

# Design and Experimental Validation of UAV Control Laws – 3D Spline-Path-Following and Easy-Handling Remote Control

Nicolas Sedlmair, Julian Theis, and Frank Thielecke

**Abstract** A complete flight control architecture with two different operating modes is developed for a 24.6 kg UAV. The first control mode provides easy-handling of the UAV for a remote pilot from the ground. All relevant control loops are designed using loopshaping techniques and gain-scheduling over airspeed. Comprehensive details of the model-based design procedure are given. The second mode provides 3D path-following capabilities using cubic spline segments between specified waypoints. A way of calculating a virtual target point on the splines is introduced with a focus on practically relevant issues such as switching between different spline segments. A nonlinear guidance law from the literature is implemented. Experimental validation of both control modes is performed in several flight tests, showing high-performance in real-world conditions.

## 1 Introduction

This paper develops a complete flight control architecture with two different operating modes for a remotely-piloted research unmanned aerial vehicle (UAV) with a mass of 24.6 kg. Comprehensive details about the design of all relevant control loops using loopshaping techniques and gain-scheduling on airspeed are provided. Further, practically important protection functions are included. The first control mode provides easy-handling of the UAV for a pilot from the ground. Such a control mode is relevant to increase safety and reduce training effort for UAV pilots. A similar control mode is, e. g., provided with the PX4 Flight Stack that is part of the open source project Dronecode [6].

---

Nicolas Sedlmair · Julian Theis · Frank Thielecke  
Institute of Aircraft Systems Engineering, Hamburg University of Technology, Nesspriel 5,  
21129 Hamburg, Germany, e-mail: nicolas.sedlmair@tuhh.de, e-mail: julian.theis@tuhh.de, e-mail: frank.thielecke@tuhh.de

The second mode provides 3D path-following capabilities using cubic spline segments between specified waypoints. Path-following is a key enabler for increasing autonomy of unmanned aerial vehicles (UAV) and has been extensively studied in the literature, see e. g. [27] for a survey. Recent approaches use nonlinear methods which, according to [2], can be divided into three categories. These categories are *vector-field-based approaches* (e. g., [15, 17, 18]), *error-regulation-based approaches* (e. g., [25]) and *virtual-target-following approaches* (e. g., [2, 16, 22]). In vector field methods, the aircraft is steered towards a desired path along a pre-defined vector field. Error-based methods directly operate on a previously defined error such as the distance from the desired path or heading error. Virtual-target-following uses a geometric solution, forcing the aircraft towards a moving point on the path. Path-following algorithms are often studied in 2D-applications. Corresponding simulation studies are found, e. g., in [12, 15, 18, 25, 27]. There are also several reported experimental validations of UAVs following 2D-paths in flight tests, e. g. [16, 17, 21, 20, 23]. In all flight tests, geometrically simple paths such as straight line segments, arcs, and combinations were used. The three dimensional path-following problem was recently addressed in [2] and investigated in simulations using cubic splines. Cubic splines can resolve very complex spatial curves while providing continuous curvature, which is in contrast to, e. g., Dubins paths [7]. Thus, it is in principle possible to select paths that satisfy kinematic constraints. A flight test with a small-scale 2.3 kg remotely-piloted aircraft flying aerobatic maneuvers is reported in [24]. The present paper uses the guidance law introduced in [24] and further provides a way of calculating a virtual target point on 3D splines with a focus on practically relevant issues such as switching between different spline segments. Experimental validation of both control modes is performed in several flight tests and proves high-performance in real-world conditions.

The paper is structured as follows. Section 2 introduces the UAV and shows aerodynamic parameter identification results. All control systems are developed in Section 3. Section 4 considers the path-following problem and Section 5 provides the experimental validation.

## 2 Scaled Test Aircraft ULTRA-Extra

The ULTRA-Extra, depicted in Figure 1, is an unmanned replica of the aerobatic aircraft *Extra 330 ML* with a scale of 1:2.5. It is driven by an electric motor delivering a maximum power of 7.2 kW. Flight test experiments lasting up to 20 minutes are possible. The aircraft has a total mass of 24.6 kg and a wingspan of 3.10 m. It can either be controlled by a flight control computer or by a safety-pilot via remote control. For data distribution purposes an Ethernet-network as well as a controller area network (CAN) are included. Data recording is realized on a separate computer. In order to control and monitor flight test experiments, a telemetry link between the aircraft and the flight test engineer is realized. The overall concept and main parts of the setup are described in [13].



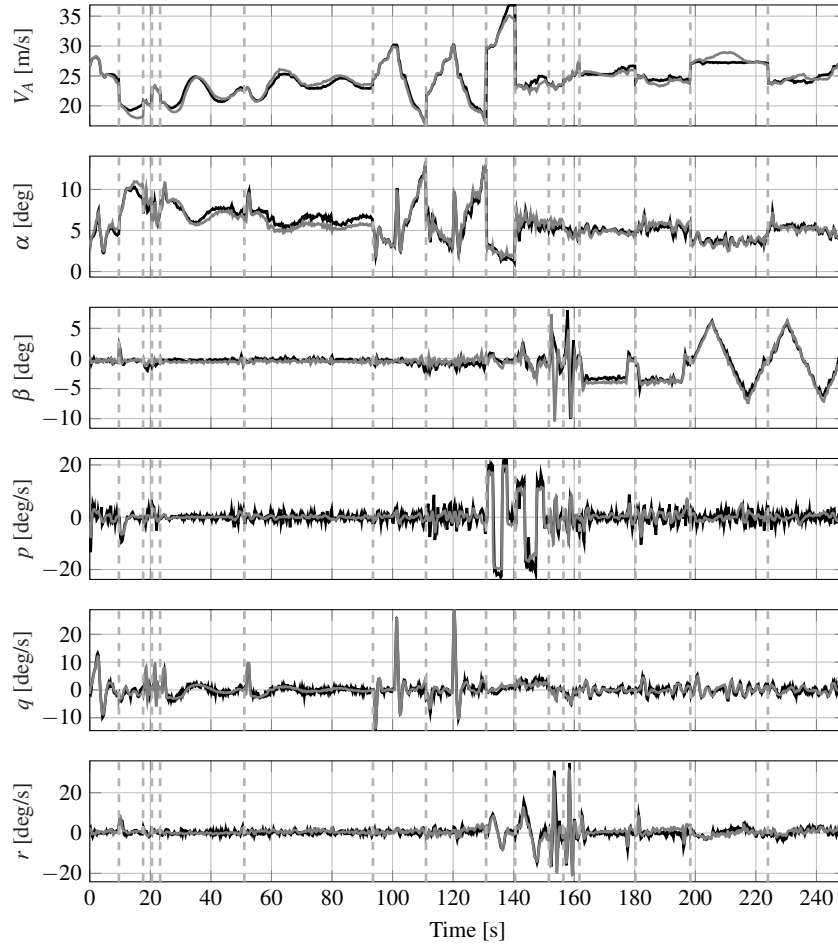
**Fig. 1** Flight test aircraft ULTRA-Extra.

## ***2.1 Hardware***

The onboard avionics comprise several sensors, computers, and radio equipment. The key component is a real-time computer that supports direct code compilation from MATLAB/Simulink. It serves as the flight control computer, hosts GNC applications, processes measurements and issues control outputs to the servos. An industry-grade high-quality navigation platform provides accelerations and rate measurements. GPS position and heading are determined using dual-antennas. Moreover, Differential GPS correction is provided by a third antenna on ground. This way, a position accuracy of up to 0.02 m is possible. Air data, such as airspeed, angle of attack, angle of sideslip, static air pressure, and air temperature are measured by a five-hole-probe. This sensor was developed in-house and calibrated in an extensive wind tunnel campaign [19]. An integrated laser sensor measures the altitude above ground.

## ***2.2 Parameter Identification***

For model-based development of GNC applications, a verified high fidelity simulation model is necessary. The ULTRA-Extra is modeled using a standard rigid body flight dynamics approach. The model comprises computations of forces and moments due to propulsion, the interaction between the landing gear and ground, and aerodynamics. An extensive flight test campaign was performed for the ULTRA-Extra in order to identify parameters for a one point gray-box aerodynamic model. A total of 24 flights were undertaken in which 148 identification maneuvers were performed. For this purpose, an in-house developed autopilot for aerodynamic parameter identification was used [14]. The output error method [11] was used to obtain the aerodynamic parameters from the data. Figure 2 shows representative results of the system identification.



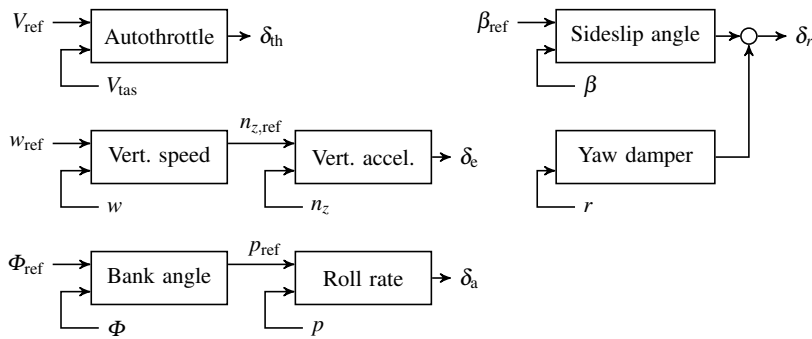
**Fig. 2** Validation of aerodynamic model, measurement (—), simulation (---), maneuver separation (- - -).

### 3 Control Systems Design

The flight control system should be applicable to fixed-wing remotely-piloted aircraft with a conventional configuration, i. e., with elevator, aileron, and rudder as control surfaces. It should further implement the required low-level functionalities for automatic path following and for automatic start and landing. A flight test engineer must be able to choose and activate different modes and must also have the possibility to control the UAV from the ground.

The classical way of controlling a remotely-piloted aircraft is directly through elevator (pitch), aileron (roll), rudder (yaw), and throttle (speed) inputs. The pilot steers the UAV from the ground and uses his or her sight as the primary “feedback

signal”, which is very different from how an onboard pilot controls an aircraft, relying mostly on sensed accelerations and visual attitude indicators. Estimating attitude from the ground, e. g., in order to keep a straight level flight, is a challenging task that requires hours of training. Providing an intuitive and “care-free” ground control mode is therefore an important aspect of the control system and drives the decision for the controller structure. Consequently, vertical speed and bank angle are selected as the primary controlled quantities. Hence, the pilot can directly control climb and sink rate of the aircraft and bank to turn. The idle state of the remote control corresponds to altitude hold straight flight. A conventional autothrottle maintains airspeed and turn coordination is implemented via sideslip angle feedback. An overview of the controller structure is given in Figure 3.



**Fig. 3** Basic flight control law overview.

### 3.1 Basic Flight Control Law

Linearization for straight level flight yields decoupled models for the longitudinal and lateral-directional dynamics. Since the ULTRA-Extra is an aerobatic aircraft, coupling effects between roll and yaw dynamics are negligible. Hence, control for all three axis can be addressed separately. A cascaded controller structure is chosen to exploit the principle of integrator chains (cf. e. g. [28]). Following this approach, linear single-input-single-output control problems with a clear physical interpretation are obtained. A qualified control design technique for such problems is classical loopshaping [10]. It can assure sufficiently high gain at low frequencies and low amplification at higher frequencies while providing robustness around the desired closed-loop bandwidth.

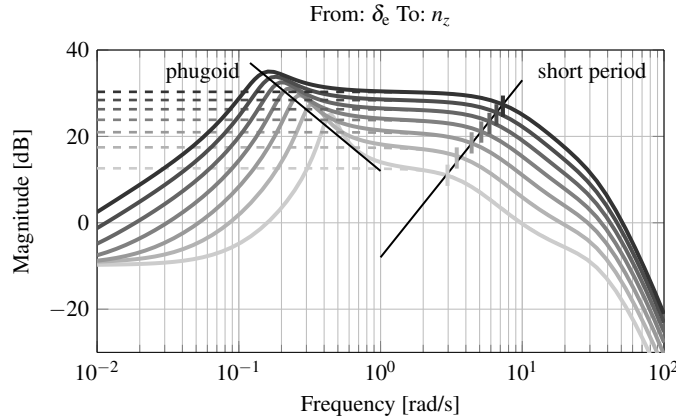
In order to obtain controllers suitable for the whole flight envelope, gain-scheduling over airspeed is applied. All controllers are designed using continuous time models. The implementation is discrete and includes anti-windup compensation and limitations on all controller outputs to ensure safe operation. Further, high-angle-of-

attack-protection is installed to avoid stall and a turn compensation crossfeed is implemented to support longitudinal control.

### 3.1.1 Longitudinal Control

The most inner longitudinal controller uses the elevator ( $\delta_e$ ) to track vertical acceleration ( $n_z$ ). Deviating from, e. g. [5] or [26], tracking the acceleration is preferred over pitch rate or angle-of-attack control. Doing so establishes an integrator chain to the vertical speed  $w$ , facilitating the design of the outer control loop.

Figure 4 shows the Bode plot of the linearized longitudinal dynamics ( $P_{n_z}(s) = n_z(s)/\delta_e(s)$ ) for different airspeeds  $V_A \in [20, \dots, 50]$  m/s. The plot shows apparent variations with airspeed, justifying gain-scheduling. Specifically, the short period frequency increases linearly with airspeed, while the phugoid frequency decreases. Further, the magnitude of the transfer function increases quadratically with airspeed. Within the considered airspeed range, the ULTRA-Extra exhibits a well-damped short period mode (damping ratio  $\zeta_{SP} > 0.7$ ), making a pitch damper loop unnecessary. At higher frequencies, servo dynamics ( $\omega_{act} \approx 34$  rad/s identified in experiments) and further known parasitic effects are visible. These parasitic effects include computational, actuator and sensor delays.



**Fig. 4** Longitudinal dynamics model,  $V=50$  m/s (—), 45 m/s (—), 40 m/s (—), 35 m/s (—), 30 m/s (—), 25 m/s (—), 20 m/s (—).

In order to design a controller, the model is reduced to the short period dynamics (dashed lines in Figure 4). The Bode plot proves accuracy of this model for frequencies above  $\omega = 1$  rad/s. Based on the reduced model, a proportional-integral controller  $C_{n_z}$  is designed to maximize bandwidth under the constraint of a phase margin of  $PM_{n_z} > 60^\circ$ . The control design leads to a controller zero in the immediate vicinity of the short period frequency (i.e. approximately linearly increasing

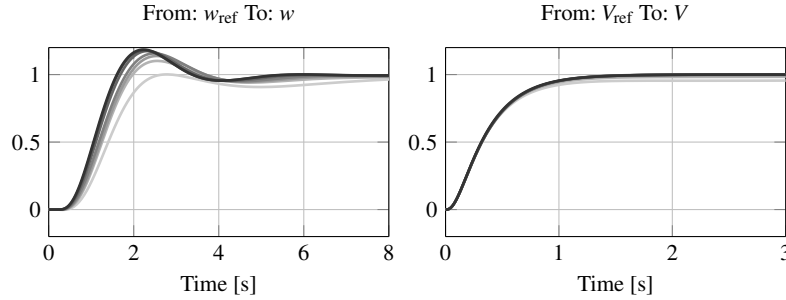
integral gain) and a proportional gain that reduces approximately quadratic with airspeed. The resulting closed-loop bandwidth over all considered airspeeds then varies from  $\omega_{nz} = 1.6$  rad/s at 20 m/s airspeed to 2.0 rad/s at 50 m/s.

With the  $n_z$  control loop closed, the vertical speed controller  $C_w$  can be selected as a proportional gain. Maximizing the control loop bandwidth under the constraint of a phase margin  $PM_w > 50^\circ$  produces a gain-schedule which increases approximately linear with airspeed. The closed-loop bandwidth varies from 0.8 rad/s at 20 m/s to 1.1 rad/s at 50 m/s. As the inner loop controls acceleration in the body-fixed frame, but the outer loop controls vertical speed with reference to earth, a small steady-state error of less than 4% results and is deemed acceptable. The controller output is a reference load factor ( $n_{z,\text{ref}}$ ), limited subject to current airspeed in order to avoid stall. This limit is adapted online from the model-based relation

$$n_{z,\text{max}} = \frac{\rho V_A^2 S}{2G} C_{L\alpha} \alpha_{\text{max}}, \quad (1)$$

using the known wing area  $S$ , current airspeed  $V_A$ , air density  $\rho$ , weight  $G$ , and angle-of-attack lift effectiveness  $C_{L\alpha}$  [1]. A maximum of  $\alpha_{\text{max}} = 12^\circ$  is chosen for the ULTRA-Extra.

The second longitudinal control loop encompasses an autothrottle to track a given airspeed as measured by the 5-hole-probe of the aircraft. A proportional-integral controller is designed to achieve a closed-loop bandwidth of 2.5 rad/s with a  $75^\circ$  phase margin. Figure 5 shows the closed-loop step responses of the longitudinal control loops over the complete airspeed envelope.

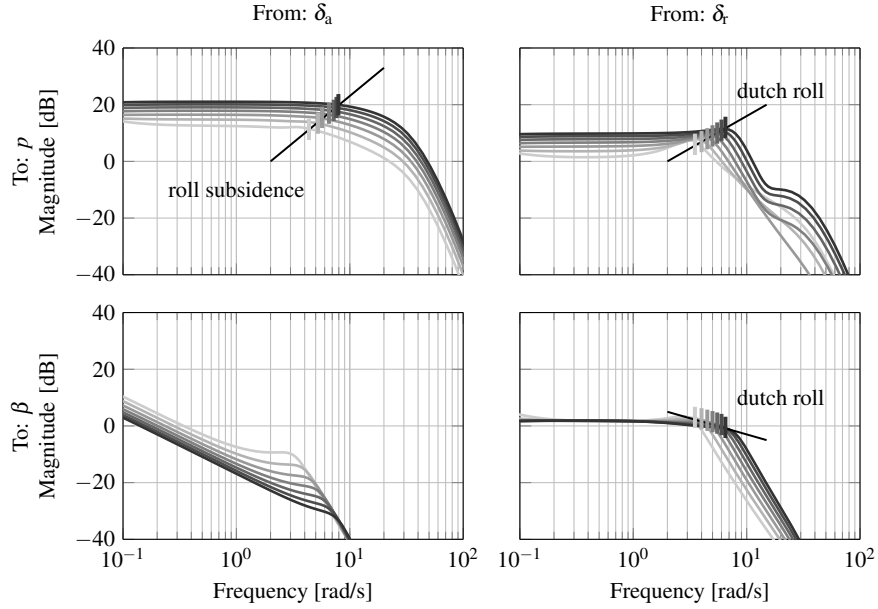


**Fig. 5** Closed-loop response to vertical speed and airspeed commands,  $V = 50$  m/s (—), 45 m/s (—), 40 m/s (---), 35 m/s (· · ·), 30 m/s (- · - ·), 25 m/s (- - -), 20 m/s (- - -).

### 3.1.2 Lateral-Directional Control

The rolling motion is controlled through aileron deflection ( $\delta_a$ ). A reduced model of the lateral dynamics is derived to apply loopshaping. It includes roll rate  $p$ , yaw rate  $r$ , velocity in  $y$ -direction ( $v$ ) (body-fixed), angle of sideslip  $\beta$ , and parasitic dy-

namics. The low roll-yaw-coupling and the variation of the dynamics with airspeed are apparent in the open-loop Bode plot shown in Figure 6.



**Fig. 6** Lateral-directional dynamics model,  $V = 50$  m/s (—), 45 m/s (---), 40 m/s (· · ·), 35 m/s (- · - ·), 30 m/s (— · —), 25 m/s (— · · —), 20 m/s (— · · · —).

Following the same integrator-chain strategy as in the longitudinal case, an inner loop is designed to track roll rate ( $p$ ) using proportional-integral control with a proportional outer control loop feeding back the bank angle ( $\Phi$ ). This architecture is slightly different from the commonly used proportional-integral bank angle control loop with proportional inner loop roll rate feedback.

The roll rate controller is designed to maximize bandwidth under a phase margin constraint of  $60^\circ$ . The integral gain is scheduled such that the controller zero remains in the vicinity of the roll subsidence mode and the proportional gain is decreased inversely proportional to airspeed squared. The result is an almost constant closed-loop bandwidth of  $\omega_p = 3.7$  rad/s for all considered airspeeds. The bank angle controller gain schedule is chosen to satisfy a  $60^\circ$  phase margin constraint, resulting in slightly increasing gain over airspeed. The closed-loop bandwidth is 1.4 rad/s at 20 m/s airspeed and 1.7 rad/s at 50 m/s. The body-fixed roll rate used in the inner feedback loop is not exactly the derivative of the bank angle  $\Phi$ , so a small steady-state error is again accepted. This control error could be reduced to zero by feeding back the inertial roll rate  $\frac{d}{dt}\Phi = p + \sin\Phi \tan\Theta q + \cos\Phi \tan\Theta r$  in the inner loop instead of the body-fixed roll rate  $p$ .

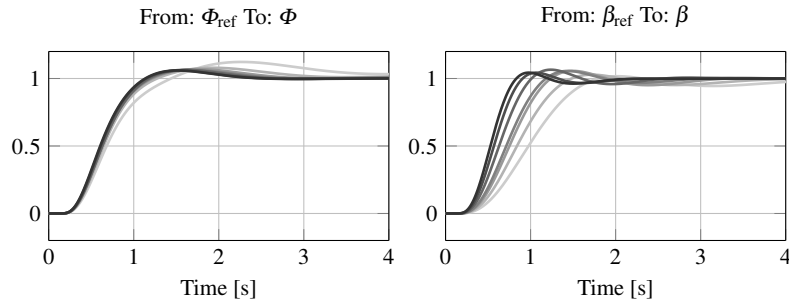


To avoid losing altitude as a result of banking the lift vector, the maximum bank angle is limited subject to current airspeed as given in Table 1.

**Table 1** Maximum bank angle command  $\Phi_{\text{cmd}}$  subject to current airspeed.

Airspeed $V_A$ [m/s]	20	25	30	35	40	45	50
$\Phi_{\text{max}}$ [deg]	30	45	45	50	50	50	50

The yaw damping is first augmented with washed-out yaw rate ( $r$ ) feedback to increase dutch roll damping to  $\zeta_{DR} > 0.4$ . Then, a proportional-integral controller is designed to track the sideslip angle ( $\beta$ ), measured by the aircraft's 5-hole probe. This control loop guarantees aerodynamically clean flight even in crosswind conditions. Further, decrab maneuvers and certain aerobatic flight maneuvers are easy to perform by directly controlling sideslip. This controller is also designed to maximize the closed-loop bandwidth such that a  $60^\circ$  phase margin constraint is satisfied. The resulting closed-loop bandwidth varies from  $\omega_\beta = 1.1$  rad/s at 20 m/s airspeed to 2 rad/s at 50 m/s airspeed.



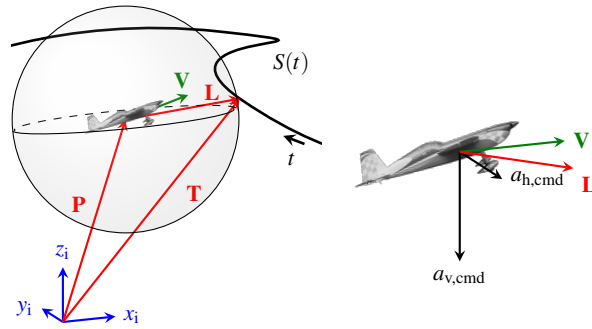
**Fig. 7** Closed-loop response to bank angle and sideslip angle commands,  $V = 50$  m/s (—), 45 m/s (---), 40 m/s (···), 35 m/s (-·-·-), 30 m/s (— — —), 25 m/s (- - - -), 20 m/s (— — —).

### 3.2 Non-Linear-Guidance-Law

While waypoint flight is easily achieved with the previously described autopilot functions (i. e., acquire heading and acquire altitude), path-following is a more challenging task. Path-following here refers to exactly following a given path in contrast to flying straight towards a given waypoint. Such capabilities are implemented using the Non-Linear-Guidance-Law (NLGL) introduced by [22][23]. The NLGL approach can be classified as a virtual target following technique. It is chosen because of its simplicity, its ability to compensate for unknown wind, and its model

independence. In this work, the original NLGL is enhanced in order to follow three dimensional paths.

Let the aircraft (position  $\vec{P}$  and velocity  $\vec{V}$  in an inertial reference frame) be in close proximity to the path  $\vec{S}(t)$ . Given a constant distance  $\|\vec{L}\|$ , a virtual target point  $\vec{T}$  on the path can be defined at the intersection of the path  $\vec{S}(t)$  and a *sphere* with radius  $R = \|\vec{L}\|$  around the aircraft. That is, the target point  $\vec{T}$  moves along the path as the UAV advances and  $\vec{L} = \vec{P} - \vec{T}$ . As shown in [22],  $\|\vec{L}\|$  is a tuning parameter that can be chosen, e. g. based on a stability analysis with a linear plant model. The guidance law computes the acceleration which is required to align the velocity vector  $\vec{V}$  with  $\vec{L}$ , i. e., it smoothly steers the aircraft on the given path. This is illustrated in Figures 8.



**Fig. 8** Target point definition and acceleration to rotate the velocity vector towards the target point.

When  $\vec{L}$  and  $\vec{V}$  are expressed in the flight-path coordinate system, the acceleration is calculated as

$$\vec{a}_{\text{cmd}} = \frac{2}{\|\vec{L}\|^2} (\vec{V} \times \vec{L}) \times \vec{V} = \begin{bmatrix} 0 \\ a_{\text{h,cmd}} \\ a_{\text{v,cmd}} \end{bmatrix}. \quad (2)$$

Thus,  $\vec{a}_{\text{cmd}}$  never has a component in the direction of flight. The acceleration commands are always normal to the current velocity vector and correspond to horizontal and vertical acceleration commands. These commands are fed as references to the basic flight control law described in Section 3.1. The horizontal acceleration  $a_{\text{h,cmd}}$  is translated to a bank angle reference

$$\Phi_{\text{cmd}} = \text{atan} \left( \frac{a_{\text{h,cmd}}}{g} \right), \quad (3)$$

where  $g$  is the gravitational acceleration. The vertical acceleration is transformed into the body-fixed coordinate system. According to [1] this leads to

$$n_{z,\text{cmd}} = -\sin(\chi - \Psi) \sin(\Theta - \gamma) \cos(\Phi) \frac{a_{h,\text{cmd}}}{g} + \cos(\Theta - \gamma) \cos(\Phi) \frac{a_{v,\text{cmd}}}{g}, \quad (4)$$

where  $\Psi$  is the current heading angle and  $\gamma$  is the flight path angle. In many cases,  $\chi - \Psi$  and  $\Theta - \gamma$  are very small and Equation (4) simplifies to

$$n_{z,\text{cmd}} = \cos(\Phi) \frac{a_{v,\text{cmd}}}{g}.$$

## 4 Spline Calculation and Path-Following Algorithm

The task of path-following is, in general, different from a simple waypoint flight. The latter usually connects consecutive waypoints by straight lines. A common solution for actual *path planning*, i. e., to define a segment between waypoints, is to create a Dubins path [7]. Both approaches generally lead to discontinuities in the path curvature, making it impossible for a UAV to exactly follow the prescribed path due to fundamental kinematic constraints. In this study, cubic splines are used to connect the waypoints, so that the resulting path is twice continuously differentiable. The waypoints are set manually in the form of a list of consecutive waypoints. Using the telemetry link between the aircraft and the ground control station, waypoint lists can be transferred during operation.

Each of the  $n$  waypoints is originally defined in terms of latitude, longitude, and elevation above the reference ellipsoid of the world geodetic system WGS84 [3]. A transformation into the Cartesian earth-centered-earth-fixed coordinate system (ECEF) is then performed and cubic splines

$$\vec{S}_k(t) = \begin{bmatrix} S_x^k(t) \\ S_y^k(t) \\ S_z^k(t) \end{bmatrix}, k \in \{1, \dots, n-1\} \quad (5)$$

are calculated using the algorithms stated in [8]. This calculation is done aboard the ULTRA-Extra once a waypoint list is provided or replaced. The polynomial coefficients are stored in matrices  $M_i^k \in \mathbb{R}^{(n-1) \times 4}$ ,  $i \in \{x, y, z\}$ ,  $k \in \{1, \dots, n-1\}$ . For the  $k$ th spline segment this leads to  $S_i^k(t) = M_i^k \cdot [1, t, t^2, t^3]^T$ . Every spline segment  $\vec{S}_k(t)$  is defined on the interval  $[0, t_{\max,k}] := \{t \in \mathbb{R} \mid 0 \leq t \leq t_{\max,k}\}$ , while the length  $t_{\max,k}$  can vary for each segment  $k$ . As the spline is fixed, the length  $t_{\max,k}$  of every segment can be computed by solving

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} S_x^k(t) - S_x^{k+1}(t=0) \\ S_y^k(t) - S_y^{k+1}(t=0) \\ S_z^k(t) - S_z^{k+1}(t=0) \end{bmatrix}. \quad (6)$$

### 4.1 Calculation of Target Point

To apply the NLGL as described in Section 3.2, Equation (2) needs to be evaluated. While the current UAV position and its velocity vector can be measured via the inertial navigation platform, the *look-ahead-vector*  $\vec{L}$ , and thus the target point  $\vec{T}$ , must be computed. This is done on the currently active spline segment preceding the currently active waypoint. If the distance to the active waypoint is less than a specified check distance, the point is considered visited and the next waypoint (and hence spline segment) is activated.

Assume that a reasonable radius  $R$  is chosen and that the Euclidean distance of the UAV to the spline is less than this radius. Such a situation is depicted in Figure 9a. The desired target point is one of the two intersection points of the sphere around the aircraft and the current spline segment  $\vec{S}_k(t)$ . Thus,  $\|\vec{S}_k(t) - \vec{P}\| = \|\vec{L}\|$  holds. This equation can be expanded and rearranged in order to obtain a polynomial of order six in  $t$  such that

$$0 = a_0 + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3 + a_4 \cdot t^4 + a_5 \cdot t^5 + a_6 \cdot t^6. \quad (7)$$

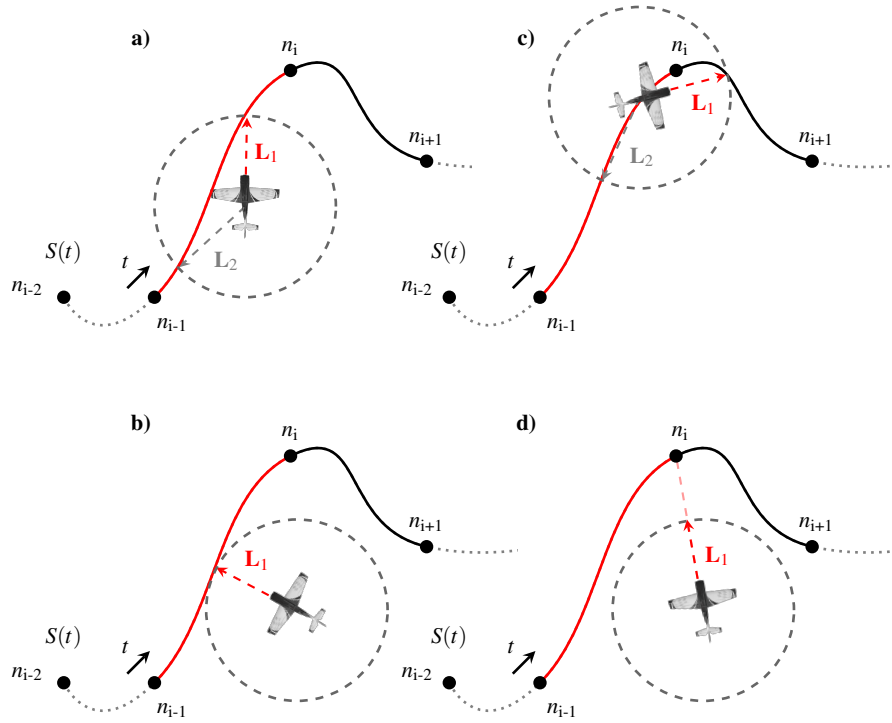
Each of the coefficients  $a_i, i \in \{0, \dots, 6\}$  of this polynomial can be calculated by an individual and fixed formula. In these formulae, only the known aircraft position, the selected radius of the sphere and the given spline parameters  $M_i^k$  are used. Then, a root search on the current spline segment  $\vec{S}_k(t)$  is done. Because of the formulation of the problem, a maximum of two real solutions  $\{t_{\text{sol}1}, t_{\text{sol}2}\}$  can be found. Since every segment has a finite length, it has a defined interval  $[0, t_{\text{max},k}] := \{t \in \mathbb{R} \mid 0 \leq t \leq t_{\text{max},k}\}$  on which the cubic polynomials  $S_i^k(t)$  are part of the spline.

In the situation depicted in Figure 9a two valid solutions exist. By choosing  $t_{\text{sol}} = \max\{t_{\text{sol}1}, t_{\text{sol}2}\}$ , the path will always be followed in the correct direction. Afterwards,

$$\vec{T} = \vec{S}_k(t_{\text{sol}}) = \begin{bmatrix} S_x^k(t_{\text{sol}}) \\ S_y^k(t_{\text{sol}}) \\ S_z^k(t_{\text{sol}}) \end{bmatrix} \quad (8)$$

yields the desired target point.

In a first implementation of the algorithm, the root search for the polynomial given in Equation (7) was done using the MATLAB routine `roots` which computes the eigenvalues of the companion matrix of the polynomial. Since the spline is given in ECEF coordinates, the companion matrix is numerically ill-conditioned. Solving Equation (7) was therefore unreliable and often led to wrong solutions. Shifting the origin of the coordinate system into, e. g., the first waypoint did not resolve the issue. In the final implementation of the target point calculation, the `fzero` algorithm of MATLAB is chosen [9]. This algorithm uses a combination of methods such as inverse quadratic interpolation, bisection, and the secant method [4].



**Fig. 9** Target point selection.

## 4.2 Target Point Selection

As an iterative method, the  $f_{\text{zero}}$  command requires an initial point  $t_{\text{init}}$  and computes a local solution in the vicinity of this initial point. Keeping the situation of Figure 9a in mind, it is not trivial to define an initial point leading to the correct solution  $t_{\text{sol}}$ . If  $t_{\text{init}}$  is chosen near the intersection point behind the aircraft, a wrong target point is obtained, guiding the aircraft in the wrong direction. To overcome this problem, both intersection points are calculated, meaning two initial points have to be defined. As long as no solution to Equation (7) is found, one initial point is chosen to be at the beginning of the current spline segment, i. e.  $t_{\text{init},1} = 0$ . In case of two valid solutions on the current segment, the maximum distance between them is exactly the diameter of the sphere around the aircraft. Hence,  $t_{\text{init},2} = t_{\text{init},1} + 2R$  is the second initial point for the root search algorithm. After the correct solution  $t_{\text{sol}}$  is computed for the first time, the initial points are updated to be  $t_{\text{init},1} = t_{\text{sol}}$  and  $t_{\text{init},2} = t_{\text{init},1} + 2R$ .

In a situation as depicted in Figure 9b, the distance between the spline and the aircraft is exactly  $R$ . Here, only one intersection point and thus only one valid solution can be found. The initial point is updated to be  $t_{\text{init},1} = t_{\text{sol}}$ . One time step later,

the distance between the spline and the aircraft is less than  $R$  and two solutions can be found. To assure the correct flight direction, again a second initial point with  $t_{\text{init},2} = t_{\text{init},1} + 2R$  for the root search is used.

Figure 9c shows a situation where the distance of the aircraft to the waypoint is larger than the check distance, but only one intersection point on the active spline segment is valid. The intersection point ahead of the UAV exceeds the permitted interval  $[0, t_{\text{max},k}]$  and is discarded. To avoid a turn in the wrong direction, Equation (7) needs to be evaluated also on the next spline segment  $\vec{S}_{k+1}$ .

In a situation in which the distance between the spline and the aircraft is larger than the diameter of the sphere, no solution to Equation (7) can be found, see Figure 9d. As evaluation of Equation (2) requires a look-ahead vector  $\vec{L}$ , the currently active waypoint is chosen as the target point in this case. Concurrently, the norm of  $\vec{L} = \vec{P} - \vec{T}$  is adjusted to the diameter of the sphere while retaining its direction. Once at least one intersection point between the spline segment and the sphere around the aircraft exists, the target point is updated again.

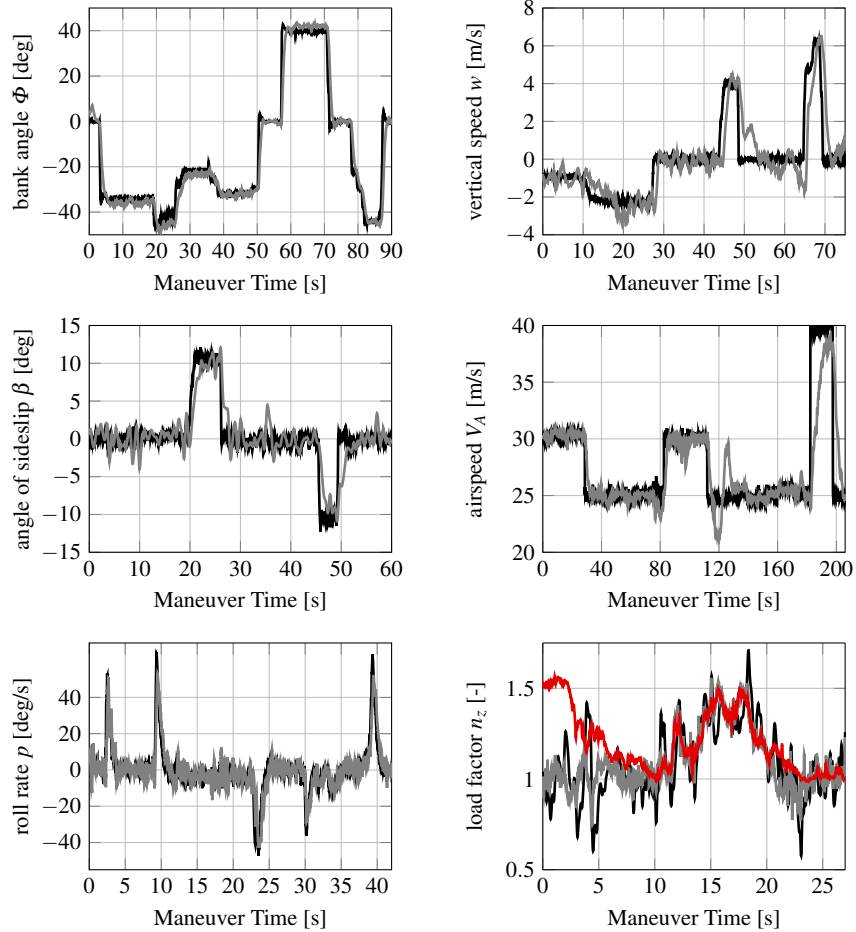
## 5 Flight Test Experiments

Prior to real flight test experiments, a comprehensive Hardware-In-The-Loop (HITL) test campaign was performed. Within this campaign, virtual test scenarios were undertaken. The goal was to validate the controllers as well as the path-planning and path-following algorithms described in the previous sections. These software modules were processed on the true hardware aboard the ULTRA-Extra. The tests confirmed the desired “easy-handling” of the aircraft in remote control mode and very good path following performance. Hence, real flight tests took place in July and August 2018. For this purpose, the Institute of Aircraft Systems Engineering possesses a ground control station which provides the necessary infrastructure to control and monitor flight test experiments [13].

### 5.1 Remote Control Mode

In order to validate the remote control mode, changing commands are issued using a commercial-off-the-shelf joystick and thrust lever. The commands are sent to the aircraft via telemetry link. Varying bank angles, climb and sink rates, airspeeds, angles of sideslip, and combinations of the aforementioned are commanded in several flights. During the campaign, the airspeed range from 20 m/s to 40 m/s was explored. Figure 10 shows exemplary results gathered in different flight tests. Each subplot in Figure 10 is to be interpreted independently of the others with time indicating the respective time of a particular flight experiment.

Tracking performance of the bank angle controller is evidently excellent, resulting in very quick responses without overshoot. The same is true for the vertical speed



**Fig. 10** Measurement data for remote control mode evaluation. Subplots show results from independent flight tests: command (—), value (—). The  $n_z$  plot additionally shows the upper limit  $n_{z,lim}$  (—) enforced by the angle-of-attack protection.

controller, although minor deviations are visible as a result of turbulence. All inner loop controllers, i. e., the angle of sideslip controller, the roll rate controller, and the load-factor controller track the issued commands tightly. Airspeed is also well controlled by the autothrottle. The only visible deviations occur in a pitch-up-pitch-down maneuver (at maneuver time 120 s) and during the last large step command, where saturation is encountered. Functionality of the angle-of-attack-protection is apparent in the load-factor data. The limit  $n_{z,lim}$  is adapted throughout the flight and bounds the  $n_z$  command. Even though the pilot demanded climb during the last quarter of the time slice, the load factor remained at  $n_z \approx 1$  to prevent stall.

In conclusion, the flight test campaign shows exceptionally good results for the remote control mode. The basic flight control law is stable throughout the flight envelope and consistently provides high performance. The controller is robust against external disturbances like wind. The desired “easy-handling” from the ground was confirmed by a total of three different pilots and the protections avoid critical situations. Therefore, all objectives are fully satisfied.

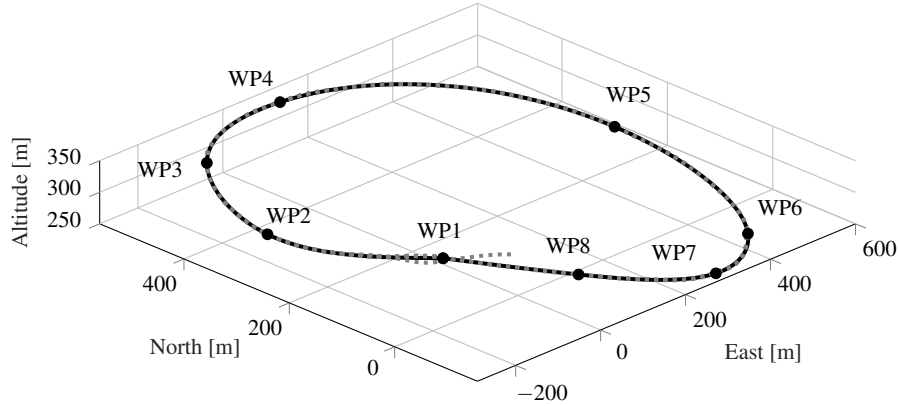
## 5.2 Path-Following Mode

For the purpose of validating the path planning and path following algorithms, different three dimensional paths were created before the start. Path-planning is done using an in-house developed software that embeds Google Earth. Within this software, waypoints can be placed while meeting legal requirements. The resulting cubic spline is calculated and displayed in Google Earth. The current position of the aircraft is also displayed such that the aircraft position can be monitored from the ground.

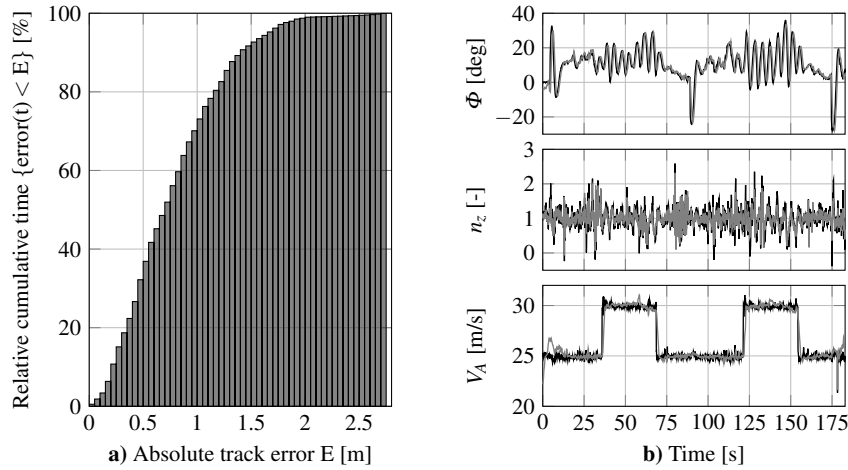
Figure 11 shows the first flight test scenario and demonstrates the high precision of the path-following mode. The desired path has eight underlying waypoints and resembles an aerodrome circuit. It includes continuous variations in altitude and segment-wise changes in commanded airspeed. GPS-altitude varies between 260 m and 337 m. Commanded airspeed is  $V_{A,\text{cmd}} = 30$  m/s between waypoints 5 and 7 and 25 m/s for the rest of the path. The waypoint check distance is set to 10 meters. The radius of the sphere (the tuning parameter of the NLGL) is set to  $R = 50$  m. The safety pilot flies the ULTRA-Extra in close proximity to the first waypoint. The path-following mode is then activated at  $t = 0$  s. The controller forces the ULTRA-Extra to approach the first waypoint. As soon as an intersection point exists, the UAV turns to follow the path. Afterwards, the aircraft flies along the spline path for two consecutive circuits. During the flight test, wind with approximately  $\bar{V}_W = 4$  m/s and a wind direction of  $\bar{\chi}_W = 150^\circ$  was present. The absolute track error during a flight time of 170 seconds was less than 2 m for 98.7% of the time and less than 1 m for 70.1% of the time (see Figure 12a). The maximum deviation from the path was 2.73 m. The tracking performance of the basic flight controller while in path-following mode is shown in Figure 12b.

A second, more challenging test scenario is depicted in Figure 13. Eight waypoints are selected to create a “roller-coaster” like path. The generated spline path varies between 203 m and 306 m in altitude. Compared to scenario one, the maximum curvature of the path is increased by a factor of approximately two. Moreover, five different airspeeds  $V_{A,\text{cmd}} \in \{20, 23, 25, 27, 28\}$  m/s are commanded. The waypoint check distance and the radius of the sphere are unchanged. During the experiment, wind with a mean speed of  $\bar{V}_W = 8$  m/s was present. The mean wind direction was measured to be  $\bar{\chi}_W = 355^\circ$ . Figure 13 reveals less precision in this scenario for some parts of the path. These deficiencies can be related to the protections described in Section 3.1.



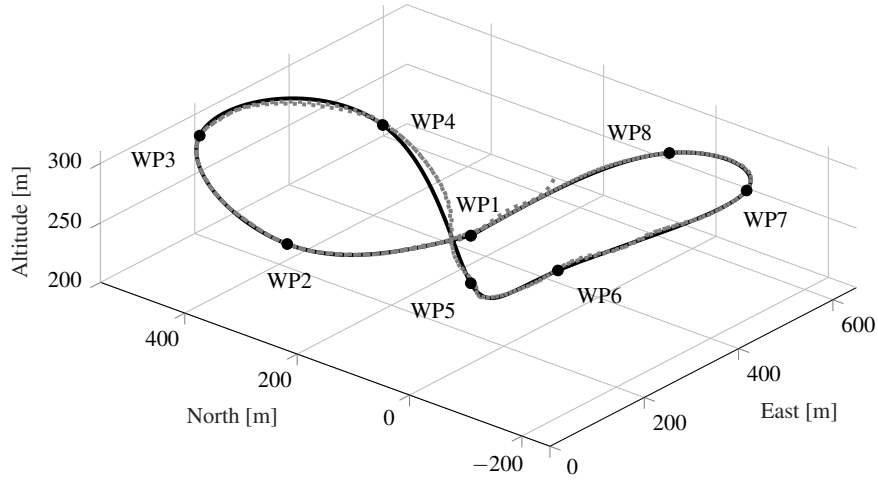


**Fig. 11** Flight test in path-following mode, aerodrome circuit scenario, desired path (—), flight path (---).

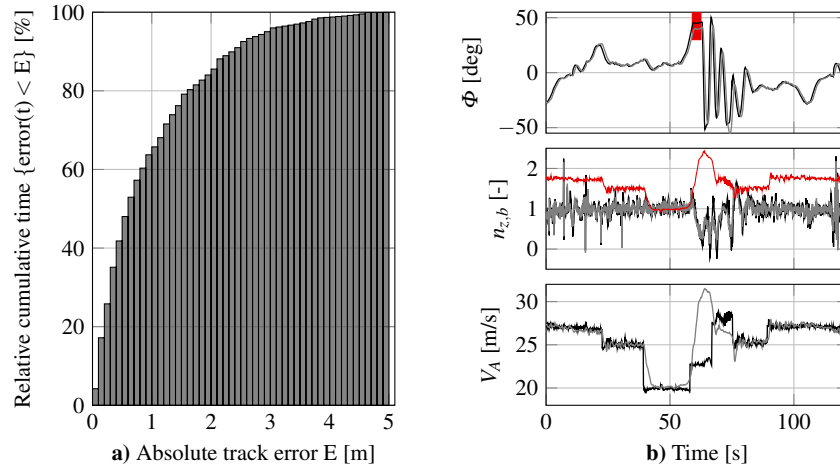


**Fig. 12** Absolute track error during path-following flight and controller performance, command (—), value (—).

The aircraft approaches the first waypoint until a first intersection point exists. After the initial transient period, the ULTRA-Extra follows the desired path. Between waypoints 1 and 3, the commanded airspeed is  $V_{A,\text{cmd}} = 27\text{ m/s}$  for the first spline segment and  $V_{A,\text{cmd}} = 25\text{ m/s}$  for the second segment. In this phase, the absolute displacement is less than 1 m for 90% of the time. The maximum track error is 1.3 m. The precision of the path-following mode is therefore comparable to the other scenario. At waypoint 3, the commanded airspeed is reduced to  $V_{A,\text{cmd}} = 20\text{ m/s}$ . Here, the desired path is still gaining altitude, but the high-alpha-protection of the basic flight control law limits the vertical acceleration (see Figure 14b). This limitation results in an increased absolute track error of about 2–2.5 m. The track error



**Fig. 13** Flight test in path-following mode, second scenario, desired path (—), flight path (---).



**Fig. 14** Absolute track error during path-following flight and controller performance, second scenario, command (—), value (—),  $n_{z,\text{lim}}$  (—).

further increases in the next segment of the path where a descent of 70 m and an airspeed of  $V_{A,\text{cmd}} = 23 \text{ m/s}$  is intended. While the vertical displacement remains small, the lateral error grows up to 16.9 m. This is again a consequence of the protections: Since the airspeed is low at the beginning of this segment, the bank angle is limited to  $\Phi_{\text{max}} = 45^\circ$  (see Table 1 and see Figure 14b, highlighted part). Hence, the necessary lateral acceleration according to Equation (2) cannot be achieved. For the rest of the path (waypoints 5 to 1), the absolute track error is less than 1 m for 64%, less than 2 m for 84% and less than 3 m for 95% of the time (see Figure 14a).

A different situation is observed regarding airspeed tracking. Here, performance limitations are evident. Even though the engine is turned off on the descending spline segment, the ULTRA-Extra accelerates to an airspeed of  $V_A = 31$  m/s (see Figure 14b). Physical limits prevent the aircraft from reducing its speed to the commanded airspeed. Further limitations of the autothrottle controller are apparent in the following ascending part of the spline path: Although maximum thrust is commanded,  $V_{A,\text{cmd}} = 28$  m/s cannot be reached.

## 6 Conclusion

A comprehensive basic flight controller designed for remote control from the ground was first presented. Excellent tracking performance was proven in flight tests and the desired “easy handling” was confirmed by three different pilots. Angle-of-attack and bank angle Protections proved to be functional. Further, a three-dimensional spline-path-following algorithm was described in detail. Its capabilities were validated in two different flight test scenarios and high precision, even in turbulent conditions, was confirmed. Nevertheless, the experimental results also showed that the protections might prevent the controller from exactly following the prescribed path. Future research towards integrated path-planning to ensure flyable paths under dynamic constraints is therefore believed to be necessary.

## References

1. Brockhaus R, Alles W, Luckner R (2011) Flugregelung. Springer, Berlin
2. Cho N, Kim Y, Park S (2015) Three-Dimensional Nonlinear Differential Geometric Path-Following Guidance Law. *Journal of Guidance, Control, and Dynamics* 38(12)
3. Decker BL (1984) World Geodetic System 1984. In: 4th Int. Geodetic Symp. on Satellite Positioning
4. Dekker TJ (1969) Finding a Zero by Means of Successive Linear Interpolation. In: Dejon B, Henrici P, Constructive Aspects of the Fundamental Theorem of Algebra, Wiley-Interscience
5. Dorobantu A, et al. (2013) An Airborne Experimental Test Platform: From Theory to Flight. In: American Control Conference 659–673.
6. Dronecode Project, Inc., A Linux Foundation Collaborative Project, [www.dronecode.org](http://www.dronecode.org)
7. Dubins L (1957) On the Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents. *American Journal of Mathematics*, 79(3):497–516
8. Engeln-Müllges G, et al. (2011) Numerik-Algorithmen. Springer, Berlin, Heidelberg
9. Forsythe GE, Malcolm MA, Moler CB, (1976) *Computer Methods for Mathematical Computations*. Prentice-Hall
10. Horowitz IM (1963) *Synthesis of Feedback Systems*. Academic Press
11. Jategaonkar R (2006) Flight Vehicle System Identification: A Time-Domain Methodology. American Institute of Aeronautics and Astronautics, Inc.
12. Kothari M, Postlethwaite I, Gu DW (2014) UAV Path Following in Windy Urban Environments. *Journal of Intelligent & Robotic Systems* 74:1013–1028
13. Krings M, Annighöfer B, Thielecke F (2013) ULTRA – Unmanned Low-cost Testing Research Aircraft. In: American Control Conference 1472–1477

14. Krings M, Henning K, Thielecke F (2013) Flight Test Oriented Autopilot Design for Improved Aerodynamic Parameter Identification. In: Chu Q, Mulder B, Choukroun D, van Kampen EJ, de Visser C, Looye G (eds) *Advances in Aerospace Guidance, Navigation and Control*. Springer, Berlin
15. Lawrence D, Frew E, Pisano W, (2008) Lyapunov Vector Fields for Autonomous Unmanned Aircraft Flight Control. *Journal of Guidance, Control, and Dynamics* 31(5):1220–1229
16. Lizarraga M, Curry R, Elkaim GH (2013) Flight Test Results for an Improved Line of Sight Guidance Law for UAVs, In: *American Control Conference* 818–823
17. de Marina HG, Kapitanyuk YA, Bronz M, Hattenberger G, Cao M (2017) Guidance Algorithm for Smooth Trajectory Tracking of a Fixed Wing UAV Flying in Wind Flows, In: *ICRA*
18. Nelson DR, Barber DB, McLain TW, Beard RW (2007) Vector Field Path Following for Miniature Air Vehicles. *IEEE Transactions on Robotics* 23(3):519–529
19. Niemann C, Montel M, Thielecke F (2014) Development of an Air Data System for an Unmanned Research Aircraft. In: *Deutscher Luft- und Raumfahrtkongress*
20. Park C, Kim HJ, Kim Y (2014) Real-Time Leader-Follower UAV Formation Flight Based on Modified Nonlinear Guidance, In: *29th ICAS Congress*
21. Park C, Cho N, Lee K, Kim Y (2015) Formation Flight of Multiple UAVs via Onboard Sensor Information Sharing. *Sensors* 15:17397–17419
22. Park S, Deyst J, How J (2004) A New Nonlinear Guidance Logic for Trajectory Tracking. In: *AIAA Guidance, Navigation and Control Conference* No 2004-4900
23. Park S, Deyst J, How J (2007) Performance and Lyapunov Stability of a Nonlinear Path Following Guidance Method. *Journal of Guidance, Control, and Dynamics* 30(6):1718–1728
24. Park S (2012) Autonomous Aerobatics on Commanded Path. *Aerospace Science and Technology*, 22(1):64–74.
25. Ratnoo A, Sujit PB, Kothari M (2011) Optimal Path Following for High Wind Flights. In: *IFAC World Congress* 12985–12990.
26. Schmidt D (2012) *Modern Flight Dynamics*. McGraw-Hill
27. Sujit PB, Saripalli S, Sousa JB (2014) Unmanned Aerial Vehicle Path Following: A Survey and Analysis of Algorithms for Fixed-Wing Unmanned Aerial Vehicles. *IEEE Control Systems* 34(1):42–59.
28. Theis J, Ossmann D, Thielecke F, Pfifer H (2018) Robust Autopilot Design for Landing a Large Civil Aircraft in Crosswind. *Control Engineering Practice* 76:54–64