33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4–9 SEPTEMBER, 2022

ICAS 2022 SWEDEN

REQUIREMENTS-BASED GENERATION OF OPTIMAL VERTICAL TAKEOFF AND LANDING TRAJECTORIES FOR ELECTRIC AIRCRAFT

Christoph Krammer[1], Felix Schweighofer[1], Daniel Gierszewski[1], Simon Philipp Scherer[1], Tuğba Akman[1], Haichao Hong[1] & Florian Holzapfel[1]

[1]Technical University of Munich, TUM School of Engineering and Design, Department of Aerospace and Geodesy, Institute of Flight System Dynamics

Abstract

The introduction of urban air mobility by means of electric vehicles capable of vertical takeoff and landing represents a major transformation of civil aviation. As traffic density increases in urban areas, novel requirements are introduced to guarantee safe operations, most notably in the EASA SC-VTOL specification. We present an automated process for generating trajectories considering such regulatory requirements and associated operational procedures while optimizing for energy efficiency. The corresponding optimal control problems are derived automatically from a flexible trajectory definition database expressing the airspace constraints and operational requirements for vertical takeoff and landing trajectories in the vicinity of vertiports. A safety gateway verifies the feasibility of the generated trajectories.

Keywords: trajectory generation, trajectory optimization, electric vertical takeoff and landing

1. Nomenclature

AFGCS	Automatic Flight Guidance and Control System
AFM	Aircraft Flight Manual
ALM	Application Lifecycle Management
AMC	Accepted Means of Compliance
ConOps	Concept of Operations
CSFL	Continued Safe Flight and Landing
EASA	European Union Aviation Safety Agency
EUROCAE	European Organization for Civil Aviation Equipment
eVTOL	Electric Vertical Take-off and Landing
FAA	Federal Aviation Administration
FATO	Final Approach and Take-Off Area
FCC	Flight Control Computer
FCS	Flight Control System
FDM	Flight Dynamic Model
FGS	Flight Guidance System
FOV	Field of View
ICAO	International Civil Aviation Organization
ILS	Instrument Landing System
JSON	JavaScript Object Notation
MOC	Means of Compliance
MPC	Model-Predictive Control
NDI	Non-Linear Dynamic Inversion
NFE	Normal Flight Envelope
NLP	Nonlinear Program
ODE	Ordinary Differential Equation
OCP	Optimal Control Problem
PinS	Point-in-Space
PSU	Provider of Services for UAM
PTS-VPT-DSN	Prototype Technical Design Specifications for Vertiports

SC-VTOL	Special Condition for small-category VTOL aircraft
SDF	Signed Distance Function
SDSP	Supplemental Data Service Provider
STAR	Standard Terminal Arrival Route
UAM	Urban Air Mobility
VLL	Very Low Level
VTOL	Vertical Take-off and Landing

2. Introduction

The introduction of manned and unmanned Electric Vertical Take-off and Landing (eVTOL) aircraft for Urban Air Mobility (UAM) demands a high degree of automation to handle the increasing traffic density in Very Low Level (VLL) airspace and U-space [1]. Therefore, automatic generation and following of safe and energy-efficient flight trajectories becomes crucial. The energy reserve in battery-powered aircraft is challenged by the high power output during the Vertical Take-off and Landing (VTOL) phases. To ensure safe, economic and sustainable operations, accurate and robust flight planning considering the system performance and uncertain environmental conditions is required. The approach/descent and departure/climb legs in the vicinity of vertiports are of particular interest. These not only cause a significant share of the energy consumption, but also include the most complex flight path geometries due to VTOL operations within urban environments. This complexity obviously necessitates improved methods for flight planning and trajectory optimization compared to the rather simple Standard Terminal Arrival Routes (STARs), straight Instrument Landing System (ILS) approaches commonly followed by fixed-wing and rotary-wing aircraft, or Point-in-Space (PinS) helicopter operations.

Trajectories may be planned offline before takeoff or online during flight. We follow the approach of offline trajectory generation combined with online trajectory selection. The main advantage of offline trajectory generation is that the results may be verified and validated before flight to ensure that the trajectories adhere to given requirements. Therefore, the options and paths available to the Automatic Flight Guidance and Control System (AFGCS) are pre-determined, leading to more deterministic behavior compared to online trajectory generation. Additionally, high computational power is required to solve trajectory optimization problems as presented in this paper, which is usually not available during flight. Online trajectory selection by the Flight Guidance System (FGS) allows to react to changes in internal or external circumstances. For example, trajectories for contingency maneuvers and Continued Safe Flight and Landing (CSFL) may be triggered in case of internal failures like propulsive degradation or performance losses. In this case, trajectories optimized for minimal flight time instead of energy consumption may be selected by the FGS. External factors that lead to a change in the flight path could be the unavailability of an UAM corridor [2] / airspace or changing environmental conditions such as wind.

In this paper, we present our design of a rule-based trajectory generation process, whose overall architecture is described in Section 3. The generation of trajectories adheres to practical requirements, stemming from sources that have been recently published as regulatory documents and standards. Section 4 states these sources of requirements for eVTOL operations and how they are imposed on the trajectories in our process. The task of generating trajectories relies on a newly proposed trajectory definition that features extensive information and portability among scenarios. The compatibility to models of different fidelity is one of the major features concerning the model integration in the proposed process. The trajectory optimization software interfaces the trajectory definitions, the requirements, and the Flight Dynamic Model (FDM) and generates optimal trajectories in terms of the desired objective in an automated manner. All trajectories can be verified by the gateway function that checks several safety-critical requirements in a closed-loop simulation. The trajectory definition database, the FDM, the optimal control approach, and the verification of results are described in Sections 5 to 8, respectively. We present an application example in Section 9. Section 10 summarizes the current state of our system and further steps toward practical implementation.

3. Trajectory Generation Architecture

Figure 1 gives an overview of the developed trajectory generation process. Requirements for the trajectories are derived from regulatory documents, standards and further sources such as a Concept of

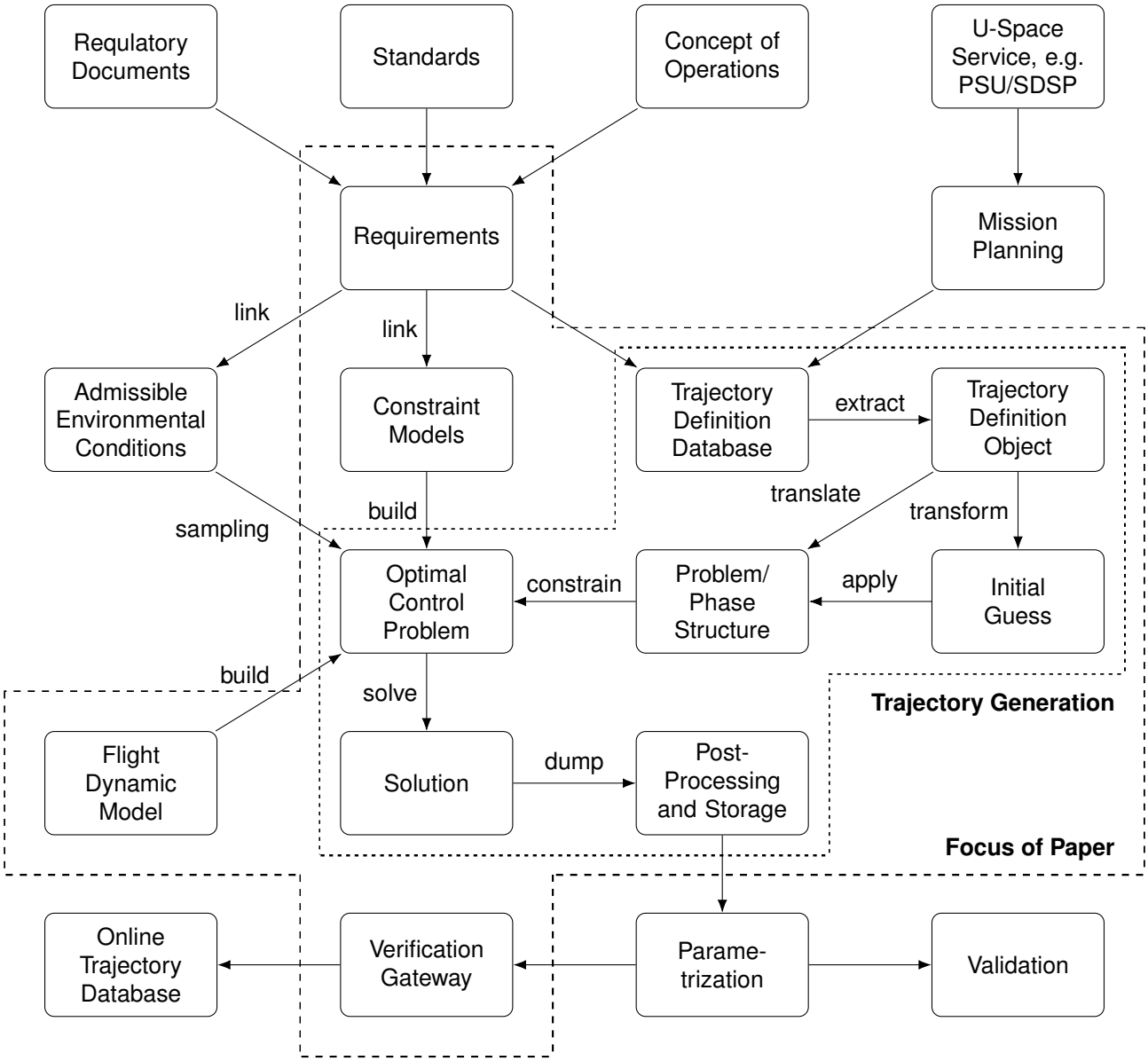

Figure 1 – Architecture of the Trajectory Generation Process

Operations (ConOps). Constraint models ensure that the requirements are imposed on the trajectories, and the environmental conditions for a FDM of the target vehicle are bound by admissible operating limits. A database of trajectory definitions expresses generic flight plans for approach/departure and cruise legs, i.e., the flight phases and associated constraints derived from the requirements. The database is created as part of the mission planning process, which may rely on information from future U-Space services such as a Provider of Services for UAM (PSU) or a Supplemental Data Service Provider (SDSP) [2].

We automatically translate the trajectory definitions into Optimal Control Problems (OCPs) for the FDM with the goal of minimizing the electric energy consumption for the eVTOL mission. These OCPs are modeled in our in-house optimal control MATLAB code FALCON.M[1] [3] and solved for various sets of environmental conditions. For reasons detailed in Section 7.4, a suitable parameterization of the resulting time-descretized trajectories is still under development and beyond the scope of this paper. After an offline feasibility and safety verification by a gateway, the parameterized trajectories are stored in an onboard database. Depending on the current environmental conditions, the FGS selects appropriate trajectories from the database, which are then passed through an online

[1] http://www.falcon-m.com/

verification gateway before serving as a reference for the trajectory-following controller.

4. Requirements

Requirements for trajectory generation may stem from various sources. For our process, we take recently published regulatory documents and standards into account, which have been released to pave the way for UAM and eVTOL operations. Most notably, the European Union Aviation Safety Agency (EASA) published the Special Condition for small-category VTOL aircraft (SC-VTOL) [4] and related Accepted Means of Compliance (AMC) [5]. To regulate the design of vertiports and their vicinity, EASA released the Prototype Technical Design Specifications for Vertiports (PTS-VPT-DSN) [6]. For example, Means of Compliance (MOC) VTOL.2115 [5] and PTS-VPT-DSN.D.445 [6] define performance limits and safety areas during takeoff and landing by means of admissible reference volumes. We are using these three-dimensional funnel-shaped objects to constrain the flight path of the vehicle, leading to trajectories that stay clear of the borders of the reference volume. Since funnel dimensions in PTS-VPT-DSN.D.485 [6] additionally include a safety area around the reference volume (to incorporate obstacle clearances around vertiports), we are using the more restrictive reference volume type 1 from MOC VTOL.2115 [5]. Furthermore, MOC VTOL.2115 [5] defines constraints for takeoff path segments such as heights, speeds and propulsive power settings. These prescriptions are also taken into account for trajectory generation.

In the USA, the Federal Aviation Administration (FAA) published the UAM ConOps [2], which envisions flights in urban areas and the necessary (technical) environment allowing such operations. In addition, the FAA is working on regulatory documents to accommodate eVTOL aircraft. For the mission planning step, we are taking the UAM concept [2] into consideration. For example, the trajectory may be constrained to lie within a certain tube of airspace representing an UAM corridor. Furthermore, committees such as European Organization for Civil Aviation Equipment (EUROCAE) working groups are advancing the standardization of future eVTOL vehicles and operations [7], which may contribute to the set of requirements that has to be considered for trajectory generation.

In addition, constraints or limitations from other sources such as the Aircraft Flight Manual (AFM) or a ConOps document need to be accounted for. For example, the admissible dynamics of the vehicle may be limited to meet passenger comfort criteria defined in the ConOps. For this purpose, we propose to introduce the concept of a trajectory generation envelope that is more restrictive with respect to, e.g., acceleration and angular rates than the admissible limits for passenger comfort or the Normal Flight Envelope (NFE) (MOC VTOL.2135 [5]). This reserves some remaining aircraft dynamics for path correction or to counteract disturbances when flying the generated trajectories without violating the passenger comfort limits or NFE. Further requirements for a specific example are listed in Section 9.2.

For our developed trajectory generation process (refer to Figure 1), the textual requirements are handled with the Application Lifecycle Management (ALM) software POLARION[2] and a customized requirements management template [8]. After populating the POLARION project with relevant information from the aforementioned sources, requirements specific to the generation of trajectories are derived and collected in a new system specification document. To account for the requirements in the OCP, they are formalized as point- or phase-specific in-/equalities and implemented as constraint models in SIMULINK. By using various POLARION work item types and link roles [8], traceability between top-level (e.g. regulatory) requirements and the lower level trajectory generation system specification is ensured. Furthermore, the in-house tool SIMPOL[3] [9] allows to establish links between requirements in the POLARION system specification and the SIMULINK constraint models. Therefore, there is full traceability between the original source of information and the requirement imposed on the OCP. The implementation of the constraint models in SIMULINK offers the advantage that the same models (with possible changes in the limit parameters or data types) may be used for an online AFGCS monitor in conjunction with a Flight Control Computer (FCC) code generation toolchain [10].

[2] https://polarion.plm.automation.siemens.com
[3] https://www.fsd.ed.tum.de/software/simpol

5. Trajectory Definition Database

As current standards for trajectory definition and navigation databases such as ARINC 424 [11] are not targeting eVTOL specifics or trajectory optimization methods, we seek to define trajectories in a custom form that lends itself well to the automatic derivation of OCPs, including the associated initial guesses. Trajectory definitions therefore include extensive information on the constraints imposed on the individual flight phases as well as on specific points of the trajectory. Ideally, the definitions are formulated in a purely semantic and aircraft-independent manner, allowing us to swap the FDM and specialized implementations of the constraints when generating trajectories for a specific aircraft type. Furthermore, our trajectory definitions are abstractions of the specific situation at individual vertiports; they are only indirectly referenced to specific positions and orientations with respect to earth. The intention behind this is to allow the reuse of generated trajectories for multiple vertiports with compatible obstacle environments and approach/departure procedures, thereby reducing computational effort and required storage size. This becomes particularly important when considering varying environmental conditions, an issue that we will address in the future. In a later stage of the project, we aim to close the gap between our trajectory definitions and established schemes such as ARINC 424 [11]. This may be achieved either by interfaces and transformations, or by extending the navigation database schemes with new elements.

Our trajectory definitions are modeled in an object-oriented manner as shown in Figure 2. A trajectory comprises one or more phases, which translate directly to the structure of a multi-phase OCP. These phases often correspond to route segments between waypoints, but are defined in a more general way. Essentially, a phase (as defined by Betts [12]) represents a continuous trajectory segment with given dynamics defined on a compact interval of the independent variable, which is time in our setting. While the phase duration may be subject to optimization, time progresses uniformly within a phase, which is why waypoint constraints and arrival time windows typically need to be imposed on the phase boundaries.

Each phase may be associated with constraints imposed on the phase start and end points (single point constraints) as well as on every point of the trajectory segment (path constraints). Constraints imposed on the phase boundaries include restrictions on the aircraft position, which are given by a reference point (waypoint or vertipad) with horizontal and vertical tolerances, as well as constraints enforcing aircraft-specific requirements at liftoff and touchdown (refer to Figure 2). The latter are implemented in generic constraint models with aircraft-specific parameters, while the database only expresses the generic semantics. For the purpose of this paper, the term vertipad denotes the Final Approach and Take-Off Area (FATO) at the destination vertiport. The liftoff and touchdown constraints reference a vertipad, whose position and orientation are passed on to the constraint implementation when assembling the OCP. Path constraints express the limitations prescribed for the different departure and approach flight phases. These typically reference a vertipad and the type of safety volume to be respected in the corresponding flight phase. The implementation of constraints for these flight phases is again independent of the database.

Additionally, obstacle clearance and FATO visibility constraints may be prescribed in individual phases. These refer to an obstacle environment defined on the trajectory phase level, which comprises a list of obstacles relevant for the specific flight phase. The obstacle clearance constraint guarantees that a given minimum distance to all obstacles is maintained at all times, whereas the FATO visibility constraint can be applied to ensure compatibility with vision-augmented automatic landing systems (refer to Section 9).

The trajectory definition is completed by a set of objective terms, which are specified for every individual phase. The objective is a weighted sum of the flight duration and energy consumption. Usually the coefficients should be the same for all phases.

Currently, the database objects are stored in a human-readable and portable JavaScript Object Notation (JSON) format. A simple variable substitution mechanism allows the user to define recurring elements at the root level of a trajectory definition and reference them from any subordinate objects, thereby reducing information duplication and avoiding inconsistencies. For example, waypoints or vertipads are defined once per trajectory, and can then be referenced in the constraints of the individual phases. Additionally, vertiports, waypoints and obstacles are each defined in separate global

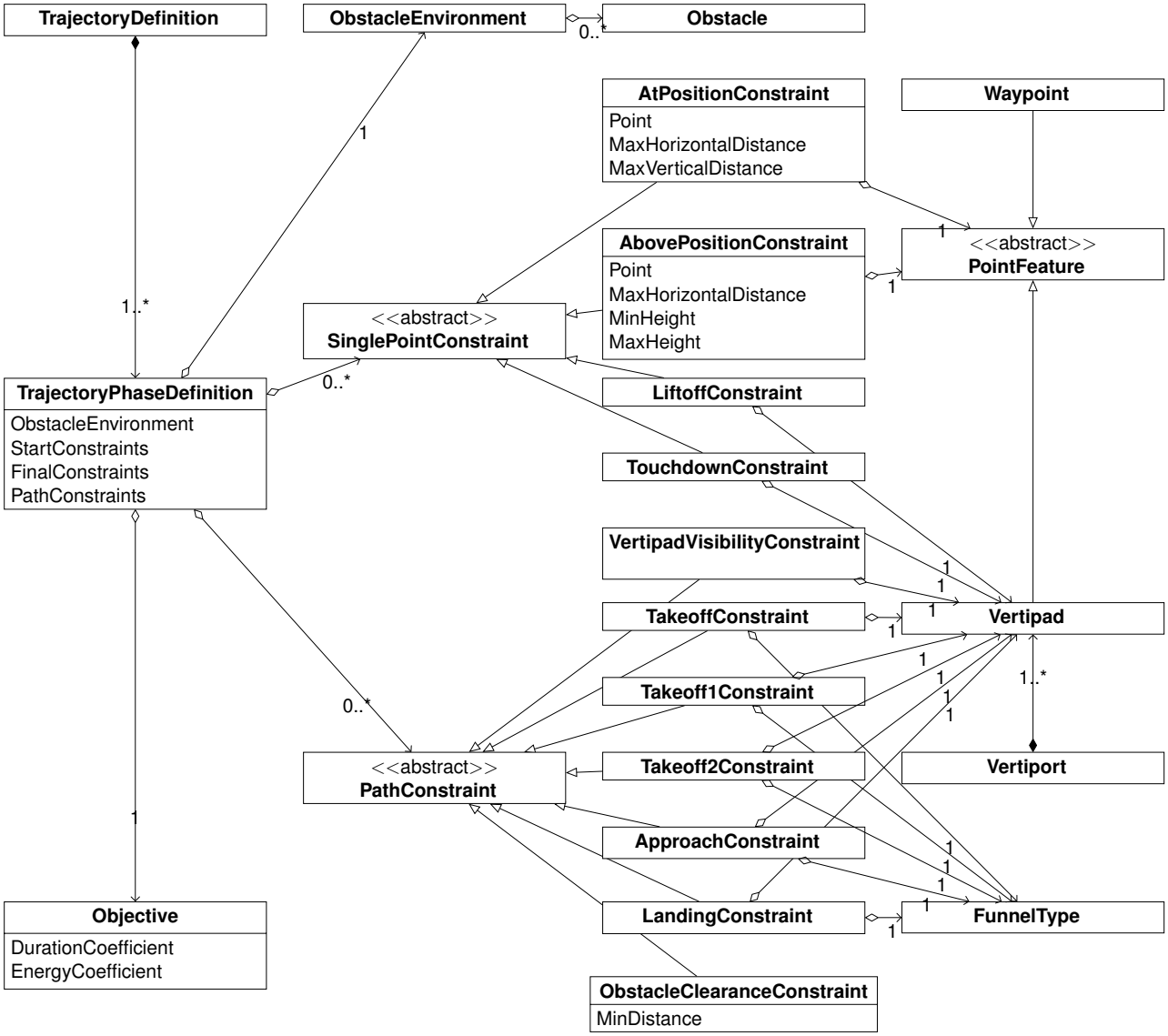

Figure 2 – Simplified Trajectory Definition Database Structure

JSON files, and referenced by their unique identifiers in the trajectory definitions. When loading the trajectory definition, these substitutions are applied before any further processing.

6. Flight Dynamic Model

Our framework is designed to work with nonlinear rigid-body FDMs. We formulate the position in a local Cartesian reference frame to facilitate the reuse of generated trajectories at multiple vertiports with compatible obstacle environments. Additionally, this improves the numerical characteristics of the OCPs compared to geographic coordinates. The assumption of a flat-earth model for the purpose of trajectory generation is justified under the consideration of short flight distances due to the limited energy reserves in battery-powered eVTOL vehicles. Since extreme attitudes are neither expected nor permitted we express the attitude by roll, pitch and yaw angles. To account for wind influence, which is expected to be significant in terms of safety margins, particularly in vertical climb or descent and with respect to energy efficiency, our model includes a constant wind field.

The FDMs may represent either the closed-loop dynamics to be provided by the Flight Control System (FCS) as reference/surrogate models formulated on a kinematic level, or the physical aircraft dynamics. Using reference/surrogate models, which specify or represent the closed-loop dynamics without modeling all subsystems of the physical plant, may offer performance benefits and allows to decouple the resulting optimal trajectories from the uncertain plant behavior. However, this approach is complicated by the need to calculate an accurate power estimate to allow for energy optimization.

Since this is inherently linked to a model of the propulsion system, we postpone this abstraction to future works.
Models can be implemented in SIMULINK with some restrictions. After eliminating discrete aspects and expressing the equations of motion as an explicit first-order Ordinary Differential Equation (ODE) without any dynamic elements (such as integrators, delays, transfer functions, etc.), the SIMULINK model can be interfaced with our optimal control code in MATLAB as detailed in Section 7. This allows us to reuse existing model components developed for control design.

7. Trajectory Generation and Optimization

7.1. Automated Problem Setup

Trajectory definitions from the database described in Section 5 directly translate to multi-phase OCPs. We implement these automatically in our in-house MATLAB-based object-oriented optimal control tool FALCON.M [3].

7.1.1. Aircraft Performance and Flight Phase Constraints

Constraint implementations with aircraft-specific parameters for various flight phases are implemented as SIMULINK models and linked to the requirements from Section 4. These constraint models include physical and operational performance limitations applicable to the specific aircraft in specific flight phases, for example path constraints applicable during the takeoff or landing phases. An extended version of FALCON.M enables us to generate function interfaces suitable for use in numerical optimal control directly from the SIMULINK implementations of the dynamics and constraints with little manual effort. Essentially, the SIMULINK Embedded Coder toolchain is used to generate a shared library efficiently evaluating the equations expressed in the SIMULINK model. FALCON.M then builds a wrapper around this library to efficiently calculate sparse finite difference Jacobian and Hessian matrices for multiple evaluation points in parallel, automatically detecting the sparsity pattern at runtime.
Among others, we impose the following constraints:

- Aerodynamic and kinematic velocity limits
- Linear and angular acceleration limits for structural integrity, passenger comfort and payload safety
- Attitude limits: It is interesting to note that according to our studies with a multicopter aircraft model with a high degree of rotational symmetry, an omnidirectional limit on the tilt angle is strongly preferable to roll and pitch angle limits. The latter can result in strong yaw oscillations if the optimization algorithm uses combined roll and pitch maneuvers to achieve higher overall tilt angles and therefore forward thrust force components.
- Liftoff and touchdown velocity and attitude point constraints
- Funnel-shaped safety volumes for takeoff and landing at FATOs
- Cruise velocity and heading constraints at the start point of an approach trajectory or at the end point of a departure
- Field of view constraints for vision-augmented automatic landing systems, ensuring that the aircraft orientation allows the onboard cameras to see the FATO at the destination vertiport during final approach (refer to Section 9)

To ensure the validity of the generated trajectories even for possible aircraft system failures and accompanying performance losses, the process needs to account for worst-case degraded aircraft performance. This may be achieved by limiting the characteristics of the FDM, leading to trajectories optimized for degraded aircraft performance. Alternatively, a backward reachability analysis may be conducted to constrain the solution of the OCP. The consideration of aircraft system failures is subject to future work.

7.1.2. Obstacle Clearance Constraint

The generic obstacle clearance path constraint is implemented as a Signed Distance Function (SDF) with respect to the obstacles referenced by the phase definition. The SDF representation is an abstraction that allows us to transparently handle arbitrary obstacle geometries; however, due to limited computational resources we currently consider only box obstacles, as these are the most accurate primitive representation of the buildings commonly found in the vicinity of anticipated vertiport locations. An overview of SDFs for various geometric primitives is given by Quilez [13].

7.1.3. FATO Visibility Constraint for Vision-Augmented Flight Guidance

FATO visibility is currently modeled by a minimum height expressed as a function of the horizontal position. The minimum height, or visibility surface, is identified using a variant of the classic `R3` algorithm for viewshed calculation [14]. We consider the FATO to be visible if and only if the straight line connecting the aircraft reference point and the vertipad center point is unobstructed. While representing the visibility constraint as a minimum height surface is quite efficient, the constraint boundary is discontinuous when non-smooth obstacle heights are considered. Accordingly, a grid discretization with high resolution has large gradients at the constraint boundary, leading us to expect numerical issues in scenarios where the aircraft needs to pass through a narrow corridor due to the line of sight being obstructed on both sides. As a side note, this visibility model is only applicable to ground-based obstacles. In the future, we plan to switch to a three-dimensional viewshed based on a distance-aided ray marching technique currently under development. This will directly use the SDF of the obstacle environment to determine visibility, thereby eliminating alternate obstacle representations in our system. At the same time, the three-dimensional visibility model will eliminate the potential issues with non-smooth obstacle heights and grid discretization, while adding full support for obstacles with no direct connection to the ground. Furthermore, we may consider the sensitivity of the line of sight with respect to aircraft position to increase robustness by ensuring sufficient visibility even if the real trajectory slightly deviates from the nominal plan.

7.1.4. Objective

The objective is specified as a weighted sum of phase durations and energy consumptions. The phase duration terms are implemented as differences of the phase start and end times, all but the first of which are decision variables in the generated FALCON.M problem. Energy consumption terms are evaluated by integrating the power output of the aircraft over each phase.
We augment these objective terms by small penalties to suppress undesirable oscillations. These do not significantly affect the resulting objective values but increase the perceived quality of the solutions as well as the convergence behavior. To increase the robustness against variations in the environmental parameters, the sensitivity of the OCP solution to the parameters may be considered in future versions of the toolchain.

7.2. Initial Guess Generation

The semantic constraint definitions allow us to automatically determine a rough initial guess for essential aircraft states, in particular the position, heading and velocity. Waypoint and vertipad positions specified at phase boundaries determine the initial approximation of the flight path. If an interior phase boundary is not associated with a known point, we interpolate linearly between all given locations. Headings are estimated from the track angle along the position guess. This can be extended to account for the wind field in the future. The velocity, if not prescribed explicitly by a cruise velocity constraint or implicitly by a touchdown/liftoff constraint, is estimated based on aircraft-specific values and the slope of the trajectory segment.
From the position, heading and velocity we also obtain an estimate of the phase durations. Most other states and controls are currently initialized to constant values. In future versions of the toolchain, we intend to enhance the initial guess generation procedure. Candidate approaches include graph-based optimizations and estimation of additional aircraft states by Kalman filtering. Ideally, an improved initial guess would allow us to obtain a reasonable estimate of the order of magnitude of the objective, to be used for automatic scaling of the optimization problem.

7.3. Solution

FALCON.M applies a trapezoidal collocation method to transcribe the infinite-dimensional OCP into a finite-dimensional Nonlinear Program (NLP). This standard parameter optimization problem is then solved by the filter line-search interior-point algorithm IPOPT [15], a general-purpose Quasi-Newton method for large and sparse problems. Within the Quasi-Newton iterations, large-scale linear systems are solved by the HSL routine `ma97` [16].

A major difficulty of this approach to optimal control is the inherent unreliability and numerical fragility of nonlinear nonconvex optimization. To apply these methods in an automated system, we need to choose discretization grids and numerical scaling in a robust way to achieve good convergence behavior. Currently, our implementation uses heuristics based on our engineering experience to determine suitable values for the discretization and scaling from the automatically generated initial guess, as well as fixed manually chosen scale factors for the aircraft states, controls and outputs. In the future, these procedures need to be improved to achieve robust convergence over a large range of scenarios. In any case, extensive safety verification is required, even though both aircraft limitations and operational constraints are considered by the trajectory optimization.

7.4. Parameterization

The trajectory optimization toolchain produces discretized state and control histories that are not necessarily suitable for onboard storage and trajectory-following control. This is because the discretized signals may have an unnecessarily high resolution in some regions, thereby increasing storage size, and also because the trajectory-following controller may require derivatives of arbitrary order as feedforward commands. Thus, we require an efficient parameterization of the optimal trajectories suitable for onboard storage and real-time evaluation on a FCC. The parameterization should be infinitely smooth to allow for the derivation of feedforward commands at high derivative orders without assuming a certain controller structure a priori. Additionally, kinematic accuracy must be assured, in particular regarding the flight path.

Candidate approaches include the trigonometric series parameterizations successfully demonstrated by Hong *et al.* [17], [18]. Additionally, convolution-based parameterizations currently under investigation may allow us to decouple the smoothness aspect from accuracy requirements. This part of the toolchain is still under development. The issues outlined in this section will be addressed by future works.

8. Verification of Generated Trajectories

The optimal control framework FALCON.M, which is used to generate the trajectories, has been developed without performing the required activities to ensure compliance with aerospace standards and guidelines, e.g., SAE ARP4754A [19], RTCA DO-178C [20] and RTCA DO-331 [21]. A qualification of the trajectory optimization framework is considered difficult and cost prohibitive due to the complexity of the deployed iterative, non-linear optimization algorithm. Instead of qualifying the software, a gateway function is introduced as proposed by [22] which functionally decouples the optimization framework from the safety-critical flight control software by verifying the optimization results before forwarding them to the AFGCS. The gateway function may thereby reduce the costly qualification burden and pave the way to apply complex iterative or even non-deterministic methods for flight trajectory planning.

The gateway function performs a closed-loop simulation of the trajectory tracking task to verify requirements related to aircraft integrity, air risk and ground risk. The simulation model allows to incorporate available information on aircraft performance, airspace restrictions and air traffic conditions. Each requirement is formulated as an inequality constraint, which is evaluated at every simulation step and has been developed independently of the constraint models used for trajectory generation. In case a constraint is violated, the simulation is aborted and the gateway function rejects the planned trajectory. The verification is performed offline before storing the trajectory data in the database and also online before forwarding the trajectory data to the flight control software. Online verification onboard the aircraft allows to account for current information on atmospheric conditions (e.g., wind, air density), aircraft system states (e.g., mass, battery state of charge) and operational constraints (e.g., airspace restrictions).

The gateway function has been developed as safety-critical software using model-based design with MATLAB SIMULINK and STATEFLOW, following the process described in [10]. The basis of the simulation model is a nonlinear, six degrees of freedom FDM, which is implemented independently of the FDM used for the trajectory generation and optimization task. The rotational inner-loop dynamics as well as the thrust dynamics are approximated by surrogate models describing the nominal system response to commands provided by the trajectory controller. The utilized trajectory controller is based on Non-Linear Dynamic Inversion (NDI) of the error dynamics between the aircraft and a reference trajectory [23]. Besides the lateral and vertical deviations to the reference trajectory and their time-derivatives, the trajectory controller requires the trajectory angles and their higher order derivatives as command inputs. These command inputs are calculated based on the geometry of the reference trajectory and the trajectory reference point, which is defined as the closest point on the reference trajectory with respect to the current aircraft position. The reference point is calculated by numerically solving a differential equation for the trajectory running parameter using information on the current aircraft position, kinematic acceleration and velocity [24]. The ODE solver used for numeric flight simulation is based on a second-order, predictor-corrector scheme, denoted as `RTAM2` [25]. The choice of the ODE solver was made to achieve a trade-off between numerical accuracy, computational burden and code complexity. Future work is seen in incorporating model uncertainties to enable a risk-based decision making regarding the feasibility of the planned trajectories.
In addition to the verification of the trajectories by means of the described gateway, validation steps are recommended before using the trajectories for flight to ensure that the trajectories are reasonable. For example, validation can range from manual inspection of the trajectories to subset simulations [26], [27] in order to estimate the exceedance of safety margins or the NFE under the consideration of environmental disturbances.

9. Examples

9.1. Aircraft Model

For specific examples, we are applying the described trajectory generation process to the publicly funded C2Land project, a collaboration between partners from academia and industry with the goal of providing a vision-augmented automatic landing system for VTOL vehicles [28]. As aircraft, we consider a nonlinear rigid-body FDM of an octadecacopter with distributed electric propulsion. In the current state of our application project, the model represents the dynamics of a physical aircraft (compared to a reference/surrogate model), with assumptions for aerodynamic and propulsive parameters. Based on a detailed SIMULINK model for control design, some simplifications are introduced for optimal control purposes. For example, the rotor speed is assumed quasi-steady and we reduce the control space by assigning the same rotor speed to groups of three rotors each, thereby approximating the system as a hexacopter.

9.2. Requirements

As example scenarios, hypothetical eVTOL operations at Munich airport (ICAO-code EDDM) are considered. We introduce an imaginary vertiport in a central location between the terminal buildings. The mission planning step yields trajectory definitions under the consideration of local circumstances and provisions of EASA SC-VTOL [4]. Each example trajectory consists of multiple flight phases with varying requirements. Table 1 gives an overview of all phase constraint models used for trajectory generation, the restricted states and the origin of the requirements. In addition to the funnel-shaped EASA reference volume [5] around the FATO (refer to Section 4), a preliminary trajectory generation envelope from the ConOps specific to the C2Land project is introduced. Due to the specific aircraft dynamics and mission scenarios, we combined the EASA takeoff segments 1 and 2 [5] in one constraint model. In addition, the propulsive power setting for all flight phases is limited to maximum continuous power. During the initial takeoff and landing phases, additional states are limited to increase the safety level when operating close to the ground. The planned missions stored in the trajectory definition database lead to additional point constraints, e.g., for the aircraft position at the transition of flight phases (for example end of takeoff segment 2) or the heading when landing at or taking off from the vertiport.

Table 1 – Overview of Phase Constraints for Trajectory Generation

Flight Phases	Constrained Signals	Sources (Rationale)
All Phases	Acceleration	ConOps (trajectory generation envelope)
	Attitude	ConOps (trajectory generation envelope)
	Angular Rate	ConOps (trajectory generation envelope)
	Angular Acceleration	ConOps (trajectory generation envelope)
Initial Takeoff	Position	SC-VTOL MOC VTOL.2115 (reference volume)
	Vertical Velocity	ConOps (safety)
	Attitude	ConOps (safety)
Takeoff Segments 1 and 2	Height	SC-VTOL MOC VTOL.2115 (takeoff path)
	Velocity	SC-VTOL MOC VTOL.2115 (takeoff path)
	Vertical Velocity	ConOps (trajectory generation envelope)
Cruise	Vertical Velocity	ConOps (trajectory generation envelope)
Approach	Position	SC-VTOL MOC VTOL.2115 (reference volume)
	Vertical Velocity	ConOps (trajectory generation envelope)
	Attitude	ConOps (optical Field of View (FOV))
Landing	Position	SC-VTOL MOC VTOL.2115 (reference volume)
	Vertical Velocity	ConOps (safety)
	Attitude	ConOps (safety)
	Attitude	ConOps (optical FOV)

Within the C2Land project, onboard optical systems are utilized to verify the relative position to the FATO in order to increase the integrity of the navigation solution. Therefore, it has to be ensured that the vertiport is visible to the optical system throughout the approach and landing phases, leading to additional constraints for the trajectory. Based on a set of relevant obstacles modeled in the trajectory definition database, we impose a FATO visibility constraint (Section 7.1.3), which restricts the flight path, and aircraft-specific field of view constraints, which restrict the attitude during the approach phase. The obstacle environment is further considered by an obstacle clearance constraint (refer to Section 7.1.2). During the landing phase, both FATO visibility and obstacle clearance are guaranteed by the prescribed safety volume at the vertipad, so the additional constraints can be omitted.

9.3. Departure Trajectory

The first example is a departure from the hypothetical vertiport at EDDM with a destination in Munich. We assume an initial climb in the eastern direction (positive direction of the plotted y-axis of the navigation frame) followed by a departure leg towards the south, crossing runway 08R/26L. Figure 3 shows the flight path for a minimum-energy departure. The aircraft attitude is indicated by small red, green and blue lines aligned with the body-fixed reference axes (forward, rightward, downward). The required obstacle clearance is shown by a green isosurface of the SDF calculated from the red box obstacles. FATO visibility is considered not relevant for the departure. At the vertipad, a geometric constraint based on the EASA reference volume [5] is imposed (refer to Section 4). The next two phase boundaries correspond to the climb segments described in EASA SC-VTOL MOC [5]; their position is subject to optimization. The final point is a prescribed waypoint, at which a given speed and flight direction must be reached to complete the departure before switching to cruise flight.

9.4. Landing Trajectory

The second example is an energy-minimal approach to the hypothetical vertiport at EDDM. Arriving from the city center located south west of the airport, the aircraft crosses runway 08R/26L from the south. We assume that arrivals and departures use separate routes and thus design a final approach from the west. We consider only the approach and landing phases in this example. Figure 6 shows that there is a corridor to the west of the tower (negative direction of the plotted y-axis of the navigation frame) where the hypothetical FATO is visible from a low altitude. The aircraft follows this corridor and the additional FOV constraints ensure that the onboard cameras can track the FATO visually during

approach. The landing phase is again constrained by the reference volume based on EASA SC-VTOL MOC [5] (refer to Section 4).

10. Conclusion and Future Work

This paper presents a toolchain to generate requirements-based energy-optimal vertical takeoff and landing trajectories. Each trajectory is optimized for a set of environmental conditions and stored in a database for selection by the online FGS. The core components are a trajectory definition database, followed by a trajectory optimization that takes requirements based on EASA SC-VTOL [4] into account, and a trajectory verification gateway ensuring their feasibility. To extend this toolchain, the following aspects are subject to future research.

The efficient and accurate generation of initial guesses based on the trajectory database and on given, possibly simplified, aircraft dynamics is a line of work under investigation. Major challenges include accounting for constraints and predicting the right velocity profile for nontrivial VTOL flight path geometries. The latter is further complicated by the wind field, which will be considered in the future. As long term goal, robust optimality with respect to uncertain environmental conditions and system parameters may be considered, as done in [29] for a VTOL departure under less complicated constraints. For the FATO visibility constraint, we are planning to switch to a three-dimensional viewshed. As outlined in Section 7.4, a suitable trajectory parameterization for the onboard database and FCS is under development. The verification gateway may be extended by the consideration of model uncertainties to allow risk-based decision making. Additionally, the in-house subset simulation toolchain [26], [27] may be applied to estimate the probability of exceedance of safety margins under consideration of environmental disturbances or plant parameter uncertainties when following the generated trajectories.

Beyond the described trajectory generation process, the next steps involve the finalization of the AFGCS for online trajectory-following control and validation in flight tests. In the long run, steps toward local refinement by onboard Model-Predictive Control (MPC) may be taken. This would allow the aircraft to perform more efficiently in case of deviations from the assumed environmental conditions and/or system degradation.

11. Acknowledgments

We would like to thank our colleagues Pranav Bhardwaj, Niclas Bähr, Maximilian Söpper, Rasmus Steffensen, Chinmaya Mishra, as well as our project partners from the Technical University of Braunschweig, f.u.n.k.e. AVIONICS GmbH and Volocopter GmbH for their valuable contributions to the C2Land project.

The presented work is part of the project C2Land, which is funded by the German Federal Ministry for Economic Affairs and Climate Action on the basis of a decision by the German Bundestag and administered by the DLR Space Administration (FKZ 50NA2004).

Supported by:

Federal Ministry for Economic Affairs and Climate Action

on the basis of a decision by the German Bundestag

12. Author Contributions and Contact Information

Feel free to contact the authors for questions or feedback:

- Christoph Krammer, christoph.krammer@tum.de: Process architecture, requirements, constraint models, trajectory definition database, flight dynamics model; initial manuscript
- Felix Schweighofer, felix.schweighofer@tum.de: Trajectory definition database, optimal control, obstacle constraints, trajectory parameterization; manuscript extension
- Daniel Gierszewski, daniel.gierszewski@tum.de: Trajectory verification, safety gateway; manuscript extension and review
- Simon Scherer, simon.scherer@tum.de: Flight dynamics model; manuscript review
- Tuğba Akman, tugba.akman@tum.de: Optimal control; manuscript review
- Haichao Hong, haichao.hong@tum.de: Trajectory parameterization; manuscript review
- Florian Holzapfel, florian.holzapfel@tum.de: Process architecture, holistic strategy

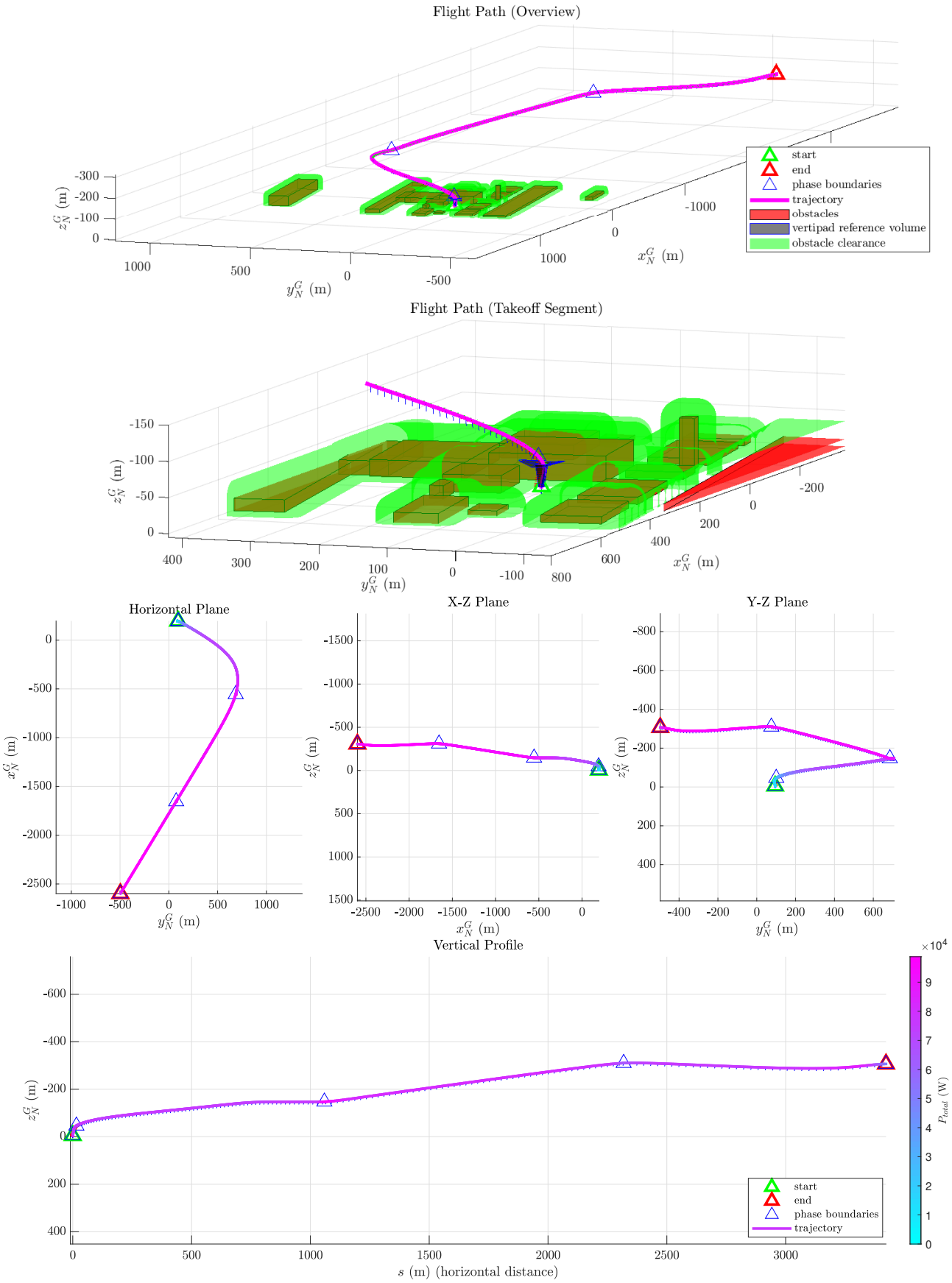

Figure 3 – Generated Trajectory for Automatic Takeoff – Flight Path

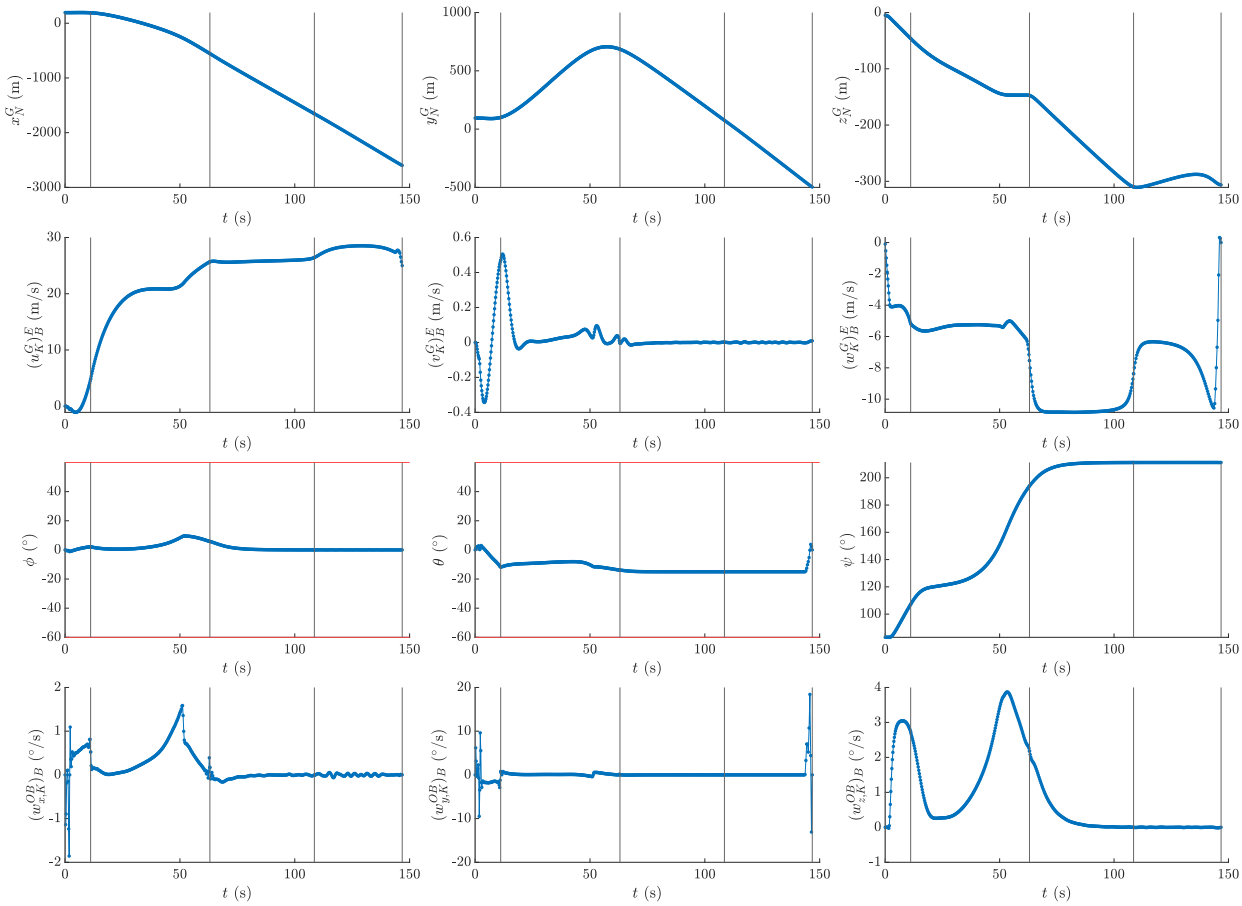

Figure 4 – Generated Trajectory for Automatic Takeoff – Aircraft States

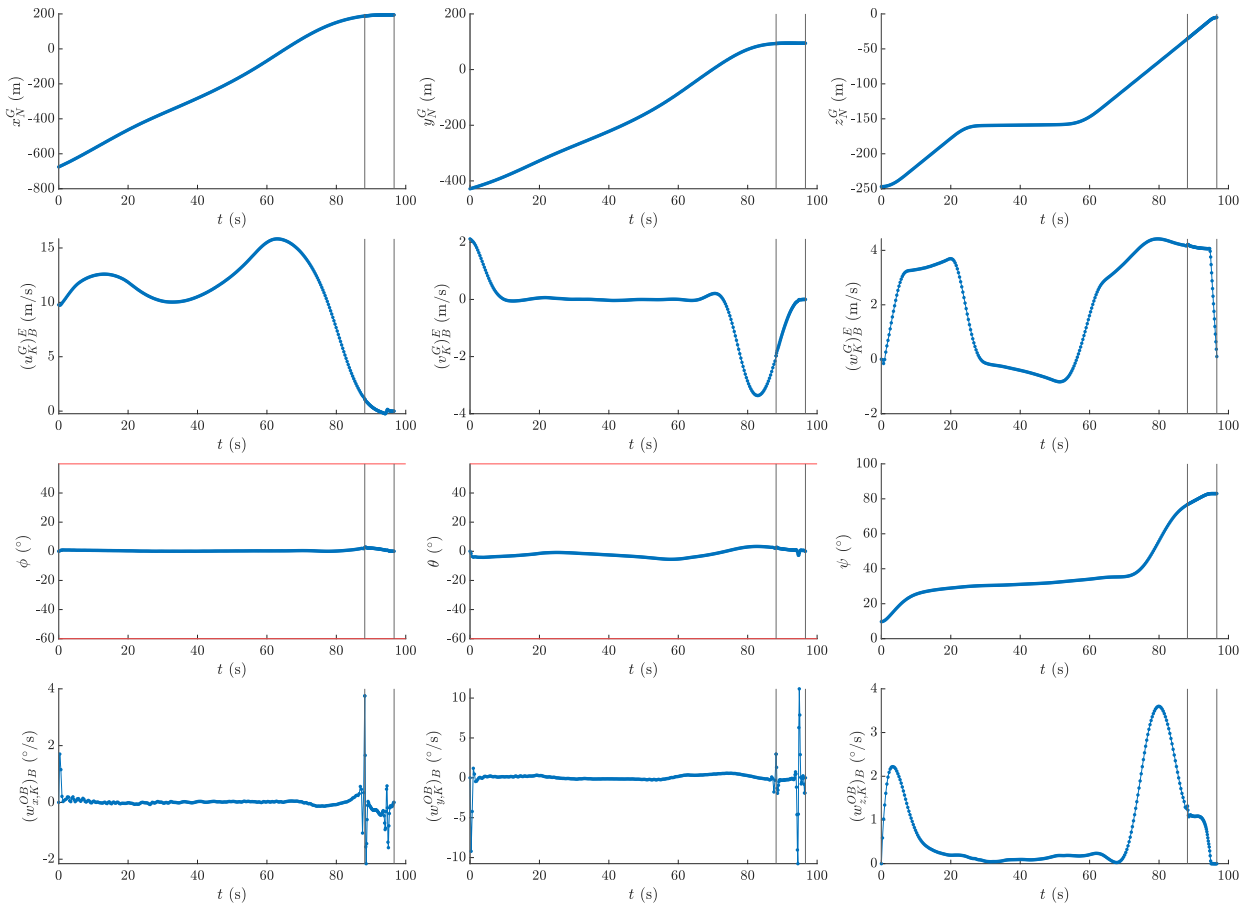

Figure 5 – Generated Trajectory for Automatic Landing – Aircraft States

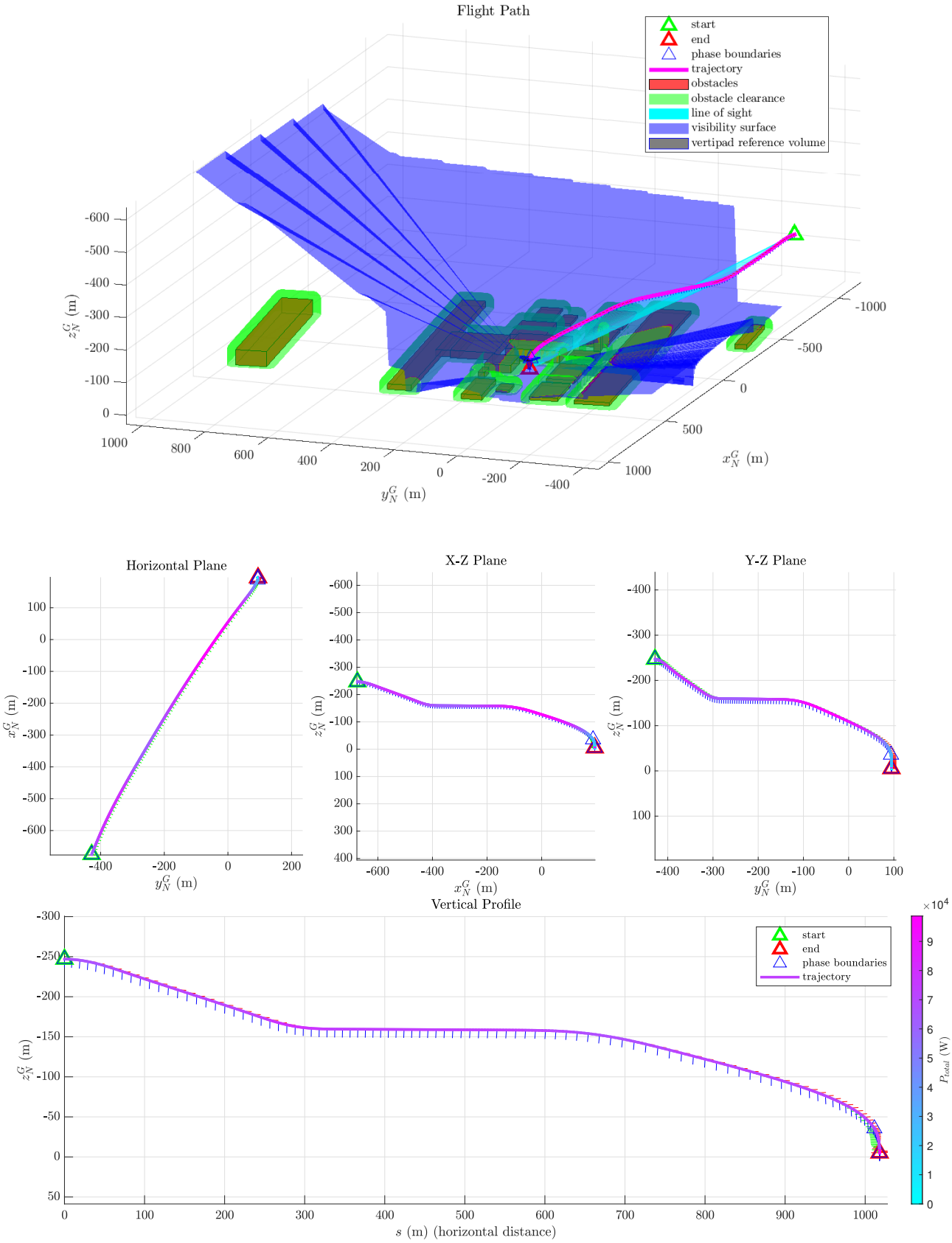

Figure 6 – Generated Trajectory for Automatic Landing – Flight Path

13. Copyright Statement

References

[1] SESAR Joint Undertaking, "U-space Blueprint," 2017. DOI: 10.2829/335092.

[2] Federal Aviation Administration, *Concept of Operations: Urban Air Mobility (UAM)*, 1st ed. FAA, 26 June 2020. [Online]. Available: https://nari.arc.nasa.gov/sites/default/files/attachments/UAM_ConOps_v1.0.pdf.

[3] Rieck M., Bittner M., Grüter B., Diepolder J., Piprek P., Göttlicher C., Schwaiger F., Hosseini B., Schweighofer F., Akman T., and Holzapfel F. *Falcon.m user guide, Version 1.27*, 2022. [Online]. Available: http://www.falcon-m.com.

[4] European Aviation Safety Agency, *Special Condition for small-category VTOL aircraft: SC-VTOL-01*, 1st ed. EASA, 2 July 2019.

[5] European Aviation Safety Agency, *Second Publication of Proposed Means of Compliance with the Special Condition VTOL: MOC-2 SC-VTOL*, 1st ed. EASA, 23 June 2021.

[6] European Aviation Safety Agency, *Prototype Technical Specifications for the Design of VFR Vertiports for Operation with Manned VTOL-Capable Aircraft Certified in the Enhanced Category, PTS-VPT-DSN*. EASA, March 2022.

[7] European Organization for Civil Aviation Equipment, *ED-278A: Concept of Operations for VTOL Aircraft Volume 1 - General Considerations*. EUROCAE, February 2022.

[8] Schmiechen K., Zafar S. A., Dmitriev K., Krammer C., Maly M., and Holzapfel F., "A Requirements Management Template in Polarion for Model-Based Development of Airborne Systems," in *Proceedings of the Software Engineering 2021 Satellite Events*, 2021.

[9] Schmiechen K., Hochstrasser M., Rhein J., Schropp C., and Holzapfel F., "Traceable and Model-Based Requirements Derivation, Simulation, and Validation Using MATLAB Simulink and Polarion Requirements," in *AIAA SciTech 2019 Forum*, Reston, Virginia: American Institute of Aeronautics and Astronautics, 2019, ISBN: 978-1-62410-578-4. DOI: 10.2514/6.2019-0500.

[10] Dmitriev K., Zafar S. A., Schmiechen K., Yi L., Saleab M., Nagarajan P., Dollinger D., Hochstrasser M., Holzapfel F., and Myschik S., "Lean and Highly-automated Model-Based Software Development Process," in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, IEEE, 2020. DOI: 10.1109/DASC50938.2020.9256576.

[11] Aeronautical Radio, Incorporated, *ARINC 424: NAVIGATION SYSTEM DATABASE*. 23 July 2018.

[12] Betts J. T., *Practical Methods for Optimal Control Using Nonlinear Programming*, 3rd ed. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2020, ISBN: 978-1-61197-619-9. DOI: 10.1137/1.9781611976199.

[13] Quilez I. *Distance functions*, Accessed 26 May 2022. [Online]. Available: https://iquilezles.org/articles/distfunctions/.

[14] Kaui B. and Zalik B., "Comparison of viewshed algorithms on regular spaced points," in *Proceedings of the 18th Spring Conference on Computer Graphics*, ser. SCCG '02, Budmerice, Slovakia: Association for Computing Machinery, 2002, pp. 177–183, ISBN: 1581136080. DOI: 10.1145/584458.584487.

[15] Wächter A. and Biegler L. T., "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006, ISSN: 0025-5610. DOI: 10.1007/s10107-004-0559-y.

[16] Science and Technology Facilities Council. *HSL: A collection of Fortran codes for large scale scientific computation*. [Online]. Available: http://www.hsl.rl.ac.uk/.

[17] Hong H., Piprek P., Afonso R. J. M., and Holzapfel F., "Trigonometric Series-based Smooth Flight Trajectory Generation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 721–728, 2021. DOI: 10.1109/TAES.2020.3008576.

[18] Hong H., Piprek P., Afonso R. J. M., and Holzapfel F., "Hierarchical Parameterization for Seamless Transition Trajectory Generation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 5, pp. 2878–2888, 2021. DOI: 10.1109/TAES.2021.3069285.

[19] Landi A. and Nicholson M., "ARP4754A / ED-79A, Guidelines for Development of Civil Aircraft and Systems - Enhancements, Novelties and Key Topics," *SAE International Journal of Aerospace*, vol. 4, no. 2, pp. 871–879, 2011, ISSN: 1946-3901. DOI: 10.4271/2011-01-2564.

[20] Radio Technical Commission for Aeronautics, *DO-178C: Software Considerations in Airborne Systems and Equipment Certification*. RTCA, 5 January 2012.

[21] Radio Technical Commission for Aeronautics, *DO-331: Model-Based Development and Verification Supplement to DO-178C and DO-278A*. RTCA, 13 December 2011.

[22] Gierszewski D. M., Fafard V., Schweighofer F., Piprek P., Speckmaier M., and Holzapfel F., "Toward onboard trajectory optimization for fuel-saving climb of aircraft with automatic flight control," in *6th CEAS Conference on Guidance, Navigation and Control (EuroGNC)*, Berlin, Germany, 2022.

[23] Schatz S. "Development and flight-testing of a trajectory controller employing full nonlinear kinematics," Dissertation, Technische Universität München, München, 2018.

[24] Mumm N. C., Schneider V., and Holzapfel F., "Nonlinear continuous and differentiable 3d trajectory command generation," in *2015 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*, 2015, pp. 1–9. DOI: 10.1109/ICARES.2015.7429822.

[25] Howe, R. M., "A new family of real-time predictor-corrector integration algorithms," *SIMULATION*, vol. 57, no. 3, pp. 177–186, 1991. DOI: 10.1177/003754979105700308.

[26] Mishra C., Schwaiger F., Bähr N. M., Sax F., Kleser M. A., Nagarajan P., and Holzapfel F., "Efficient Verification and Validation of Performance-Based Safety Requirements using Subset Simulation," in *AIAA SciTech 2021 Forum*, Reston, Virginia: American Institute of Aeronautics and Astronautics, 2021, ISBN: 978-1-62410-609-5. DOI: 10.2514/6.2021-0072.

[27] Mishra C., Schwaiger F., Blum C., Kleser M. A., Krammer C., and Holzapfel F., "Efficient Estimation of Probability of Exceeding Performance Limits for Automatic Landing Systems using Subset Simulation," in *AIAA SciTech 2022 Forum*, Reston, Virginia: American Institute of Aeronautics and Astronautics, 2022, ISBN: 978-1-62410-631-6. DOI: 10.2514/6.2022-1894.

[28] Krammer C., Scherer S., Mishra C., and Holzapfel F., "Concept for a Vision-Augmented Automatic Landing System for VTOL Aircraft," in *AIAA AVIATION 2021 Forum*, Reston, Virginia: American Institute of Aeronautics and Astronautics, 2021, ISBN: 978-1-62410-610-1. DOI: 10.2514/6.2021-3217.

[29] Piprek P. and Holzapfel F., "Robust Trajectory Optimization of VTOL Transition Maneuver Using Bi-Level Optimal Control," in *ICAS 31st Congress of the International Council of the Aeronautical Science*, 2018. [Online]. Available: https://www.icas.org/ICAS_ARCHIVE/ICAS2018/data/preview/ICAS2018_0224.htm.