Predicting aircraft trajectory uncertainties for terminal airspace design evaluation

Zhu, Xinting; Hong, Ning; He, Fang; Lin, Yu; Li, Lishuai; Fu, Xiaowen

Published in:
Journal of Air Transport Management

Published: 01/10/2023

Document Version:
Final Published version, also known as Publisher's PDF, Publisher's Final version or Version of Record

License:
CC BY-NC-ND

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1016/j.jairtraman.2023.102473

Publication details:

Citing this paper
Please note that where the full-text provided on CityU Scholars is the Post-print version (also known as Accepted Author Manuscript, Peer-reviewed or Author Final version), it may differ from the Final Published version. When citing, ensure that you check and use the publisher's definitive version for pagination and other details.

General rights
Copyright for the publications made accessible via the CityU Scholars portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights. Users may not further distribute the material or use it for any profit-making activity or commercial gain.

Publisher permission
Permission for previously published items are in accordance with publisher's copyright policies sourced from the SHERPA RoMEO database. Links to full text versions (either Published or Post-print) are only available if corresponding publishers allow open access.

Take down policy
Contact lbscholars@cityu.edu.hk if you believe that this document breaches copyright and provide us with details. We will remove access to the work immediately and investigate your claim.

Download date: 28/10/2023
Predicting aircraft trajectory uncertainties for terminal airspace design evaluation

Xinting Zhu, Ning Hong, Fang He, Yu Lin, Lishuai Li, Xiaowen Fu

ABSTRACT

The terminal airspace that surrounds an airport is the area with high flight density and complex structure. Aircraft are asked to follow the standard arrival and departure routes in terminal airspace, yet the actual trajectories may deviate due to air traffic control (ATC) instructions, pilots’ decisions, surveillance and flying performance variations, etc. Predicting aircraft trajectories considering such uncertainties plays a crucial role in evaluating a redesign of the standard routes. Traditional simulation approaches for generating aircraft trajectories in a terminal airspace are cumbersome to use as it requires a detailed setup for each new scenario, while most existing data-driven methods can only be used in an airspace with historical trajectories, not applicable to new structure designs or other terminal areas. To fill in gap, in this paper, we develop a new model based on Multilayer Perceptron Neural Network (MLPNN) to predict aircraft trajectories with uncertainties for terminal airspace design evaluations. A key feature of the proposed model is that it is trained on existing standard routes yet it can be applied to new standard routes to generate trajectories. The enabler of the model’s transferability is a novel input-and-output construction method for feature representations of raw trajectory data based on domain knowledge, including trajectory reconstruction, feature engineering, and output designing. After the input-and-output construction, a supervised learning model based on MLPNN is built to predict the standard deviations from the extracted features using historical trajectory data of existing standard routes. Once the model is built, trajectories with uncertainty can be simulated, through applying Gaussian distribution and exponential moving average algorithms, even on newly designed standard routes, where no aircraft have flown yet. Subsequently, new terminal airspace designs could be evaluated for their safety, efficiency, and environmental implications based on the simulated trajectories. The proposed model was tested on real-world operational data. Results showed that the model can quantify the characteristics of aircraft trajectories that are transferable across standard routes, and generate trajectories for new standard routes. We also demonstrated the proposed model on evaluating deficiencies on fuel consumption of actual arrival trajectories compared with the designed arrival routes. The generated trajectories showed 23%-37% more fuel consumption on average than the standard arrival routes in the terminal airspace of Hong Kong International Airport, which was validated with actual flight data.

1. Introduction

The terminal airspace that surrounds an airport is the area with high flight density and complex structure. In fact, nearly half of all fatalities, about 49%, occur on approach and landing in terminal airspace from 2009 through 2019 (Boeing Commercial Airplanes, 2020). The changes in airport and its terminal airspace may affect safety, fuel consumption, noise impact, and pollutant impact (M. Z. Li et al., 2018; M. Z. Li and Ryerson, 2017; Rosenow and Fricke, 2019). Therefore, a comprehensive evaluation of terminal airspace design is essential to support the safe, efficient, and environmental-friendly operations of new airports.

To evaluate the safety, efficiency, and environmental impact of airspace redesign, generating aircraft trajectories considering uncertainties plays a crucial role. In the terminal area, there are standard...
procedures, such as Standard Instrument Departure routes (SID) and Standard Arrival Routes (STAR). However, actual trajectories may be different from the standard ones, as shown in Fig. 1. Such deviations may be caused by many factors, such as weather conditions, air traffic control (ATC) instructions, pilots’ decisions, the performance of aircraft, etc (T. Li and Wan, 2021). Particularly a large amount of trajectory non-conformance happened in terminal airspace come from congestions and flight separation requirements. Even in good weather conditions, there might be many VFR flights, resulting in congestion and trajectory deviations if the airport is not strategically slot constrained (e.g., many US airports). Thus, trajectory generation models to characterize such deviations and predict aircraft trajectories with probabilistic uncertainties are needed as an essential step for the analysis of airspace operations.

In the field of aviation, there is a vast amount of research related to aircraft trajectory prediction. Most of these studies focus on predicting future aircraft motion for air traffic conflict detection and resolution. These studies account for the uncertainty affecting the aircraft future positions for the avoidance of collision, via a multiple model method (Jilkov et al., 2019; Liu and Hwang, 2011; Yepes et al., 2007), the worst-case approach (Huang and Tomlin, 2009; Tomlin et al., 1998), a Sequential Monte Carlo method (Lymeropoulos and Lyster, 2010; Visintini et al., 2006), a Hidden Markov Model (Ayhan and Samet, 2016; Seah and Hwang, 2009) and others. These researches focus on quantifying the probability of conflicts between two aircraft or more and perform trajectory prediction for individual aircraft, not for a fleet of aircraft in a particular airspace.

Another line of studies related to trajectory prediction is on the dynamic terminal route concept, developing trajectory models to optimize aircraft movement in terminal airspace (Murça and Müller, 2015). Pfeil developed an A*-based algorithm for the design of three-dimensional (3D) terminal routes under uncertain weather conditions (Pfeil, 2011). Zhou et al. introduce a fast-marching method to generate 3D SIDs and STARs at a strategic level, where a simulated annealing method is used to find the optimal order (Zhou et al., 2014). Sidiropoulos et al. propose a priority-based framework to design dynamic arrival and departure routes for the multi-airport system (Sidiropoulos et al., 2018). These methods aim at designing new routes but not quantifying the uncertainties of flying existed routes.

Traditional methods for evaluating new designs of arrival/departure routes in a terminal airspace are simulation-based. Several simulation models and products are available, including the Terminal Area Route Generation and Traffic Simulation (TARGETS) developed by MITRE (CSSI Inc.; MITRE, 2021) and the Future ATM Concept Evaluation Tool (FACET) (Billmorla et al., 2001) developed by NASA Ames Research Center. However, these simulation tools either require intensive inputs or human-in-the-loop setup to obtain the uncertainties involved with flying the standard routes, or lack of the capability to simulate the uncertainties.

Recent efforts have been made to train a statistical or machine learning model on historical surveillance data directly, to capture nondeterministic influences for aircraft trajectory prediction (Bongiorno et al., 2017). One important branch is studying flight historical data by regression analysis, such as a multiple linear regression (S. Hong and Lee, 2015), a functional regression (Tastambekov et al., 2014), a local polynomial regression (Hamed et al., 2013), a stepwise regression (de Leege et al., 2013), and neural networks (Alligier et al., 2015; Khan et al., 2021). These approaches have returned successful outcomes for improving the trajectory prediction accuracy. Another branch involves clustering turning points or clustering historical aircraft trajectories in terminal airspace (Barratt et al., 2018; Gariel et al., 2011; Jarry et al., 2020; Mahboubi and Kochenderfer, 2017; Marzuoli et al., 2014; Murça et al., 2020; Murça and Halsman, 2019; Olive and Morio, 2019; Ren and Li, 2018). However, a common limitation of these methods is that they can only be used on existing standard routes of the current airport where historical trajectory data are available, while they cannot be applied to routes under design, terminal airspace with new structures, or other airports.

To fill this gap in trajectory generation, we develop a novel machine learning model for predicting aircraft trajectory uncertainties in a terminal airspace with new route structures. The primary idea of our proposed method is to extract features that could be generalized for different route structures from the historical surveillance data. The proposed model can capture statistical properties of the aircraft trajectory uncertainty in a terminal airspace. The main advantage of our model compared with existing methods is that it is transferable to new standard routes. By introducing a trajectory reconstruction procedure, the features are re-constructed from raw trajectory data and the features are universal to different route structures.

This paper is structured as follows: The proposed framework is described in Section 2. Model evaluation and testing are presented in Section 3. In Section 3, we also apply the proposed approach for fuel consumption evaluation in the terminal airspace of Hong Kong International Airport. Finally, conclusions and future works are presented in Section 4.

2. Methodology

We propose a machine learning approach to model the uncertainty of aircraft trajectories in terminal airspace and generate trajectories for terminal airspace redesign analysis. The model, referred to as Trajectory Generation Model in this paper, predicts the deviations of actual aircraft
trajectories from the designed routes. The simulated trajectories generated model can then be used for a number of analysis, e.g. fuel efficiency evaluation, environmental impact assessment, airspace safety assessment, etc., as illustrated in Fig. 2.

The model uses a transfer learning approach, training on existing standard routes but can be applied to new standard routes for trajectory generation. The inputs of the model are historical trajectory data and existing standard arrival/departure routes. Then, the uncertainty of trajectories relative to standard routes is characterized and parameterized via the input-and-output construction step and the supervised learning via MLPNN step. Lastly, when a new design of standard routes is given, the model can simulate aircraft trajectories flying these newly designed routes, with probabilistic uncertainties on a system level.

Overall, three modules are included in the proposed model: 1) input-and-output construction, 2) supervised learning via MLPNN, and 3) trajectory generation, as illustrated in Fig. 3. In this paper, we demonstrate how this model can be used for fuel consumption analysis, as an example of the follow-up evaluation analysis. The details are provided in the following sub-sections.

1. Input-and-Output Construction

A key challenge of such a model lies in how to make it adaptable on different terminal airspace routes and airports. Airports vary in size, layout, runway configuration, etc., and the corresponding terminal airspace structures differ as well. If a learning model is directly trained on raw trajectory data, it won’t be applicable to new standard routes or new airports where trajectory data are not available yet. Therefore, the
There are three main complications with the trajectories in their original format. First, the raw trajectories are specific to each runway. Features directly extracted from the raw data would be meaningful for other runways, standard routes, or airports. Second, the trajectories may contain outliers or noises. Third, the trajectories are not of equal length.

We tackle the second complication first as it is a simple data filtering process. We use the exponential moving average (EMA) algorithm (Wang et al., 2018) to smooth trajectory data, acting as low-pass filters to remove high-frequency noise. Despite its simplicity, it is highly competitive with other more sophisticated trajectory smoothing methods. The simplest form of exponential smoothing is given by the formula (Chazal et al., 2011),

$$ P_i = \frac{1}{\alpha} + (1 - \alpha) P_{i-1}, i > 1 $$

where, $\alpha$ represents the degree of weighting decrease, which is a constant smoothing factor between 0 and 1. $P_i$ denotes one of the aircraft 3D position variables ($P_x, P_y, P_z$) at the data point $i$ and $P_i$ is the value after smoothing by EMA at the data point $i$.

Then, to solve the second complication, we build new Cartesian coordinate systems in each runway to transform position data into coordinates that are universal to any runway, as illustrated in Table 1 and Fig. 4. According to the landing slot of the aircraft, its 3D position in coordinate systems in each runway to transform position data into coordinates that are universal to any runway, as illustrated in Table 1 and Fig. 4. According to the landing slot of the aircraft, its 3D position in coordinate systems in each runway to transform position data into coordinates that are universal to any runway, as illustrated in Table 1 and Fig. 4. According to the landing slot of the aircraft, its 3D position in coordinate systems in each runway to transform position data into coordinates that are universal to any runway, as illustrated in Table 1 and Fig. 4. According to the landing slot of the aircraft, its 3D position in coordinate systems in each runway to transform position data into coordinates that are universal to any runway, as illustrated in Table 1 and Fig. 4.

There are three main complications with the trajectories in their original format. First, the raw trajectories are specific to each runway. Features directly extracted from the raw data would be meaningful for other runways, standard routes, or airports. Second, the trajectories may contain outliers or noises. Third, the trajectories are not of equal length.

We tackle the second complication first as it is a simple data filtering process. We use the exponential moving average (EMA) algorithm (Wang et al., 2018) to smooth trajectory data, acting as low-pass filters to remove high-frequency noise. Despite its simplicity, it is highly competitive with other more sophisticated trajectory smoothing methods. The simplest form of exponential smoothing is given by the formula (Chazal et al., 2011),

$$ P_i = \frac{1}{\alpha} + (1 - \alpha) P_{i-1}, i > 1 $$

where, $\alpha$ represents the degree of weighting decrease, which is a constant smoothing factor between 0 and 1. $P_i$ denotes one of the aircraft 3D position variables ($P_x, P_y, P_z$) at the data point $i$ and $P_i$ is the value after smoothing by EMA at the data point $i$.

Then, to solve the second complication, we build new Cartesian coordinate systems in each runway to transform position data into coordinates that are universal to any runway, as illustrated in Table 1 and Fig. 4. According to the landing slot of the aircraft, its 3D position in coordinate systems in each runway to transform position data into coordinates that are universal to any runway, as illustrated in Table 1 and Fig. 4. According to the landing slot of the aircraft, its 3D position in coordinate systems in each runway to transform position data into coordinates that are universal to any runway, as illustrated in Table 1 and Fig. 4. According to the landing slot of the aircraft, its 3D position in coordinate systems in each runway to transform position data into coordinates that are universal to any runway, as illustrated in Table 1 and Fig. 4.

### Table 1
The right-handed cartesian coordinate systems for each runway.

<table>
<thead>
<tr>
<th>The Origin</th>
<th>The Positive Orientation of Axis</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold position of the runway</td>
<td>X: The opposite direction of the runway axis</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td>Y: Rotating X-axis 90° counter clockwise axis</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td>Z: Vertically upward axis</td>
<td>m</td>
</tr>
</tbody>
</table>

Fig. 4. The framework of the trajectory generation model.

### Table 2
Five key informative features for supervised learning.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIS</td>
<td>Accumulated horizontal flying distance to the runway threshold</td>
<td>m</td>
</tr>
<tr>
<td>ALT</td>
<td>Relative altitude above the touchdown zone, namely Z</td>
<td>m</td>
</tr>
<tr>
<td>RAR</td>
<td>Relative angle to the positive X-axis</td>
<td>rad</td>
</tr>
<tr>
<td>TAV</td>
<td>Track angle variation between adjacent data points</td>
<td>rad</td>
</tr>
<tr>
<td>DTW</td>
<td>Horizontal distance to nearest waypoint from STAR profiles</td>
<td>m</td>
</tr>
</tbody>
</table>

Input-and-Output Construction module is developed to reconstruct trajectory data into designated features that capture the underlying patterns of uncertainties relative to the standard routes, so that these features could be easily adapted to other airports to test different procedures. This module includes a unified method to process original ungraded data to standard trajectory data of high quality, extract key features to be used as model inputs, and to design the outputs for subsequent supervised learning. There are three submodules: a) Trajectory Reconstruction, b) Feature Engineering, and c) Output designing in the Input-and-Output Construction.

1. Trajectory Reconstruction

To make our model a general approach, that can be trained by different data sources and applied in various terminal airspace, we firstly conduct the trajectory reconstruction. Trajectory reconstruction is the process of processing original ungraded data into standard trajectory data of high quality. The Automatic Dependent Surveillance-Broadcast (ADS-B) data is selected as the data source of the aircraft trajectory prediction model. Each ADS-B data record we collected contains.

1. Type of operation (departure/arrival),
2. Aircraft number,
3. Coordinated Universal Time (UTC) timestamp $t$,
4. Position information, longitude $P_x$, latitude $P_y$, and altitude $P_z$,
5. Track $\psi$,
6. Ground speed $V_h$,
7. Vertical velocity $V_v$.

Each record with the same aircraft number belongs to an aircraft $j$, and the group of all records for that aircraft forms the trajectory $T_j, i = 1, \ldots, n$, where $n$ is the total number of trajectories in the dataset.

There are three main complications with the trajectories in their original format. First, the raw trajectories are specific to each runway. Features directly extracted from the raw data would be meaningful for other runways, standard routes, or airports. Second, the trajectories may contain outliers or noises. Third, the trajectories are not of equal length.

We tackle the second complication first as it is a simple data filtering process. We use the exponential moving average (EMA) algorithm (Wang et al., 2018) to smooth trajectory data, acting as low-pass filters to remove high-frequency noise. Despite its simplicity, it is highly competitive with other more sophisticated trajectory smoothing methods. The simplest form of exponential smoothing is given by the formula (Chazal et al., 2011),

$$ P_i = \frac{1}{\alpha} + (1 - \alpha) P_{i-1}, i > 1 $$

where, $\alpha$ represents the degree of weighting decrease, which is a constant smoothing factor between 0 and 1. $P_i$ denotes one of the aircraft 3D position variables ($P_x, P_y, P_z$) at the data point $i$ and $P_i$ is the value after smoothing by EMA at the data point $i$.

Then, to solve the second complication, we build new Cartesian coordinate systems in each runway to transform position data into coordinates that are universal to any runway, as illustrated in Table 1 and Fig. 4. According to the landing slot of the aircraft, its 3D position in coordinate systems in each runway to transform position data into coordinates that are universal to any runway, as illustrated in Table 1 and Fig. 4. According to the landing slot of the aircraft, its 3D position in coordinate systems in each runway to transform position data into coordinates that are universal to any runway, as illustrated in Table 1 and Fig. 4. According to the landing slot of the aircraft, its 3D position in coordinate systems in each runway to transform position data into coordinates that are universal to any runway, as illustrated in Table 1 and Fig. 4.

### 2. Feature Engineering

Based on domain knowledge of air traffic control and flight operations, we engineered five key features from the reconstructed trajectory data, which describes the relative position of aircraft in reference to the standard route. These five key features, as illustrated in Table 2 and Fig. 5, are derived based on the reconstruction trajectory data and STAR profiles. The \textit{DIS} is the aircraft’s accumulated horizontal flying distance from its position to the runway threshold, and it could be obtained from the position data $(X, Y, Z)$. The \textit{ALT}, namely $Z$, means the relative altitude above the touchdown zone. The \textit{‘RAR’} is defined as the relative angle to X-axis in the right-handed coordinate system and its range is $(-\pi, \pi]$. We assume angles on the coordinate plane have an initial side on the ground plane and the terminal side of the angle is the side where the ground plane projective point of aircraft located. Positive angles on the coordinate plane are angles that go in a counterclockwise direction, while negative angles on the coordinate plane are angles that go in a clockwise direction. The \textit{TAV} represents the track angle variation between adjacent data points. The first data point is the landing site of aircraft, and select 150 nautical miles (NM) DIS backward for the trajectory. We set a data point at each 3 NM interval so that all trajectories have the length of 51 data points. Trajectories that are significantly shorter than 150 NM are discarded. Then, trajectories that are shorter than 150 NM are extrapolated and trajectories that are longer than 150 NM are truncated. Meanwhile, the parameters of ADS-B data, such as track $\psi$, ground speed $V_h$, and vertical velocity $V_v$, are obtained in each data points through linear interpolation.

The reconstructed trajectories have the same length after distance-aligning processing, which includes recomputed parameters: 3D position information $(X, Y, Z)$, track $\psi$, ground speed $V_h$, and vertical velocity $V_v$.

### 2. Feature Engineering

Based on domain knowledge of air traffic control and flight operations, we engineered five key features from the reconstructed trajectory data, which describes the relative position of aircraft in reference to the standard route. These five key features, as illustrated in Table 2 and Fig. 5, are derived based on the reconstruction trajectory data and STAR profiles. The \textit{DIS} is the aircraft’s accumulated horizontal flying distance from its position to the runway threshold, and it could be obtained from the position data $(X, Y, Z)$. The \textit{ALT}, namely $Z$, means the relative altitude above the touchdown zone. The \textit{‘RAR’} is defined as the relative angle to X-axis in the right-handed coordinate system and its range is $(-\pi, \pi]$. We assume angles on the coordinate plane have an initial side on the ground plane and the terminal side of the angle is the side where the ground plane projective point of aircraft located. Positive angles on the coordinate plane are angles that go in a counterclockwise direction, while negative angles on the coordinate plane are angles that go in a clockwise direction. The \textit{TAV} represents the track angle variation between adjacent data points. The first data point is the landing site of aircraft, and its \textit{TAV} is set to 0 rad. By deductive calculation backward step-by-step, it’s simple to derive \textit{TAV} values for the other data points based on the parameter track $\psi$. The \textit{DTW} refers to the horizontal distance to the nearest waypoint from STAR profiles. For each data
point, we calculate the horizontal distance between the aircraft and all waypoints from the relevant STAR profile, and the minimum value is the target value.

In summary, each data point in the reconstruction trajectory now contains five key informative features, namely DIS, ALT, RAR, TAV, and DTW, which are underlying attributes of the flight trajectory, relative to the standard route.

3. Output Designing

As for target outputs for the supervised learning in the next module, we use standard deviations to summarize the uncertainties of trajectories, as depicted in Table 3 and Fig. 6, which include 3D position information \((X, Y, Z)\), ground speed \(V_h\), and vertical velocity \(V_v\), in the reconstructed trajectory data.

To summarize, each data point now contains five key informative features (DIS, ALT, RAR, TAV, and DTW) as inputs, and five output variables \((\sigma_X, \sigma_Y, \sigma_Z, \sigma_{GS}, \sigma_{VS})\) designed for supervised learning.

2. Supervised Learning via MLPNN

After accomplishing the input-and-output construction, the next step is to develop the MLPNN model to predict designed outputs, based on the key informative features extracted from the reconstruction trajectory data and STAR (or SID) profiles. MLPNN is one of the most commonly used artificial neural networks. In this study, MLPNN is tested against a few other advanced regression methods, which are support vector regression (SVR), Random Forest (RF), and Gaussian process regression (GPR). Details on the performance comparison of different regression methods can be found in Section 3.3.

Here, we provide a brief introduction of the MLPNN model used in the paper. MLPNN normally consists of three sequential layers: input layer, hidden layer, and output layer. In this study, each input vector is

\[ x_i = [\text{DIS}, \text{ALT}, \text{RAR}, \text{TAV}, \text{DTW}] \]

which is a set of underlying attributes related to the flight trajectory. Further, each output vector \( y_i = [\sigma_X, \sigma_Y, \sigma_Z, \sigma_{GS}, \sigma_{VS}] \) is a set of standard deviations of 3D position information and aircraft velocity information.

The structure of MLPNN employed for the standard deviation prediction is shown in Fig. 7. The goal of the MLPNN model is to develop a neural network that can accurately predict the standard deviations based on the input features of future data, by updating the weights in the network from the errors calculated of each training. For each neuron \( j \) in the hidden layer, it would sum up its input signals \( x_i \), after multiplying them by their respective connection weights \( w_{ji} \). The output of each neuron is described as follows:

\[ y_j = f \left( \sum w_{ji} x_i + b \right) \]  (2)

where, \( b \) is a bias term and \( f \) is an activation function. The commonly used activation functions include the sigmoid function, the hyperbolic tangent, or a Rectified Linear Unit (ReLU) \((\text{Nair and Hinton, 2010})\).

During the training process, the weight \( w_{ji} \) is adjusted to minimize the differences between the desired and actual values of the output neurons. The minimization objective function is defined as,

\[ E = \frac{1}{2} \sum (y_d - y_j)^2 \]  (3)

where, \( y_d \) is the desired value and \( y_j \) is the actual value of the output for

Table 3

<table>
<thead>
<tr>
<th>Outputs</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sigma_X )</td>
<td>The standard deviation of the recomputed parameter ( X )</td>
<td>m</td>
</tr>
<tr>
<td>( \sigma_Y )</td>
<td>The standard deviation of the recomputed parameter ( Y )</td>
<td>m</td>
</tr>
<tr>
<td>( \sigma_Z )</td>
<td>The standard deviation of the recomputed parameter ( Z )</td>
<td>m</td>
</tr>
<tr>
<td>( \sigma_{GS} )</td>
<td>The standard deviation of the recomputed ground speed ( V_h )</td>
<td>m/s</td>
</tr>
<tr>
<td>( \sigma_{VS} )</td>
<td>The standard deviation of the recomputed vertical velocity ( V_v )</td>
<td>m/s</td>
</tr>
</tbody>
</table>

\[ \text{Fig. 5. Five key informative features for supervised learning.} \]

\[ \text{Fig. 6. The standard deviation of recomputed 3D position information (X, Y, Z).} \]

\[ \text{Fig. 7. The structure of MLPNN employed for the standard deviations prediction.} \]
3. Trajectory Generation

In the last module, we develop a method to simulate aircraft trajectories based on the output of the MLPNN module by applying Gaussian distribution and exponential moving average algorithms. The MLPNN trained in the previous step is not specific to a STAR. It can be transferred directly to another STAR and simulate its associated trajectories considering uncertainty. We apply a Gaussian distribution to simulate trajectories when given a new STAR in terminal airspace. First, we translate the new STAR into the five features as specified in Module I. Next, we use the trained MLPNN to obtain the five standard deviations. Lastly, we randomly generate the 3D position information \( (X, Y, Z) \), ground speed \( V_h \), and vertical velocity \( V_v \) according to five Gaussian distributions. These five Gaussian distributions have means, \( \mu_i = (X_i, Y_i, Z_i, V_{hi}, V_{vi}) \), based on the new STAR, and standard deviations, \( \sigma_i = (\sigma_{Xi}, \sigma_{Yi}, \sigma_{Zi}, \sigma_{GSi}, \sigma_{VSi}) \), calculated from the MLPNN module. The notations and the probability density functions (PDF) of these five simulated parameters in data point \( i \) on a trajectory are formulated as,

\[
X_i' \sim N(X_i, \sigma_{Xi}); \quad \text{PDF} \Rightarrow f(a) = \frac{1}{\sigma_{Xi} \sqrt{2\pi}} e^{-\frac{(a - X_i)^2}{2\sigma_{Xi}^2}} \tag{4}
\]
\[
Y_i' \sim N(Y_i, \sigma_{Yi}); \quad \text{PDF} \Rightarrow f(b) = \frac{1}{\sigma_{Yi} \sqrt{2\pi}} e^{-\frac{(b - Y_i)^2}{2\sigma_{Yi}^2}} \tag{5}
\]
\[
Z_i' \sim N(Z_i, \sigma_{Zi}); \quad \text{PDF} \Rightarrow f(c) = \frac{1}{\sigma_{Zi} \sqrt{2\pi}} e^{-\frac{(c - Z_i)^2}{2\sigma_{Zi}^2}} \tag{6}
\]
V′hi ∼ N(Vhi, σGSi); PDF ⇒ f(d) = \frac{1}{\sigma_{GSi} \sqrt{2\pi}} e^{-\frac{(d-Vhi)^2}{2\sigma_{GSi}^2}} (7) 

V′vi ∼ N(Vvi, σVSi); PDF ⇒ f(e) = \frac{1}{\sigma_{VSi} \sqrt{2\pi}} e^{-\frac{(e-Vvi)^2}{2\sigma_{VSi}^2}} (8)

Based on the PDFs, we can create five normally distributed sets of random numbers \((X′_i, Y′_i, Z′_i, V′_{hi}, V′_{vi})\) for each data point, and then generate new trajectories by making connections of these simulated points \((X, Y, Z)\) in order of data point sequencing number \(i\).

After simulating new trajectories, we do trajectory smoothing for the rationality of track angle variation using the EMA algorithm. The form of exponential smoothing for a simulated trajectory is given by the formulas,

\[ S_i = \begin{cases} X_i, & i = 1 \\ \beta X_i + (1 - \beta) S_{i-1}, & i > 1 \end{cases} \] (9)

Table 6
Overview of ADS-B data after input-and-output construction.

<table>
<thead>
<tr>
<th>Number of Trajectories</th>
<th>Data Points for Each Trajectory</th>
<th>Number of Datasets</th>
<th>Inputs of Datasets</th>
<th>Outputs of Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>30,798</td>
<td>51</td>
<td>1570698</td>
<td>(DIS, ALT, RAR, TAV, DTW)</td>
<td>(σX, σY, σZ, σGS, σVS)</td>
</tr>
</tbody>
</table>

Fig. 10. Sensitive analysis of smoothing factor \(a\).

Fig. 11. Coordinate system transformation for HKIA.

Fig. 12. Four types of arrival routes in Hong Kong terminal airspace.
where, $\beta$ represents the constant smoothing factor. $X_i$ is one of the new position variables at the data point $i$ and $S_i$ is the value after smoothing by the EMA at the data point $i$.

4. Preparation for Airspace Analysis

The trajectories generated from the previous modules are now ready to be used for the following evaluation analysis of airspace designs. Here, to demonstrate how to use the simulated trajectories for a fuel consumption evaluation. The fuel consumption estimation model, as shown in Fig. 8, is for fast assessment of system-level fuel consumption about terminal airspace redesigns, which is proposed by Hong and Li (N. Hong and Li, 2018). The inputs of this model are the aircraft’s altitude, the air temperature, the calibrated airspeed, flying distance, and the aircraft mass when descending. Some of these inputs are calculated from the simulated trajectories, while some are obtained from Meteorological Aerodrome Reports (METARs) and other sources. The detailed process is illustrated in Table 4. The output is the fuel burn when flying the simulated trajectories.

3. Evaluation and testing

The Airport Authority Hong Kong plan to expand Hong Kong International Airport (HKIA) into a three-runway system in project Master Plan 2030 by Airport Authority Hong Kong (2011), as shown in Fig. 9. In this context, we demonstrate the proposed trajectory generation model via a case study of HKIA terminal airspace to show how it can be used for fuel efficiency evaluation of airspace designs.

4. Operational data description

The ADS-B dataset used in this study is collected from Flightradar24 every minute for 3 months, namely January, May, and June in 2016, covering all flights landing at HKIA. Each item of the dataset consists of several attributes, which include the flight ID, latitude, longitude, altitude, track, origin airport, destination airport, aircraft type, date, and monitoring time. The ground speed and vertical velocity in our analysis are derived from the position measurements. The data description of the ADS-B dataset is given in Table 5.

### Table 7
Detailed setting of the train-test split.

<table>
<thead>
<tr>
<th>Arrival Routes</th>
<th>Training Set</th>
<th>Testing Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABBEY</td>
<td>9990</td>
<td>7719</td>
</tr>
<tr>
<td>BETTY</td>
<td>4843</td>
<td>8246</td>
</tr>
<tr>
<td>CANTO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIERA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 8
Parameter settings for MLPNN model.

<table>
<thead>
<tr>
<th>Hidden Layer Size</th>
<th>Activation Function</th>
<th>Weight Optimization</th>
<th>Exponential Decay Rate for 'Adam'</th>
<th>L2 Penalty Parameter</th>
<th>Random state value</th>
<th>Mini-batch size</th>
</tr>
</thead>
<tbody>
<tr>
<td>[5, 5, 3]</td>
<td>'ReLU'</td>
<td>'Adam' optimizer</td>
<td>$\beta_1 = 0.9; \beta_2 = 0.999$</td>
<td>0.0001</td>
<td>1</td>
<td>64</td>
</tr>
</tbody>
</table>

2. Input-and-Output Construction of ADS-B data

In this part, the objective is to process ADS-B data to the standard trajectory data of high quality, extract key features as inputs, and finally design the outputs for subsequent supervised learning. We handle the outlier and noise problems by using the EMA algorithm, with the smoothing factor $\alpha$ setting to 0.5 based on testing. The value of smoothing factor is decided by the sensitive analysis as shown in Fig. 10. We calculated the RMSE between raw trajectory and smoothed trajectory as a function of alpha ranging from 0 to 1 with a step of 0.1. Before conducting the trajectory reconstruction, we identify the landing runway in HKIA of each flight based on the position information. Then, we build new Cartesian coordinate systems for each runway, as shown in Fig. 11, to perform the trajectory reconstruction process. The 30,798 reconstructed trajectories all have the same length with 51 data points. Each data point includes recomputed parameters: 3D position information $X$, $Y$, $Z$, track $\psi$, ground speed $V_g$, and vertical velocity $V_v$. After that, the feature engineering and target outputs designing are executed for supervised learning based on the above-mentioned methods in Section 2.1. The detailed information of processed ADS-B data after input-and-output construction is summarized in Table 6.

After input-and-output construction, the identification of which STAR the flight trajectories belong to is performed. There are four main kinds of STARs, namely ABBEY, BETTY, CANTO, and SIERA, and they are different from each other in terms of trajectory characteristics. We classify the trajectories based on waypoints information from the STARs profiles, and the classification results are shown in Fig. 12. To test the transfer learning capability of our model, we train the model on three different datasets.

### Table 9
Parameter settings for RF, SVR, GPR, and OLS models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter Settings Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>(i) Number of trees $n_{trees} = 100$; (ii) The maximum depth of the tree $d_{max} = 11$; (iii) Criterion is 'MSE'; (iv) The features to consider when splitting $f_{max} = 3$</td>
</tr>
<tr>
<td>SVR</td>
<td>(i) Kernel used is 'RBF'; (ii) Kernel coefficient $\gamma = 0.2$; (iii) Regularization parameter $C = 1$; (iv) Epsilon value $\epsilon = 0.1$</td>
</tr>
<tr>
<td>GPR</td>
<td>(i) Kernel used is 'RBF'; (ii) Kernel parameters optimizer is 'L-BFGS-B' algorithm; (iii) Value added to the diagonal of the kernel matrix $\beta = 1e^{-10}$</td>
</tr>
</tbody>
</table>

### Table 10
Comparison of model Performance by prediction accuracy.

<table>
<thead>
<tr>
<th>Method</th>
<th>PA - 1-MAPE</th>
<th>$\sigma_x$</th>
<th>$\sigma_y$</th>
<th>$\sigma_z$</th>
<th>$\sigma_{GS}$</th>
<th>$\sigma_{VS}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>76.42%</td>
<td>81.34%</td>
<td>87.48%</td>
<td>85.94%</td>
<td>87.39%</td>
<td>87.77%</td>
</tr>
<tr>
<td>SVR</td>
<td>80.92%</td>
<td>77.99%</td>
<td>87.49%</td>
<td>82.81%</td>
<td>85.46%</td>
<td>86.17%</td>
</tr>
<tr>
<td>GPR (MEAN)</td>
<td>56.37%</td>
<td>44.52%</td>
<td>70.78%</td>
<td>80.96%</td>
<td>86.91%</td>
<td>87.47%</td>
</tr>
<tr>
<td>MLPNN</td>
<td>83.63%</td>
<td>86.91%</td>
<td>91.25%</td>
<td>87.99%</td>
<td>87.39%</td>
<td>87.77%</td>
</tr>
</tbody>
</table>
STARs and test it on the fourth STAR as depicted in Fig. 13 and Table 7, to simulate the situation when given a new or redesigned STAR profile in current terminal airspace.

3. Evaluation and Testing of MLPNN

To compare with MLPNN, we tested three other regression methods: 1) the RF method, which is suitable for complicated tasks, owing to its capacity in handling big data with numerous variables running into thousands (Genuer et al., 2017); 2) the SVR method, a kernel-based technique (Awad and Khanna, 2015); 3) the GPR method, which has internal structure to capture uncertainties in predictions (Boyle and Frean, 2005).

The metrics used to evaluate the models is the prediction accuracy (PA), which is defined to be calculated by the formula,

\[
PA = 100\% - MAPE
\]

where MAPE is the abbreviation of commonly used metrics mean absolute percentage error, and it is defined by the formula,

\[
MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{A_i - F_i}{A_i} \right|
\]

where \(A_i\) is the actual value and \(F_i\) is the forecast value. The models with high PA values are preferred.

Table 8 and Table 9 report the parameter settings for the MLPNN.
model, and the other four models, respectively. For the RF model training, the number of trees was set to 100. For SVR and GPR models training, the Gaussian radial basis function (RBF) kernel was used and defined as 
$$\exp\left(\frac{-\|S_1-S_2\|^2}{2\sigma^2}\right)$$, where $S_1$ and $S_2$ are two points and $\sigma$ is the kernel hyper-parameter. For GPR models, the kernel's hyper-parameters are optimized during fitting based on the 'L-BFGS-B' algorithm (Zhu et al., 1997). All the parameters in these models are optimized based on grid search and selected in the predefined ranges, and the best combination is selected according to the criterion, mean squared error (MSE) or root-mean-square error (RMSE).

The performances of the RF, SVR, GPR and MLPNN models trained by the processed ADS-B data are summarized in Table 10. We found that MLPNN performed significantly better, and was more accurate by 3% on average than the other four models. The estimation accuracy of MLPNN could achieve 87.43% on average, implying that the relationship between the features extracted and the designed output variables is well built for trajectory generation.

5. Trajectory generation in Hong Kong airspace

We simulate trajectories for arrival procedure design SIERA-6D based on the MLPNN module using Gaussian distribution and exponential moving average algorithms.

First, we do the trajectory reconstruction and feature engineering for the STAR, SIERA-6D. Next, the five standard variations are predicted from the developed MLPNN model. For the data point $i$ in SIERA-6D, we have the mean $\mu_i = (X_i, Y_i, Z_i, V_{hi}, V_{vi})$ and the standard deviations $\sigma_i = (\sigma_{X_i}, \sigma_{Y_i}, \sigma_{Z_i}, \sigma_{GS_i}, \sigma_{VS_i})$. Based on the assumptions and the PDFs illustrated in Section 2.3, we create five normally distributed sets of random numbers for each data point and then generate new trajectories by making connections of these simulated points $(X'_i, Y'_i, Z'_i, V'_{hi}, V'_{vi})$ in order of data point sequencing number $i$. Then, trajectory smoothing is conducted by applying the EMA algorithm with the smoothing factor $\beta$ set to 0.3. In this study, we simulated 10,000 new trajectories.

In this study, we focus on the designated flying ranges for different STARs: (1) for ABBEY-2B, the range of study is from waypoint ABBEY to the threshold of the runway; (2) for BETTY-2B, the range is from waypoint BETTY to the threshold of the runway; (3) for CANTO-2B, the range is from waypoint CANTO to the threshold of the runway; (4) for SIERA-6D, the range is from waypoint SIERA to the threshold of the runway.

The comparisons of the published arrival procedures, the actual trajectories, and the simulated trajectories from the proposed model, are shown in Fig. 14 for training sets and Fig. 15 for testing on SIERA-6D. The results show that even the model is trained on ABBEY-2B, BETTY-2B, and CANTO-2B, it can simulate trajectories for SIERA-6D and show

<table>
<thead>
<tr>
<th>Table 11</th>
<th>Fuel burn in terminal airspace estimation results.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuel Consumption</td>
<td>Mean Value</td>
</tr>
<tr>
<td>FDR Data</td>
<td>1136.8 kg</td>
</tr>
<tr>
<td>Simulation Data</td>
<td>1126.5 kg</td>
</tr>
<tr>
<td>Estimation Accuracy</td>
<td>99.10%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 12</th>
<th>Actual fuel burn compared with ideal fuel burn in terminal airspace.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABBEY-2B</td>
<td>BETTY-2B</td>
</tr>
<tr>
<td>Ideal Fuel Burn (kg)</td>
<td>600.0</td>
</tr>
<tr>
<td>Actual Fuel Burn (Mean) (kg)</td>
<td>823.6</td>
</tr>
<tr>
<td>Over Consumption (kg)</td>
<td>223.6</td>
</tr>
<tr>
<td>Percentage of Over Consumed (%)</td>
<td>37.3</td>
</tr>
<tr>
<td>Estimated Fuel Burn (kg)</td>
<td>835.1</td>
</tr>
<tr>
<td>Estimation Accuracy (%)</td>
<td>98.6</td>
</tr>
</tbody>
</table>
high similarity compared with the actual trajectories.

5. Fuel Consumption Estimation and Analysis

In this part, we evaluate the arrival procedure SIERA-6D of Hong Kong terminal airspace from the perspective of fuel efficiency, using a fuel consumption estimation model developed by Hong and Li (N. Hong and Li, 2018). This fuel consumption estimation model was trained based on the Digital Flight Data Recorder (FDR) data of 2686 flights landing at HKIA during November in 2014, and from April to June in 2015. As shown in Fig. 8, the inputs of the fuel consumption estimation model are the aircraft altitude, the air temperature, the calibrated airspeed, flying distance, and the aircraft mass when descending. The outputs are the average fuel burn in the terminal area when flying in SIERA-6D designs. The prediction accuracy of the fuel consumption estimation model is 96.6% on average. The detailed information about the fuel consumption estimation model training process can be found in (N. Hong and Li, 2018).

Feeding the simulated trajectories of SIERA-6D into the fuel consumption model, we obtain the fuel burn if SIERA-6D were flew. Compared with the actual fuel consumption from the FDR data, the fuel consumption of the simulated trajectories is in agreement, as shown in Table 11 and Fig. 16.

To further demonstrate the value of generating trajectories
considering uncertainties, we analyze the difference between the ideal fuel burn if aircraft could follow STAR precisely and the fuel burn in practice. The results are shown in Table 12. When flying SIERA-6D, the ideal fuel burn is 900 kg if the aircraft could fly exactly as the designed procedure from waypoint SIERA to the threshold of the runway. This reference number is obtained from an airline’s flight planning system. The actual fuel burn obtained from FDR data is more than 1100 kg, which indicates that the current operations of SIERA-6D in Hong Kong airspace are not fuel-efficient. Similar analyses are conducted for four STARs. Compared with the ideal fuel consumption values, the actual fuel consumption is 37.27%, 31.13%, 23.31% and 26.31% more on average for ABBEY-2B, BETTY-2B, CANTO-2B and SIERA-6B/6D, respectively. Fig. 17 shows the flight profiles of the actual operations compared with the ideal ones (highlighted in red). Currently, aircraft fly much lower than the planned profiles, which is a major factor that causes the fuel over-consumption of flights that arrive at Hong Kong airport. If the proposed trajectory generation model was used, similar numbers can be obtained via a fast system-level simulation approach without requiring the FDR data.

6. Conclusion

In this research, we developed a novel trajectory generation model based on MLPNN, which predicts aircraft trajectories with uncertainties for airspace redesign analysis. The proposed framework included three parts: 1) input-and-output construction, 2) supervised learning via MLPNN, and 3) trajectory generation. Trajectory reconstruction of operational data, feature engineering, and output designing for supervised learning were performed first. Next, the MLPNN was built to predict the relationship between the standard deviations and extracted features. Finally, the trajectories with uncertainties were simulated by applying Gaussian distribution and EMA algorithms.

The proposed framework was evaluated and tested on ADS-B data in Hong Kong terminal airspace. We found that the MLPNN model achieved good prediction performance and it was better than other machine learning methods based on RF, SVR, and GPR, respectively. We also demonstrated the usage of the proposed framework in estimating the real fuel consumption, to evaluate current STAR designs from the perspective of fuel efficiency in Hong Kong terminal airspace. In addition, our proposed framework can be adapted to evaluate standard procedures in other terminal airspaces.

However, in this model, the shape of simulated trajectories is closely linked to the probability distribution functions assumed in this research; however, we assume that simulated trajectories follow Gaussian distribution, and it may not hold for some special segments of the trajectories in real operations. For example, the current model cannot capture the characteristics of holding patterns. Besides, we only considered the diagonal variance estimation of Gaussian while ignoring the prediction of covariance, which will be further investigated. Other distribution assumptions will have potential to improve the Gaussian assumption such as truncated Gaussians could be applied when defining the normal range and would like to investigate the possibility of extreme values outside the range. Normalizing flow will be another worthy direction to assume the distribution since it can model the underlying complex probabilistic distributions. Dynamical system approach could be another direction to explore.

Future work is planned to focus on two parts, 1) improvement on the feature engineering part to capture special patterns of the flight routes, such as the holding pattern, 2) extension of the proposed model to trajectory generation of other types of vehicles, such as unmanned aerial vehicles, freighters, maritime vessels cars, etc.

Author statement

Xinting Zhu: Methodology, Software, Writing- Reviewing and Editing; Ning Hong: Methodology, Visualization, Writing- Original draft preparation; Fang He: Data curation, Investigation; Yu Lin: Software, Validation; Lishuai Li: Conceptualization, Supervision, Project administration, Funding acquisition; Xiaowen Fu: Supervision.

Data availability

The data that has been used is confidential.

Acknowledgment

The work was supported by the Hong Kong Research Grants Council General Research Fund (Project No. 11215119 and 11209717), ITF - Guangdong-Hong Kong Technology Cooperation Funding Scheme (Ref. No. GHP/145/20), and CityU Strategic Interdisciplinary Research Grant (Project Ref. No. 2022SIRG035).

References


