

PRELIMINARY AIRCRAFT DESIGN WITHIN A MULTIDISCIPLINARY AND MULTIFIDELITY DESIGN ENVIRONMENT

Sebastian Wöhler ⁽¹⁾, Georgi Atanasov ⁽²⁾, Daniel Silberhorn ⁽³⁾, Benjamin Fröhler ⁽⁴⁾, Thomas Zill ⁽⁵⁾

(1) German Aerospace Center (DLR), Institute of System Architectures in Aeronautics, Hein-Saß-Weg 22, 21129 Hamburg, Germany, Sebastian.woehler@dlr.de

(2) German Aerospace Center (DLR), Institute of System Architectures in Aeronautics Hein-Saß-Weg 22, 21129 Hamburg, Germany, Georgi.atanaov@dlr.de

(3) German Aerospace Center (DLR), Institute of System Architectures in Aeronautics, Hein-Saß-Weg 22, 21129 Hamburg, Germany, Daniel.silberhorn@dlr.de

(4) German Aerospace Center (DLR), Institute of System Architectures in Aeronautics Hein-Saß-Weg 22, 21129 Hamburg, Germany, Benjamin.froehler@dlr.de

(5) German Aerospace Center (DLR), Institute of System Architectures in Aeronautics Hein-Saß-Weg 22, 21129 Hamburg, Germany, Thomas.zill@dlr.de

KEYWORDS: conceptual aircraft design, multifidelity, collaborative engineering, multidisciplinary design and analysis, openAD

ABSTRACT:

In this paper, the preliminary aircraft design tool openAD is presented. Over the past years the DLR has developed a multidisciplinary and multifidelity design environment for aircraft design to evaluate and assess various concepts and technologies at aircraft level [1], [2], [3]. Thereby, a valid and consistent design can be derived by a minimum set of top-level aircraft requirements. This design environment is highly modular where the domain specific tools are provided and hosted across the different DLR institutes. As part of the design process, openAD was developed as the key enabler for overall aircraft design and assessment.

1. MOTIVATION

OpenAD is a preliminary aircraft design tool that is based on well-understood and mostly publicly available handbook methods. It is programmed object oriented in the Python scripting language and is easily expandable to generate a consistent initial evaluation of an air vehicle design. The current design space is valid for aircraft from 19 passengers (Dornier 228 size) up to 800 passengers (A380 size) with enhanced capabilities on the design and thermodynamic cycle calculation of turbofan and turboprop engines. Dedicated to tube and wing concepts, the software offers wide capabilities to design, among others, strut braced wings, canards, fuselage mounted engines and any combination, depicted in *Figure 1*. By making use of the common language CPACS, as described by Alder et al [4], for import and export, it can easily be linked to software within the DLR and of external partners. OpenAD is used for overall

aircraft model calculations, design space explorations and parameter optimizations as a standalone tool and within a broader design environment. Hence, it is an essential part of the multidisciplinary and multifidelity design workflow, used to derive an initial design and synthesize higher fidelity disciplinary results into the design. To analyze and optimize a wide range of conventional and novel aircraft configurations, the tool gets constantly enhanced and extended. As a core capability of the DLR Institute of System Architectures in Aeronautics, openAD is applied in German national research projects LuFo V-3 AVACON and LuFo V-3 SynergIE as well as in the European research projects Clean Sky 2 LPA ADEC, Clean Sky 2 LPA Hyper-F and Clean Sky 2 Airframe ITD NACOR.



Figure 1 Use cases of the openAD design space

The development of the collaborative design environment for multidisciplinary and multifidelity aircraft design and analysis at DLR was started in the late 2000s out of the realization that more in-depths analyses in the early design phases were required to improve decision making processes, to reduce the time required for intricate studies and ultimately to improve the vehicle design itself. OpenAD originated in 2010 [2] from the need to provide initial aircraft performance data on conceptual level in such multidisciplinary design studies. Early on, it became apparent that these distributed collaborative design workflows, which are usually comprised of analysis capabilities of

varying levels of modelling fidelity, require not only such provider of initial solutions but also a tool that could dynamically adapt to the results provided by the disciplinary tools and then perform the required synthesis calculations. Back then this functionality was allocated to the tool VAMPzero [5]. OpenAD builds on these developments and after major overhauls of its core conceptual design methods it provides the basis for detailed aircraft concept studies at DLR.

2. COLLABORATIVE DESIGN ENVIRONMENT

The multidisciplinary and multifidelity collaborative design environment established among the DLR institutes is based on the open source RCE integration framework [6]. The workflow utilizes the capabilities to publish and distribute tools remotely from different DLR sites and integrate them into an automated design environment. In addition, RCE is not only used to connect and manage standalone tools but it also offers its own toolkit with modules for design of experiments, optimization algorithms and convergers among others to compose workflows, nested loops and sub-workflows.

The collaborative design environment is segmented into level of fidelity ranging from L0 to L2 to ensure an appropriate calculation effort at each section. A flowchart of the workflow is depicted in *Figure 2*.

To launch the design workflow a minimum set of top level aircraft requirements (TLARs) and design parameters are required. For the TLARs the design range, cruise altitude, Mach number, payload definition, take-off field and landing field length as well as the reserve mission specification are mandatory. Within the design parameters, configurational decisions have to be made. The wing vertical position, the engines location, the empennage arrangement and the landing gear configuration will be fixed. If no decisions are made, a conventional configuration as the Airbus A320 or Boeing 737 aircraft is set as default. Likewise, calibration factors and design constraints can be defined. For the initial sizing, the design parameters for wing loading and thrust-to-weight-ratio can be specified or varied as a design of experiment or optimization target among others.

The aircraft design process, shown in *Figure 2*, is initiated by openAD within the L0 segment. Based on TLARs and design parameters, the tool derives the geometrical layout, mass breakdown, aerodynamics and engine performance along with the mission performance based on empiric-, semi-empiric and analytical methods described in chapter 3. On a conceptual level, first studies can already be performed deploying openAD without

any additional tools or segments of the workflow. The results are exported in the CPACS format and are forwarded to the L1 convergence loop.

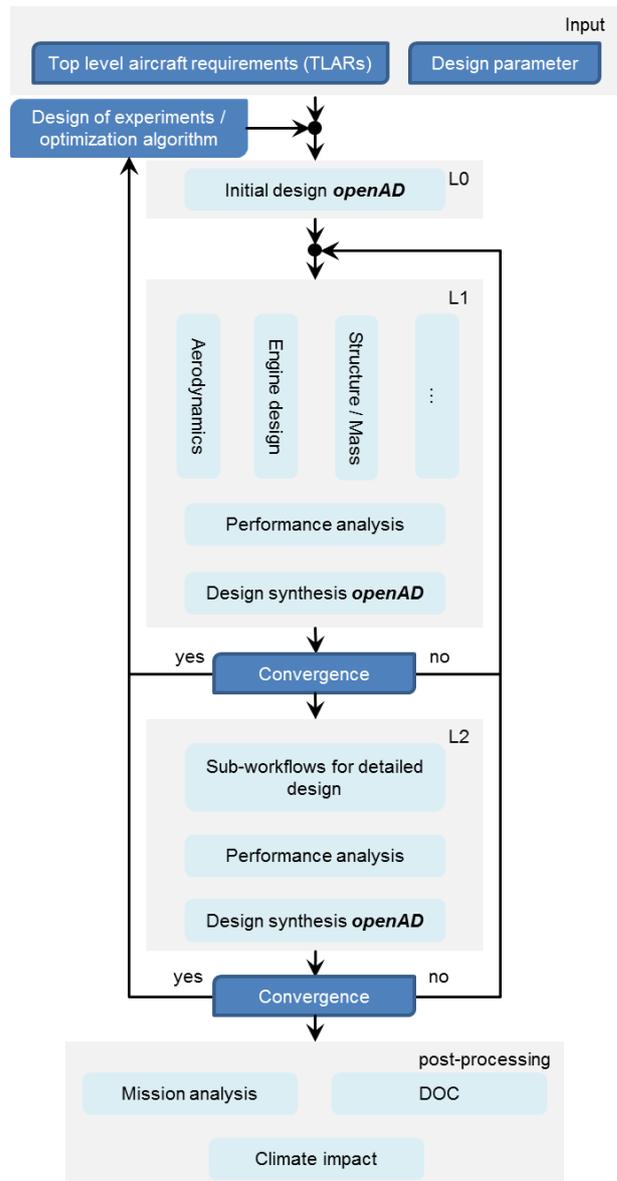


Figure 2 Flowchart of the design workflow

Within the L1 section dedicated disciplinary tools are introduced to increase and refine the quality and accuracy of the design studies performed using simple physic based or surrogate models of higher fidelity methods. The core of this part of the workflow is composed by aerodynamics, engine performance and mass estimation. “Lifting Line” is used for the aerodynamic flow simulation, calculating the lift distribution, induced drag and pitching moments [7]. Additionally the tool “HandbookAero” adds the results of the viscous drag for each component based on semi-empirical

methods [8], [9]. The wave drag is estimated within Python script as described in chapter 3.2. In summation, the tools and scripts define the aerodynamic characteristics for the designed aircraft dependent of the geometry defined in CPACS file.

For the engine performance, predefined decks can be imported. In the majority of the projects mentioned earlier engine decks were provided by the DLR Institute of Propulsion Technology or other project partners where a detailed design was performed. For deviation of the design point or case studies, engine scaling rules are applied to rubber the thrust and other engine parameters according to the requirements defined in the L0 segment of the workflow within applicable margins. If no engine performance is available, openAD is capable of computing a simplified cycle calculation described in chapter 3.3.

The wing mass is calculated by “CLA”, a reduced order model of a wing analysis tool using beam theory to replace the semi-empirical approach to consider aero elastic effects and CFRP wings in the design as described in [10].

Beside the three described disciplines, additional modules can be integrated to calculate for example the center of gravity shift during the design mission based on a predefined defueling scenario. Subsequently, the results of the parallel executable disciplines are merged. In this case the component and design masses will be updated and the engine and aero performance will be substituted and forwarded to the performance analysis. “AMC” was developed at the DLR Institute for System Architectures in Aeronautics and is used to estimate the 2D stepwise mission performance. An aero-engine performance optimized step climb will be computed while also initial cruise altitude optimization and trim drag calculation at each time or mass step can be performed.

In each loop, the results are handed over to the design synthesis in openAD. The corresponding values are updated and a design iteration is performed. At this point, parameters are iterated that are directly dependent on the maximum take-off mass since the component masses and the fuel mass needed for the design mission is fed back. Hence, the so called snowball effects will be calculated and a new consistent design is generated. If the variation of the maximum take-off mass compared to the start of the L1 is within the convergence bound, the convergence loop is closed.

Within the L2 segment, the tools and scripts are located in sub workflows to limit the complexity of the primary design workflow. Detailed design studies can be conducted, as for example the

design of the main wing and tail plane primary structure using finite element methods. Afterwards the results are again fed back into AMC and openAD for synthetization.

If convergence is reached, the aircraft design is completed and the post processing can be conducted. For the mission analysis, the payload-range diagram with any desired number of ranges is calculated using AMC while also an assessment mission can be evaluated. In last step, direct operating cost and the climate impact can be analyzed based on the aircraft performance.

For specific studies of novel aircraft configurations or technology assessments, the design workflow can be adapted or extended with suitable modules with respect to the configurations and level of fidelity. Due to the utilization of the standard exchange format CPACS across all tools, scripts adaptations and extensions of the workflow as well as the exchange of certain modules can be implemented easily and flexible. Furthermore, CPACS is used to assure a consistent aircraft model base throughout the aircraft design process. Besides the aircraft description, CPACS also offers the “toolspecifics” section where particular tool settings for each module can be defined locally or within the workflow input.

3. OPENAD TOOL DESCRIPTION

OpenAD is a software tool for preliminary aircraft design developed in Python. It is based on well-understood and mostly publicly obtainable handbook methods [8], [11], [9], [12], [13], [14], [15] and own methods for disciplines for which no adequate or only insufficient methods could be sourced from the literature. OpenAD can interpret the CPACS tool specifics data as an input to define calculation settings and to manipulate calculation parameter. Even more important, openAD can be used to generate a CPACS output file where the relevant data is exported to the CPACS schema and additional output can be written into the tool specifics.

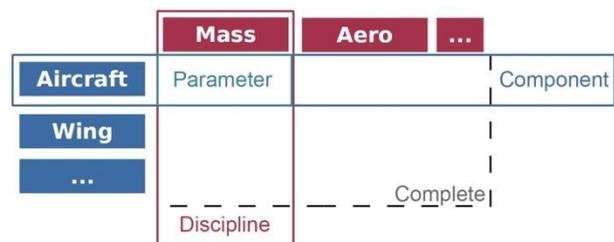


Figure 3 Arrangement of the main classes in openAD (extracted from [5])

Due to the fact that openAD is based on an object oriented structure it is highly flexible. Furthermore, the structure, distinguishes feature aspects: file handling, convergence- and process control as well as design aspects: parameter definition, calculation methods. The design aspects are grouped into components, disciplines and parameters, whereas the parameters contain the actual design knowledge in a way that makes extensions easy to implement (see *Figure 3*). Each parameter holds several attributes that can directly be accessed and adapted in the CPACS tool specifics.

Available parameter attributes:

- *value*
- *unit*
- *factor*
- *status*
- *cpacsPath*
- *upper-/ lower bound*

The *value* is defined in the corresponding unit while the units can be arbitrary strings. SI units (international system of units) are preferred and a unit consistency check, a configuration option of openAD, can be executed at each run. The *factor* may be defined for every parameter to introduce calibration or technology factors which are not directly modelled within the knowledge base. For the CPACS export of the parameters, the respective location may be given in the *cpacsPath*. If the design space for a specific parameter is only valid for a certain range, boundaries can be defined in the *upper-* and *lower bound* attributes. The *status* determines the handling of the parameter during the calculation. Available options are “fixed”, “init” and “calc”. An openAD run is separated into an estimate run and a full calculation execution. If a *value* is defined in the input file it will be set to fixed and will not be calculated nor overwritten in any case. If a *status* is defined as “init”, it will be used in the estimate run and will be reevaluated during the calculation run. The “calc” setting is the default *status* and indicates that the value will be calculated and checked for convergence. Therefore, the convergence criteria can be predefined. For both runs, the execution will be terminated when all parameters are converged. The implementation of estimation and calculation run aims at improving the performance and ensures robustness for the aircraft design process. *Factor*, *value* and *status* allow a simple but efficient way to manipulate every parameter within the knowledge base

enabling the diverse use within the aircraft design environment.

The design of new configurations includes outer geometry as well as the mass breakdown of the aircraft, engine performance, aerodynamics, systems and mission analysis. Hereafter, methods will be described that are specific to openAD and are not documented in the literature.

3.1 Masses and center of gravity

The focus of the implementation of mass methods for openAD was put on sensitivities rather than the highest achievable accuracy. During the selection process, different methods were compared and if required, methods with high accuracy were not selected if the necessary sensitivities were not represented. For example the mass method for the main landing gear in the Luftfahrttechnisches Handbuch [12] is only dependent on the maximum take of mass and was not selected as default in favor of the mass method from NASA FLOPS [14] due to the additional dependency on the main landing gear length. Additionally, for each component several mass methods are implemented and can be selected by the user if desired.

Corresponding to the mass, methods for the center of gravity for each component are implemented. While performing trade studies on different configurations, the need to divide the component mass of the overall system into individual sub-systems was identified. The center of gravity for the sum of all systems is not sufficient to dissolve the influence of each system on the overall aircraft center of gravity. Therefore, methods for the mass and center of gravity for all aircraft systems according to the airbus weight chapters were derived based on statistics. Empirical or semi-empirical methods from the literature were not accurate enough for a state-of-the art aircraft compared to reference data.

3.2 Aerodynamic modelling

The aerodynamic drag modelling in openAD is based on handbook methods [8], [9], [13] and has the following breakdown:

$$(1) \quad c_D = c_{D0} + c_{Di} + c_{Dw}$$

Where: c_D is the drag coefficient of the aircraft
 c_{D0} is the aircraft zero-lift drag coefficient
 c_{Di} is the lift-dependent drag coefficient (excluding compressibility effects)
 c_{Dw} is the wave drag coefficient

Zero-lift drag coefficient

The zero-lift drag coefficient calculation follows the methodology described by Raymer [9]. The zero-lift drag of the aircraft is the sum of the zero-lift drag coefficients of all containing components. These are depending on:

- The wetted area of the respective component.
- The Reynolds Number, which is dependent of the flight condition (Mach number, altitude, and temperature deviation from the standard atmosphere) as well as on the mean aerodynamic chord or length of the component.
- The form coefficient, which depends on the geometric properties of the component.
- The interference drag factor, which is defined individually for each component

Lift-dependent drag coefficient

The lift-dependent drag coefficient (see Eq. (1)) is calculated as follows:

$$(2) \quad c_{Di} = l \cdot c_L + k \cdot c_L^2$$

Where: c_L is the lift coefficient of the aircraft
 l is the coefficient for linear dependency of drag on the lift coefficient
 k is the coefficient for quadratic dependency on lift

The linear dependency on lift is set by default to zero, thus its main function is that it can be used by the user for polar calibration. The factor for the quadratic dependency on lift is calculated by the classic equation for a 3D wing:

$$(3) \quad k = \frac{1}{\pi \cdot AR \cdot e} \quad (\text{for } c_L < c_{L,PB})$$

Where: AR is aspect ratio of the wing
 e is the Oswald coefficient
 $c_{L,PB}$ is the polar break lift coefficient

As shown in Eq. (3), the modelling assumes a quadratic polar approximation up to a certain lift coefficient, which is defined as the “polar break” lift coefficient. For higher values, the quadratic lift dependency coefficient increases significantly causing a much faster decline in the lift-over-drag polar than for a symmetrical polar (see Figure 4). The polar break modelling assumes that the wing is optimized for a certain lift coefficient. This

coefficient is set to the mid-cruise condition by default.

Wave drag coefficient

The wave drag calculation is adopted from Torenbeek [8], which gives the dependency of wave drag on the Mach number for a certain divergence Mach number. The divergence Mach number depends on the overall wing geometry and on the lift coefficient. The calculation method is given by Torenbeek [13]. If the wing design is known from a reference aircraft or from a Hi-Fi calculation method, a combination of a divergence Mach number and a lift coefficient can be specified in the input, which calibrates the method to this combination.

Aerodynamic Polars

The drag calculation method described in the previous sections is summarized in Figure 4. An example of calculated aerodynamic polars is given in Figure 5. The output of openAD also includes an aerodynamic map generated for CPACS.

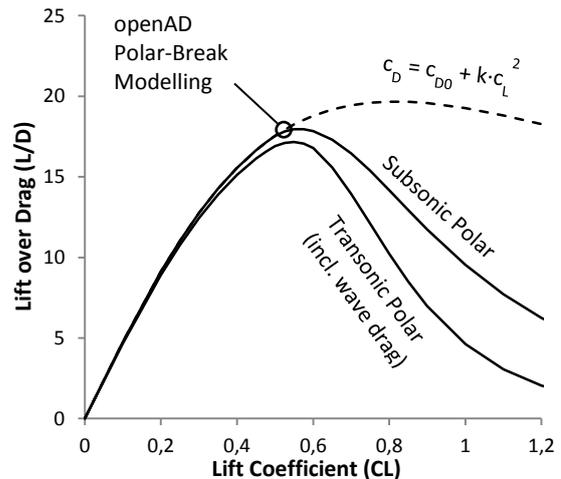


Figure 4 Aerodynamic polar modelling in openAD

There are multiple possibilities to adjust and calibrate the polars:

- A global drag factor can be specified, which affects all calculated drag values.
- A factor on the zero-lift drag can be set either for the entire aircraft or for each component separately.
- An additional zero-lift-drag coefficient can be specified, which will be added on top of the aircraft zero-lift-drag coefficient. This value remains constant for all polar points.

- A factor for linear dependency on lift can be specified (see Lift-dependent drag coefficient).
- The Oswald factor can be set to a fixed number or a factor can be applied on its calculation (see Lift-dependent drag coefficient).
- The wave drag calculation can be calibrated by specifying a combination of a divergence Mach number and a lift coefficient (see Wave drag coefficient).

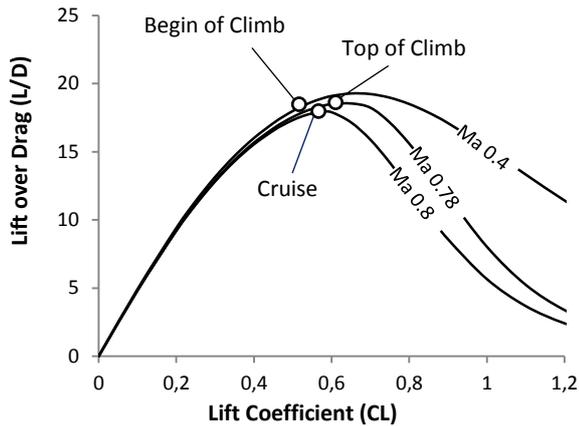


Figure 5 Aerodynamic polar examples for a generic transonic airliner.

3.3 Geometry Definition

As default, openAD is set to design a conventional aircraft configuration. Changes deviating of the conventional configuration can be defined in the tool input, whereas masses, center of gravity and the geometry of each component will be adjusted due to the parametrization.

The fuselage geometry definition is based on an inside out approach where the cockpit, passenger number and cabin definition is sizing fuselage length and height. Different classes can be implemented and the number of galleys and toilets is determined by comfort standards and the design range. The number of seats abreast sizes the width and height whereas the deviation of a circular cross-section can be specified within the input parameters. As a function of the average diameter, the nose and tail necking is derived.

The wing planform is divided into 4 segments and thereof 5 stations:

- Center station at the symmetry line of the fuselage width.
- Root station at the fuselage-wing intersection.

- Kink station at the kink of a triple trapezoid wing.
- Mid station can be defined at any span wise position between kink and tip. Default is set to 95% of the half wing span.
- Tip station at the most outboard position of the wing.

The wing geometry is build up from the trailing edge of the wing. If a constant dihedral is chosen over all segments, the trailing edge would be a continuous line from center to tip station in the back-view perspective. The planform can either be defined by global parameters or locally for each station and segment. Dependencies are set for both options where the geometry gets analytically solved (see Figure 6).

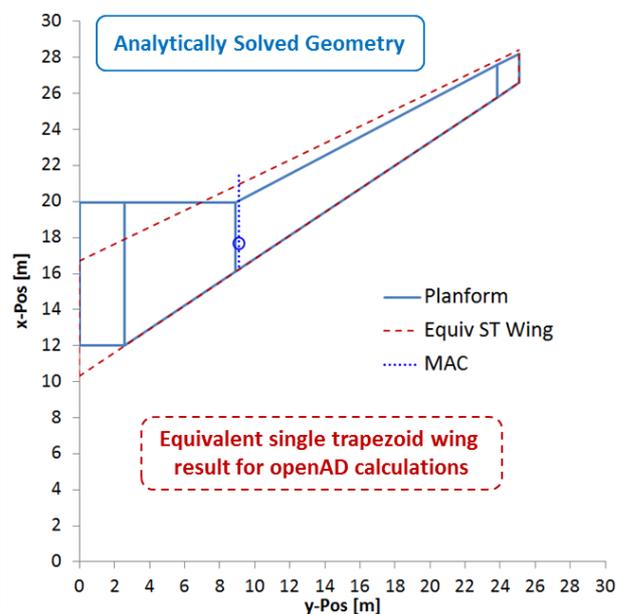


Figure 6 Wing planform definition in openAD

By utilizing the global geometry parameters, a conventional single- or triple trapezoid wing can be defined by a minimum set of parameters (aspect ratio, twist, span, kink span wise position, quarter chord sweep, thickness and dihedral). For any arbitrary shape or precise planform definition, local parameters are to be adjusted, e.g. the angles of incidence, leading and trailing edge sweeps or dihedrals. Examples of feasible planforms are shown in Figure 1. An equivalent single trapezoid wing is calculated by the program based on the defined planform, which is needed for certain aero and flight mechanic methods.

Based on the defined planform, an initial moveable layout will be derived. The layout is fully parametric and engine cut outs are placed based on the

engine position. Any desired number of slats, spoilers, flaps and ailerons can be defined or left out if chosen. In *Figure 7* a typical wing moveable layout is depicted including the engine cut out and split ailerons. Due to the dependencies implemented, the movables are aligned with the parametric spar placement.

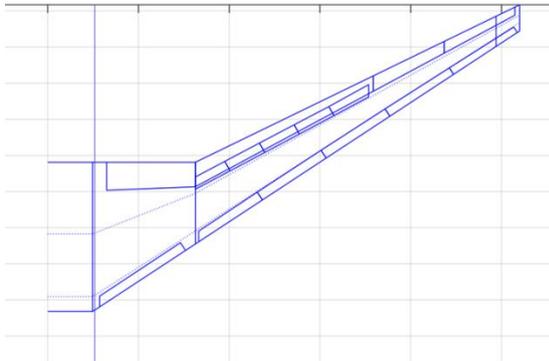


Figure 7 Wing moveable parametrization in openAD

Design options for the tail plane are a conventional empennage, a t-tail or a canard and vertical tail plane arrangement. The surfaces are sized dependent on the volume coefficients and lever arms. For the default positioning, the front spar of the vertical tail plane gets aligned with the rear pressure bulk head and the horizontal stabilizers front spar gets aligned with the respective rear spar of the vertical tail plane. In a t-tail configuration both front spars get matched at the top intersection. In a canard design, an appropriate location has to be chosen in the input section. However, the user is fully flexible in overwriting the default positioning of the empennage surfaces. The turbofan or turboprop engines can either be attached to the fuselage or the wing. Thereby, any position along the fuselage or in span wise location can be defined. The pylons get positioned and sized accordingly. Any number of engines between one and four can be implemented. In the case of three engines, the third engine can be placed individually. The engines geometry is dependent on thrust and the bypass ratio, whereas the nacelle geometry is sized consequently. For the landing gear, options for fuselage attachment and wing attachment are provided. In either case the landing gear length is sized to offer sufficient clearance angle to the engine and tip as well as to the tail section for the take-off rotation. The main landing gear position along the fuselage axes is based on the center of gravity of the aircraft.

3.4 Engine performance modelling

Multiple methods for calculating the engine performance are implemented in openAD.

- Option 1: a generic calculation.
- Option 2: gas turbine cycle calculation.
- Option 3: a generic thrust lapse behavior with a cycle calculation for the fuel consumption.

Option 1 is the default setting, which is the most simple and robust option. This option is the best choice when the tool is used in a workflow which includes an engine performance module that offers a detailed engine performance calculation. For stand-alone studies, where the engine performance is known or only partially known, Option 2 and Option 3 are generally the better choice.

The cycle calculation is based on a simplified, mostly analytical method with pre-set gas turbine definition parameters, which can be changed by the user. The calculation assumes ideal gas and does not include a gas-turbine cooling loop. The inputs needed for the cycle calculation module are:

- Bypass Ratio (BPR) or Fan Pressure Ratio (FPR) at mid-cruise condition, which is the engine design point.
- Overall Pressure Ratio (OPR) at the design point.
- Maximal Turbine Entry Temperature (TET).

Some optional inputs are:

- TET values for the different characteristic flight phases (take-off, climb, and cruise).
- Switch for a turboprop calculation (the default is set to turbofan).
- Switch for offtakes calculation (turned off by default). When turned on, the tool assumes a conventional bleed architecture with appropriate mechanical and bleed offtake values depending on the cabin size and number of passengers. The offtake values can also be manually specified by the user.

The outputs of the engine cycle calculation (Option 2) are:

- Thrust, shaft power, and fuel consumption calculation for the characteristic flight conditions (take-off, climb, and cruise).
- Fan diameter sizing.

This setup is able to provide a rough estimate of the fuel efficiency and the thrust lapse behavior for a specified engine. Calibrating the specific fuel consumption parameter to a known value allows for a reliable estimation of the engine behavior throughout the different aircraft flight points needed for the mission calculation.

Since the engine cycle calculation takes into account the flight altitude and speed of the aircraft, it allows for basic mission parameter trade-off studies already at the conceptual aircraft design level.

3.5 Mission fuel calculation

In openAD, the fuel consumption is calculated for each segment of the flight (see *Figure 8*) based on the design- and reserve mission definition. The fuel masses of each segment are grouped into mission block fuel, reserves and fuel on board. The mission block fuel is added up by the following parts:

- Taxi-out fuel needed prior to take off to get to the take-off position.
- Take off fuel consumed during the take-off run and the initial climb segment up to 1500ft.
- Climb fuel from begin of climb at 1500ft to top of climb at initial cruise altitude.
- Cruise fuel from end of climb till start of descent.

- Descent fuel from end of cruise altitude to 1500ft.
- Approach and landing fuel from 1500ft till end of runway at the destination airport.
- Taxi-in fuel needed from runway to the gate.

The reserves are arranged into:

- Diversion fuel necessary to reach an alternate airport.
- Loiter fuel for holding in above the alternate airport.
- Contingency as percentage of the block fuel, this is typically set to 5% or to 3% if it can be assumed, that an alternate airport is available enroute [16].

The fuel on board is composed of the mission block fuel and the reserves. However, the taxi-in fuel is not included in the total fuel despite being part of the block fuel. According to the Acceptable Means of Compliance [16] taxi-in fuel has not to be accounted in the preflight fuel planning. Thus, for taxi-in from the runway to the gate the reserves can be utilized and no additional mass needs to be carried along for each flight.

For taxi-out, approach and landing as well as taxi-in, the fuel mass is calculated top-down based on the fuel flow and the time needed at each segment. The time for each of those segments is set to default values, which emulate by usual airline operations. The fuel flow rate is calculated analytically dependent on the engine and mission inputs [17].

In the climb calculation, a Breguet-like method is

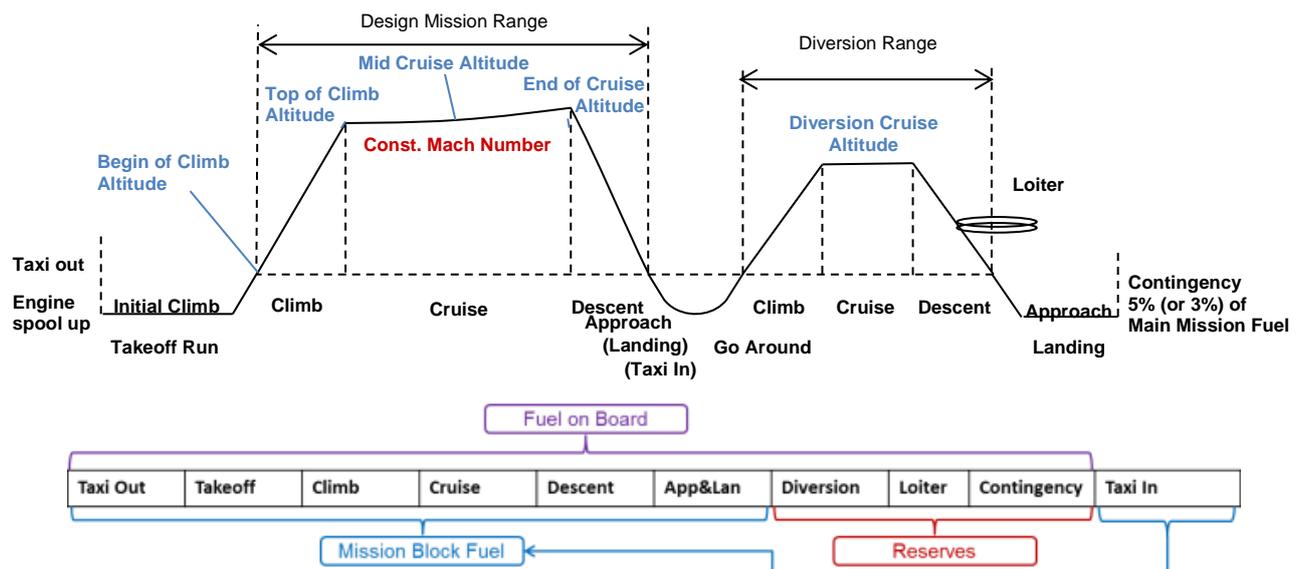


Figure 8 Design mission segment and fuel breakdown definition

analytically solved, where thrust, fuel flow and velocity are assumed to behave linearly with the altitude between begin and top of climb. The descent calculation is conducted with a similar approach where the lift over drag ratio is assumed to be constant and the thrust is set to zero. For the cruise calculation two options are implemented:

- Breguet equation approach with constant lift coefficient and velocity
- Breguet equation approach with constant flight altitude and velocity

For the reserve mission the climb and descent fuel masses are derived and corrected from the main mission results, whereas the cruise segment is calculated using the Breguet equation with a constant lift coefficient and constant velocity. The fuel mass accounted for loiter is based on the Breguet endurance equation.

4. VALIDATION AND SENSITIVITIES

The introduction provided in chapter 1 claims a viable design space for aircraft from 19 to 800 seats is claimed. Therefore, this chapter highlights the results of reference aircraft calculation for projects and design studies carried out with openAD. The payload range characteristics are presented for aircraft summarized in *Table 1*. Subsequently, sensitivities on key aircraft parameters are shown for the Airbus A320neo similar aircraft calculated in openAD. Alterations of aerodynamic and engine efficiency as well as changes of masses are depicted to demonstrate the impact on the aircraft performance.

The Dornier 228, British Aerospace S31 and Beechcraft 1990D represent the commuter class aircraft characterized by turboprop engines flying at low Mach numbers of around 0.4 on regional ranged missions. The reference aircraft selected cover a wide range of configurations. The variation of high or low wings and pressurized or non-pressurized cabins comprise the broad design capabilities of openAD that were validated along the studies performed by Atanasov et al. [18]. For the short range segment, the Airbus A320neo and Boeing B737 Max8 are applied as use cases. For these configurations, comprehensive and consistent data sets are available which enables an extensive validation of the openAD design functionalities.

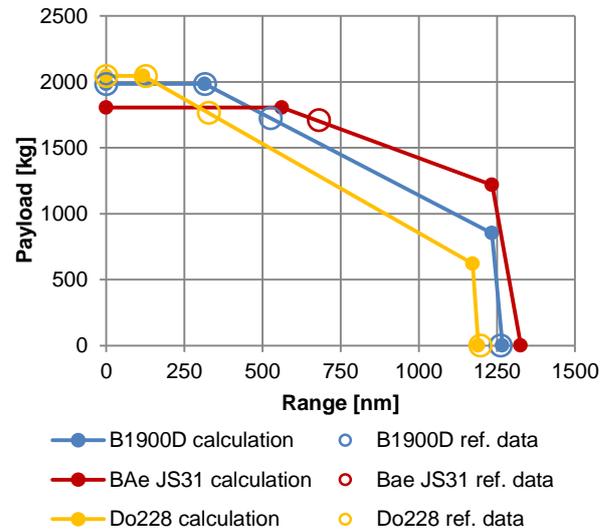


Figure 9 Payload-Range characteristics comparison of openAD calculation and reference data for the commuter range mission

Table 1 Aircraft characteristics for the use cases presented

| Top level aircraft characteristics | Unit | Dornier 228 | BAe Jetstream S31 | Beechcraft 1900D | Airbus A320neo | Boeing 737 MAX 8 | Airbus A380 |
|------------------------------------|------|-------------|-------------------|------------------|----------------|------------------|-------------|
| pax typical layout | [-] | 19 | 19 | 19 | 165 | 162 | 555 |
| payload at design point | [t] | 1.7 | 1.7 | 1.7 | 17.0 | 18.5 | 52.7 |
| range at design point | [nm] | 370 | 680 | 525 | 3100 | 3000 | 7980 |
| cruise Mach number | [-] | 0.36 | 0.42 | 0.46 | 0.78 | 0.78 | 0.85 |
| maximum take-off mass | [t] | 6.4 | 6.95 | 7.77 | 79.0 | 82.2 | 575.0 |
| wing span | [m] | 16.97 | 15.85 | 17.64 | 35.8 | 35.9 | 79.8 |
| number of engines | [-] | 2 | 2 | 2 | 2 | 2 | 4 |
| type of engines | [-] | Turboprop | Turboprop | Turboprop | Turbofan | Turbofan | Turbofan |

The Airbus A380 represents the largest commercial aircraft, introduced for passenger transport, ever build. Along internal studies, the geometry and initial performance of the aircraft were modeled in openAD and the methods verified accordingly. The A380 defines the upper boundary of the design space validated in openAD and shows results for the long range mission.

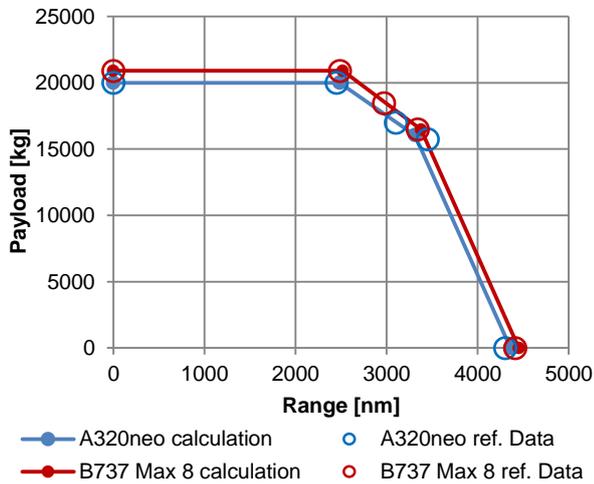


Figure 10 Payload-Range characteristics comparison of openAD calculation and reference data for the short range mission

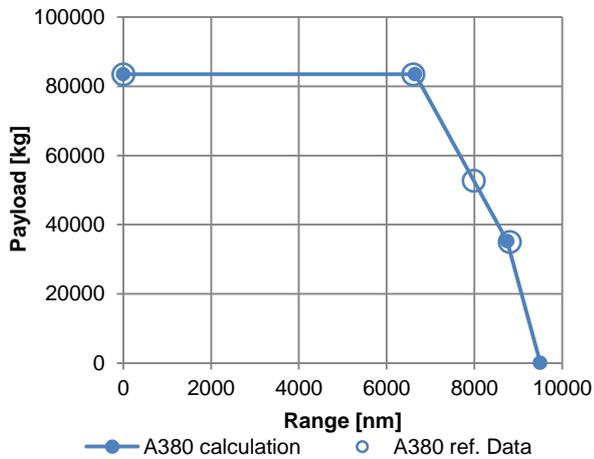


Figure 11 Payload-Range characteristics comparison of openAD calculation and reference data for the long range mission

In Figure 9, Figure 10 and Figure 11 the payload-range characteristics of the aircraft described in Table 1 ranging from commuter to long range mission are presented. The results of the openAD calculations are shown and compared to reference data [18], [19], [20], [21]. For this purpose, the payload-range diagram is particularly convenient. The presented use cases are set-up at their specific design point in openAD. The corner points

are directly linked to the design masses, payload definition and useable fuel capacity (see [8]). The slope of the substitution lines is mainly dependent on the aerodynamic and propulsion efficiency of the aircraft. The underlying aerodynamic performance calculated by openAD is exemplarily shown in Figure 12 for the 19 class seater and in Figure 13 for the A320neo similar aircraft.

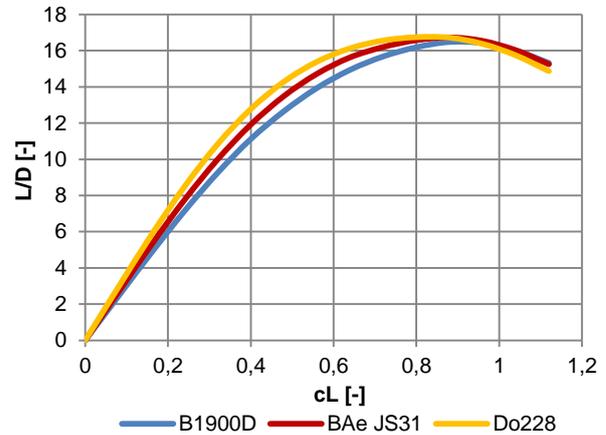


Figure 12 Aerodynamic polars of the 19 seater aircraft calculated by openAD at mid cruise condition

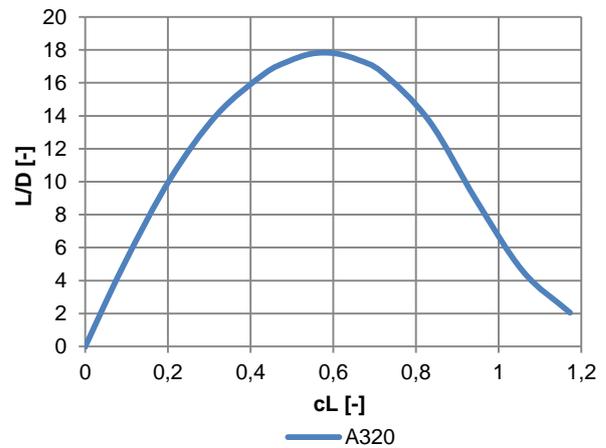


Figure 13 Aerodynamic polar of the A320neo similar aircraft calculated by openAD at mid cruise condition

Across all ranges and seating capacities, the comparisons of the openAD results to the reference data demonstrate a high level of accuracy. Besides the TLAR definition and configurational decisions, only little effort is needed to match the reference performance. Adaptations and calibration to engines and aerodynamics are limited to a minimum parameter set. The largest deviations are seen on the masses across all aircraft. This is mainly due to conditional architecture, which is unique to each design e.g.

the double deck configuration of the A380 fuselage or the rectangular cross-section of the Dornier 228. *Figure 14* highlights the sensitivities of the A320neo similar configuration based on the implementation of methods in openAD. The parameters specific fuel consumption (SFC), operating empty mass (OEM) and total drag (cD) are varied by $\pm 5\%$. Within such a range, the influence is expected to be linear for smaller variations. The parameter changes presented are independent from technology development and only highlight the impact on aircraft level. Generic factors are applied as described in chapter 3 for the aerodynamics and engine performance. For the mass variation, a mass component at the OEM center of gravity is implemented.

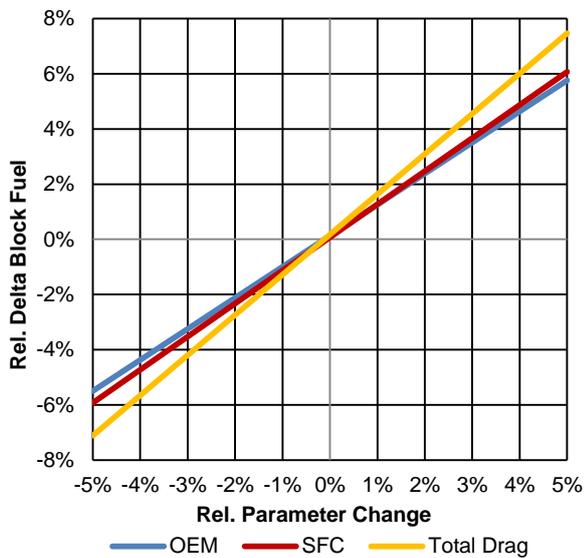


Figure 14 Impact of parameter variations of operating empty mass (OEM), specific fuel consumption (SFC) and total drag (cD) on block fuel for the A320neo similar calculated by openAD

For each parameter change, an entire sizing loop within openAD is carried out, whereas the complete geometry, engine performance and aerodynamic behavior as well as the mission performance are recalculated. The sensitivity study identifies that the total drag variation results in the highest overall impact. By decreasing the total drag of the aircraft, the thrust requirement is reduced and hence, a resized engine with less installed thrust is required. Albeit somewhat lower, the impact of mass and SFC are on a similar level, which is the general expectation.

5. CONCLUSION

The predecessor to openAD, VAMPzero was developed with the focus on supporting the design and synthesis in a multifidelity and multidisciplinary

design environment. Its primary task was to manage the output of disciplinary tools and process these into a consistent design rather than initiating an accurate aircraft design. At this point, VAMPzero was depending on the availability of disciplinary tools. In cases where these were not available, VAMPzero had to provide the required capabilities on a low fidelity level. Therefore, the need to enhance the design capabilities in VAMPzero was identified.

Over the recent years, extensive work was invested into the development of the overall aircraft design capabilities. OpenAD's (renaming due to the extensive rework of the knowledge base) latest version is now in a state, where it can be utilized as a standalone tool for conceptual design studies and in the DLR design environment to support detailed design studies as described. Going more into detail, the underlying sizing principles and methods distinctive to openAD are highlighted. The object oriented programming in python and the establishment of the main classes enable fast and flexible extensions to the knowledge base.

This paper also highlights the application of openAD in aircraft design studies ranging from a 19 seater to 800 seats across turboprop and turbofan engine applications for regional, short and long range missions. The consistent results in this wide variety of use cases provide an indication of the dependability and flexibility of the tool. However, due to the complex nature of aircraft design, the flexible modelling capability also imposes the need of a high degree of familiarity with the structure and functionality of the software as a prerequisite on the user.

The work presented emphasizes the current status of openAD. Corresponding to ongoing project and research, the tool is under successive development and enhancements are introduced permanently to broaden and refine the design space.

ACKNOWLEDGEMENT

The authors would like to thank Johannes Hartmann, Matthias Strack and all colleagues and students involved in the development of openAD and previously VAMPzero.

REFERENCES

1. Liersch, C., Hepperle, M., A distributed toolbox for multidisciplinary preliminary aircraft design, CEAS Aeronautical Journal, vol. 2, no. 1–4, pp. 57–68, Aug. 2011.

2. Zill, T., Böhnke, D., Nagel, B., Preliminary Aircraft Design in a Collaborative Multidisciplinary Design Environment, 11th AIAA Aviation Technology, Integration, and Operations Conference, Virginia Beach, USA, 2011.
3. Moerland, E., Pfeiffer, T., Böhnke, D., Jepsen, J., Freund, S., Liersch, C., Chiozzotto, G., Klein, C., Scherer, J., Hasan, Y., On the Design of a Strut-Braced Wing Configuration in a Collaborative Design Environment, AIAA AVIATION Forum, Denver, USA, 2017.
4. Alder, M., Moerland, E., Jepsen, J., Recent advances in establishing a common language for aircraft design with CPACS, Aerospace Europe Conference 2020, Bordeaux, France, 2020.
5. Boehnke, D., Nagel, B., Gollnick, V., An Approach to Multi-Fidelity in Distributed Design Environments, IEEE Aerospace Conference, Big Sky, USA, 2011.
6. German Aerospace Center, Institute of Simulation and Software Technology, 2019. Available: <https://rcenvironment.de/>. [Accessed 2019]
7. Horstmann, K.H., Ein Mehrfach-Traglinienverfahren und seine Verwendung für Entwurf und Nachrechnung nicht-planarer Flügelanordnungen, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt, Braunschweig, Germany, 1987.
8. Torenbeek, E., Synthesis of Subsonic Airplane Design, Kluwer Academic Publishers, Delft, Netherlands, 1984.
9. Raymer, D.P., Aircraft Design: A Conceptual Approach, AIAA Education Series, Washington DC, USA, 2012.
10. Chiozzotto, G.P., Initial Weight Estimate of Advanced Transport Aircraft Concepts Considering Aeroelastic Effects, 55th AIAA Aerospace Sciences Meeting, Texas, USA, 2017.
11. Roskam, J., Airplane Design, Part 1-7, Kansas: Design, Analysis and Research Corporation, Kansas, USA, 1985.
12. Luftfahrttechnisches Handbuch, Ausgabe 2018. Available: <https://www.lth-online.de/>. [Accessed 2019]
13. Torenbeek, E., Advanced Aircraft Design, Wiley Aerospace Series, Delft, Netherlands, 2013.
14. Wells, D.P., Horvath, B.L., McCullers, L.A., The Flight Optimization System - Weights Estimation Method, NASA Langley Research Center, Virginia, USA, 2017.
15. Jenkinson, L.R., Simpkin, P., Rhodes, D., Civil Jet Aircraft Design, Butterworth Heinemann, United Kingdom, 1999.
16. European Aviation Safety Agency, Acceptable Means of Compliance and Guidance Material to Part-CAT, Initial issue, 2012.
17. Rick, H., Gasturbinen und Flugantriebe, VDI-book series Springer Vieweg, Berlin, Germany, 2014.
18. Atanasov, G., van Wensveen, J., Peter, F., Zill, T., Electric Commuter Transport Concept Enabled By Combustion Engine Range Extender, Deutscher Luft- und Raumfahrtkongress, Bielefeld, Germany, 2019.
19. Airbus S.A.S., A320 Aircraft Characteristics Airport and Maintenance Planning, Rev Feb 01/18, 2018.
20. Boeing Commercial Airplanes, 737 MAX Airplane Characteristics for Airport Planning, Revision A, 2017.
21. Airbus S.A.S., A380 Aircraft Characteristics Airport and Maintenance Planning, Rev Dec 01/16, 2018.