EuroGNC Bristol, UK June 11th-13th 2024

Trajectory Planning for Efficient BVLOS Drone Flights over Agricultural Points of Interest

Max Hartmann Research Associate, RWTH Aachen University, Institute of Flight System Dynamics, 52056, Aachen, Germany. max.hartmann@fsd.rwth-aachen.de

Nicolai Voget Research Engineer, flyXdrive GmbH, 52074, Aachen, Germany. voget@flyxdrive.com

Sebastian Seitz Chief Engineer, RWTH Aachen University, Institute of Flight System Dynamics, 52056, Aachen, Germany. seitz@fsd.rwth-aachen.de

Dieter Moormann Head of Institue, RWTH Aachen University, Institute of Flight System Dynamics, 52056, Aachen, Germany. office@fsd.rwth-aachen.de

ABSTRACT

Long-range unmanned flights beyond visual line of sight pose multiple challenges on planning and execution of flight missions. For example, topography, transitions between geographical zones each with different height limitations, or limits defined in the operational authorization might yield to highly varying elevation. All this has to be considered during mission planning. Additionally, a high degree of automation is needed for this planning in order to complete a vast amount of flights within a limited time. In order to accomplish these challenges, we present a toolchain of methods that converts externally provided waypoints into a flyable trajectory respecting the aircraft's flight envelope as well as position dependent restrictions. Due to the wind dependency of trajectory planning we split the toolchain into both a pre-flight and an in-flight process. Given a list of 2D waypoints, the pre-flight process calculates a 2D flight path. Then, area-based restrictions are applied to the flight path, resulting in a height corridor and speed limits coupled to the flight path progress. Based on a given list of points of interest, sections of the path are calculated on which special actions (e.g. activating payload equipment) will be triggered to fulfill the mission. During flight, receding horizon control is used to convert the restricted flight path into a trajectory with fixed speed and altitude profiles.

Keywords: BVLOS, Trajectory Planning, Receding Horizon

Nomenclature

$h_{\mathrm{c,l}}$	=	lower tolerance limit at the currently regarded $l_{\mathrm{pos,cur}}$
$h_{\mathrm{c,u}}$	=	upper tolerance limit at the currently regarded $l_{\mathrm{pos,cur}}$
l_{pos}	=	positional length, cumulative length along the path from the path's starting point describing the path progress
$l_{\mathrm{pos,cur}}$	=	l_{pos} at the currently regarded tolerance limits
$l_{\mathrm{pos,end}}$	=	l_{pos} at the end of an interval
$l_{\mathrm{pos,start}}$	=	l_{pos} at the start of an interval

$l_{\text{pos},\times}$ = l_{pos} at the crossing point $p_\times$
$p_{\max}$ = point restricting the line of maximum inclination
$p_{\min}$ = point restricting the line of minimum inclination
$p_\times$ = crossing point of corridor boundary lines
V_{K} = ground speed along the path

1 Introduction

Unmanned Aerial Vehicles (UAVs) are increasingly used in various domains, such as agriculture [1] and mining [2], rescue missions [3], medical goods transportation [4], and infrastructure inspection [5]. These applications benefit from technological advancements, automation, and the development of Beyond Visual Line of Sight (BVLOS) capabilities. BVLOS operations have become a prominent topic in environmental monitoring in recent years [6]. They have the potential to transform the monitoring and management of agricultural landscapes by providing timely and high-resolution aerial imagery, which is essential for optimizing crop yields, resource allocation, and decision-making processes.

In July 2023, a flight campaign with unmanned aerial vehicles was carried out to photograph several thousand agricultural fields throughout the German Free State of Saxony. In total, a flight distance of more than 10 000 km had to be covered throughout the whole federal state. Up to four UAVs simultaneously performed highly automated BVLOS flights controlled by an operator to allow for fast completion of the flight campaign. The employed UAVs are tilt-wing UAVs, allowing for fast and energy efficient flight over long distances while also being able to start and land vertically and perform hover-flight. To fulfill the campaign in a short timeframe, an efficient coverage of all fields was necessary, which brought many challenges especially for mission and path planning as well as path control [7].

To enable an efficient coverage of the complete flight area, long-range unmanned flights of up to 80 km each were executed. Various restrictions — such as a high variation of elevation due to topography, transitions between geographical zones each with different height corridors, or limits defined in the operational authorization — had to be considered during mission planning for each flight. In conclusion, a high degree of automation was needed for this planning to fulfill a vast amount of flights within a limited time.

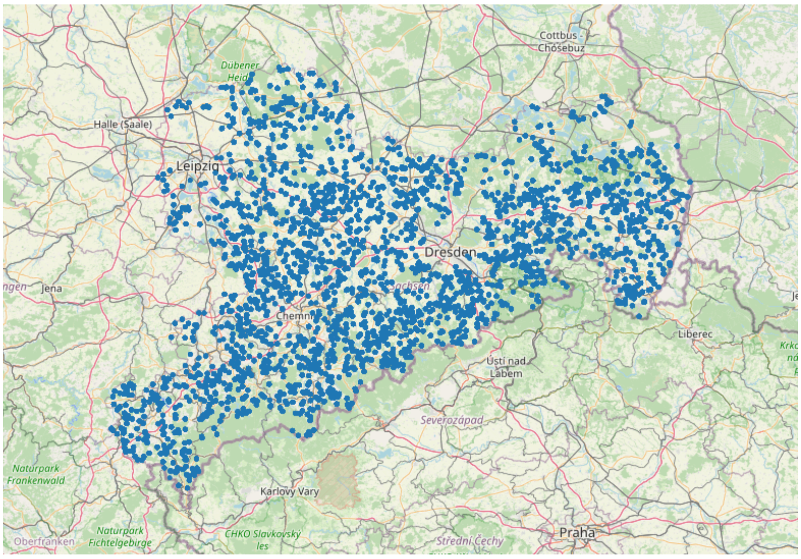


Fig. 1 Figure showing the extension of the flight campaign. Map data © OpenStreetMap

To accomplish these challenges we propose a set of methods to plan a trajectory for the UAV based on given waypoints, taking into account the aircraft's flight mechanics constraints as well as position dependent restrictions. While mission planning can be carried out well before the flights, our trajectory planning algorithm considers wind, requiring periodic replanning in-flight. Thus, we split our toolchain into both a pre-flight and an in-flight process.

The architecture of the proposed system is introduced in section 2, giving an overview of the structure and the interdependencies of the components. Subsequently, each component is presented in detail starting with the path planner in section 3. Afterwards, the trajectory controller is reviewed in detail in section 4. Finally, real-world examples from the extensive flight campaign are discussed, showing the results of the presented set of methods in execution.

2 Architecture

To account for different requirements on the components of the flight guidance system as well as to allow for reusability and flexibility, the complete system is subdivided into several smaller components, each fulfilling one specialized task [8]. As can be seen in figure 2, we therefore define the relevant components as path planner, trajectory planner and trajectory controller. Path planner and trajectory planner will be described in detail in sections 3 and 4, respectively.

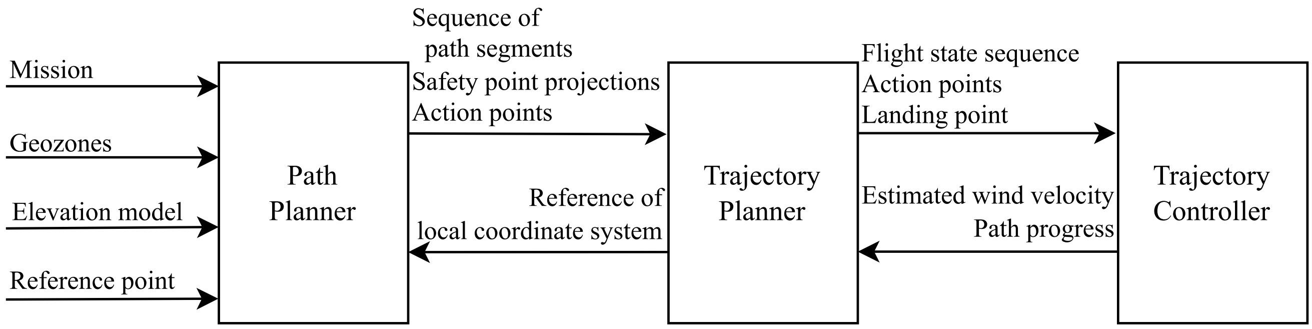


Fig. 2 Architecture of the relevant components of the flight guidance system

For the flight campaign, the general mission for each flight was to take photos at several points of interest. An automated mission generation algorithm calculates the takeoff and landing point as well as a list of two-dimensional waypoints to fulfill the mission, as presented in [9]. Those waypoints are specified by geographic coordinates and annotated with a maximum radius to be flown at the waypoint. Additionally, the mission includes the points of interest, also defined by geographic coordinates and annotated with an id. Lastly, safety points specified by geographic coordinates, a height and a corresponding radius, as presented in [10], are included in the mission.

During mission planning the mission is only defined in the horizontal plane. In the path planner, an allowed height band, consisting of an upper and a lower height limit, will be added. This height band has to meet various requirements given by organizational or legal procedures. Firstly, as introduced in [11] upper limits to the height band are imposed by the operational authorization. Secondly, geographical zones like nature reserve areas and power lines limit the lower limit of the height band. All those geographical zones annotated with their restrictions are therefore passed to the path planner. As all restrictions are defined as a height above ground, an elevation model is fed into the path planner to allow for path planning based on altitude. The resulting path, calculated by the path planner, is a sequence of two-dimensional path segments superimposed by the altitude band, giving a three-dimensional flight path. Additionally, action points are defined, which are placed on the flight path in proximity to the points of interest. At those points the camera will be turned on or off during flight. Also, for each safety point positions on the flight path are calculated at which this safety point is applicable. All data is then passed on to the trajectory planner.

In the trajectory planner the three-dimensional flight path, the current state of the aircraft, and wind conditions are used to precompute a series of flight states to be executed in the trajectory controller. In the trajectory planner a coupled altitude and speed planning in the sense of receding horizon control is carried out, which allows for an efficient and fast flight, yielding a list of preplanned flight states with a fixed altitude profile. The trajectory planner uses a wind estimation from the UAV's flight state controller and the progress on the flight path which is calculated in the trajectory controller. The list of preplanned flight states, beginning at the current location of the aircraft on the flight path and ending in a safe state at a safety point, is then passed on to the trajectory controller. Also, the upcoming action points are sent to the trajectory controller as it will trigger the camera when the UAV reaches the action point.

3 Path Planning

The routing process from [9] produces missions described by route waypoints and the points of interest covered by this route, each point's position defined by a pair of geographic latitude and longitude. In contrast, the path which is to be passed to the trajectory planner is described by points in a cartesian local coordinate system. Therefore, all geographic positions get transformed into this local coordinate system first. Next, the horizontal course of the flight path is calculated from the converted waypoints. Given this two-dimensional path, restrictions of altitude and velocity along the path can be determined. Additionally, for every point of interest the section of the path is calculated along which images shall be captured. Finally, all points on the path are determined from which the aircraft could deviate to head for one of the given safety points.

3.1 Constructing two-dimensional path from waypoints

To construct the two-dimensional path from the given waypoints, first each pair of consecutive waypoints gets connected through a straight line segment. Then, each resulting corner is replaced with a circular arc that is tangential to both straight line segments crossing at the respective waypoint. The radius of this circular arc is limited by two factors: First, it may not exceed the desired radius defined for the corresponding waypoint. Second, the transition from the straight line segment to the resulting arc must not be further away from the waypoint than the center point of the straight line segment. Thereby, intersection of consecutive arcs is prevented even in case the path's geometry does not allow the desired turn radius.

The resulting path is a sequence of straight line segments and circular arcs. Each of these path elements can be described completely by its type (straight or cricular), 2D starting position, 2D starting direction, element length, and—for circular arcs—its radius and turn direction. Any point on the path is then identified unambiguously using the cumulative length along the path from the path's starting point up to the desired point. In the following, this length is denoted as *positional length* or l_{pos}.

3.2 Location dependent restrictions

Following the calculation of the two-dimensional path, the respective maximum speed and the height band inside which the aircraft may fly along the path have to be determined. Their exact values depend on the actual location of each point on the path, as restrictions stem from different geographical areas. Most importantly, flight operation is permitted only within flight geographies approved by authorities. As described in [11], the extension of a flight geography depends—among other parameters—on respective maximum inertial speed and maximum height above ground level (AGL) which the aircraft may attain. Using multiple flight geographies for different combinations of maximum speed and height enables the aircraft to cover areas close to buildings, even though at slow speed, while flight in greater distance to buildings can be performed more efficiently at high speed. Additionally, flying over nature conservation areas as well as crossing power lines is only allowed above certain minimum heights.

Intersecting the previously calculated two-dimensional path with the geographic boundaries of the different areas yields all points on the path at which speed or height restrictions change. For each of these so-called *joints*, the following parameters are saved:

- l_{pos}
- 2D coordinates
- Outgoing height range

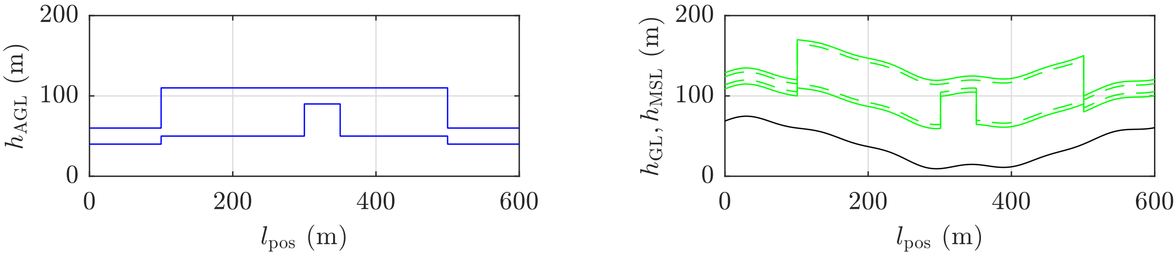
- Outgoing maximum speed

For a fixed path, l_{pos} alone would suffice to unambigously define the point of the joint. However, [12] explains that both the starting and ending point of the route will be moved to the aircraft's actual takeoff location, thereby altering the locations resulting from previously calculated values of l_{pos}. Then again, the location alone also isn't eligible as a path may cross the same exact location in different directions, e.g. during departure and approach, leading to joints with different restrictions for the same coordinates. However, using the combination of the originally calculated l_{pos} and the corresponding location, l_{pos} can be updated correctly after moving the starting point. This is done by following the altered path beginning at the saved l_{pos} until the locally minimal distance to the location is found.

3.3 Altitude limits

The height limits determined in the previous step are all defined w.r.t. ground level. In contrast, the aircraft's controller tracks altitude, being the distance to the geoid or mean sea level (MSL). There are various reasons for this setup: First of all, the aircraft's potential energy and thus its dynamics relate to altitude. Therefore, maintaining constant altitude leads to less energy consumption, regardless of resulting changes in height caused by varying ground elevation. Also, altitude is measured by the equipped GNSS receiver whereas the aircraft does not contain any sensor for measuring height.

To transform the previously determined height range into an equivalent altitude range, first the two-dimensional path is discretized into points with one meter spacing. For each point, the respective ground elevation h_{GL} is interpolated from EU's digital elevation model [13]. Then, the altitude limits are calculated by $h_{\text{MSL}} = h_{\text{AGL}} + h_{\text{GL}}$, see Fig. 3. This process yields a high number of joints, e.g. 60 000 joints for a typical route length of 60 km. To reduce this number, the course of both upper and lower altitude limits are approximated through a sequence of linear segments. However, this approximation cannot be performed via a classical line fitting approach such as linear least squares because the resulting range must not leave the original one, i.e. the approximation for the lower/upper altitude limit must nowhere be smaller/greater than the original limit, respectively. Instead, a different approach is used as described in detail in the following section.



(a) Lower and upper height limits

(b) Resulting altitude limits in green, ground elevation in black

Fig. 3 Transformation of height limits to altitude limits

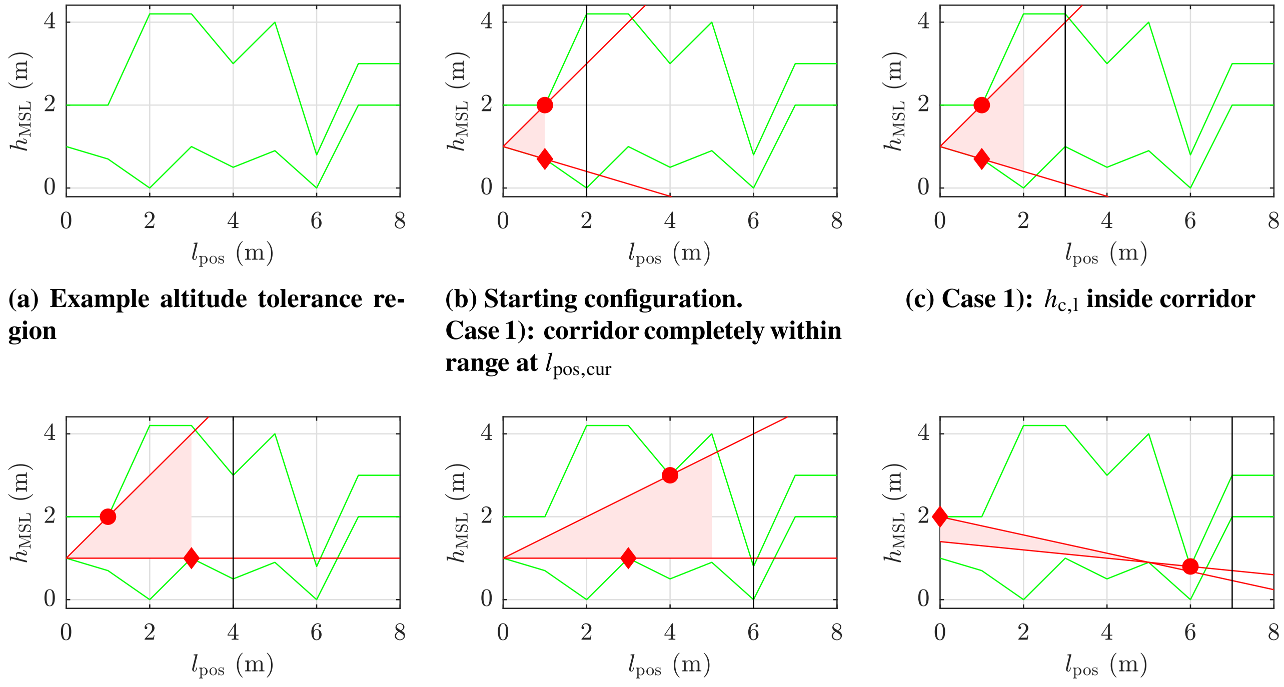
3.3.1 Approximation of altitude limits

To begin with, the available altitude range is divided into three regions: a tolerance region within which the approximation for the upper limit must lie, an analogous tolerance region for the lower limit, and a region in between, which is guaranteed to be available for the aircraft even after approximation. In Fig. 3, these three regions are separated by dashed lines. The height of each tolerance region was chosen to be a quarter of the range's total height at each point, but at most 5 m.

Given a tolerance region, a corridor is spanned by two straight lines being defined by three points: the crossing point of both lines, $p_{\times}$, a point restricting the maximum inclination, p_{max}, and a point restricting the minimum inclination, p_{min}. The resulting corridor then specifies the range of all valid approximation lines going through $p_{\times}$. $p_{\times}$ is always defined by a point on the lower tolerance border. This can be justified as follows. If there exists a line segment approximating a section of the tolerance region that does not touch the lower tolerance border at any point, this line segment can be moved downward in parallel until it touches the lower tolerance border. This parallel line must then also comply to the same section of the tolerance region: No upper limit can be violated because the new line is below the original valid approximating line. Also, all lower limits are satisfied because the line gets shifted downwards only until it touches the lower tolerance border.

During the following approximation, $\left[l_{\text{pos,start}}, l_{\text{pos,end}}\right]$ denotes the range of the positional length for which the regarded corridor is known to comply with the given tolerance region; $h_{\text{max}}(l_{\text{pos}})$ and $h_{\text{min}}(l_{\text{pos}})$ indicate the points on the line of maximum and minimum inclination at l_{pos}, respectively; $h_{\text{c,u}}$ and $h_{\text{c,l}}$ are the upper and lower altitude boundaries at the currently regarded positional length, $l_{\text{pos,cur}}$.

At the start of the approximation, $p_{\times}$ is set to the leftmost point on the lower tolerance border while p_{max} and p_{min} are set to the directly following points on the upper and lower boundary of the tolerance region, respectively. Moving along the tolerance region, the following conditions are checked. For better comprehension, Fig. 4 displays changes to the corridor for a sample tolerance region.



(a) Example altitude tolerance region

(b) Starting configuration. Case 1): corridor completely within range at $l_{\text{pos,cur}}$

(c) Case 1): $h_{\text{c,l}}$ inside corridor

(d) Case 1): $h_{\text{c,u}}$ inside corridor

(e) Case 2): $h_{\text{c,u}}$ below corridor

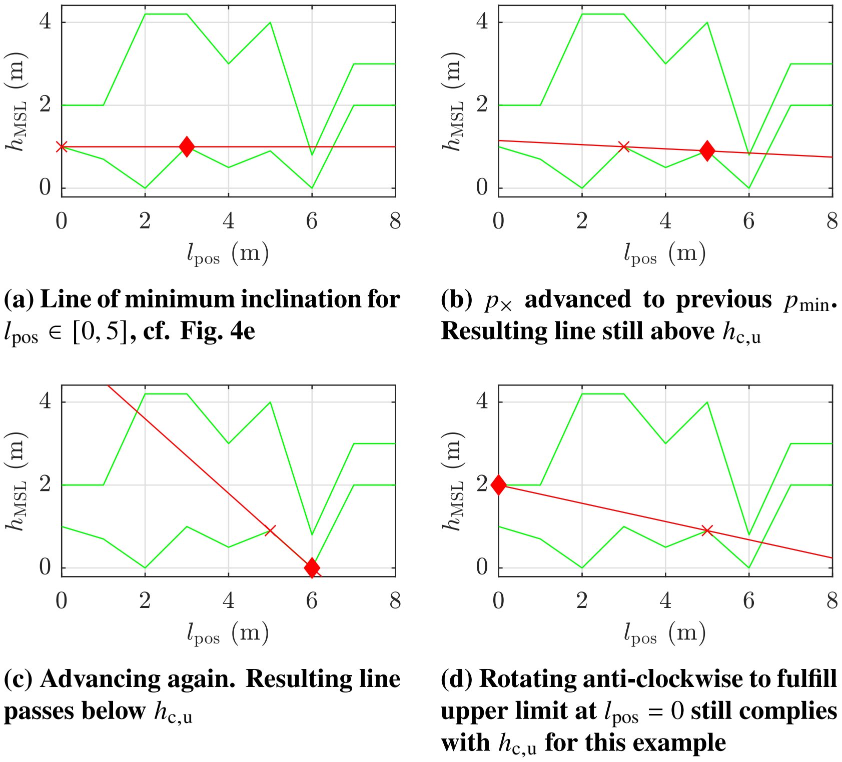
(f) Case 3): $h_{\text{c,l}}$ above corridor

Fig. 4 Changes to the corridor of approximating lines when iterating along a tolerance region. The filled area represents the section of the corridor that is known to comply with the tolerance region. $l_{\text{pos,cur}}$ is denoted by a black vertical bar, p_{max} by a disc and p_{min} by a diamond. The resulting corridor displayed in (f) stays within the given tolerance region for $l_{\text{pos}} \in [0, 6]$. No approximation is possible for $l_{\text{pos}} \in [0, 7]$ with one single line segment, therefore a new approximation starts at $l_{\text{pos}} = 6$.

1) As long as $h_{\text{c,l}} \leq h_{\text{max}}(l_{\text{pos,cur}})$ and $h_{\text{c,u}} \geq h_{\text{min}}(l_{\text{pos,cur}})$, at least part of the corridor complies with the tolerance range at $l_{\text{pos,cur}}$. If $h_{\text{c,l}} > h_{\text{min}}(l_{\text{pos,cur}})$, p_{min} gets replaced with $\left(l_{\text{pos,cur}}, h_{\text{c,l}}\right)$. The same way, p_{max} is set to $\left(l_{\text{pos,cur}}, h_{\text{c,u}}\right)$ if $h_{\text{c,u}} < h_{\text{max}}(l_{\text{pos,cur}})$. If the corridor gets modified, the resulting corridor is always a true subset of the previous one because both lines get rotated around their crossing point towards each other. Therefore, the new corridor still satisfies all limits between $l_{\text{pos,start}}$ and $l_{\text{pos,end}}$. Finally, $l_{\text{pos,end}}$ is increased to $l_{\text{pos,cur}}$ and the next point is examined.
2) If $h_{\text{c,u}} < h_{\text{min}}(l_{\text{pos,cur}})$, the currently regarded tolerance range at $l_{\text{pos,cur}}$, $\left[h_{\text{c,l}}, h_{\text{c,u}}\right]$, has no intersection with the corridor at all. However, if both p_{min} and $p_{\times}$ lie on the lower boundary,

moving $p_\times$ might yield a corridor that satisfies all tolerances within $\left[l_{\text{pos,start}}, l_{\text{pos,cur}}\right]$. This is discussed in more detail in the following paragraphs. If a new corridor is found this way, set $l_{\text{pos,end}} = l_{\text{pos,cur}}$ and continue with the next point. Otherwise, the original corridor's line of minimum inclination is used as approximation for $\left[l_{\text{pos,start}}, l_{\text{pos,end}}\right]$ and a new approximation run starts from $l_{\text{pos,end}}$.

3) If none of the preceding conditions apply, $h_{\text{c,l}}$ must be above the corridor. In this case, there can be no straight line that fulfills all limits for $\left[l_{\text{pos,start}}, l_{\text{pos,cur}}\right]$: Moving the existing line of maximum inclination upwards in parallel would exceed p_{max} while increasing its gradient would violate p_{max} or $p_\times$. Therefore, the lower corridor line is used as approximation for $\left[l_{\text{pos,start}}, l_{\text{pos,end}}\right]$ and a new approximation run starts from $l_{\text{pos,end}}$.



(a) Line of minimum inclination for $l_{\text{pos}} \in [0, 5]$, cf. Fig. 4e

(b) $p_\times$ advanced to previous p_{min}. Resulting line still above $h_{\text{c,u}}$

(c) Advancing again. Resulting line passes below $h_{\text{c,u}}$

(d) Rotating anti-clockwise to fulfill upper limit at $l_{\text{pos}} = 0$ still complies with $h_{\text{c,u}}$ for this example

Fig. 5 Advancing $p_\times$ and rotating the line of minimum inclination to satisfy $h_{\text{c,u}}$ for $l_{\text{pos,cur}} = 6$. The cross denotes $p_\times$ while p_{min} is again represented by a diamond.

For case 2), if p_{min} lies to the right of $p_\times$, it may be possible to construct a new corridor by rotating the line of minimum inclination clockwise around p_{min} as shown in Fig. 5. As this rotation causes the new line to be above the old line everywhere left of p_{min}, all lower limits within $\left[l_{\text{pos,start}}, l_{\text{pos}}(p_{\text{min}})\right]$ are still being adhered to. Also, the new line drops right of p_{min}, leading to all upper limits within $\left[l_{\text{pos}}(p_{\text{min}}), l_{\text{pos,end}}\right]$ still being valid. All lower limits between $l_{\text{pos}}(p_{\text{min}})$ and $l_{\text{pos,end}}$ must have been on or below the original line of minimum inclination. Also, $h_{\text{c,l}}$ lies below this line, as even $h_{\text{c,u}}$ does. Therefore, there exists at least one lower limit within $\left]l_{\text{pos}}(p_{\text{min}}), l_{\text{pos,cur}}\right]$ such that the line connecting p_{min} and this point does not violate any lower limit within this range and the resulting inclination is equal to or lower than the one of the original line of minimum inclination. Then, p_{min} is redefined to be this point and $p_\times$ to the previous p_{min}.

Given this new line of minimum inclination, two conditions remain to be checked. First, for the new corridor to be valid $h_{\text{c,u}}$ needs to be above this line. If this condition is not met, $p_\times$ and p_{min} have to be moved again as described previously. This will result in a valid line eventually because at the latest the lower limits for $l_{\text{pos,end}}$ and $l_{\text{pos,cur}}$ define a line that does not lie above $h_{\text{c,u}}$. Second, rotating the line of minimum inclination clockwise can cause it to violate upper limits left of the resulting $p_\times$. Therefore, once the first condition is met, all upper limits for $\left[l_{\text{pos,start}}, l_{\text{pos}}(p_\times)\right[$ have to be checked. If an upper limit is below the line of minimum inclination, p_{min} is replaced by this limit, rotating the line of minimum inclination anti-clockwise around $p_{\times}$. This can cause the line going above $h_{\mathrm{c,u}}$ again, in which case no corridor exists for $\left[l_{\mathrm{pos,start}}, l_{\mathrm{pos,cur}}\right]$. Otherwise, a corridor is found that lies completely within the tolerance range for $\left[l_{\mathrm{pos,start}}, l_{\mathrm{pos,cur}}\right]$.

Applying the proposed approximation algorithm to both the upper and lower tolerance region yields piecewiese linear segments for upper and lower altitude limits.

3.4 Projection points

The main objective of the whole mission is to perform specific actions such as taking pictures in the vicinity of given points of interest (POIs). These POIs are defined as id—coordinate pairs. Additionally, d is a globally predefined distance along the path defining the range around each POI inside which pictures shall be taken. For every POI, the positional length of the nearest point on the path is determined. Then, a trigger annotated with the POI's id is set at $l_{\mathrm{pos}} - d$ to activate the camera and another one at $l_{\mathrm{pos}} + d$ to deactivate it again. During flight, each trigger is executed as soon as the aircraft's current l_{pos} exceeds the trigger's positional length. When activated, the camera takes pictures at a constant frequency, which are then annotated with the camera's respective location and orientation as well as the corresponding POI's id.

Finally, to enable safe early landing as presented in [14], positions on the path have to be determined from where it is safe to head for a safety point in a straight line. Besides its location, every given safety point holds a safety radius that defines a disk around the safety point inside which no obstacles are present above the safety height. To minimize flight distance off the pre-planned route, only points on the path are selected which have a locally minimal distance to a safety point, i.e. where the connection to the safety point is perpendicular to the path's direction. For every such path point, l_{pos} is saved together with the local coordinates of the reachable safety point.

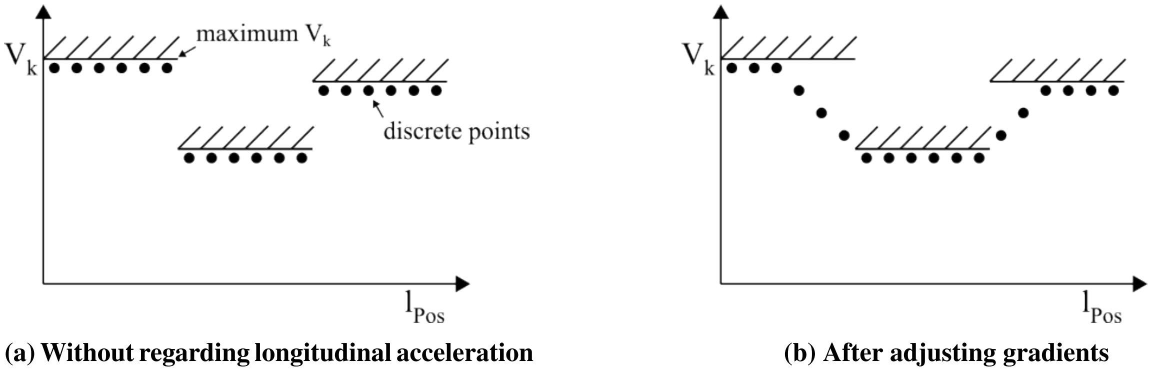
4 Trajectory Planning

The three-dimensional flight path which was calculated as described in the previous section still needs to be converted to commands for the flight state controller of the tilt-wing UAV. To allow for feasible flight state commands, the flight dynamics constraints of the UAV have to be considered. From the space of possible flight states, the course should be selected to minimize flight time. To solve this problem a receding horizon control approach is employed. This approach computes a series of flight states, taking into account the three-dimensional flight path, the current state of the aircraft, and wind conditions. These flight states are subsequently sent to the UAV's flight state controller.

For receding horizon control, the flight path is discretized at equidistant points up to a predefined horizon. For each of these points a flight state is preplanned. Each flight state consists of ground speed, vertical speed, altitude, and, optionally, radius, turn angle, and center of the turn. Each flight state can be identified by its position on the path, described by its l_{pos} (see section 3). The list of preplanned flight states will be referred to as state plan in the following sections. Using the assumptions of [15] and the input data, the majority of this data can be derived directly from the given path, leaving only the horizontal speed and altitude to be determined through more complex computation.

The computation of these values occurs in two separate steps. The first step focuses on determining the horizontal speed, following the work of [15] and [16] closely. Initially, the maximum ground speed V_{K} is calculated for every flight state, factoring in the flight dynamics constraints of the UAV but ignoring acceleration limitations. An example of this initial calculation is illustrated in Fig. 6a.

However, these speeds are not practically achievable, primarily due to the high gradient of the ground speed, which the UAV cannot feasibly follow with a limited acceleration. To address this, V_{K} is lowered



(a) Without regarding longitudinal acceleration

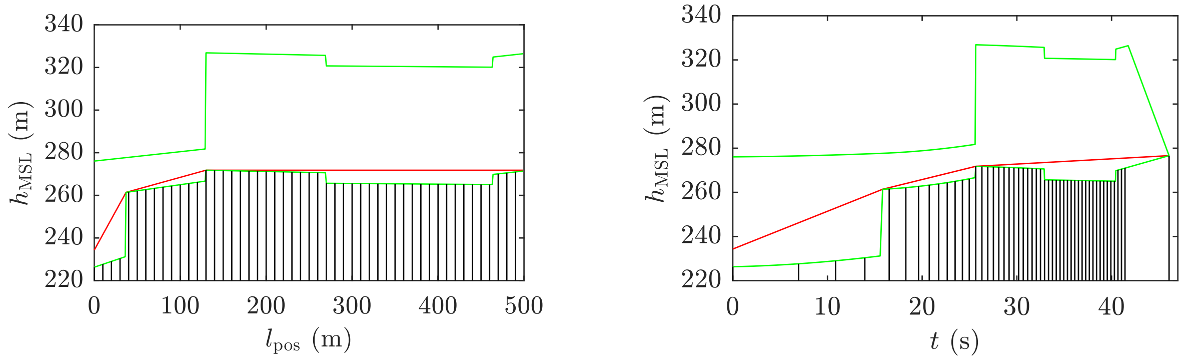
(b) After adjusting gradients

Fig. 6 Horizontal speed profile (see [15])

where needed in order to fulfill the UAV's acceleration limits. A result of this method is illustrated in Fig. 6b. [15] Next, the altitude profile is calculated. The input path contains an altitude band, as described in section 2, comprising upper and lower bounds. These bounds are represented by green lines in Fig. 7a. The position of every tenth state plan point is represented by a vertical bar.

The calculation of the altitude profile aims for constant vertical speed, while still accounting for the flight dynamics of the UAV and staying inside the bounds of the altitude range. It also aims for the shortest path through the given altitude boundaries, i.e. the altitude should change as little as possible. A first approach for this altitude profile can be seen in Fig. 7a, represented by the red line. It is calculated using the same method as described in section 3.3, with the additional restriction that altitude must not change at the transition between consecutive segments. Therefore, $p_\times$ is set to the current aircraft's altitude for the first segment. For the following segments, $p_\times$ is fixed to the altitude at the respective previous segment's end. Therefore, case 2) from section 3.3 is handled the same as case 3).

This results in a constant climb angle. The vertical speed w, however, is calculated by $w = -\frac{\Delta h}{\Delta t}$, where Δh describes the change in altitude between points and Δt the time between points. Since Δt depends on V_K, changes of V_K along a segment of constant climb angle lead to varying vertical speeds. This will result in turn in a higher vertical speed w.



(a) In spatial domain

(b) In time domain

Fig. 7 Altitude profile of the state plan. Green lines denote altitude limits. Every tenth state plan point is marked by a vertical bar. The red line depicts the optimal altitude profile.

Therefore, the altitude range is transformed into the time domain using V_K. This will cause the state plan points to be no longer equidistant, since the time between points varies with the horizontal speed V_K. The warped altitude profile can be seen in 7b. Applying the previously explained method on these warped boundaries yields the altitude profile which is represented by the red line in 7b.

Due to the limited horizon of the state plan changes of the boundaries beyond the horizon are not taken into account. Therefore, the aircraft will fly at constant altitude even if a boundary crosses this altitude, as long as this crossing is behind the horizon. To counter this problem, the altitude range is extended by a so called guiding point, denoting the next altitude that must be cleared by the UAV. This point is calculated by approximating the altitude range in the spatial domain, but this time continuing until the segment's end point lies beyond the horizon. This end point is then used as guiding point. Fig. 8 displays this method, with the first section being similar to Fig. 7a.

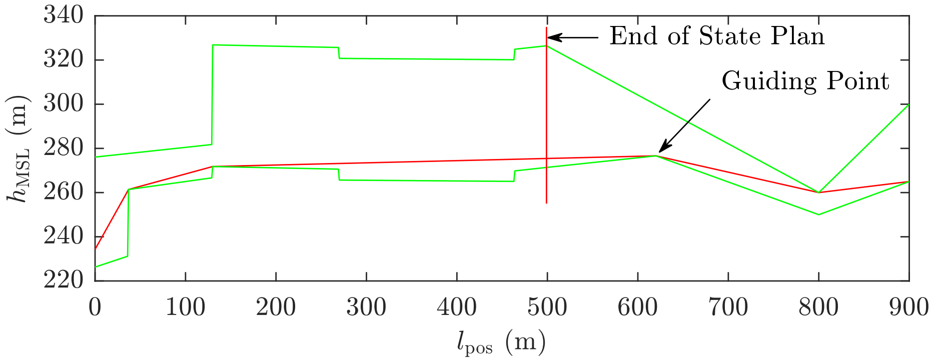


Fig. 8 Determination of guiding point

Using this guiding point, the altitude profile can be calculated as previously described, now aiming for the guiding point at the end. This can be seen at the right side of Fig. 7b. Evaluating the lines at every point, the altitude for every flight state can be determined. Using these altitudes, the vertical speed w is calculated by interpolation with the previously described equation of $w = -\frac{\Delta h}{\Delta t}$.

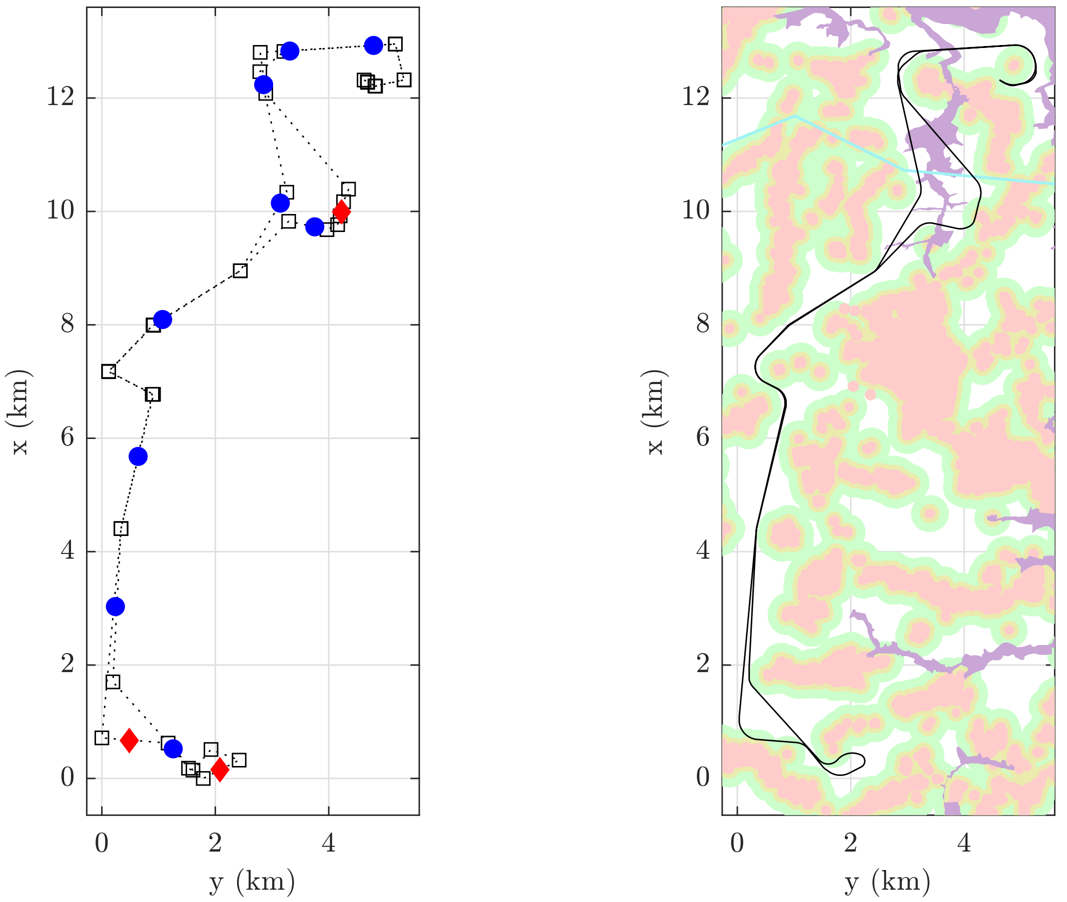
The resulting vertical speeds are then compared to the flight dynamics constraints of the UAV. If w exceeds these limits at any point, Δt between the respective points is increased by lowering V_{K}. However, because vertical and horizontal speed are interdependent, the necessary reduction cannot be calculated directly, but has to be performed iteratively. After the horizontal speeds have been lowered the accelerations are adjusted again, following the method depicted in Fig. 6b. Since Δt was changed, the warping of the altitude range is affected directly. Therefore, a new altitude profile is calculated for the new warped altitude range. This iterative process is repeated until the vertical speed limits are satisfied at every point.

5 Validation

The methods presented in this paper are validated using a real flight that was executed during the flight campaign. The mission input consists of 38 waypoints, three POIs, and nine safety points, see Fig. 9a. Following the method presented in section 3.1, the resulting flight path consisting of linear segments and circular arcs is shown in Fig. 9b together with the permitted flight geographies as well as the geographical zones.

Intersecting the resulting path with the given flight geographies yields the limits for inertial speed and height as shown in Fig. 10. From the speed limits it can be seen where the path does not lie inside the flight geography permitting high speed: during start and landing, at the segment around $x = 7$ km, which is passed both about $l_{\mathrm{pos}} = 10.6$ km and $l_{\mathrm{pos}} = 27.6$ km, and in the vicinity of the three POIs. For the first and third POI the path even needs to pass through the flight geography with $V_{\mathrm{K,max}} = 10$ m/s. In Fig. 10b, both upper and lower height limit are reduced at the exact same regions of l_{pos} where speed is reduced. Additionally, the lower limit is raised to 90 m at multiple sections in the first and last quarter of the flight. This happens where the path crosses power lines or passes through nature reserve areas.

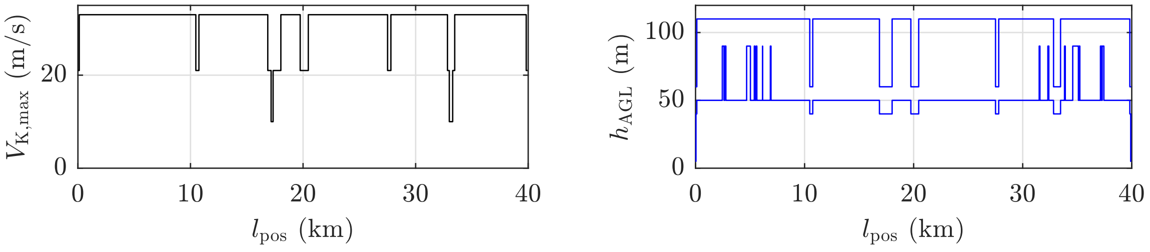
Superimposing the height range with the ground elevation profile from Fig. 11 gives an altitude range with 39 958 points each for the upper and lower limit. Applying the approximation method presented in section 3.3, both range boundaries can be represented by 147 piecewise linear segments each.



(a) Mission input: Waypoints (black squares), POIs (red diamonds) and safety points (blue dots)

(b) Flight path with geographic regions. Flight geographies with reduced speed limit are filled green and yellow; red areas are forbidden; purple denotes nature reserve areas; the light blue line represents a power line.

Fig. 9 Mission input and resulting flight path for a real flight mission. Starting and landing position is at (12.3 km, 4.6 km). Flight path is followed clockwise, i.e. the POI at (0.7 km, 0.5 km) is visited first and the one at (10 km, 4.2 km) last.



(a) Speed limits

(b) Height range

Fig. 10 Limits determined by intersecting flight path with geographical zones

Using the flight path and height range, the trajectory planner calculates a new state plan once every second using the method described in section 4. The effective velocity commands for the examined flight can be seen in figure Fig. 12a. The aerodynamic velocity only needs to be lowered at locations on the path where the allowed inertial velocity is limited, as can be seen for the ranges starting at l_{pos} = 10.6, 17 and 33 km. Due to the prevailing wind, the moderate lowering of $V_{K,max}$ starting at l_{pos} = 19.7 km and l_{pos} = 27.6 km did not even require any reduction of V_A.

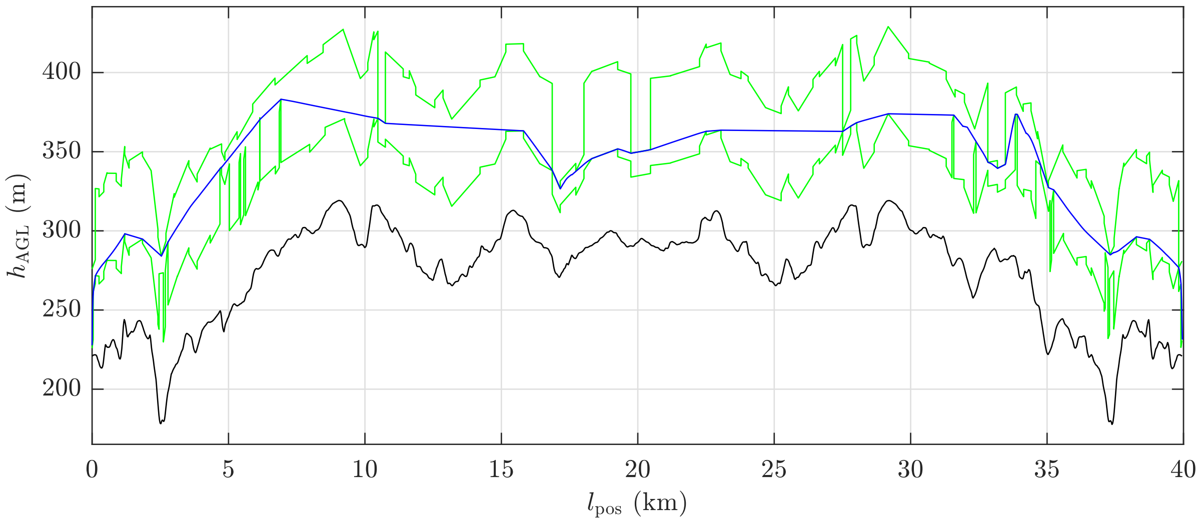
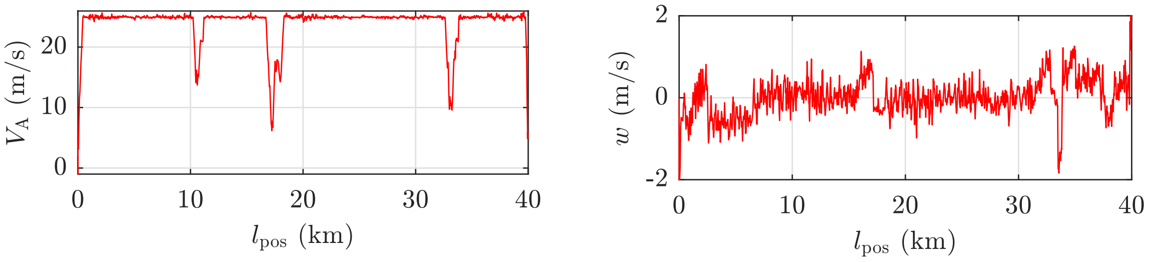


Fig. 11 Approximation for altitude range in green, ground elevation in black, resulting altitude command in blue



(a) Aerodynamic velocity

(b) Vertical velocity

Fig. 12 Speed commands

The resulting altitude command is represented by the blue line in figure Fig. 11. Additionally, the resulting vertical speed is shown in figure Fig. 12b. The vertical speed only reaches the limits of ±2 m/s during takeoff and landing. The effect of the guiding point can be seen especially in the area between $l_{\mathrm{pos}} = 7\,\mathrm{km}$ and $l_{\mathrm{pos}} = 10.5\,\mathrm{km}$: Given the state plan's horizon of 500 m, the aircraft would fly at constant altitude until just before $l_{\mathrm{pos}} = 10\,\mathrm{km}$. Only then would the steep change of the upper altitude limit come into sight. However, due to the guiding point a constant low sinking speed is commanded already from $l_{\mathrm{pos}} = 7\,\mathrm{km}$.

6 Conclusion

In this paper we proposed a set of methods to plan a trajectory for UAVs based on given waypoints, taking into account the aircraft's flight mechanics constraints as well as position dependent restrictions. While path planning can be carried out well before flight, our trajectory planning algorithm includes the consideration of wind, requiring periodic in-flight replanning.

The path planner converts a given list of waypoints into a sequence of path segments consisting of linear segments and circular arcs. Then, an allowed height band, consisting of an upper and a lower height limit, is determined. This height band has to meet various requirements, given by organizational constraints, like the operational authorization, or legal requirements, for example for nature reserves or power lines. Subsequently, the height band is combined with the ground elevation profile of the path to obtain altitude limits. Finally, these limits are approximated through linear segments.

In the trajectory planner the planned flight path, the current state of the aircraft, and wind conditions are used to precompute a series of flight states which will then be executed in the trajectory controller. Therein a coupled altitude and speed planning is carried out using receding horizon control, yielding a preplanned list of flight states with a defined altitude profile.

Using the presented method, an extensive flight campaign was executed, accumulating a total flight distance of more than 10 000 km. Based on one of these flight missions, the proposed method was validated in detail.

References

[1] Md. Abrar Istiak, M.M. Mahbubul Syeed, Md Shakhawat Hossain, Mohammad Faisal Uddin, Mahady Hasan, Razib Hayat Khan, and Nafis Saami Azad. Adoption of unmanned aerial vehicle (uav) imagery in agricultural management: A systematic literature review. *Ecological Informatics*, 78:102305, 2023. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2023.102305.

[2] He Ren, Yanling Zhao, Wu Xiao, and Zhenqi Hu. A review of uav monitoring in mining areas: current status and future perspectives. *International Journal of Coal Science & Technology*, 6(3):320–333, Sep 2019. ISSN: 2198-7823. DOI: 10.1007/s40789-019-00264-5.

[3] Reem Ashour, Sara Aldhaheri, and Yasmeen Abu-Kheil. *Applications of UAVs in Search and Rescue*, pages 169–200. Springer International Publishing, Cham, 2023. ISBN: 978-3-031-32037-8. DOI: 10.1007/978-3-031-32037-8_5.

[4] Evan Ackerman and Eliza Strickland. Medical delivery drones take flight in east africa. *IEEE Spectrum*, 55(1), 2018. DOI: 10.1109/MSPEC.2018.8241731.

[5] Matúš Tkáč and Peter Mésároš. Utilizing drone technology in the civil engineering. *Selected Scientific Papers - Journal of Civil Engineering*, 14(1), 2019. DOI: 10.1515/sspjce-2019-0003.

[6] Elena Politi, Panagiotis Rodosthenous, Ricardo dos Reis, Morten Larsen, Iraklis Varlamis, and George Dimitrakopoulos. Analysis of the new market trends of uavs for safe bvlos operations. *Journal of Physics: Conference Series*, 2716:012056, 03 2024. DOI: 10.1088/1742-6596/2716/1/012056.

[7] Johanna Holsten, Dagmar Huth, Norbert Siepenkötter, and Dieter Moormann. Highly automated bvlos drone operations: A large scale flight campaign for agricultural observation in saxony, germany. In *Proceedings of the 2024 CEAS Specialist Conference on Guidance, Navigation and Control (EuroGNC)*, Bristol, UK, Apr. 2024. DOI: TBD.

[8] Sebastian Seitz, Philipp Hartmann, Isabelle Barz, and Dieter Moormann. *Fault-Tolerant Flight Guidance System for UAS Enabling Full U-space Integration*. DOI: 10.2514/6.2022-1146.

[9] Tobias Islam, Sebastian Seitz, and Dieter Moormann. Automated mission generation and dispatching for bvlos drone operations. In *CEAS Conference on Guidance, Navigation and Control, EuroGNC, June 2024*, Bristol, UK, June 2024.

[10] Patrick Osterloh, Johannes Krimphove, Eva König, Tobias Islam, Johanna Holsten, and Dieter Moormann. Robust and fault-tolerant flight control architecture for bvlos drone operations. In *CEAS Conference on Guidance, Navigation and Control, EuroGNC, June 2024*, Bristol, UK, June 2024.

[11] Fabian Baader, Ann-Kristin Sturm, Philipp Müller, Johanna Holsten, and Dieter Moormann. Operational permit: Application for flexible flight operations in the eu-2019/947 specific category. In *CEAS Conference on Guidance, Navigation and Control, EuroGNC, June 2024*, Bristol, UK, June 2024.

[12] Yavor Dobrev, Nicolai Voget, Johannes Krimphove, Lennart Daniemeier, Sebastian Seitz, Johanna Holsten, and Dieter Moormann. A flight operation strategy for highly automated parallel bvlos operations. In *CEAS Conference on Guidance, Navigation and Control, EuroGNC, June 2024*, Bristol, UK, June 2024.

[13] European Environment Agency. European digital elevation model (eu-dem). https://www.eea.europa.eu/data-and-maps/data/eu-dem. Accessed: 2024-05-15.

[14] Eva König, Sebastian Seitz, Nicolai Voget, Lennart Danielmeier, and Dieter Moormann. Safety procedure using path planning methods for tilt-wing unmanned aerial vehicles. In *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 24–31, 2023. DOI: 10.1109/ICUAS57906.2023.10155850.

[15] Philipp Hartmann. *Vorausschauende Flugbahnregelung für Kippflügelflugzeuge*. Dissertation, RWTH Aachen University, Aachen, 2017. DOI: 10.18154/RWTH-2017-10495, https://publications.rwth-aachen.de/record/710450.

[16] Isabelle Barz. *Flugbahnplanung für Kippflügelflugzeuge zur Unterstützung im Rettungseinsatz*. Dissertation, RWTH Aachen University, Aachen, 2023. DOI: 10.18154/RWTH-2023-03181, https://publications.rwth-aachen.de/record/954410.