



# Exploration of optimal control based surrogate modeling as a basis for fuel efficient 4D aircraft routing on graphs

Felix Schweighofer<sup>1</sup> · Benedikt Grüter<sup>1</sup> · Florian Holzapfel<sup>1</sup>

Received: 2 February 2023 / Revised: 20 November 2023 / Accepted: 19 December 2023 / Published online: 4 February 2024  
© The Author(s) 2024

## Abstract

Fuel efficient coordination of aircraft operations in the Terminal Maneuvering Area of large airports can contribute to the reduction of the environmental impact of air traffic. To exploit the full potential of the air traffic system, coordinated routing in the Terminal Maneuvering Area, runway assignment and scheduling need to be optimized considering detailed models of the aircraft dynamics and performance. Due to the inhomogeneous nature and level of detail of these combined discrete and continuous decision problems, the optimization of the overall operations poses significant challenges. As part of an integrated approach to the solution of this hybrid problem, this work explores the generation of surrogate models for the fuel consumption of individual aircraft using optimal control methods to exploit the physical capability of the aircraft within an operationally permissible envelope. The surrogate models approximate the predicted fuel consumption of a given point-mass aircraft model on short generic trajectory segments. The trade-off between flight duration and fuel consumption on such segments is analyzed, focusing on the influences of initial aircraft mass, altitude, distance, mean climb angle, along-track wind velocity and linear wind shear. An extensive description of the optimal control-based data generation and surrogate modeling methodology is followed by a discussion of the effects of parameter variation. Based on an illustrative case study, the applicability of the approach is critically analyzed.

**Keywords** Optimal control · Surrogate model · Air Traffic Management

## 1 Introduction

This section explains the context (Sect. 1.1), focus and contribution (Sect. 1.2) as well as the structure (Sect. 1.3) of the present work.

### 1.1 Context: 4D aircraft routing/scheduling

In light of global warming, it becomes increasingly important to reduce the emissions of air traffic, particularly in

large-scale commercial aviation. This need for higher efficiency is reflected by high-level political agendas in some parts of the world, such as the *Flight Path 2050* [10] program by the European Commission, as well as more technical strategies for the evolution of the air traffic system, such as the *European ATM Master Plan* [38] within the SESAR Joint Undertaking.

Efforts to reduce air traffic emissions may target the individual aircraft level, for example by enhancing engines and aerodynamic efficiency or by transitioning from fossil fuels to synthetic fuels, hydrogen or even battery-electric power trains—all of these based on renewable energy. While such measures can significantly increase efficiency on the physical aircraft level, operational aspects offer additional potential for improvement, oftentimes at a lower cost of implementation. This applies to individual aircraft operations, fleet-level coordination by operators and system-level coordination by Air Traffic Control (ATC). Modern navigation, guidance and information exchange systems successively enable the safe introduction of additional degrees of freedom, as in Continuous Climb Operations (CCO)/Continuous Descent

---

✉ Felix Schweighofer  
felix.schweighofer@tum.de

Benedikt Grüter  
benedikt.grueter@tum.de

Florian Holzapfel  
florian.holzapfel@tum.de

<sup>1</sup> Institute of Flight System Dynamics, Department of Aerospace and Geodesy, TUM School of Engineering and Design, Technical University of Munich, Boltzmannstraße 15, 85748 Garching, Germany

Operations (CDO), Trajectory-Based Operations (TBO) and Performance-Based Navigation (PBN), see [38]. Such procedures enable operators and ATC to consider ecological factors in addition to the primary goal of safe traffic flow. For example, CCO/CDO and TBO grant operators more control over the trajectory, while individually arranged PBN corridors enable ATC to safely handle the additional flexibility.

Optimization methods are key to exploiting these features of modern air traffic systems with the goal to reduce environmental impact and cost while increasing airspace capacity. For example, open climb/descent instructions allow operators to choose the most efficient vertical profile according to their aircraft and missions [5]. On the fleet level, accurate scheduling of multiple flights helps save fuel while matching time-varying demands and logistic constraints, see [32]. At the same time, Air Traffic Management (ATM) strives to coordinate flights on the overall system level in a safe, fair and efficient way while continually adapting to uncertainties arising from unforeseen events, weather and emergencies [10, 38]. While fuel savings are of both environmental and economic interest, recent ATM research also aims at reducing indirect climate effects, for example by optimizing contrail formation, as in [41].

ATM optimization has been subject to research for a long time. Operational systems include Arrival Manager (AMAN) decision support systems for human controllers, see [23] for example. Studies have also targeted the integration of ATM and onboard Flight Management Systems (FMS), as in [28]. Still, the remaining potential for efficiency gains continues to inspire the development of new methods.

Many authors focus on either wide-area routing on graphs (in 2D/3D space or in 3D/4D spacetime) or arrival scheduling and runway allocation problems, which are often formulated and solved to global optimality using Mixed-Integer Linear Programming (MILP) based methods. Examples include 3D spacetime routing under uncertainty, see [13], 4D routing in [4, 6, 7], as well as scheduling and runway allocation, see [1] (multi-runway scheduling), [14, 24] (with uncertainties), or [9, 33] (with arrival route selection). A weakness of such methods is that they cannot naturally accommodate detailed flight dynamics and fuel consumption models. The question of how to quantify the fuel consumption on a discretized route network is thus often neglected. Furthermore, scheduling problems are often formulated without explicitly modeling the routing aspect with its associated spatial separation constraints. Notably, the recent works [36, 37] implement free routing with separation using discrete optimization methods and quite sophisticated physical aircraft model approximations, without relying on a graph discretization of the route network.

The desire to incorporate more accurate models of the aircraft behavior has also led to optimal control-based methods, with increasingly detailed approaches being developed

thanks to the unprecedented availability of computational resources. Research activities include optimal control methods based on point-mass aircraft dynamics and Nonlinear Program (NLP) transcription, such as those presented in [2, 3, 11, 12, 15] as well as dynamic programming methods [20–22, 30]. Explicitly modeling the physical aircraft behavior, such methods are particularly well-suited to the planning of individual trajectories—the discrete and combinatorial aspects arising from multi-aircraft coordination, however, have been found to introduce significant difficulties. For example, integrating the arrival sequencing and trajectory optimization as in [17, 19] can yield global optima, but as the number of aircraft increases, computational performance suffers strongly from a disproportionate growth of the number of hypothetically possible discrete sequences and pairwise separation constraints, an issue known as the curse of dimensionality.

Hybrid methods like the bi-level approach in [18] allow for the application of continuous optimal control methods to individual trajectories while solving discrete aspects of the coordination problem by means like genetic algorithms, see [16]. An alternative approach is studied in [43–46]: Here, a relation between feasible arrival times and the associated cost index is calculated by solving the Optimal Control Problem (OCPs); this relation is then used as a basis for MILP arrival sequencing problems considering time separation constraints at given merge points. Similarly, [39, 40] calculate candidate CDO trajectories for multiple aircraft and solve Mixed-Integer Programming (MIP) models for optimal coordination. The question of whether to start with individual OCPs and subsequently solve the discrete coordination problem, or to start with the coordination problem and subsequently find the corresponding trajectories, is addressed by [35].

In the related application of truck routing on highway networks, Watling et al. [47] solve OCPs to minimize fuel consumption on each highway link, varying entry and exit speeds as well as travel time. The tabulated results are used as weights for the shortest path problem in the space-time expanded network. The present, independently conducted study follows a similar overall concept, but introduces surrogate models as an abstraction and focuses on the application aspects of generating the required aircraft trajectory solutions.

## 1.2 Focus and contribution: optimization of TMA traffic

This work addresses the gap between system-level optimization methods unable to handle detailed aircraft models, and aircraft-level optimal control methods incapable of solving the coordination aspects in a scalable way. Using

optimal control, we explore the generation of surrogate models of the flight performance as an interface.

Continuing the line of work by Grüter et al. [16] with a new problem decomposition, this study is part of an exploratory effort to develop and evaluate a holistic approach for traffic optimization in the Terminal Maneuvering Area (TMA) with the goal of reducing overall fuel consumption and delays. In our joint research project *HOTRUN (Holistic Optimization of Trajectories and Runway Scheduling)*, Hoch et al. [25, 26] approach the multi-aircraft routing and scheduling problem on the system level using a graph-based MILP method. Complementary to this high-level optimization, our work is concerned with providing estimates of the best-case fuel consumption for individual aircraft on short trajectory segments, which serve as a basis for modeling objectives and physical feasibility in the routing and scheduling problem. The problem decomposition is described in more detail in Sect. 2.1.

To address the lack of detailed physical aircraft models in the literature on graph-based routing and scheduling, we employ optimal control methods based on a realistic flight dynamics model. We introduce application-specific warmstart and parallelization procedures that enable the efficient generation of a large number of sample solutions. Based on these, we discuss the effects of individual parameter variations and generate surrogate models that can be evaluated efficiently for a large number of trajectory segments in the discrete routing problem. We discuss the challenges associated with this step and critically analyze the results in a 4D graph routing context.

Watling et al. [47] apply a similar problem decomposition to a route planning problem for trucks on a highway network. However, they solve OCPs for specific highway segments and discrete boundary conditions; we instead generate surrogate models with the goal to support a continuous parameter space. Furthermore, we address in detail the generation of the required OCPs solutions for aircraft trajectories with a higher number of parameters and point out the associated difficulties.

### 1.3 Structure

Section 2 formulates the OCPs for trajectory segments, describes the data generation procedures, and discusses the derivation of surrogate models. Section 3 discusses the influence of individual parameters on the relation between flight time and fuel consumption, and analyzes the model performance in the 4D graph routing context. Section 4 summarizes the outcome of the study and gives an outlook on future research potential.

## 2 Methodology

This section describes the assumptions, problem statements and methods forming the basis for the subsequent analysis. Section 2.1 describes the overall framework and identifies the subproblems to be addressed. Section 2.2 states the general assumptions underlying the study. Section 2.3 formulates the OCPs which generate the data for the surrogate models derived in Sect. 2.4. Section 2.5 defines scenarios for verification of the obtained models.

### 2.1 Overall framework and interfaces

The air traffic coordination problem comprises continuous dynamics on the aircraft/trajectory level as well as discrete aspects like arrival sequences on the system level. The holistic optimization of the overall system is therefore considered difficult.

A spacetime graph discretization approach with adaptive refinement developed by Hoch et al. [25, 26] in the *HOTRUN* project handles the high-level routing and scheduling problem using a MILP optimization. The ultimate goal is to find discretized 4D trajectories for a set of aircraft in the TMA such that an optimal compromise between the overall fuel consumption on the system level and a fair distribution of schedule deviations at the Final Approach Fix (FAF) is reached. Constraints include safe separation between multiple aircraft, modeled based on the distances between discretized trajectory segments.

The system-level optimization depends on a model of the fuel consumption of individual aircraft that provides fuel burn estimates for edges in a spacetime graph. This fuel consumption model may, in principle, represent any aircraft or pilot behavior. For the present study, we assume that every individual aircraft is operated optimally with respect to its fuel consumption while following the prescribed 4D route. Based on this assumption we consider a subproblem concerned with quantifying the minimum fuel consumption of individual aircraft on 4D trajectory segments given by the graph discretization of the airspace and planning time horizon.

One approach to the solution of this subproblem is to apply optimal control methods to flight dynamics models which provide an estimate of the instantaneous fuel flow as a function of aircraft states and controls. To that end, we formulate continuous-time OCPs to minimize the fuel consumption on trajectory segments parameterized by their geometry, flight duration, initial aircraft mass and wind conditions, subject to physical and operational limitations. While reference [47] considers explicit velocity boundary conditions in a classical shortest-path problem, the

MILP routing and scheduling method by [25, 26] does not account for velocity, climb angle and track angle at graph nodes to reduce the size of the coordination problem. We, therefore, introduce appropriate assumptions in Sect. 2.2 to create the corresponding surrogate models.

The MILP approach in [25] results in a very large number of edges in the routing/scheduling graph. Because of this, solving OCPs for every individual edge, as done in [43–46] for complete approach trajectories, or for highway links in [47], would result in a prohibitive computational effort and impractical runtimes. Thus our main focus is to explore the construction of surrogate models approximating the minimum fuel consumption for a relevant range of trajectory segment parameters based on optimal control. This approach relates to the Universal Value Function Approximation (UVFA) concept in the field of Approximate Dynamic Programming (ADP) and Reinforcement Learning (RL), though for a true UVFA we would need to explicitly consider all aircraft states at the segment boundaries.

Our explorative study can thus be subdivided in the following phases:

*Data generation* Section 2.3 deals with the subproblem to generate fuel-minimal trajectories in significant numbers using optimal control methods.

*Model derivation* Section 2.4 addresses the derivation of efficient surrogate models from the given optimal trajectories.

*Verification* Section 2.5 discusses verification scenarios.

## 2.2 Application-specific assumptions

The following assumptions are defined for all subsequent analyses:

- A-1 All trajectory segments are short and straight (lateral deviations are allowed, though).
- A-2 Any effects of Earth curvature and rotation are insignificant and can be safely neglected because of A-1.
- A-3 Only low-bandwidth maneuvers typically encountered in commercial transport aircraft are of interest.
- A-4 The aircraft dynamics can be represented by an augmented point mass model due to A-3. A bank angle rate limit is sufficient to represent the rotational dynamics. An aerodynamic sideslip angle of zero is assumed.
- A-5 The effect of mass change during flight along any *single* trajectory segment is negligible in accordance with A-1. The initial mass, however, may vary, and it may be ben-

eficial to explicitly model mass change for computational reasons.

- A-6 The wind field on every trajectory segment can be modeled by a horizontal reference wind velocity aligned with the segment and a linear wind gradient over altitude, based on A-1. Vertical wind velocity is not considered and the wind profile is the same everywhere on the trajectory segment.
- A-7 The atmospheric conditions correspond to the ICAO Standard Atmosphere (ISA).
- A-8 The aircraft configuration, i.e., high-lift devices and landing gear, is fixed on every individual trajectory segment.
- A-9 In general, the aircraft may use an altitude corridor ranging from the specified start altitude to the specified final altitude, extended by  $h_c := 30.48$  m, i.e., one flight level, in both directions. Beyond this, CCO/CDO is required for every trajectory segment with a specified altitude change exceeding  $\pm h_c$ .
- A-10 Lateral maneuvers deviating from the straight trajectory segments (A-1) are restricted according to Required Navigation Performance (RNP) 0.1, corresponding to a corridor half-width of  $y_c := 185.20$  m. For simplicity, we consider neither the spatial extents of the aircraft nor the composition of the RNP error budget.
- A-11 Every trajectory segment starts and ends in steady-state straight and level flight.
- A-12 The Calibrated Airspeed (CAS) at the start and end of every trajectory segment corresponds to the mean CAS on this segment.

Assumptions A-1 and A-3 are based on the overall TMA coordination problem statement and the chosen decomposition described in Sect. 1.2. Assumptions A-2, A-4 and A-5 follow from these and are generally accepted standard practice for similar problems. Assumptions A-6, A-7 and A-8 are approximations introduced to limit the dimensionality of the considered parameter space; we consider them reasonable based on A-1 and the explorative nature of the present study, but there remains potential for future extension. Assumptions A-9 and A-10 are justified within the TMA, which is our focus area. Assumption A-12 is introduced due to the limitations of the system-level coordination approach developed in parallel by Hoch et al., which does not explicitly model the velocity at the segment boundaries (see also Sect. 2.3.2), as well as to limit the dimensionality of the considered parameter space. Analogously, assumption A-11 reduces the dimensionality of the parameter space.

### 2.3 Optimal trajectory segments

This section describes the approach taken to generate a large number of optimal solutions for trajectory segments as a basis for surrogate modeling in Sect. 2.4. Section 2.3.1 introduces the aircraft model underlying the analysis. Section 2.3.3 formulates two parameterized OCPs. Sections 2.3.4 and 2.3.5 explain how we solve a large number of instances efficiently and effectively. Section 2.3.6 discusses limitations inherent to the methodology and its implementation.

#### 2.3.1 Aircraft dynamics and performance model

We consider an extended point-mass aircraft model formulated in a local Cartesian reference frame according to assumptions A-2 and A-4. Note that beyond the ease of implementation these assumptions allow us to consider generic trajectory segments which are not referenced to any global position except for altitude.

The aerodynamic and propulsion forces as well as aircraft-specific physical and operational limitations are modeled according to BADA 4.2 [31], provided by EUROCONTROL. Discontinuities are smoothly approximated to enable the application of gradient-based optimization methods with analytical derivatives. The state vector of our aircraft model has  $N_x = 10$  dimensions and is defined as

$$\mathbf{x} := [x, y, h, V_K, \chi_K, \gamma_K, \mu_A, \bar{C}_L, \bar{C}_T, m]^\top \tag{1a}$$

with the along-track and cross-track positions  $x$  and  $y$ , geodetic altitude  $h$ , kinematic velocity  $V_K$ , track angle  $\chi_K$ , flight path angle  $\gamma_K$ , bank angle  $\mu_A$ , normalized lift and thrust coefficients  $\bar{C}_L \in [0, 1]$  and  $\bar{C}_T \in [0, 1]$ , as well as the mass  $m$  (omitted for some problem statements). We limit the bank angle to  $|\mu_A| \leq 30$  deg. The normalized lift coefficient maps to the range from zero to the maximum lift coefficient expressed by a BADA constraint, while the normalized thrust coefficient maps to the thrust range allowed by the BADA model. By modeling the lift and thrust coefficients as dynamic states, we can impose rate limits as bounds on their derivatives, which we introduce as controls. The control vector of dimension  $N_u = 4$ ,

$$\mathbf{u} := [\dot{\bar{C}}_L, \dot{\bar{C}}_T, \dot{\mu}_A, \delta_{sb}]^\top, \tag{1b}$$

comprises the lift and thrust coefficient rates and the bank angle rate, which we limit to  $|\dot{\bar{C}}_L| \leq 0.01 \text{ s}^{-1}$ ,  $|\dot{\bar{C}}_T| \leq 0.2 \text{ s}^{-1}$ , and  $|\dot{\mu}_A| \leq 5 \text{ deg s}^{-1}$ , as well as a normalized speedbrake command  $\delta_{sb} \in [0, 1]$ . The limits were designed based on preliminary studies with the goal to suppress numerical chattering and obtain smooth, gradual, realistic maneuvers

suitable for the application of interest. We further define an output vector,

$$\mathbf{y} := [\text{Ma}, V_{\text{CAS}}, \dot{m}_F, n_{z,K}]^\top \tag{1c}$$

comprising the MACH number  $\text{Ma}$ , CAS  $V_{\text{CAS}}$ , fuel mass flow  $\dot{m}_F$  and load factor  $n_{z,K}$ . For passenger comfort we restrict the latter to  $-1.2 \leq n_{z,K} \leq -0.8$  (negative values representing upward accelerations). Further aircraft-specific limits on states and outputs are taken from BADA, such as upper bounds on  $\text{Ma}$  or  $V_{\text{CAS}}$ .

We skip the customary description of the equations of motion, which is not particularly relevant in the following, and instead refer the critical reader to Appendix 1. In essence the aircraft model is represented by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{2a}$$

$$\mathbf{c}(\mathbf{x}, \mathbf{u}) \leq \mathbf{0}. \tag{2b}$$

Herein, the inequality constraint function  $\mathbf{c}$  aggregates all limits imposed on  $\mathbf{x}$ ,  $\mathbf{u}$  and  $\mathbf{y}$ .

Finally, we point out that our methodology currently makes use of BADA models, but does not depend on them. Other aircraft models and data sources can be integrated in the future.

#### 2.3.2 Trajectory segment definition

Based on the overall system architecture described in Sect. 2.1, we define a trajectory segment appearing in the MILP routing/scheduling approach by a tuple

$$\mathcal{S} := (\hat{m}_s, \hat{h}_s, \hat{x}_f, \hat{\gamma}_K, \hat{u}_W, \mathcal{D}_h \hat{u}_W; t_f). \tag{3}$$

This comprises the initial mass  $\hat{m}_s$  and altitude  $\hat{h}_s$ , the slope  $\hat{\gamma}_K$  (defined as the elevation angle between the start and end point), the along-track ground distance  $\hat{x}_f$ , as well as the longitudinal wind velocity  $\hat{u}_W$  and its gradient with respect to altitude  $\mathcal{D}_h \hat{u}_W$ , and, finally, the flight duration  $t_f$ . Depending on the context, we assume all of these parameters to be given, or we vary parameters, in particular  $t_f$ , to obtain sets of solutions for similar segments. For ease of notation, we will write  $\mathcal{S}$  in any case. Figure 1 illustrates the parameterization. The final altitude,

$$\hat{h}_f = \hat{h}_s + \hat{x}_f \tan(\hat{\gamma}_K), \tag{4}$$

is determined by  $\mathcal{S}$ ; the reference wind velocity is applied at the mean altitude. Note that the aircraft is not required to follow a straight line between the start and end point.

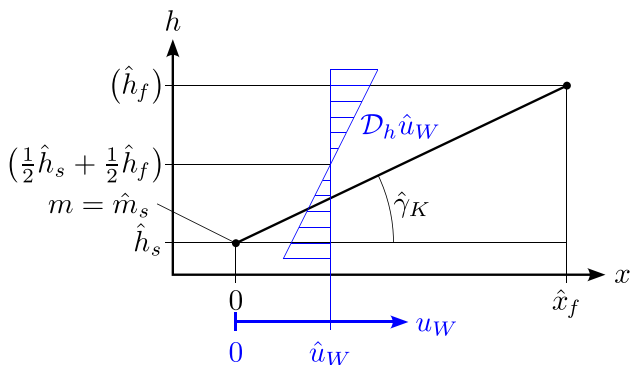


Fig. 1 Trajectory segment parameterization: initial mass, initial height, mean climb angle, distance, and wind shear profile

### 2.3.3 Optimal control problem formulation

We redefine the aircraft states and controls as signals on the continuous finite time horizon  $\mathcal{T} = [0, t_f]$ ,

$$\mathbf{x} : \mathcal{T} \rightarrow \mathbb{R}^{N_x}, t \mapsto \mathbf{x}(t), \tag{5a}$$

$$\mathbf{u} : \mathcal{T} \rightarrow \mathbb{R}^{N_u}, t \mapsto \mathbf{u}(t), \tag{5b}$$

and introduce the parameter vector

$$\mathbf{p} \in \mathbb{R}^{N_p}, \tag{5c}$$

to be defined later for specific problem variants.

Using these, we formulate a nonlinear Optimal Control Problem (OCP) for the flight along a trajectory segment during  $\mathcal{T}$ . The objective of the form

$$\min_{(\mathbf{x}(\cdot), \mathbf{u}(\cdot), \mathbf{p})} (J(\mathbf{x}(\cdot), \mathbf{u}(\cdot), \mathbf{p}) + P(\mathbf{x}(\cdot), \mathbf{u}(\cdot))) \tag{6}$$

includes a primary cost term  $J$  and a penalty term  $P$ . The penalty term serves as a means to suppress undesirable aspects of feasible trajectories. Both terms are defined later.

The trajectory is subject to the differential constraint

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad \forall t \in \mathcal{T} \tag{7a}$$

given by the nonlinear aircraft dynamics  $\mathbf{f}$  and accompanied by physical and operational limitations of the aircraft

$$\mathbf{c}(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{0} \quad \forall t \in \mathcal{T} \tag{7b}$$

as defined by the aircraft model in Sect. 2.3.1; the parameterized position boundary conditions

$$\begin{bmatrix} h(0) \\ x(0) \\ y(0) \end{bmatrix} = \begin{bmatrix} \hat{h}_s \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} h(t_f) \\ x(t_f) \\ y(t_f) \end{bmatrix} = \begin{bmatrix} \hat{h}_f \\ \hat{x}_f \\ 0 \end{bmatrix} \tag{7c}$$

given by the trajectory segment definition  $\mathcal{S}$ , see Sect. 2.3.2 and Fig. 1; the altitude corridor constraint

$$\min(\hat{h}_s, \hat{h}_f) - h_c \leq h(t) \leq \max(\hat{h}_s, \hat{h}_f) + h_c \quad \forall t \in \mathcal{T} \tag{7d}$$

according to assumption A-9; the continuous descent/climb constraint

$$\begin{cases} \gamma_K(t) \leq 0 \quad \forall t \in \mathcal{T}, & \text{if } \hat{h}_s > \hat{h}_f + h_c \\ \gamma_K(t) \geq 0 \quad \forall t \in \mathcal{T}, & \text{if } \hat{h}_s < \hat{h}_f - h_c \end{cases} \tag{7e}$$

inactive for trajectory segments without significant altitude change (assumption A-9); the lateral corridor constraint

$$-y_c \leq y(t) \leq y_c \quad \forall t \in \mathcal{T} \tag{7f}$$

implementing assumption A-10; the trim constraints

$$\begin{bmatrix} \gamma_K(0) \\ \chi_K(0) \\ \dot{V}_K(0) \\ \dot{\gamma}_K(0) \\ \dot{\chi}_K(0) \end{bmatrix} = \begin{bmatrix} \gamma_K(t_f) \\ \chi_K(t_f) \\ \dot{V}_K(t_f) \\ \dot{\gamma}_K(t_f) \\ \dot{\chi}_K(t_f) \end{bmatrix} = \mathbf{0} \tag{7g}$$

imposing steady-state straight and level flight at the initial and final boundary (assumption A-11); the additional velocity constraints

$$V_{CAS}(0) = V_{CAS}(t_f) \tag{7h}$$

$$V_{CAS}(0) = t_f^{-1} \int_{t=0}^{t_f} V_{CAS}(t) dt \tag{7i}$$

according to assumption A-12.

Note that the altitude corridor constraint (7d) comes into effect only in case of near-horizontal flight. Otherwise, it is superseded by the CCO/CDO constraint (7e).

The CAS constraints (7h) and (7i) are designed to achieve a reasonable velocity profile even though the velocity at the segment boundaries is not prescribed. Without such constraints, optimization would typically result in descent segments accelerating from a very low velocity to a very high velocity, and climb segments starting at a very high velocity and ending at a very low velocity. This behavior is undesirable because it exploits aspects that are not modeled explicitly to obtain formally better but unreasonable solutions. Equating the boundary velocities and the mean velocity ensures that the overall speed level is maintained while allowing for temporary deviations. This constraint is applied to CAS to capture the natural altitude dependency of the optimal (in terms of efficiency) airspeed. It results in

velocity profiles where True Airspeed (TAS) increases with increasing altitude.

The objective consists of a primary objective  $J$  and a penalty term  $P$  designed to support the convergence of the direct optimal control solver (Sect. 2.3.4) and suppress undesirable oscillations in the solution. The terms are weighted such that  $P$  is insignificant compared to  $J$ . The latter is expected to be on the order of  $10^1$  to  $10^2$  (see below).

Most prominently, solutions on which a high flight time is imposed tend to exhibit zigzagging behavior with alternating lateral excursions from the mean flight path. These can be explained by the optimization algorithm seeking to maintain an admissible and efficient air speed while satisfying the long imposed flight duration. Flying on a straight path in such cases requires a low speed and, therefore, incurs a high fuel consumption if this low speed is even feasible. Zigzagging elongates the flight path and thereby allows for a higher speed and higher efficiency. Contrary to this, short imposed flight durations inherently require high speed on short paths, so they yield straight trajectories.

We consider the zigzagging undesirable from an operational perspective and, therefore, suppress it. Penalties on the lateral position,  $\int_T y^2 dt$ , and on the bank angle,  $\int_T \mu_A^2 dt$ , effectively reduce this phenomenon even at small weights of  $10^{-5} \text{ m}^{-2} \text{ s}^{-1}$  and  $10^{-5} \text{ deg}^{-2} \text{ s}^{-1}$ .

Additionally, a penalty on the normalized speed brake control,  $\int_T \delta_s b^2 dt$ , weighted by  $10^{-3} \text{ s}^{-1}$ , prevents the optimizer from getting stuck in local optima with increased speed brake usage. Speed brakes are still used where necessary, typically in a steep descent.

A small quadratic penalty on the normalized thrust command rate,  $\int_T \dot{C}_T^2 dt$ , expresses a preference for solutions with gradual changes in thrust. It is weighted by  $10^{-2} \text{ s}^{-1}$  and thus contributes a maximum cost on the order of  $10^{-2}$  to the overall objective, assuming an expected flight time of not more than a few hundred seconds.

Analogously, we penalize the normalized lift coefficient command rate by  $\int_T \dot{C}_L^2 dt$  with a weight of  $10^{-2} \text{ s}^{-1}$ . Combined with the corresponding rate limit (Sect. 2.3.1) this helps ensure gradual maneuvers to satisfy assumption A-3.

From the base problem thus formulated we construct two independent parameterized problem statements which differ in their primary objective functions  $J$ , the bounds imposed on the final time and the treatment of mass change:

**Problem  $\mathcal{T}$**  minimizes the flight time, i.e.,  $J = q_t t_f$  where  $q_t = 1 \text{ s}^{-1}$  (which yields large values compared to the penalties). In this case, the flight time is subject to optimization, i.e., we define the parameter vector from Eq. 5c as  $\mathbf{p} := t_f$  with  $N_p = 1$ . According to assumption A-5 we consider  $m = \hat{m}_s$  a constant.

**Problem  $\mathcal{F}$**  minimizes the total fuel consumption by maximizing the final mass, i.e.,  $J = -q_m m(t_f)$  with  $q_m = 1 \text{ kg}^{-1}$  (again, typical values are significantly larger than the penalty terms). In this case, the flight time is fixed, i.e., effectively  $N_p = 0$ , and we model the dynamic mass change to integrate the fuel flow together with the other states.

Based on the trajectory segment definition in Sect. 2.3.2, a straight-forward approach to data generation would be to create a grid of segment parameters according to eq. 3 and solve Problem  $\mathcal{F}$  independently at every grid point, possibly in parallel. Though simple, this approach lacks both efficiency and effectiveness. The former is because common solution methods based on NLP transcriptions rely on an initial guess and can perform significantly better when solving multiple closely related problem instances in a row than they do with independent or strongly varying instances. The latter is due to the initial guess not only affecting computational performance but also the outcome of the optimization; solving a sequence of closely related instances tends to increase the success rate compared to independent runs.

Because of this, we seek to exploit similarities between trajectory segments. Due to the specific characteristics of the problems at hand we choose to subdivide the data generation task as follows:

- A sweep over  $t_f$  reveals the relation between flight duration and fuel consumption on a trajectory segment with otherwise constant parameters, see Sect. 2.3.4.
- A directed, hierarchical exploration of the remaining parameter space, see Sect. 2.3.5, covers the flight envelope, running a  $t_f$  sweep at every sample.

### 2.3.4 Fuel versus time: a nonlinear trade-off

We model the OCPs formulated in Sect. 2.3.3 using an extended development version of the optimal control code FALCON.M<sup>1</sup> [34], which provides object-oriented facilities for the formulation of continuous-time problems in MATLAB and their transcription into sparse NLPs. The resulting NLPs are then solved iteratively by the large-scale local NLP solver IPOPT [48], a filter line search interior-point optimization algorithm, using the linear solver routine `ma57` from the HSL [27] package for sparse linear algebra subproblems.

A trapezoidal collocation scheme with a uniform time discretization of 50 samples is applied in our studies. This discretization is very coarse; it corresponds to steps

<sup>1</sup> <https://falcon-m.com/>.

of approximately 1 s to 7 s depending on  $t_f$ . For exemplary cases, we confirmed that a higher resolution does not significantly affect the results relevant for our project, i.e., the relation of fuel consumption and flight time for a given trajectory segment definition  $S$ .

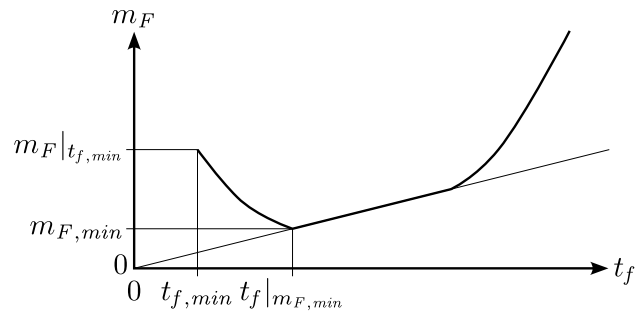
Our preliminary numerical studies indicate that Problem  $\mathcal{F}$ , in variants with a fixed or free flight time, is inherently more difficult than Problem  $\mathcal{T}$ , judging from relatively high sensitivity to parameter values, numerical scaling and initial guess. In the case of large imposed flight times, we observe that the optimization converges to elongated flight paths with zigzagging lateral deviations allowing to maintain an efficient airspeed, as described in Sect. 2.3.3. This leads us to suspect that an associated worsening of convergence behavior may be due to ambiguity of the solution, for example concerning the placement and side of the lateral excursions, as well as the initial guess being less accurate since it may not include the same excursions. Accordingly, the convergence of maximum-time problems also proves difficult. Nevertheless, we can generate a large number of fuel-minimal solutions with varying flight times in an acceptable time frame by implementing the application-specific procedure detailed in the following.

Given a generic trajectory segment  $S$ , we first solve Problem  $\mathcal{T}$ . A primitive initial guess for the dynamic states is usually sufficient, though we may use existing solutions from related instances. A Broyden-Fletcher-Goldfarb-Shanno (BFGS) approximation to the second derivative of the NLP works well here. We obtain the minimum flight duration  $t_{f,\min}^*$  and the associated fuel consumption  $m_F|_{t_{f,\min}^*}$ . If this step fails for a specific instance we cannot draw any straight-forward conclusions on the feasibility of the flight along the corresponding trajectory segment, due to the inherent fragility of the numerical solution methods.

Next, we solve Problem  $\mathcal{F}$ , setting

$$t_f := t_{f,\min} := t_{f,\min}^* + t_{f,\text{offset}} \tag{8}$$

and initializing with the solution of Problem  $\mathcal{T}$ . From here on, we consider  $t_{f,\min} = t_{f,\min}^* + t_{f,\text{offset}}$  the minimum flight duration; the offset  $t_{f,\text{offset}} := 1$  s is introduced to avoid potential infeasibility caused by small deviations of the aircraft model, which includes a dynamic mass state not modeled in Problem  $\mathcal{T}$ , or by differences in the numerical setup; it is considered negligible in our application scenario. Without these small differences in the problem formulation and implementation, and without the time offset, a valid solution for Problem  $\mathcal{T}$  would mean that there must be a valid solution for Problem  $\mathcal{F}$  as well. Though our approach does not allow us to draw such strict conclusions, we can reasonably expect that there is a solution to Problem  $\mathcal{F}$  at  $t_f$  in all but



**Fig. 2** Schematic relation of flight duration and fuel consumption on a descent segment: short durations (high speeds) yield high fuel consumptions, moderate durations allow for idle descent, long durations (low speeds) yield high fuel consumptions due to increased induced drag

exceptionally sensitive cases (which are not of particular interest), and should obtain this solution unless numerical issues prevent convergence. We obtain the minimum fuel consumption  $m_{F,\min}|_{t_{f,\min}}$  that allows the aircraft to traverse the trajectory segment in minimum time.

Starting from the point  $(t_{f,\min}, m_{F,\min}|_{t_{f,\min}})$  on the PARETO frontier of time and fuel consumption, we solve a sequence of Problem  $\mathcal{F}$  instances with successively increasing  $t_f$  in steps of  $t_{f,\text{step}}$ . In every step, the numerical optimization is initialized at the previous solution and thereby achieves convergence up to the specified tolerances quickly in most cases. Notably, there is a trade-off between the step size in  $t_f$ , accuracy of the  $m_F$  values and computational performance. Typically, small steps can be expected to increase the convergence speed of individual problems and represent the resulting  $(t_f, m_F)$  curve more accurately, whereas large steps reduce the number of problems solved, thus potentially reducing overall computation times as well as the resolution of the results.

This insight suggests that step-size adaptation strategies may yield better trade-offs between performance and accuracy than fixed-step sizes, as observed in many other well-known numerical methods. Indeed, we find that very good results can be obtained by increasing or reducing the  $t_f$  step size by a constant factor (within prescribed bounds) based on a linear prediction of the next  $m_F$  value and a given prediction error bound. However, this approach requires careful tuning of the adaptation strategy and can result in unsatisfactory convergence characteristics and long computation times in some cases. Here, the key issue is the unpredictable runtime of NLP solvers, which does not reliably correlate with the parameter variation step size. Consequently, we use fixed steps of  $t_{f,\text{step}} := 2$  s for all analyses presented in this paper.

Furthermore, we find that evaluating the analytical second derivative of the NLP instead of its BFGS approximation



improves convergence during this flight time sweep in Problem  $\mathcal{F}$ . Note that this is contrary to the finding for Problem  $\mathcal{T}$ , which tends to be solved much more reliably and efficiently with a BFGS approximation. This observation can be explained by the relatively small step size in the  $t_f$  sweep, which typically results in a sequence of very similar solutions, meaning that we may profit from the locally superlinear convergence characteristics of the NLP solver. When varying other parameters, as described in Sect. 2.3.5, the variation steps are usually larger and the focus is on achieving robust convergence of Problem  $\mathcal{T}$  before starting the time sweep.

Figure 2 illustrates a typical shape of the resulting  $(t_f, m_F)$  relation for a descent trajectory segment. Short flight times, i.e., high speeds, cause high fuel consumption due to parasitic drag. At moderate flight times the fuel consumption drops, but at some point slowing down further starts to require significantly more fuel due to an increase in induced drag. The fuel consumption curves sometimes exhibit a linear region in idle descent, as depicted, otherwise the behavior is consistent with the fuel-vs-time PARETO front generated by [30] for a longer trajectory including climb, cruise and descent phases.

For the operation of individual aircraft the high-speed region between  $t_{f,\min}$  and the minimum fuel time is clearly most attractive, as it provides good trade-offs between the two cost factors time and fuel. For system-level optimization, however, higher flight times with an additional, moderate increase in fuel consumption may be acceptable or even beneficial on some trajectory segments if this improves the overall efficiency. Nevertheless, we deem it justified to terminate the  $t_f$  sweep before reaching excessively high flight durations. In addition to avoiding unattractive flight conditions for individual aircraft, such a cut-off helps drastically reduce computation times. This is because the fuel-minimal trajectories at high  $t_f$  tend to be less clearly distinguished than at high speed; numerical experiments indicate that such solutions often exhibit step descents, where the exact number and distribution of steps is highly sensitive to the numerical configuration. Physically it seems likely that many local optima arise in these cases, hindering convergence. Based on these preliminary observations, we terminate the procedure once either the flight duration exceeds a threshold,

$$t_f > \max(1.5 t_{f,\min}, t_{f,\min} + 5(t_f|_{m_{F,\min}} - t_{f,\min})), \tag{9}$$

or the fuel consumption becomes excessive:

$$m_F > \max(1.5 m_{F,\min}, m_{F,\min} + 1.5(m_F|_{t_{f,\min}} - m_{F,\min})) \tag{10}$$

Additionally, we terminate once a problem instance fails to converge to a valid solution.

### 2.3.5 Expanding the parameter envelope

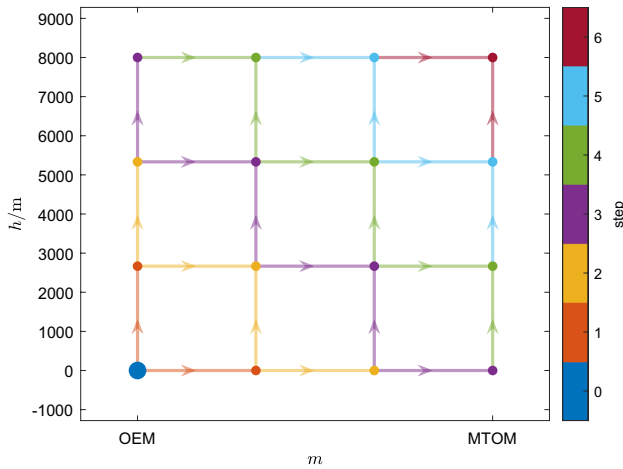
Our analyses as well as the generation of surrogate models require a large number of sample solutions covering the part of the flight envelope contained in the parameter space for trajectory segments. Parallel computing and solver warm starts are essential to obtain these sample solutions within a reasonable timeframe, but data generation is not per se a time-critical task. Since it is fully decoupled from any real-time routing and scheduling, computing resources only limit the number of data points that we can generate in practice and may thereby limit the quality of surrogate models derived from the data. Real-time routing and scheduling though is only affected by the surrogate model evaluation time.

We employ a custom framework built on top of MATLAB'S PARALLEL COMPUTING TOOLBOX to guide the parameter sweeps for data generation. A naïve approach is to iterate over a grid of parameters and rely on the easily accessible `parfor` feature for parallelization. This, however, requires that the order of execution is completely arbitrary—an assumption that is not suitable for taking advantage of warm start functionalities of NLP solvers, which should use a closely related solution as a basis. Therefore, we model a parameter sweep as a graph, where every node or *task* represents a combination of parameters and every edge expresses a dependency. The latter may represent either hard dependencies that strictly need to be available for a task to run, or soft dependencies in the sense that at least a given number of the predecessor nodes needs to be solved. The graph of hard dependencies must be a Directed Acyclic Graph (DAG) for a valid execution sequence to exist. For soft dependencies, this requirement does not hold, but we satisfy it nonetheless.

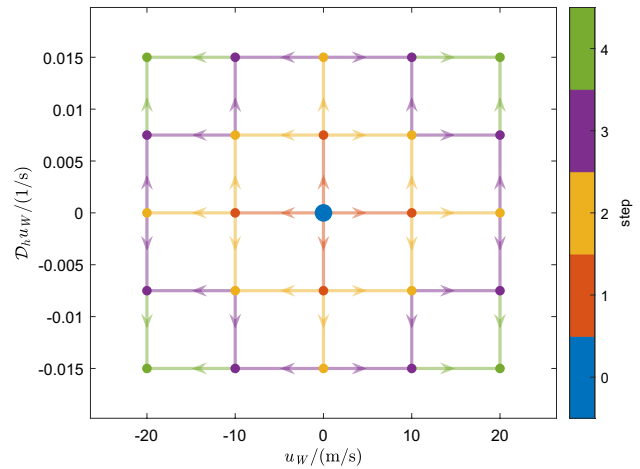
Using dependencies, we can specify the direction of the parameter sweep in a very natural and generic way even in multiple dimensions. For example, this allows us to model a wave front expansion of the parameter envelope, thereby avoiding to spend computing time on parameter combinations that are far beyond the capabilities of the aircraft. At the same time, we use the dependencies to initialize the optimization problems.

Sweeps are orchestrated by a controller object in the client MATLAB session. Tasks whose dependencies are satisfied are scheduled and executed in parallel on a pool of worker instances by means of MATLAB'S `parfeval()` functionality. The framework is set up such that the Optimal Control Problem (OCP)/NLP structures for Problem  $\mathcal{T}$  and Problem  $\mathcal{F}$  are constructed only once on each worker using FALCON.M and reused for every task.

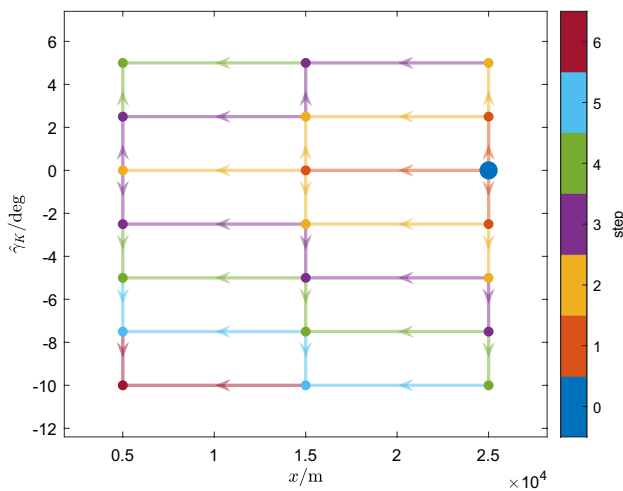
Figures 3, 4, 5 illustrate the individual layers of the hierarchical dependency structure employed in the present study. All relations are formulated as soft dependencies requiring at least one valid result for initialization. The  $(\hat{m}_s, \hat{h}_s)$  graph



**Fig. 3** Mass and altitude sweep structure: envelope expansion from low to high mass/height



**Fig. 5** Wind speed and gradient sweep structure: successive increase in absolute wind speed and shear



**Fig. 4** Distance and climb angle sweep structure: successive increase of mean absolute slope

in Fig. 3 forms the basis; the root node is in the lower left corner. At every node of this graph, we append the  $(\hat{x}_f, \hat{\gamma}_K)$  dependency structure shown in Fig. 4; then we expand every resulting node by the  $(\hat{u}_w, \mathcal{D}_h \hat{u}_w)$  combinations illustrated by Fig. 5. The nodes are color-coded by their distance from the root node of their respective layer, measured in discrete steps, to show the expected direction of expansion. In essence, we gradually expand the envelope starting from low mass and low altitude to high mass and high altitude, from horizontal flight at the maximum segment distance to climb and descent on short segments, and from zero wind to positive and negative wind speed and wind shear. We include edge cases such as segments starting at the Operational Empty Mass (OEM),  $\hat{m}_s = m_{\text{OEM}}$ , and descents to negative altitudes in exceedance of the lowest real-world geographic

**Table 1** Data generation task status and duration

|            | Tasks | Duration <sup>a</sup> (s) |                    |                    |
|------------|-------|---------------------------|--------------------|--------------------|
|            |       | Total                     | Mean               | Median             |
| Total      | 8400  | 152,260                   | 23.52 <sup>b</sup> | 12.37 <sup>b</sup> |
| Successful | 6154  | 145,493                   | 23.64              | 12.56              |
| Failed     | 317   | 6767                      | 21.27              | 2.13               |
| Skipped    | 1929  | 0                         | –                  | –                  |

<sup>a</sup>Wallclock time, sum over all worker instances

<sup>b</sup>Excluding skipped cases

depressions as a basis for interpolation, even though they themselves are not actually feasible. For the OEM cases, we relax any lower bounds on the aircraft mass; for negative altitudes no action is required as we do not model the Earth’s surface.

Note that we apply the parameter sweep procedure presented in this section to all parameters except the flight duration  $t_f$ . The calculation of  $t_{f,\min}$  (Problem  $\mathcal{P}$ ) and sweep over  $t_f$  (Problem  $\mathcal{F}$ ) is performed sequentially within every task to avoid communication and initialization overheads associated with parallelization.

Using the parameter grid shown in Figs. 3, 4, 5 and described above, we generate data for the subsequent surrogate model development. The parameter sweep is run in parallel on a hexacore desktop computer (INTEL CORE i5-8600K with 32 GB of RAM), using four MATLAB worker instances and one client instance. The computation times of the tasks are stated in Table 1. Each of the tasks includes a  $t_f$  sweep. Note that the mean duration of the failed tasks is almost equal to the mean duration of successful tasks since there is often no way for the NLP solver to quickly determine that a problem is likely infeasible. Tasks whose dependencies are

**Table 2** Number of valid trajectories per successful data generation task and trajectory computation times

|                              | Min  | Max   | Mean | Median |
|------------------------------|------|-------|------|--------|
| Solutions                    |      |       |      |        |
| Problem $\mathcal{T}$        | 1    | 1     | 1    | 1      |
| Problem $\mathcal{F}$        | 1    | 112   | 25.9 | 22     |
| Solver time <sup>a</sup> (s) |      |       |      |        |
| Problem $\mathcal{T}$        | 0.17 | 4.85  | 0.41 | 0.39   |
| Problem $\mathcal{F}$        | 0.12 | 59.12 | 0.58 | 0.29   |

<sup>a</sup>Wallclock time per trajectory

not satisfied due to failures are skipped. For the successful tasks, we determine the number of resulting trajectories and the relative performance of Problem  $\mathcal{T}$  and Problem  $\mathcal{F}$ , as stated in Table 2. We observe that the time required to solve Problem  $\mathcal{T}$  is relatively stable, with values ranging from 0.17 to 4.85 s. For Problem  $\mathcal{F}$  we obtain a better median performance, but the spread is ten times larger, leading to worse mean times. In total, we obtain 173725 valid trajectories, among these 167,571 solutions of Problem  $\mathcal{F}$  instances, within 10 h and 41 min. These only serve as a basis for surrogate modeling in Sect. 2.4, so the time expended for the trajectory computations is irrelevant for real-time use of the derived surrogates. However, computational performance limits the size of the parameter space that can be covered and the ability to scale the approach to the large number of different aircraft types to be considered in practical ATM applications.

### 2.3.6 Limitations

A major limitation of the MILP optimization method for graph-based TMA routing and scheduling in the *HOTRUN* project is the impracticality of modeling the aircraft velocity and direction of flight at all nodes. As explained in Sects. 2.1 and 2.2, we resort to heuristic constraints based on the notion of CAS and the mean velocity to obtain reasonable velocity profiles on every individual segment, and we assume steady-state straight and level flight at the segment boundaries.

Furthermore, due to limited computational resources, our nonlinear optimal control approach cannot cover very high-dimensional parameter spaces with sufficient resolution. This means that the optimal control methodology introduces similar limitations to those described above.

The local optimization method employed to solve the OCPs/NLPs, see Sect. 2.3.4, may get stuck in a local optimum which is not the desired solution. In fact, we observe switches in the solution structure during the  $t_f$  sweep. In some cases, no valid solution is obtained.

The dependency-based parameter sweep approach described in Sect. 2.3.5 has three potential drawbacks

compared to independent solver runs. First, since failed tasks are not necessarily physically infeasible, dependent tasks may be inadvertently skipped; we attempt to avoid this issue to some degree using soft dependencies, which allow tasks to run once any neighboring task succeeds. Second, tasks may fail due to the chosen initialization, and we do not retry them with a new initial guess. Third, the hierarchical expansion of the parameter envelope means that physically feasible cases may be skipped because of an assumed dependency on physically infeasible cases. For example, we only consider cases with wind or wind shear after successfully solving the corresponding scenario without wind. Nevertheless, we think that these limitations are acceptable for an explorative study.

## 2.4 Surrogate models

This section deals with the step of aggregating the generated trajectory samples in surrogate models for real-time use. Section 2.4.1 describes the interface and other requirements; Sect. 2.4.2 presents a simple modeling approach.

### 2.4.1 Interface and requirements

To determine edge weights in a large routing and scheduling graph, surrogate models are required to efficiently provide estimates of the fuel consumption as well as the admissible flight envelope. Both need to be expressed as functions of the trajectory segment parameters  $\mathcal{S}$ , which in turn correspond to the attributes of edges in the graph. Modeling the envelope boundary is a very important aspect in our aerospace application, as the complete set of considered aircraft and environment parameters affects the physical capability of the aircraft to traverse a given trajectory segment in a given time. While it is possible to model infeasibility as infinite (or artificially exaggerated) fuel consumption, we consider it more appropriate to formulate a surrogate model with two independent functions, one predicting the fuel consumption and the other representing the flight envelope. The latter can be modeled as an implicit function indicating set membership.

It is desirable for the surrogate models to be not only computationally efficient, but also deterministic, meaning that the evaluation is guaranteed to succeed in a finite time. This makes such models applicable in a real-time context where the optimal control methods used for data generation are too slow and unreliable, given their convergence characteristics.

### 2.4.2 Grid-based approach

The hierarchical envelope expansion approach described in Sect. 2.3.5 is designed to support working with a regular

grid of parameter values. Thus it is a natural approach to collect the results in a multidimensional array, with dimensions corresponding to  $\mathcal{S}$  (see Sect. 2.3.2), and apply linear interpolation at the query points.

However, it is to be considered that the dataset always contains gaps due to individual cases failing for nonphysical reasons. This is a major incentive for the surrogate modeling approach; besides yielding shorter evaluation times, the surrogate model can smooth over the small interior regions of the parameter space where the data generation procedure fails even though valid solutions are known to physically exist. In a 7D grid the interpolated value of every query point is based on the values at  $2^7 = 128$  neighboring vertices, meaning that the chance of at least one of these neighbors being an invalid data point is excessively high. Because of this, linear interpolation is not immediately applicable if the envelope boundaries are not aligned with the grid dimensions.

To alleviate this issue, we fill small gaps and extrapolate slightly beyond the valid region. To this end, we first define a distance-based weighting scheme approximating a truncated multivariate GAUSSIAN kernel with a radius of 2 grid points and a standard deviation of 1 grid step. Centered at every missing point, we use this kernel to calculate a weighted sum of valid grid values, as well as the sum of weights corresponding to valid data points in the vicinity. Both operations can be implemented conveniently and efficiently as discrete convolutions. The gaps are then filled with the weighted mean arising from these values.

Since this approach not only fills gaps in the interior of the feasible domain, but also extrapolates beyond its (unknown) boundary, we construct an implicit function approximation of the boundary based on the original dataset. This is achieved by assigning 1 to every valid grid point and 0 to every invalid grid point, and smoothing this array using a discrete convolution with a GAUSSIAN kernel with a radius of 1 grid step. The boundary is then approximated by the manifold at a value of 0.5.

The grid-based surrogate model enables the evaluation of both approximate fuel consumption and envelope membership of graph edges at a rate of more than  $8 \times 10^6 \text{ s}^{-1}$  on the desktop computer also used for data generation. This is clearly several orders of magnitude faster than solving the underlying OCPs. Furthermore, the surrogate model evaluation is deterministic.

Appendix 2 gives an overview of alternative surrogate modeling methods, which are not implemented in the present study based on preliminary experiments.

## 2.5 Verification

Within the scope of this study, we limit ourselves to the analysis of exemplary cases to gain a basic understanding of the applicability of the developed approach.

As a basis for discrete routing, we generate a spacetime graph. In the present study we consider only the vertical plane, although the surrogate models can just as well be integrated into the 4D routing/scheduling framework by Hoch et al. [25, 26]. We build a graph extending over 150 km in the horizontal dimension and over an altitude range from 0 to 10,000 m; the discretization is 10,000 m or 15,000 m in the horizontal dimension and 200 m in the vertical dimension. Implicitly enforcing a CDO constraint we generate edges with climb angles in the range from  $-10$  to  $0$  deg by connecting the neighboring nodes resulting from the discretization. For the time dimension, we choose a discretization of either 5 s or 10 s, with a sufficiently long time horizon of 1000 s. Along the time dimension, instances of spatially neighboring nodes are connected if they are reachable within the velocity range of the aircraft. Note that this results in a velocity discretization whose resolution increases with decreasing speed. Since the high-speed region of the fuel-vs-time trade-off is more attractive, future work should reconsider the graph construction.

We consider three distinct wind conditions: no wind, constant headwind ( $u_w = -15 \text{ m s}^{-1}$ ), and a linear wind shear ( $\mathcal{D}_h u_w = 0.003 \text{ s}^{-1}$  with  $u_w = -10 \text{ m s}^{-1}$  at  $h = 5000 \text{ m}$ ). In each case, we calculate the wind speed and gradient at the nominal midpoint of every trajectory segment, according to Fig. 1.

We model a descent from 8000 to 3000 m over a ground distance of 150 km and choose the start node as well as a set of final nodes accordingly. As start mass, we choose the mean of  $m_{\text{OEM}}$  and  $m_{\text{MTOM}}$  of the aircraft model used for data generation.

Based on the spacetime discretization and the wind field, we estimate edge weights using the fuel consumption surrogate model. To find fuel-minimal discrete trajectories, we resort to a standard shortest path algorithm. In the ATM application, a MILP method may be used instead.

Finally, we solve a multi-phase Optimal Control Problem (OCP) following the discrete trajectory for comparison. We spare the reader the full problem formulation; essentially, we model an Optimal Control Problem (OCP) as a series of trajectory segments similar to Problem  $\mathcal{F}$ , with one segment per edge of the discrete solution and fixed times at the segment boundaries. However, we omit the boundary conditions (7c), the trim constraints (7g) and the mean CAS constraint (7i). Additionally, we apply the CAS constraint (7h) not for every individual phase/segment, but for the full trajectory, and enforce continuity of all aircraft states at the phase boundaries.

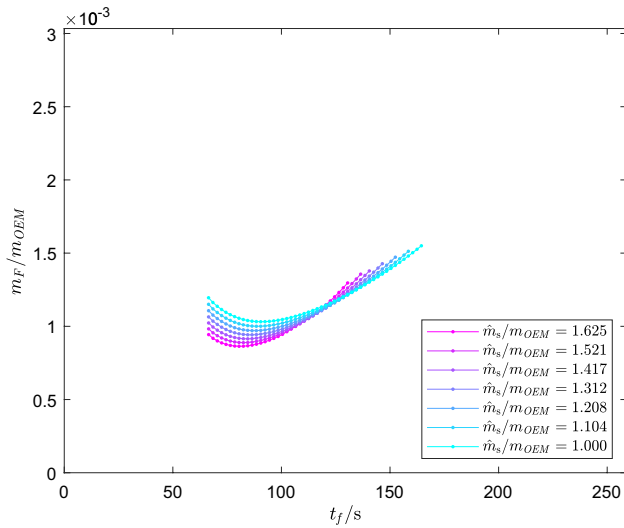


Fig. 6 Fuel/time relation at varying mass: higher mass is advantageous in the fast descent

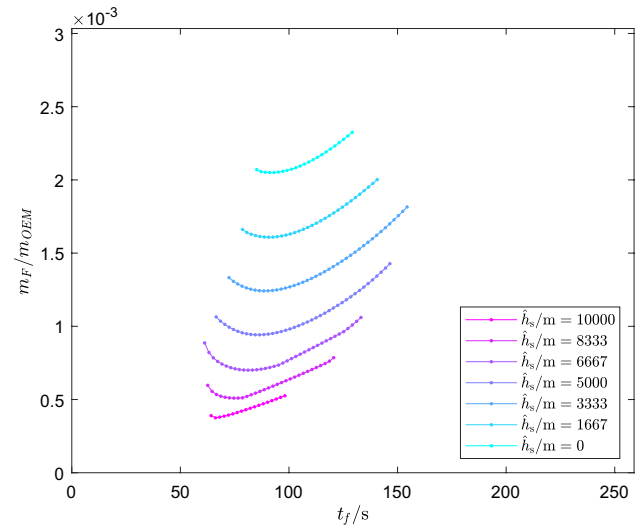


Fig. 7 Fuel/time relation at varying altitude: fuel consumption is lower at higher altitude

A comparison of the discrete and continuous trajectories and fuel consumptions then gives insight into the expected error magnitude.

Further verification aspects to consider in the future include the numerical accuracy and grid independence of the optimal control-based data generation, the envelope coverage of the surrogate models and the accuracy of discrete solutions with respect to free-flight OCP solutions.

### 3 Results and discussion

Section 3.1 discusses the optimal trajectory segments obtained from the OCPs. Section 3.2 presents a case study to analyze the applicability of the surrogate models in graph-based trajectory planning.

#### 3.1 Optimal trajectory segments

This section describes the optimal trajectory segments obtained by the procedures discussed above. We consider an exemplary short-range turbofan aircraft according to BADA. In Sect. 3.1.1 we describe the effects of individual parameter variations on the resulting fuel/time relations. Section 3.1.2 gives an impression of the results for simultaneous parameter variations.

##### 3.1.1 Effects of isolated parameter variations

To foster an understanding of the shape of the parameter effects to be captured by surrogate models, we define an exemplary descent trajectory:

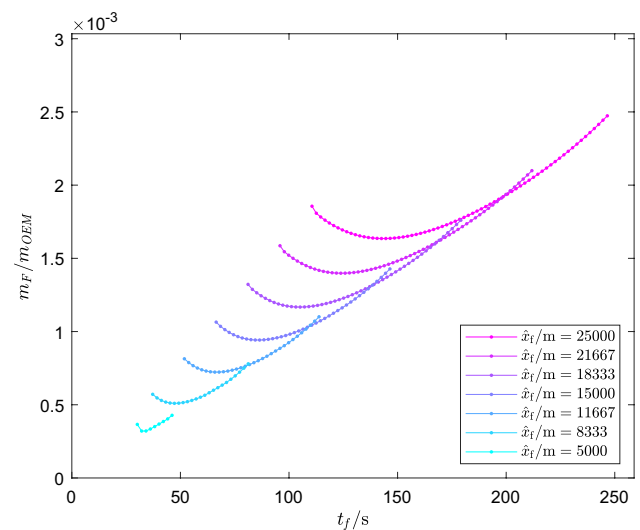


Fig. 8 Fuel/time relation at varying distance: longer distances increase the range of feasible durations

$$\begin{aligned}
 \hat{m}_s &= \frac{1}{2}(m_{OEM} + m_{MTOM}) \\
 \hat{h}_s &= 5000 \text{ m} \\
 \hat{x}_f &= 15,000 \text{ m} \\
 \hat{\gamma}_K &= -3 \text{ deg} \\
 \hat{u}_W &= 0 \text{ m s}^{-1} \\
 \mathcal{D}_{\hat{h}} \hat{u}_W &= 0 \text{ s}^{-1}
 \end{aligned}
 \tag{11}$$

We then vary each individual parameter, with increments in both directions, starting from the nominal value. For each resulting combination of parameter values we apply the procedure from Sect. 2.3.4 to identify the minimum fuel

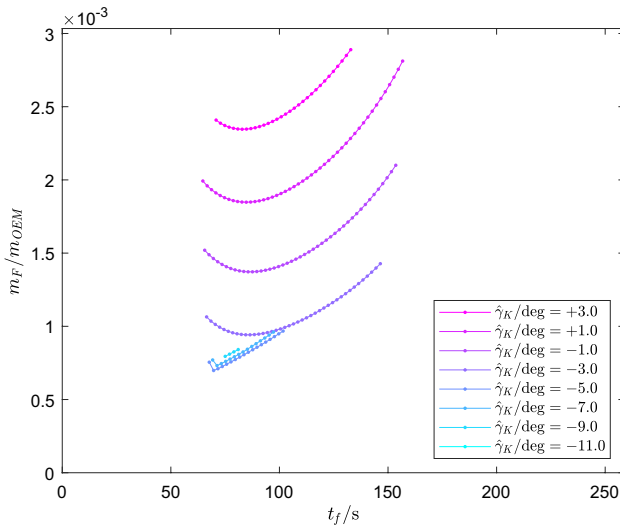


Fig. 9 Fuel/time relation at varying slope

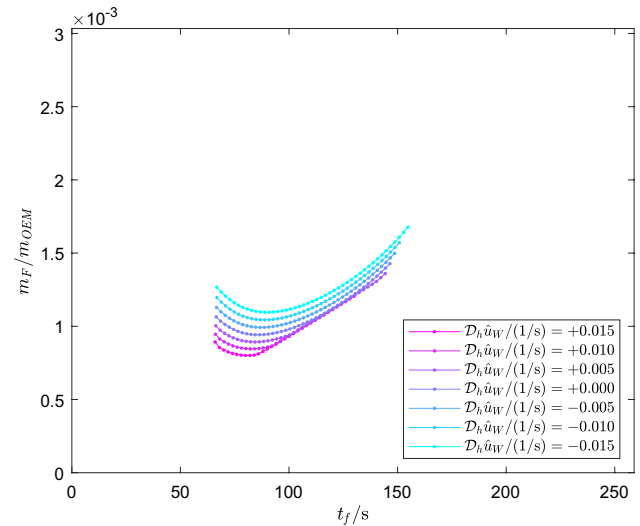


Fig. 11 Fuel/time relation with wind shear: in descent, headwind costs more fuel at high altitude

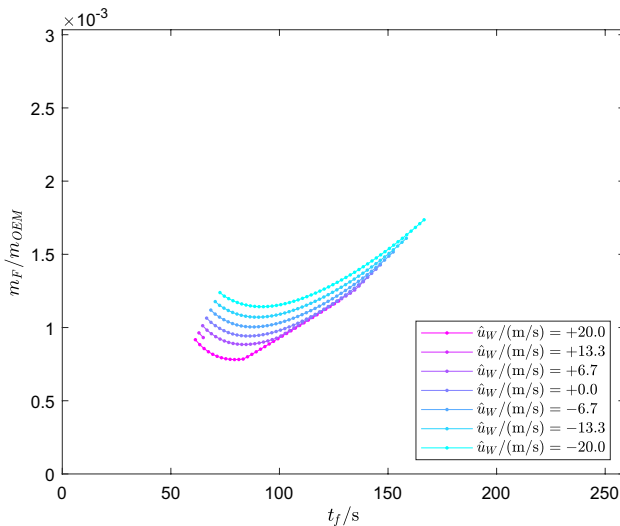


Fig. 10 Fuel/time relation at varying wind velocity: effects diminish at low speed

consumption as a function of flight duration. The results are presented in Figs. 6, 7, 8, 9, 10, 11.

Figure 6 shows the effect of the start mass varying from  $m_{OEM}$  to  $m_{MTOM}$ . We observe that low mass leads to higher fuel consumption at high speed (low duration), but also to a lower sensitivity with respect to flight duration, compared to cases at higher mass. For a descent segment, this can be explained by the gravity component tangent to the inclined flight path compensating a higher share of the drag if the mass is higher, thus reducing the required thrust and fuel flow. Conversely, at low speed more lift is required and thus more induced drag must be overcome, giving an advantage to a lighter aircraft. The minimum flight time is unaffected

by the mass variation, indicating that it is limited by independent factors such as the upper bound on CAS. Note that we cannot draw any conclusions on the maximum flight duration, as it is determined by the termination criteria set in Sect. 2.3.4.

Figure 7 shows the fuel-vs-time curve for varying start altitude. As expected, fuel consumption decreases with increasing altitude (and decreasing air density), while the maximum speed increases, resulting in lower minimum times (note that a constant CAS limit corresponds to an altitude-dependent TAS limit). Only the two cases at the highest altitudes do not match this expectation, but this may be due to numerical issues. Furthermore, we observe that a linear region appears at high altitude, indicating a transition from powered to idle descent in a certain speed range (cf. Fig. 2). This is a significant nonlinearity that needs to be well represented in surrogate models, since the idle descent regime comes with low fuel consumption.

Figure 8 shows the effect of varying distance. Since we consider absolute fuel consumptions and flight durations, the curves shift and scale quite linearly with the distance. Increasing distance also broadens the range of feasible durations.

Figure 9 shows the effects of slope variation. The steep descent case at  $\hat{\gamma}_K = -11$  deg failed because the final CAS constraint could not be achieved, and the case at  $\hat{\gamma}_K = -9$  deg is either almost infeasible or sensitive to numerical issues. The cases at  $\hat{\gamma}_K = -7$  deg and  $\hat{\gamma}_K = -5$  deg yield virtually the same results and clearly exhibit an idle descent section, which disappears when further increasing the slope. The point of minimum fuel consumption shifts to a lower speed when leaving the idle descent regime, whereas the minimum time varies only marginally.

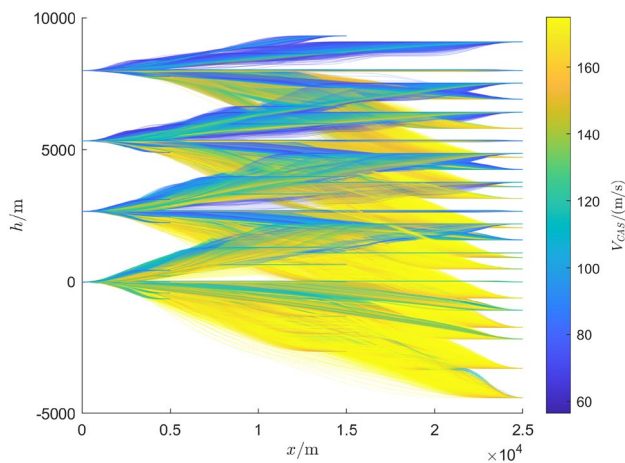


Fig. 12 Vertical profiles and CAS

Figures 7, 8, 9 indicate that fuel consumption rises at low flight time (high speed) can exhibit significantly different shapes, depending on the trajectory segment parameters. At short flight durations, the fuel consumption gradient tends to be larger, and relatively sharp kinks appear at the high-speed end of the idle descent region.

Figure 10 considers a varying wind speed. The case at  $\hat{u}_W = 13.33 \text{ m s}^{-1}$  failed after a single  $t_f$  step, but the remaining cases exhibit a high degree of regularity. Negative wind velocities correspond to a headwind (cf. Fig. 1) and lead to higher flight durations and fuel consumption. However, the effect diminishes with increasing flight duration, i.e., decreasing (kinematic) velocity. In cases with positive wind speed we observe lower flight durations and lower fuel consumption. In addition, the idle descent regime can be clearly seen here.

Figure 11 depicts the impact of varying wind shear. The linear wind gradient assumed here has no effect on the minimum flight time. However, it significantly affects the fuel consumption, with negative wind shear (meaning stronger headwind at higher altitude) resulting in higher fuel consumption than positive wind shear (meaning stronger

headwind at lower altitude) for this descent segment. The effect is more pronounced at high speeds than at low speeds.

### 3.1.2 Simultaneous parameter variations

Figure 12 shows the vertical profiles of a set of 167,571 fuel-minimal trajectory segments, color-coded by CAS. We ask the sophisticated reader to excuse the rasterized nature of this depiction, which makes it impossible to discern all individual lines but occupies less storage, and point out the diversity of the flight paths resulting from parameter variation. These range from almost straight to stepwise climb and descent profiles; the latter are in principle undesirable as CCO/CDO can yield higher efficiency [5], but the prescribed variation of the flight duration forces the trajectories to assume such shapes. We do not present the overall  $(t_f, m_F)$  curves here, as the results from individual parameter variation discussed in Sect. 3.1.1 provide more insight.

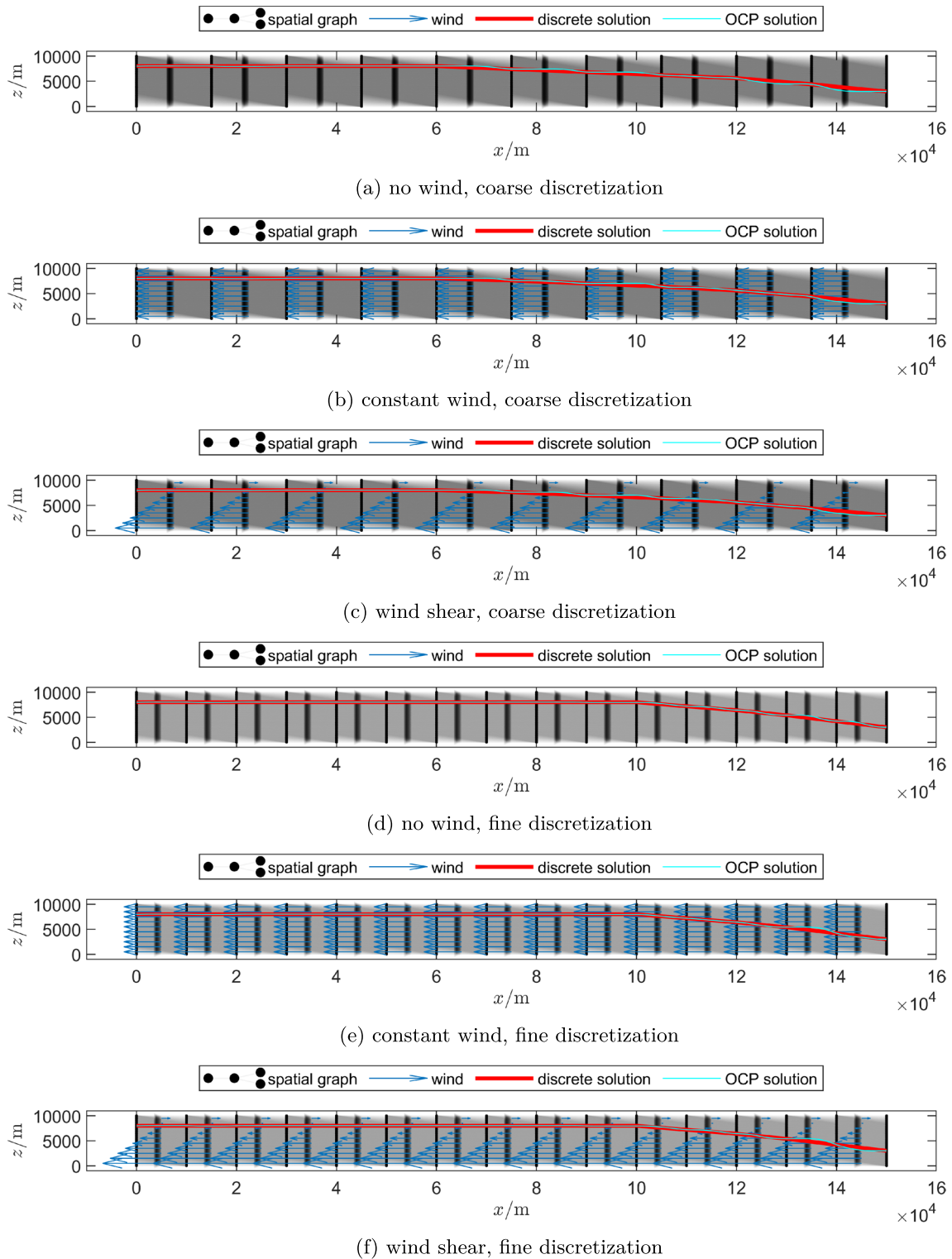
### 3.2 Verification: case study

Running the verification scenarios specified in Sect. 2.5 we obtain the results listed in Table 3; Fig. 13 presents the associated flight paths, and Fig. 14 the fuel consumptions. It is immediately apparent that the spacetime discretization has a strong influence on the resulting flight duration. Not only is there a difference of one twentieth in-flight durations between the cases with coarse discretization and the cases with fine discretization; also within each group the variation of wind speed and wind shear does not change the flight duration. This indicates that an even finer discretization may be required to take full advantage of the trade-off between flight duration and fuel consumption discussed earlier. However, further analysis is required to determine if the choice of space or time discretization is more critical.

In the presented cases the graph-based solution with a coarse discretization slightly overestimates the fuel-consumption, whereas with a fine discretization it underestimates the fuel consumption by a larger amount; the latter

**Table 3** Case study: results of graph-based routing and relative fuel consumption mismatch compared to a continuous OCP for a descent from 8000 to 3000 m over a distance of 150 km

| Case | Discretization |         |         | Wind  |   | Flight duration $t_f$ (s) | Fuel mismatch ratio (relative to OCP) |
|------|----------------|---------|---------|---|---|---------------------------|---------------------------------------|
|      | $x$ (m)        | $h$ (m) | $t$ (s) | $u_W$ at $h = 5000 \text{ m}$ ( $\text{m s}^{-1}$ ) | $\mathcal{D}_h u_W$ ( $\text{s}^{-1}$ ) |                           |                                       |
| a    | 15,000         | 200     | 10      | 0   | 0                                       | 800                       | 0.034                                 |
| b    | 15,000         | 200     | 10      | - 15  | 0                                       | 810                       | 0.025                                 |
| c    | 15,000         | 200     | 10      | - 10  | 0.003                                   | 800                       | 0.021                                 |
| d    | 10,000         | 200     | 5       | 0   | 0                                       | 785                       | - 0.079                               |
| e    | 10,000         | 200     | 5       | - 15  | 0                                       | 785                       | - 0.133                               |
| f    | 10,000         | 200     | 5       | - 10  | 0.003                                   | 785                       | - 0.072                               |

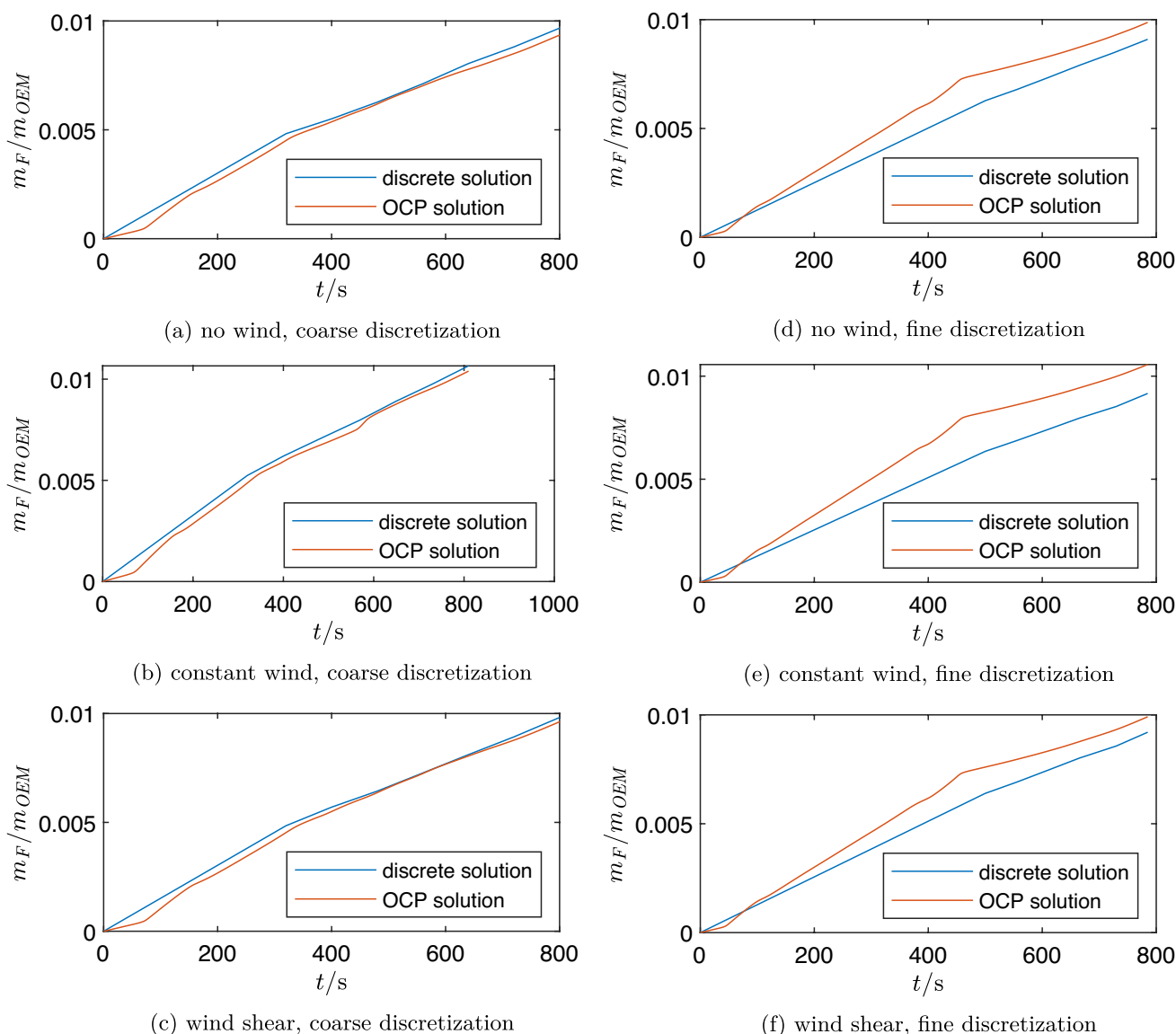


**Fig. 13** Case study: flight paths for a descent from 8000 to 3000 m over a distance of 150 km

can be clearly seen in Fig. 14. This is unexpected because the edge weights are determined from trajectory segment solutions with more constraints, compared to the full trajectory OCP; also, the edge weights in the graph are calculated

without taking into account that the aircraft mass reduces from one trajectory segment to the next. However, the simplifying assumptions on the CAS at the boundaries of individual trajectory segments may be a factor contributing





**Fig. 14** Case study: fuel consumption from graph-based routing and OCP for a descent from 8000 to 3000 m over a distance of 150 km

to the deviation. Future work may address this issue and perhaps lead to conservative estimates. Nevertheless, in the considered cases, we observe that the relative mismatch is of a magnitude that we consider acceptable, given the simplifying assumptions of this study. This, of course, does not yet indicate any correspondence of the results to reality. Further research is also required to compare these solutions to unrestricted optimal trajectories, i.e., trajectories calculated by optimal control methods without imposing constraints to follow the discrete reference solution in space and time.

The optimal control flight paths are shown in Fig. 13 often exhibit undesirable step descents, even though no constraints were imposed on the flight path angle at the segment boundaries and indeed some waypoints are crossed at nonzero flight path angles. We explain this behavior by

the combination of CDO constraints and fixed waypoint arrival times which strongly restricts the solutions. Future work should address this issue; possible remedies include improved assumptions on the segment boundary states or a more explicit treatment of these, as well as a relaxation of waypoint times. The latter, of course, needs to be designed in close coordination with the discrete planning method.

Furthermore, it is to be expected that the fuel cost models may perform worse in some situations, especially at the boundaries of the feasible region. Nevertheless, the case studies confirm that the overall concept of discrete 4D minimum-fuel trajectory planning for aircraft based on optimal control results for individual trajectory segments is feasible.

## 4 Conclusion and outlook

In this paper, we demonstrate an efficient way to generate segment-based surrogate models for the fuel consumption of aircraft, which may serve as a basis for subsequent graph-based trajectory optimization.

In Sect. 2.3, we introduce procedures that enable the fast generation of a large number of optimal control solutions for short aircraft trajectory segments. We find that a directed expansion of the parameter envelope using warm start procedures is an effective method to cover the parameter space. Minimum-time and minimum-fuel problems exhibit different convergence characteristics; solving a minimum-time problem first and then solving minimum-fuel problems with successively increasing prescribed flight duration turns out to be an efficient procedure to investigate the relationship between flight time and fuel consumption. How to reliably distinguish physical and computational infeasibility of individual problem instances remains an open question.

Section 2.4 investigates a grid-based surrogate modeling approach with smoothing to fill in missing data points. This is found to be a practical method for approximating fuel consumption and the flight envelope, but future research may reconsider alternative methods such as those mentioned in Appendix 2 to improve scalability and reduce the reliance on individual data points.

Section 3.1 exemplarily analyzes the individual influences of aircraft mass, altitude, distance, slope, wind velocity and wind shear on the fuel-vs-time trade-off observed on a short descent trajectory segment. We demonstrate that even the isolated effects of the parameters exhibit significant nonlinearities that must be represented correctly by surrogate models. Furthermore, this analysis may serve as a basis for the derivation of simple approximations capturing the main effects.

Section 3.2 demonstrates the basic feasibility of the hybrid optimization approach by comparing graph-based routing solutions with the associated continuous trajectories obtained from OCPs. Fuel consumption predictions are found to agree well in cases with a coarse discretization, but significant mismatches arise for a fine discretization. The neglected boundary conditions of velocity and flight path angle are identified as a weakness that should be reconsidered in the future.

Future work may also consider additional modeling aspects, such as configuration changes, as well as other aircraft classes and operating environments. For example, fully automated flight planning and guidance is envisioned in the Urban Air Mobility (UAM) domain, where the energy consumption of electric Vertical Take-Off and Landing (eVTOL) aircraft is a critical factor,

making trajectory optimization methods particularly relevant [29]. Additionally, parametric uncertainties of wind and aircraft mass may be considered, with the goal of quantifying their effects or generating trajectories that are robust against deviations from the nominal parameters.

## Appendix 1: Aircraft model: equations of motion

This section outlines core aspects of the aircraft model structure.

The position propagation is given by

$$\dot{x} = V_K \cos(\gamma_K) \cos(\chi_K), \quad (12a)$$

$$\dot{y} = V_K \cos(\gamma_K) \sin(\chi_K), \quad (12b)$$

$$\dot{h} = V_K \sin(\gamma_K). \quad (12c)$$

We incorporate a model of the ISA to calculate the static air density  $\rho$ , temperature  $\theta$ , pressure  $p$  and speed of sound  $a$  as functions of  $h$ . The wind velocity  $u_w$  is modeled as a linear function of altitude with a given gradient  $\mathcal{D}_h u_w$ .

The aerodynamic velocity vector in the local North-East-Down (NED) reference frame is given by

$$(\mathbf{V}_A)_O = [\dot{x}, \dot{y}, -\dot{h}]^\top - [u_w, 0, 0]^\top \quad (13)$$

From this, we determine the orientation of the aerodynamic reference frame  $A$ . CAS can be calculated according to BADA [31].

The total force vector  $(\mathbf{F}_T^G)_K$  is calculated in the kinematic reference frame; it comprises gravity (assuming a constant gravitational acceleration  $g = 9.80665 \text{ m s}^{-2}$ ), aerodynamic forces and propulsion forces. The propulsive and aerodynamic forces are calculated using BADA models, which we cannot reproduce in this paper. The interested reader may refer to [31]. We map the normalized thrust command variable  $\bar{C}_T$  to the admissible range dynamically calculated according to BADA equations. We obtain a thrust force  $T$  and the associated fuel mass flow  $\dot{m}_F$ . Since the considered aircraft model does not include full attitude information, we assume that the thrust force is collinear to the aerodynamic velocity. The fuel mass flow results in a mass reduction according to  $\dot{m} = -\dot{m}_F$  in case the mass state is modeled.

Analogously, BADA yields a maximum lift coefficient as a function of the aircraft states; we scale this coefficient by the command variable  $\bar{C}_L$  to obtain the lift coefficient  $C_L$ . Then we obtain the drag coefficient  $C_D$  as a function of  $C_L$ , aircraft states and controls. Using the dynamic pressure resulting from the atmosphere model and aerodynamic

velocity, as well as the reference wing area according to BADA, we calculate the lift and drag forces  $L$  and  $D$ .

Given the total force  $(\mathbf{F}_T^G)_K$  in the kinematic reference frame, the translation dynamics yield

$$\dot{V}_K = m^{-1} [1, 0, 0] (\mathbf{F}_T^G)_K, \quad (14a)$$

$$\dot{\chi}_K = m^{-1} [0, 1, 0] (\mathbf{F}_T^G)_K (V_K \cos(\gamma_K))^{-1}, \quad (14b)$$

$$\dot{\gamma}_K = -m^{-1} [0, 0, 1] (\mathbf{F}_T^G)_K V_K^{-1}. \quad (14c)$$

We calculate the total load factor as

$$n_{z,K} = m^{-1} [0, 0, 1] (\mathbf{F}_T^G)_K - g \cos(\gamma_K) \quad (15)$$

## Appendix 2: Candidate methods for surrogate modeling

This section points out several methods that may allow other researchers to improve on the grid-based surrogate modeling approach from Sect. 2.4.

The grid-based approach is not ideal when it comes to covering high-dimensional parameter spaces, as the number of samples scales exponentially with the dimension. Instead, Design of Experiments (DoE) methods may be applied to resolve higher-dimensional relations with fewer samples. Resulting in irregularly placed samples, this leads to the problem of scattered data interpolation in high dimensions. For the application under investigation, it is important to be able to efficiently evaluate the resulting model at a very large number of sample points. At the same time, the admissible parameter envelope needs to be represented, i.e., there needs to be a distinction between feasible and infeasible regions.

A basic approach that yields highly efficient surrogate models is to create a Linear Least Squares (LLS) or Non-linear Least Squares (NLS) fit using a set of basis functions or a parameterized model function. This yields continuous approximations without explicitly including the underlying data points (suitable for DoE) and allows incorporating prior knowledge, for example, basic analytical models of the density/altitude dependence of aircraft dynamics. While we made attempts along these lines in the early stages of the study, it became evident that the nonlinear cross-couplings between the parameters do not permit us to develop reasonably accurate model structures in higher dimensions. Nevertheless, it may be possible to create a suitable hierarchical function structure corresponding to the hierarchical envelope expansion introduced in Sect. 2.3.5. Alternatively, Artificial Neural Networks (ANNs) may be considered as a generic model structure; however, we expect these require a very high number of sample points. With all LLS/NLS methods,

the envelope may be represented either by explicit functions describing its boundaries or by an implicit function indicating set membership; coming up with a suitable basis functions is generally not trivial.

DELAUNAY triangulation is a method rather well-suited to the representation of the admissible envelope. From given data points, this method generates simplexes in the parameter space, within which barycentric interpolation allows to estimate intermediate values. The simplexes formed by valid data points inherently approximate the admissible envelope. In non-convex regions,  $\alpha$ -shapes can be applied to avoid extrapolation. However, constructing a full DELAUNAY triangulation does not scale well with the dimension of the parameter space. For this reason [8] introduces a sparse DELAUNAY triangulation around a given query point. While applicable to high-dimensional spaces, our preliminary experiments indicated that the performance might not suffice for the expected number of model evaluations.

Methods for scattered data interpolation or approximation based on Radial Basis Functions (RBFs), such as SHEPARD interpolation (see [42], for example) allow constructing an implicit indicator function to represent the envelope. However, preliminary experiments indicated that the density of data points required for high-dimensional problems may exceed our resources.

**Acknowledgements** The authors would like to thank Benno Hoch and Frauke Liers (Friedrich-Alexander Universität Erlangen-Nürnberg) for inspiring discussions throughout the joint research project *HOTRUN* and their efforts concerning the system-level optimization. Furthermore, the authors would like to thank their colleagues Johannes Diepolder, Tuğba Akman, Haichao Hong and Alexander Zwenig for comments and discussion.

**Author contributions** The authors contributed to this study as follows: Felix Schweighofer: problem formulation, implementation, data generation procedures, surrogate models, analysis, manuscript. Benedikt Grüter: project initiation, strategy, early-stage support, discussion, review. Florian Holzappel: strategy, guidance, vision.

**Funding** Open Access funding enabled and organized by Projekt DEAL. This research was funded by the German Federal Ministry for Economic Affairs and Climate Action through the project *HOTRUN—Holistic Optimization of Trajectories and Runway Scheduling* in the fifth call of the *Federal Aviation Research Programme* (LuFo-V), under Grant No. 20E1720A.

**Data Availability** Not applicable.

### Declarations

This document has been created with or contains elements of Base of Aircraft Data (BADA) Family 4 Release 4.2 which has been made available by EUROCONTROL to INSTITUTE OF FLIGHT SYSTEM DYNAMICS, TECHNICAL UNIVERSITY OF MUNICH. EUROCONTROL has all relevant rights to BADA. ©2021 The European Organisation for the Safety of Air Navigation (EUROCONTROL). All rights reserved. EUROCONTROL shall not be liable for any direct, indirect, incidental or consequential damages arising out of or in con-

nection with this product or document, including with respect to the use of BADA.

**Conflict of interest** The authors have no competing interests to declare that are relevant to the content of this article.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., Abramson, D.: Scheduling aircraft landings—the static case. *Transp. Sci.* **34**(2), 180–197 (2000). <https://doi.org/10.1287/trsc.34.2.180.12302>
2. Bittner, M., Rieck, M., Grüter, B., Holzapfel, F.: Optimal conflict free approach trajectories for multiple aircraft. In: ENRI International Workshop on ATM/CNS (EIWAC) (2015)
3. Bittner, M., Rieck, M., Grüter, B., Holzapfel, F.: Optimal approach trajectories for multiple aircraft considering disturbances and configuration changes. In: ICAS 30th International Congress of the International Council of the Aeronautical Sciences (2016)
4. Blanco, M., Borndörfer, R., Hoang, N.-D., Kaier, A., Schienle, A., Schlechte, T., Schlobach, S. (eds.) Solving Time Dependent Shortest Path Problems on Airway Networks Using Super-Optimal Wind (2016). <https://doi.org/10.4230/OASICS.ATMOS.2016.12>
5. Brain, D., Bastin, M.: The benefits of CCO/CDO operations—European task force findings. In: ICAO 2019 Environmental Report, pp. 145–152 (2019). <https://www.icao.int/environmental-protection/Pages/envrep2019.aspx>
6. Chaimatanan, S., Delahaye, D., Mongeau, M.: A hybrid metaheuristic optimization algorithm for strategic planning of 4D aircraft trajectories at the continental scale. *IEEE Comput. Intell. Mag.* **9**(4), 46–61 (2014). <https://doi.org/10.1109/MCI.2014.2350951>
7. Chaimatanan, S., Delahaye, D., Mongeau, M.: Aircraft 4D trajectories planning under uncertainties. In: 2015 IEEE Symposium Series on Computational Intelligence, pp. 51–58. *IEEE* (2015). ISBN 978-1-4799-7560-0. <https://doi.org/10.1109/SSCI.2015.18>
8. Chang, T.H., Watson, L.T., Lux, T.C.H., Butt, A.R., Cameron, K.W., Hong, Y.: Algorithm 1012: DELAUNAYSPARSE: interpolation via a sparse subset of the Delaunay triangulation in medium to high dimensions. *ACM Trans. Math. Softw. (TOMS)* **46**(4), 1–20 (2020). <https://doi.org/10.1145/3422818>
9. Chen, H., Zhao, Y.J.: Sequential dynamic strategies for real-time scheduling of terminal traffic. *J. Aircr.* **49**(1), 237–249 (2012). <https://doi.org/10.2514/1.C031503>
10. European Commission: Flightpath 2050: Europe's Vision for Aviation: Maintaining Global Leadership and Serving Society's Needs. Publications Office of the European Union (2012). <https://doi.org/10.2777/15458>
11. Fisch, F., Bittner, M., Holzapfel, F.: Optimal scheduling of fuel-minimal approach trajectories. In: 3rd International Air Transport and Operations Symposium (2012)
12. Fisch, F., Bittner, M., Holzapfel, F.: Optimal scheduling of fuel-minimal approach trajectories. *J. Aerosp. Oper.* **2**(3–4), 145–160 (2014)
13. Franco, A., Rivas, D., Valenzuela, A.: Optimal aircraft path planning considering wind uncertainty. In: 7th European Conference For Aeronautics and Space Sciences. <https://doi.org/10.13009/EUCASS2017-254>
14. Fürstenau, N., Heidt, A., Kappelke, M., Liers, F., Mittendorf, M., Weiß, C.: Pre-tactical planning of runway utilization under uncertainty: optimization and validation. In: Dirk, S. (ed.) Proceedings of the SESAR innovations days 2015. Eurocontrol (2015)
15. Gardi, A., Sabatini, R., Kistan, T.: Multiobjective 4D trajectory optimization for integrated avionics and air traffic management systems. *IEEE Trans. Aerosp. Electron. Syst.* **55**(1), 170–181 (2019). <https://doi.org/10.1109/TAES.2018.2849238>
16. Grüter, B., Bittner, M., Rieck, M., Diepolder, J., Holzapfel, F.: Optimal sequencing in atm combining genetic algorithms and gradient based methods to a bilevel approach. In: ICAS 30th International Congress of the International Council of the Aeronautical Sciences (2016)
17. Grüter, B., Bittner, M., Rieck, M., Diepolder, J., Holzapfel, F.: Bilevel homotopic aircraft sequencing using gradient-based arrival time assignment and direct optimal control. In: 57th Israel Annual Conference on Aerospace Sciences, IACAS 2017 (2017)
18. Grüter, B., Diepolder, J., Akman, T., Holzapfel, F.: Integrated bi-level arrival and departure time assignment for optimal trajectories. In: ENRI International Workshop on ATM/CNS 2017 (EIWAC 2017) (2017)
19. Grüter, B., Diepolder, J., Piprek, P., Akman, T., Holzapfel, F.: Nonlinear model predictive control for optimal aircraft sequencing. In: ICAS 31st Congress of the International Council of the Aeronautical Science (2018)
20. Harada, A.: Dynamic programming applications to flight trajectory optimization. *IFAC Proc. Volumes* **46**(19), 441–446 (2013). <https://doi.org/10.3182/20130902-5-DE-2040.00145>
21. Harada, A.: Air traffic efficiency evaluation by trajectory optimization and ads-b data. In: ICAS 30th International Congress of the International Council of the Aeronautical Sciences (2016)
22. Harada, A., Wickramasinghe, N.K., Brown, M., Fukuda, Y.: Analysis of air traffic efficiency using dynamic programming trajectory optimization. In: ICAS 29th International Conference (2014)
23. Hasevoets, N., Conroy, P., Eurocontrol: AMAN implementation guidelines and lessons learned. <https://www.skybrary.aero/sites/default/files/bookshelf/2416.pdf>
24. Heidt, A.: Uncertainty Models for Optimal and Robust ATM Schedules. Doctoralthesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) (2017). <http://nbn-resolving.de/urn:nbn:de:bvb:29-opus4-82536>
25. Hoch, B., Liers, F.: An integrated rolling horizon and adaptive-refinement approach for disjoint trajectories optimization. *Optim. Eng.* (2022). <https://doi.org/10.1007/s11081-022-09719-2>
26. Hoch, B., Liers, F., Neumann, S., Zaragoza Martínez, F.J.: The non-stop disjoint trajectories problem: Preprint (2020). [http://www.optimization-online.org/DB\\_HTML/2020/09/8015.html](http://www.optimization-online.org/DB_HTML/2020/09/8015.html)
27. HSL: A collection of fortran codes for large scale scientific computation. <http://www.hsl.rl.ac.uk/>
28. Korn, B., Helmke, H., Kuenz, A.: 4D trajectory management in the extended TMA: coupling AMAN and 4D FMS for optimized approach trajectories. In: ICAS 25th International Congress of the Aeronautical Sciences (2006)
29. Krammer, C., Schweighofer, F., Gierszewski, D., Scherer, S., Akman, T., Hong, H., Holzapfel, F.: Requirements-based generation of optimal takeoff and landing trajectories for electric aircraft. In: 33rd Congress of the International Council of the Aeronautical Sciences (2022)

30. Miyazawa, Y., Wickramasinghe, N.K., Harada, A., Miyamoto, Y.: Dynamic programming application to airliner four dimensional optimal flight trajectory. In: AIAA Guidance, Navigation, and Control (GNC) Conference, Reston, Virginia, 2013. American Institute of Aeronautics and Astronautics (2013). ISBN 978-1-62410-224-0. <https://doi.org/10.2514/6.2013-4969>
31. Nuic, A., Mouillet, V., Eurocontrol: User manual for the Base of Aircraft Data (BADA) family 4: Eec technical/scientific report no. 12/11/22-58
32. Oliveira, A.V.M., Narcizo, R.R., Caliari, T., Morales, M.A.V., Prado, R.: Estimating fuel-efficiency while accounting for dynamic fleet management: testing the effects of fuel price signals and fleet rollover. *Transp. Res. Part D Transp. Environ.* **95**, 102820 (2021). <https://doi.org/10.1016/j.trd.2021.102820>
33. Pacciarelli, D., D'Ariano, A., Pistelli, M.: Aircraft retiming and rerouting in vicinity of airports. *IET Intel. Transp. Syst.* **6**(4), 433–443 (2012). <https://doi.org/10.1049/iet-its.2011.0182>
34. Rieck, M., Bittner, M., Grüter, B., Diepolder, J., Piprek, P., Götlicher, C., Schwaiger, F., Hosseini, B., Schweighofer, F., Akman, T., Holzapfel, F.: FALCON.m user guide: Version 1.27. <http://www.falcon-m.com>
35. Samà, M., Palagachev, K., D'Ariano, A., Gerdtts, M., Pacciarelli, D., Hochreiter, R.: Terminal control area aircraft scheduling and trajectory optimization approaches. In: ITM Web of Conferences, vol. 14, no. 3, p. 00008 (2017). ISSN 2271-2097. <https://doi.org/10.1051/itmconf/20171400008>
36. Schmidt, J., Fügenschuh, A.: A two-time-level model for mission and flight planning of an inhomogeneous fleet of unmanned aerial vehicles. In: Cottbus Mathematical Preprints, volume COMP # 16 (2021a). <https://doi.org/10.26127/BTUOpen-5461>
37. Schmidt, J., Fügenschuh, A.: Planning inspection flights with an inhomogeneous fleet of micro aerial vehicles. In: Cottbus Mathematical Preprints, volume COMP # 21 (2021b). <https://doi.org/10.26127/BTUOpen-5656>
38. SESAR Joint Undertaking. European ATM Master Plan: Digitalising Europe's Aviation Infrastructure: Executive View. Publications Office of the European Union (2020). ISBN 978-92-9216-134-7. <https://doi.org/10.2829/695700>
39. Sáez, R., Dalmau, R., Prats, X.: Optimal assignment of 4D close-loop instructions to enable CDOs in dense TMAs. In: 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), pp. 1–10. IEEE (2018). ISBN 978-1-5386-4112-5. <https://doi.org/10.1109/DASC.2018.8569597>
40. Sáez, R., Polishchuk, T., Schmidt, C., Hardell, H., Smetanová, L., Polishchuk, V., Prats, X.: Automated sequencing and merging with dynamic aircraft arrival routes and speed management for continuous descent operations. *Transp. Res. Part C: Emerg. Technol.* **132**, 103402 (2021). <https://doi.org/10.1016/j.trc.2021.103402>
41. Teoh, R., Schumann, U., Stettler, M.E.J.: Beyond contrail avoidance: efficacy of flight altitude changes to minimise contrail climate forcing. *Aerospace* **7**(9), 121 (2020). <https://doi.org/10.3390/aerospace7090121>
42. Thacker, W.I., Zhang, J., Watson, L.T., Birch, J.B., Iyer, M.A., Berry, M.W.: Algorithm 905: SHEPPACK: modified Shepard algorithm for interpolation of scattered multivariate data. *ACM Trans. Math. Softw.* **37**(3), 1–20 (2010). <https://doi.org/10.1145/1824801.1824812>
43. Toratani, D.: Application of merging optimization to an arrival manager algorithm considering trajectory-based operations. *Transp. Res. Part C Emerg. Technol.* **109**, 40–59 (2019). <https://doi.org/10.1016/j.trc.2019.09.015>
44. Toratani, D., Itoh, E.: Effects on optimal merging trajectories with allocation optimization from the trade-off between fuel consumption and flight time. *SICE J. Control Meas. Syst. Integr.* **11**(3), 182–189 (2018). <https://doi.org/10.9746/jcmsi.11.182>
45. Toratani, D., Delahaye, D., Ueno, S., Higuchi, T.: Merging optimization method with multiple entry points for extended terminal maneuvering area. In: EIWAC 2015, 4th ENRI International Workshop on ATM/CNS, Tokyo, Japan (2015). <https://hal-enac.archives-ouvertes.fr/hal-01288387>
46. Toratani, D., Wickramasinghe, N.K., Itoh, E.: Study on the arrival manager maximizing the benefit of four-dimensional trajectory based operations. In: ICAS 31st Congress of the International Council of the Aeronautical Science (2018). [http://www.icas.org/ICAS\\_ARCHIVE/ICAS2018/data/preview/ICAS2018\\_0315.htm](http://www.icas.org/ICAS_ARCHIVE/ICAS2018/data/preview/ICAS2018_0315.htm)
47. Watling, D.P., Connors, R.D., Chen, H.: Fuel-optimal truck path and speed profile in dynamic conditions: an exact algorithm. *Eur. J. Oper. Res.* (2022). <https://doi.org/10.1016/j.ejor.2022.07.028>
48. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* **106**(1), 25–57 (2006). <https://doi.org/10.1007/s10107-004-0559-y>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.