Minimum Waypoint Spacing for VTOL Rotorcraft Trajectory Guidance using B-Spline Midpoint Refinement

Bryan C. H. Chu
PhD Candidate, GN&C
Arizona State University
Tempe, AZ, USA

Dr. Spring Berman
Associate Professor
Arizona State University
Tempe, AZ, USA

ABSTRACT

Given a cubic piecewise-polynomial spline representation of a Vertical Take-off and Landing (VTOL) rotorcraft guidance trajectory in the monomial basis, we propose a new method for defining the stopping criterion for midpoint refinement of its representation in the B-spline basis. This method incorporates a model of the rotorcraft flight dynamics, consisting of the wind-axes equations of motion for a point-mass particle under the assumptions of constant altitude, constant rotor speed, and negligible side force in the direction mutually perpendicular to the velocity vector and the thrust vector from the main rotor(s). Under such conditions, the rotorcraft's ability to maneuver and change the ground track angle of the velocity vector is limited by its current speed and the available command accelerations along and normal to the velocity vector. The command acceleration limits are a function of available installed engine power, internal acceleration limiters in the vehicle control laws, and maximum allowable body-axis rates and Euler attitudes. Moreover, these acceleration limits constrain the rate of reorientation of the velocity vector. Our method defines the spline control points as the waypoints of the flight plan and iteratively refines these points until their spacing is too close for the vehicle to reorient its velocity vector to align with the bearing to the next waypoint. This stopping criterion permits a dense spacing of waypoints, since the vehicle is not required to fly through successive waypoints or along the legs connecting them. Five motion primitives were used to demonstrate the algorithm. The motion primitives were scaled in size and speed to match ADS-33/MIL-DTL-32742 Mission Task Elements (MTEs) such as the Slalom and Pirouette, as well as future proposed MTEs for VTOL uncrewed autonomous systems (UAS). We evaluated the tracking performance of a simulated single main-rotor tail-rotor helicopter for flight plans based on two of the MTEs, and we examined how closely the wind-axes flight dynamics model matches these high-fidelity simulations for turn maneuvers in different speed regimes. These investigations indicate directions for further development of our method in simulated scenarios and eventual flight tests. The benefit of this method is that it requires only minimal software changes to extend the capability of in-service autopilot and coupled flight director algorithms to fly generic trajectories that can be represented by cubic piecewise-polynomial splines.

NOTATION

Δ = Waypoint (WP) change distance [ft]
V = Wind-axes velocity magnitude [fps]
χ = Wind-axes velocity ground track angle [deg]
n_{xw}, n_{yw}, n_{zw} = Resultant aerodynamic forces along, across, and normal to flight path, normalized by weight [g's]
g = Gravitational acceleration constant [ft/s^2]
$f(x)$ = Functional representation of cubic spline
θ_b = Desired bearing of next WP from previous WP
θ_t = Bearing to point of velocity tangency from previous WP
r_t = Radial distance to point of velocity tangency [ft]

Presented at the Vertical Flight Society's 11th Biennial Autonomous VTOL Technical Meeting, Phoenix, AZ, February 4-6, 2025. Copyright © 2025 by the Vertical Flight Society. All rights reserved.

INTRODUCTION

Current flight guidance algorithms for in-service, passenger-carrying VTOL rotorcraft are designed to guide the vehicle along straight-line paths between pairs of sparsely spaced waypoints. While these algorithms are highly effective and robust to disturbances (e.g. wind) during flight, there has long been a need for algorithms that can guide rotorcraft along more complex, curvilinear trajectories. This need is due to the proliferation of advanced automatic trajectory planners that can be used to route autonomous aircraft through densely populated urban areas and re-route them around unexpected obstacles. These trajectory planners typically output trajectories that can be represented by cubic piecewise-polynomial splines, and in some cases require higher-order polynomials.

Most published path-following and trajectory-tracking algorithms for aircraft are designed for fixed-wing bank-to-turn maneuvering and typically require extensive changes to in-service autopilot and guidance software architecture and code.

(Ref. 1) and (Ref. 2) provide an overview of various guidance strategies for fixed-wing unmanned aerial vehicles (UAVs) and quadrotors. (Ref. 3) and (Ref. 4) discuss two of the most widely-used "modern" (as they were developed many years after waypoint-following algorithms for straight-line paths) guidance algorithms, which to the authors' knowledge have yet to be certified on a production VTOL vehicle. These guidance algorithms are vector-field guidance (VFG) and the nonlinear guidance law (NLGL), which is based on proportional navigation missile guidance for following arbitrary trajectories. (Ref. 5) uses VFG to command a small unmanned helicopter to follow different planar paths such as a straight line, circle, ellipsoid, and cubic spline. NLGLs have received considerable attention since their introduction, with various improvements developed in (Refs. 6–9). While VFG and NLGLs have been shown to significantly improve aircraft tracking performance over legacy straight-line waypoint-following algorithms for trajectories in simulations and experiments with small-scale platforms, these guidance algorithms are not well-suited for certification and implementation on production vehicles. In our prior work (Ref. 10), we proposed a new guidance optimization scheme for spacing waypoints on spline trajectories, the *bounded area minimization algorithm*, that can be implemented with existing waypoint guidance software on VTOL rotorcraft. While the algorithm was effective at reducing cumulative cross-track error when the waypoint spacing was sufficiently large, this benefit was not realized for densely-spaced waypoints.

We develop a flight guidance algorithm that enables VTOL rotorcraft to fly more densely-spaced sequences of waypoints that more closely approximate a given spline trajectory. The proposed algorithm takes advantage of fundamental properties of cubic polynomial splines, one of the most commonly used types of spline for VTOL path following in the literature (Refs. 11–13), to enable legacy waypoint guidance algorithms to fly arbitrary spline trajectories. Specifically, it transforms a cubic spline from its monomial basis form $(1,\ x,\ x^2,\ x^3)$ to the B-spline form, in which the coefficients of the basis functions represent the locations of the spline's control points. The convex hull of the control points forms the *control polygon*, which bounds the entirety of the spline trajectory and is a gross discrete approximation of the spline. Successive *midpoint refinements*, the insertion of additional knots midway between the original spline knots, produces a sequence of control polygons that converge to the spline defined by the original control polygon (Refs. 14–16). After several midpoint refinements, the refined control polygon is visually indistinguishable from the original spline, as illustrated in Figure 1. This is why this methodology is typically used in Computer Aided Geometric Design (CAGD) to approximate splines with straight lines on computer screens.

Given a cubic spline trajectory to be flown by a VTOL rotorcraft, our algorithm defines the control points of the spline as the waypoints of the vehicle's flight plan. To our knowledge, this is the first rotorcraft flight guidance algorithm to define the control polygon of a spline or one of its refinements as the guidance trajectory. This approach requires the specification

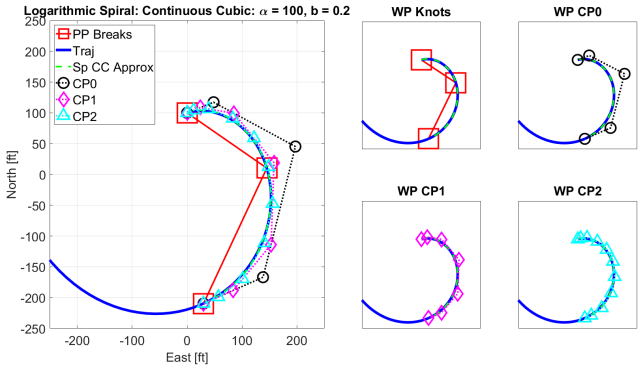

Figure 1. Cubic piecewise-polynomial spline approximation (Sp CC Approx) of a logarithmic spiral trajectory (Traj), with waypoints defined either as knots that connect two piecewise polynomials (PP Breaks), as in our previous work (Ref. 10), or as control points of successive midpoint refinements of the control polygon (CP0, CP1, CP2), as proposed in this paper. The logarithmic spiral can be written in polar coordinates as $r = \alpha \cdot e^{b\theta},\ \theta \in \mathbb{R}$.

of a *stopping criterion* for midpoint refinement of the control polygon. The stopping criterion is usually defined in terms of a threshold on the geometric distance between successive control points or the closest geometric distance from the control polygon to the spline. However, if the spline's control points are to be used as waypoints in a vehicle's flight plan, the midpoint refinement stopping criterion should account for the dynamics of the vehicle that follows the flight plan. We propose a stopping criterion that is based on a simplified flight dynamics model consisting of the wind-axes equations of motion for VTOL rotorcraft, which was first introduced in (Ref. 17).

The next three sections discuss details of B-spline theory, the definition of the flight dynamics-based stopping criterion for control polygon midpoint refinement, and a proposed algorithm for computing recommended waypoint sequences along the control polygon based on this stopping criterion. Then, we present the algorithm's recommended waypoint sequences for five motion primitives using the simplified wind-axes flight model at different vehicle speeds and identify conditions under which flight dynamics constraints are violated. Next, we compare trajectories of the wind-axes flight model to those of a full nonlinear VTOL rotorcraft simulation, performed using a MATLAB implementation of the GENHEL model (Refs. 18–20), during right-handed turn maneuvers at different speeds and different maximum resultant aerodynamic forces along and normal to the flight path. Finally, we present cross-track errors from nonlinear vehicle simulations for trajectories based on the ADS-33 Slalom and Emergency Stop Mission Task Elements (MTEs) (Ref. 21). We conclude with a summary of main results of our study and comments on further evaluation and possible extensions of the algorithm.

B-SPLINE THEORY AND MIDPOINT REFINEMENT

Spline Forms and Change of Basis

As discussed in (Ref. 14), any polynomial spline can be converted from its *monomial basis form* to its *B-spline basis form*. We specifically consider piecewise cubic polynomial splines, since they can be used to represent most typical flight maneuvers. A full spline trajectory used for flight planning may consist of up to 150–200 knots with cubic polynomial splines linking successive knots together, but simpler trajectories can also be comprised of as few as 3 knots. A knot (or more precisely, knot point) is a location in 3-dimensional space where two polynomial spline segments are joined together. See Section I.A in (Ref. 10) for a description of the full 3-dimensional representation of a cubic spline segment between two knots. To simplify the derivation of the spline change of basis, here we will consider a single spatial coordinate of the spline, denoted by the cubic polynomial function $f : \mathbb{R} \to \mathbb{R}$, with the understanding that the derivation extends to all three coordinates. Given knots $n-1$ and n with spatial coordinates $f_{n-1} \in \mathbb{R}$ and $f_n \in \mathbb{R}$, respectively, the coordinate f of the n^{th} segment of the piecewise spline, which lies between these two knots, can be represented in the monomial basis as follows:

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3, \quad x \in [x_{n-1},\, x_n], \qquad (1)$$

where $f(x_{n-1}) = f_{n-1}$ and $f(x_n) = f_n$ are the knot coordinates and $a_i \in \mathbb{R}$, $i = 0, 1, 2, 3$, are constant coefficients. The monomial basis is useful for efficiently evaluating the spline coordinate f at an arbitrary value of $x \in [x_{n-1},\, x_n]$.

Other widely-used bases for the spline space include the Lagrange basis, the Bernstein basis (Bézier curves), and the B-spline basis. The B-spline basis is more efficient than the monomial basis for spline construction, since it enables the specification of smoothness requirements on the function f at the knots. In addition, the B-spline basis is the most useful for our application because the coefficients of a B-spline give the locations of the control points which define the spline's control polygon. (Ref. 22) derives the B-spline form by starting with a quadratic spline in the monomial basis, transforming it to the Lagrange basis via the inversion of a Vandermonde matrix, and then using geometric linear combination arguments to arrive at the Bernstein basis for Bézier curves and finally the B-spline form. The B-spline form can be arrived at more directly by use of the Cox-deBoor recursion relationship, but the basis transformation method used in (Ref. 22) highlights the fact that the original spline is unchanged when converting between the monomial and B-spline basis forms. APPENDIX A presents the same transformation equations for a cubic spline that were performed for a quadratic spline in (Ref. 22).

Properties of Spline Control Polygons

The vertices of the spline control polygon comprise the sequence of control points (Ref. 14). The control polygon can be easily computed by converting a spline to its B-spline form and obtaining the control point locations from the B-spline basis coefficients. The control polygon has several useful properties:

- Strong Convex Hull Property: The B-spline curve is contained in the convex hull of its control polygon.
- Variation Diminishing Property: If the B-spline curve is in a plane, no straight line intersects the curve more times than it intersects the curve's control polygon.
- Affine Invariance: If an affine transformation is applied to a B-spline curve, the resulting transformed curve can be constructed from the affine images of the original curve's control points.

The strong convex hull property provides an advantage to approximating a B-spline using midpoint refinement of its control polygon, rather than sampling the spline at its knot locations. A sequence of straight-line paths through adjacent pairs of knots sampled from the spline is not guaranteed to enclose the entire spline and may cut off parts of the spline if the sampling is not dense enough.

MIDPOINT REFINEMENT STOPPING CRITERIA BASED ON FLIGHT DYNAMICS

Although waypoints could theoretically be placed as close together as possible to best approximate the spline in a geometric sense, there are practical limitations in most automatic flight path guidance algorithms that make such oversampling of a spline infeasible. First, there is a limit to the number of waypoints that can be stored in the array buffer that is used to store a complete flight plan sequence in code. Second, since waypoint latitude and longitude coordinates are stored in double-precision format, waypoints placed too closely together may be interpreted as a single waypoint due to round-off error and precision limitations. Finally, a chain of waypoints could be placed so closely together that the removal of some intermediate waypoints might leave the trajectory unaltered, implying that many of the waypoints are redundant.

In addition to the previously mentioned limitations, there are restrictions on waypoint spacing that arise from the constraints on a particular aircraft's flight dynamics. In order to determine a physics-based practical minimum waypoint spacing, we introduce a wind-axes-based simplified rotorcraft model in the next section.

Simplified Wind-Axes Model of Rotorcraft Dynamics

Rotorcraft flight models are notoriously complicated, especially in the low-speed regime where aerodynamic interactions between the rotors and fuselage are highly unpredictable and not amenable to compact integrable equations of motion math models. In order to derive a practical flight dynamics-based stopping criterion for midpoint refinement that could be run in real-time on a guidance, navigation and control

flight computer, the model must be relatively simple, while still capturing the salient physical effects on the vehicle dynamics. Given these requirements, we chose an augmented wind-axes model from (Refs. 17,23,24) as the basis for deriving the flight dynamics-based stopping criterion. This model is low-dimensional, derived from first principles, and valid over the entire speed range of VTOL flight, including low-speed, cruise, and high-speed flight conditions. While the model is not as accurate as many higher-fidelity models that include multiple rotor states, Computational Fluid Dynamics (CFD), and inertial cross-couplings, the model's wind-axes formulation, which does not require the vehicle heading to align with the ground track angle, enables its application to the wide variety of flight conditions that are typically encountered by many VTOL aircraft (but not by fixed-wing vehicles). The state-space equations of the wind-axes flight model for 2-dimensional planar flight can be written as:

$$\dot{x} = \begin{bmatrix} \dot{V} \\ \dot{\chi} \end{bmatrix} = \begin{bmatrix} g n_{xw} \\ \pm \frac{g}{V} \sqrt{n_{zw}^2 - 1} \end{bmatrix} \tag{2}$$

where g is the gravitational constant, V is the velocity magnitude, χ is the velocity ground track angle, and (n_{xw}, n_{zw}) represent the resultant weight-normalized forces along and normal to the flight path, respectively. The $\pm$ sign accounts for the fact that a right-hand turn results in increasing χ which corresponds to a positive value of $\dot{\chi}$, whereas a left-hand turn results in decreasing χ and thus a negative value of $\dot{\chi}$. We approximate n_{xw} and n_{zw} as constant step inputs. Step inputs were deemed sufficient for the purpose of determining minimum waypoint spacing to simplify the mathematics, although in practice, a human pilot or autonomous flight computer will typically command ramp inputs or higher-order inputs. A full derivation of the model in Equation (2) and its associated simplifying assumptions are described in APPENDIX B.

In addition to these state-space equations, which describe the evolution of the vehicle's trajectory, (Ref. 17) includes an estimate of the power required to accomplish a given maneuver. In low-speed flight, there are no sources of energy exchange other than installed engine power since there is no excess potential energy from losing altitude or excess kinetic energy from bleeding off speed, and any excess rotor energy is assumed to be negligible due to constant rotor rpm. This power estimate was crucial for evolving the model past a point-mass model that was only applicable to rotorcraft at higher speeds. The power equations from (Ref. 17) are:

$$\begin{aligned} P_R^* &= P_{ip}^* + \frac{1-M}{M} + \frac{1}{4}\frac{f_e}{A} V^{*3} \\ P_{ip}^* &= n_{xw} V^* + \sqrt{(n_{xw})^2 + (n_{zw})^2} \cdot v^* \\ v^* &= \frac{(n_{xw})^2 + (n_{zw})^2}{\sqrt{(P_{ip}^*)^2 + V^{*2} n_{zw}^2}} \end{aligned} \tag{3}$$

with the following definitions of terms:

$$\begin{aligned} P_R &= \text{Power required for steady level flight [hp]} \\ P_{ip} &= \text{Induced + propulsive power above that required} \\ &\quad \text{for steady level flight [hp]} \\ v &= \text{Induced velocity magnitude [fps]} \\ M &= \text{Rotor figure of merit} \\ f_e &= \text{Vehicle equivalent flat-plate drag area [ft}^2\text{]} \\ A &= \text{Rotor disc area [ft}^2\text{]} \\ (\,\cdot\,)^* &= \text{Indicates normalized quantities} \end{aligned} \tag{4}$$

Note that while power required is often a limiting factor in being able to accomplish a maneuver when there are no other sources of energy to exchange, other limiting factors may also need to be accounted for, such as bounds on the allowable attitudes and rates (if the vehicle is piloted) and structural load limitations. Since our waypoint spacing method was found to only have limited applicability to the low-speed flight condition, the power equations from Equation (3) were not used for any of the nonlinear simulations, although this will be an area of future work.

Although Equations (2) and (3) describe a simplified two-state rotorcraft flight dynamics model, it is not straightforward to analytically integrate these state-space equations to solve for the vehicle trajectory as a function of time. Moreover, knowing the exact vehicle position as a function of time, while useful, is not necessary for the solution we are proposing.

Minimum WP Spacing Based on Velocity Vector Bearing

Figure 2 illustrates the most common options for an aircraft's trajectory as the aircraft changes between two legs in its flight plan, e.g., from the leg connecting waypoints $n-1$ and n to the leg connecting waypoints n and $n+1$. In our analysis of minimum waypoint (WP) spacing, we will first assume that the aircraft executes Option 2: Late Change OVERFLY, i.e., overflying its current waypoint before changing to the next waypoint. We consider Option 1: Regular Change at the end of this subsection. We do not address Options 3 and 4 because they are not as commonly exercised and typically are not used to precisely follow a given spline trajectory. These options are often used to reroute a vehicle around pop-up obstacles or to execute a Return to Base maneuver in the event of a lost communication link.

Differently from the OVERFLY option illustrated in Figure 2, we do not assume that the vehicle flies through successive waypoints. Instead, we determine the minimum waypoint spacing, d, based on the location at which the vehicle's velocity vector would have the same direction as the vector from its most recent waypoint to its next one. To illustrate this criterion, Figure 3 displays hypothetical flight paths (solid and dashed brown lines) that start at a waypoint $n-1$. At its current location along the flight path, the aircraft (yellow circle) is at distance r_t and bearing θ_t from waypoint $n-1$. Suppose that θ_b is the desired bearing of the next waypoint, n,

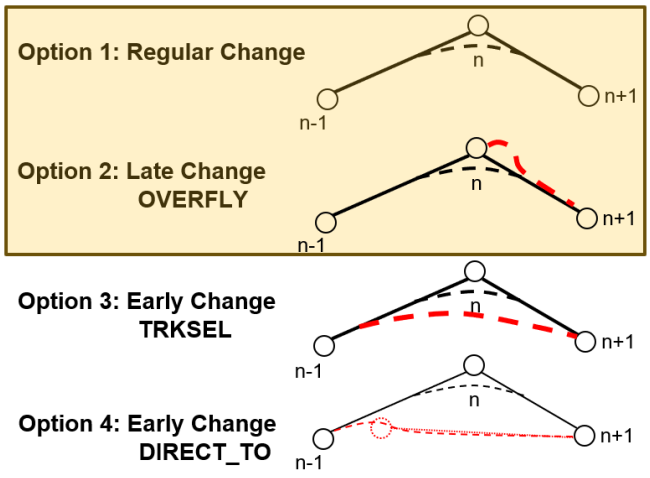

Figure 2. Common Waypoint Change Options (TRKSEL = Track Select)

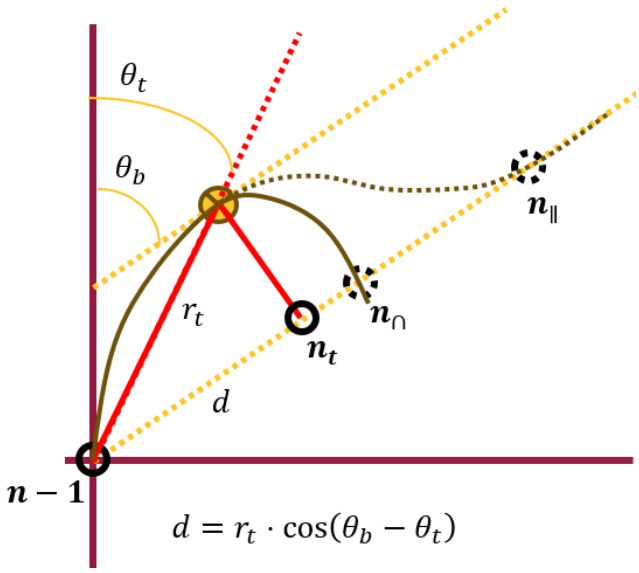

Figure 3. Three Options (n_t, $n_\cap$, and $n_\parallel$) for Waypoint n with Desired Bearing θ_b from Waypoint $n-1$

from $n-1$. Three possible options for n with this bearing, n_t, $n_\cap$, and $n_\parallel$, are shown along the dashed yellow line extending from waypoint $n-1$, which we will refer to as the *waypoint line*. At the location where the aircraft's velocity vector, which is tangent to its flight path (indicated by the dashed yellow line through the aircraft), has bearing θ_b, the point n_t marks the projection of the aircraft's position onto the waypoint line. The point $n_\cap$ marks the intersection of the flight path with the waypoint line in the case where the aircraft's velocity can have arbitrary bearing at this intersection. In the case where the velocity must have bearing θ_b at this intersection, the intersection is marked by the point $n_\parallel$.

From Figure 3, it is evident that the length of the leg (straight-line path) between waypoint $n-1$ and each possible waypoint n is smallest for $n = n_t$ and largest for $n = n_\parallel$. The point $n = n_\cap$ could be even closer to waypoint $n-1$ than n_t if the flight path can curve backward toward $n-1$. However, since we approximate the spline trajectory using midpoint refinement of its control polygon, thus defining the sides of the polygon as the legs of the flight plan, we specify that the aircraft's flight path should at least align with these legs; i.e., its velocity vector should converge to the same bearing as each leg, which can be achieved by setting waypoint n to either n_t or $n_\parallel$. If the aircraft has enough space to maneuver, we can also specify that it should converge to zero cross-track error with respect to each leg. Since we aim to define the minimum possible waypoint spacing, we set the waypoint n to n_t, which is closer to waypoint $n-1$ than $n_\parallel$ is. Thus, as illustrated in Figure 3, the minimum waypoint spacing d is defined as:

$$d = r_t \cdot \cos(\theta_b - \theta_t) \tag{5}$$

Note that $d < 0$ for $(\theta_b - \theta_t) \in (\frac{\pi}{2}, \frac{3\pi}{2})$ rad. Since negative waypoint spacings are not possible, d is set to zero when $(\theta_b - \theta_t)$ is in this range (i.e., there is no lower bound on the leg length).

We can determine the minimum waypoint spacing for the Regular Change option, which involves the aircraft changing ground track angle before it reaches its next waypoint, using our analysis for the OVERFLY option. We consider an aircraft in CRUISE flight, under the conditions that ground track angle changes are initiated with bank angle changes and turns are coordinated through either the zeroing of side-slip or zeroing of side-acceleration. The *change distance* Δ is defined as the distance from the next waypoint at which the aircraft switches from the current leg to the next leg. Assuming that the aircraft flies at a constant bank angle during the change and thus inscribes a circle between the two legs, the minimum change distance Δ_{CRUISE} can be computed as follows (see (Ref. 10), Section I.B:)

$$\Delta_{CRUISE} = \frac{V^2}{g \cdot \tan\phi_{max}} \tan\left|\frac{\Delta\psi_L}{2}\right| \tag{6}$$

where $\Delta\psi_L$ is the ground track angle change between successive legs and ϕ_{max} is the maximum allowable bank angle command. This distance can be computed *a priori*, and thus the minimum waypoint spacing d_{RC} for the Regular Change option is given by:

$$d_{RC} = d - \Delta_{CRUISE} \tag{7}$$

where d is defined in Equation (5). Note that the minimum spacing is lower for the Regular Change option than for the OVERFLY option because during Regular Change, the vehicle initiates the ground track angle change before reaching its original target waypoint.

Monotonicity of the Ground Track Angle Function

It is important to confirm that the ground track angle $\chi(t)$ is a well-behaved function that is always either increasing or decreasing with time t, since it is not physically realistic for the

aircraft's trajectory to have discontinuities and, as mentioned earlier, we do not consider trajectories that curve backward toward the previous waypoint. Toward this end, we show that $\chi(t)$ is either monotonically increasing or decreasing for fixed values of n_{xw} and n_{zw}. Given an initial velocity magnitude V_o, the state-space model in Equation (2) with the positive value of $\dot{\chi}$ (indicating a right-hand turn) can be integrated analytically to obtain the solution $\chi(t)$:

$$\chi(t) = \frac{1}{n_{xw}} \left(\sqrt{n_{zw}^2 - 1} \right) \cdot \ln \left(\frac{V_o + g n_{xw} t}{V_o} \right), \quad t \geq 0. \tag{8}$$

It can be assumed that $n_{zw} \geq 1$, since aerobatic low-g pushover maneuvers are not being considered, only 2D maneuvers that alter the direction of the velocity vector in the local level plane. For $n_{xw} > 0$ and $t > 0$, we have that $(V_o + g n_{xw} t)/V_o > 1$ and so its natural logarithm is positive, yielding $\chi(t) > 0$. For $n_{xw} < 0$ and $t > 0$, $(V_o + g n_{xw} t)/V_o < 1$ and its natural logarithm is negative, which also yields $\chi(t) > 0$. Thus, irrespective of whether the vehicle is accelerating ($n_{xw} > 0$) or decelerating ($n_{xw} < 0$) during a right-hand turn, $\chi(t)$ is always positive, i.e., monotonically increasing. Using a similar argument, for a negative value of $\dot{\chi}$ (left-hand turn), it can be shown that $\chi(t)$ is always negative, i.e., monotonically decreasing.

Extension to Chains of Waypoints

The proposed criterion for waypoint spacing can be applied to chains of waypoints that form an entire flight plan. Figure 4 shows the path (solid brown line) of an aircraft that flies a chain of three waypoints, labeled $n-1$, n, and $n+1$, in the OVERFLY condition. Waypoint n has been placed at point n_t in Figure 3. As the aircraft approaches waypoint n, the guidance logic will switch to waypoint $n+1$ based on a combination of conditions, such as reaching a certain radial distance from waypoint $n+1$ or crossing the line perpendicular to the leg between waypoints $n-1$ and n that intersects waypoint n. Immediately thereafter, the same criterion can be applied as the vehicle travels from n to $n+1$ to specify that the vehicle aligns its velocity vector in the same direction as the leg between n and $n+1$, which generates the value of d for this leg. Using this criterion, successive waypoints can be similarly defined for the entire spline trajectory. If the aircraft flies the waypoints in the Regular Change condition instead of OVERFLY, the waypoint spacing can be augmented by the minimum change distance defined in Equation (6).

WAYPOINT SPACING ALGORITHM

The proposed algorithm, Algorithm 1, takes as input a set of parameters and specifications that include a cubic piecewise-polynomial spline trajectory, f; the waypoint change option (see Figure 2); and the vehicle's entry speed, V_e, and entry ground track angle, χ_e, defined as the vehicle's speed and ground track angle, respectively, before its first waypoint. The control polygon is computed by converting the piecewise-polynomial spline into its B-spline form and obtaining the locations of the polygon's vertices (the control points) from the

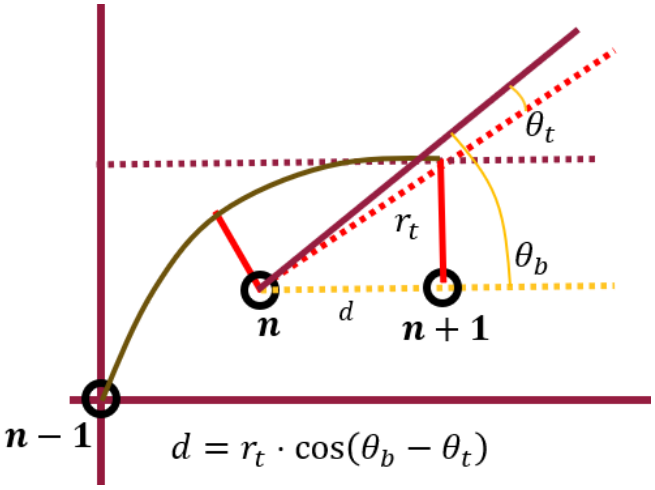

Figure 4. Illustration of Waypoint $n+1$ Placement in a Chain of Three Waypoints

B-spline coefficients. The area A_{kn} enclosed by the spline and the straight-line path through the spline knots, and the area A_{cp} enclosed by the spline and the control polygon, are computed using Green's Theorem, which can be applied to find the area A of a region bounded by a closed curve C (see (Ref. 10), Section II):

$$A = \frac{1}{2} \oint_C (-y dx + x dy) \tag{9}$$

The algorithm performs midpoint refinement of the control polygon to find the polygon that most closely approximates the trajectory, i.e., yields the smallest area A_{cp}, subject to lower bounds on the lengths of its sides, i.e., the legs of the flight plan. Indexing the legs by $l = 1, \ldots, L$, each leg has length $d_{leg,l}$, and its corresponding lower bound is the minimum waypoint spacing for either the OVERFLY or Regular Change option, denoted by $d_{min,l}$. Midpoint refinements are stopped if $d_{leg,l} < d_{min,l}$ for any leg l. If A_{cp} for the resulting control polygon is less than A_{kn}, then the waypoints are defined as the control points; otherwise, the waypoints are defined as the spline knots. To simplify notation, the subscript l will be dropped from quantities that vary with leg l (e.g., $d_{min,l}$ will be written as d_{min}) when we are referring to an arbitrary leg of the flight plan.

The minimum waypoint spacing d (if explicitly indexed by leg, d_l) in Equation (5) is computed from the position $(x(t_d), y(t_d))$ of the aircraft in an inertial coordinate frame fixed at its previous waypoint (e.g., waypoint $n-1$ in Figure 3), where t_d is the time the aircraft takes to reach distance r_t and bearing θ_t from its previous waypoint. The velocity ground track angle at this time, $\chi(t_d)$, equals θ_b (technically, $\theta_{b,l}$), the desired bearing of the next waypoint from the previous waypoint, and this relation can be used to solve for t_d. Hence, Equation (5) can be written as:

$$\begin{aligned} d &= r_t \cdot \cos(\theta_b - \theta_t) \\ &= \sqrt{x(t_d)^2 + y(t_d)^2} \cdot \cos\left(\theta_b - \tan^{-1}\left(\frac{y(t_d)}{x(t_d)} \right) \right) \end{aligned} \tag{10}$$

Algorithm 1 Waypoint Spacing using Control Polygon Midpoint Refinement with Flight Dynamics-Based Constraints

Require: A cubic piecewise-polynomial spline trajectory, f, constructed using three knots and represented in the monomial basis; waypoint change option (OVERFLY or Regular Change); n_{xw} option (zero or nonzero); n_{zw}; $\Delta\psi_L$; ϕ_{max}; V_e; χ_e; V_f

$A_{kn} \leftarrow$ Area of region enclosed by f and the straight-line path through the knots, from Equation (9)
if $(A_{kn} == 0)$ **then** ▷ f is a straight line
  Flight plan waypoints $\leftarrow$ Knot locations
else
  $TOO\_CLOSE \leftarrow 0$ ▷ Boolean variable that indicates a violation of flight dynamics-based constraints on waypoint spacing
  while $(!TOO\_CLOSE)$ **do**
    Compute control points $C_0, ..., C_{N_{CP}-1}$, the vertices of the control polygon: on first iteration, via conversion of f to the B-spline basis (see APPENDIX A); on subsequent iterations, via midpoint refinement of the control polygon
    $A_{cp} \leftarrow$ Area of region enclosed by f and the control polygon, from Equation (9)
    for $(l = 1, ..., L)$ **do** ▷ The control polygon has L sides (legs of flight plan)
      Compute $\theta_{b,l}$ from the control polygon
      if $(n_{xw} == 0)$ **then**
        $t_d, x(t_d), y(t_d) \leftarrow$ Equations (13), (14)
      else
        $n_{xw}, t_d, t_{dec} \leftarrow$ Equations (15), (19), (20)
        $x(t_d), y(t_d) \leftarrow$ Equation (21) if $t_{dec} \geq t_d$; Equation (22) otherwise
      end if
      $d_{min,l} \leftarrow$ Equation (11)
      $d_{leg,l} \leftarrow$ Length of leg l, the distance between control points C_{l-1} and C_l
    end for
    $TOO\_CLOSE \leftarrow 1$ if any $d_{leg,l} < d_{min,l}$; 0 otherwise
  end while
  if $(A_{cp} < A_{kn})$ **then**
    Flight plan waypoints $\leftarrow$ Control points
  else
    Flight plan waypoints $\leftarrow$ Knot locations
  end if
end if
return Flight plan waypoints

Thus, the distance d_{min} is defined as follows for the OVERFLY and Regular Change options:

$$d_{min} = \begin{cases} d \text{ from Eq. (10)}, & \text{OVERFLY} \\ d_{RC} \text{ from Eq. (7) with } d \text{ from Eq. (10)}, & \text{Regular Change} \end{cases} \tag{11}$$

We compute d for two cases: (1) $n_{xw} = 0$, i.e., there is zero along-track acceleration or deceleration; and (2) $n_{xw} \neq 0$. In both cases, there is constant normal acceleration $n_{zw} > 1$ to maintain level flight.

In the case $n_{xw} = 0$, we assume that the waypoints are sufficiently close together such that the aircraft's speed does not change considerably between waypoints, and we set this constant speed to V_e. The state-space model in Equation (2) reduces to:

$$\begin{aligned} V(t) &= V_e = constant \\ \chi(t) &= \frac{g}{V_e}\left(\sqrt{n_{zw}^2 - 1}\right) \cdot t \end{aligned} \tag{12}$$

Setting $\chi(t_d) = \theta_b$ in Equation (12) and solving for t_d yields:

$$t_d = \frac{V_e}{g}\left(\sqrt{n_{zw}^2 - 1}\right)^{-1} \cdot \theta_b \tag{13}$$

Then Equations (12) and (13) can be used to compute the aircraft's position coordinates at time t_d as the following expressions, which are straightforward to integrate:

$$\begin{aligned} x(t_d) &= \int_0^{t_d} V(t) \cdot \cos(\chi(t)) dt \\ &= \int_0^{t_d} V_e \cdot \cos\left[\frac{g}{V_e}\left(\sqrt{n_{zw}^2 - 1}\right) \cdot t\right] dt \\ y(t_d) &= \int_0^{t_d} V(t) \cdot \sin(\chi(t)) dt \\ &= \int_0^{t_d} V_e \cdot \sin\left[\frac{g}{V_e}\left(\sqrt{n_{zw}^2 - 1}\right) \cdot t\right] dt \end{aligned} \tag{14}$$

In the case $n_{xw} \neq 0$, we assume a constant n_{xw}, approximated as the aircraft's average n_{xw} throughout the maneuver, and compute the aircraft's speed at each waypoint (control point) based on this value of n_{xw}. We define $d_{tot} = \sum_{l=1}^{L} d_{leg,l}$ as the total length of all legs of the flight plan. Let V_j denote the aircraft's speed at control point C_{j-1}, $j = 1, ..., N_{CP}$. V_1 is the aircraft's entry speed, V_e. We assume that the aircraft is decelerating to a desired final speed, V_f, during its maneuver, such that $V_{N_{CP}}$ should be V_f. Typical values of V_f are around 5 knots for a hover target, as most of the effort to achieve hover is applied to bring V_f close to zero. The value of n_{xw} is computed from the constraint that $V_{N_{CP}} = V_f$:

$$n_{xw} = \frac{V_f^2 - V_1^2}{2 \cdot d_{tot}} \tag{15}$$

If $n_{xw} > n_{xw,max}$, n_{xw} is set to $n_{xw,max}$, and a higher value of V_f than desired is accepted. The aircraft's speed at each control point is then given by:

$$V_1 = V_e; \quad V_j = (V_1^2 + 2 \cdot n_{xw} \cdot \sum_{i=1}^{j-1} d_{leg,i})^{1/2}, \quad j = 2, ..., N_{CP} \tag{16}$$

Given the aircraft speed V_j at the control point at the beginning of a leg, the state-space model in Equation (2) can be integrated to obtain:

$$\begin{aligned} V(t) &= V_j + g n_{xw} t \\ \chi(t) &= \int_0^t \frac{g}{V(t)}\left(\sqrt{n_{zw}^2 - 1}\right) dt \end{aligned} \tag{17}$$

Combining the two expressions in Equation (17) and integrating the expression for $\chi(t)$ yields:

$$\chi(t) = \frac{1}{n_{xw}} \left(\sqrt{n_{zw}^2 - 1} \right) \ln \left(\frac{V_j + g n_{xw} t}{V_j} \right) \tag{18}$$

Setting $\chi(t_d) = \theta_b$ in Equation (18) and solving for t_d yields:

$$t_d = \frac{V_j}{g n_{xw}} \left(\exp \left\{ \theta_b \cdot \frac{n_{xw}}{\sqrt{n_{zw}^2 - 1}} \right\} - 1 \right) \tag{19}$$

We can use Equation (17) to solve for the time t_{dec} that the aircraft takes to decelerate to V_{j+1}, the computed speed at the control point at the end of the leg:

$$t_{dec} = \frac{V_{j+1} - V_j}{g n_{xw}} \tag{20}$$

We will first address the case where $t_{dec} \geq t_d$. Equations (18) and (19) can be used to compute the aircraft's coordinates at time t_d:

$$\begin{aligned} x(t_d) &= \int_0^{t_d} V(t) \cdot \left(\cos \left[\frac{\sqrt{n_{zw}^2 - 1}}{n_{xw}} \cdot \ln \left| \frac{V(t)}{V_j} \right| \right] \right) dt \\ y(t_d) &= \int_0^{t_d} V(t) \cdot \left(\sin \left[\frac{\sqrt{n_{zw}^2 - 1}}{n_{xw}} \cdot \ln \left| \frac{V(t)}{V_j} \right| \right] \right) dt \end{aligned} \tag{21}$$

where $V(t)$ is defined in Equation (17). These integrals are not easily solved in closed form, but can be efficiently computed with quadrature. In tests on a 13th Gen Intel Core i7-13700 2.10 GHz processor with 64.0 GB of RAM, it took approximately 0.02 seconds to numerically integrate over 150 instances of Equation (21), without further code optimization. Thus, there are no foreseen barriers to numerically integrating Equation (21) in real-time on a flight control computer.

We next address the case where $t_{dec} < t_d$. In this case, the aircraft performs the maneuver in two phases: (1) turning while decelerating to V_{j+1} from time 0 to t_{dec}, then (2) pure turning at constant V_{j+1} from time t_{dec} to t_d. The coordinates $(x(t_{dec}), y(t_{dec}))$ of the aircraft at time t_{dec} can be computed from Equation (21), with the upper limit of integration t_d replaced by t_{dec} from Equation (20). The aircraft's coordinates at time t_d are then computed by adding $(x(t_{dec}), y(t_{dec}))$ to the expressions in Equation (21), integrated from t_{dec} to t_d:

$$\begin{aligned} x(t_d) &= x(t_{dec}) + \int_{t_{dec}}^{t_d} V_{j+1} \cdot \cos \left[\frac{\sqrt{n_{zw}^2 - 1}}{n_{xw}} \cdot \ln \left| \frac{V_{j+1}}{V_j} \right| \right] dt \\ y(t_d) &= y(t_{dec}) + \int_{t_{dec}}^{t_d} V_{j+1} \cdot \sin \left[\frac{\sqrt{n_{zw}^2 - 1}}{n_{xw}} \cdot \ln \left| \frac{V_{j+1}}{V_j} \right| \right] dt \end{aligned} \tag{22}$$

The ground track angle at time t_d is found by integrating Equation (2) over the two time intervals $[0, t_{dec}]$ (with $V(t)$ defined in Equation (17)) and $[t_{dec}, t_d]$ (with $V(t) = V_{j+1}$):

$$\chi(t_d) = \theta_b = \sqrt{n_{zw}^2 - 1} \left(\frac{1}{n_{xw}} \ln \left| \frac{V_{j+1}}{V_j} \right| + \frac{g}{V_{j+1}} (t_d - t_{dec}) \right) \tag{23}$$

ALGORITHM PERFORMANCE ON MOTION PRIMITIVES

Algorithm 1 was tested on five motion primitives: straight line, sinusoid, semicircle, decaying exponential, and logarithmic spiral. The straight-line primitive was left unscaled, as it was simply a test of the algorithm's ability to reject a case where no control polygon midpoint refinement was required. The sinusoid and semicircle primitives were sized to be applicable to common ADS-33/MIL-DTL-32742 MTEs such as the Slalom and Pirouette. The exponential was sized to match a future proposed MTE, the Emergency Stop, that is tailored specifically for VTOL UAS (Ref. 21). Finally, the logarithmic spiral was sized with an initial radius based on the Pirouette and modeled after a spiral search pattern that is currently in operation on certain cargo/utility VTOL platforms. For all motion primitives, the algorithm was run with the OVERFLY waypoint change and $n_{xw} = 0$ options; for the exponential, the algorithm was additionally run with the $n_{xw} \neq 0$ option.

The following subsections present the minimum waypoint spacings (i.e., minimum leg lengths), d_{min} from Equation (11), that are generated by Algorithm 1 for each motion primitive (except the straight line), given different aircraft entry speeds V_e. Instead of executing Lines 7-19 based on the Boolean condition in Line 6, these lines were repeated three times to produce an initial control polygon (CP0) and two midpoint refinements (CP1, CP2). This was done to identify any legs of CP0, CP1, and CP2 whose length, d_{leg}, is shorter than the minimum length d_{min} computed for that leg, indicating that this leg violates flight dynamics-based constraints on its length. In addition, the area A_{kn} enclosed by the spline and the baseline (BL) path through the knots and the area A_{cp} enclosed by the spline and each control polygon (CP0, CP1, CP2) are shown for comparison.

Straight Line

The straight-line trajectory is a trivial case, but must be tested to ensure that the algorithm handles it properly. Since the area enclosed by the trajectory and the straight-line path through the waypoints (initially, the knots) is zero, successive midpoint refinements would not reduce this area, so the algorithm terminates without inserting any additional waypoints (control points), as expected.

Sinusoid: Slalom

The sinusoid motion primitive can be used to approximate the ADS-33 Slalom MTE, as shown in Figure 5. It is clear from the figure that the given spline representation of the sinusoid is closely approximated by the control polygon after only one or two midpoint refinements (CP1, CP2). We apply the specification for the Slalom MTE that the aircraft flies at a constant speed throughout the maneuver, which we define as V_e. We compute waypoint spacings for four values of V_e defined in the Slalom MTE standards: 15 knots (desired) and 30 knots (adequate) in Degraded Visual Environments (DVE), and 40

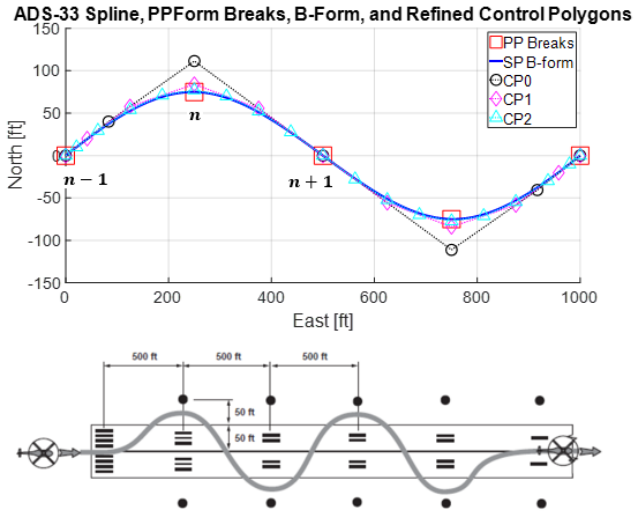

FIGURE 59. Suggested course for slalom maneuver

Figure 5. *Top:* **Cubic piecewise-polynomial spline approximation of a sinusoid (SP B-form), shown with knots (PP Breaks) and control points of successive midpoint refinements of the control polygon (CP0, CP1, CP2).** *Bottom:* **ADS-33 Slalom MTE from (Ref. 25).**

knots (desired) and 60 knots (adequate) in Good Visual Environments (GVE). For CP0, CP1, and CP2, Table 1 displays the values of d_{leg} in bold and the values of d_{min} computed for each entry speed V_e. Values of d_{min} for which $d_{leg} < d_{min}$ are highlighted in red, along with the corresponding entry speed. The table shows that the legs of CP0 all exceed d_{min} for $V_e = 15$ and 30 knots, but two legs are shorter than d_{min} for $V_e = 40$ and 60 knots.

The areas A_{cp} for CP0, CP1, and CP2 are about 90%, 25%, and 7%, respectively, of the area A_{kn}, indicating that the successive midpoint refinements produce flight plans that approximate the spline trajectory more and more closely, thus reducing the tracking error for the maneuver. However, this comes at the cost of incurring violations of flight dynamics-based constraints (henceforth simply referred to as "violations"): the number of legs that are shorter than d_{min} generally increases with successive midpoint refinements and increasing entry speed. No constraints are violated at the lowest entry speed of 15 knots for any of the control polygons.

Semicircle: Pirouette

As shown in Figure 6, the semicircle motion primitive models the ADS-33 Pirouette MTE, a slow, constant-speed circular maneuver with a radius of 100 ft. It represents one of the most challenging low-speed rotorcraft maneuvers and epitomizes "heading independent of ground track," since the nose of the vehicle must always be pointed at the center of the circle as the vehicle moves around the circle's perimeter. Figure 6 shows that the spline representation of the semicircle is very

Table 1. Sinusoid Motion Primitive: Waypoint Spacings for Constant Aircraft Speed

Entry Ground Track Angle: 90 deg

Bounded Area [ft²]		Grd Track Chg [deg]		Separation Distances [ft]									
BL 5123	CP0 (5pts) 4579	-26,4,44,4	CP	93		180		180		93			
			15	23		5		29		5			
			30	91		19		116		19			
			40	161		33		206		33			
			60	362		75		463		75			
CP1	CP1 (7pts) 1267	-26,2,13, 23,13,2	CP	47		91	128		128		91		47
			15	23		2	14		21		14		2
			30	91		10	54		84		54		10
			40	161		17	96		149		96		17
			60	362		38	217		334		217		38
CP2	CP2 (11pts) 368	-26,1,4,6,9, 12,9,6,4,1	CP	23	46	67	65	63	63	65	67	46	23
			15	23	1	5	7	10	13	10	7	5	1
			30	91	5	19	27	39	51	39	27	19	5
			40	161	9	33	49	70	90	70	49	33	9
			60	362	19	75	109	158	203	158	109	75	19

RED indicates violation of flight dynamics-based constraints

Table 2. Semicircle Motion Primitive: Waypoint Spacings for Constant Aircraft Speed

Entry Ground Track Angle: 0 deg

Bounded Area [ft²]		Grd Track Chg [deg]		Separation Distances [ft]									
BL 5708	CP0 (5pts) 4292	0,45,90, 45	CP	50		141		141		50			
			05	0		4		3		4			
			10	0		13		9		13			
			15	0		29		20		29			
			30	0		117		79		117			
CP1	CP1 (7pts) 855	0,27,37, 53,37,27	CP	25		56	84		84		56		25
			05	0		3	4		4		4		3
			10	0		10	12		13		12		10
			15	0		23	27		30		27		23
			30	0		93	109		119		109		93
CP2	CP2 (11pts) 102	0,14,22, 19,22,28,22 ,19,22,14	CP	13	26	40	40	39	39	40	40	26	13
			05	0	2	3	2	3	3	3	2	3	3
			10	0	6	9	8	9	11	9	8	9	6
			15	0	14	20	18	20	24	20	18	20	14
			30	0	58	81	73	81	95	81	73	81	58

RED indicates violation of flight dynamics-based constraints

closely approximated by the control polygon after two midpoint refinements (CP2). We define the aircraft's speed as V_e and compute waypoint spacings for four speeds defined in the Pirouette MTE standards: 5 knots (desired) and 10 knots (adequate) in DVE, and 15 knots (desired) and 30 knots (adequate) in GVE.

Table 2 displays the same information as Table 1 for the semicircle motion primitive. The table shows that for each control polygon refinement, all legs exceed d_{min} for $V_e = 5$ and 10 knots. For CP0 and CP1, this is also true for $V_e = 15$ knots; however, refining the control polygon to CP2 incurs a violation at this speed (the last leg is shorter than d_{min}). Thus, the semicircle maneuver can likely be performed at speeds greater than 15 knots, depending on how many refinements of the control polygon are applied.

Note that the semicircle motion primitive can be flown using existing guidance algorithms for circular trajectories that are even simpler than control polygon midpoint refinement. A comparison of these methods is a possible area of future work.

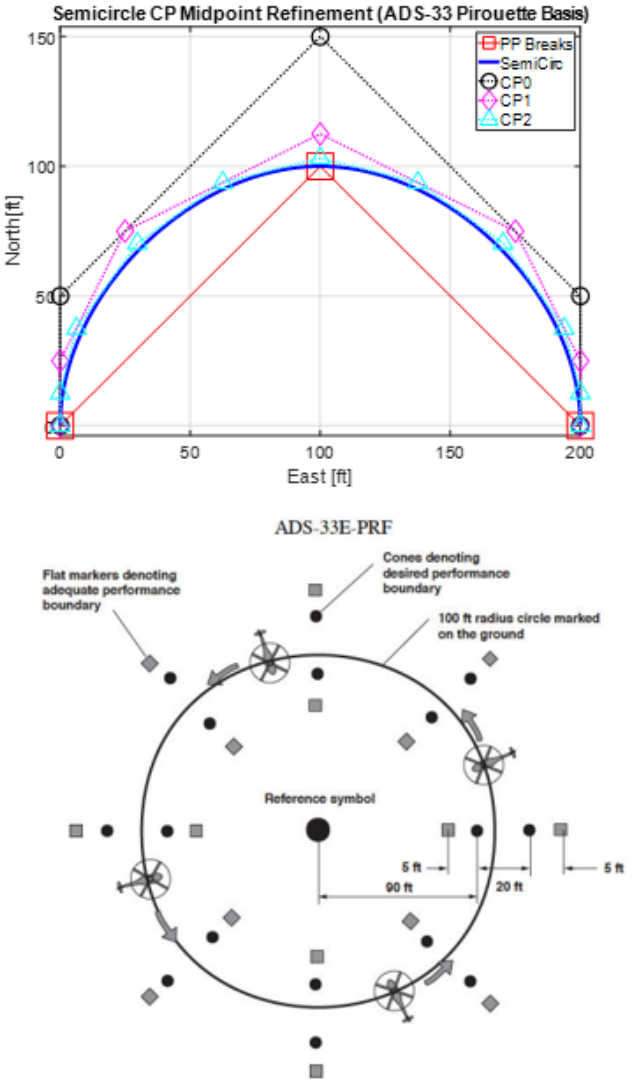

Figure 25. Suggested course for pirouette maneuver

Figure 6. *Top:* **Cubic piecewise-polynomial spline approximation of a semicircle (SemiCirc), shown with knots (PP Breaks) and control points of successive midpoint refinements of the control polygon (CP0, CP1, CP2).** *Bottom:* **ADS-33 Pirouette MTE from (Ref. 25).**

Exponential: Emergency Stop

The Emergency Stop is a newer MTE, first proposed in (Ref. 21), that was designed for VTOL UAS. One of the primary objectives of this maneuver is to *check the ability to accomplish an unexpected stop from a representative autonomous operation speed with limited forward distance travel.* The trajectory of a vehicle during this maneuver can be modeled by the exponential motion primitive. Assuming an aircraft Froude scaling of 1, the desired and adequate values of the pertinent requirements from (Ref. 21) are summarized in Table 3.

The exponential Emergency Stop maneuver used for testing

Table 3. Emergency Stop MTE Performance Standards

Metric	Desired	Adequate
Minimum speed from which deceleration is initiated, V_{decel}	60 kts	50 kts
Maximum allowable forward travel after deceleration is initiated	350 ft	600 ft

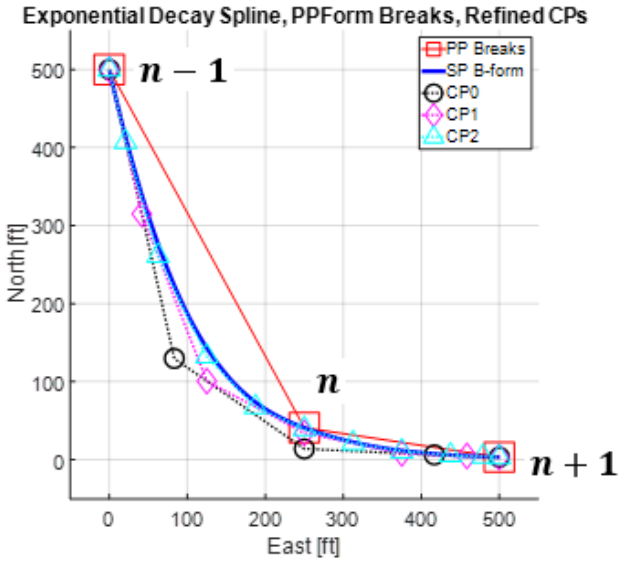

Figure 7. Cubic piecewise-polynomial spline approximation of a decaying exponential (SP B-form), shown with knots (PP Breaks) and control points of successive midpoint refinements of the control polygon (CP0, CP1, CP2).

Algorithm 1 is shown in Figure 7. The aircraft enters from the north while decelerating and turning to the east. The MTE requirements allow for the aircraft to execute the turn in either a coordinated or uncoordinated fashion, as long as it concludes in a steady hover with minimal altitude loss. We first specify that the aircraft flies at a constant speed (i.e., $n_{xw} = 0$), its entry speed V_e, which is set to four values with a maximum of 60 knots, the desired value in Table 3. The aircraft's maximum allowable forward travel was set to 500 ft, which is between the desired and adequate values in the table.

Table 4 displays the same information as the previous tables for the exponential motion primitive. The table shows that for speeds between 40 and 60 knots, each control polygon (CP0, CP1, and CP2) has one or more legs that are shorter than d_{min}, and thus incur a violation. Since these speeds overlap with the range of speeds (50–60 knots) at which the Emergency Stop MTE is meant to be flown, we attempt to eliminate these violations by running Algorithm 1 with $n_{xw} \neq 0$, which modifies the waypoint spacing calculations to account for the vehicle's deceleration during the maneuver.

A comparison of waypoint spacings for the constant-speed (CS; $n_{xw} = 0$) and decelerating (DCL; $n_{xw} \neq 0$) cases for each

Table 4. Exponential Motion Primitive: Waypoint Spacings for Constant Aircraft Speed

Entry Ground Track Angle: 180 deg

Bounded Area [ft²]		Grd Track Chg [deg]	Separation Distances [ft]									
BL 23508	CP0 (5pts) 9347	-13, -43, -32,-1	CP	379		203		167		83		
			15	14		29		26		1		
			30	54		115		102		5		
			40	96		205		181		9		
			60	217		461		408		19		
CP1	CP1 (7pts) 3551	-13, -9, -41, -16, -9, 0	CP	190	230	142	128	83	42			
			15	14	10	28	16	10	0			
			30	54	39	114	64	39	0			
			40	96	70	202	114	70	0			
			60	217	158	454	256	158	0			
CP2	CP2 (11pts) 675	-13,-3,-10, -18, -22, -8, -7, -5, -3, 0	CP	95	152	143	90	69	65	63	63	42	21
			15	14	4	11	18	20	9	8	6	4	0
			30	54	14	43	70	81	36	32	23	14	0
			40	96	25	77	125	144	63	56	41	25	0
			60	217	57	173	281	324	142	126	92	57	0

RED indicates violation of flight dynamics-based constraints

Table 5. Exponential Motion Primitive, Control Polygon CP0: Waypoint Spacings for Decreasing Aircraft Speed

Entry Ground Track Angle: 180 deg

Bounded Area [ft²]		n_{xw} Required	Separation Distances [ft]				
BL 23508	CP0 (5pts) 9347	-0.386 [ft/s²]	CP	379	203	167	83
			15 kts CS	14	29	26	1
			15 kts DCL	14	16	8	0
			V_{entry}	15kts	11.1kts	8.2kts	4.7kts
Ground Trk Change [deg]		-1.541 [ft/s²]	30 kts CS	54	115	102	5
			30 kts DCL	54	63	31	0
			V_{entry}	30kts	22.1kts	16.4kts	9.5kts
-13, -43, -32,-1		-2.739 [ft/s²]	40 kts CS	96	205	181	9
			40 kts DCL	96	111	55	1
			V_{entry}	40kts	29.5kts	21.9kts	12.6kts
		-6.163 [ft/s²]	60 kts CS	217	461	408	19
			60 kts DCL	217	251	123	2
			V_{entry}	60kts	44.3kts	32.9kts	19.0kts

RED indicates violation of flight dynamics-based constraints

Table 6. Exponential Motion Primitive, Control Polygon CP1: Waypoint Spacings for Decreasing Aircraft Speed

Entry Ground Track Angle: 180 deg

Bounded Area [ft²]		n_{xw} Required	Separation Distances [ft]						
BL 23508	CP1 (7pts) 3551	-0.393 [ft/s²]	CP	190	230	142	128	83	42
			15 kts CS	14	10	28	16	10	0
			15 kts DCL	14	8	14	5	2	0
			V_{entry}	15 kts	13.1 kts	10.4 kts	8.4 kts	5.9 kts	3.4 kts
Ground Track Change [deg]		-1.573 [ft/s²]	30 kts CS	54	39	114	64	39	0
			30 kts DCL	54	30	55	20	6	0
			V_{entry}	30 kts	26.3 kts	20.9 kts	16.7 kts	11.7 kts	6.8 kts
-13, -9, -41, -16, -9, 0		-2.796 [ft/s²]	40 kts CS	96	70	202	114	70	0
			40 kts DCL	96	54	98	35	11	0
			V_{entry}	40kts	35kts	27.8kts	22.3kts	15.7kts	9.1kts
		-6.292 [ft/s²]	60 kts CS	217	158	454	256	158	0
			60 kts DCL	217	121	220	80	24	0
			V_{entry}	60kts	52.5kts	41.8kts	33.4kts	23.5kts	13.6kts

RED indicates violation of flight dynamics-based constraints

Table 7. Exponential Motion Primitive, Control Polygon CP2: Waypoint Spacings for Decreasing Aircraft Speed

Entry Ground Track Angle: 180 deg

Bounded Area [ft²]		n_{xw} Req	Separation Distances [ft]										
BL 23508	CP2 (11pts) 675	15 kts	CP	95	152	143	90	69	65	63	63	42	21
			CS	14	4	11	18	20	9	8	6	4	0
		-0.4 [ft/s²]	DCL	14	3	7	9	8	3	2	1	0	0
			V_{entry}	15	14.1	12.5	10.8	9.5	8.4	7.3	5.9	4.2	2.4
		30 kts	CS	54	14	43	70	81	36	32	23	14	0
Ground Track Change [deg]		-1.6 [ft/s²]	DCL	54	13	30	36	33	11	7	4	1	0
			V_{entry}	30	29.2	25	21.5	19	16.9	14.6	11.9	8.4	4.8
-13,-3,-10, -18, -22, -8, -7, -5, -3, 0		40 kts	CS	96	25	77	125	144	63	56	41	25	0
		-2.84 [ft/s²]	DCL	96	22	53	64	58	20	13	6	2	0
			V_{entry}	40	37.6	33.3	28.7	25.4	22.5	19.4	15.8	11.2	6.5
		60 kts	CS	217	57	173	281	324	142	126	92	57	0
		-6.38 [ft/s²]	DCL	217	50	120	144	131	45	30	14	4	0
			V_{entry}	60	56.3	49.9	43	38	33.7	29.1	23.8	16.8	9.7

RED indicates violation of flight dynamics-based constraints

of the three control polygons is shown in Tables 5, 6, and 7. In the decelerating case, V_f is set to 0 knots, n_{xw} *required* is computed from Equation (15), and the rows labeled V_e in the tables list the aircraft's speed at the start of each leg. For the control polygon CP0, Table 5 shows that accounting for the aircraft's deceleration between control points reduces the minimum distance required to reorient along the spline. Due to this reduction in waypoint spacing, there are no longer any violations at a speed of 40 knots and only one violation at 60 knots, where the magnitude $d_{min,l} - d_{leg,l}$ of the violation is significantly lower than that of the two violations at 60 knots in the constant-speed case. Similarly, for control polygons CP1 and CP2, Tables 6 and 7 show that the decelerating case also reduces the number of violations and their magnitudes relative to the constant-speed case at speeds of 40 and 60 knots. Future versions of the algorithm can account for the particular magnitude $d_{min,l} - d_{leg,l}$ of any violations in determining minimum waypoint spacing, rather than just whether $d_{min,l} - d_{leg,l} > 0$.

Logarithmic Spiral: Search Pattern

The logarithmic spiral is a typical search pattern used to cover large distances without having to constantly change the direction of aircraft travel and with only small adjustments in airspeed and bank angle, depending on ambient wind conditions. Starting from the last known location of a target (e.g., a lost person) the aircraft can systematically expand its search radius without the risk of revisiting terrain or missing the target. Figure 8 shows that, as for other motion primitives, the spline representation of the logarithmic spiral is very closely approximated by the control polygon after two midpoint refinements (CP2).

Table 8 displays the same information as the previous tables for the logarithmic spiral. As expected, flight dynamics-based constraints limit the speed at which the spiral can be flown, given the allowable cross-track error metrics. The table shows that violations are incurred for all control polygons at speeds of 30 knots and above, indicating that this waypoint spacing method is not suitable recommended for higher speeds. The results show why the logarithmic spiral is not typically flown with waypoints operationally.

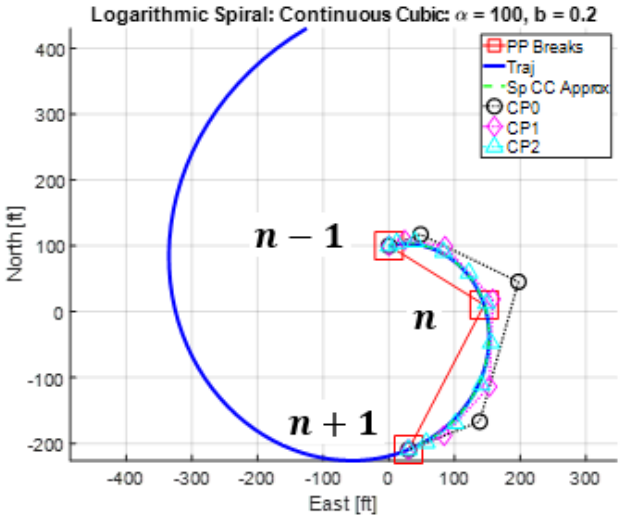

Figure 8. Cubic piecewise-polynomial spline approximation (SP CC Approx) of a logarithmic spiral trajectory (Traj), shown with knots (PP Breaks) and control points of successive midpoint refinements of the control polygon (CP0, CP1, CP2).

Summary of Results for Motion Primitives

As demonstrated in the previous subsections, the proposed algorithm can be used to define a feasible waypoint sequence that follows a complicated trajectory while maintaining legacy guidance algorithms and not requiring an excessive number of waypoints. For the sinusoid, semicircle, and exponential motion primitives, the control polygon refinements generally yield waypoint spacings that do not violate flight dynamics-based constraints at speeds required for the corresponding MTEs (the Slalom, Pirouette, and Emergency Stop). Cases where violations are incurred, and thus the desired MTE would not be flyable with the computed waypoint spacing, can be identified beforehand using our methodology. Because the algorithm was developed based on the simplified wind-axes flight model, which does not include some real-world physical effects, we assessed its utility for waypoint guidance under the more realistic conditions and vehicle dynamics of a high-fidelity nonlinear simulation, as described in the next section.

NONLINEAR SIMULATIONS

The core flight dynamics model used in the nonlinear simulation was GENHEL, a generic nonlinear single main-rotor tail-rotor (MR-TR) helicopter model, with the UH-60A parameters in Table 9. The basis of this nonlinear model and its implementation are discussed in (Refs. 18–20). APPENDIX C describes how the weight-normalized aerodynamic forces (n_{xw}, n_{yw}, n_{zw}) are extracted from this flight model implementation. We first compared the predictions of the simplified wind-axes model and the simulated aircraft trajectories for

Table 8. Logarithmic Spiral Motion Primitive: Waypoint Spacings for Constant Aircraft Speed

Entry Ground Track Angle: 90 deg

Bounded Area [ft²]		Grd Track Chg [deg]	Separation Distances [ft]											
BL 16822	CP0 (5pts) 9617	-68,53, 80,45	CP	116		220		165		51				
			15	28		30		24		29				
			30	112		119		97		117				
			40	199		211		173		207				
			60	448		476		388		466				
CP1	CP1 (7pts) 2924	-68,26,41, 44,39,28	CP	58		101	133	108		62	26			
			15	28		23	28	29		28	24			
			30	112		91	114	116		112	95			
			40	199		161	202	206		198	168			
			60	448		362	454	463		446	379			
CP2	CP2 (11pts) 868	-68,12,23, 21,20,23, 21,20,23, 16	CP	29	53	71	65	60	54	50	46	27	13	
			15	28	13	21	20	19	21	20	19	21	16	
			30	112	51	84	79	76	84	79	76	84	64	
			40	199	90	149	140	135	149	140	135	149	114	
			60	448	203	334	314	303	334	314	303	334	256	

RED indicates violation of flight dynamics-based constraints

Table 9. GENHEL UH-60A Example Parameters

Parameter	Variable	Value
Air Density	Atm.rho	1.225 kg/m^3
MR No. Blades	Helo.MR.nb	4
MR Speed	Helo.MR.omega	27 rad/s (258 RPM)
MR Chord	Helo.MR.c	1.73 ft
MR Lift-Curve Slope	Helo.MR.a	5.37 1/rad
MR Flap Hinge Offset	Helo.MR.e	1.25 ft
Primary Mission GW	Helo.Mass	16,260 lb

right-handed turn maneuvers, and then evaluated the simulated aircraft's trajectory-tracking performance for flight plans based on the Slalom and Exponential Stop MTEs. The performance of the simulated aircraft at flying these relatively simple maneuvers can provide insight into flight conditions under which our algorithm can generate feasible waypoint sequences for more complex maneuvers.

Trajectory Comparisons with Wind-Axes Model

The trajectories of the simplified wind-axes model and the nonlinear simulation were compared for two series of right-handed turn maneuvers, flown under the following conditions with different combinations of $(n_{xw,max}, n_{zw,max})$:

- *High-speed cruise:* Entry speed $V_e = 100$ knots, constant bank angle, with ball-centered turn coordination entry conditions
- *Low-speed cruise:* Entry speed $V_e = 20$ knots, changing ground track angle, fixed aircraft nose heading due north (0°)

For these two series of maneuvers, $(n_{xw,max}, n_{zw,max})$ were computed as shown in Tables 10 and 11. In the tables, $\dot{v}_{xc,max}$ and $\dot{v}_{yc,max}$ are the longitudinal and lateral maximum acceleration commands in the local level frame, and a_z is the accelerometer measurement along the z-axis of the body frame.

Table 10. Maximum Acceleration and Bank Angle Commands Converted to $(n_{xw,max}, n_{zw,max})$ For Increasingly Aggressive Maneuvers at High-Speed Cruise (100 kts), Ball-Centered Turn Coordination

$\dot{v}_{xc,max}$ $[g's]$	ϕ_{max} $@n_{zw,max}$	a_z $(T=W)$	$n_{xw,max}$ $(\dot{v}_{xc,max})$	$n_{zw,max}$ $(1/\cos\phi_{max})$
0.1	10°	1.0	0.1	1.0154
0.2	15°	1.0	0.2	1.0353
0.3	30°	1.0	0.3	1.1547
0.5	45°	1.0	0.5	1.4142
1.0	60°	1.0	1.0	2.0000

Table 11. Maximum Acceleration and Deceleration Commands Converted to $(n_{xw,max}, n_{zw,max})$ For Increasingly Aggressive Maneuvers at Low-Speed Cruise (20 kts); n_{yw} assumed negligible

$\dot{v}_{(x,y)c,max}$ $[g's]$	MR Thrust Tilt $@n_{zw,max}$	$n_{xw,max}$ $(\dot{v}_{xc,max})$	$n_{zw,max}$ $\sqrt{\dot{v}_{xc}^2+\dot{v}_{yc}^2+1.0}$
0.1	8.0°	0.1	1.0100
0.2	15.8°	0.2	1.0392
0.3	23.0°	0.3	1.0863
0.5	35.3°	0.5	1.2247
1.0	54.7°	1.0	1.7321

Note that in the low-speed cruise case (Table 11), $n_{zw,max}$ does not depend on the maximum bank angle command ϕ_{max}, since the aircraft heading is independent of ground track angle and turns do not have to be coordinated. (Moved text here:) In low-speed flight, the fundamental mechanism of rotorcraft maneuverability is different than in high-speed flight: although the aircraft is capable of making coordinated turns (either ball-centered or zero side-slip), its speed is typically well within bounds imposed by side-slip, structural, or critical wind azimuth limitations, and thus it does not need to perform a coordinated turn to change ground track angle. For example, the most efficient way for an aircraft in low-speed flight to achieve a 180° change in ground track angle is to decelerate, go through hover, and then re-establish its new velocity vector.

Figure 9 compares the planar trajectories of the nonlinear simulation and wind-axes model for the high-speed cruise case with $n_{xw,max} = -0.2$ g's and ϕ_{max} varied from 10° to 60°. The nonlinear simulation 6DOF autopilot was set up with a flight plan that included an OVERFLY waypoint at (0E ft, 500N ft) bearing north and a subsequent waypoint placed immediately behind bearing due south, which forced the autopilot guidance to command and hold a turn at the maximum bank angle to reacquire the next leg. As Figure 9 shows, there is close agreement between the trajectories of the wind-axes model and the nonlinear simulation at the benign bank angles of 10° and 15°, with growing discrepancies as the bank angle increases up to 60° (i.e., as the turn becomes more aggressive). This is in part because it is difficult for the aircraft autopilot to hold a perfectly coordinated turn at high bank angles while maintaining constant deceleration. Thus, a major limitation of any

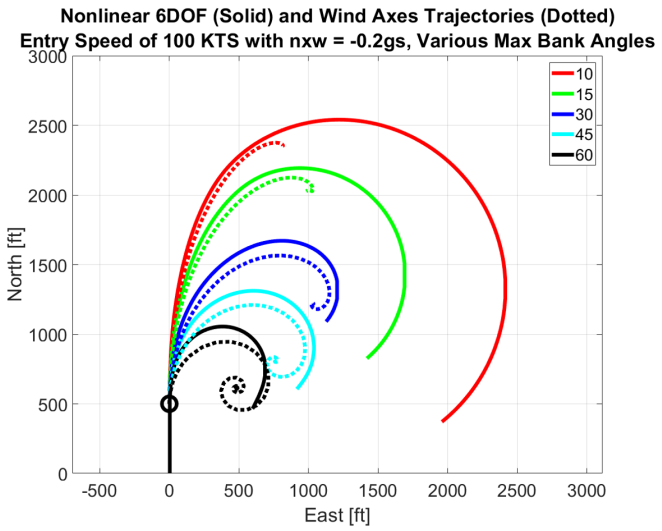

Figure 9. Trajectories of Nonlinear Simulation 6DOF Autopilot (Solid Lines) and Wind-Axes Model (Dotted Lines) for High-Speed Cruise (100 kts) OVERFLY with $n_{xw,max} = -0.2$ g's. Legend shows Maximum Bank Angle Command ϕ_{max} (deg).

automatic flight path guidance algorithm that is based on the simplified wind-axes model will be the ability of the vehicle control laws to maintain commanded attitudes and accelerations so that the model is valid.

Figure 10 shows trajectories of the nonlinear simulation for the low-speed cruise case with $n_{xw,max} = -0.2$ g's and $n_{zw,max} = 1.04$, in a scenario where the vehicle must make a 135° change in ground track angle via a Regular Change maneuver at different change distances Δ from the waypoint. At larger distances Δ of 25-75 ft, the vehicle performs a combined pitch and bank change to acquire the new track. At smaller distances Δ of 5-15 ft, the vehicle decelerates to a hover and backtracks to reacquire the new ground track. Both types of maneuvers are unique to the low-speed regime in that they are both uncoordinated, involving high amounts of side-slip (the nose of the aircraft is pointed due north with excursions $< \pm 1.0°$ during the entire maneuver) and considerable amounts of side-force. It was observed that for the same values of Δ, the vehicle never backtracks for ground track angle changes between -90° and 90° and always backtracks for changes of 180°.

Figure 10 illustrates that the minimum waypoint spacing algorithm must be used with discretion for low-speed flight, due to the types of maneuvers that aircraft can perform at low speeds which are not possible at high speeds. While low-speed uncoordinated turns can be approximated (albeit with considerable error) by the simplified wind-axes model even without the addition of the power equation constraint that we use to determine minimum waypoint spacing, low-speed backtracking maneuvers require a different application and integration of the flight model. In this case, the vehicle simply slows to a hover and resumes the flight plan at the current waypoint, so

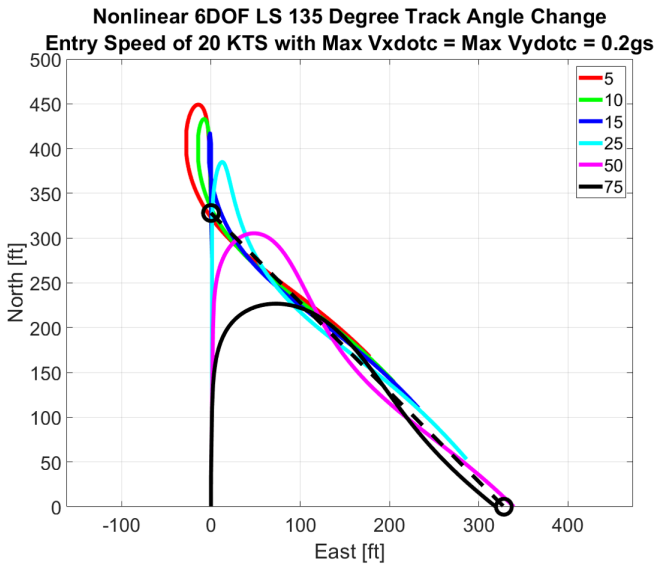

Figure 10. Trajectories of Nonlinear Simulation 6DOF Autopilot for Low-Speed Cruise (20 kts) REGULAR CHANGE, with $n_{xw,max} = -0.2$ g's, $n_{zw,max} = 1.04$ g's. Vehicle Performs 135° Change in Ground Track Angle. Legend shows Change Distance Δ (ft) Before Waypoint Due North of Origin.

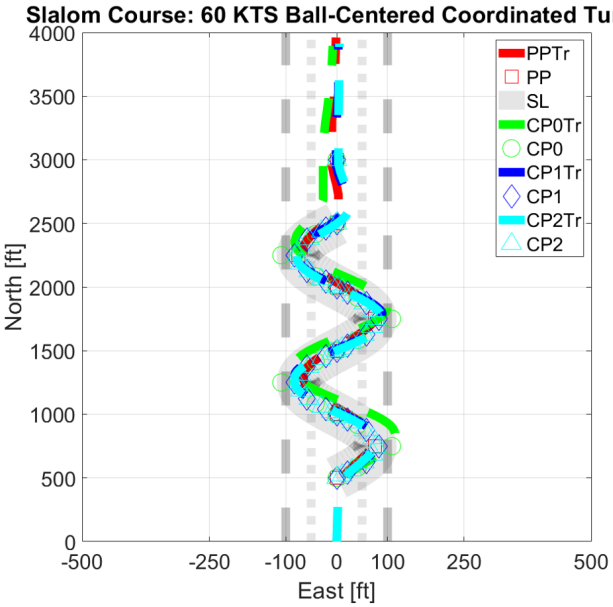

Figure 11. Slalom MTE course (SL) and nonlinear simulation trajectories (PPTr, CP0Tr, CP1Tr, CP2Tr) for flight plans with waypoints defined either as knots (PP) of a cubic piecewise-polynomial spline approximation of the MTE, or as control points of successive midpoint refinements of the control polygon (CP0, CP1, CP2). Vertical dashed lines indicate the maximum and minimum excursion from the centerline required with each turn.

there is no additional waypoint along the projecte flight-path to define. Moreover, the ability of the vehicle to slow to a hover and redirect its velocity vector without turning indicates that the constraint of a minimum distance to the next waypoint is not necessary for velocity vector reorientation in this case. For these reasons, the development of waypoint guidance algorithms that accommodate these types of low-speed maneuvers is a direction for future work.

Performance on Motion Primitives

Since the algorithm was found to be consistently applicable to maneuvers involving high-speed coordinated turns, with the aforementioned limitations on its application to low-speed maneuvers, comparisons of nonlinear simulation trajectories were performed for flight plans based on the Slalom and Emergency Stop MTEs at high entry speeds of 60 knots. The flight plan waypoints were defined either as knots (PP) of the spline approximation to the corresponding sinusoid or exponential motion primitive, or as the control points of successive midpoint refinements of the control polygon (CP0, CP1, CP2). Both MTEs were flown using the Regular Change option.

Figure 11 shows the simulated trajectories for the Slalom MTE, which was flown at constant speed and constant altitude as much as possible, in accordance with the MTE requirements. The change distance Δ was set to 196.8 ft, a constant approximation to the expression for Δ in Equation (6). The figure shows that the trajectories for all flight plans except CP0 meet the MTE requirement that cross-track error not exceed 100 ft from the centerline (some waypoints in CP0 slightly exceed this threshold). Successive midpoint refinements improve the aircraft's tracking error with respect to the spline trajectory, at the cost of adding waypoints in the flight plan. The particular use case will determine the necessary degree of tracking accuracy, and thus the appropriate waypoint sequence.

Table 12 shows the cumulative cross-track error, defined in Appendix A of (Ref. 10), and its maximum value and standard deviation for the CP0, CP1, and CP2 waypoint sequences based on the Slalom MTE. All values are normalized relative to the corresponding values for the PP sequence, which are scaled to 1. The errors were calculated by comparing the simulated vehicle trajectory to a sinusoid with the same wavelength and amplitude as the Slalom MTE. The CP0 sequence results in higher cross-track error values than the PP sequence, which is expected since the initial control polygon is a gross approximation of the spline trajectory. However, the refined control polygons CP1 and CP2 result in decreasing cross-track error relative to the PP sequence, at the cost of increasing numbers of waypoints in the flight plan.

Figure 12 shows the simulated trajectories for the Emergency Stop MTE, in which the aircraft decelerated to $V_f = 0$. The change distance Δ was manually tuned for each waypoint, due to the greater discrepancy between the wind-axes model and the simulated trajectories for this MTE at low speeds than for

Table 12. Slalom MTE Normalized Cross-Track Error (ft) Statistics for Waypoint (WP) Sequences Defined as Spline Knots (PP) or Control Points (CP0, CP1, CP2)

Sequence	Num WPs	Cumulative	Max	StdDev
PP	9	1.00	1.00	1.00
CP0	17	3.08	2.86	3.08
CP1	25	0.75	0.80	0.89
CP2	42	0.59	0.71	0.71

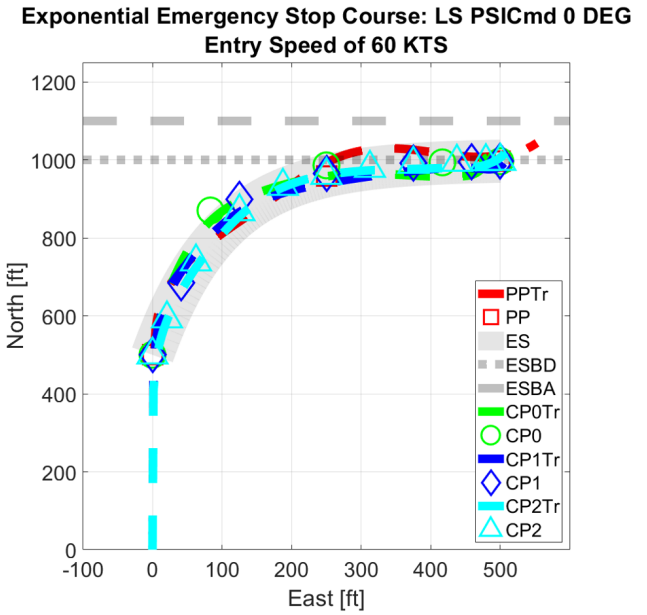

Figure 12. Emergency Stop MTE (ES) and nonlinear simulation trajectories (PPTr, CP0Tr, CP1Tr, CP2Tr) for flight plans with waypoints defined either as knots (PP) of a cubic piecewise-polynomial spline approximation of the MTE, or as control points of successive midpoint refinements of the control polygon (CP0, CP1, CP2). Horizontal dashed lines indicate the boundaries of desired (ESBD) and adequate (ESBA) maximum allowable forward travel.

the Slalom MTE at a constant cruise speed. Note, however, that as long as the aircraft's deceleration to hover during the Emergency Stop does not involve significant values of n_{yw}, the waypoint spacing criteria based on the wind-axes model should still be applicable. Table 13 shows the same cross-track error statistics as Table 12 for the Emergency Stop MTE. The table indicates that, as for the Slalom MTE, successive refinement of the control polygon results in decreasing cross-track error.

CONCLUSIONS

We have shown that control polygon midpoint refinement can be used to generate waypoints for autonomous flight path guidance driven by legacy waypoint guidance algorithms. The critical component of our proposed waypoint spacing algorithm is a robust stopping criterion for midpoint refinement

Table 13. Emergency Stop MTE Normalized Cross-Track Error (ft) Statistics for Waypoint (WP) Sequences Defined as Spline Knots (PP) or Control Points (CP0, CP1, CP2)

Sequence	Num WPs	Cumulative	Max	StdDev
PP	3	1.00	1.00	1.00
CP0	5	0.88	0.75	0.72
CP1	7	0.65	0.41	0.38
CP2	11	0.29	0.31	0.31

based on realistic vehicle flight dynamics. An augmented wind-axes model was used for real-time computation of the stopping criterion. The algorithm was tested on five motion primitives for different vehicle entry speeds to determine conditions under which the computed waypoint sequences do not violate flight dynamics-based constraints. In addition, the tracking performance of high-fidelity nonlinear simulations was assessed for a subset of waypoint sequences, and the wind-axes model was compared to the nonlinear simulations for particular turn maneuvers to identify discrepancies that could affect the applicability of the algorithm in practice.

The main results of this study are the following:

1. The simplified wind-axes flight dynamics model was very effective at defining minimum waypoint spacing for high-speed cruise conditions. This agrees with (Refs. 17, 23, 24), which demonstrate that at high cruise speeds, a point-mass approximation for helicopter flight dynamics is usually quite accurate.

2. For low-speed flight, the wind-axes flight dynamics model was not applicable for defining minimum waypoint spacing in many important use cases. This was evidenced by the nonlinear simulation results for the low-speed 135° change in ground track angle, where the vehicle can perform "backtracking" maneuvers that do not require minimum waypoint spacing.

3. Anticipating variations in vehicle speed at leg changes is important for capturing change distance variations over complex splines. This was shown by comparing the number and magnitude of violations of flight dynamics-based constraints for constant-speed versus decelerating trajectories that track the exponential Emergency Stop MTE.

Future work involves further exploration of the applicability of our waypoint spacing algorithm to the low-speed maneuverability regime and development of an alternative expression for minimum waypoint spacing that accounts for the ability of the vehicle to backtrack at low speeds and short change distances.

Author contacts:

Bryan Chu: bcchu1@asu.edu
Spring Berman: spring.berman@asu.edu

APPENDIX A

This appendix discusses the change of basis from a cubic polynomial spline in the monomial basis to the B-spline form by way of the Lagrange basis and Bernstein basis. It closely follows the derivation in (Ref. 22) that was performed on quadratic polynomial splines.

A cubic polynomial function of $x \in \mathbb{R}$ can be written in the monomial basis as follows, where $a_i \in \mathbb{R}$:

$$\begin{aligned} f(x) &= a_0 + a_1 x + a_2 x^2 + a_3 x^3 \\ &= \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix} \cdot \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix}^{\mathrm{T}} \equiv \mathbf{a}^{\mathrm{T}} \cdot \mathbf{m}(x) \end{aligned} \tag{24}$$

The Lagrange basis form of a cubic polynomial can be written as follows, where x_m and b_m, $m = 0,1,2,3$, are the node locations, coefficients of the basis set, and node indices:

$$\begin{aligned} f(x) = \; & b_0 l_0(x) + b_1 l_1(x) + b_2 l_2(x) + b_3 l_3(x) \\ &= \begin{bmatrix} b_0 & b_1 & b_2 & b_3 \end{bmatrix} \cdot \begin{bmatrix} l_0(x) & l_1(x) & l_2(x) & l_3(x) \end{bmatrix}^{\mathrm{T}} \\ &\equiv \mathbf{b}^{\mathrm{T}} \cdot \mathbf{l}(x), \\ & l_j(x) = \prod_{\substack{0 \le m \le 3 \\ m \ne j}} \frac{(x - x_m)}{(x_j - x_m)} \end{aligned} \tag{25}$$

The conversion of a cubic polynomial from the monomial basis to the Lagrange basis is readily accomplished by use of the Vandermonde matrix, $\mathbf{V}$. The Lagrange basis functions in $\mathbf{l}(x)$ can be found as follows:

$$\mathbf{l}(x) = (\mathbf{V}^{\mathrm{T}})^{-1} \cdot \mathbf{m}(x), \quad \mathbf{V}^{\mathrm{T}} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_0 & x_1 & x_2 & x_3 \\ x_0^2 & x_1^2 & x_2^2 & x_3^2 \\ x_0^3 & x_1^3 & x_2^3 & x_3^3 \end{bmatrix}, \tag{26}$$

and the Lagrange basis coefficients, $\mathbf{b}$, can thus be calculated as follows from the monomial coefficients, $\mathbf{a}$:

$$\begin{aligned} \mathbf{a}^{\mathrm{T}} \cdot \mathbf{m}(x) \; = \; \mathbf{b}^{\mathrm{T}} \cdot \mathbf{l}(x) \; &= \; \mathbf{b}^{\mathrm{T}} \cdot (\mathbf{V}^{\mathrm{T}})^{-1} \cdot \mathbf{m}(x) \\ \implies \mathbf{b}^{\mathrm{T}} \; &= \; \mathbf{a}^{\mathrm{T}} \cdot \mathbf{V}^{\mathrm{T}} \end{aligned} \tag{27}$$

The Bernstein basis, which is the basis for Bézier curves, is given by the following, where $\beta_i \in \mathbb{R}$ and $n = 3$:

$$\begin{aligned} f(x) &= \beta_0 B_{0,n}(x) + \beta_1 B_{1,n}(x) + \beta_2 B_{2,n}(x) + \beta_3 B_{3,n}(x) \\ B_{i,n}(x) &= \binom{n}{i} x^i (1-x)^{n-i}, \quad i = 0,1,2,\ldots,n \end{aligned} \tag{28}$$

The relationship between the Bernstein basis coefficients and the monomial basis coefficients can be found by equating the coefficients that multiply the same powers of x, similar to the procedure for determining the relationship between the Lagrange and monomial basis coefficients:

$$\begin{aligned} \beta_0 &= a_0, & \beta_2 &= a_2 + 2a_1 + 3a_0, \\ \beta_1 &= \frac{a_1}{3} + a_0, & \beta_3 &= a_3 + 3a_2 + 5a_1 + 7a_0. \end{aligned} \tag{29}$$

We can similarly obtain the relationship between the B-spline basis coefficients and the monomial basis coefficients. The B-spline representation of a cubic polynomial is as follows, where $C_0, C_1, C_2, C_3 \in \mathbb{R}$ are the spline control points, x_i are the knots of the spline, and $B_{i,3}$ are the B-spline basis functions:

$$\begin{aligned} f(x) &= C_0 B_{-3,3}(x) + C_1 B_{-2,3}(x) + C_2 B_{-1,3}(x) + C_3 B_{0,3}(x), \\ B_{i,d}(x) &= \frac{x - x_i}{x_{i+d} - x_i} B_{i,d-1}(x) + \frac{x_{i+d+1} - x}{x_{i+d+1} - x_{i+1}} B_{i+1,d-1}(x), \\ B_{i,0}(x) &= \begin{cases} 1 & x_i \le x \le x_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{30}$$

If we assume a uniform knot spacing of $h \equiv x_{i+1} - x_i$ for all $i = 0, \pm 1, \pm 2, \pm 3, \ldots$, then the basis functions for the cubic B-spline form can be simplified to (Ref. 26):

$$B_{i,3}(x) = \frac{1}{6h^3} \begin{cases} (x - x_{i-2})^3, & x_{i-2} \le x \le x_{i-1}, \\ -3(x - x_{i-1})^3 + 3h(x - x_{i-1})^2 + & \\ \quad 3h^2(x - x_{i-1}) + h^3, & x_{i-1} \le x \le x_i, \\ -3(x_{i+1} - x)^3 + 3h(x_{i+1} - x)^2 + & \\ \quad 3h^2(x_{i+1} - x) + h^3, & x_{i-1} \le x \le x_i, \\ (x_{i+2} - x)^3, & x_{i+1} \le x \le x_{i+2}, \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

Over a normalized interval spacing such that $h \equiv 1$, the B-spline basis functions can be further simplified to:

$$\begin{aligned} B_{-3,3}(x) &= \frac{1}{6}\left(-x^3 + 3x^2 - 3x + 1\right), \\ B_{-2,3}(x) &= \frac{1}{6}\left(3x^3 - 6x^2 + 4\right), \\ B_{-1,3}(x) &= \frac{1}{6}\left(-3x^3 + 3x^2 + 3x + 1\right), \\ B_{0,3}(x) &= \frac{1}{6}\left(x^3\right), \end{aligned} \tag{32}$$

and thus, Equation (30) can be written as a linear combination of the control points as:

$$f(x) = \frac{1}{6} \begin{bmatrix} x^3 & x^2 & x & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}. \tag{33}$$

Setting the expressions for $f(x)$ in Equations (24) and (33) equal to each other, we obtain the following relationships between the monomial basis coefficients and the B-spline basis

coefficients over the normalized interval $x \in \mathbb{R}$:

$$\begin{aligned} a_0 &= \frac{1}{6}\left(C_0 + 4C_1 + C_2\right), \\ a_1 &= \frac{1}{6}\left(-3C_0 + 3C_2\right), \\ a_2 &= \frac{1}{6}\left(3C_0 - 6C_1 + 3C_2\right), \\ a_3 &= \frac{1}{6}\left(-C_0 + 3C_1 - 3C_2 + C_3\right). \end{aligned} \tag{34}$$

This set of linear equations can be easily inverted to find the control point locations in terms of the monomial basis coefficients.

APPENDIX B

This appendix summarizes the derivation of the wind-axes flight model, Equation (2), including the necessary simplifications. The derivation closely follows that in (Ref. 23), with some further developments from (Ref. 17). The following notation is used:

$\mathbf{F}_w$ = Total force on vehicle in wind-axes frame
m = Vehicle mass
$\mathbf{a}_w$ = Vehicle acceleration vector in wind-axes frame
$\mathbf{V}_w$ = Vehicle velocity vector in wind-axes frame
$\mathbf{(i, j, k)}_w$ = Unit vectors along axes of wind-axes frame
p_w = Vehicle body roll rate in wind-axes frame
q_w = Vehicle body pitch rate in wind-axes frame
r_w = Vehicle body yaw rate in wind-axes frame
$\omega_w = [p_w \; q_w \; r_w]^T$
ϕ_w = Wind-axes bank angle about $\mathbf{V_w}$
γ = Wind-axes velocity flight path angle
χ = Wind-axes velocity ground track angle

The equations of motion for an aircraft relative to a flat, non-rotating Earth in the wind-axes frame are derived from Newton's second law of motion:

$$\mathbf{F}_w = m\mathbf{a}_w = m\frac{d\mathbf{V}_w}{dt} \tag{35}$$

Defining $V = ||\mathbf{V}_w||$, the rate of change of the vehicle velocity vector relative to the wind-axes frame is given by:

$$\begin{aligned} \frac{d\mathbf{V}_w}{dt} &= \frac{d(\mathbf{i}_w V)}{dt} = \mathbf{i}_w\dot{V} + \frac{d\mathbf{i}_w}{dt}V \\ &= \mathbf{i}_w\dot{V} + (\omega_w \times \mathbf{i}_w)V \\ &= \mathbf{i}_w\dot{V} + (\mathbf{j}_w r_w - \mathbf{k}_w q_w)V \end{aligned} \tag{36}$$

The components of the sum of the forces in the wind-axes frame, where the gravitational acceleration vector is expressed in this frame, can then be written as:

$$\begin{aligned} \sum F_{x_w} &= mgn_{xw} - mg\sin\gamma = m\dot{V} \\ \sum F_{y_w} &= mgn_{yw} = mVr_w \\ \sum F_{z_w} &= mgn_{zw} - mg\cos\gamma = -mVq_w \end{aligned} \tag{37}$$

The angular rates in the wind-axes frame can be related to the reference rates in the inertial frame as follows:

$$\begin{bmatrix} p_w \\ q_w \\ r_w \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\gamma \\ 0 & \cos\phi_w & \sin\phi_w\cos\phi_w \\ 0 & -\sin\phi_w & \cos\phi_w\cos\gamma \end{bmatrix} \begin{bmatrix} \dot{\phi}_w \\ \dot{\gamma} \\ \dot{\chi} \end{bmatrix} \tag{38}$$

Then, from Equation (37), the components of the vehicle's acceleration in the wind-axes frame are:

$$\begin{aligned} a_{x_w} &= \dot{V} \\ a_{y_w} &= Vr_w = V(-\dot{\gamma}\sin\phi_w + \dot{\chi}\cos\phi_w\cos\gamma) \\ a_{z_w} &= -Vq_w = -V(\dot{\gamma}\cos\phi_w + \dot{\chi}\sin\phi_w\cos\phi_w) \end{aligned} \tag{39}$$

Combining Equations (37) and (39), we arrive at:

$$\begin{aligned} \dot{V} + g\sin\gamma &= gn_{xw} \\ V\dot{\gamma} + g\cos\gamma &= gn_{zw}\cos\phi_w - gn_{yw}\sin\phi_w \\ V\dot{\chi}\cos\gamma &= gn_{zw}\sin\phi_w + gn_{yw}\cos\phi_w \end{aligned} \tag{40}$$

Under the assumption that the side-force term n_{yw} is zero, these are the wind-axes equations of motion derived in (Ref. 17) for a single main-rotor tail-rotor helicopter at low speed. The side-force term n_{yw} is zero for tilt-rotors and tandem rotors, and for fixed-wing aircraft that are flying zero sideslip or ball-centered. For single main-rotor tail-rotor helicopters, it is true in general that n_{yw} is zero or negligible when flying ball-centered and performing coordinated turns. Even under steady trimmed conditions, where there is significant sideslip, the wind-axes model in Equation (40) still applies. However, for low-speed dynamic maneuvers where significant changes in sideslip occur, n_{yw} should not be approximated as zero.

Setting $n_{yw} = 0$ and assuming zero flight path angle and zero climb rate ($\gamma = 0, \dot{\gamma} = 0$) in Equation (40), we obtain:

$$\begin{aligned} \dot{V} &= gn_{xw} \\ g &= gn_{zw}\cos\phi_w \\ V\dot{\chi} &= gn_{zw}\sin\phi_w \end{aligned} \tag{41}$$

Finally, the last two equations of Equation (41) can be combined by eliminating the dependence on ϕ_w:

$$\begin{aligned} \cos\phi_w = \frac{1}{n_{zw}} \;\Rightarrow\; \sin\phi_w &= \pm\frac{\sqrt{n_{zw}^2 - 1}}{n_{zw}} \\ \Rightarrow\; \dot{\chi} = \frac{gn_{zw}}{V}\sin\phi_w &= \pm\frac{g}{V}\sqrt{n_{zw}^2 - 1} \end{aligned} \tag{42}$$

The expressions for $\dot{V}$ in Equation (41) and $\dot{\chi}$ in Equation (42) comprise the wind-axes model in Equation (2). Note that the sign of $\dot{\chi}$ depends on the value of ϕ_w, which is positive when the aircraft is banking clockwise about $\mathbf{V}_w$ and negative when it is banking counterclockwise.

APPENDIX C

This appendix discusses the computation of the resultant weight-normalized aerodynamic forces along, across, and normal to the flight path, denoted by n_{xw}, n_{yw}, and n_{zw}, respectively, in the nonlinear 6DOF GENHEL rotorcraft simulation. Note that these forces are not to be confused with the body accelerations, as there are many cases in VTOL rotorcraft flight where they are not the same.

The nonlinear 6DOF equations of motion for an aircraft relative to a flat, non-rotating Earth in an inertial reference frame are given by (Ref. 27, Chapter 3.2):

$$\begin{aligned}
\dot{u} &= F_x/m - g\sin\theta + rv - qw \\
\dot{v} &= F_y/m + g\sin\phi\cos\theta - ru + pw \\
\dot{w} &= F_z/m + g\cos\phi\cos\theta + qu - pv
\end{aligned}$$

$$\begin{aligned}
\dot{p} = {} & (I_{zz}M_x + I_{xz}M_z - \{I_{xz}(I_{yy} - I_{xx} - I_{zz})p + \\
& [I_{xz}^2 + I_{zz}(I_{zz} - I_{yy})]r\}q)/(I_{xx}I_{zz} - I_{xz}^2) \\
\dot{q} = {} & [M_y - (I_{xx} - I_{zz})pr - I_{xz}(p^2 - r^2)]/I_{yy} \\
\dot{r} = {} & (I_{xz}M_x + I_{xx}M_z + \{I_{xz}(I_{yy} - I_{xx} - I_{zz})r + \\
& [I_{xz}^2 + I_{xx}(I_{xx} - I_{yy})]p\}q)/(I_{xx}I_{zz} - I_{xz}^2)
\end{aligned} \tag{43}$$

The transformation matrix $L_{\phi\theta\psi}$ from the body-fixed frame to the local-level frame is defined as follows, where $c()$ and $s()$ represent the cos() and sin() functions, respectively:

$$L_{\phi\theta\psi} = \begin{bmatrix} c(\psi)c(\theta) & s(\psi)s(\theta) & -s(\theta) \\ c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & c(\phi) & s(\phi)c(\phi) \\ 0 & -s(\phi) & c(\phi)c(\theta)c(\phi) \end{bmatrix} \tag{44}$$

Given the translational velocities (u, v, w) from the solution of Equation (43), the local level velocities in the Earth inertial frame can be computed using $L_{\phi\theta\psi}$:

$$\begin{bmatrix} V_{EX} \\ V_{EY} \\ V_{EZ} \end{bmatrix} = L_{\phi\theta\psi} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{45}$$

The ground track angle χ, measured clockwise from true north, can be computed as:

$$\chi = \tan^{-1}\left(\frac{V_{EY}}{V_{EX}}\right) \tag{46}$$

The local level accelerations can be computed by multiplying $L_{\phi\theta\psi}$ by the accelerations in the body-fixed frame:

$$\begin{bmatrix} \dot{V}_{EX} \\ \dot{V}_{EY} \\ \dot{V}_{EZ} \end{bmatrix} = L_{\phi\theta\psi} \begin{bmatrix} F_x/m \\ F_y/m \\ F_z/m \end{bmatrix} \tag{47}$$

The weight-normalized forces along and across the flight path can now be calculated as:

$$\begin{aligned}
n_{xw} &= \dot{V}_{EX}\cos\chi - \dot{V}_{EY}\sin\chi \\
n_{yw} &= \dot{V}_{EY}\cos\chi + \dot{V}_{EX}\sin\chi
\end{aligned} \tag{48}$$

Assuming that there is no rotor flapping relative to the rotor mast and that the mast is perpendicular to the nose of the aircraft, the weight-normalized force normal to the flight path is given by:

$$n_{zw} = F_z/m \tag{49}$$

In typical dynamic and unsteady maneuvering flight, the forces n_{xw}, n_{yw}, and n_{zw} are not constant. Special cases such as turning at a constant bank angle and slowing to a desired speed at a constant deceleration are exceptions that are used in this study; these maneuvers require tight control of vehicle attitudes in order to maintain constant n_{xw}, n_{yw}, and n_{zw}.

REFERENCES

1. Sujit, P. B., Saripalli, S., and Sousa, J. B., "An Evaluation of UAV Path Following Algorithms," European Control Conference, Zurich, Switzerland, 2013.

2. Rubí, B., Perez, R., and Morcego, B., "A Survey of Path Following Control Strategies for UAVs Focused on Quadrotors," *Journal of Intelligent & Robotic Systems*, Vol. 98, 2020, pp. 241–265.

3. Nelson, D., Barber, B., McLain, T., and Beard, R., "Vector Field Path Following for Small Unmanned Air Vehicles," American Control Conference, Minneapolis, MN, 2006.

4. Park, S., Deyst, J., and How, J., "Performance and Lyapunov Stability of a Nonlinear Path-Following Guidance Method," *Journal of Guidance, Control, and Dynamics*, Vol. 30, (6), 2007, pp. 684–686.

5. Chen, Y., Zeng, W., and Wu, Y., "Vector Field-Based Guidance for Planar Smooth Path-Following of a Small Unmanned Helicopter," *Hindawi: Journal of Robotics*, 2023, pp. 684–686.

6. Curry, R., Lizarraga, M., Mairs, B., and Elkaim, G. H., "L2+, An Improved Line of Sight Guidance Law for UAVs," American Control Conference, Washington, DC, 2013.

7. Cho, N., Kim, Y., and Park, S., "Nonlinear Differential Geometric Path-Following Guidance Law," *Journal of Guidance, Control, and Dynamics*, Vol. 38, (12), 2015.

8. Furieri, L., Stastny, T., Marconi, L., Siegwart, R., and Gilitschensiki, I., "Gone with the Wind: Nonlinear Guidance for Small Fixed-Wing Aircraft in Arbitrarily Strong Windfields," American Control Conference, Seattle, WA, 2017.

9. Chen, Q., and Lu, Y., "An Improved Nonlinear Path-Following Method with On-line Transition Trajectory for Fixed-Wing Unmanned Aerial Vehicles," *International Journal of Advanced Robotic Systems*, Vol. 19, (3), 2022.

10. Chu, B., Keller, J., and Berman, S., "Trajectory Waypoint Spacing for Spline-Based Flight Plans," Vertical Flight Society 6th Decennial Aeromechanics Specialists' Conference, Santa Clara, CA, February 2024.

11. Paduano, J., Wissler, J., Drozeski, G., *et al.*, "TALOS: An Unmanned Cargo Delivery System for Rotorcraft Landing to Unprepared Sites," American Helicopter Society 71st Annual Forum, Virginia Beach, VA, 2015.

12. Petit, P., J., W., B., F., and Greiser, S., "Waypoint Based Online Trajectory Generation and Following Control for the ACT/FHS," AIAA SciTech Forum, 2019.

13. Takahashi, M. D., Fujizawa, B. T., Lusardi, J. A., *et al.*, "Comparison of Autonomous Flight Control Performance Between Partial- and Full-Authority Helicopters," *Journal of Guidance, Control, and Dynamics*, Vol. 45, (5), February 2022, pp. 885–901.

14. deBoor, C., *A Practical Guide to Splines*, Springer-Verlag, New York, NY, 2001.

15. Lane, J. M., and Riesenfeld, R. F., "A Theoretical Development for the Computer Generation and Display of Piecewise Polynomial Surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, (1), January 1980, pp. 35–46.

16. Mork, H., "Spline Approximation Via the Control Polygon," M.S. Thesis, University of Oslo, 2012.

17. Curtiss, H. C., and Price, G., "Studies of Rotorcraft Agility and Maneuverability," Tenth European Rotorcraft Forum, The Hague, The Netherlands, August 1984.

18. Hilbert, K. B., "A Mathematical Model of the UH-60 Helicopter," NASA TM 85890, 1984.

19. Talbot, P. D., Tinling, B. E., Decker, W. A., and Chen, R. T. N., "A Mathematical Model of a Single Main Rotor Helicopter for Piloted Simulation," NASA TM 84281, 1982.

20. Orhan, E. H., "Generic Nonlinear Helicopter Model," https://www.mathworks.com/matlabcentral/fileexchange/72828-generic-nonlinear-helicopter-model, MATLAB Central File Exchange, March 10, 2024.

21. Ivler, C. M., Russell, K., Gong, A., *et al.*, "Toward a UAS Handling Qualities Specification: Development of UAS-Specific MTEs," Vertical Flight Society 78th Annual Forum, Fort Worth, TX, 2022.

22. Ezhov, N., Neitzel, F., and Petrovic, S., "Spline Approximation, Part 2: From Polynomials in the Monomial Basis to B-splines - A Derivation," *Mathematics*, Vol. 9, (18), 2021, pp. 2198.

23. Wood, T. L., *et al.*, "Maneuver Criteria Evaluation Program," USAAAMRDL Technical Report 74-32, 1974.

24. Wells, C. D., and Wood, T. L., "Maneuverability - Theory and Application," *Journal of the American Helicopter Society*, Vol. 18, (1), January 1973.

25. "ADS-33E-PRF, Aeronautical Design Standard Performance Specification Handling Qualities Requirements for Military Rotorcraft," U.S. Army Aviation and Missile Command, Aviation Engineering Directorate, Redstone Arsenal, Alabama, 2000.

26. Munguia, M., "Cubic B-spline: Derivation with two applications," M.S. Thesis, University of Texas-Pan American, 2014.

27. Stengel, R. F., *Flight Dynamics*, Princeton University Press, Princeton, NJ, 2005.