  $G^1$  and  $G^2$  continuity paths.

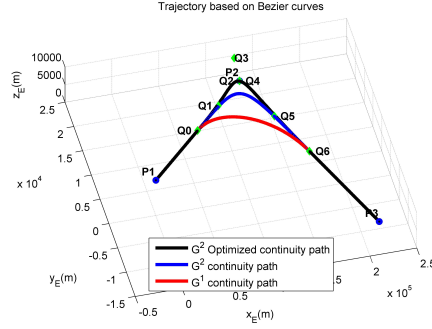


Figure 4: 100m deviation optimized path using  $Q_3$ .

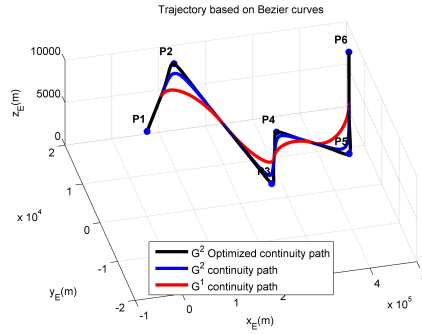


Figure 5: 100m deviation optimized path.

Table 4: Control points for an aircraft changing airways.

1nm=1852m	X(m)	Y(m)	Z(m)
$P_1$	0	92,600	10,000
$P_2$	92,600	92,600	10,000
$P_3$	92,600	0	10,000

Table 5: Arc lengths and times at different velocities

Arc length(m)		Time(s)		
		170m/s	200m/s	230m/s
$l_1$	46,300	$t_0$	0	0
		$t_1$	272.35	201.30
$l_2$	84,571	$t_2$	769.83	569
$l_3$	46,300	$t_3$	1,042.18	770.3

shown. Now, let's consider three different speeds for an aircraft to go through these control points, the numerical values of the timestamps where the Bezier curves are joined change according to Table 5. Also, the transition between legs will generate different load factors depending on the speed, shown in Figure 7. On the other hand, independent from the speed, the curvature of the path remains constant. Once that the relation of the load-factor and velocity is stated, let's explore the curvature constraints due to load factor limits keeping a constant velocity. In the example above, the distance from the trajectory to the point  $P_2$  is 7,161.66m, so, let's propose the trajectory of an aircraft turning closer to  $P_2$  with a constant velocity of 200m/s. If the optimal reshaping of Section 3 is asked to modify the trajectory such that a maximum of 10m deviation from  $P_2$  is achieved, the

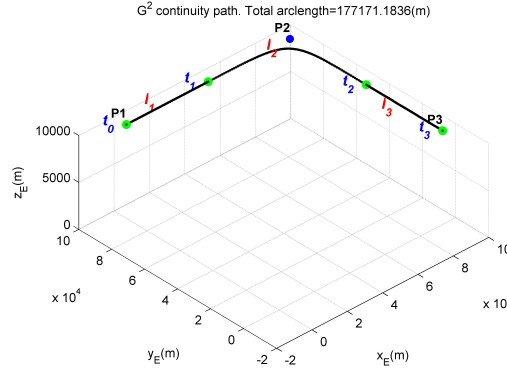


Figure 6: Control points of example with Arc lengths and Times.

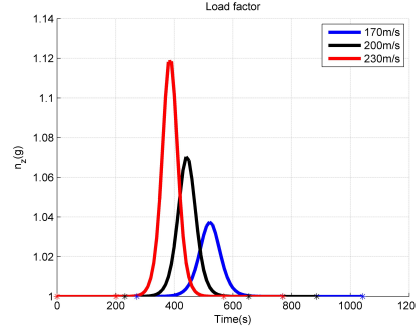


Figure 7: Load factor at different velocities.

Table 6: Auxiliary control points to force a high load factor

	X(m)	Y(m)	Z(m)
$P_{aux1}$	77,043.2	92,600	10,000
$P_{aux2}$	92,600	77,043.2	10,000

load factor remains within bounds. This scenario is depicted in Figure 8, where a zoom to  $P_2$  shows how the generated trajectory is bended before and after the control point to satisfy the maximum deviation constraint. The load factor of the generated trajectory reaches a pike of  $\approx 1.425g$  and a curvature of  $\approx 2.5 \times 10^{-4} (m^{-1})$ . The distance from this trajectory to  $P_2$  is 1.36m.

Consequently, in order to have a trajectory close to the control point  $P_2$  while increasing significantly the load factor, two auxiliary control points are added (see Table 6), such that the direction of the path is not bended, like in the case when the optimal reshaping is done. A zoom close to the control point  $P_2$  shows the generated trajectory, which passes at a distance of 1,203.15m from  $P_2$  (Fig. 8). Under this conditions, the load factor reaches the 2.5g limit. The curvature of the path pikes at  $\approx 5.6 \times 10^{-4} (m^{-1})$ , close enough to the 1,779.55m radius limit stated at the end of Section 3. Thus, if an aircraft at 200m/s is commanded to pass through "L" shape control points, a single turn maneuver leaves the aircraft at 1,203.15m from the middle control point before infringing load factor limits, on the other hand, the proposed reshaping algorithm generates a flyable trajectory at 1.36m from the same point. Numerical computations can be done for any other velocity and the pitching motion.

## 5 CONCLUSIONS

As the curvature of the path is forced to be zero at the joints between Bezier curves,  $G^2$  continuity is assured. As the time interval for each Bezier curve needs to be known for the time-parametrization, velocity/time constraints are assumed, leading to an inherent computation of the arc length for each Bezier curve. The closer the optimal path is to the control points, the bigger the curvature becomes, such that a maximum curvature is defined and controlled by the deviation distance defined by the user, establishing a tradeoff between curvature-speed-load factor of the trajectory. This approach for 4D smooth trajectories generation, opens the way to 4D guidance along the time-parametrized Bezier curves under path and flyability constraints. The adoption of these trajectories should ease air traffic management in congested areas, helping to solve air traffic conflicts and improving



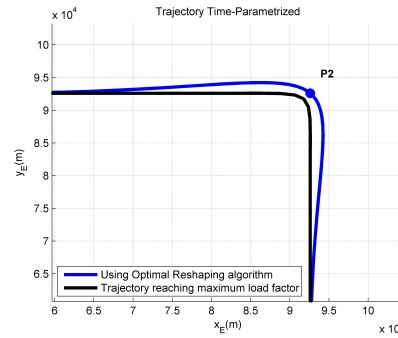


Figure 8: Trajectories close to  $P_2$  with a 10m maximum deviation and forcing a maximum load factor.

on-board guidance systems performance.

## REFERENCES

- Anderson, E. P., R. W. Beard, & T. W. McLain (2005). Real-time dynamic trajectory smoothing for unmanned air vehicles. *IEEE Transactions on Control Systems Technology* 13(3).
- Bakolas, E., Y. Zhao, & P. Tsiotras (2011). Initial guess generation for aircraft landing trajectory optimization. *AIAA Guidance, Navigation, and Control Conference, Portland, Oregon, USA*.
- Barsky, B. A. & T. D. DeRose (1984, October). Geometric continuity of parametric curves. technical report no. ucb/csd 84/205. *University of Berkeley, USA*.
- Delahaye, D., S. Puechmorel, P. Tsiotras, & E. Feron (2014). *Mathematical Models for Aircraft Trajectory Design: A survey*. In: *Electronic Navigation Research Institute (eds) Air Traffic Management and Systems. Lecture Notes in Electrical Engineering*, Volume 290. Springer.
- Dubins, L. E. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics* 79(3), 497–516.
- FAA (2017). *Code of Federal Regulations (CFR)-Title 14 Aeronautics and Space*. FAA.
- Judd, K. B. & T. W. McLain (2001). Spline based path planning for unmanned air vehicles. *AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada*.
- Lee, H., H. Kim, & H. J. Kim (2016). Planning and control for collision-free cooperative aerial transportation. *IEEE Transactions on Automation Science and Engineering* PP(99), 1–13.
- Scheuer, A. & T. Fraichard (1997). Continuous-curvature path planning for car-like vehicles. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Walter, R. (2015). *Digital Avionics Handbook. Chapter 24* (3rd ed.). CRC Press.
- Yang, K. & S. Sukkarieh (2010). An analytical continuous-curvature path-smoothing algorithm. *IEEE Transactions on Robotics* 26(3).
- Yang, L., D. Song, J. Xiao, J. Han, L. Yang, & Y. Cao (2015). Generation of dynamically feasible and collision free trajectory by applying six order bezier curve and local optimal reshaping. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany*.