Delft University of Technology

OpenAP

An open-source aircraft performance model for air transportation studies and simulations

Sun, Junzi; Hoekstra, Jacco M.; Ellerbroek, Joost

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

OpenAP: An Open-Source Aircraft Performance Model for Air Transportation Studies and Simulations

Junzi Sun * , **Jacco M. Hoekstra and Joost Ellerbroek**

Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1,
2629 HS Delft, The Netherlands; j.m.hoekstra@tudelft.nl (J.M.H.); j.ellerbroek@tudelft.nl (J.E.)

* Correspondence: j.sun-1@tudelft.nl

Abstract: Air traffic simulations serve as common practice to evaluate different concepts and methods for air transportation studies. The aircraft performance model is a key element that supports these simulation-based studies. It is also an important component for simulation-independent studies, such as air traffic optimization and prediction studies. Commonly, contemporary studies have to rely on proprietary aircraft performance models that restrict the redistribution of the data and code. To promote openness and research comparability, an alternative open performance model would be beneficial for the air transportation research community. In this paper, we introduce an open aircraft performance model (OpenAP). It is an open-source model that is based on a number of our previous studies, which were focused on different components of the aircraft performance. The unique characteristic of OpenAP is that it was built upon open aircraft surveillance data and open literature models. The model is composed of four main components, including aircraft and engine properties, kinematic performances, dynamic performances, and utility libraries. Alongside the performance model, we are publishing an open-source toolkit to facilitate the use of this model. The main objective of this paper is to describe each main component, their connections, and how they can be used for simulation and research in practice. Finally, we analyzed the performance of OpenAP by comparing it with an existing performance model and sample flight data.

1. Introduction

Many current air traffic management studies are aimed at solving the challenges caused by the increasing demand for air transportation. In the likely future scenario wherein air transportation is continuously being utilized more, innovative strategies will have to be investigated to maintain the safety and efficiency of flights, while keeping up with the growing capacity demand. To evaluate different concepts, large-scale air traffic simulations with future air traffic density need to be undertaken. Though several commercial air traffic simulators exist, the growing demand for more transparency in research data and research tools has made the open-source option stand out. For example, BlueSky [1], an open-source air traffic simulator, has become popular in the air traffic management research community.

Among other things, the aircraft performance model constitutes a core part of traffic simulators. The domain of aircraft performance or flight mechanics studies characteristics of aircraft such as speed, altitude, thrust, drag, and fuel consumption. Based on the aircraft dynamic model, it provides methods with which to compute flight trajectories. Many air traffic management studies are dependent on the aircraft performance model. Currently, many of these studies depend on the proprietary model Base of Aircraft Data (BADA), including BADA 3 and BADA 4 [2,3]. No comparable open alternative to BADA

is available. One also developed by Eurocontrol, the General Aircraft Modelling Environment (GAME) is a kinematic-only performance model [4] that can be used for air transport research. Compared to BADA, it is less frequently used by air transport researchers. Previous research in [5] has produced a simplified aircraft performance model, named COALA, which uses part of the BADA model data. From the aviation industry, almost all aircraft manufacturers provide commercial performance data and tools for their aircraft. For example, Airbus has developed the Performance Engineering Programs (PEP) [6] that can be used as a standalone software tool. In addition, third-party commercial tools that model aircraft performance also exist. One of the examples is PIANO software [7].

Due to the proprietary nature of these models and tools though, users are limited by the strict license agreements, which makes it difficult to redistribute and to use the models commercially or openly. For open-source projects like BlueSky, lacking a performance model that can be redistributed would pose a challenge for sharing the tool as a complete open simulator. In earlier research, we built a preliminary performance model that is purely based on models from the literature [8]. However, the lack of up-to-date information led to the research completed in this paper.

In this paper, we present an open aircraft performance model, OpenAP, which can be used as an alternative to BADA for future air traffic management studies. It is a model produced by the combination of open data, literature models, and estimated parameters originating from open data. The processes of obtaining different model components have been previously published by the authors, which are discussed and summarized in later sections. Our final goal is to provide an open model that can be used, distributed, and modified freely under an open-source license. Additionally, a toolkit is to be published alongside this paper (available at: https://github.com/tudelft-cns-atm/openap). The OpenAP toolkit includes both the model data and the python code libraries. It thus allows quick access and model integration in potential applications.

The results presented in this paper largely benefit from the recently expanded adoption of automatic dependent surveillance-broadcast (ADS-B) and Enhanced Mode-S (EHS) technologies. A large quantity of flight data is transmitted by aircraft through these surveillance methods. This data can be freely received by researchers. With this open surveillance data, new opportunities for data-driven modeling emerge. Previous studies have proven the potential of the data-driven method using flight data. Some examples of these studies involve inferring aircraft mass [9,10], filtering and estimating flight states [11], modeling fuel consumption [12], and predicting the performance [13].

For instance, one of the data-driven components in OpenAP is the aircraft kinematic performance model, which is a result of our earlier study [14,15]. The kinematic model deals with parameters such as aircraft speed, altitude, vertical rate, and distances. These parameters can be directly observed from surveillance data. By deploying adequate data mining and estimation methods at different phases of flight, the kinematic performance can be accurately modeled. However, a data-driven approach is not suitable for most of the dynamic performance components, since there are often too many unknowns to solve given the limited set of equations and observables from flight data. The aircraft thrust and fuel flow are two examples. In these cases, we must address the modeling challenge based on engine performance instead of flight data. Literature models and other open data are used for thrust and fuel flow. In general, modeling the dynamic performance components (thrust, drag, and fuel flow) is a crucial component of the studies that led to the publishing of OpenAP.

In the following section of the paper, the details of the different OpenAP components are described. In Section 2, the basis of the OpenAP model and toolkit is explained. In Section 3, the aircraft and engine property model is discussed. In Sections 4 and 5, the kinematic and dynamic performance models are presented respectively. Other related utility libraries are shown in Section 6. In Section 7, examples and demonstrations of OpenAP are presented. In sections eight and nine, discussion and conclusions are presented.

2. OpenAP Building Blocks

The OpenAP model consists of four main components, which are the *aircraft properties*, *kinematic performances*, *dynamic performances*, and *utilities*. In each main component, there can be multiple sub-components. Figure 1 shows the structure of OpenAP model at the current stage.
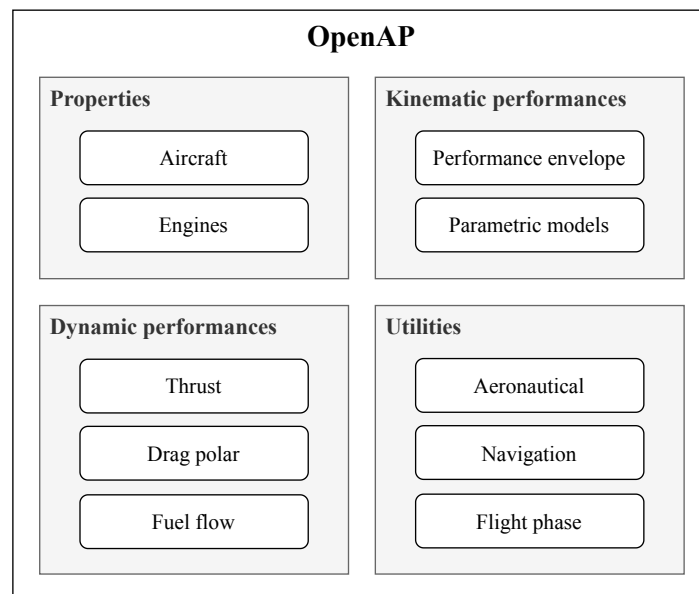

Figure 1. The structure of OpenAP.

In OpenAP, multiple formats exist to effectively store different types of data. The main file format is *YAML* format [16], which, according to its developer, is a human-friendly data serialization standard. It is primarily a key-value based document definition that can be easily interpolated by both humans and programming languages. For example, aircraft properties and drag polar models are presented using the YAML file format. YAML is effective for storing a key-value-like data structure but not efficient for larger tabular data. In these cases, the *fixed width text* files are used. They can be easily processed by computers while still maintaining readability for users. For example, the engine and kinematic performance databases are stored using fixed-width text files.

OpenAP aims to be a practical model with easy applicability. Hence, we have designed the toolkit alongside the model using one of the most common scientific programming languages—*Python*. The toolkit interacts with the model data and implements the equations for different calculations. It gives other high-level applications easy access to the OpenAP model. For example, one can obtain the aircraft configurations and parameters of a specific engine as shown in Appendix C.1.

3. Aircraft and Engine Properties

Following the previous example, the property module includes aircraft characteristics and engine configurations. It also provides performance data for a large number of engines. Currently, the most common aircraft types are included in this module. A list of supported aircraft is shown in Table A1 of the Appendix A . For each aircraft type, parameters such as dimensions, limits, nominal cruise conditions, and engine options are given. For example, the Airbus A320 aircraft parameters are shown in Appendix B.1.

The unit for dimensions and height in the configuration files is the meter, while the unit for weight is the kilogram. The aircraft data were gathered based on public data from aircraft manufacturers and the literature—for example, in [17,18].

Performance data on around 400 different engines are presented in the OpenAP engine databases (available at */openap/openap/data/engine* of the OpenAP repository). Each engine has 11 parameters that

are related to its performance. These parameters are indicated in Table 1. The first eight parameters are available for all engines. They are derived based on the ICAO aircraft engine emissions databank [19]. The last three parameters are only available for some engines; they were obtained from [20].

Table 1. Engine performance parameters

Parameter	Notation	Remarks
name	-	engine common identifier
manufacture	-	-
bpr	λ	bypass ratio
pr	-	pressure ratio
max_thrust	T_0	maximum static thrust, sea level (unit: N)
fuel_c3	C_{ff3}	fuel flow coefficient, 3rd order term (unit: kg/s)
fuel_c2	C_{ff2}	fuel flow coefficient, 2nd order term (unit: kg/s)
fuel_c1	C_{ff1}	fuel flow coefficient, 1st order term (unit: kg/s)
cruise_thrust	T_{cr}	thrust at the top of climb (unit: N)
cruise_mach	M_{cr}	cruise Mach number for the thrust condition
cruise_alt	h_{cr}	cruise Mach altitude for the thrust condition (unit: ft)

4. Kinematic Performance

The kinematic model studies the motion of the aircraft without considering the force acting on the aircraft. In OpenAP, the kinematic performance database is based on the results of a data-driven statistical model (WRAP) which was proposed in our previous study [15].

The WRAP model divides the flight into seven flight phases. It models different performance parameters at each flight phase accordingly. These performance indicators are acceleration, speed, vertical rate, altitude, and distance. Sorted by the flight phase, the parameters for kinematic performance are listed in Table 2.

By combining all the parameters, the kinematic module can describe the trajectory from takeoff until landing for the aircraft types as listed in Table A1.

The trajectory of the aircraft can be constructed using a set of ordinary differential equations. Under the International Standard Atmosphere (ISA) condition and without wind, the differential equations are expressed as:

$$\frac{dx}{dt} = V \sin \psi \tag{1}$$

$$\frac{dy}{dt} = V \cos \psi \tag{2}$$

$$\frac{dh}{dt} = V_S \tag{3}$$

$$\frac{dV}{dt} = a \tag{4}$$

where V represents the ground speed of the aircraft, which is the same as true airspeed without wind. It can be computed from either calibrated airspeed or Mach number depending on the flight phase.

The true airspeed can be computed from calibrated airspeed using the following equations:

$$V = \sqrt{\frac{7p}{\rho} \left[\left(1 + \frac{q_c}{p} \right)^{\frac{2}{7}} - 1 \right]} \tag{5}$$

$$q_c = p_0 \left[\left(1 + \frac{\rho_0}{7p_0} V_{cas}^2 \right)^{\frac{7}{2}} - 1 \right] \tag{6}$$

where q_c represents the impact pressure. In cases where the Mach number is given, the true airspeed can be calculated using temperature (τ) and speed of sound (a) as:

$$V = a_0 M \sqrt{\frac{\tau}{\tau_0}} \tag{7}$$

where ρ, p, and τ can be obtained from the ISA model if one knows the altitude of the aircraft. For cases other than ISA, deviations from ISA conditions, such as temperature, pressure, and wind, are considered to compute the actual ground speed. It is worth noting that in real operations, pressure, temperature, and wind can be derived based on aircraft Enhanced Mode S Surveillance (EHS) data and ADS-B data, which is described in detail in [21].

Table 2. Performance parameter in OpenAP kinematic model.

Takeoff	V_{lof}	liftoff speed	m/s
	d_{tof}	takeoff distance	km
	$\bar{a}_{\text{tof}}$	mean takeoff acceleration	m/s^2
Initial climb	$V_{\text{cas,ic}}$	calibrated airspeed	m/s
	$V_{S\text{ic}}$	vertical rate	m/s
Climb	$R_{\text{top,cl}}$	range to the top of climb	km
	$h_{\text{cas,cl}}$	constant CAS crossover altitude	km
	$V_{\text{cas,cl}}$	constant CAS	m/s
	$V_{S\text{cas,cl}}$	vertical rate during constant CAS climb	m/s
	$h_{\text{mach,cl}}$	constant Mach climb crossover altitude	km
	M_{cl}	constant Mach number	-
	$V_{S\text{mach,cl}}$	vertical rate at constant Mach climb	m/s
	$V_{S\text{precas,cl}}$	vertical rate before constant CAS climb	m/s
Cruise	R_{cr}	cruise range	km
	$h_{\text{init,cr}}$	initial cruise altitude	km
	h_{cr}	cruise altitude	km
	M_{cr}	cruise Mach number	-
Descent	$R_{\text{top,de}}$	range from the top of descent	km
	M_{de}	constant Mach number	-
	$h_{\text{mach,de}}$	constant Mach descent crossover altitude	km
	$V_{S\text{mach,de}}$	vertical rate at constant Mach descent	m/s
	$V_{\text{cas,de}}$	constant CAS	m/s
	$h_{\text{cas,de}}$	constant CAS crossover altitude	km
	$V_{S\text{cas,de}}$	vertical rate at constant CAS descent	m/s
	$V_{S\text{postcas,de}}$	vertical rate after constant CAS descent	m/s
Final approach	$V_{\text{cas,fa}}$	calibrated airspeed	m/s
	$V_{S\text{fa}}$	vertical rate	m/s
	$\angle_{\text{fa}}$	path angle	deg
Landing	V_{tcd}	touchdown speed	m/s
	d_{lnd}	braking distance	km
	$\bar{a}_{lnd}$	mean braking deceleration	m/s^2

Each parameter from the WRAP kinematic model is provided with a default value, minimum value, maximum value, and parametric statistical model. The minimum and maximum values provide the boundaries of the parameter, while the parametric statistical model can be used to sample values from a given probability density function (normal, gamma, or beta). With the OpenAP toolkit, all kinematic performance parameters can be queried using the functions defined in Appendix C.2. Each of the functions returns the default, minimum, maximum, and parameters to construct the probability density function that reflects the performance indicator. The model default value refers to the most common value for each parameter. The detail on obtaining these parameters can be found in [15].

5. Dynamic Performance

To further extend OpenAP capabilities, the dynamic module is included. In addition to speed, vertical rate, and distance, the dynamic module deals with more performance parameters, specifically, the mass and forces. The main challenge of constructing a complete dynamic model is the lack of open data on aircraft mass, thrust, and drag models. OpenAP accomplishes the goal by using models from the literature and open flight data. Another important part of the dynamic module is to express the fuel consumption of different engines during operation. To achieve this, the ICAO aircraft engine emissions databank was used to construct the fuel flow model.

5.1. Aircraft Dynamic Model

Aircraft motion in OpenAP is defined using a four-degrees-of-freedom *point-mass* model, which describes the aircraft translations in three perpendicular axes, and the roll rotation. In this point-mass model, the forces are assumed to be applied at the aircraft's center of gravity. The model can also be expressed with a set of ordinary differential equations. Assuming the ISA and zero wind, these equations can be written as:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = V \sin \psi \cos \gamma \tag{8}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = V \cos \psi \cos \gamma \tag{9}$$

$$\frac{\mathrm{d}h}{\mathrm{d}t} = V \sin \gamma \tag{10}$$

$$\frac{\mathrm{d}\psi}{\mathrm{d}t} = \frac{g \tan \phi}{V \cos \gamma} \tag{11}$$

$$\frac{\mathrm{d}V}{\mathrm{d}t} = \frac{T - D}{m} - g \sin \gamma \tag{12}$$

$$\frac{\mathrm{d}m}{\mathrm{d}t} = f_{\text{fuel}}(T, h) \tag{13}$$

The dynamic components introduced in these differential equations are mass, thrust, drag, and fuel flow. In Figure 2, forces acting on the aircraft in the dynamic model are illustrated.
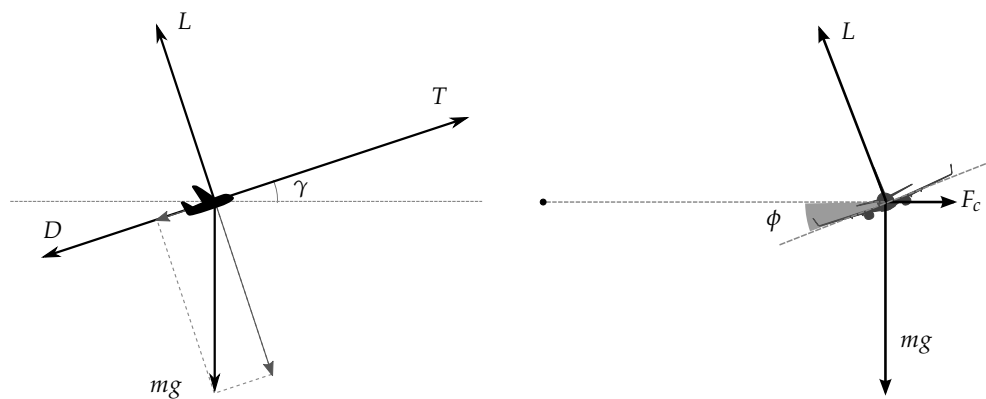

Figure 2. Forces acting on the aircraft in the point-mass dynamic performance model.

On the left-hand side of the figure, different forces are viewed from the side view. The right-hand side of the figure shows the forces during the banking from the front view. F_c refers to the centrifugal force, which is not explicitly expressed in the model.

The net thrust is expressed as the product throttle setting (δ_T) and the maximum thrust. Based on an empirical model proposed by [22], the maximum thrust of the aircraft (f_{thr}) is expressed as a function that is dependent on the aircraft altitude, speed, and vertical rate:

$$T = \delta_T \, f_{\text{thr}}(h, V, V_S) \tag{14}$$

The drag force of the aircraft in Equation (12) can be calculated knowing the dynamic pressure (q), the wing surface (S), and the drag coefficient (C_D):

$$D = C_D q S \tag{15}$$

$$q = \frac{1}{2}\rho V^2 \tag{16}$$

The fuel flow model used in Equation (13) is a function that is dependent on the net thrust of the aircraft's engines. It is constructed using the fuel flow coefficients proposed in the engine performance database that is shown in Table 1. The details of the thrust, drag, and fuel flow models are explained in the rest of this section.

5.2. Thrust

The thrust model included in OpenAP can calculate maximum takeoff thrust and maximum thrust separately during the flight. The key parameters that determine the maximum thrust of the turbofan engine are the maximum static thrust at sea level (T_0), the bypass ratio (λ), and the rated engine thrust at cruise. Most numerical coefficients in the thrust equations are obtained empirically from available performance data from multiple engines according to [22].

5.2.1. Takeoff Thrust

Considering the altitude of the runway, the takeoff thrust of an engine is modeled as:

$$\frac{T}{T_0} = A - \frac{0.377(1+\lambda)}{\sqrt{(1+0.82\lambda)G_0}}ZM + (0.23 + 0.19\sqrt{\lambda})XM^2 \tag{17}$$

where the coefficient of the parameter is calculated as follows:

$$G_0 = 0.0606\lambda + 0.6337 \tag{18}$$

$$A = -0.4327 \left(\frac{p}{p_0}\right)^2 + 1.3855\frac{p}{p_0} + 0.0472 \tag{19}$$

$$Z = 0.9106 \left(\frac{p}{p_0}\right)^3 - 1.7736 \left(\frac{p}{p_0}\right)^2 + 1.8697\frac{p}{p_0} \tag{20}$$

$$X = 0.1377 \left(\frac{p}{p_0}\right)^3 - 0.4374 \left(\frac{p}{p_0}\right)^2 + 1.3003\frac{p}{p_0} \tag{21}$$

5.2.2. Climb and Cruise Thrust

The climb is divided into three segments to compute the thrust. The segments are 0–10 k ft, 10–30 k ft, and 30–40 k ft. An additional key parameter is the engine thrust during the cruise at 30 k ft according to [22]. In OpenAP, a small modification is made so that the cruise altitude is not fixed at 30 k ft. Instead, the cruise altitude from the aircraft and engine property module is used.

(i)   For the en-route segment from 30 k ft to 40 k ft, the thrust ratio is modeled as follows:

$$\frac{T}{T_{\text{cr}}} = c_1 \ln \left(\frac{p}{p_{\text{cr}}}\right) + c_2 \tag{22}$$

where c_1 and c_2 are calculated as:

$$c_1 = -0.4204 \left(\frac{M}{M_{cr}} \right) + 1.0824$$

$$c_2 = \left(\frac{M}{M_{cr}} \right)^{-0.11}$$

(23)

(ii) For the segment from 10 k ft to 30 k ft, the thrust is derived similarly, with the exception that the reference calibrated airspeed is used instead of Mach number. The thrust can be approximated as:

$$\frac{T}{T_{cr}} = c_3 \left(\frac{p}{p_{cr}} \right)^{c_4}$$

(24)

where the coefficients c_3 and c_4 are calculated as follows:

$$c_3 = \left(\frac{V_{cas}}{V_{cas,cr}} \right)^{-0.1}$$

$$c_4 = -0.335 \left(\frac{V_{cas}}{V_{cas,cr}} \right) + c_5$$

(25)

In order to find the value of c_5, a look-up table is used in [22]. To simplify the computation, a linear approximation for c_5 is adopted in OpenAP as follows:

$$c_5 = 2.667 \times 10^{-5} \, V_S + 0.8633$$

(26)

where the unit of V_S is in m/s.

(iii) For the climbing segment from takeoff up to 10 k ft, a linear model is used to approximate the thrust ratio, with respect to the thrust ratio at 30 k ft. The relationship is as follows:

$$\frac{T}{T_{cr}} = c_6 \left(\frac{p}{p_{cr}} \right) + \left[\frac{T_{10}}{T_{cr}} - c_6 \left(\frac{p_{10}}{p_{cr}} \right) \right]$$

(27)

where T_{10} is the thrust at 10k ft. It can be computed by first using Equation (24). Coefficient c_6 is introduced to incorporate the variations in speeds and vertical rates. In [22], a look-up table is also used to find the value of c_6. In OpenAP, a bi-variate polynomial function is constructed to approximate this coefficient instead:

$$c_6 = -1.2043 \times 10^{-1} \left(\frac{V_{cas}}{V_{cas,cr}} \right) - 8.8889 \times 10^{-9} \, V_S^2$$

$$+ 2.4444 \times 10^{-5} \, V_S + 4.7379 \times 10^{-1}$$

(28)

where the units of V_{cas} and V_S are both in m/s.

5.2.3. Reference Cruise Thrust

One of the parameters used in the thrust model is the rated thrust at cruise altitude (T_{cr}). However, this information is not always published by aircraft or engine manufacturers. Available engine reference data are included in the engine performance (shown in Table 1). When T_{cr} is not available, we rely on

the empirical model proposed in [20], where the cruise thrust model is constructed for the maximum static thrust as:

$$T_{\text{cr}} = 0.2 \, T_0 + 890 \tag{29}$$

where the units of T_{cr} and T_0 are both in N.

Using the OpenAP toolkit, the maximum thrust of the aircraft during the flight can be conveniently computed following the example in Appendix C.3.

5.3. Drag

To compute the drag of aircraft, we first need to determine the drag coefficient. The drag coefficient is dependent on the drag polar and the lift. In our previous study [23,24], drag polar models for common aircraft types were estimated (data available at */openap/openap/data/dragpolar* of the OpenAP repository).

When an aircraft is airborne, the equations to calculate the drag coefficient are expressed as follows:

$$C_D = C_{D0} + kC_L^2 \tag{30}$$

$$C_L = \frac{L}{qS} = \frac{mg \cos \gamma}{qS} \tag{31}$$

where the zero-lift drag coefficient (C_{D0}) and lift-induced drag coefficient (k) are provided by OpenAP. In the OpenAP toolkit, these parameters at different flight phases are presented. An example of the drag polar model is shown in Appendix B.2.

Currently, the OpenAP model contains the drag polars for around 20 of the most common aircraft types. For illustration purposes, the drag polars for several common Airbus and Boeing aircraft types are visualized in Figure 3.
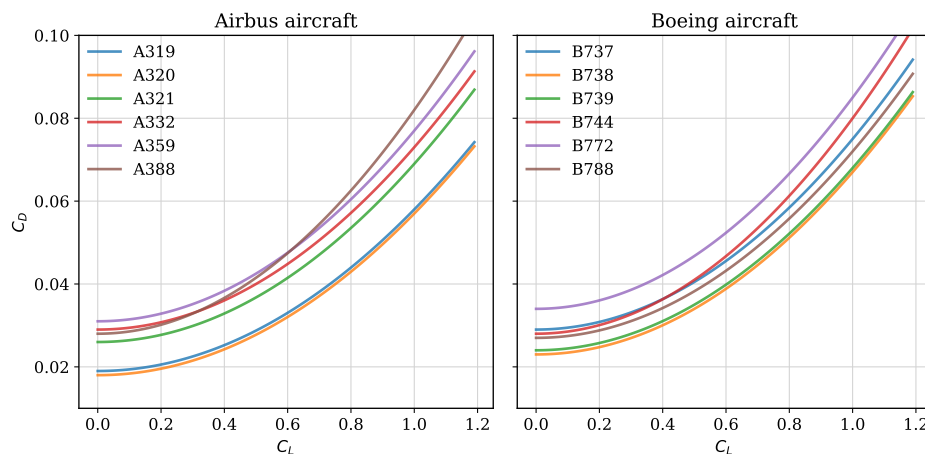

Figure 3. Drag polar estimated for common Airbus and Boeing aircraft under clean configurations at low Mach number.

5.4. Flaps

In addition to the drag polar of clean configuration, the OpenAP toolkit provides the model to compute the increase in drag coefficient under different flaps configurations. Using an empirical model ([25], p.109), this increase in drag coefficient due to flap deflection can be computed as:

$$\Delta C_{D,f} = \lambda_f \left(\frac{c_f}{c}\right)^{1.38} \left(\frac{S_f}{S}\right) \sin^2 \delta_f \tag{32}$$

where c_f/c and S_f/S are flap to wing chord ratio and flap to wing surface ratio respectively. δ_f is the flap deflection angle. When exact c_f/c and S_f/S are not available, they are both assumed to be 0.15, which is an empirical approximation based on existing aircraft data. λ_f is dependent on the type of flaps. According to [25], λ_f is set to be 1.7 for plain and split flaps and 0.9 for slotted flaps.

Changes in lift-induced drag coefficient due to flaps can also be modeled. The deflection of flaps affects the Oswald efficiency factor (e). Based on data from several existing aircraft, a linear relationship was found by [26]. The model to calculate such an increase in e due to flaps (denoted as Δe_f) is:

$$
\begin{aligned}
\Delta e_f &= 0.0046\,\delta_f \quad &\text{(for rear-mounted engines)} \\
\Delta e_f &= 0.0026\,\delta_f \quad &\text{(for wing-mounted engines)}
\end{aligned}
\tag{33}
$$

5.5. Compressibility

When an aircraft flies at a higher speed than critical Mach number, the compressibility effect should not be ignored. In OpenAP, the increase of drag coefficient due to wave drag $\Delta C_{D,w}$ is modeled as [27]:

$$
\Delta C_{D,w} = \begin{cases} 0 & M \le M_{\text{crit}} \\ 20\,(M - M_{\text{crit}})^4 & M > M_{\text{crit}} \end{cases}
\tag{34}
$$

where M_{crit} is the critical Mach number which is given by OpenAP. In Figure 4, the compressibility effect on drag polar for one aircraft is shown.
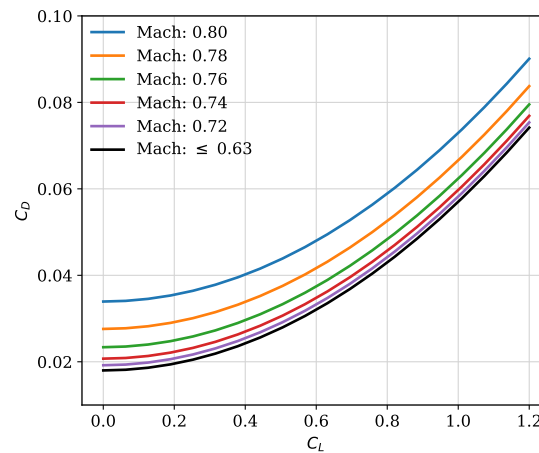

Figure 4. Changes of drag polar due to compressibility (A320).

5.6. Landing Gear

The landing gear adds a significant amount of drag to an aircraft when it is extended. It is retracted as soon as the aircraft becomes airborne and only extended shortly before landing. There are limited studies that quantify the drag coefficients of aircraft landing gear. In this paper, we adopted the model proposed by [28] to calculate the increased drag coefficient created by landing gear:

$$
\Delta C_{D,g} = \frac{W}{S} K_{\text{uc}}\, m_{\text{max}}^{-0.215}
\tag{35}
$$

where W/S is the wing loading, m_{max} refers to the maximum mass of an airplane, and K_{uc} is a factor that relates to the flap deflection angle. In principle, the value of K_{uc} is lower when more flap deflection is applied. This is because the flow velocity along the bottom of the wing decreases when flaps are deployed, which leads to a lower drag on the landing gear. For simplification, it is taken as 3.16×10^{-5} in this paper based on the data provided by [28].

5.7. Final Drag Polar Model

In summary, combining all the components discussed in this section, the zero-lift drag coefficient is calculated as:

$$C_{D0} = C_{D0,clean} + \Delta C_{D,f} + \Delta C_{D,g} + \Delta C_{D,w} \tag{36}$$

Additionally, the lift-induced drag coefficient is calculated as:

$$k = \frac{1}{1/k_{clean} + \pi A \Delta e_f} \tag{37}$$

Knowing the aircraft mass, airspeed, altitude, and flight path angle, the drag of the aircraft can be computed following the example in Appendix C.4.

5.8. Fuel Flow

The engine fuel flow model was constructed based on the public data obtained from the *ICAO aircraft engine emissions databank* [19]. Fuel flow indicators for most turbofan engines are included in this database. It defines fuel flow under four different conditions, which are takeoff, climb-out, approach, and idle. The power settings of the engine are defined at 100%, 85%, 30%, and 7% of engine maximum power respectively. Under the static sea-level testing condition, engine fuel flow data were obtained and published.

In order to create a model that can compute actual fuel consumption under different thrust, these four data points are fitted with a 3rd-degree polynomial model. The coefficients for the polynomial function are defined as C_{ff3}, C_{ff2}, and C_{ff1} (as shown in Table 1), with the unit of kg/s. The fuel flow profile defines the relationship between fuel flow and percentage of the maximum static thrust. Denoting $f_{fuel,SL}$ as the function representing fuel flow (kg/s) at sea level, the dynamic fuel flow model can be computed knowing the net thrust:

$$f_{fuel,SL}(T) = C_{ff3} \left(\frac{T}{T_0}\right)^3 + C_{ff2} \left(\frac{T}{T_0}\right)^2 + C_{ff1} \frac{T}{T_0} \tag{38}$$

where all the coefficients and maximum static thrust T_0 of each engine are given in the engine database of OpenAP. For the same amount of thrust, fuel flow increases with altitude. This difference can be seen by the thrust's specific fuel consumption (SFC) at sea-level and cruise conditions. In Figure 5, the differences for multiple engines are shown.
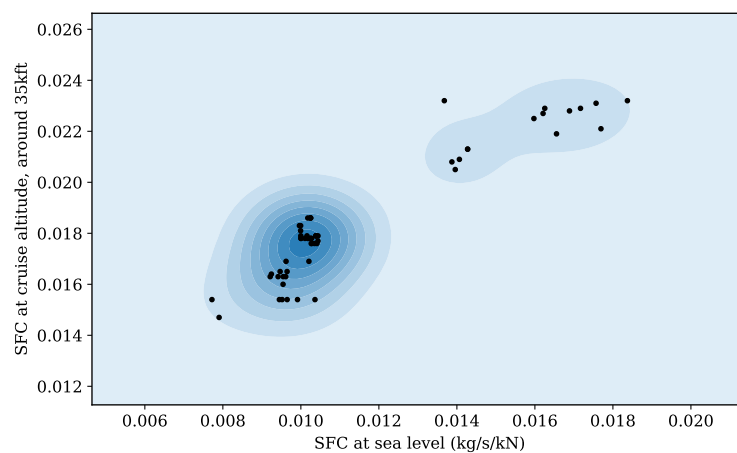

Figure 5. Differences in specific fuel consumption at sea level and cruise altitude.

Based on the limited data on SFC at sea level and cruise altitude, we propose a simplified linear correction factor ($C_{\text{ff,ch}}$) that takes into consideration the altitude effect. When SFC is available at the cruise altitude, the correction factor is calculated as:

$$C_{\text{ff,ch}} = \frac{\text{SFC}_{\text{CR}} - \text{SFC}_{\text{SL}}}{h_{\text{CR}}}$$
$$= \frac{\text{SFC}_{\text{CR}} - f_{\text{fuel,SL}}(T_0)/T_0}{h_{\text{CR}}} \qquad \left(\text{unit:} \frac{kg}{s \times kN \times m}\right) \tag{39}$$

where the h_{CR} is the cruise altitude for the corresponding SFC.

Figure 6 illustrates the coefficient for engines with available SFC_{CR}. The mean value was found to be around 6.7×10^{-7}, which is used as a default value for engines where SFC_{CR} is not available.
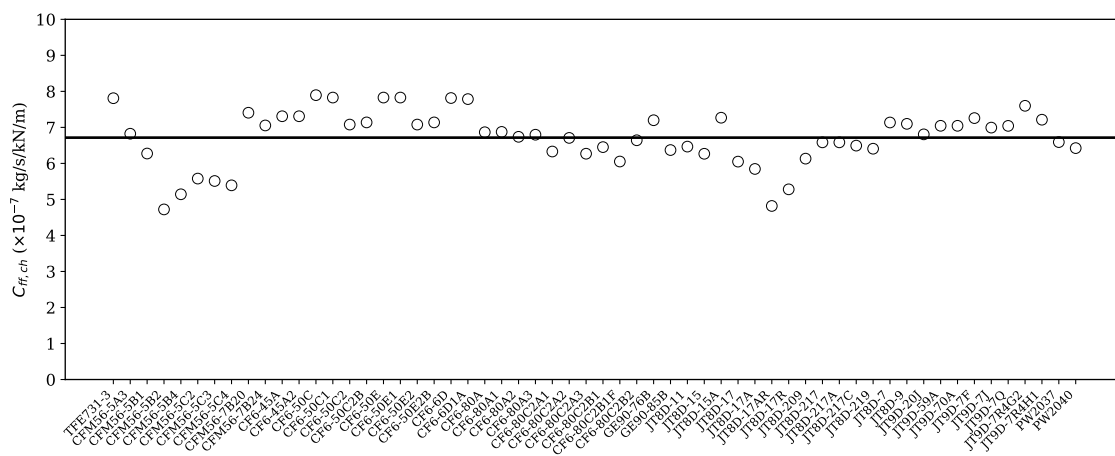

Figure 6. Difference in specific fuel consumption at sea level and cruise altitude.

Combining the fuel flow at sea level and the fuel flow with altitude correction, the dynamic fuel flow can be computed. The final fuel flow model is expressed as:

$$f_{\text{fuel}}(T, h) = f_{\text{fuel,SL}}(T) + C_{\text{ff,ch}} \times T \times h$$
$$= C_{\text{ff3}} \left(\frac{T}{T_0}\right)^3 + C_{\text{ff2}} \left(\frac{T}{T_0}\right)^2 + C_{\text{ff1}} \frac{T}{T_0} + C_{\text{ff,ch}} \times T \times h \tag{40}$$

In OpenAP, the fuel flow can be computed under three conditions: (1) with a known net thrust of all engines; (2) during the takeoff with a known thrust setting and an unknown aircraft mass; (3) during the flight, knowing the mass, speed, altitude, and flight path angle. In general, when net thrust is not known, it can be calculated based on the dynamic model (Equation (12)). Examples of calculating the fuel flow are given in Appendix C.5.

6. Other Databases and Utility Libraries

Besides the core components explained in the earlier sections, OpenAP also provides several utility libraries to facilitate calculations related to aircraft performance.

6.1. Flight Phase

Based on our earlier research [29], the flight phase library enables the easy use of the fuzzy logic flight phase identification algorithm. Providing the trajectory (with time, altitude, speed, and vertical rate), one can obtain corresponding flight phases for the entire trajectory. The flight data are labeled with the ground, climb, cruise, descent, and leveling phases.

Functions in the flight phase library also enable the extraction of specific flight segments among all available flight phases, which are takeoff, initial climb, climb, cruise, descent, final approach, and landing. Examples are shown in Appendix C.6.

6.2. Aeronautical Calculations

Derived from the open-source BlueSky simulator, the aeronautical library provides common calculations related to aircraft performance. For example, this library provides the functions to compute distance and bearing. It is also able to calculate the standard atmospheric conditions at different flight levels. In addition, speed conversions among calibrated airspeed (CAS), true airspeed (TAS), and Mach number (e.g., Equations (5) and (7)) are easily made with this library.

6.3. Navigation Database

The navigation database provides information on airport and navigation data (available at */openap/openap/data/nav* of the OpenAP repository). The data in the navigation database are based on the open data published by X-Plane [30]. The navigation library is designed to give easy access to these datasets. For example, the airport location can be queried using its ICAO code. One can also use this library to get the closest airport or fixes of a specific location. Examples are shown in Appendix C.7.

7. Analysis of OpenAP

In this section, analyses are conducted to examine the OpenAP model. Firstly, maximum thrust profiles computed using OpenAP and BADA 3 are compared. Then, using a test flight from the TU Delft Cessna Citation aircraft, both thrust and fuel flow from OpenAP are compared to the onboard data. In addition, the same values are computed from the BADA model to serve as a comparison.

7.1. Trajectory Generation

The OpenAP toolkit comes with a kinematic trajectory generator that can be used to quickly generate a large number of flight trajectories. The generator creates trajectories based on the WRAP kinematic model (Equations (1)–(4)). Users can specify the initial conditions and time step, or leave them to the program. When left to the program, the performance parameter values can be requested as the default values or random values to be drawn from the corresponding probability density function. When a complete trajectory needs to be generated, speed profiles of the climb, cruise, and descent phases are coupled based on the cruise altitude and Mach number. An example of this trajectory generation method is explained in Appendix C.8.

In the following use case, several simple flight trajectories are generated. In Figures 7 and 8, the vertical profiles of a number of generated climbing and descent trajectories are shown. The default (most common) profile is indicated in red, while a set of randomly generated profiles is shown in gray.
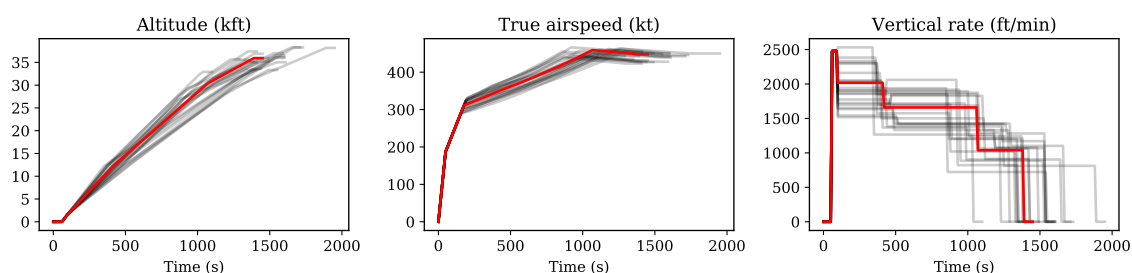

Figure 7. Generated climbing trajectory (Airbus A320).
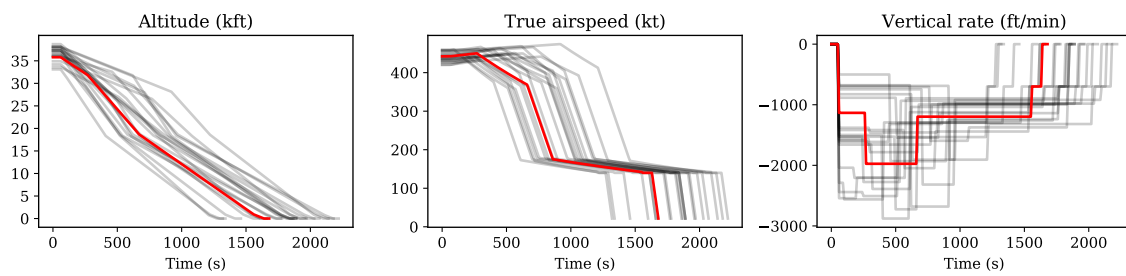
Figure 8. Generated descent trajectory (Airbus A320).

Based on the noise characteristics from ADS-B, the trajectory generator also allows simulated observation noise to be added to generated trajectories. In Figure 9, one such example is shown as a complete trajectory with Gaussian noise added.
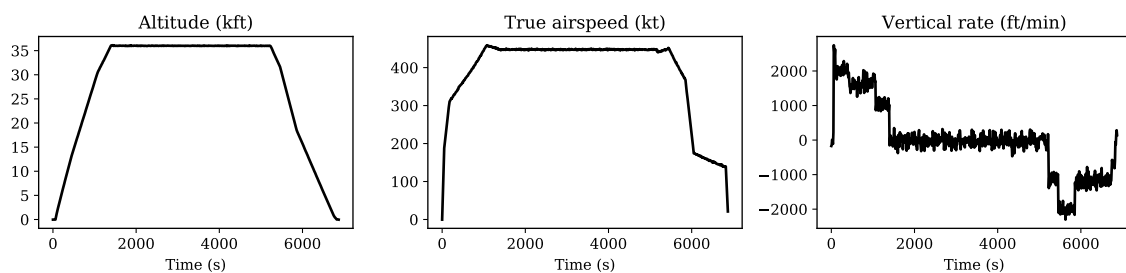

Figure 9. Generated complete trajectory with noise added (Airbus A320).

7.2. Comparison with BADA

7.2.1. Thrust Model

To demonstrate the difference between OpenAP and BADA, we computed the maximum thrust of an aircraft (Airbus A320 with V2500-A1 engines) under different flight conditions. The results are shown in Figure 10. The maximum thrust was computed using different combinations of altitude and speed. OpenAP and BADA results are illustrated in red and blue respectively.
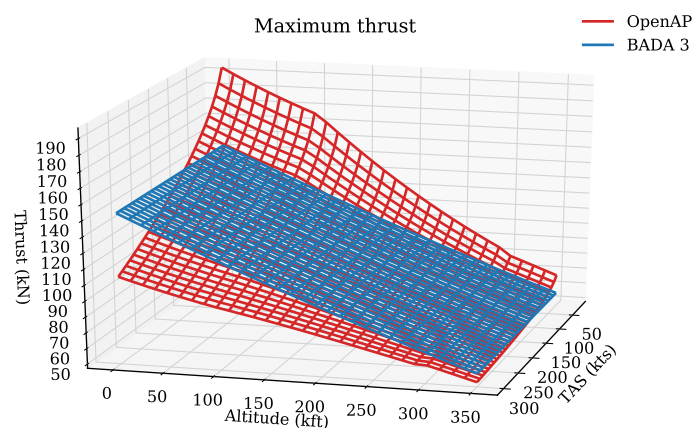

Figure 10. A320 maximum thrust (V2500-A1 engine).

A large difference is visible when the aircraft is at low altitude. The difference between OpenAP and BADA becomes smaller when the aircraft is flying at a higher altitude. One of the major reasons for the difference is that BADA 3 does not model thrust dynamic due to the change of airspeed. In contrast, OpenAP shows a clear nonlinear relationship between aircraft speed and thrust. The performance is more realistic for a lower altitude at a lower speed.

7.2.2. Drag Polar

Due to the restrictions associated with the BADA license, exposure of specific BADA coefficients is not permitted. Instead of showing the exact difference between BADA and our model for each aircraft type, the overall statistics for all available aircraft are shown in Figure 11. The mean absolute differences for C_{D0} and k were both found to be around 0.007.
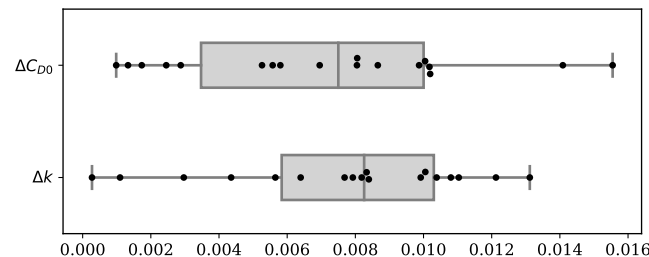

Figure 11. Absolute difference of drag polar between BADA (3.12) and the model derived in this paper under clean configuration.

This gives a good indication of what the level of difference in the outcome of this study is, compared to well-established performance models. However, we should be cautious when interpreting this difference as the estimation accuracy. Since neither the method nor the data used for constructing BADA drag polar are published, it is hard to identify the uncertainties in the BADA model. Thus, the difference between the two models should not be directly used to quantify the accuracy of our model.

7.2.3. Fuel Flow Model

The main difference between OpenAP and BADA fuel flow models is the input parameters. The inputs for BADA fuel flow calculation are thrust and speed, while the inputs for OpenAP fuel flow calculation are thrust and altitude.

We can visually compare the inputs and outputs of both fuel flow models in Figure 12, which shows the different variability of the two fuel flow models regarding different input parameters.
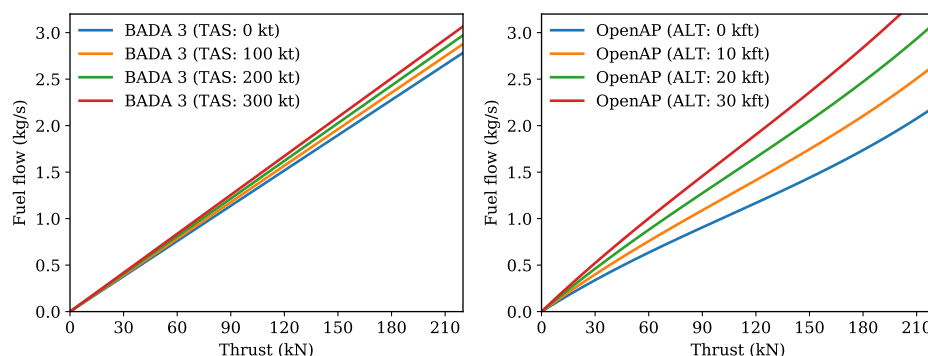

Figure 12. A320 maximum thrust (V2500-A1 engine).

It can be seen that the performances of both the fuel flow model are similar at a lower thrust (for example, during the cruise). However, while the aircraft is flying with a higher thrust (for example, during the climb), the OpenAP fuel flow model is corrected concerning the altitude of the aircraft. It is also worth emphasizing that the inputs for both models are different. Thus, a better way to evaluate fuel flow estimations has to be based on real flight trajectories, which is shown in the following section.

7.3. Analysis Based on Flight FMS Data

To further evaluate the thrust and fuel flow models, we used a real flight to examine the performance of OpenAP. The flight data were gathered from the TU Delft Cessna Citation II flight data recorder. The flight data contained states such as position, altitude, speed, vertical rate, acceleration, and fuel flow, and the takeoff mass of the aircraft.

Based on the onboard data, we could reconstruct the true net thrust of the engine during the flight based on drag and acceleration. This was to be used as a reference when examining the thrust model. In Figure 13, the maximum thrust computed from OpenAP and BADA are shown alongside the net thrust computed from flight data.
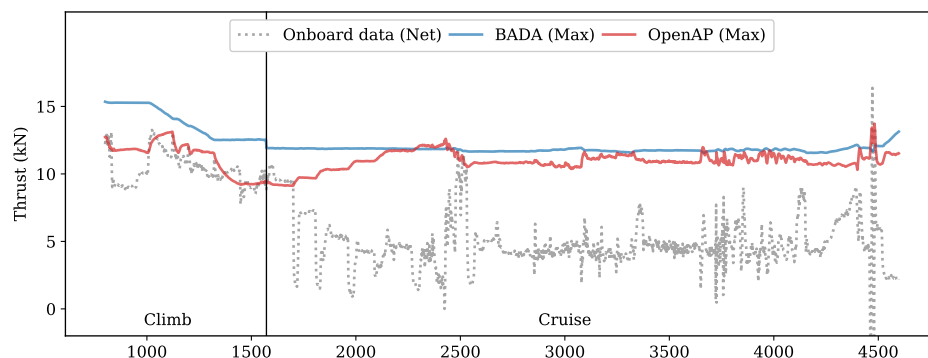

Figure 13. Maximum thrust profile.

The flight recorder data also include the fan rotation speed (N1), which is expressed as a percentage of the maximum rotation speed. By combining the N1 value and the maximum thrust computed from OpenAP, we were able to derive the estimated net thrust during the flight. The same method was also applied to the BADA model to serve as a comparison. In Figure 14, the estimated net thrusts are shown against the true net thrust. The distribution of estimation error based on BADA and OpenAP is also shown in this figure.
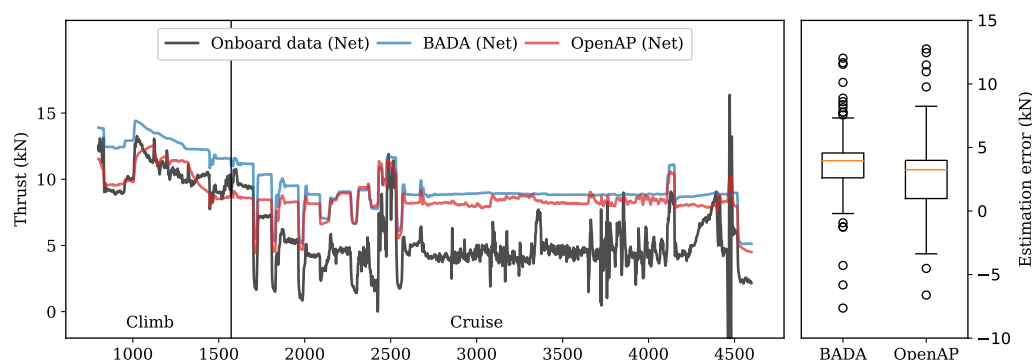

Figure 14. Estimated net thrust based on maximum thrust profile and N1 settings.

It is noticeable that during the climb, the open thrust model can approximate the thrust with the same level accuracy or better than BADA. During the cruise, a relatively large difference existed for both models. It is important to note that this large difference could be due to the nonlinear relationship between the fan rotation speed and the actual throttle setting. It is also worth noting the difference between this practice flight and a normal passenger flight. Firstly, the aircraft was cruising at a lower altitude (around 10,000 ft) instead of the optimal altitude (commonly above 30,000 ft). Secondly, the flight procedure was designed to test the limits of flight performance. Thus, drastic changes in speed and altitude existed during the flight. These factors would contribute to the difference between

true thrust and estimated thrust using either OpenAP or BADA. Nonetheless, we can see similar estimation results emerge using OpenAP and BADA.

Using the recorded engine fuel flow data, the fuel flow model was evaluated. In Figure 15, the fuel flow is shown based on the the net thrust calculated based on onboard data. Results from OpenAP and BADA are illustrated alongside the actual fuel flow. The estimation error distributions of fuel flow using BADA and OpenAP are also shown.
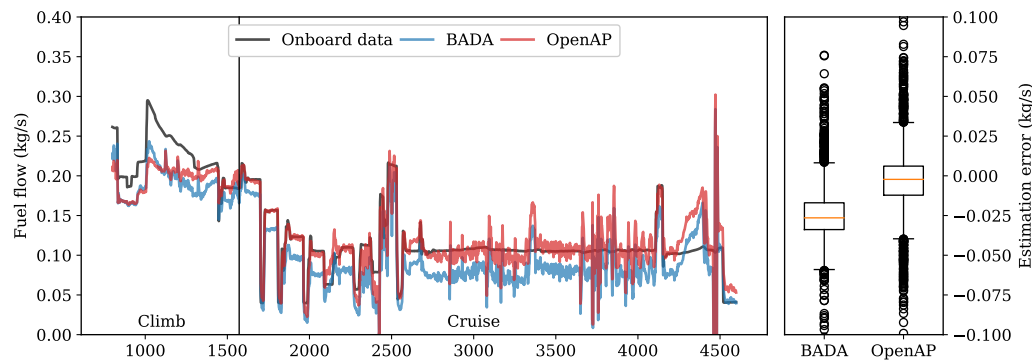

Figure 15. Estimated fuel flow based on the net thrust.

Overall, OpenAP displays slightly better accuracy than BADA. It is also noticeable that the difference is higher when thrust is higher (during the climb). This can be seen as a reflection of the result that has been found in Figure 12.

8. Discussions

The OpenAP model and its associated toolkit are still under development. In this section, we address the limitations and plans for this aircraft performance model.

8.1. Limitations

As mentioned in the earlier sections, the thrust model included in OpenAP is based on the studies of [20,22], where only turbofan engines are modeled. Hence, OpenAP does not yet cover aircraft that are equipped with other engine types.

The thrust and fuel models used in OpenAP are based on external sources. Hence, the uncertainty in these third-party models and data affects the accuracy of the OpenAP model. Limited access to accurate flight data from a wider range of aircraft types also limits the validation of OpenAP.

As we are relying on a Bayesian estimation method, this creates uncertainty in the drag polar values described in this paper. There is also a certain level of uncertainty inherited from the thrust models and data, which is explained in [23].

Currently, only the most common civil aircraft types are covered in OpenAP. This is due to the availability of flight data. Though the aircraft diversity may be a limiting factor for some applications, the data-driven approach used to construct OpenAP allows more aircraft to be included with the growing volume of flight data in the future.

8.2. Future Development

The current research plan for OpenAP focuses on enhancing the models and providing more functionalities that can be beneficial for air transport research. Our current research efforts include:

- OpenAP Emission: Research is being conducted to incorporate an open emission model for emission and climate-related air transport studies.

- OpenAP Optimization: Based on the dynamic model, a trajectory optimization model shall be developed. This will further enhance the trajectory generation capability of OpenAP, extending from kinematic models to dynamic models.
- WRAP model: As more ADS-B data is collected, we plan to update the WRAP kinematic model every few years to include new aircraft types, as well as to improve models for less common aircraft types.

8.3. Sharing and Community Contributions

OpenAP model data and source code for the toolkit are published under the GNU GPL v3 license. This allows anyone to freely use (including commercial and private use), distribute, and change the content. The advantage of having such an open license is to allow easy access and encourage contributions from the air traffic management research community. The source code is published on GitHub, where anyone can take part in the future development of the model.

Part of OpenAP is derived from flight data based on our previous studies, for example, the drag polar and kinematic models. However, we welcome more accurate information to be submitted, as long as the data can be open and have been obtained from legitimate sources.

Currently, there are a limited number of flight test data included in the OpenAP repository as part of the open flight datasets. They are from the TU Delft Cessna Citation II aircraft. We welcome the input of flight test data from other aircraft types. The flight identifier can be anonymized. However, it would be ideal to maintain aircraft and engine types, airspeed, aircraft mass, and fuel flow information in the flight data.

9. Conclusions

In this paper, we explain the details of a new open aircraft performance model, OpenAP. The OpenAP model, which is the combined outcome of several of our previous studies, is aimed at providing the air traffic management research community an open alternative to current closed-source aircraft performance models. Accompanied by an off-the-shelf user library written in the Python programming language, the performance model consists of four major components, which are the aircraft and engine properties, kinematic performance, dynamic performance, and utility libraries.

The characteristics of 27 common aircraft and 400 turbofan engines have been included in the OpenAP properties library. The aircraft database provides basic information on aircraft dimensions, limits, and engine configurations. The engine database provides information such as maximum thrust, bypass ratio, cruise performance, and fuel flow coefficients for each engine type. The kinematic performance is produced by our previous data-driven model (WRAP). It describes parameters that are related to speed, vertical rate, distance, and height in seven distinct flight phases. The OpenAP toolkit enables easy access to these parameters. The dynamic performance module provides models that describe engine maximum thrust, fuel consumption, and drag polar. It allows OpenAP to be used independently for studies that require flight dynamics.

To provide more options for future air traffic management studies, future research on extending OpenAP will be focused on aircraft emission models and noise models. We also plan to include the trajectory optimization and prediction functionalities in the OpenAP toolkit. To promote open science and collaborations, all OpenAP model data and the toolkit source code are published on GitHub under the GNU GPL license, where future contributions and requests are welcomed.

Nomenclature

x	downrange position (m)
y	cross-range position (m)
h	altitude (m)
h_{cr}	reference cruise altitude (ft)
a	acceleration (m/s^2)
g	gravitational acceleration (m/s^2)
V	aircraft speed (m/s)
V_S	vertical speed (m/s)
ψ	heading angle (deg)
γ	flight path angle (deg)
ϕ	bank angle (deg)
δ_T	thrust setting (-)
T	net thrust (N)
T_0	maximum static thrust, sea level (N)
T_{10}	thrust at altitude of 10,000 feet (N)
T_{cr}	thrust at the top of climb (N)
D	total drag (N)
L	total lift (N)
m	aircraft mass (kg)
q	dynamic pressure (Pa)
q_c	impact pressure (Pa)
p_0	pressure at sea level (Pa)
p_{10}	pressure at altitude of 10,000 feet (Pa)
p	pressure (Pa)
p_{cr}	pressure at reference cruise altitude (Pa)
ρ	air density (kg/m^3)
ρ_0	air density at sea level (kg/m^3)
τ	air temperature (K)
τ_0	air temperature at sea level (K)
M	Mach number (-)
M_{cirt}	critical Mach number (-)
M_{cr}	reference cruise Mach number (-)
C_D	drag coefficient (-)
C_L	lift coefficient (-)
C_{D0}	zero-lift drag coefficient (-)
$C_{D,w}$	wave drag coefficient (-)
C_{ff}	fuel flow coefficient (-)
k	lift-induced drag coefficient (-)
e	Oswald factor (-)
S	wing area (m^2)
λ	bypass ratio (-)

Appendix A. Available Aircraft Types

Table A1. Current available aircraft types in OpenAP.

Type	Aircraft Model	Engine Options
A319	Airbus A319	CFM56-5B5, CFM56-5B6, CFM56-5A4, CFM56-5A5, CFM56-5B7, V2522-A5, V2524-A5, V2527M-A5
A320	Airbus A320	CFM56-5-A1, CFM56-5-A1, CFM56-5A3, CFM56-5B4, CFM56-5B5, CFM56-5B6, V2500-A1, V2527-A5, V2527E-A5
A321	Airbus A321	CFM56-5B1, CFM56-5B2, V2530-A5, CFM56-5B3, CFM56-5B1, CFM56-5B2, V2533-A5, V2530-A5
A332	Airbus A330-200	CF6-80E1A2, CF6-80E1A4, CF6-80E1A3, PW4168A, PW4170, Trent 772
A333	Airbus A330-300	CF6-80E1A2, CF6-80E1A4, CF6-80E1A3, PW4164, PW4168, PW4168A, Trent 768, Trent 772, Trent 772
A343	Airbus A340-300	CFM56-5C2, CFM56-5C3, CFM56-5C4
A359	Airbus A350-900	Trent XWB-84
A388	Airbus A380-800	Trent 970-84, Trent 972-84, GP7270
B734	Boeing 737-400	CFM56-3B-2, CFM56-3C-1
B737	Boeing 737-700	CFM56-7B20, CFM56-7B22, CFM56-7B24, CFM56-7B26, CFM56-7B27
B738	Boeing 737-800	CFM56-7B24, CFM56-7B26, CFM56-7B27
B739	Boeing 737-900	CFM56-7B24, CFM56-7B26, CFM56-7B27
B744	Boeing 747-400	PW4062, CF6-80C2B1F, RB211-524G
B748	Boeing 747-8	GEnx-2B67
B752	Boeing 757-200	PW2037, RB211-535E4
B763	Boeing 767-300	JT9D-7R4D, PW4056, CF6-80C2B2
B772	Boeing 777-200ER	GE90-94B, PW4077, Trent 895
B773	Boeing 777-300	GE90-110B1, PW4090, Trent 892
B77W	Boeing 777-300ER	GE90-115B
B788	Boeing 787-8	GEnx-1B70, GEnx-1B67, GEnx-1B64, Trent 1000-E2, Trent 1000-C2, Trent 1000-A1
B789	Boeing 787-9	GEnx-1B75, GEnx-1B74, Trent 1000-K2, Trent 1000-J2, Trent 1000-A2
C550	Cessna Citation II	JT15D-4
E145	Embraer ERJ145 (LR)	AE3007A1
E170	Embraer E170	CF34-8E5, CF34-8E6
E190	Embraer E190 (LR)	CF34-10E5
E195	Embraer E195 (LR)	CF34-10E5
E75L	Embraer E175 (LR)	CF34-8E5, CF34-8E6

Appendix B. OpenAP Sample Data

Appendix B.1. Aircraft Property Data Structure

Data location: */openap/openap/data/aircraft/a320.yml*

```
aircraft: Airbus A320

fuselage:
  length:   37.57
  height:   4.14
  width:    3.95

wing:
  area:     124
  span:     35.8
  mac:      4.29
  sweep:    25
  t/c:      null

flaps:
  type:     single-slotted
  area:     21.1
  bf/b:     0.780

limits:
  MTOW:     78000
  MLW:      66000
  OEW:      42600
  MFC:      24210
  MMO:      0.82
  ceiling:  12500

cruise:
  height:   11000
  mach:     0.78

engine:
  type: turbofan
  mount: wing
  number: 2
  default: CFM56-5A3
  options:
    A320-111: CFM56-5-A1
    A320-211: CFM56-5-A1
    A320-212: CFM56-5A3
    ...
   ...
```

Appendix B.2. Drag Polar Data Structure

Data location: */openap/openap/data/dragpolar/a320.yml*:

```
aircraft: Airbus A320

cd0:
  clean:      0.018
  initclimb:  0.020
  finalapp:   0.024

k:
  clean:      0.039
  initclimb:  0.036
  finalapp:   0.034

e:
  clean:      0.798
  initclimb:  0.850
  finalapp:   0.902

gears:        0.017
mach_crit:    0.632
```

Appendix C. OpenAP Python Library Programming Interface Examples

The example in this paper reflects only the current version of the OpenAP Python library. Changes can be expected in the future.

Appendix C.1. Access OpenAP Aircraft and Engines

```
from openap import prop

aircraft = prop.aircraft('A320')
engine = prop.engine('CFM56-5B4')

print(aircraft['wing']['span'])
print(engine['max_thrust'])
```

Appendix C.2. Access WRAP Parameters

```
from openap import WRAP

wrap = WRAP(ac='A320')

param = takeoff_speed()
param = takeoff_distance()
param = takeoff_acceleration()
param = initclimb_vcas()
param = initclimb_vs()
param = climb_range()
param = climb_const_vcas()
param = climb_const_mach()
param = climb_cross_alt_concas()
param = climb_cross_alt_conmach()
param = climb_vs_pre_concas()
param = climb_vs_concas()
param = climb_vs_conmach()
param = cruise_range()
param = cruise_alt()
param = cruise_init_alt()
param = cruise_mach()
param = descent_range()
param = descent_const_mach()
param = descent_const_vcas()
param = descent_cross_alt_conmach()
param = descent_cross_alt_concas()
param = descent_vs_conmach()
param = descent_vs_concas()
param = descent_vs_post_concas()
param = finalapp_vcas()
param = finalapp_vs()
param = landing_speed()
param = landing_distance()
param = landing_acceleration()
```

Appendix C.3. Calculate Aircraft Thrust

```
from openap import Thrust

thrust = Thrust(ac='A320', eng='CFM56-5B4')

T = thrust.takeoff(tas=100, alt=0)
T = thrust.climb(tas=200, alt=20000, roc=1000)
T = thrust.cruise(tas=230, alt=32000)
```

where the inputs of the function are true airspeed (in knots), altitude (in feet), and rate of climb (in feet/minute). The outputs are maximum thrust forces (in Newton).

Appendix C.4. Calculate Aircraft Drag

```
from openap import Drag

drag = Drag(ac='A320')

D = drag.clean(mass=60000, tas=200, alt=20000)

# higher drag due to compressibility
D = drag.clean(mass=60000, tas=250, alt=20000)

# increase of drag due to flaps
D = drag.nonclean(mass=60000, tas=150, alt=1000,
  flap_angle=20, path_angle=10, landing_gear=False)

# increase of drag due to the landing gear
D = drag.nonclean(mass=60000, tas=150, alt=200,
  flap_angle=20, path_angle=10, landing_gear=True)
```

The inputs of these functions are mass (in kilogram), true airspeed (in knots), and altitude (in feet). The outputs are the drag force (in Newton).

Appendix C.5. Calculate Aircraft Fuel Flow

```
from openap import FuelFlow

fuel = FuelFlow(ac='A320', eng='CFM56-5B4')

FF = fuel.at_thrust(acthr=50000, alt=0)
FF = fuel.at_thrust(acthr=50000, alt=20000)
FF = fuel.enroute(mass=60000, tas=200, alt=20000, path_angle=3)
FF = fuel.enroute(mass=60000, tas=230, alt=32000, path_angle=0)
FF = fuel.takeoff(tas=100, alt=0, throttle=1)
```

The inputs of these functions are thrust (in Newton), mass (in kilogram), true airspeed (in knots), altitude (in feet), flight path angle (in degrees), and throttle setting (between 0 and 1).

Appendix C.6. Inferring Flight Phase

```
from openap import FlightPhase

FP = FlightPhase()
FP.set_trajectory(ts, alt, spd, roc)

labels = FP.phaselabel()                # produce the phase label for each data point
indices = FP.flight_phase_indices()     # return the set of indices each phase starts
```

where we first set the trajectory data. The last two functions produce the flight phase labels and the indices of the data point when a different flight phase starts.

Appendix C.7. Use of Navigation Library

```
from openap import nav

print(nav.airport('EHAM'))
print(nav.closest_airport(52.011, 4.357))

print(nav.fix('EH155'))
print(nav.closest_fix(52.011, 4.357))
```

Appendix C.8. Generation of Flight Trajectories

```
from openap.traj import Generator

trajgen = Generator(ac='a320')

trajgen.enable_noise()   # enable Gaussian noise in trajectory data

data_cl = trajgen.climb(dt=10, random=True)  # using random paramerters
data_cl = trajgen.climb(dt=10, cas_const_cl=280, mach_const_cl=0.78, alt_cr=35000)

data_de = trajgen.descent(dt=10, random=True)
data_de = trajgen.descent(dt=10, cas_const_de=280, mach_const_de=0.78, alt_cr=35000)

data_cr = trajgen.cruise(dt=60, random=True)
data_cr = trajgen.cruise(dt=60, range_cr=2000, alt_cr=35000, m_cr=0.78)

data_all = trajgen.complete(dt=10, random=True)
data_all = trajgen.complete(dt=10, alt_cr=35000, m_cr=0.78,
                            cas_const_cl=260, cas_const_de=260)
```

References

1. Hoekstra, J.M.; Ellerbroek, J. BlueSky ATC Simulator Project: An Open Data and Open Source Approach. In Proceedings of the 7th International Conference on Research in Air Transportation, Philadelphia, PA, USA, 20–24 June 2016; pp. 1–8.

2. Gallo, E.; Navarro, F.A.; Nuic, A.; Iagaru, M. Advanced aircraft performance modeling for ATM: BADA 4.0 results. In Proceedings of the 2006 IEEE/AIAA 25th Digital Avionics Systems Conference, Portland, OR, USA, 15–19 October 2006; pp. 1–12.

3. Nuic, A.; Poles, D.; Mouillet, V. BADA: An advanced aircraft performance model for present and future ATM systems. *Int. J. Adapt. Control Signal Process.* **2010**, *24*, 850–866. [CrossRef]

4. Calders, P. *G.A.M.E. Aircraft Performance Model Description*; DIS/ATD Unit, DOC.CoE-TP-02002; Eurocontrol: Brussels, Belgium, 2002.

5. Rosenow, J.; Fricke, H. *Flight Performance Modeling to Optimize Trajectories*; Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV: Bonn, Germany, 2016.

6. Airbus. Getting to grips with Aircraft Performance. *Flight Oper. Support Line Assist.* **2002**, *2*, 11–16.

7. Simos, D. *PIANO: PIANO User's Guide Version 4.0*; Lissys Limited: London, UK, 2004.

8. Metz, I.; Hoekstra, J.; Ellerbroek, J.; Kügler, D. Aircraft performance for open air traffic simulations. In Proceedings of the AIAA Modeling and Simulation Technologies Conference, San Diego, CA, USA, 4–8 January 2016; p. 3522.

9. Alligier, R.; Gianazza, D.; Durand, N. Ground-based estimation of the aircraft mass, adaptive vs. least squares method. In Proceedings of the ATM 2013, 10th USA/Europe Air Traffic Management Research and Development Seminar, Chicago, IL, USA, 10–13 June 2013; p. 1.

10. Sun, J.; Ellerbroek, J.; Hoekstra, J. Aircraft initial mass estimation using Bayesian inference method. *Transp. Res. Part Emerg. Technol.* **2018**, *90*, 59–73, [CrossRef]

11. Sun, J.; Blom, H.A.P.; Ellerbroek, J.; Hoekstra, J.M. Aircraft Mass and Thrust Estimation Using Recursive Bayesian Method. In Proceedings of the 8th International Conference on Research in Air Transportation, Barcelona, Spain, 26–29 June 2018.

12. Chati, Y.S.; Balakrishnan, H. A Gaussian process regression approach to model aircraft engine fuel flow rate. In Proceedings of the ACM 8th International Conference on Cyber-Physical Systems, Pennsylvania, PA, USA, 18–20 April 2017; pp. 131–140.

13. Hrastovec, M.; Solina, F. Prediction of aircraft performances based on data collected by air traffic control centers. *Transp. Res. Part Emerg. Technol.* **2016**, *73*, 167–182. [CrossRef]

14. Sun, J.; Ellerbroek, J.; Hoekstra, J. Modeling aircraft performance parameters with open ads-b data. In Proceedings of the 12th USA/Europe Air Traffic Management Research and Development Seminar, Seattle, WA, USA, 27–30 June 2017.

15. Sun, J.; Ellerbroek, J.; Hoekstra, J.M. WRAP: An open-source kinematic aircraft performance model. *Transp. Res. Part Emerg. Technol.* **2019**, *98*, 118–138. [CrossRef]

16. Ben-Kiki, O.; Evans, C.; Ingerson, B. *Yaml Ain't Markup Language (YAML) Version 1.1*; Tech. Rep; yaml.org. 2005; p. 23. Available online: https://yaml.org/spec/1.1/ (accessed on 1 October 2019).

17. Jenkinson, L.R.; Simpkin, P.; Rhodes, D.; Jenkison, L.R.; Royce, R. *Civil Jet Aircraft Design*; Arnold: London, UK, 1999; Volume 338.

18. Paul, J. *Jane's All The World's Aircraft. 2004–2005*; Jane's Information Group Inc.: Alexandria, Egypt, 2005.

19. ICAO. *Aircraft Engine Emissions Databank*; International Civil Aviation Organization: Montreal, QC, Canada, 2018.

20. Svoboda, C. Turbofan engine database as a preliminary design tool. *Aircr. Des.* **2000**, *3*, 17–31. [CrossRef]

21. Sun, J.; Vû, H.; Ellerbroek, J.; Hoekstra, J.M. Weather field reconstruction using aircraft surveillance data and a novel meteo-particle model. *PLoS ONE* **2018**, *13*, e0205029. [CrossRef] [PubMed]

22. Bartel, M.; Young, T.M. Simplified Thrust and Fuel Consumption Models for Modern Two-Shaft Turbofan Engines. *J. Aircr.* **2008**, *45*, 1450–1456. [CrossRef]

23. Sun, J.; Hoekstra, J.M.; Ellerbroek, J. Aircraft drag polar estimation based on a stochastic hierarchical model. In Proceedings of the SESAR Innov. Days 2018, Salzburg, Austria, 3–7 December 2018.

24. Sun, J.; Hoekstra, J.M.; Ellerbroek, J. Estimating aircraft drag polar using open flight surveillance data and a stochastic total energy model. *Transp. Res. Part Emerg. Technol.* **2020**, *114*, 391–404. [CrossRef]

25. McCormick, B.W. *Aerodynamics, Aeronautics, and Flight Mechanics*; Wiley: New York, NY, USA, 1995; Volume 2.

26. Obert, E. *Aerodynamic Design of Transport Aircraft*; IOS Press: Amsterdam, The Netherlands, 2009.

27. Gur, O.; Mason, W.H.; Schetz, J.A. Full-configuration drag estimation. *J. Aircr.* **2010**, *47*, 1356–1367. [CrossRef]

28. Mair, W. *Aircraft Performance*; Cambridge University Press: Cambridge, UK, 1996; Volume 5.

29. Sun, J.; Ellerbroek, J.; Hoekstra, J. Flight Extraction and Phase Identification for Large Automatic Dependent Surveillance–Broadcast Datasets. *J. Aerosp. Inf. Syst.* **2017**, *14*, 566–571. [CrossRef]

30. Peel, R.A. Data Development Documentation. *X-Plane Developer*. 2019. Available online: https://developer.x-plane.com/docs/data-development-documentation/ (accessed on 1 October 2019).