Bézier Curve Path Planning for Parafoil Terminal Guidance

Lee Fowler
Texas A&M University, College Station, Texas 77843 and
Jonathan Rogers
Georgia Institute of Technology, Atlanta, Georgia 30332
DOI: 10.2514/1.I010124

Autonomous parafoil terminal guidance plays a critical role in landing accuracy but is inherently difficult due to underactuation and path disturbances caused by winds. Terminal path planners must generate flight paths that deliver the parafoil as close as possible to the destination while landing into the wind. This paper presents a novel path-planning scheme capable of highly general trajectory shapes that enable successful path planning from a wide set of initial conditions in constrained three-dimensional environments. The path-planning approach parameterizes the two-point boundary value guidance solution in the form of one or more Bézier curves that connect the current parafoil location with a final approach point downwind of the target. Online nonlinear optimization of curve parameters allows for path regeneration given changing winds. Furthermore, generality of the trajectory shape provides a wide set of initial conditions at which terminal guidance may be initiated in obstacle-constrained scenarios, reducing coupling between landing accuracy and terrain constraints in the target area. Additional advantages include yaw-angle smoothness guarantees and ease of path-length optimization. Following a description of the path generator, six-degree-of-freedom simulation results show example cases in which the path planner computes terminal guidance solutions in realistic terrain with changing winds. Monte Carlo simulations verify robustness of the proposed trajectory generation technique.

Nomenclature

- \( d_{mid} \) = two-dimensional distance from Bézier curve junction point to target
- \( d_i \) = \( i \)th control point magnitude
- \( f_{ok} \) = optimizer relative convergence threshold
- \( H_p \) = model predictive control prediction horizon
- \( k_m, k_{cp} \) = path planner penalty function gains
- \( k_{yr} \) = linear coefficient relating parafoil turn rate to descent speed
- \( N \) = number of points used to discretize Bézier curve
- \( N_c \) = number of Bézier curves used to create a terminal guidance path
- \( Q, R \) = model predictive control weighting matrices
- \( t_{pre} \) = model predictive controller preturn value
- \( V_{ss} \) = parafoil steady-state airspeed
- \( V_z \) = parafoil steady-state descent speed
- \( W_x, W_y, W_z \) = wind components in the inertial frame
- \( x, y, z \) = parafoil inertial coordinates
- \( x_t, y_t, z_t \) = target inertial coordinates
- \( \zeta_{mar} \) = parafoil altitude margin
- \( \zeta_{miss} \) = estimated altitude miss distance for a candidate terminal guidance path
- \( \zeta_r \) = altitude miss distance threshold for curve acceptance
- \( \gamma \) = parafoil heading angle
- \( \xi \) = parafoil crab angle
- \( \psi \) = parafoil yaw angle

I. Introduction

Autonomous parafoils are attractive for airdrop resupply missions in that they may be released far from the target yet arrive with considerable accuracy. This large standoff distance reduces risk to cargo aircraft, which can maintain long distances from the drop zone to enhance safety or efficiency. However, unknown or changing wind conditions inherently limit precision airdrop accuracy. Although the strategy for guidance from release to touchdown may be varied, most guidance systems employ an energy-management phase, where the parafoil dissipates altitude and estimates wind, a homing phase, in which the parafoil approaches the target area, and a terminal phase, which consists of final maneuvers such that the parafoil impacts the target while flying into the wind [1]. Terminal guidance is often most critical to landing accuracy in that there is little opportunity to correct for guidance errors, given wind disturbances. Several factors inherently limit landing accuracy. First, parafoils are underactuated vehicles in that traditionally only lateral control is available and glide slope control is extremely limited. Second, wind disturbances occur at the same order of magnitude as the vehicle airspeed, resulting in flight paths that are very sensitive to changes in nominal
winds. It is reasonably common that wind speeds approach vehicle airspeeds, making upwind flight difficult if not impossible. Uncertainty in wind conditions as well as obstacle or terrain limitations mean that the parafoil may enter terminal guidance from a wide array of initial conditions, and thus general trajectory shapes may be required to provide robust path-planning solutions especially in complex drop zones with three-dimensional path limitations.

Simple parafoil terminal guidance strategies have been proposed in which the parafoil spirals over the target area and proceeds in a straight line path to the target at an optimal time [1,2], although it has been shown that landing into the wind cannot always be guaranteed [3]. Several other guidance schemes have been proposed including the use of trajectory databases, model predictive control, or direct glide slope control [4–8]. Recently, optimal trajectory paths have been explored, such as a bandwidth-limited trajectory planner proposed by Carter et al. [9]. Rademaker et al. used a hybrid optimal control scheme combined with modified Dubins paths [10] to accomplish terminal path planning, while Slegers and Yakimenko developed a terminal guidance law using calculus of variations and inverse dynamics in a virtual domain [11]. Most recently, Rogers and Slegers [12,13] developed a method to generate terminal trajectory paths as the solution to a very general nonlinear minimization problem. These guidance solutions were meant to protect against unknown future wind disturbances. To reduce computational complexity, the terminal guidance law was parameterized as a constant-rate turn, and the path was recomputed at discrete intervals during terminal guidance.

Calise and Preston [14] developed a swarming parafoil control law that included a collision-avoidance algorithm. The guidance law used a zero-effort miss calculation to determine collision probabilities between vehicles. Many previous studies (with the exception of [13]) have not specifically addressed the problem of three-dimensional obstacle avoidance, which can be a primary guidance constraint in mountainous or urban terrain. Difficulty of the guidance problem is compounded further if terminal guidance must be initiated from less-than-optimal initial conditions due to wind uncertainty or terrain constraints. Oftentimes, landing accuracy suffers in direct proportion to the severity of terrain constraints in the target area.

New terminal guidance schemes offering increased generality of the trajectory shape may provide an easier mechanism for obstacle avoidance in these challenging scenarios while still minimizing miss distance and enforcing dynamic constraints.

The path-planning methodology proposed here parameterizes the parafoil terminal flight path as a sequence of Bézier curves. Use of splines and Bézier curves for path generation is not a new idea and has been explored in both the ground and aerial robotics communities. Magid et al. [15] developed a spline-based path planner for ground robots, with nonlinear optimization of path parameters performed through the Nelder–Mead simplex method. Choi et al. [16] used a sequence of Bézier curves for ground robot path planning, accounting for waypoint and corridor constraints. Connors and Elkhairi [17] proposed a similar algorithm using cubic splines. For air vehicle applications, Jung and Tsiotras [18] have proposed a planar path-generation scheme based on a library of B-spline path templates, which may be joined together to form a global path. Yang and Sukkarieh [19] developed a UAV path planner capable of obstacle avoidance using a series of linear segments connected by Bézier curves to enhance smoothness. Macharet et al. [20] and Mittal and Deb [21] have derived full three-dimensional path planners using Bézier curves, in which curves are optimized using evolutionary algorithms. Although these authors have successfully employed spline-based planners for air and ground vehicle applications, no work to date has been performed in the parafoil community exploring how such guidance parameterizations may be used to improve performance of autonomous parafoil systems.

This paper extends previous work on spline-based path planning to the parafoil terminal guidance problem. The algorithm developed here seeks to create an optimized, flyable path from any arbitrary point in the terminal area to a final approach point downwind of the target, assuming such a path exists. The path is composed of one or more Bézier curves whose parameters are optimized to minimize the touchdown distance error from the target. The curves are generated under hard constraints on maximum curvature and penalties on other parameters which stimulate convergence to a viable solution. Terrain obstacles are considered as nonlinear constraints within the optimization function. The algorithm is built as a tiered optimizer in which a path is first computed using a single Bézier curve. If no feasible path is found, increasing numbers of Bézier curves are chained together until convergence to a feasible path is achieved. Resulting paths can be regenerated quickly as winds shift and previous paths become suboptimal. The primary advantage of the proposed method is that the trajectory shape of the solution is more general than in many previous methods and can accommodate a wide variety of initial positions and heading angles with respect to the wind-target geometry at initiation of terminal guidance. This path flexibility enables convergence to a feasible solution even when the guidance path is severely constrained by three-dimensional terrain or obstacles, potentially reducing coupling between landing accuracy and the presence of drop zone obstacles. The natural price to pay for this flexibility in path geometry is a higher computational burden compared to previous methods (such as that proposed in [11]) and potential convergence issues. However, detailed simulation studies conducted here demonstrate that real-time execution and robust convergence may be expected for onboard implementations in many realistic terrain and wind scenarios.

The paper begins by reviewing some characteristics of cubic Bézier curves given fixed end points and control points. Then, the path-planning process is described in which control point locations are varied to enforce path-length and curvature constraints as well as various other optimization criteria. Example terminal guidance paths are generated for several approach scenarios in which three-dimensional terrain limits the approach path and winds vary spatially and temporally. Feasibility of these paths is evaluated using a six-degree-of-freedom (6DOF) parafoil model. Finally, Monte Carlo simulations are conducted to demonstrate the method’s robustness given randomized approach scenarios and changing wind conditions during terminal guidance.

II. Parafoil Terminal Path Planning with Bézier Curves

Autonomous parafoil guidance is initiated by release from the cargo aircraft. Following successful deployment and inflation of the parafoil canopy, a standard guidance strategy is typically executed in the form of an energy-management phase, a homing phase, and a terminal guidance phase, as illustrated in Fig. 1. Energy management consists of some type of loitering maneuver in which winds are estimated and an optimal altitude is computed to enter the homing phase. Once the parafoil has descended to this optimal altitude, the homing phase is initiated in which the parafoil proceeds approximately downwind toward the target area to a turn initiation point (TIP). Upon arrival at the TIP, terminal guidance is initiated in the form of final maneuvers that allow the parafoil to achieve ground impact at or near the target while flying into the wind. Oftentimes terminal maneuvers will culminate in arrival at a final approach point (FAP), after which the parafoil flies straight to the target to achieve landing in a level attitude. Note that energy management and homing do not play as crucial a role in precision landing as terminal guidance because errors in these initial phases may be corrected during terminal flight. Furthermore, note that this downwind-to-upwind approach scenario is highly idealized; given wind shifts, release errors, or problems with canopy inflation, many actual scenarios may involve crosswind approaches or other suboptimal geometries. Terminal path planners must be flexible enough to deal with such scenarios.

This paper describes an algorithm to generate smooth, flyable paths between the TIP and FAP. All scenarios considered in this paper assume that energy management and homing have been completed, although not necessarily in an optimal fashion. As a result, terminal guidance may be initiated from a wide variety of initial conditions, some of which may require highly flexible path geometries that differ substantially from the single half-turn path shown in Fig. 1.
A. Bézier Curve Background

A cubic Bézier curve can be completely defined by two control points and two end points. A line from the end point to its corresponding control point dictates the slope of the curve at that end point. In the terminal guidance scenario, two independent control points are required because heading angles at both end points are geometrically constrained (the first by the parafoil’s initial heading, the second by the wind direction). This necessitates use of a cubic Bézier curve rather than, for instance, a quadratic Bézier curve, which has a single control point that serves both end points. Given end points \((x_1, y_1)\) and \((x_2, y_2)\) and their respective control points \((x_1^*, y_1^*)\) and \((x_2^*, y_2^*)\), a cubic Bézier curve is defined by the following parametric equations:

\[
\begin{align*}
  x &= a_x \tau^3 + b_x \tau^2 + c_x \tau + x_1 \\
  y &= a_y \tau^3 + b_y \tau^2 + c_y \tau + y_1
\end{align*}
\]

where \(0 \leq \tau \leq 1\). Note that the independent curve parameter \(\tau\) is a nonlinear function of time along the path, and the relationship between the two is determined numerically as described in the next section. The coefficients in Eqs. (1, 2) are given by the following equations:

\[
\begin{align*}
  c_x &= 3(x_2^* - x_1) \\
  b_x &= 3(x_2^* - x_1) - c_x \\
  a_x &= x_2 - x_1 - c_x - b_x
\end{align*}
\]

with analogous expressions for \(a_y, b_y,\) and \(c_y\). Examples of several cubic Bézier curves are shown in Fig. 2 with end points and control points labeled. Note the following properties of Bézier curves, proofs of which may be found in [22].

Property 1: End-point interpolation property: Given a cubic Bézier curve \(B(t)\) with end points \(b_0, b_3\) and control points \(b_1, b_2\), it can be shown that \(B(0) = b_0\) and \(B(1) = b_3\).

Property 2: End-point tangent property: Given a cubic Bézier curve \(B(t)\) with end points \(b_0, b_3\) and control points \(b_1, b_2\), it can be shown that \(B'(0) = 3(b_1 - b_0)\) and \(B'(1) = 3(b_3 - b_2)\). Thus, a line connecting points \(b_0\) and \(b_1\) is tangent to the Bézier curve at \(B(0) = b_0\), and likewise a line connecting \(b_2\) and \(b_3\) is tangent to the Bézier curve at \(B(1) = b_3\).

These properties are used in the next section to ensure parafoil position and yaw-angle continuity throughout a generated guidance path. Note that the Bézier curve formulation provided here ensures yaw angle \(C^1\) continuity only, meaning that although yaw-angle continuity is guaranteed, yaw rate continuity is not [19]. It is also important to note that other path parameterizations, including many spline formulations and the constant-rate turn employed as the open-loop path in [12], do not guarantee \(C^1\) continuity in yaw angle given fixed locations of the path end points at the parafoil’s current position and the FAP.

B. Bézier Curve Parafoil Path Generation

A single Bézier curve terminal flight path is specified in two dimensions by the locations of the two end points \((x_1, y_1)\) and \((x_2, y_2)\), and the two control points \((x_1^*, y_1^*)\) and \((x_2^*, y_2^*)\). The path is constructed by setting \((x_1, y_1)\) at the parafoil’s current location and \((x_2, y_2)\) at the final approach point (FAP). Next, control point \((x_1^*, y_1^*)\) is fixed along a line in the direction of the parafoil’s current heading angle at a distance \(d_1\) from point
Likewise, control point \((x_1, y_1)\). Likewise, control point \((x_2, y_2)\) is fixed along a line directly downwind of the FAP at a distance \(d_2\). This geometry is shown in Fig. 3. Note that fixing the first control point \((x_1', y_1')\) along the current path of the parafoil guarantees a smooth transition from the homing phase or upon recalculation of the current guidance curve. Fixing the second control point \((x_2', y_2')\) along a line passing through the curve’s end point and parallel to the wind guarantees entry into the final approach facing into the wind. For the single-curve solution, the distances \(d_1\) and \(d_2\) are the optimization variables. Note that the curve shape is highly flexible depending on the values of \(d_1\) and \(d_2\) and the initial and final headings of the curve, as previously shown in Fig. 2. Also note that, although not explored here, the final heading direction may be allowed to vary from the wind direction in practical implementations of the algorithm, ostensibly leading to a larger solution space.

Feasible Bézier curve terminal guidance paths are obtained by solving a constrained nonlinear minimization problem. The problem is formulated through application of several assumptions that allow the parafoil to be modeled kinematically, enabling suitable estimates of flight path quantities (time along the path, heading angle, etc.) to be determined quickly for a candidate path. First, it is assumed that the parafoil turns slowly enough so that roll and sideslip angles may be ignored. Second, it is assumed that the parafoil maintains a constant horizontal airspeed \(V_{ss}\). Finally, it is assumed that the parafoil’s descent rate is a function of yaw rate, so that

\[
\dot{z} = V_{z} - k_{yr}|\psi| + W_{z}
\]

where \(V_{z}\) is a constant steady-state descent rate, \(\dot{\psi}\) is the Euler yaw-angle rate, and \(W_{z}\) is the vertical wind velocity (in practical implementations, more complicated descent rate dependencies may be easily incorporated if required). For the parafoil used in these studies (described in the Appendix), \(V_{z} = 13.2\) ft/s, and \(k_{yr} = 1.2\) ft/s (or radians per second). Typically, the coupling parameter between yaw rate and descent rate \(k_{yr}\) must be identified for a specific parafoil through system identification. Thus, for path-planning purposes, parafoil motion can be adequately represented by the kinematic model given by

\[
\begin{align*}
\dot{x} &= W_x + V_{ss} \cos \psi \\
\dot{y} &= W_y + V_{ss} \sin \psi \\
\dot{z} &= V_{z} - k_{yr}|\psi|
\end{align*}
\]

where \(W_x\), \(W_y\), and \(W_z\) are wind components in the inertial frame \(I\).

Two types of constraints are enforced during Bézier curve optimization. The first is a constraint on the maximum possible parafoil turn rate. Turn rates along a Bézier curve path are estimated first by discretizing the path into \(N\) points \(\{b_1, b_2, \ldots, b_N\}\) equally spaced along the curve in terms of arc length. Ground speed between each point is then calculated as a composition of the horizontal airspeed \(V_{ss}\) and winds using a standard wind triangle as shown in Fig. 4. These ground-speed values are used to obtain the time it takes to travel from point to point between all the discretized locations on the Bézier curve, yielding \(N - 1\) time intervals \(\{\Delta t_1, \ldots, \Delta t_k, \ldots, \Delta t_{N-1}\}\). In addition, \(N - 1\) yaw angle values may be determined by computing heading angles between successive points on the Bézier path and computing the required crab angle to maintain that course for a given ground speed as shown in Fig. 4. This yields \(N - 1\) yaw-angle values given by \(\{\psi_1, \ldots, \psi_k, \ldots, \psi_{N-1}\}\). Finally, yaw rate is computed simply as a forward-looking, first-order finite difference according to

\[
\psi_k = \frac{\psi_{k+1} - \psi_k}{\Delta t_{k+1/2}}
\]
where \( \Delta t_{k+1/2} = (\Delta t_{k+1} + \Delta t_k)/2 \). Equation (8) yields \( N - 2 \) yaw rate values \( \{\dot{\psi}_1, \ldots, \dot{\psi}_k, \ldots, \dot{\psi}_{N-2}\} \) that may be used to determine the maximum absolute yaw rate on the Bézier curve path. Paths that yield yaw rate magnitudes above a given value \( \dot{\psi}_{\text{max}} \) are considered infeasible and rejected. For the parafoil used in these studies, \( \dot{\psi}_{\text{max}} = 0.76 \text{ rad/s} \). Once a feasible path is found, the computed yaw angles are sent to the model predictive inner-loop controller described in Sec. III for terminal tracking.

The second constraint imposed during path generation is a restriction that the Bézier curve path must not intersect an obstacle given the parafoil’s predicted altitude upon arrival at the obstacle. This is computed within the optimization routine by determining whether any of the discretized points on the Bézier curve infringe upon terrain features or obstacles (or similarly a fixed buffer zone around the obstacle). Note that although there is potential to “miss” an intersection between the Bézier path and an obstacle due to the path discretization, in practice the discretization is performed at a much finer resolution than the smallest obstacle size to ensure that intersections with all obstacles are captured.

Bézier curve optimization is performed with respect to a cost function defined as

\[
J(B) = z_{\text{miss}}
\]

Altitude error \( z_{\text{miss}} \) is measured with respect to the target after addition of a 60 ft straight path from the Bézier curve end point (final approach point) to the target. Computation of \( z_{\text{miss}} \) requires numerical integration of the descent rate given in Eq. (6) along the entire Bézier curve path plus the final approach addition, which clearly depends on the yaw rate values, initial altitude, and total time to traverse the path. Assuming that the parafoil traverses the candidate path, let \( z_N \) be the predicted parafoil altitude upon reaching the target. Then, \( z_{\text{miss}} \) is defined as

\[
z_{\text{miss}} = |z_N - z_t|
\]

For the single Bézier curve path, the terminal guidance optimal planning problem can therefore be stated as follows.

Given the current parafoil position \((x, y, z)\) and target location \((x_t, y_t, z_t)\), find control point distances \(d_1\) and \(d_2\) that minimize

\[
J(B) = z_{\text{miss}}
\]

subject to the following.

1) \( \max(|\dot{\psi}_1|, |\dot{\psi}_2|, \ldots, |\dot{\psi}_{N-1}|) < \dot{\psi}_{\text{max}} \).
2) The path \( B(\tau) \) does not intersect any terrain obstacles.

Note that, in general, this is a nonconvex optimization problem due to the nonconvexity of the constraint functions.

C. Multiple Curve Paths

The optimization routine using a single Bézier curve may not converge to a suitable trajectory solution for several reasons. In some cases, a single-curve path with the appropriate length that also satisfies the terrain and yaw rate constraints may not exist. This is especially evident in cases with numerous terrain obstructions, or when path planning is initiated from approach geometries that differ significantly from a standard downwind-to-upwind approach (i.e., the geometry shown in Fig. 1). In such cases, multiple connected cubic Bézier curves may be used instead, adding significant flexibility to the solution while also increasing computational burden. Figure 5 shows an example path using two Bézier curves. The two Bézier curve solution is constructed by setting the first end point of the first curve to the current parafoil location and the second end point of the second curve to the FAP. The two curves meet at a shared end point, termed the curve “midpoint”. The first control point for the first curve and the second control point for the second curve lie along lines determined by the parafoil’s heading and wind direction, respectively, as in the single Bézier curve case. The other two control points must lie along a line such that each control point and the midpoint are collinear to avoid yaw-angle discontinuities. These geometric constraints lead to five additional optimization parameters in addition to those in the single Bézier curve case, resulting in seven total optimization parameters. These parameters are \(d_1, d_2, d_3, d_4, x_{\text{mid}}, y_{\text{mid}}, \) and the angle \( \theta_{\text{mid}} \).

Increased geometric flexibility of the path may be achieved by adding additional curves, providing further benefit in cases of highly constrained drop zones or when energy management and homing are executed poorly and significant energy must be dissipated during terminal guidance. Figure 6 shows an example terminal path using three connected cubic Bézier curves. Note that, in this case, there are 12 variables that must be optimized. In general, a path consisting of \( N_C \) connected cubic Bézier curves requires optimization of \( 5(N_C - 1) + 2 \) independent variables, leading to rapidly increasing computational burden as \( N_C \) grows. In the next studies, a maximum of \( N_C = 3 \) was chosen as this was found to be sufficient in almost any obstacle scenario considered by the authors. Practical implementations may support structures with additional curves if sufficient onboard computational resources are available. Alternatively, precomputed curves may be inserted into the Bézier curve chain in...
high-altitude, highly constrained scenarios to avoid run time or convergence penalties associated with the $5(N_C - 1) + 2$ scaling of optimization parameters.

Two additional penalties are added to the cost function for the multiple Bézier curve optimization. The distances from curve midpoints to the target are penalized to avoid paths that wander far from the target area. Such paths are generally inadvisable due to uncertainty in wind conditions. In addition, dissimilarity in control point magnitudes at each midpoint is penalized to minimize yaw acceleration at each curve junction. Yaw acceleration is minimized through this penalty because, at each curve junction, the two control points and the midpoint are constrained to lie along a line. As a result, control points magnitudes dictate curvature of the Bézier path on either side of the junction, and thus similar control point magnitudes will yield similar curvatures and associated yaw rates (thus yaw acceleration will be minimized). Resulting curves that minimize yaw acceleration are likely to offer more favorable tracking performance. The resulting cost function is therefore

$$J(B) = z_{\text{miss}} + \sum_{i=1}^{N_C-1} k_m d_{\text{mid}_i} + \sum_{i=1}^{N_C-1} k_{\text{cp}} |d_{2i+1} - d_{2i}|$$  \hspace{1cm} (11)$$

where $d_{\text{mid}_i}$ is the distance between the $i$th curve midpoint and the target, and $d_j$ is the $j$th control point magnitude. Note that these two additional penalties are meant strictly to steer the optimizer toward paths with improved robustness to wind disturbances and lower yaw rate demands but have not been observed to have detrimental effects on optimizer convergence. Thus, $k_m > 0$ and $k_{\text{cp}} > 0$ are held to relatively small values to ensure that the optimizer prioritizes miss distance well above any of the other factors. In the cases shown next, $k_m = 1.5 \times 10^{-6}$ and $k_{\text{cp}} = 1.0 \times 10^{-6}$ were chosen as suitable values through limited experimentation.

D. Path Planner Structure and Execution

A closed-loop feedback path-planning system is constructed for parafoil terminal guidance based on the Bézier curve optimization process described previously. Figure 7 shows a block diagram of the overall path planner flow of execution. At initiation of terminal guidance, a terminal path is planned through solution of a set of tiered optimization problems, as shown in Fig. 8. When the path planner requests a new path solution, curve optimization is performed first using a single curve. The optimization process is considered successful if the nonlinear optimizer converges to a specified relative cost function tolerance of $f_{\text{tol}}$, and if the estimated altitude miss $z_{\text{miss}}$ is below a specified value $z_r$. If the two criteria given by $f_{\text{tol}}$ and $z_r$ are met, then the path is considered valid and sent to the tracking controller. If either of the two criteria is violated, then $N_C$ is incremented by 1 and the optimization problem is attempted again. If a valid path cannot be found for $N_C = 3$, then path optimization is considered to have failed, and the path planner either initiates a so-called fallback position or reverts to the previous valid Bézier curve solution if one has been found. Note that, because of the nonconvexity of the nonlinear optimization problem posed in Eq. (11), strict convergence or run-time guarantees cannot be provided in general, and thus fallback plans are necessary to ensure robustness during real-time guidance.

![Fig. 7 Bézier curve path planner flow of execution.](image-url)
Two fallback positions are defined in cases when the path planner is unsuccessful in solving for a valid Bézier curve path at initiation of terminal guidance (i.e., the $f_{tol}$ or $z_r$ requirements are violated during the initial attempt to solve for a terminal guidance path or an optimization time threshold is exceeded). Define the altitude margin $z_{max}$ as the amount of excess altitude available with respect to the target altitude if the parafoil executes a constant rate turn and proceeds directly to the target in a straight line trajectory. Thus, altitude margin is defined as

$$z_{max} = \min \left( \sqrt{\frac{(x-x_d)^2 + (y-y_d)^2}{(W_x + V_h \cos \psi)^2 + (W_y + V_h \sin \psi)^2}} - z_{turn} + z - z_r \right)$$

where $x$, $y$, and $z$ are the current parafoil inertial coordinates; $x_d$, $y_d$, and $z_d$ are the target coordinates; $z_{turn}$ is the altitude loss incurred during the constant rate turn computed from Eq. (6); and $\psi$ is the final parafoil yaw angle once the turn is complete. If a valid path has not been found and $z_{max} < 0$, then a Dubins path is computed as the optimal route to the target that minimizes total miss distance (as described in [10]), and the planning process is terminated. If $z_{max} > 0$ and no valid path has been found, then the parafoil is commanded to enter a circular pattern of radius $r_c$ centered at the target, during which new Bézier curve optimizations are regularly attempted. Once a valid Bézier curve solution is found, the planner exits the circular path and executes the computed Bézier path. Figure 7 shows a detailed flowchart of the planning process and the described fallback procedures. Note that, in nearly all of the simulation studies shown next, once the circular pattern fallback procedure is initiated, a valid Bézier curve solution is usually found within a few replanning cycles.

Figure 8 shows the Bézier curve optimizer execution path, which takes place every time a new path is requested. Simpler curves with $N_C = 1$ are attempted using a succession of initial guesses discussed in the next section, followed by more complex curves with $N_C = 2$ and $N_C = 3$. The optimizer exits once the nonlinear optimization routine converges to a relative cost function tolerance of $f_{tol}$, as long as the path exhibits an estimated miss distance less than $z_r$. If the optimization routine cannot satisfy these criteria even with $N_C = 3$, then curve computation is considered to have failed.

Wind uncertainty, sensor errors, and the kinematic model assumptions used in the path planner inevitably lead to tracking error from the planned trajectory as the parafoil tracks desired turn rates. Thus a path that was optimal upon arrival at the TIP may not be optimal later on during terminal guidance. As shown in [10–12], a feedback path-planning system is required to replan in cases where tracking error exceeds a desired threshold and it is no longer feasible to track the original path.

Replanning is initiated upon exceeding a threshold of total error, which is calculated as the total three-dimensional distance $d_{err}$ between the parafoil’s current position and its predicted location along the terminal flight path at the current time:

$$d_{err}(t) = \sqrt{(x-x_{des})^2 + (y-y_{des})^2 + (z-z_{des})^2}$$

(13)

where $x_{des}$, $y_{des}$, $z_{des}$ denote the desired position at time $t$. The terminal path is replanned if $d_{err}$ exceeds a threshold distance $d_{max}$, where

$$d_{max} = \frac{z_0}{z_0} d_{tol} + d_0$$

(14)

In Eq. (14), $z_0$ is the initial altitude of the parafoil at initiation of terminal guidance, and $d_{tol}$ and $d_0$ are constants. Note that maximum acceptable tracking error $d_{max}$ decreases linearly with altitude, so that less tracking error is accepted as the parafoil nears the ground. This reflects the increased urgency associated with replanning at lower altitudes where tracking error has less time to be corrected. In the simulations shown next, $d_{tol} = 150$ ft and $d_0 = 5$ ft. Also, note that, in implementations of the path-planning algorithm shown next, replanning is initiated not from the parafoil’s current location but rather from the predicted location in 1 s to account for latency in the path-planning process. As shown in the examples next, the majority of paths are planned in 1–2 s using the authors’ implementation, and thus this was determined to be a reasonable value.

E. High Wind Conditions

A fundamental assumption in the path planner formulation described in Sec. II.B is that the parafoil can physically track a candidate path considered by the optimization routine. This assumption is necessary to compute the time required for the parafoil to traverse the candidate path, providing an estimate of the parafoil’s altitude once it reaches the target. However, if path planning is initiated with the estimated wind greater than the parafoil airspeed $V_{as}$, any path segments that require the parafoil to advance upwind or maintain crab angles in excess of 90 deg are unfeasible.
In these situations, there is no suitable way to evaluate candidate paths quickly using a kinematic model of the parafoil. Although it may be technically possible to evaluate such candidate paths using a higher-fidelity simulation model (for instance, the 6DOF model employed in the model predictive guidance scheme in [12]), such a step would greatly increase computational burden. Instead, in the path-planning system implemented here, if the wind speed exceeds the parafoil airspeed, then the wind magnitude used by the planner is set to 95% of the parafoil airspeed and the FAP is moved upward of the target. This essentially restricts the optimization algorithm from considering any paths with segments that advance upward. If wind speed is greater than $V_{ws}$, then the parafoil will not be able to track the planned path exactly (because the winds used to plan the path are different from the actual winds), and thus tracking error develops. However, the feedback nature of the path planner mitigates the effects of tracking error on overall miss distance because new paths are continually replanned. Therefore, although an “optimal” path cannot be generated in winds greater than $V_{ws}$ using the kinematic model assumption, in practice the system yields reasonable results as demonstrated in Sec. IV (as long as terminal guidance is initiated upward of the target).

F. Initial Guess Selection

For a single curve, two regions of the parameter space (consisting of $d_1$ and $d_2$) may be identified that produce paths consistent with the parafoil maximum yaw rate constraint. Consider Fig. 9, in which $d_1$ is varied and $d_2$ is constrained such that $d_1 = d_2$. The left-hand and right-hand figures show paths consistent with the parafoil maximum yaw rate, while the path depicted in the center figure is likely to exceed the yaw rate constraint. Therefore, two feasible regions of the parameter space exist as shown in Fig. 10. Initial guesses for the single-curve case are constructed so as to place the optimizer somewhere in one of these favorable regions. The first initial guess is chosen as $d_1 = d_2 = 0.4r_0$, where $r_0$ is the parafoil’s current three-dimensional distance to the target, while the second initial guess is chosen as $d_1 = d_2 = 3r_0$.

For $N_C > 1$, initial guess selection is more complex because identification of favorable convergence regions in higher-dimensional spaces is significantly more difficult, especially given terrain constraints that vary on a case-by-case basis. For that reason, general initial guesses have been derived through experimentation with example terrain profiles that promote convergence to a valid solution. For $N_C = 2$, an initial guess is formulated by placing $x_{mid}$ and $y_{mid}$ equidistant between the parafoil and the target, and $\theta_{mid}$ pointing directly toward the target such that the middle two control points, the midpoint, and the target are all collinear. Two subsequent initial guesses are constructed by shifting $x_{mid}$ and $y_{mid}$ closer and farther from the target, respectively. For $N_C = 3$, an initial guess is formulated again by aligning both curve midpoints ($x_{mid}$, $y_{mid}$) and ($x_{mid}$, $y_{mid}$) as well as control points $d_2$, $d_3$, $d_4$ along a line connecting the parafoil location and the target such that curve midpoints are equally spaced. An equivalent initial guess selection procedure may be used for $N_C > 3$. The general scheme described previously has proven suitable for the rather arbitrary and differing terrain examples studied here and does not necessarily represent an optimized initial guess selection procedure. More complex methods for deriving initial guesses using a given terrain profile may perhaps be derived but are beyond the scope of this work.

Finally, in cases of curve replanning, the likelihood of convergence may be improved by using the remaining segment of the previous Bézier curve solution as an initial guess. De Casteljau’s algorithm [23] enables a Bézier curve to be subdivided into two segments; thus, a curve parameterization (curve end points and control points) may be easily derived for the remaining segment of the previous Bézier path. Let $b_0^1$, $b_1^1$, $b_2^1$, and $b_3^1$ be the four points describing the path being currently tracked by the parafoil (in the multiple-curve case, this represents the current single-curve segment of the multiple-curve path). Furthermore, let $\tau_c \in [0, 1]$ be the Bézier parameter at the parafoil’s current location along the path. Then define the sequence of points given by [25]
r_B does not yield a solution that satisfies the as shown in Fig. 11. At each curve replanning cycle, the previous curve initial guess is used first, followed by the progression shown in Fig. 8 if this does not yield a solution that satisfies the $f_{\text{tol}}$ and $z_c$ criteria. Use of the previous curve solution as the first initial guess was observed to greatly improve convergence characteristics.

G. Algorithm Implementation

The path-planning algorithm described previously was implemented in C++ using the open-source nonlinear optimization library NLopt [24]. After some experimentation, a derivative-free optimization algorithm, constrained optimization by linear approximation, was chosen due to the absence of analytical expressions for the constraint and cost function gradients [25]. A cost function relative tolerance of $f_{\text{tol}} = 1 \times 10^{-2}$ was chosen as the primary convergence criterion, and a maximum run time of 1.5 s was set for each optimization attempt (i.e., each path optimization converges to altitude miss distances much less than 1 ft due to the low time constant, and $f_{\text{tol}}$ was set to $z_c = 20$ ft. In the vast majority of cases, the optimizer converges to altitude miss distances much less than 1 ft due to the low $f_{\text{tol}}$ value except when altitude margin is less than zero or in highly constrained terrain conditions. However, this strict convergence criteria did not impose a noticeable run time penalty. Path-planning computations are performed on a standard desktop computer using an Intel Xeon processor running at 3.2 GHz.

III. Parafoil Model and Inner-Loop Control

A simulation architecture was constructed to evaluate performance of the proposed Bézier path planner. In the path-planning formulation, a kinematic model is used, and it is assumed that the parafoil tracks the computed path perfectly. In reality, the parafoil experiences turning dynamics and nonzero sideslip, and thus performance must be determined using a higher-fidelity model. A six-degree-of-freedom (6DOF) parafoil simulation, similar to that used in [11–13], is employed here to evaluate tracking performance. Degrees of freedom include the parafoil-payload system mass center inertial coordinates and the vehicle yaw, pitch, and roll Euler angles ($\phi$, $\theta$, and $\psi$ respectively). The model assumes a constant canopy incidence angle and includes drag of the payload in the system aerodynamic coefficients. Apparent mass is approximated by assuming the parafoil has a plane of symmetry. Details of this model can be found in [26].

An inner-loop control strategy is required to generate asymmetric brake deflections such that the parafoil remains on the desired trajectory output from the path planner. The Bézier curve path planner produces a desired $x$-$y$ time history from the current point to impact, which, combined with the kinematic model discussed in Sec. II, yields a yaw-angle time history. This yaw-angle time history may be interpolated at equal time intervals along the path. Any number of inner-loop controllers may be used to track this desired trajectory, but by sampling the desired yaw rate time history over a finite horizon, model predictive control (MPC) may be implemented relatively easily. Recently, Ward [27] showed that a single-degree-of-freedom linear turn model may be used quite effectively for parafoil inner-loop tracking, which has proved suitable in other recent studies [12]. This discrete linear turn rate model is given as

$$\begin{bmatrix} \bar{\psi} \\ \bar{\tau}_B \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \Delta \\ 0 & 1 - \Delta/\tau_T \end{bmatrix} \begin{bmatrix} \bar{\psi} \\ \bar{\tau}_B \end{bmatrix}_k + \begin{bmatrix} 0 \\ B\Delta/\tau_T \end{bmatrix} \delta_{\psi,k} \tag{16}$$

where $\bar{\psi}$ is the parafoil yaw rate, the state vector is $x = [\bar{\psi} \, \bar{\tau}_B]^T$, $\Delta$ is the discrete sampling period, $B$ is the control sensitivity, $\tau_T$ is the turning time constant, and $\delta_{\psi}$ is the parafoil asymmetric brake deflection [27]. Other models that incorporate Dutch roll, sideslip, and other dynamic effects may be used in practice to improve tracking performance as needed.

Let the vector $\bar{\psi} = [\psi_{k+1}, \psi_{k+2}, \ldots, \psi_{k+H_p}]^T$ represent discrete desired yaw angles over a prediction horizon $H_p$. Model predictive control solves for the optimal control sequence $U = [u_{k}, u_{k+1}, \ldots, u_{k+H_p-1}]^T$ governed by the linear plant $x_{k+1} = Ax_k + Bu_k$ with output $y_k = Cx_k$. For the single-degree-of-freedom turn model used in this MPC scheme, $A$ and $B$ are the matrices in Eq. (16), and the output matrix is $C = [0 \, 1]^T$. Given the current state vector $x_k$ and control vector $U$, a future estimate of the state may be obtained according to

$$\hat{\psi} = K_{CA}x_k + K_{CAB}U \tag{17}$$

where

$$b_i' = (1 - \tau_j)b_i^{j-1} + \tau_j b_{i+1}^{j-1} \quad j = 1, 2, 3; \quad i = 0, \ldots, 3 - j \tag{15}$$

The new curve end points are given by $b_i^0$ (the parafoil’s current location) and $b_i^1$, while the new curve control points are given by $b_i^2$ and $b_i^3$ as shown in Fig. 11. At each curve replanning cycle, the previous curve initial guess is used first, followed by the progression shown in Fig. 8 if this does not yield a solution that satisfies the $f_{\text{tol}}$ and $z_c$ criteria. Use of the previous curve solution as the first initial guess was observed to greatly improve convergence characteristics.

Fig. 11 Bézier curve segmentation using De Casteljau’s algorithm. Path to left of current position denotes original curve; path to right of current position denotes new curve used for replanning initial guess.
Model predictive control solves for $U$ by casting the problem as a finite-time discrete optimal control problem with quadratic cost given by

$$J = (\psi - \hat{\psi})^T Q (\psi - \hat{\psi}) + U^T R U$$

where $Q \in H_p \times H_p$ is a symmetric positive semidefinite matrix, and $R \in H_p \times H_p$ is a symmetric positive definite matrix, which penalize tracking error and control, respectively. The solution to this optimization problem is found analytically to be [28]

$$U = (K_{CA}^T Q K_{CA} + R)^{-1} K_{CA}^T Q (\psi - K_C X_k)$$

Only the first control value $u_k$ is applied at each update as the optimal control problem is solved at each MPC time step. This MPC scheme is used to compute asymmetric brake deflections given a desired Bézier curve path over a prediction horizon $H_p = 10$ s.

The ideal terminal guidance scenario shown in Fig. 1 assumes that the parafoil can transition to the final turn from zero turn rate instantaneously. This is a direct result of the kinematic model assumptions employed in Sec. II. In reality, turning dynamics have a nonzero time constant $\tau_T$. To compensate for turning dynamics, the transition from homing to tracking the final maneuvers is advanced by $t_{pre}$ seconds, where $t_{pre}$ is a “pretum” parameter that may be selected based on the turning time constant $\tau_T$ of the particular parafoil system under consideration. This method was employed successfully in previous studies [12,13].

### IV. Results

Performance of the proposed path-planning algorithm is demonstrated through detailed example simulations and Monte Carlo studies. The parafoil model used to generate these results is a small-scale vehicle with a steady-state horizontal airspeed of approximately 24.8 ft/s and descent rate of approximately 13.5 ft/s in level flight. Additional parafoil parameters are provided in the Appendix. For all simulations, it is assumed that energy management and homing are performed imperfectly (due to wind shifts or other factors), and thus initiation of terminal guidance may not be performed in an ideal geometry. The target is assumed to lie at the origin, and the final approach point is placed 60 ft downwind of the target.

Table 1 lists all path planner parameters used throughout this section, and the Appendix lists the model predictive control weighting matrix values. Note that performance of the path planner proved to be quite insensitive to many of the parameters in Table 1 (such as $k_a$, $k_c$, which are designed simply to add minor gradient to the cost function). Unless otherwise noted, the parafoil is perturbed by random wind gusts throughout terminal descent, where gusts are generated based on the Dryden gust model outlined in [29]. Note that vertical winds are not included in these studies, but if estimates are available, they may be easily incorporated in the path planner through the $W_z$ terms in Eq. (6). To obtain a wind estimate for use by the path planner, actual winds are filtered through a 4 s moving average using a Gaussian additive noise term with a standard deviation of 10% of the actual wind value (modeled after the simulated wind estimation scheme used in [12,13]). Furthermore, to improve simulation fidelity, actual digital terrain and elevation data (DTED) are used to demonstrate performance of the path-planning system in realistic terrain environments. Section IV.A details several example simulations, while Sec. IV.B demonstrates overall system performance through Monte Carlo simulation.

#### A. Example Simulations

Example simulations demonstrate performance of the algorithm in some notional landing scenarios in which the path planner must avoid realistic terminal constraints and adapt to wind disturbances. The first landing scenario is designed to demonstrate nominal performance of the path planner in a benign geometry. In this case, the parafoil begins terminal guidance heading downwind, where the wind blows to the north at 10 ft/s. To demonstrate ideal performance, this is the only example in this section where randomized wind gusts are not included and perfect knowledge of the wind is assumed. The parafoil’s position at handoff to terminal guidance is $-350$ ft cross range, $-100$ ft down range, and 500 ft altitude. Figure 12 shows the planned path and the flown path. Note that a single Bézier curve path is sufficient to create a feasible terminal guidance solution, and the final impact point approximately overlays the target. For this single-curve case, computation of the Bézier curve solution required a run time of 0.075 s.

A second scenario is considered using actual DTED data from the vicinity of Stone Mountain, Georgia. Nominal winds are from the Northeast at approximately 14.5 ft/s. Randomized wind gusts generated in accordance with a Dryden gust model are present throughout the trajectory, varying both wind magnitude and direction. Magnitude deviations between 0 and 90% of the parafoil airspeed are allowed. Note that this landing scenario creates a highly constrained environment for the path planner, especially if the vehicle is expected to land into the wind. Figure 13 shows a three-dimensional plot of the drop zone and the path flown by the parafoil. Figure 15 shows a time history of wind magnitude. Note that, in these simulations, as in all simulations following, a

### Table 1 Path-planner parameters used in simulation studies

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_a$</td>
<td>Midpoint distance penalty weighting factor</td>
<td>$1.5 \times 10^{-6}$</td>
</tr>
<tr>
<td>$k_c$</td>
<td>Control point dissimilarity weighting factor</td>
<td>$1.0 \times 10^{-6}$</td>
</tr>
<tr>
<td>$z_c$</td>
<td>Altitude miss distance acceptance threshold</td>
<td>20 ft</td>
</tr>
<tr>
<td>$f_{tol}$</td>
<td>Optimizer convergence relative tolerance</td>
<td>$10^{-2}$</td>
</tr>
<tr>
<td>$d_{tol}$</td>
<td>Replanning error threshold, linear term</td>
<td>150 ft</td>
</tr>
<tr>
<td>$d_0$</td>
<td>Replanning error threshold, constant term</td>
<td>5 ft</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of points used to discretize each Bézier Curve</td>
<td>500</td>
</tr>
<tr>
<td>$r_f$</td>
<td>Radius for circular path fallback plan</td>
<td>150 ft</td>
</tr>
<tr>
<td>$H_p$</td>
<td>Model predictive control receding horizon</td>
<td>2.5 s</td>
</tr>
<tr>
<td>$t_{pre}$</td>
<td>Model predictive control pretum</td>
<td>1.4 s</td>
</tr>
</tbody>
</table>
moving average filter including noise is used within the simulation to generate imperfect wind estimates used by the parafoil guidance system. Thus deviations from the planned path are caused by kinematic model error, controller tracking error, wind gusts, and wind estimation errors. This example yields an overall miss distance of 34 ft, which is clearly higher than the ideal case shown in Fig. 12, which did not include gust disturbances. Seven total paths are planned between the initial path and replanning cycles, with run times varying between a minimum of 0.03 s for one of the single-curve paths at the end of the trajectory and a maximum of 4.4 s for one of the two-curve paths near the beginning of terminal guidance. As evident in the first replanning cycle in Fig. 13, if different path shapes offer similar performance to the optimizer, then small changes in wind estimates may initiate a switch between these path shapes.

Another example is presented in a further constrained terrain environment. In this case, DTED data of a steep canyon is used and the target located at the bottom of the canyon. Wind is along the canyon from the west at a nominal magnitude of 6 ft/s, but gusts as low as 2 ft/s and as high as 12 ft/s affect the parafoil throughout flight. Again, the simulated wind estimator is used to provide the planner with imperfect knowledge of the wind. Figure 16 shows the vehicle trajectory and desired paths, while Fig. 17 shows a three-dimensional plot of the path flown. Miss distance in this example is 38 ft. Five total Bézier curve paths are planned, with a minimum solve time of 0.06 s for one of the single-curve paths and a maximum of 2.9 s for one of the two-curve paths. Note the s-turn shaped Bézier path planned near the end of the trajectory to expend energy as the wind magnitude suddenly decreases due to a gust.
B. Monte Carlo Simulations

Several Monte Carlo simulations in various terrain environments are performed to evaluate the planner’s effectiveness under randomized wind and initial conditions. All simulations use 200 trajectories, where initial locations are randomized in an $800 \times 800$ ft$^2$ area around the target such that the parafoil begins with $z_{mar} > 0$. In the first simulation, flat terrain is used, and the parafoil starts at 500 ft altitude. Initial heading angles point directly toward the target, with a deviation of $\pm 90$ deg, according to a uniform distribution. Wind direction is randomized between zero and $2\pi$, and wind magnitude is uniformly distributed between 0 and 90% of the parafoil airspeed (22.3 ft/s). Furthermore, randomized wind gusts perturb the parafoil throughout the trajectories. Gusts are generated according to a Dryden model, with a standard deviation of 4 ft/s in magnitude and a standard deviation of 30 deg in direction from the nominal wind (thus, there are periods during some trajectories where winds are greater than parafoil airspeed). The simulated wind estimator used in the example simulation sections is also included here, as in all Monte Carlo examples next. Figure 18 shows Monte Carlo impact points for the flat terrain case. Target-centered 50% circular error probable (CEP), representing the median horizontal miss distance from the target, is 31 ft for this case, while 90% CEP is 117 ft. Figure 19 shows a histogram of path planner execution times for each initial planning and replanning cycle for this flat terrain case, while Table 2 describes run-time and path-planning statistics. Note the interesting bimodal distribution in Fig. 19, indicating that many single-curve paths can be planned extremely quickly, but more complex geometries (consisting of more curves) take a bit longer. Mean and median solution times for the Bézier curve path planner are approximately 1.32 and 1.50 s, respectively, demonstrating that the closed-loop system achieves attractive accuracy at reasonable computational burden.

Planner effectiveness is more clearly demonstrated when considering constrained drop zones. A Monte Carlo simulation of 200 trajectories is performed using the Stone Mountain terrain. As in the previous example case, the target is located at the bottom of the mountain on the downwind
side, thus constraining the final approach path if landing into the wind is desired. Nominal winds blow from the north with a magnitude uniformly distributed between 0 and 22 ft/s, and direction normally distributed with a standard deviation of 30 deg. A gust model identical to that in the first Monte Carlo simulation is employed. Figure 20 shows Monte Carlo impact points overlaying a contour map of the terrain. A 50% CEP of approximately 38 ft and 90% CEP of 184 ft are observed in this case, indicating that accuracy is only marginally affected by the presence of an obstacle in the final approach path. Table 2 lists run time statistics for this case, showing mean and median solve times of 1.34 and 1.50 s, respectively. Run-time histograms for this case are similar to that shown in Fig. 19 and are not included here.

One of the primary motivations behind the proposed planner is to decouple landing accuracy from the presence of terrain constraints as much as possible through geometric path flexibility. This is further demonstrated in the final two Monte Carlo simulations presented here, involving the canyon terrain used previously. The target is again embedded at the bottom of the 160-ft-wide canyon. In the first Monte Carlo simulation in the canyon terrain, wind magnitudes are again uniformly distributed between 0 and 22 ft/s, and nominal winds blow from the west with a directional standard deviation of 30 deg. Random gusts are present, with a standard deviation of 4 ft/s in magnitude and 30 deg in direction from the nominal wind. As in all Monte Carlo simulations, the initial location is randomized in an 800 x 800 ft grid around the target, and initial heading is randomized as well, as described previously. Thus, in the majority of cases, significant path flexibility is required during terminal guidance to achieve successful entry into the canyon near the target. Figure 21 shows the impact dispersion pattern with a 50% CEP of 44 ft (90% CEP of 256 ft), and Table 2 lists the run-time statistics. Again, run-time histograms looked similar to that shown in Fig. 16 and have been omitted for space reasons. Note that, even with the highly constrained approach environment, the CEP increases only 13 ft over the flat terrain case and over 85% of the impacts land on the floor of the canyon in the target vicinity. Geometrical path flexibility is critical in this case to ensure that an appropriate descent can be planned without flying too far from the target.
A final Monte Carlo simulation examines the case of high winds using the example canyon geometry. All simulation parameters remain the same, except nominal winds are uniformly distributed between 22 and 37 ft/s (90 to 150% of vehicle airspeed). No gusts are included in this example. Although all initial locations are upwind of the target, this by no means guarantees a flyable entry path to the canyon given limitations on advancing flight upwind. Figure 22 shows the impact point dispersion pattern for this case, while Table 2 lists the run time statistics. Clearly, the majority of impacts land downwind from the target, as expected, but the 50% CEP value of 289 ft represents attractive performance in such a difficult landing scenario, especially considering initial conditions are located in an 800 × 400 ft area upwind of the target. Note that the 90% CEP value for this case increases to 517 ft. In many cases, it was observed that, although the parafoil was unable to track the desired path for the reasons described in Sec. II.E, feedback path planning generally steered the parafoil into an approach scenario where it was directly upwind of the target and facing into the wind, as would be desired in landing conditions where wind-to-airspeed ratios are greater than one. To further this point, it was observed that, in 80% of cases, the parafoil landed facing within 5 deg of the upwind direction.

The data in Table 2 show that the majority of solutions use either one or two curves, with only a handful requiring three curves for a valid solution. Furthermore, run-time values in Fig. 19 demonstrate from an empirical standpoint that real-time performance of the proposed guidance system may be achieved. Planner fallback positions coupled with the optimizer maximum run time limits described in Sec. II allow convergence issues to be handled in a robust manner. Overall, results demonstrate that geometric path flexibility enables suitable terminal guidance paths to be generated even in highly constrained environments, potentially reducing coupling between terrain constraints and landing accuracy.

### V. Conclusions

A novel terminal guidance path-planning scheme has been proposed for autonomous parafoils. The algorithm parameterizes the flight path using single or connected cubic Bézier curves, which offer significant geometrical flexibility during terminal guidance maneuvers. A path-optimization problem was formulated, given constraints on parafoil maximum turn rate and three-dimensional terrain avoidance. Results demonstrated that the feedback algorithm is highly flexible and can handle difficult drop-zone terrain environments. Monte Carlo simulations verified that the algorithm produces reasonable flight paths that lead to low miss distances in unfavorable geometries and gusty wind conditions. Overall, the Bézier curve path-planning algorithm was shown to be an attractive path-planning scheme by offering geometrically flexible trajectory shapes that maintain landing accuracy in the presence of terrain obstacles.
Appendix: Example Parafoil and Controller Parameters

Table A1  Example parafoil inertial and aerodynamic parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>System mass ( m_B )</td>
<td>0.145 slug</td>
</tr>
<tr>
<td>Included mass ( m_I )</td>
<td>0.006 slug</td>
</tr>
<tr>
<td>Steady-state aerodynamic velocity ( V_{as} )</td>
<td>24.8 ft/s</td>
</tr>
<tr>
<td>Steady-state descent rate ( V_c )</td>
<td>13.5 ft/s</td>
</tr>
<tr>
<td>Canopy reference area ( S )</td>
<td>10.0 ft²</td>
</tr>
<tr>
<td>Canopy span ( b )</td>
<td>4.55 ft</td>
</tr>
<tr>
<td>Canopy chord ( c )</td>
<td>2.25 ft</td>
</tr>
<tr>
<td>Incidence angle ( \Gamma )</td>
<td>-12 deg</td>
</tr>
<tr>
<td>Payload inertia matrix elements, slug-ft² ( I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz} )</td>
<td></td>
</tr>
<tr>
<td>Canopy inertia matrix elements, slug-ft² ( I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz} )</td>
<td></td>
</tr>
<tr>
<td>Elements of the apparent mass matrix, slug</td>
<td>( P = 0.040, Q = 0.010, R = 0.0018 )</td>
</tr>
<tr>
<td>Distance from ( C ) to payload mass center ( z_{CS} )</td>
<td>0.3 ft</td>
</tr>
<tr>
<td>Distance from ( C ) to parafoil mass center ( x_{CB} )</td>
<td>0.5 ft</td>
</tr>
<tr>
<td>Distance from ( C ) to canopy rotation ( z_{CR} )</td>
<td>-2.25 ft</td>
</tr>
<tr>
<td>Distance from ( C ) to canopy rotation ( z_{CR} )</td>
<td>-0.5 ft</td>
</tr>
<tr>
<td>Distance from ( R ) to canopy aero center ( x_{RP} )</td>
<td>0.633 ft</td>
</tr>
<tr>
<td>Distance from ( R ) to apparent mass center ( x_{RM} )</td>
<td>0.59 ft</td>
</tr>
<tr>
<td>Distance from ( R ) to apparent mass center ( x_{RM} )</td>
<td>0.20 ft</td>
</tr>
<tr>
<td>Maximum brake deflection ( \delta )</td>
<td>0.75 ft</td>
</tr>
</tbody>
</table>

Aerodynamic coefficients

\[
\begin{align*}
C_{D0} &= 0.15, \\
C_{D2} &= 0.90, \\
C_{T0} &= -0.15, \\
C_{T2} &= 0.25, \\
C_{cd} &= 0.68, \\
C_{uw} &= 0.0, \\
C_{mp} &= -0.265, \\
C_{lw} &= -0.355, \\
C_{sb} &= -0.0003, \\
C_{ur} &= -0.02, \\
C_{au} &= 0.004, \\
C_{DS} &= 0.40
\end{align*}
\]

The example parafoil inertial and aerodynamic parameters are given in Table A1. The model predictive control weighting matrices are

\[
Q = \text{diag}(1.5, 1.5, 1.0, 0.8, 0.6, 0.3, 0.3, 0.15, 0.15, 0.15)
\]

\[
R = \text{diag}(0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.0)
\]

References


doi:10.2514/1.12251


doi:10.2514/2.6933


doi:10.2514/1.44802


doi:10.1243/09544101AJERG749


doi:10.2514/1.59782


doi:10.2514/1.28586


