

# Generation of Vertical Profiles with Neural Ordinary Differential Equations Trained on Open Trajectory Data

Gabriel Jarry <sup>\*</sup>,<sup>1</sup> and Xavier Olive <sup>\*</sup>,<sup>2</sup>

<sup>1</sup>Aviation Sustainability Unit, EUROCONTROL Innovation Hub, Brétigny-sur-Orge, France

<sup>2</sup>ONERA DTIS, Université de Toulouse, France

\*Corresponding author: [gabriel.jarry@eurocontrol.int](mailto:gabriel.jarry@eurocontrol.int); [xavier.olive@onera.fr](mailto:xavier.olive@onera.fr)

(Received: 6 Nov 2025; Revised: 2 Feb 2026; Accepted: 12 Mar 2026; Published: 19 Mar 2026)

(Editor: Martin Strohmeier; Reviewers: Raúl Sáez, Manuel Waltert, and Yutong Chen)

## Abstract

Recent advances in Neural Ordinary Differential Equations (Neural ODE) have shown that high-fidelity aircraft dynamics can be learned from flight recorder data, but such proprietary datasets remain largely inaccessible. In this study, we extend these principles to open trajectory data by training a Neural ODE model on Automatic Dependent Surveillance–Broadcast (ADS-B) and Mode S Enhanced Surveillance (EHS) information retrieved from the OpenSky Network. The model focuses on reconstructing the vertical dynamics of transport aircraft using only openly available surveillance variables. The resulting framework learns continuous-time dynamics that remain physically consistent through embedded kinematic relations, demonstrating that realistic vertical profiles can be generated solely from open surveillance data. This work contributes to reproducible, data-driven performance modelling and supports the broader adoption of open, physics-guided learning methods in aviation research.

**Keywords:** Neural Ordinary Differential Equations; Aircraft Performance; Trajectory Generation; ADS-B

## 1. Introduction

Accurate trajectory prediction and generation are essential to modern air traffic management (ATM), airline operations, and environmental performance modelling. Reliable forecasts of aircraft motion have applications in conflict detection and resolution, sector capacity planning, and the implementation of trajectory-based operations. Beyond operational safety and efficiency, realistic trajectory models also enable quantitative assessments of fuel consumption, emissions, and noise exposure which are key metrics in the transition toward sustainable aviation.

Traditional approaches to aircraft performance modelling rely on analytical formulations of the point-mass equations of motion, such as those implemented in the Base of Aircraft Data (BADA) [1, 2]. These models provide interpretable and physically consistent representations of nominal aircraft behaviour but depend on manufacturer-supplied performance parameters and simplifying assumptions that can limit their fidelity under real-world conditions. Empirical analyses of surveillance data often reveal systematic deviations between simulated and observed trajectories, particularly in the climb and descent phases, where speed management and mass evolution exert strong influence. Open-source initiatives such as OpenAP [3, 4] have addressed some of these limitations by offering transparent, data-driven performance parametrizations derived from large-scale ADS-B data.

In parallel, the increasing availability of digital flight data has enabled a new generation of data-driven performance models. Studies using Quick Access Recorder (QAR) data demonstrate that neural networks and continuous-time formulations such as Neural Ordinary Differential Equations (Neural ODEs) can reconstruct high-fidelity dynamics directly from measured flight parameters [5]. These approaches capture both deterministic aircraft physics and operational variability, producing continuous and differentiable representations of real flight behaviour.

However, QAR data remain proprietary and inaccessible to most researchers and operational stakeholders. This work aims to extend the same methodological principles to open trajectory data collected through the OpenSky Network [6]. By leveraging globally available ADS-B and Mode S Enhanced Surveillance (EHS) information, we demonstrate that it is possible to approximate the vertical, longitudinal dynamics of transport aircraft with a Neural ODE trained solely on open data. The proposed framework reconstructs altitude and speed evolution along the flight path while preserving physical consistency through analytical kinematic relations. Methodologically, this work demonstrates how hybrid architectures combining analytical constraints with learnable dynamics can compensate for the lack of observability (e.g., unknown mass, thrust) inherent to open datasets.

Compared with QAR-based approaches, this study emphasizes the challenges and opportunities specific to open surveillance data: hidden control inputs, incomplete coverage of Mode S registers, and the need for careful preprocessing and intent inference. Despite these constraints, our findings show that open data can reproduce key dynamical patterns observed in QAR-based studies, thereby contributing to transparent, reproducible, and scalable aircraft performance modelling.

The remainder of this paper is organized as follows. Section 2 reviews existing work on model-based, data-driven, and hybrid approaches to aircraft trajectory prediction. Section 3 describes the data sources and preprocessing steps used to construct the training dataset. Section 4 presents the Neural ODE formulation and its integration with analytical kinematic relations. Results and evaluation metrics are discussed in Section 5, followed by limitations and perspectives in Section 6.

## 2. Literature review

Trajectory prediction and generation underpin a wide range of ATM and airline operations, from conflict detection and sequencing to environmental assessment and strategic planning. Two complementary approaches dominate: model-based methods based on aircraft performance and point-mass dynamics, and data-driven methods that learn spatio-temporal regularities from large-scale surveillance and weather datasets.

Physics-based point-mass formulations and performance databases remain the historical backbone of operational trajectory prediction. The EUROCONTROL BADA model provides aircraft-type parameters and procedures for 4D trajectory computation under forecast conditions [1, 2], while OpenAP extends these capabilities to the research community with an open-source framework [3, 4]. Beyond canonical models, the literature formalizes trajectory representation, compression, and optimization using splines, principal component analysis, and connections to optimal control and wavefront propagation under constraints [7]. Recent efforts by Poll and Schumann refine performance characteristics and cruise fuel modelling from aerodynamic theory and empirical data [8]. These approaches provide interpretability and controllability, but rely on accurate state and intent estimates and may under-represent operational variability.

Data-driven methods have achieved notable gains in 4D trajectory prediction using supervised learning with surveillance and meteorological features. Early work by Ayhan and Samet demonstrated machine-learning-based prediction of arrival times and trajectories using historical and weather data [9]. Sequence models then became prevalent: Liu and Hansen proposed a seq2seq long short-

term memory (LSTM) [10], Pang et al. combined convolutional encoders of convective weather “cubes” with recurrent decoders [11], and Ma and Tian introduced a convolutional neural networks (CNN)-LSTM on ADS-B data [12]. Online frameworks fuse point-mass physics, BADA parameters, and ADS-B conformance to update intent and improve estimated time of arrival precision [13]. Generative models address multi-modality and error accumulation: generative adversarial network (GAN)-based forecasters produce full-sequence 4D trajectories [14], variational autoencoders (VAE)/GAN generators improve realism in sparse regimes [15], privacy-preserving LSTM-GANs synthesize flows while mitigating re-identification risk [16], and imitation-learning methods such as TrajGAIL learn realistic route choices and flow-level statistics [17]. Data-driven models excel at capturing variability and airline-specific practices, though generalization off-distribution and flyability remain challenging [18].

Hybrid strategies combine the adaptability of data-driven approaches with the consistency of physics-based models, either by calibrating physics-based predictors using data or by embedding known dynamics into neural architectures. Calibration work has improved climb predictions by inferring aircraft-specific parameters from radar data [19]. Physics-informed learning enforces equations of motion and aerodynamic limits via loss functions or architectural constraints. Neural ODEs formalize these approaches as continuous-time dynamical systems, trained efficiently with adjoint methods and evaluated using ODE solvers [20, 21]. Aerospace applications include learning residual dynamics under icing [22], hypersonic glide vehicle prediction [23], and differentiable predictive control for unmanned aerial vehicles [24]. Practical challenges remain regarding stiffness, training cost, stability, and hybrid dynamics with discrete modes [25]. Recently, a study introduced the NODE-FDM architecture [5], demonstrating that Neural ODEs trained on Quick Access Recorder (QAR) data can accurately reconstruct high-fidelity aircraft dynamics. Building on this foundation, the present study investigates whether comparable continuous-time formulations can be learned from open surveillance data alone.

Beyond prediction, ATM applications also require the generation of realistic trajectories and flows with quantified realism and flyability. Statistical density models, such as Gaussian mixture models and vine copulas, combined with dimensionality reduction, reproduce flow statistics and go-around patterns while balancing realism and tractability [26, 27]. Evaluation frameworks assess operational realism, statistical coherence, similarity to observations, and flyability via simulator replay [28]. Interactive FPCA-based pipelines support clustering, deformation, and consistent generation for what-if analyses, for instance in noise footprint studies [29]. Aviation-focused GANs synthesize approach paths and identify atypical trajectories for safety management [30], while recent adversarial maximization–minimization methods assemble trajectories from mined manoeuvres under performance constraints, improving fidelity, diversity, and controller consistency [31].

Overall, model-based methods provide transparency and physical consistency but struggle with latent intent and operational diversity, while data-driven predictors capture variability and multi-modality yet must address generalization, uncertainty, and physical plausibility, and physics-guided hybrids, including Neural ODE frameworks, offer a promising middle ground for robustness and realism. Within this spectrum, our approach falls into the physics-guided hybrid category, combining large-scale ADS-B data with analytical kinematics, regression-based propulsion modeling, and a Neural ODE formulation. Compared with purely model-based approaches such as BADA, it emphasizes accurate reconstruction of altitude and speed trajectories while remaining suitable for environmental impact assessments. Furthermore, unlike discrete sequence models (e.g., LSTMs), the continuous-time Neural ODE formulation naturally handles irregular sampling inherent to surveillance data and allows for the seamless integration of physical constraints, ensuring both operational realism and kinematic consistency.

### 3. Data sources

The dataset used in this study was extracted from the historical databases of the OpenSky Network [6]. We selected a representative subset of aircraft covering multiple ICAO aircraft type designators (typecode), with particular care to include a broad spectrum of airline operators so as to mitigate any operator-specific biases in flight practices or configuration preferences. The final sample comprises eleven major categories, summarized in Table 1. For each selected aircraft type, we retrieved ADS-B trajectories recorded worldwide at 20-day intervals between 1 October 2024 and 15 October 2025, providing a temporally balanced dataset across one year of operations.

**Table 1.** Number of selected aircraft and airlines by typecode

typecode	unique aircraft	unique airlines
A320	99	60
B738	98	39
A20N	97	46
B38M	97	24
A21N	97	38
A321	96	39
A333	95	36
A359	90	24
A319	87	28
AT76	85	28
E190	84	18

Each trajectory was matched with metadata on departure and destination airports using the flights table from the OpenSky Network database. For the same set of flights, we retrieved corresponding Enhanced Mode S (EHS) messages, specifically those containing downlinked reports for registers BDS 4,0, BDS 5,0, and BDS 6,0, which were decoded following the logic detailed in [32] and implemented in rs1090<sup>1</sup>. The variables available in these data blocks are summarized in Table 2.

Since EHS data availability depends on Secondary Surveillance Radar (SSR) configurations and interrogation schemes, only trajectory segments where these reports were present were retained. The resulting data were filtered to remove anomalous or inconsistent samples, using criteria similar to those in [33], and segments with missing values or durations shorter than four minutes were discarded. All valid trajectories were then resampled at 4-second intervals to obtain a uniform temporal resolution suitable for subsequent analysis.

**Table 2.** Information available for each Broadcast Data Selector (BDS)

<b>ADS-B available in broadcast mode</b>	
BDS0,5	latitude, longitude, barometric altitude
BDS0,8	callsign
BDS0,9	ground speed, true track angle, vertical rate
<b>EHS available upon query by a SSR</b>	
BDS4,0	selected altitude (FMS or MCP)
BDS5,0	true air speed (TAS), roll angle
BDS6,0	indicated air speed (IAS), Mach number

<sup>1</sup><https://docs.rs/rs1090/>

To complement the Mode S and ADS-B data, each trajectory was cross-matched with meteorological fields from the ERA5 reanalysis dataset [34], provided by the European Centre for Medium-Range Weather Forecasts (ECMWF). The matching was performed using the *fastmeteo* Python library [35], which enables efficient spatiotemporal interpolation of meteorological variables along aircraft trajectories. Wind components and ambient temperature were extracted at the aircraft's position and time. Using these variables, the true airspeed (TAS), Mach number, and calibrated airspeed (CAS) were consistently recalculated.

The Mode S EHS data directly provided the selected altitude (BDS 4,0), but not the commanded targets for speed or vertical rate. To infer these quantities, an automated plateau detection algorithm was developed and applied to the time series of Mach number, calibrated airspeed (CAS), and vertical speed. Each signal was optionally smoothed using either a Savitzky–Golay filter or a rolling mean to suppress short-term fluctuations while preserving the general trend.

The smoothed series were then differentiated, and quasi-steady intervals were identified when the absolute first derivative remained below a fixed tolerance for at least a minimum duration. For each such plateau, the mean signal value was assigned as the representative selected or commanded parameter. Mach plateaus detected above 20,000 ft were labelled as *mach\_sel*. To avoid redundancy, these intervals were excluded from the subsequent CAS analysis so that constant calibrated airspeed segments (*cas\_sel*) were only detected in climb and descent phases.

The same procedure was applied to the vertical-speed series to identify sustained climb or descent segments with a mean vertical rate magnitude exceeding 50 ft/min for at least 30 s. When no stable phase was detected, the selected vertical rate (*vz\_sel*) was set to zero. This process yielded three continuous variables, *mach\_sel*, *cas\_sel*, and *vz\_sel*, representing the estimated commanded targets for the aircraft's longitudinal and vertical dynamics.

Aircraft mass is a key determinant of climb performance and overall energy management, yet it cannot be directly observed from ADS-B or Mode S Enhanced Surveillance (EHS) data. Moreover, fuel flow information is generally unavailable in open datasets. To provide the model with contextual information related to the flight mission and an indirect indicator of the aircraft mass evolution, we introduce two geometric proxies: (i) the great-circle distance from the current aircraft position to the departure airport, and (ii) the great-circle distance to the destination airport. As a result, the model can implicitly account for mass-dependent effects without relying on proprietary or non-observable parameters.

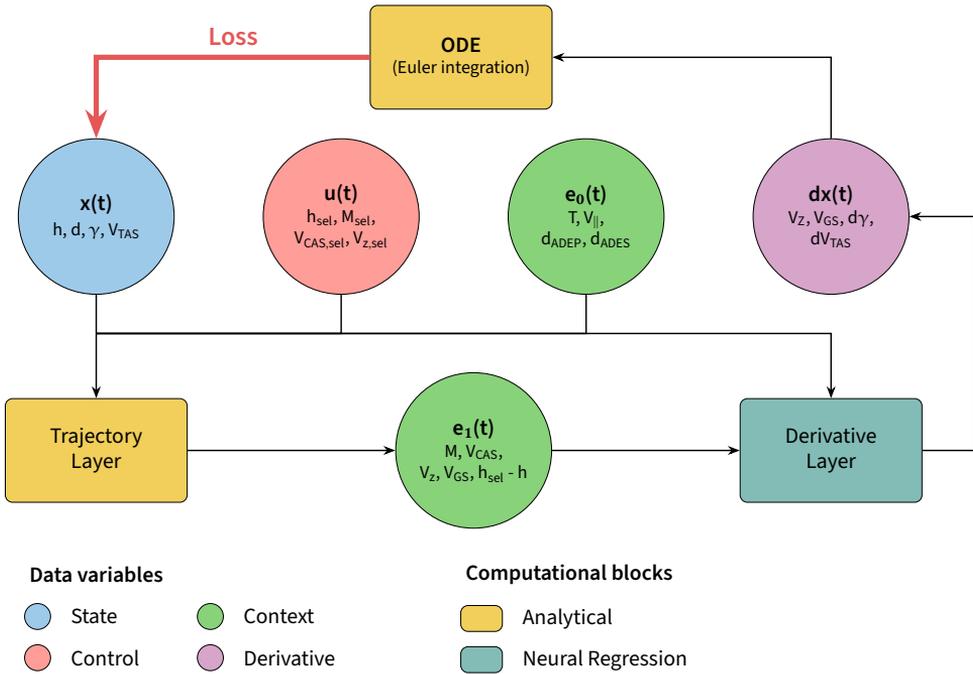
After all filtering, enrichment, and resampling steps, we obtained a curated set of trajectories for each aircraft type, summarized in Table 3. The training and test sets were split by aircraft identifier to avoid data leakage across subsets. Each type includes approximately one hundred unseen flights reserved for testing, while the remainder were used for training. Flight-hour statistics are provided to illustrate the total data volume and ensure that both short- and long-range aircraft are proportionally represented in the final dataset.

## 4. Methodology

We used a simplified version of the NODE-FDM architecture [5]. This architecture is organised as a modular, physics-informed architecture designed to estimate the temporal evolution of aircraft states (Figure 1). The simplified model combines analytical equations and a Neural ODE formulation to ensure both physical consistency and flexibility, with a focus on the vertical profile and speed management. In this study, the scope is limited to the longitudinal dimensions of the trajectory: altitude, true airspeed, flight path angle, and along-track distance. Lateral dynamics (e.g. track angle and horizontal position) are not included at this stage and are left for future investigation.

**Table 3.** Training, validation, and test dataset composition by aircraft type.

Type	Train		Validation		Test	
	Traj.	Hours	Traj.	Hours	Traj.	Hours
A320	1925	2184.7	476	533.3	139	143.2
A20N	1879	2161.1	449	564.1	159	151.4
A21N	1637	2223.3	404	514.2	108	144.9
A319	2714	2582.0	671	768.7	175	189.6
A321	1874	2159.3	443	584.7	146	173.2
A333	956	778.0	222	175.3	105	105.9
A359	916	792.7	223	206.0	102	83.0
AT76	1980	1079.3	477	253.0	172	105.2
B38M	1349	1921.0	300	467.9	115	179.7
B738	2125	2461.2	470	585.6	154	191.1
E190	3948	3498.5	903	864.9	208	161.3

**Figure 1.** Architecture of the model representing the single-step derivative estimation. The full trajectory is built autoregressively through ODE integration.

The state vector is defined as:

$$x(t) = \{h(t), d(t), \gamma(t), V_{TAS}(t)\}, \quad (1)$$

where  $h$  denotes the altitude,  $d$  the along-track distance,  $\gamma$  the flight path angle,  $V_{TAS}$  the true air-speed. This state vector corresponds to the a simplified standard point-mass formulation (without mass), widely applied in aircraft trajectory prediction and optimization.

**Table 4.** List of parameters used in the model with source units and converted SI units.

Feature	Symbol	Source Unit	SI Unit	Description
<b>State variables <math>x(t)</math></b>				
altitude	$h$	ft	m	Standard altitude above mean sea level
distance along track	$d$	nm	m	Along-track distance from departure
flight path angle	$\gamma$	deg	rad	Angle between velocity vector and horizon
true airspeed	$V_{TAS}$	kt	m/s	Speed relative to the surrounding air mass
<b>Control variables <math>u(t)</math></b>				
selected altitude	$h_{sel}$	ft	m	Altitude target extracted from EHS messages
selected Mach	$M_{sel}$	kt	m/s	Estimated Mach control
selected speed	$V_{CAS,sel}$	kt	m/s	Estimated CAS control
selected vertical speed	$V_{z,sel}$	ft/min	m/s	Estimated vertical speed control
<b>Context and intermediate variables <math>e(t)</math></b>				
<b>Context variables <math>e_0(t)</math></b>				
Air temperature	$T$	K	K	Ambient static air temperature (ERA5)
headwind component	$V_{\parallel}$	kt	m/s	Wind component along the trajectory (ERA5)
departure airport distance	$d_{ADEP}$	nm	m	Great circle distance to departure airport from current position
arrival airport distance	$d_{ADES}$	nm	m	Great circle distance to arrival airport from current position
<b>Trajectory variables <math>e_1(t)</math></b>				
Mach number	$M$	-	-	Ratio of $V_{TAS}$ to speed of sound
Calibrated airspeed	$V_{CAS}$	kt	m/s	Airspeed derived from dynamic pressure
vertical speed	$V_z$	ft/min	m/s	Rate of climb or descent
ground speed	$V_{GS}$	kt	m/s	Speed relative to the ground
selected altitude difference	$h_{sel} - h$	ft	m	Difference between selected and current altitude

The model architecture (Figure 1) processes this state vector through two sequential layers to compute the state derivatives  $dx/dt$  needed for trajectory integration:

- A **trajectory layer** (analytical): Given the current state  $x(t)$ , control inputs  $u(t)$ , and context variables  $e_0(t)$ , it analytically derives intermediate kinematic variables  $e_1(t)$  (Mach number, CAS, vertical speed, ground speed).
- A **derivative layer** (neural network): Implemented as a structured layer (detailed below), it takes these intermediate variables to estimate  $dV_{TAS}$  and  $d\gamma$ , while  $V_z$  and  $V_{GS}$  are passed directly. These derivatives are then integrated using an ODE solver (Euler scheme in this work) to generate continuous-time trajectories.

The evolution of the state vector is modelled as:

$$\frac{dx}{dt} = f_{\theta}(x(t), u(t), e(t)), \quad (2)$$

where  $x(t)$  denotes the aircraft state vector at time  $t$ ,  $u(t)$  represents the control variables, and  $e(t)$  corresponds to the context and intermediate variables (see Table 4 for a complete list). Note that the control inputs  $u(t)$  and context variables  $e_0(t)$  are exogenous inputs available for all  $t$ , and thus do not need to be estimated autoregressively.  $f_{\theta}$  is represented by the **derivative layer** (a neural network). We compute the predicted trajectories through numerical integration of the ODE using an explicit Euler solver, implemented via the `odeint` function from the `torchdiffeq` library [20]. In this work, we use an explicit Euler scheme to integrate the Neural ODE, as the effective temporal resolution of the data (1–4 s after interpolation) and the smooth, non-stiff nature of aircraft dynamics make higher-order solvers such as 4th order Runge–Kutta (RK4) unnecessary. Euler integration was found to be numerically stable and computationally efficient, though future work could explore higher-order methods to assess whether noticeable gains in accuracy can be achieved at this sampling rate.

The **trajectory layer** provides deterministic conversions from the state variables and context conditions to additional kinematic variables, ensuring consistency with physical laws. Notably, vertical speed  $V_z$  and ground speed  $V_{GS}$  are computed analytically in this layer and passed directly to the **derivative layer** without further transformation. Specifically:

$$V_z = V_{TAS} \cdot \sin(\gamma), \quad (3)$$

$$M = \frac{V_{TAS}}{a}, \quad a = \sqrt{\gamma_{air} R T}, \quad (4)$$

$$V_{GS} = V_{TAS} - V_{\parallel}, \quad (5)$$

where  $V_z$  is the vertical speed,  $M$  the Mach number,  $a$  the local speed of sound,  $\gamma_{air}$  the ratio of specific heats,  $R$  the perfect gas constant,  $T_{OAT}$  the outside air temperature, and  $V_{\parallel}$  the headwind component.

The **derivative layer** is implemented using a structured layer [5], which serves as a reusable building block within the model architecture. Each structured layer consists of three main components: an input normalizer, a backbone, and multiple output heads. The inputs are first normalized according to their empirical statistics and concatenated into a fixed-size representation. This representation is processed by the backbone, a three-layer fully connected network with 48 hidden neurons per layer, followed by ReLU activations. The extracted features are then passed to task-specific output heads, each implemented as a two-layer perceptron with 48 neurons per layer. This design allows the model to share a common latent representation while adapting flexibly to heterogeneous output variables (continuous or binary). Finally, the outputs of each head are denormalized to recover values on the original physical scale. We deliberately excluded dropout layers from our Neural ODE architecture. Unlike in discrete architectures (e.g., ResNets), applying standard dropout within continuous-time models is known to destabilize training, as the stochastic masking conflicts with the smooth dynamics required for stable ODE integration [36].

The model is trained by minimizing the difference between predicted and observed trajectories over autoregressive sequences of  $N = 60$  time steps. For each training sample, the ODE is integrated forward from an initial state  $x(t_0)$  to generate a sequence of state predictions. The composite loss function aggregates errors across all time steps and output groups (state variables, aircraft angles, and engine parameters):

$$\mathcal{L} = \sum_{k=1}^N \sum_{i \in C} \alpha_i \ell_i(y_i^{\text{pred}}(t_k), y_i^{\text{true}}(t_k)), \quad (6)$$

where  $\ell_i$  is the error metric for feature  $i$ ,  $\alpha_i$  is a weighting coefficient, and  $C = \{x(t), \}$  (see Table 4). In other words, the loss function directly compares the model's predicted variables with those recorded by the ADS-B data. The resulting error signals are then used to adjust the weights of the **derivative layer**, gradually improving its prediction accuracy.

In this paper,  $\ell_i$  corresponds to a mean squared error (MSE) and the weighting coefficients  $\alpha_i$  are chosen inversely proportional to the empirical standard deviation of each variable, in order to mitigate differences in scale and absolute value. This formulation allows balancing between accuracy in integrated state values, and regressed engine and aircraft variables.

The model is trained on one GPU (RTX A6000 Ada Gen) using PyTorch for 40000 gradient steps with batch size of 512 and checkpoints on a validation set composed of 20% of the train set. Optimization was performed with the AdamW optimizer ( $\text{lr} = 1 \times 10^{-4}$ ,  $\text{weight\_decay} = 1 \times 10^{-4}$  corresponding to L2 regularization).

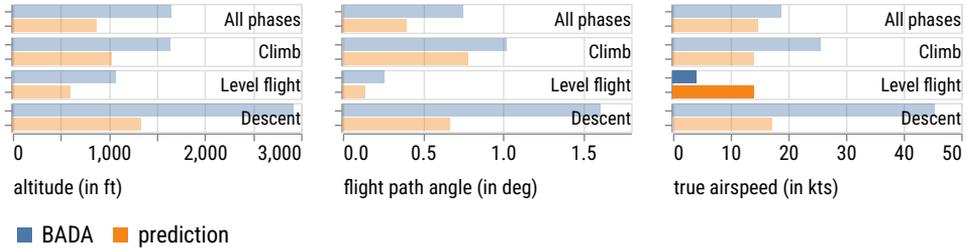
## 5. Results

To generate physically consistent benchmark trajectories from flight data, we used a BADA-based methodology combining the pyBADA library, in particular routines from the TCL.py (Trajectory Calculation Library) module, and the BADA 4.2 performance coefficients, together with control inputs mapped from the ADS-B data. In this setup, the BADA4 aircraft performance model was instantiated for each aircraft typecode, while trajectory propagation relied on the TCL routines `accDec_time`, `constantSpeedLevel`, `constantSpeedRating_time`, and `constantSpeedROCD_time`, applied over fixed 4-second intervals. For each ADS-B record, the current aircraft state (altitude, true airspeed, Mach number, temperature deviation from ISA, configuration, and mass) was extracted and mapped to the corresponding BADA4 input variables. Calibrated airspeed and Mach number were derived from standard atmosphere relationships, and along-track wind is also incorporated to ensure consistency with the recorded operational conditions. Importantly, the evaluation presented here concerns the entire BADA-based trajectory generation pipeline, not only the underlying BADA4 performance model.

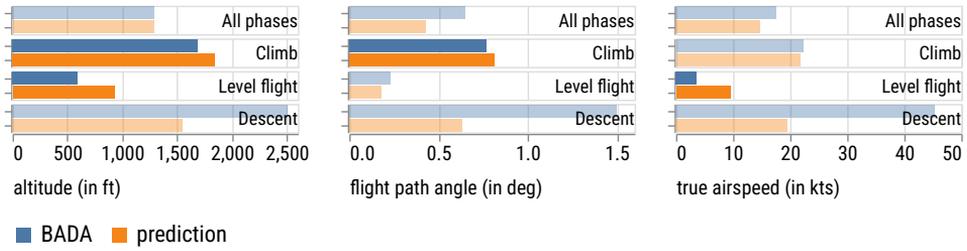
Depending on the identified flight attitude (climb, level, or descent) and the difference between commanded and actual speeds, the appropriate TCL routine was selected: `constantSpeedLevel` for cruise, `constantSpeedRating_time` for climb or descent at constant speed, `accDec_time` when speed adjustments were required, and `constantSpeedROCD_time` when a vertical speed target had been engaged. At each integration step, the most recent simulated state of altitude, airspeed, rate of climb or descent, and mass was fed back into the subsequent computation, thereby ensuring a continuous and dynamically consistent simulation of the trajectory. Since aircraft mass is not directly available from ADS-B or Mode S data, the BADA default assumption of 85% of the Maximum Take-Off Weight (MTOW) was applied at the first available point of the recorded trajectory (depending on ADS-B coverage).

Figure 2 presents a quantitative comparison of mean absolute errors (MAE) between BADA and the proposed neural model for four representative aircraft types (A319, A359, and B738, E190), evaluated across flight phases (all phases, climb, level flight, and descent) and variables (altitude, flight path angle, and true airspeed). Specifically, evaluation is performed on full trajectories generated autoregressively from the initial state ( $t = 0$  to  $T$ ), ensuring that the metrics capture the long-term accumulation of errors. Overall, the proposed model achieves slightly lower errors than BADA for

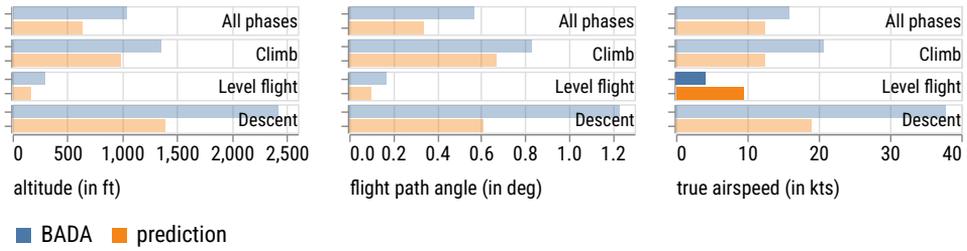
### A319 performance model comparison



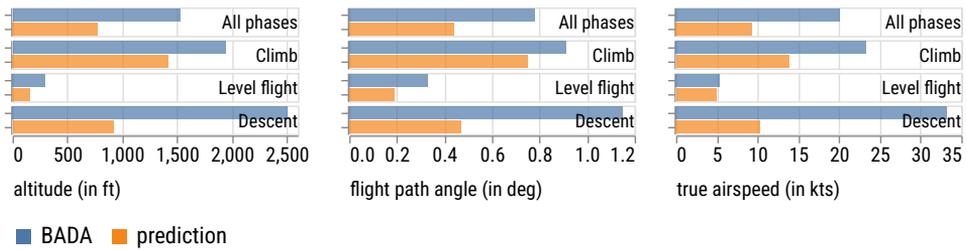
### A359 performance model comparison



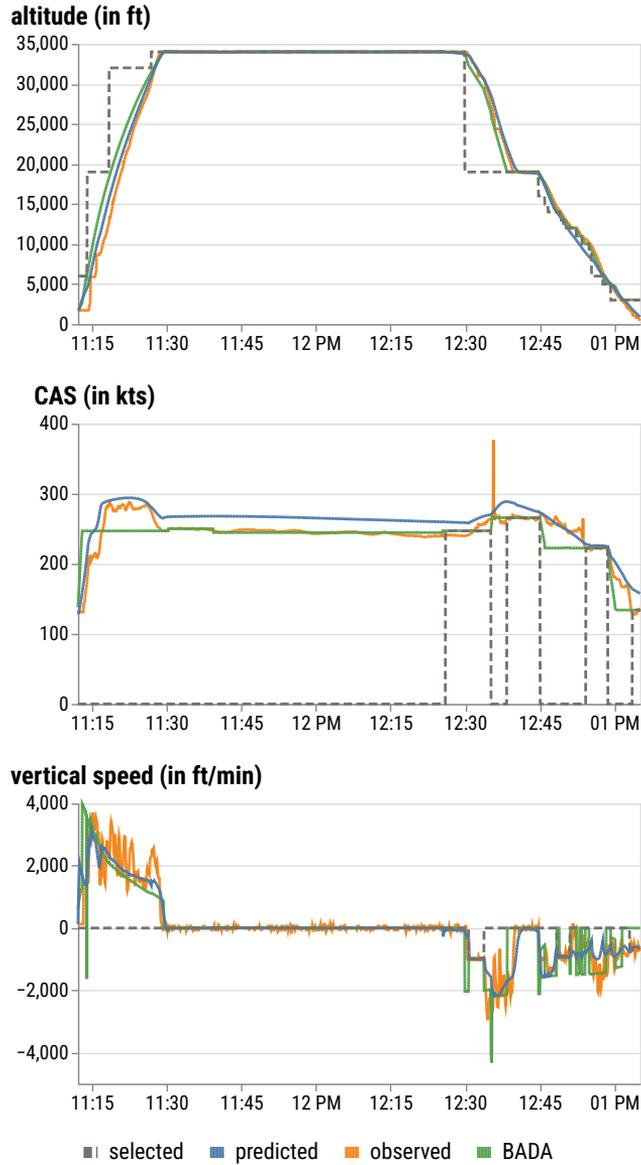
### B738 performance model comparison



### E190 performance model comparison



**Figure 2.** Comparison of mean absolute errors (MAE) between BADA and the proposed model for four aircraft types (A319, A359, B738, E190). Results are reported for altitude, flight path angle, and true airspeed (TAS), averaged over all phases, climb, level flight, and descent. Instances where BADA achieves lower errors are highlighted with full opacity.



**Figure 3.** Example of trajectory reproduction from A319 ADS-B data. The figure compares the recorded ADS-B trajectory (orange) with the model prediction (blue) and the BADA-based benchmark (green) across altitude, CAS and vertical speed. The selected altitude, CAS and vertical speed is shown as a red dashed-line. Time is represented on the x-axis in seconds (one point every 4 s).

most metrics and flight phases. For instance, for the A319, the MAE in altitude over all phases decreases from approximately 1,650 ft with BADA to 872 ft with the proposed approach, while the error in flight path angle is reduced from  $0.75^\circ$  to  $0.40^\circ$ . A similar trend is observed for the A359, confirming the improved capability of the neural model to reproduce vertical dynamics across aircraft categories.

However, during level flight, BADA performs slightly better. For example, for the A319, the MAE in true airspeed is about 3.8 kt for BADA compared to 14.0 kt for the proposed model, and similar patterns are found for the A359, B738 and E190. This phase corresponds to long, steady cruise segments, often at constant Mach, where the neural model exhibits small drifts in maintaining target speeds and perfectly flat altitude segments. These local deviations, while minor in magnitude, highlight the inherent difficulty for data-driven models to reproduce stable control regimes governed by subtle autopilot dynamics. The general trends observed in the quantitative analysis are further illustrated by the example trajectory of an A319 shown in Figure 3.

## 6. Discussion and limitations

The present study demonstrates that Neural ODE models can learn physically consistent vertical aircraft dynamics directly from open surveillance data. While the results confirm the feasibility and potential of such an approach, several limitations must be acknowledged regarding the scope of the model, the data quality, and the methodological assumptions.

### 6.1 Interpretation of Benchmarking Results

It is important to emphasize that BADA is a well-established and physically consistent performance model, widely used in both operational and simulation contexts. Its comparatively higher errors in this benchmark do not indicate a lack of validity, but rather reflect methodological differences. In this analysis, the evaluation concerns the entire BADA-based trajectory generation pipeline, coupling the BADA 4 performance model with trajectory control routines driven by operational inputs, rather than the performance equations alone. BADA is not designed to replicate individual recorded flights with noisy control input, but to provide a reliable and generalisable description of aircraft performance across a wide envelope of flight conditions. In contrast, the proposed architecture is directly trained on the ADS-B data used for evaluation, which naturally gives it an advantage in reproducing the observed flight dynamics.

The comparison also reflects differences in mass modelling assumptions. A default reference mass corresponding to  $0.85 \times \text{MTOW}$  was used for BADA, while the neural model might have learned a latent representation of aircraft mass as a function of city-pair distance and other contextual features. This additional flexibility likely contributes to its improved performance in reproducing climb and descent profiles across different aircraft and operational contexts.

### 6.2 Limitations

A first limitation is that the current model is restricted to the longitudinal dynamics of the trajectory, i.e., the evolution of altitude and speed along the aircraft's longitudinal axis, expressed through true airspeed. In this formulation, the aircraft is effectively represented in a body-fixed reference frame aligned with its nose, so only motion in the vertical plane is captured. Consequently, lateral behaviour, including variations in track angle, banking, and turn dynamics, is not represented. The framework therefore cannot yet support fully four-dimensional trajectory prediction, nor operational applications in which lateral manoeuvres and route choices play a decisive role. Extending the model to incorporate lateral dynamics represents a natural direction for future research.

The focus on vertical-longitudinal dynamics was a deliberate choice, as this axis encompasses the

most critical unobservable physical dependencies. In particular, modelling the energy share factor, how an aircraft partitions its surplus power between climbing and accelerating, represents a significantly higher challenge for data-driven models than lateral kinematics. While lateral motion involves specific hurdles such as turn detection and the modelling of orthodromic navigation patterns, the vertical axis remains the primary bottleneck for physically-consistent trajectory reconstruction from surveillance data.

A second limitation lies in the observability and reconstruction of control inputs. Many key parameters, such as the selected speed or vertical rate, are not directly available from surveillance data and must be inferred from the temporal evolution of Mode S Enhanced Surveillance (EHS) signals through plateau detection algorithms. While these heuristics enable the reconstruction of plausible targets, they remain sensitive to noise, resolution, and local turbulence, potentially introducing uncertainty in the estimated control variables. Furthermore, the coverage of EHS messages is uneven across regions and dependent on secondary surveillance radar (SSR) configurations, which may bias the dataset toward well-instrumented airspaces such as Europe and North America.

A third limitation concerns the quality of the *selected altitude* variable derived from Mode S EHS reports (BDS 4,0). As noted in the results analysis, this parameter is often affected by decoding artefacts, inconsistencies, and spurious step changes that propagate into evaluation metrics and can distort apparent accuracy in altitude reproduction. Part of the discrepancy in BADA's tracking performance originates from this imperfect reference. In principle, this issue could be mitigated by re-estimating selected altitudes from trajectory patterns, following the same logic as for speed targets. However, the objective of this work was primarily to assess the feasibility of learning vertical dynamics from open surveillance data, rather than to refine individual control parameters. Future developments will aim to integrate more robust methods for reconstructing and validating Mode S variables, improving the reliability of altitude-related evaluations.

It is worth noting that a similar Neural ODE methodology was previously applied to high-fidelity Quick Access Recorder (QAR) data [5]. QAR data provides significantly higher temporal resolution (4–25 Hz) and direct access to propulsion parameters such as engine thrust settings (N1), fuel flow, and aircraft mass, as well as precise autopilot targets recorded by the flight management system. In contrast, the present study relies on reconstructed control intent from Mode S EHS signals, which are subject to the noise and decoding artefacts discussed earlier. As a result, the QAR-based approach benefits from cleaner inputs and richer physical observability, enabling more accurate performance estimation and fuel consumption modelling. Despite these differences, the present ADS-B-based framework achieves promising vertical trajectory reconstruction, suggesting that the continuous-time formulation and embedded kinematic constraints provide sufficient regularization to compensate for input uncertainty. The main trade-off lies in scalability versus interpretability: ADS-B enables global, open-access trajectory modelling, while QAR supports detailed performance analysis but requires proprietary airline data.

From a modelling perspective, the proposed architecture is only partially physics-informed. Beyond the analytical kinematic relations embedded in the trajectory layer, the Neural ODE component is not explicitly constrained by aerodynamic or performance equations. While this formulation ensures smooth and continuous-time dynamics, it does not guarantee adherence to physical limits, such as thrust or drag bounds. Although no unrealistic behaviours were observed in the experiments, such effects cannot be entirely ruled out when extrapolating outside the training distribution or under atypical control conditions. Introducing stronger physics-based regularisation, such as energy-balance constraints or aerodynamic envelopes, would enhance generalisation and robustness. In particular, penalizing energy changes during level-flight segments could help reduce the small drifts observed in cruise (Section 5), forcing the model to better respect steady-state equilibrium conditions.

Another structural limitation is that the model learns relative motion patterns but cannot infer absolute performance quantities. Since aircraft mass, thrust, and fuel flow are unobservable in open datasets, the model does not capture the underlying energy or propulsion balance governing climb performance. Consequently, while the generated trajectories are realistic in shape, they cannot directly support quantitative assessments of fuel consumption or emissions at this stage. Combining open data with partial physical calibration or probabilistic mass estimation could help bridge this gap.

Overall, the proposed framework demonstrates that continuous-time aircraft dynamics can be learned from open surveillance data while preserving physical consistency through embedded kinematic relations. Nonetheless, the above limitations underline that this approach remains a research prototype rather than an operational model. Addressing these challenges, through improved observability, stronger physical constraints, uncertainty quantification, and domain adaptation, will be key to advancing the use of open data for reliable, interpretable, and reproducible trajectory modelling.

## 7. Conclusion

This paper presented a physics-guided neural framework for generating vertical aircraft profiles from open surveillance data using Neural Ordinary Differential Equations (Neural ODEs). The proposed approach combines analytical kinematic relations with a learnable continuous-time dynamical model trained on ADS-B and Mode S Enhanced Surveillance (EHS) data retrieved from the OpenSky Network. Despite the inherent limitations of open data, the model shows promising performance for altitude reconstruction, while further work is required to better capture speed profiles.

Evaluation against BADA-based benchmarks showed that the Neural ODE model reproduced vertical dynamics with overall good absolute errors across most flight phases, particularly during climb and descent. However, the comparison with BADA was partly blurred by the limited quality of several Mode S-derived control variables, notably the *selected altitude* and the estimated *speed* and *Mach* targets. These parameters, reconstructed from noisy and irregular signals, introduce uncertainty in both training and evaluation, making precise phase alignment and control-law interpretation challenging. Despite these data imperfections, the Neural ODE framework exhibited notable robustness to noisy or incomplete inputs, maintaining stable predictions and smooth state evolution. This resilience suggests that continuous-time learning, coupled with embedded physical relations, can help absorb inconsistencies in open surveillance data.

Future work will focus on improving the reconstruction and filtering of Mode S-derived features to reduce uncertainty in control-related variables. Better estimates of selected targets, applied consistently in both training and evaluation, should sharpen the comparison with reference performance models such as BADA. Beyond this, extending the model toward lateral dynamics and incorporating stronger physical constraints, for instance through energy-balance or aerodynamic regularisation, will be key to achieving higher realism and generalisation.

## Author contributions

Conceptualization (G.J), Methodology (G.J), Software (G.J), Validation (G.J), Formal analysis (*all*), Investigation (*all*), Data Curation (*all*), Writing – Original Draft (*all*), Writing – Review & Editing (*all*), Visualization (X.O), Project administration (X.O), Funding acquisition (X.O)

## Funding statement

The authors are grateful to the EC for supporting the present work, performed within the NEEDED project, funded by the European Union's Horizon Europe research and innovation programme under

grant agreement no. 101095754 (NEEDED). This publication solely reflects the authors' view and neither the European Union, nor the funding Agency can be held responsible for the information it contains.

## Open data statement

All data used in this study are openly available from the OpenSky Network databases. The dataset can be reproduced using the provided download and preprocessing scripts, which automatically retrieve ADS-B and Mode S Enhanced Surveillance (EHS) data through the OpenSky API.

## Reproducibility statement

All code used to download and preprocess the data, train the models, perform trajectory inference, and generate the figures presented in this paper is available at: <https://github.com/eurocontrol-asu/node-fdm/>. The authors welcome collaboration and can provide guidance upon reasonable request.

The baseline trajectory prediction routines rely on the pyBADA library, available at: <https://github.com/eurocontrol-bada/pybada>. Please note that the BADA 4.2 performance coefficients are distributed by EUROCONTROL under license and must be requested separately.

## References

- [1] Angela Nuic, Damir Poles, and Vincent Mouillet. "BADA: An advanced aircraft performance model for present and future ATM systems". In: *International journal of adaptive control and signal processing* 24.10 (2010). DOI: 10.1002/acs.1176.
- [2] Angela Nuic and Vincent Mouillet. "User Manual for the Base of Aircraft Data (BADA) Revision 4". In: *EUROCONTROL Experimental Centre, EEC Technical/Scientific Report No. 12/11/22-58 v1.3* (2016).
- [3] Junzi Sun, Jacco M Hoekstra, and Joost Ellerbroek. "OpenAP: An open-source aircraft performance model for air transportation studies and simulations". In: *Aerospace* 7.8 (2020). DOI: 10.3390/aerospace7080104.
- [4] Junzi Sun. "OpenAP.top: Open flight trajectory optimization for air transport and sustainability research". In: *Aerospace* 9.7 (2022). DOI: 10.3390/aerospace9070383.
- [5] Gabriel Jarry, Ramon Dalmau, Xavier Olive, and Philippe Very. *A Neural ODE Approach to Aircraft Flight Dynamics Modelling*. Proceedings of the SESAR Innovation Days 2025. 2025. arXiv: 2509.23307 [cs.LG].
- [6] Matthias Schäfer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, and Matthias Wilhelm. "Bringing up OpenSky: A Large-Scale ADS-B Sensor Network for Research". In: *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. 2014. DOI: 10.1109/ipsn.2014.6846743.
- [7] Daniel Delahaye, Stéphane Puechmorel, Panagiotis Tsiotras, and Eric Féron. "Mathematical models for aircraft trajectory design: A survey". In: *Air Traffic Management and Systems: Selected Papers of the 3rd ENRI International Workshop on ATM/CNS*. 2014. DOI: 10.1007/978-4-431-54475-3\_12.
- [8] Dia Poll and Ulrich Schumann. "An estimation method for the fuel burn and other performance characteristics of civil transport aircraft in the cruise. Part 1 Fundamental quantities and governing relations for a general atmosphere". In: *The Aeronautical Journal* 125.1284 (2021). DOI: 10.1017/aer.2020.62.
- [9] Samet Ayhan and Hanan Samet. "Aircraft trajectory prediction made easy with predictive analytics". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016. DOI: 10.1145/2939672.2939694.

- [10] Yulin Liu and Mark Hansen. *Predicting Aircraft Trajectories: A Deep Generative Convolutional Recurrent Neural Networks Approach*. 2018. arXiv: 1812.11670 [cs.LG].
- [11] Yutian Pang, Houpu Yao, Jueming Hu, and Yongming Liu. “A recurrent neural network approach for aircraft trajectory prediction with weather features from sherlock”. In: *Proceedings of the AIAA Aviation 2019 Forum*. 2019. doi: 10.2514/6.2019-3413.
- [12] “A hybrid CNN-LSTM model for aircraft 4D trajectory prediction”. In: *IEEE access* 8 (2020). doi: 10.1109/ACCESS.2020.3010963.
- [13] Junfeng Zhang, Jie Liu, Rong Hu, and Haibo Zhu. “Online four dimensional trajectory prediction method based on aircraft intent updating”. In: *Aerospace Science and Technology* 77 (2018). doi: 10.1016/j.ast.2018.03.037.
- [14] Xiping Wu, Hongyu Yang, Hu Chen, Qinzhi Hu, and Haoliang Hu. “Long-term 4D trajectory prediction using generative adversarial networks”. In: *Transportation Research Part C: Emerging Technologies* 136 (2022). doi: 10.1016/j.trc.2022.103554.
- [15] Xinyu Chen, Jiajie Xu, Rui Zhou, Wei Chen, Junhua Fang, and Chengfei Liu. “TrajVAE: A variational autoencoder model for trajectory generation”. In: *Neurocomputing* 428 (2021). doi: 10.1016/j.neucom.2020.03.120.
- [16] Jimeng Rao, Song Gao, Yuhao Kang, and Qunying Huang. *LSTM-TrajGAN: A Deep Learning Approach to Trajectory Privacy Protection*. 2020. arXiv: 2006.10521 [cs.LG].
- [17] Seongjin Choi, Jiwon Kim, and Hwasoo Yeo. “TrajGAIL: Generating urban vehicle trajectories using generative adversarial imitation learning”. In: *Transportation Research Part C: Emerging Technologies* 128 (2021). doi: 10.1016/j.trc.2021.103091.
- [18] Gabriel Jarry, Ramon Dalmau, Philippe Very, and Junzi Sun. “On the generalization properties of deep learning for aircraft fuel flow estimation models”. In: *Transportation Research Part C: Emerging Technologies* 176 (2025). doi: 10.1016/j.trc.2025.105143.
- [19] Richard Alligier and David Gianazza. “Learning Aircraft Operational Factors to Improve Aircraft Climb Prediction: A Large Scale Multi-Airport Study”. In: *Transportation Research Part C: Emerging Technologies* 96 (2018). doi: 10.1016/j.trc.2018.08.012.
- [20] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. “Neural Ordinary Differential Equations”. In: *Proceedings of the 32nd Conference on Neural Information Processing Systems*. Montréal, Canada, 2018. arXiv: 1806.07366 [cs.LG].
- [21] Eldad Haber and Lars Ruthotto. “Stable architectures for deep neural networks”. In: *Inverse problems* 34.1 (2017). doi: 10.1088/1361-6420/aa9a90.
- [22] Jinyi Ma, Yiyang Li, Jingqi Tu, Yiming Zhang, Jianliang Ai, and Yiqun Dong. “Development and Implementation of Physics-Informed Neural ODE for Dynamics Modeling of a Fixed-Wing Aircraft Under Icing/Fault”. In: *Guidance, Navigation and Control* 4.01 (2024). doi: 10.1142/S2737480723500243.
- [23] Shaoning Lu and Yue Qian. “Enhanced trajectory forecasting for hypersonic glide vehicle via physics-embedded neural ODE”. In: *Drones* 8.8 (2024). doi: 10.3390/drones8080377.
- [24] Jongho Park and Mingu Kim. “Data-Driven Regulator of Rotary-Wing UAV Utilizing Neural Ordinary Differential Equations”. In: *International Journal of Aeronautical and Space Sciences* (2025). doi: 10.1007/s42405-025-01007-4.
- [25] Lars Ruthotto and Eldad Haber. “Deep neural networks motivated by partial differential equations”. In: *Journal of Mathematical Imaging and Vision* 62.3 (2020). doi: 10.1007/s10851-019-00903-1.
- [26] Timothé Krauth, Jérôme Morio, Xavier Olive, Benoit Figuet, and Raphael Monstein. “Synthetic Aircraft Trajectories Generated with Multivariate Density Models”. In: *Proceedings of the 9th OpenSky Symposium*. Brussels, Belgium, 2021. doi: 10.3390/engproc2021013007.
- [27] Timothé Krauth, Adrien Lafage, Jérôme Morio, Xavier Olive, and Manuel Waltert. “Deep Generative Modelling of Aircraft Trajectories in Terminal Maneuvering Areas”. In: *Machine Learning with Applications* 11 (2022). doi: 10.1016/j.mlwa.2022.100446.

- [28] Xavier Olive, Junzi Sun, Mayara Conde Rocha Murça, and Timothé Krauth. “A Framework to Evaluate Aircraft Trajectory Generation Methods”. In: *Proceedings of the 14th USA/Europe Air Traffic Management Research and Development Seminar*. 2021.
- [29] Gabriel Jarry, Almoctar Hassoumi, Daniel Delahaye, and Christophe Hurter. “Interactive trajectory modification and generation with FPCA”. In: *CEAS Aeronautical Journal* 13.2 (2022). doi: 10.1007/s13272-022-00577-3.
- [30] Gabriel Jarry, Nicolas Couellan, and Daniel Delahaye. “On the use of generative adversarial networks for aircraft trajectory generation and atypical approach detection”. In: *Proceedings of the ENRI International Workshop on ATM/CNS*. 2019. doi: 10.1007/978-981-33-4669-7\_13.
- [31] Xuhao Gui, Junfeng Zhang, Xinmin Tang, Daniel Delahaye, and Jie Bao. “A Novel Aircraft Trajectory Generation Method Embedded with Data Mining”. In: *Aerospace* 11.8 (2024). doi: 10.3390/aerospace11080648.
- [32] Junzi Sun, Huy Vũ, Joost Ellerbroek, and Jacco Hoekstra. “pyModeS: Decoding Mode S Surveillance Data for Open Air Transportation Research”. In: *IEEE Transactions on Intelligent Transportation Systems* (2019). doi: 10.1109/TITS.2019.2914770.
- [33] Xavier Olive, Jan Krummer, Benoit Figuet, and Richard Alligier. “Filtering Techniques for ADS-B Trajectory Preprocessing”. In: *Journal of Open Aviation Science* 2.2 (2025). doi: 10.59490/joas.2024.7882.
- [34] Hans Hersbach et al. “The ERA5 global reanalysis”. In: *Quarterly Journal of the Royal Meteorological Society* 146.730 (2020), pp. 1999–2049. doi: 10.1002/qj.3803.
- [35] Junzi Sun and Esther Roosenbrand. “Fast contrail estimation with OpenSky data”. In: *Journal of Open Aviation Science* 1.2 (Nov. 2023). doi: 10.59490/joas.2023.7264.
- [36] Jonghun Lee, YongKyung Oh, Sungil Kim, and Dongyoung Lim. *Neural Regenerative Stochastic Differential Equation: Dropout Scheme for Neural Differential Equations*. 2025. URL: <https://openreview.net/forum?id=Mq23uJ6sIm>.