Fourteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2021)

A Framework to Evaluate Aircraft Trajectory Generation Methods

Xavier Olive*, Junzi Sun†, Mayara Condé Rocha Murça‡, Timothé Krauth*§

*ONERA – DTIS
Université de Toulouse
Toulouse, France

†Faculty of Aerospace Engineering,
Delft University of Technology,
Delft, the Netherlands

‡Division of Civil Engineering
Aeronautics Institute of Technology
São José dos Campos, São Paulo, Brazil

§Centre for Aviation, School of Engineering
Zurich University of Applied Sciences
Winterthur, Switzerland

***Abstract*—Aircraft trajectory generation is a widely addressed problem with applications including emergency trajectory generation, collision risk models, air traffic flow and capacity management or airspace design. State of the art methods to generate individual trajectories and optimise some performance or emergency criterion may lack of realism with respect to common situations implemented by air traffic controllers. On the other hand, statistical data-driven methods to generate aircraft trajectories excel at imitating operational practice but may be difficult to implement even in simulations due to aircraft performance limitations. This contribution proposes a common baseline to compare literature and bleeding-edge methods to generate air traffic trajectories. Keeping in mind that the most appropriate criterion should always depend on the targeted application, we present here an extensive set of metrics to evaluate the quality of generated trajectories, before assessing two generation methods in light of these indicators.**

***Keywords*—air traffic management, trajectory generation, simulation, evaluation, statistical methods**

I. Introduction

Aircraft trajectory generation is the problem which consists in creating a full set of timestamped coordinates for an aircraft flying through a given airspace. Such generated flight paths should reflect the physical behaviour of a potential aircraft (e.g. airplane, helicopter, UAV) and statistical properties of the spatio-temporal bounds it evolves in. When the generation process is aimed to be computed online, while the aircraft is flying, we talk about trajectory prediction.

Applications for such techniques are many: trajectory generation allows for simulation of the traffic flows and Air Traffic Control (ATC) workload to support airspace design processes [1], [2]; it allows for prediction of aircraft trajectories and resulting demand in airport/airspace resources to assist air traffic flow and capacity management at strategic and tactical levels, especially in future Trajectory-Based Operations environment [3]; it can help in the design of emergency procedures in future Flight Management Systems for single pilot operations and in collision risk modelling [4], which raises additional challenges, including the need to sample *outlying* trajectories with characteristics of situations leading to a risk increase.

Historically, researchers have been tackling the problem of trajectory generation from a guidance and control approach, as a dynamic programming [5] or online optimisation [6], [7] problem, sometimes with variants to tackle weather uncertainties [8]. Such physical and dynamical models perform well within such a scope but do not usually consider operational real-time constraints, interaction with ATC or human factors. On the other hand, data-driven statistical approaches aim at imitating operational situations, examples ranging from Generative Adversarial Networks (GAN) [9] to Gaussian Mixture Models (GMM) [10]. Despite some kind of weak statistical guarantee that such generated trajectories are reasonable as they are designed to be close to sample trajectories which have already been flown before, no physical model like BADA [11] or OpenAP [12] is integrated and nothing constrains such generated trajectories to be flyable.

Best suited approaches obviously depend on the intended application; however, the literature reveals that in spite of convincing results on specific applications, a set of common metrics to objectively compare the performance of such methods in terms of physical soundness and statistical realism is currently lacking: model-driven methods would score high on physical aspects, while data-driven methods would favour statistical relevance over the law of physics. While the evaluation of trajectory prediction methods is natural—compare the predicted and the actual trajectory—, and some simple applications may settle for the distance between a generated trajectory and existing ones, it may be hard to converge to a reasonable set of metrics for a wide range of applications, including those where generated trajectories are expected to follow an atypical pattern.

After a literature review in Section II, we present in Section III a set of baseline metrics to evaluate the performance of generation methods. Further in Section IV, we consider two data-based generation methods with respect to the proposed framework before initiating a discussion and concluding in Sections V and VI about the implications of our approach.

II. Literature Review

Trajectory generation and prediction have been extensively studied by the scientific community over the past decades. Applications are many, and the motivation behind each contribution results in fundamentally different methods focusing on flight trajectory creation.

Trajectory generation embodies several different meanings. The most common one refers to the generation of an optimal flight route, be it time or fuel efficient. Some focus on the creation of a whole trajectory: in [5], the authors applied their method to a Terrain Following Problem, whereas in [6], they calculate a whole commercial flight trajectory which minimises the fuel consumption given a set of path constraints.

Trajectory generation can also be computed online, in an attempt to prevent hijacking [7], to avoid hazardous weather [8] (rerouting), or even to find an emergency path to the closest most appropriate airport given the nature of the failure [13]. The latter also gives a method to obtain a configuration reducing airspace congestion, minimizing cost and respecting connection plans. Trajectory prediction for anticipation of potential conflicts or delays is another common application. Among many available methods, Shi et al. [14] focus on weather or ATC uncertainties, while Wang et al. study Unmanned Aerial Systems (drones) in restricted air-spaces for collision avoidance [15]. Trajectory prediction heavily relies on aircraft performances: BADA [11] and OpenAP [12] provide an extensive way of modelling aircraft performances for all flight phases of a given trajectory.

Further, collision risk models are another common application of trajectory generation methods: Blom et al. [16] present stochastic models for aircraft trajectories in order to deduce a collision risk model. Eckstein [17] use trajectory generation to oversample its database and estimate collision risks with Monte-Carlo methods. These frameworks may benefit from further contributions [1], [10], [18] focusing on the generation of artificial trajectories aiming to be "reasonable" compared to existing ones.

The best fitted trajectory generation method can greatly differ from an application to another. We can however classify them into two main categories: *model-driven generation methods* and *data-driven generation methods*: both approaches focus on very different aspects, so the final choice heavily rely on the final purpose.

A. Model-driven trajectory generation

Model-driven trajectory generation is the classical approach, based on kinematics and aerodynamics models. The generation process is mostly based on the optimisation of a given criterion, usually, the fuel consumption. Waller et al. [5] solve this problem through dynamic programming: they look for an optimal path in a large-scale discrete solution space. Soler et al. [6] minimize a cost function under constraints, designed to ensure that the solution found actually follows physical kinematics equations. Additional constraints may help design trajectories to avoid buildings [7], prohibited areas [13], or evolving bad weather [8].

The other main approach is based on a physical or statistical modelling of the different characteristics of an airplane. The most famous aircraft performance models include BADA [11] and OpenAP [12]. Such methods are deterministic and provide accurate flyable trajectories but do not take into account general uncertainties like ATC manoeuvres, human factors, or unexpected weather.

A common way to address uncertainty is to introduce randomness to the flight mechanics equations so as to model a trajectory by a stochastic process in continuous time, leading to some stochastic differential equations [16]. An other approach uses Monte-Carlo simulations to estimate the probabilistic distribution of positions using simple kinematic equations [15]; however, such simplifications are made in the physical equations that may no longer be relevant for 3D trajectories.

B. Data-driven trajectory generation

Data-driven trajectory generation is currently booming with the hype around data science and artificial intelligence. ML-based methods predict or generate trajectories using only past observations. No simplifications in the physical models are required and uncertainties are easier to take into account.

Again, variants are many, and heavily depend on the task to achieve. Jacquemart et al. [4] use Markov chain processes to introduce randomness to a 2D straight flight path with constant speed, before using importance splitting to estimate conflict probabilities. Poppe et al. [1] target the generation of nominal behaviours for a flight route and focus on clustering, with a K-means algorithm to define clusters of 1D climb profiles. Centroids of each cluster are then considered as the mainstream trajectory of the route. Murça et al. [10] initiate their approach with a clustering analysis to identify a set of trajectory patterns, which are then modeled with a probability distribution function. Sampling trajectories from the probabilistic model generates trajectories looking very familiar to ATC in charge of the airspace.

Dimensionality reduction is an additional tool available to improve the sampling of statistically coherent data. For instance, Eckstein et al. [17] project trajectories on dimensions with the most variance with a Principal Component Analysis (PCA). The trajectory generation can be performed in the latent space, before serving as input to the inverse linear operation. Jarry et al. extend the process to functional PCA, considering trajectories as continuous objects [18]. Generative Adversarial Networks [9] are one of the most common non-linear alternative to f-PCA. However, as such approaches based on the sampling from fitted probability distributions excel at generating new trajectories similar to the most commonly encountered ones, they can fail to generate emergency trajectories or situations falling in the "outlier" category.

All metrics will rely on definitions of trajectory distances which are the sinews of war. The most reviewed definitions of distances [19], [20] consider trajectories as samples in $\mathbf{R}^n$ and compare trajectories point by point. As an attempt to address the *curse of dimensionality* causing data samples to become very sparse in high dimensional space, hence making it very difficult to have two trajectories to become close to each other, the literature in trajectory clustering addresses this issue by finding original metrics [21] or by projecting trajectories to a lower dimension space prior to computing Euclidean distances [22].

III. Framework and metrics

A. Expectations about generated trajectories

Trajectory generation is a popular topic with many different applications ranging from flight management systems to deconfliction strategies or safety analyses. Expectations set on those generation methods highly depend on the application framework: therefore, the best suited method will depend on the question asked and on the definition of the optimisation criterion.

In most use cases, generated trajectories are intended to serve realistic simulations, but the focus for the realism depends on the application. In the following, we make a distinction between a *trajectory* which concerns the behaviour of a single aircraft, and a *scenario* which focuses on the interaction between two or more aircraft. Expectations for common trajectory generation use cases are:

- *Free flight and automatic deconfliction.* Dense air traffic scenarios with a large number of flights are generated. Usually, random flights generated at boundaries of the experiment airspace. In these simulations, flights do not follow predefined waypoints and often do not need to be based on historical flight data. The expectation is that flights need to be physically plausible (i.e., within the performance boundaries);
- *Human factor studies and ACTO training.* Here, very realistic trajectories are expected. A generated flight should be able to imitate real flights on the radar screen, for example, following predefined waypoints and flying with common speed and altitude restrictions. Scenarios should implement reasonable separations, have a realistic mix of different aircraft categories and may focus on less common situations (QFU reconfigurations, thunderstorms, etc.) for specialised training;
- *Flight safety analyses.* These studies aim at estimating the probabilities of rare events, incidents such as collision risks, runway excursions, etc. A common approach is to model each risk with fault trees, decision barriers aiming at mitigating the precursors leading to dangerous situations: e.g. go around if unstable approach or TCAS resolution advisory if aircraft separation is infringed. Such conditional probabilities may be estimated with Monte-Carlo simulations but as precursors have a low probability of occurrence, a common workaround is to use importance sampling or subset sampling to favour the generation of risky situations and better estimate resulting probabilities. Those situations should be realistic but ensuring the realism is a difficult task: the deeper the condition in the fault tree, the less common the situation;
- *U-Space simulation.* Simulation of UAVs often involves different performance models than aircraft. Since these studies often deal with future urban U-Space scenarios, there is no historical data available. Hence, UAV flights are generated based on performance models and constrains of U-Space;
- *Emergency trajectories.* This topic is mostly about the generation of flyable trajectories under degraded capabilities: engine loss situations (the Hudson river case), support to pilot with reduced crew, etc. The generated trajectories aim at determining the most reasonable alternative airfield to reach, and to feed the generated trajectory to a Flight Management System (FMS). Past emergency situations, e.g., 7700 squawk codes in ADS-B large-scale databases [23]; may be considered for offline comparison, but no two such situations being similar, it is hard to leverage this knowledge for a most realistic trajectory generation method;
- *Air traffic flow and capacity management (ATFCM).* Trajectory generation can here be used to estimate the future workload in given airspaces and avoid placing unnecessary regulations [1]. Studies are mostly about the estimation of possible implemented ATC strategies (regulations, rerouting) and prediction of times of entry and exit in given airspaces. Aircraft performance is here a minor contributing factor: trajectory generation can be coarse grain; a good knowledge of operational practice has a higher impact on the quality of the models.

Eventually, expectations boil down to two main categories:

1) *operational realism*: generated trajectories and situations should look familiar to experts, pilots and air traffic controllers. Realism is not about real, past or usual patterns (safety analyses would be keen on situations identified as leading to a high probability for the studied risk), rather about experts who would not say, for a good reason, that "this situation would actually never happen";
2) *physical realism*: generated trajectories should be consistent with the performance of the aircraft. This criterion is however limited by the accuracy of the model, and by limitations of aircraft flying with degraded performances, e.g. limited climbing rate;

but all applications do not call for the same amount of realism on both aspects. Table I below attempts to summarize these considerations.

	operational realism	physical realism
free flight	n/a	average
human factor	**high**	average
ATCO training	**high**	average
safety analyses	**outliers**	average
UAV	n/a	**high**
emergency situations	n/a	**high**
ATFCM	**high**	low

TABLE I. Expectations for trajectory generation with respect to common use cases

B. A set of proposed metrics

In light of hereabove considerations, we propose in the following a set of metrics to address different needs of realism with respect to the application framework. These metrics attempt to reduce the performance of a generation method to a scalar, and are either operation or physics oriented.

a) Expert evaluation: this metric assesses the operational relevance of generated trajectories. A possible protocol is to ask air traffic controllers or pilots familiar with a given

airspace to determine whether a generated trajectory or situation has actually happened in real life. Candidate trajectories or situations would be randomly drawn from a large set containing real and generated samples: a natural metric would be the *f-score*, i.e. the harmonic mean of precision (how many selected items are relevant? how many detected trajectories were generated?) and recall (how many relevant items are selected? how many generated trajectories were identified?). If experts are fooled by generated trajectories, it is reasonable to claim the generation method performs well.

The weakness of this approach lies in the selection of experts: the metric is based on the evaluation of their performance at detecting simulated trajectories, and assumes all experts are reasonably skilled at this unusual selection task.

b) Similarity evaluation: this metric is particularly suitable when the question asked is whether a given situation has already been encountered in a reference dataset. The corresponding definition of this similarity score $s(t)$ would be the minimal distance d between the generated trajectory t and all samples t_r in the reference dataset:

$$s(t) = \min_{t_r \in \text{history}} d(t, t_r) \tag{1}$$

The difficulty here lies in the definition of the reference dataset which should contain as many possible patterns as possible for the score to be relevant. The risk is indeed that an unusual pattern, real or generated, scores low because it is unrepresented in the reference dataset. The definition for d can be chosen among distances defined in [19].

c) Statistical evaluation: this metric refers to the statistical properties of a set of generated trajectories or situations. Rather than computing a distance from an individual situation to a reference dataset, the comparison of the distribution of the generated set with the distribution of the real trajectories set should be the center of attention. The Kolmogorov–Smirnov test determines whether the generated samples and the reference data follow the same distribution but many other metrics are available for probability distributions. The Kullback–Leibler divergence tends to zero when the two distributions match; the Kolmogorov distance, the Total Variation distance, or the Wasserstein distance are valid alternatives, the latter being well known for its application in Generative Adversarial Networks (GAN).

The probability density functions may be defined on large dimension spaces with finely resampled trajectories or after projecting to a latent space of lower dimension. The final score will highly depend on the definition of the reference dataset, which determines the reference probability density function. When the topic revolves around safety studies and atypical risky situations are expected, the reference dataset may need to be formed of wisely selected subsamples so that the reference probability density function matches the expectations to be met by generated trajectories.

d) Physical evaluation: this metric tackles with the difficulty to generate trajectories expected to be reasonable from the point of view of the laws of physics with considerations of aircraft performances.

In this approach, we implement such a metric by measuring the distance between a given trajectory and its replayed

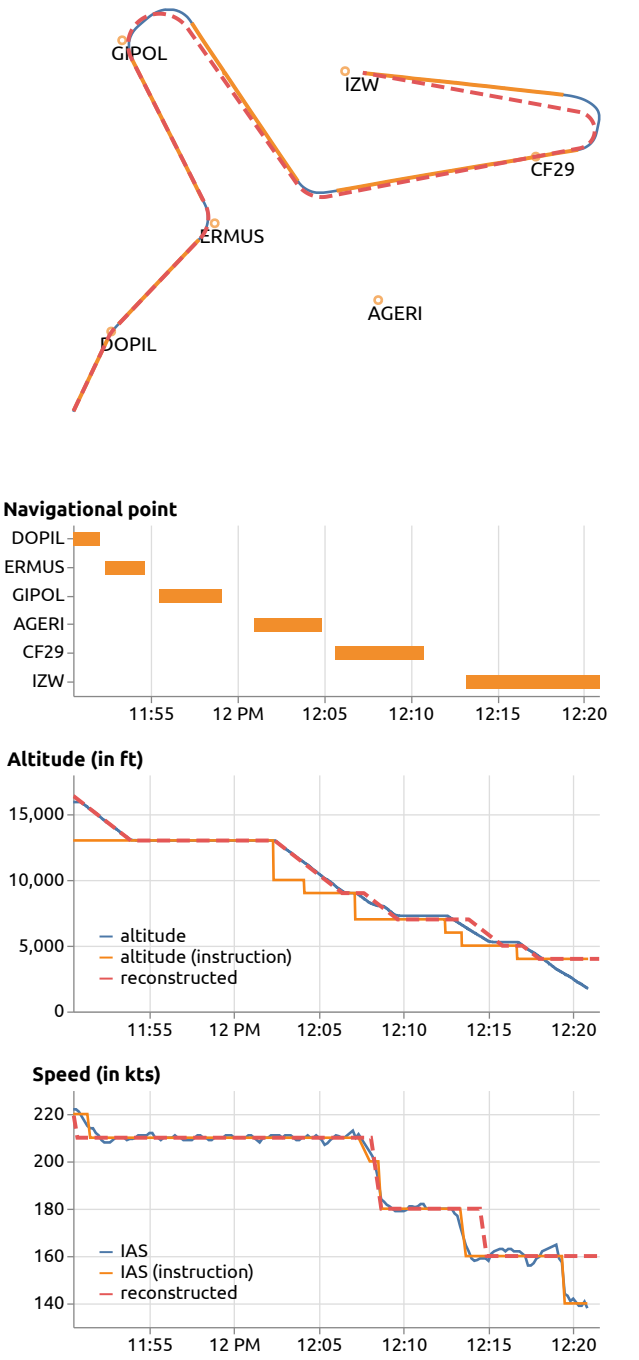

Figure 1: Trajectory of an aircraft landing at Zürich airport (`SWR563` on November 28th, 2019), with the corresponding navigational points, altitude and speed settings; and the reconstructed trajectory in dashed lines.

version in the open-source air traffic simulator BlueSky [24]. We decompose trajectories in segments aligned on navigational beacons defined in the considered geographical scope, with specific altitude settings in the FMS decoded from Enhanced Mode S information (*selected altitude*, BDS 4,0) and with constant indicated airspeed (*heading and speed report*, BDS 6,0) [25]. From these segments (see Figure 1), we generate corresponding ATC instructions leading to pilot actions, and compare each recomputed trajectory with the original one.

This approach relies on several hypotheses, especially on the assumption that the simulator and aircraft performance model are mature enough so that the distribution of distances between real trajectories and their reconstructed counterparts is much narrower than for generated trajectories. It also assumes that all the maneuvers implemented in the geographical scope are all efficiently modelled and reproducible in the simulator. When we evaluate the generation method in Section IV-A:

- we filter out any trajectory with holding patterns;
- we ignore the parts of trajectories after the last navigational point before aligning on the runway

as BlueSky implements neither automatic holding patterns nor ILS landing at the moment. These are not limitations of our approach, but of the chosen tool to reconstruct the trajectories.

C. Guidelines for assessing trajectory generation methods

The focal point when considering trajectory generation methods is the context of the application. A general evaluation of the whole process is possible, and we limit ourselves to such general remarks in next Section. However, the final decision to prefer a method over on another will only depend on the targeted application and will reflect the priorities in terms of expectations.

Section III-B presents a set of metrics that may prove useful when evaluating generation methods. Some of these metrics happen to be the founding stones of a wide range of common generation methods. Therefore, one had better try to prove that trajectories generated after a statistical analysis are actually flyable, rather than measuring how close they match the distribution they have been sampled from. Conversely, assessing the flyability of trajectories calculated by a real Flight Management System, knowing the physical parameter of the aircraft with the best precision, seems vain: evaluating how such a trajectory in an operational context, where the aircraft has to interact with many other ones, would make more sense. The rule of thumb is to favour metrics that do not serve as bases of the implementation, of which the generation methods are assessed.

Finally, whenever possible, implementing the same metrics on a large number of generated trajectories, as well as on a significant number of real trajectories matching the desired criterion is a good way to secure the analysis and to ensure the chosen metrics remains fully relevant when faced with real samples.

IV. Case studies

The methodology is tested with a dataset including a total of 19,480 trajectories landing at Zurich airport (`LSZH`) between October 1st and November 30rd 2019. We relied on The OpenSky Network [26] database to properly label trajectories landing at `LSZH`.

All trajectories have been requested and preprocessed with the help of the traffic Python [27] library which downloads OpenSky data, converts the data to structures wrapping pandas data frames and provides a specialised semantics for aircraft trajectories (e.g., intersection, resampling or filtering). In particular, it iterates over trajectories based on contiguous timestamps of data reports from a given `icao24` identifier: all trajectories are first assigned a unique identifier and resampled to one sample per second before the bearing of the initial point within 40 nautical miles (ASMA area) is computed.

The data presented in this paper is now provided as a generic import in the `traffic` library, triggering a download from a corresponding figshare repository [28] if the data is not in the cache directory of the user.

A. Generation of terminal airspace operations

a) Methodology: The first trajectory generation method was purely data-driven based on the methodology presented by Murça and Oliveira [10]. The approach starts with a trajectory clustering analysis to identify the major arrival trajectory patterns in the terminal area. For this, the flight trajectory data was resampled to generate trajectory vectors with the same number p of tracking observations (p was set to 200 to have a fine trajectory representation). The resampling approach linearly interpolates the spatial position for timestamps $t = \{0, \tau, 2\tau, ..., T\}$ evenly spaced throughout the flight duration T, where $\tau = \frac{T}{(p-1)}$. The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [29] was applied on the set of trajectory vectors, yielding twenty-six trajectory patterns. The trajectory patterns identified were then modeled in the form of a Gaussian Mixture Model (GMM) [30] to create a probabilistic trajectory model. The probability density function is given by a weighted sum of Gaussian densities, where each Gaussian component of the mixture models a particular trajectory pattern:

$$p(X) = \sum_{z=1}^{N} \pi_z \mathcal{N}(X; \mu_z, \Sigma_z) \tag{2}$$

where $X = \{\mathrm{lat}, \mathrm{lon}, \mathrm{alt}, \tau\}$ is the multivariate random variable that represents the multi-dimensional 4D trajectory, N is the number of Gaussian components in the mixture, μ_z and Σ_z are the mean vector and the covariance matrix of the Gaussian component that models the z^{th} trajectory pattern and π_z are the components weights in the mixture, which satisfy:

$$0 \leq \pi_z \leq 1 \tag{3}$$

$$\sum_{z=1}^{N} \pi_z = 1 \tag{4}$$

With the estimated parameters of the mixture $\theta = \{\pi_z, \mu_z, \Sigma_z\}$, trajectory generation is performed through sampling from the probabilistic model as follows:

1) Sample z from $Z \sim \text{Categorical}(\pi)$;
2) Sample x from $X/Z = z \sim \mathcal{N}(\mu_z, \Sigma_z)$

This method is strongly oriented towards operational realism: expert evaluation has been conducted in [10], and both similarity and statistical evaluations would make little sense here as the method is explicitly built to optimize these metrics.

b) Results: We present here a physical evaluation based on the replay of generated trajectories in the BlueSky simulator. We compare two different modelling approaches of the trajectories for the simulator:

1) based on navigational points: we detect alignments on official navigational points in the area (by comparing the track angle of the trajectory and the bearing to the navigational point, with a tolerance of one degree) and pilot the simulator with `DIRECT TO` instructions;
2) based on a line simplification with the Douglas–Peucker algorithm: we detect the most salient points in the trajectory and pilot the simulator with `DIRECT TO` instructions on those artificially defined way points.

We replayed two sets of trajectories:

1) the reference dataset of real trajectories (without self-intersections to avoid a well-known limitation of the BlueSky simulator);
2) a large number of trajectories (5,000) generated with this GMM-based approach (excluding trajectories from five self-intersection patterns identified, for the same reason above).

Not all trajectories replay with the same level of accuracy. In an attempt to quantify the replicability of trajectories in the simulator,

- we trim the original trajectory before it aligns on the ILS (to avoid another limitation of the BlueSky simulator);
- we trim the replayed or original trajectory so as to compare what was flown during the exact same duration;

then, we compute various trajectory distances between the two paths based on the `traj-dist` Python library[1], and plot the distributions of distances for all reconstructed trajectories.

The library implements 9 different metrics with slightly different distribution profiles: we present in Figure 2 only two of them, the Dynamic Time Warping (DTW) and the Symmetric Segment-Path Distance (SSPD). We choose here a plot of cumulative densities on the same graph, for a fair comparison between all approaches. We found that:

- trajectory modelling based on line simplification (dashed lines) tends to yield better reconstructions than when based on navigational points (solid lines). However, better reconstructed real trajectories score better when based on navigational points;
- artificially generated trajectories score very well with a line simplification modelling, suggesting that the GMM generates flyable trajectories, but that realism in terms of alignment on navigational points of STAR procedures is lacking for some trajectories.

[1]https://github.com/bguillouet/traj-dist

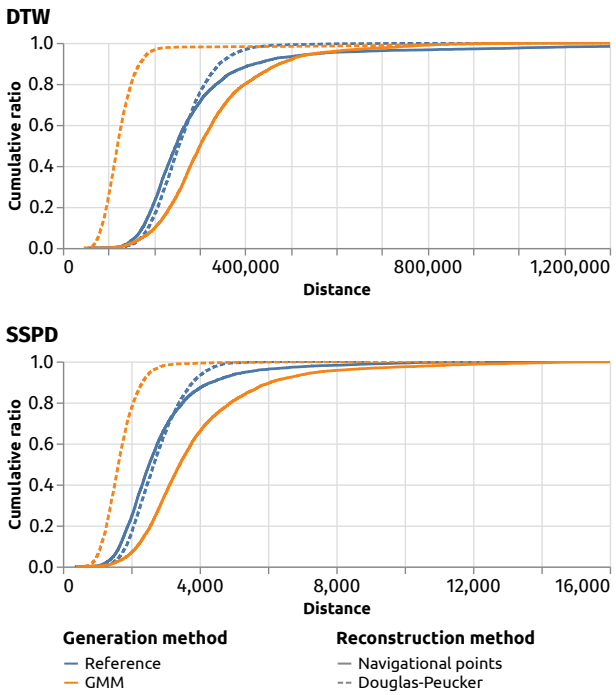

Figure 2: Cumulative distribution of the distances (DTW and SSPD metrics) between trajectories (real or generated) and their reconstruction through the BlueSky simulator using a navigational point or a line simplification (Douglas–Peucker) modelling. The faster the curves comes close to 1, the more trajectories are correctly reconstructed.

Figure 3 shows the the SSPD distance distributions by arrival trajectory pattern based on reconstructed trajectories with navigational points. It is interesting to note that the patterns naturally have different levels of difficulty in reconstruction. Moreover, generated trajectories that are more poorly aligned with navigational beacons are more present for some of the patterns.

Figure 4 focuses on poorly regenerated real and generated trajectories:

- (with the reference dataset) sequencing in the TMA involves more than alignment to navigational points: while alignments on `ERMUS`, `GIPOL` and `IKL` (ILS) seem operationally sound, the alignement on `ZH110` could be artificial. It also misses the tromboning action and leads to a higher distance between both trajectories;
- when based on line simplification, the BlueSky simulator faces a limitation about the maximum bank angle allowed; this parameter is not present in ADS-B and is hard to deal with in a generic manner;
- (with generated trajectories) matched navigational points seem particularly artificial and do not reflect the reality of the possible operations. This may be due to the many non straight segments forming the trajectory, a potential limitation with respect to operational realism. Line simplification yields a decent reconstruction, apart from the aforementioned bank angle issue.

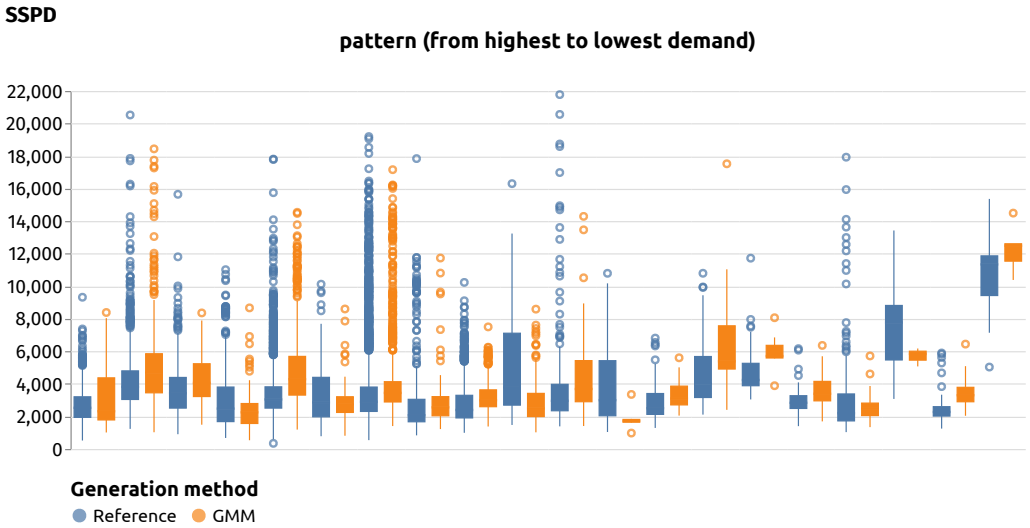

Figure 3: Boxplots of SSPD distances between trajectories (reference and generated) and their reconstruction through the BlueSky simulator using navigational points for each arrival trajectory pattern.

To wrap up, this analysis suggests that the proposed method, which is based on a statistical density representation of traffic incoming at Zurich airport:

- is able to generate feasible flyable trajectories with respect to the law of physics, despite being purely data-driven;
- generates trajectories with segments that can be more poorly aligned with navigational beacons from STAR procedures, especially for some particular patterns;
- is able to generate trajectories associated with the main patterns observed in the airspace, but fails to reproduce outlier trajectory behaviors harder to reconstruct that can be seen in practice (see the better scores than with real trajectories on the Douglas–Peucker line simplification approach).

Finally, the analysis is limited by the representation capabilities of the chosen simulator.

B. Generation of vertical profiles

a) Methodology: Vertical profile include altitude, vertical rated, and speed of the flight. Undoubtedly, the previously mention data-driven GMM can be used for the vertical profile generate. However, we want to make use of different type of data-driven approach to generate these flights for our analysis.

In [31], a parametric statistical model was introduced to construct the kinematic models, WRAP (also part of the OpenAP model), for the entire flight for common airliners. Here, we make use of the descent part of the WRAP model to generate vertical profiles for different aircraft types.

For descent flight, WRAP models key parameters such as constant Mach then constant CAS segments, crossover altitudes, and vertical speeds at different segments. All these parameters are generated from a large number of flights and modeled with parametric statistical models, which allows random sample to be drawn from a set of predefined distributions. By combining different key parameters independently drawn from these models, we can generate trajectories [12] that resemble descent flight in real operations.

b) Results: In this analysis, we constructed a dataset with 5,000 randomly generated vertical profiles, with a mix of aircraft types, descent ranges, and speed schedules. Similar to the approach of the previous case study, we use BlueSky to evaluate these generated trajectories. Most importantly, in order to independently evaluate this trajectories without involving parameters from OpenAP, BlueSky is configured with BADA performance model.

Figure 5 plots the vertical profile of a generated trajectory and of its reconstructed profile through BlueSky: both altitude profiles fit exceptionally well, which indicates that the vertical trajectory can be very well reconstructed by the BlueSky simulator. In terms of speed profile, a significant difference appears between the two trajectories. This can be explained by the different default deceleration value used in OpenAP ($-0.2\,\mathrm{m/s^2}$) compared to BlueSky ($-0.5\,\mathrm{m/s^2}$) during the segment after the constant CAS descent.

Since the vertical position consists of only one parameter (altitude), the metrics used for evaluation are much simpler. We present in Figure 6 three evaluation metrics, resp. the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Median Absolute Error (MedAE). These three metrics are calculated for each pair of generated and reconstructed trajectories.

In these plots, we can see that most of the generated trajectories have relatively small errors compared to reconstructed trajectories in the simulator. However, a small number of outliers still exist. Figure 7 shows the vertical profile for one of the outliers drawn from the dataset. The most striking difference lies in the divergence between the two trajectories at lower altitudes (below 10,000 ft). The generated trajectory has a higher than desired vertical rate, which is limited by the performance model in the simulator.

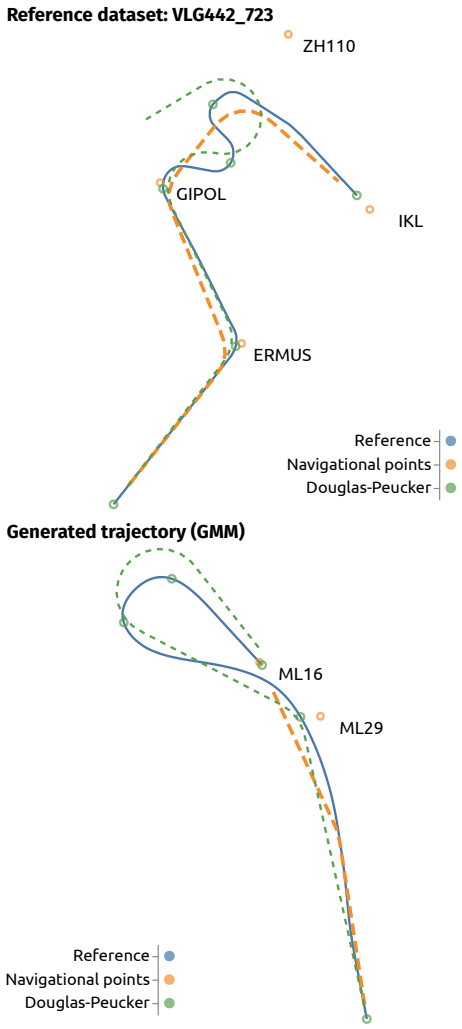

Figure 4: Examples of trajectories failing to reconstruct through the BlueSky simulator. In the reference dataset (real trajectories), tromboning instructions are hard to reconstruct; the maximum bank angle also becomes a limitation with the line simplification (Douglas–Peucker) modelling.

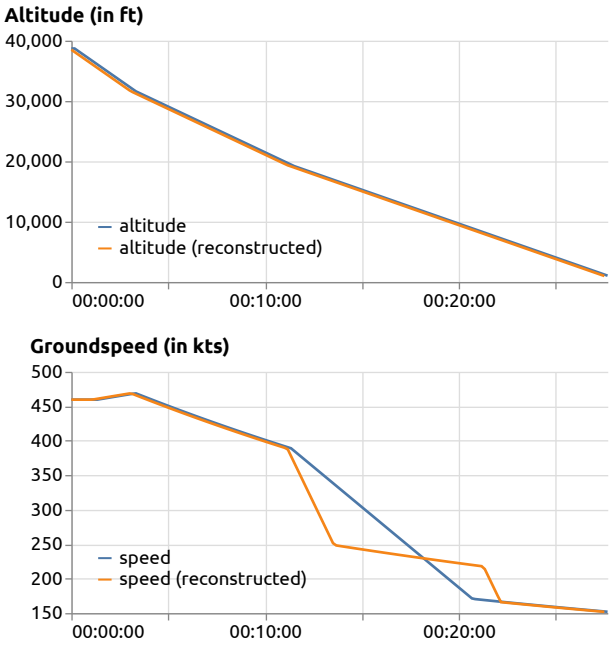

Figure 5: An example of generated and reconstructed vertical flight profiles (166 data points in both trajectories). Generated trajectory is in Blue, while reconstructed trajectory is in orange.

V. Discussion

Section IV applies our proposed approach on two existing trajectory generation methods: the first method generates 4D trajectories based on the identification of main flows in a given airspace, after sampling from a Gaussian Mixture Model; the second method focuses on the physical characterisation of performance for the vertical part of the trajectory using statistical kinematic models constructed from past flown trajectories.

While other metrics would be available for purely physical generation approaches, we focus here on the evaluation of the physical realism for statistically generated trajectories. Both approach relies on a fast-time air traffic simulator, BlueSky, to evaluate the physical realism of our methods. Even through BlueSky is a well established simulator within the ATM research community, we acknowledge that this evaluation approach comes with the a risk of evaluating the relevance of the method and the performance of the simulator simultaneously.

Keeping these limitations in mind, we come up with the following key takeaways:

1) When evaluating a generation method, one shall clearly state the expectations for the use case (see Section III-A) and consider relevant metrics in accordance with the original requirement (Section III-B);
2) To compare multiple generation methods, one shall avoid using a metric of which one of the methods relies on. Further, one should be cautious when using the metric as objective for improving the generation method, as the method could become biased towards the metric;
3) One shall evaluate results from different metrics to assess whether the generation method is appropriate for the application being considered. For example, Figure 2 revealed that the generation method was able to reproduce well the main patterns observed in the terminal airspace but was not able to generate outlying trajectory behaviors that are naturally harder to reconstruct. Therefore, it would be more suitable for an air traffic flow and capacity management application than for a Monte-Carlo simulation in collision risk modelling.

VI. Conclusion

After an extensive literature review about trajectory generation methods, in this paper, we detailed the fundamental elements required to build an trajectory generation evaluation framework: common expectations, objective quantitative metrics and guidelines to evaluate generation methods. With case studies, we applied this methodology for evaluating two generation methods with a new critical eyes. The approach

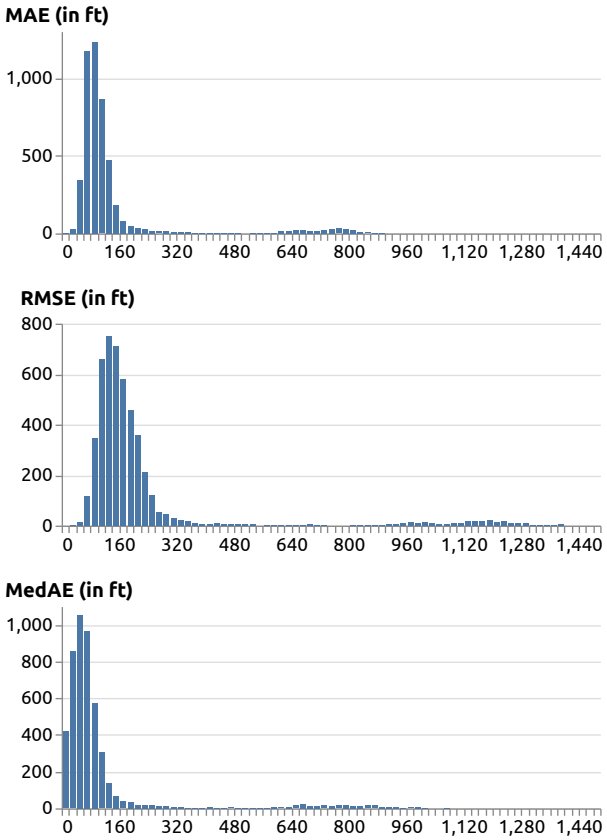

Figure 6: Distributions of error metrics used for comparing vertical profiles of generated and simulated trajectories, showing an average error of about 100 ft.

helped identifying corner cases and room for improvement in both methods when considering multiple use cases. More importantly, the evaluation framework provides an objective way to compare the performance of different methods whether a model-driven or data-driven approach is pursued. Future works shall apply this evaluation framework to objectively assess and compare the data-driven trajectory generation performance with new methods.

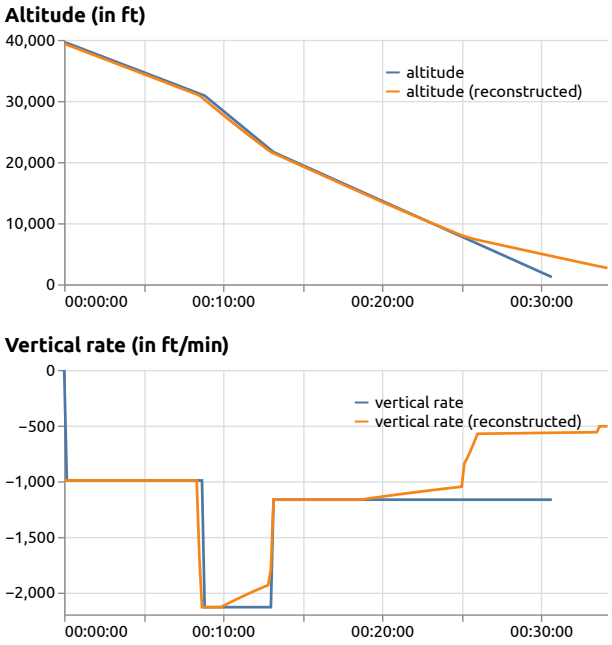

Figure 7: Sample generated flight and its reconstructed vertical profile. This example shows a limitation due to the limitation of descent rate at final approach.

References

[1] M. Poppe and J. Buxbaum, "Clustering climb profiles for vertical trajectory analysis," in *Proceedings of the 10th SESAR Innovation Days*, 2020.

[2] G. Sabhnani, A. Yousefi, I. Kostitsyna, J. Mitchell, V. Polishchuk, and D. Kierstead, "Algorithmic traffic abstraction and its application to nextgen generic airspace," in *Proc. 10th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, 2010.

[3] S. Mondoloni and N. Rozen, "Aircraft trajectory prediction and synchronization for air traffic management applications," *Progress in Aerospace Sciences*, vol. 119, p. 100640, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037604212030052X

[4] D. Jacquemart and J. Morio, "Conflict probability estimation between aircraft with dynamic importance splitting," *Safety Science*, vol. 51, no. 1, pp. 94–100, Jan. 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925753512001300

[5] M. Waller, J. Rigopoulos, D. Blackman, and T. Berreen, "Considerations in the application of dynamic programming to optimal aircraft trajectory generation," in *IEEE Conference on Aerospace and Electronics*, 1990, pp. 574–579. [Online]. Available: http://ieeexplore.ieee.org/document/112828/

[6] M. Soler, A. Olivares, and E. Staffetti, "Hybrid Optimal Control Approach to Commercial Aircraft Trajectory Planning," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 3, pp. 985–991, May 2010. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.47458

[7] R. B. Patel and P. J. Goulart, "Trajectory Generation for Aircraft Avoidance Maneuvers Using Online Optimization," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 1, pp. 218–230, Jan. 2011. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.49518

[8] M. Kamgarpour, V. Dadok, and C. Tomlin, "Trajectory generation for aircraft subject to dynamic weather uncertainty," in *49th IEEE Conference on Decision and Control (CDC)*. Atlanta, GA, USA: IEEE, Dec. 2010, pp. 2063–2068. [Online]. Available: http://ieeexplore.ieee.org/document/5717889/

[9] G. Jarry, N. Couellan, and D. Delahaye, "On the Use of Generative Adversarial Networks for Aircraft Trajectory Generation and Atypical Approach Detection," in *Proceedings of the 6th ENRI International Workshop on ATM/CNS*, Tokyo, Japan, Oct. 2019. [Online]. Available: https://hal-enac.archives-ouvertes.fr/hal-02267170

[10] M. C. R. Murça and M. de Oliveira, "A Data-Driven Probabilistic Trajectory Model for Predicting and Simulating Terminal Airspace Operations," in *Proceedings of the 39th IEEE/AIAA Digital Avionics Systems Conference (DASC)*, 2020.

[11] A. Nuic, D. Poles, and V. Mouillet, "BADA: An advanced aircraft performance model for present and future ATM systems," *International Journal of Adaptive Control and Signal Processing*, vol. 24, no. 10, pp. 850–866, Aug. 2010. [Online]. Available: http://doi.wiley.com/10.1002/acs.1176

[12] J. Sun, J. M. Hoekstra, and J. Ellerbroek, "OpenAP: An Open-Source Aircraft Performance Model for Air Transportation Studies and Simulations," *Aerospace*, vol. 7, no. 8, p. 104, Jul. 2020. [Online]. Available: https://www.mdpi.com/2226-4310/7/8/104

[13] D. Delahaye, S. Puechmorel, P. Tsiotras, and E. Féron, "Mathematical models for aircraft trajectory design: A survey," in *Air Traffic Management and Systems*, 2014, pp. 205–247.

[14] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, "LSTM-based flight trajectory prediction," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[15] C. J. Wang, S. K. Tan, and K. H. Low, "Three-dimensional (3D) Monte-Carlo modeling for UAS collision risk management in restricted airport airspace," *Aerospace Science and Technology*, vol. 105, p. 105964, 2020.

[16] H. Blom, B. Bakker, M. Everdij, and M. Van Der Park, "Collision risk

modeling of air traffic," in *2003 European Control Conference (ECC)*. IEEE, 2003, pp. 2236–2241.

[17] A. Eckstein, "Data driven modeling for the simulation of converging runway operations," *Proceedings of the 4th International Conference on Research in Air Transportation*, 2010.

[18] G. Jarry, H. Almoctar, D. Delahaye, and C. Hurter, "Interactive Trajectory Modification and Generation with FPCA," in *ICRAT 2020, 9th International Conference for Research in Air Transportation*, 2020.

[19] P. Besse, B. Guillouet, J.-M. Loubes, and R. François, "Review and Perspective for Distance Based Trajectory Clustering," *arXiv:1508.04904 [cs, stat]*, Aug. 2015. [Online]. Available: http://arxiv.org/abs/1508.04904

[20] D. Delahaye, S. Puechmorel, S. Alam, and E. Féron, "Trajectory mathematical distance applied to airspace major flows extraction," in *Proceedings of the 5th ENRI International Workshop on ATM/CNS*, 2017, pp. 51–66.

[21] X. Olive and J. Morio, "Trajectory clustering of air traffic flows around airports," *Aerospace Science and Technology*, vol. 84, pp. 776–781, Jan. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S127096381731129X

[22] X. Olive, L. Basora, B. Viry, and R. Alligier, "Deep Trajectory Clustering with Autoencoders," in *Proceedings of the 9th International Conference on Research in Air Transportation*, 2020.

[23] X. Olive, A. Tanner, M. Strohmeier, M. Schafer, M. Feridun, A. Tart, I. Martinovic, and V. Lenders, "OpenSky Report 2020: Analysing in-flight emergencies using big data," in *Proceedings of the 39th {IEEE}/{AIAA} Digital {Avionics} {Systems} {Conference} ({DASC})*, 2020, p. 10.

[24] J. M. Hoekstra and J. Ellerbroek, "Bluesky atc simulator project: an open data and open source approach," in *Proceedings of the 7th International Conference on Research in Air Transportation*, vol. 131. FAA/Eurocontrol USA/Europe, 2016, p. 132.

[25] J. Sun, *The 1090 Megahertz Riddle: A Guide to Decoding Mode S and ADS-B Signals*, 2nd ed. TU Delft OPEN Publishing, 2021.

[26] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. 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, pp. 83–94.

[27] X. Olive, "traffic, a toolbox for processing and analysing air traffic data," *Journal of Open Source Software*, vol. 4, no. 39, p. 1518, Jul. 2019. [Online]. Available: http://joss.theoj.org/papers/10.21105/joss.01518

[28] X. Olive and L. Basora, "Reference data sets for detection and identification of significant events in historical aircraft trajectory data." Dec 2019. [Online]. Available: https://doi.org/10.6084/m9.figshare.11406735.v1

[29] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996, pp. 226–231.

[30] G. J. McLachlan and K. E. Basford, *Mixture Models: Inference and Applications to Clustering*. New York: Marcel Dekker, 1988.

[31] J. Sun, J. Ellerbroek, and J. M. Hoekstra, "Wrap: An open-source kinematic aircraft performance model," *Transportation Research Part C: Emerging Technologies*, vol. 98, pp. 118–138, 2019.