aerospace

MDPI

Article

A Framework for Aircraft Conceptual Design and Multidisciplinary Optimization †

Saeed Hosseini , Mohammad Ali Vaziry-Zanjany * and Hamid Reza Ovesy

Department of Aerospace Engineering, Amirkabir University of Technology, Tehran 159163-4311, Iran; saeed.hosseini@aut.ac.ir (S.H.); ovesy@aut.ac.ir (H.R.O.)

* Correspondence: vaziry@aut.ac.ir; Tel.: +98-21-6454-3235

† This paper is a highly modified version of a manuscript prepared to be presented at the AIAA SciTech 2024 Conference and it was withdrawn later by the authors. The withdrawn abstract can be accessed from: https://arc.aiaa.org/doi/abs/10.2514/6.2024-0050 (accessed on 4 January 2024).

Abstract: In this research, the architecture and the functionalities of the LAMBDA (Laboratory of Aircraft Multidisciplinary Knowledge-Based Design and Analysis) framework for the design, analysis, and optimization of civil aircraft are presented. The framework is developed in MATLAB R2022a and comprises a modular architecture, which gives the potential for the use of different methods and fidelities for each discipline. The methods can be selected from a set of built-in methods or custom user-defined scripts. Disciplinary modules of the LAMBDA are Requirements, Weight, Sizing, Geometry, Aerodynamics, Engine, Performance, Cost, Emission, and Optimization. This framework has been used for different types of design and optimization problems. When it is applied for the design and optimization of a novel regional TBW (Truss-Braced Wing) aircraft, the operating cost has been reduced by 7.7% in the optimum configuration compared to the base configuration.

Keywords: aircraft design; framework; optimization; multidisciplinary; conceptual design

check for updates

Citation: Hosseini, S.; Vaziry-Zanjany, M.A.; Ovesy, H.R. A Framework for Aircraft Conceptual Design and Multidisciplinary Optimization. *Aerospace* **2024**, *11*, 273. https://doi.org/10.3390/aerospace11040273

Academic Editor: Dieter Scholz

Received: 20 February 2024
Revised: 28 March 2024
Accepted: 28 March 2024
Published: 30 March 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

It is estimated that air travel is growing by 4% annually [1], which follows an increment in the number of aircraft, and consequently, the emissions produced. In response to this, environmental requirements are becoming more and more stringent to mitigate the environmental impacts resulting from increased aircraft emissions for 2030 and 2050 [2]. As TAW (Tube-and-Wing) aircraft configuration has reached a matured state over the past few decades, new environmental targets can be met by the development of novel configurations. Many novel configurations are investigated to achieve these targets, which include TBW (Truss-Braced Wing) (Figure 1) [3], BWB (Blended Wing Body) [4], BLI (Boundary Layer Ingestion) [5], HEP (Hybrid Electric Propulsion) [6], TEP (Turboelectric Propulsion) [7], Hydrogen Propulsion [8], and Box-Wing [9].

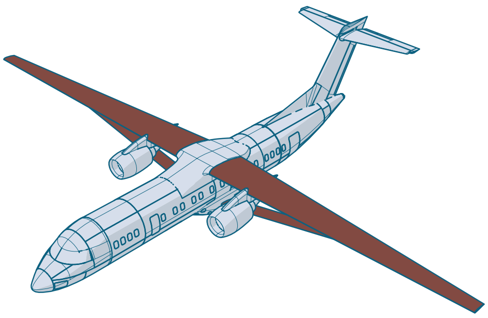


Figure 1. Typical TBW Configuration [10].

These novel configurations should not only provide the capability to increase performance and reduce emissions significantly, but they should also be economically viable. The conceptual design and optimization of novel configurations require the integration of many disciplines (such as weight, aerodynamics, propulsion, cost, emissions, etc.) into a unified design process. On the other hand, aircraft design is an iterative process by nature, and the design loop should be repeated until the design converges. Applying the traditional design methods manually, such as Roskam [11], Torenbeek [12], and Raymer [13], would be time-consuming and computationally inefficient. This issue intensifies when the problem includes optimization, which may require the examination of thousands of design sets. Furthermore, many design tasks are repetitive and work-intensive, particularly for high-fidelity methods (such as grid generation, geometry modeling, etc.), and there will be a considerable reduction in workload if these processes are automated in a computer program. To tackle these issues, aircraft conceptual design frameworks are developed and widely used to optimize the aircraft and evaluate new technologies.

In 1976, NASA (National Aeronautics and Space Administration) ARC (Ames Research Center) developed the program ACSYNT (Aircraft Synthesis) to automate the conceptual design, optimization, and sensitivity analysis of aircraft and included six disciplines: geometry, trajectory (mission performance), aerodynamics, propulsion, structure, and mass properties (weight and balance) [14]. The geometry is generated within the tool, which allows for rapid iteration, but it can be burdensome for complex geometry modeling. Later in 1984, NASA LaRC (Langley Research Center) introduced the FLOPS (Flight Optimization System), which included optimization, mission performance, propulsion, weight estimation, take-off/landing, and aerodynamics modules [15]. In this tool, empirical formations were used to estimate the aerodynamic and weight characteristics of the aircraft rapidly, while requiring a smaller number of inputs compared to other empirical methods [16]. When the weight module was compared with the textbook methods, the lack of thorough documentation and ambiguity in weight breakdown was highlighted by Horvath [16].

The development of program PASS (Program for Aircraft Synthesis Studies) was begun in 1988 by Kroo at Stanford University for aircraft conceptual design and included a module for cost analysis. The emphasis was put on the application of AI (Artificial Intelligence) in the conceptual design [17], and the modularity of this tool allowed for the application of different levels of fidelity depending on the application. A program for aircraft sizing and performance analysis, namely, Piano [18], was started as Ph.D. research by Simos in 1984 [19], and the software was introduced for use in 1990. This program, which is an off-the-shelf commercial tool, features a graphical interface and has modules for geometry, mass, aerodynamics, engine, performance, emissions, cost, and optimization. The program has a focus on performance and emission analyses and includes a large database of commercial aircraft derived from public and proprietary data sources.

In 1990, the commercially off-the-shelf software AAA (Advanced Aircraft Analysis) was introduced by Roskam [20], with emphasis on the user-friendly interface and application of empirical methods. The initial release of the software included weight sizing, performance sizing, geometry, drag estimation, engine, weight, stability and control, cost, and dynamics. The software, which is suitable for speeds lower than a Mach number of 0.7, has a focus on the estimation of stability and control characteristics and adjusting the results using a database of similar aircraft [21]. An expert system for aircraft design was developed in 1991 at Cranfield University by Seung-Hyeog [22], which included parametric design, wing, fuselage, engine, tail, landing gear, weight, and cost analysis modules. PrADO (Preliminary Aircraft Design and Optimization Program) software was initially developed in 1990 at the Technical University of Brunschwig for the conceptual design of hypersonic aircraft [23] and later was extended for the conceptual design of civil subsonic aircraft. RDS (Raymer Design Software) software was introduced in 1992 by Raymer as a student analysis tool and included weight, aerodynamics, weight, sizing, performance, and cost modules [24]. The tool has its own CAD (Computer-Aided Design) module and now comes in two versions: student (free) and professional (paid).

In 2002, the QCARD (Quick Conceptual Aircraft Research and Development) was developed for aircraft conceptual design by Isikveren at KTH Royal Institute of Technology, which was written in MATLAB, and emphasis was placed on the graphical interface and real-time interaction [25]. The tool included mostly quasi-analytical formulas for geometry, weights, aerodynamics, engine, stability and control, performance, cost, and optimization. The user can define the inputs interactively and observe the results in real-time. The MDOPT (Multidisciplinary Design Optimization) was introduced by LeDoux from Boeing in 2004 [26] for air vehicle optimization and included a GUI (Graphical User Interface) for user input. Noticeably, this tool incorporates CFD (Computational Fluid Dynamics) simulations for aerodynamic analysis. A software called DEE (Design and Engineering Engine) was developed at Delft University of Technology by La Rocca in 2009, with great emphasis on the flexibility of aircraft shape generation and FEM (Finite Element Model) automation [27]. An aircraft design tool, namely, CEASIOM (Computerized Environment for Aircraft Synthesis and Integrated Optimization Methods), tailored for more accurate flight dynamics and stability and control properties estimation, was developed under the coordination of KTH Royal Institute of Technology with funds from the EU (European Commission) in 2008 [28] based on the QCARD.

The PreSTo (Preliminary Sizing Tool) tool, which is a set of Microsoft Excel worksheets, was introduced in 2011 by Seeckt from Hamburg University of Applied Sciences and included modules for configuration, propulsion, sizing, cabin, fuselage, wing, tails, mass, stability and control, landing gear, aerodynamics, performance, cost, and geometry [29]. Though the implementation of the framework using spreadsheet tools will increase the speed of analysis and reduce the pre-processing time, it is not suitable for high-fidelity problems. As a part of the NASA N+3 program in 2010, the TASOPT (Transport Aircraft System Optimization) program was developed by Drela at MIT (Massachusetts Institute of Technology), with an emphasis on environmental constraints (such as noise and emission), and it was used for the design and optimization of BLI aircraft [30].

The VAMPzero [31] was developed at DLR (German Aerospace Center) for aircraft conceptual design that uses CPACS (Common Parametric Aircraft Configuration Schema) [32] as the data model for storing and communicating aircraft data. A geometry-oriented tool called RAPID (Robust Aircraft Parametric Interactive Design) was developed at Linköping University in 2013, in which CATIA was used as the core geometry engine for modeling the wing, fuselage, cabin, windshield, wingtip, interior, nacelle, pylon, engine, and control surfaces [33]. This tool was integrated into an aircraft design framework, namely, CAD-Lab, which included modules for aerodynamic analysis and system simulation [34]. A more generic tool for pre-designing technical systems, including aircraft, was presented by DLR in 2013 [35]. A data model named ADDAM (Aircraft Design DAta Model) was developed in 2015 at TUM (Technical University of Munich) using MATLAB language for storing and communicating aircraft data in the ADEBO (Aircraft DEsign BOx) tool, which includes sizing, airfoil aerodynamics, aircraft aerodynamics, weight, mission performance, and geometry [36]. Python language was used for the development of PyPAD (Python module for Preliminary Aircraft Design) at Politecnico di Milano in 2015 for the preliminary design of aircraft, and an emphasis was put on high-fidelity structure sizing methods [37]. In 2015, SUAVE (Stanford University Aerospace Vehicle Environment) was developed for the multi-fidelity design of unconventional aircraft using Python language [38]. The JPAD (Java toolchain of Programs for Aircraft Design), which is an open-source Java-based application for aircraft conceptual design, was introduced in 2016 by the Aircraft Design Research Group at the University of Naples and is capable of parametrizing the aircraft geometry, generating the CAD model, and employing semi-empirical methods for the rapid estimation of weight, aerodynamics, performance, stability and control, and cost [39].

RADE (Rapid Airframe Design Environment) was developed at Georgia Institute of Technology in 2018 as "a modular, parametric, and multidisciplinary design framework, to address challenges associated with higher fidelity structural design in the conceptual phase" [40]. In 2019, the GENUS aircraft conceptual design environment was developed at Cranfield University using the Java language, and modules for systems, certification, flight testing, reliability, maintainability, and manufacturing were considered in addition to the commonly used modules, i.e., geometry, performance, aerodynamics, weight, and stability [41]. The demand for the design and optimization of novel configurations, such as electric and hybrid-electric propulsion systems, has motivated NASA LaRC in 2018 to develop the LEAPS (Layered and Extensible Aircraft Performance System) in Python language to replace FLOPS, which was in use for the past few decades [42]. A framework named DELWARX (Distributed Design Optimization of Large Aspect Ratio Wing Aircraft with Rapid Transonic Flutter Analysis in Linux) was developed via Python language for the Linux operating system by Khan at Virginia Polytechnic and State University for the MDO (Multidisciplinary Design Optimization) of transonic transport aircraft, with emphasis on distributed computing and transonic flutter analysis [43]. In recent years, many projects for the development of collaborative aircraft design frameworks have been conducted in DLR [44–48]. With a great focus on remote collaboration between disciplinary experts, the AGILE framework is developed by researchers from Europe, Canada, and Russia for aircraft design, analysis, and optimization [49].

The need for an aircraft design framework existed from the beginning of the conceptual design of the TBW aircraft at AUT (Amirkabir University of Technology (Tehran Polytechnic)) in 2010. Initially, spreadsheet tools, stand-alone scripts, and available software were used to perform the sizing, design, and sensitivity analyses. Later in 2015, there was a decision to use an integrated single framework for the design and optimization activities. To this aim, existing and available tools were analyzed according to the published literature, and many of them were examined at different levels of implementation. In most of the investigated tools, it was found that for each disciplinary analysis, a single method is hard-coded in the software, which provides low or no levels of flexibility for implementing an alternative method. The ability to add new methods is critical for developing novel configurations, where empirical methods cannot be used, and the user should provide physics-based methods. Furthermore, the reviewed tool highly relies on internal methods, mostly without any link to external tools. Leveraging the potential of dedicated and validated high-fidelity external tools can improve the accuracy and the applicability of the framework.

In the end, it was decided that a new aircraft design framework should be developed, as none of the existing tools fulfilled the requirements completely. Many reasons contributed to this decision: (1) lack of proper and complete documentation for many of these frameworks; (2) many of them were not easily available or accessible; (3) many of these tools had a close architecture, for which it was not possible to add custom methods and processes; and (4) additionally, there was a strong preference to maintain complete control over the software development process, rather than using an existing tool or participating in an open-source project. For these reasons, the work on the development of LAMBDA (Laboratory of Aircraft Multidisciplinary Knowledge-Based Design and Analysis) started in 2015 with a focus on the design and optimization of the TBW configuration, and currently, the framework has progressed to a stage that performs essential aircraft design and optimization solutions for conventional and novel civil aircraft. LAMBDA is architectured to be a multidisciplinary aircraft design and optimization framework, which has the capability of adding and using new methods of analysis or incorporating multiple levels of fidelity, without modifications to the core code.

This paper aims to present the architecture, modules, functionalities, and applications of LAMBDA. At first, the framework development overall process and software architecture are discussed in Section 2. The analysis and interface engineering modules are presented and described in Section 3. In the end, the result of applying the tool for the design and optimization of many conventional and novel aircraft are presented in Section 4.

2. Framework Development

In the following subsections, the overall process for the development of the framework is presented briefly.

2.1. Framework Requirements

After studying the available tools, a list of requirements was prepared. Many requirements were considered for the development of LAMBDA:

1. **Extensibility**: The framework should be capable of extending the methodologies to user-defined methods, formulations, and processes easily.
2. **Flexibility**: The software should provide freedom for the user to choose from a variety of solvers, formulations, and methods for a specific problem.
3. **Multifidelity**: The framework should be capable of employing different levels of fidelities at a single design job.
4. **Usability**: The framework should be usable not only by framework developers but also by any aircraft designer.
5. **Modularity**: The framework should have a modular architecture that allows for the addition of new modules and updates of existing modules.
6. **Integrability**: The framework should be able to integrate both COTS (Commercial Off-The Shelf) and in-house developed design, analysis, and optimization tools.
7. **Consistency**: To ease future and independent development, a consistent coding style, naming convention, data architecture, and behaviors should be used.
8. **Accuracy**: The framework should provide estimations that are as accurate as possible.
9. **Scalability**: The framework should effectively utilize and exploit parallel computing.
10. **Diversity**: Though a single language may be used for the core engine of the framework, it should be possible to employ and use other languages in the framework development.
11. **Efficiency**: The framework should provide acceptable results within a reasonable time unless high-fidelity methods are selected.
12. **Applicability**: The framework should provide real-world usefulness and should be applicable to practical problems in aircraft design.
13. **Adaptability**: The framework application should not be limited to a single configuration or technology, and it should be possible to evaluate different configurations and technologies.
14. **Visuality** The framework shall enable users to automatically or manually generate various types of plots and provide customization options for the plot attributes.

2.2. Framework Architecture

The framework architecture has been subject to many changes since the beginning of the framework development. The current architecture allows for high levels of modularity and flexibility, which enable the addition of new modules and methods easily. The architecture consists of five layers (see Figure 2). The user provides the primary input file, which may include links to custom methods and data that will override the default methods. The user interface has two primary duties: to process and validate the input files, and to prepare the output files according to the user requirements. The core engine module handles the data transfer between the internal modules and defines the sequence of their operation depending on the requested analysis. Internal modules are responsible for providing the required technical data, and they can employ internal methods or use one or many external tools. Each external tool has an interface module, which translates the data to/from the external tool. Outside of the framework, the required external tool should be available on the computing machine.

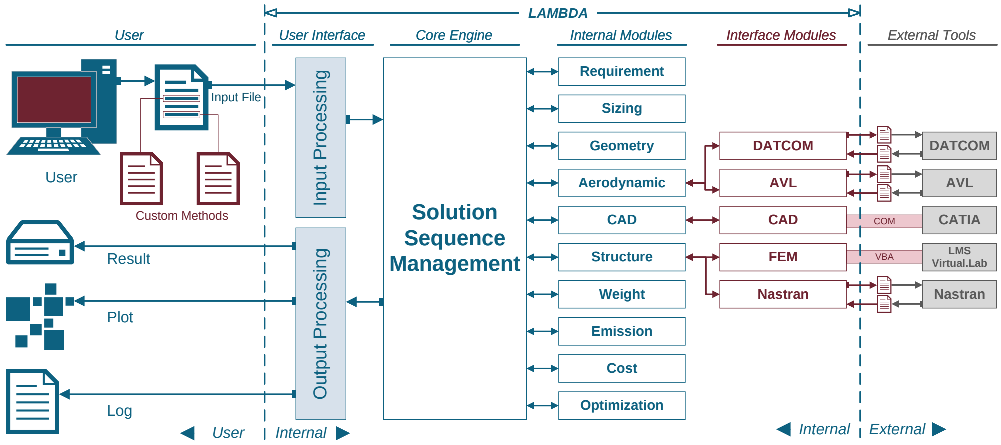


Figure 2. Framework architecture.

2.3. Code Language

Existing frameworks have used a variety of languages for this aim, such as Fortran [14], Java [41], Python [42], and MATLAB [28]. Among these tools, MATLAB and Python have been used widely in academia and industry for engineering analysis. In the end, it is decided to use MATLAB [50] for many reasons: (1) MATLAB provides robust numerical computing capabilities; (2) MATLAB provides a vast array of specialized toolboxes for various engineering disciplines; (3) MATLAB excels in data visualization, offering a wide range of plotting functions and visualization tools; and (4) MATLAB is available on multiple platforms, including Windows, macOS, and Linux.

2.4. User Interface

Two major options were available for the user interface: GUI (Graphical User Interface) and TUI (Textual User Interface). Though the GUI approach is sometimes easier to learn and use, it is decided to implement Namelist TUI for the framework. In this approach, all inputs and commands to the software are written in a text file with a specific format that is understandable to the framework. The benefits of this approach are: (1) Namelist inputs can be easily created and modified; (2) the Namelist file can be scripted programmatically for batch processing; (3) Namelist inputs are easy to version control; (4) text-based input files contribute to the reproducibility of simulations; (5) Namelist inputs might be more efficient than navigating through a GUI in large-scale simulations; and (6) Namelist inputs are platform-independent. In addition to the input file, the framework provides CLI (Command Line Interface) in which it is possible to send direct commands to the framework. Furthermore, the framework creates a comprehensive log of events in a text file, which is helpful for debugging. The results of the simulation are stored in local files, for which it is possible to retrieve the results later.

2.5. Object-Oriented Programming

Initially, functional programming was used for the development of the framework, but later, OOP (Object-Oriented Programming) was selected, and previous functions were converted to classes. The OOP has enabled the efficient application of complex data structures and the exchange of data handles between the modules.

2.6. Data Exchange

Since the framework will be used for MDO, which includes many disciplines, it is very important to have consistent and robust data exchange within the framework. LAMBDA uses the `Aircraft` object handle for data exchange among different disciplines. In this approach, the object `Aircraft` is created when the solution starts, and each discipline updates the related properties inside the `Aircraft` using functions and methods. This method: (1) ensures that each module has access to the latest data; (2) enables faster processing; (3) reduces input/output processing overhead; (4) simplifies data sharing in parallel computing; (5) supports more complex data structures; and (6) makes debugging easier.

2.7. Execution Sequence

When the user calls the program for a job, a sequence of high-level steps is executed by the framework, as presented in Figure 3. The first step is to ask the user for the input file. This input file contains all the required information for the job: solution type, save options, plot options, print options, aircraft model, analysis methods, fidelity levels, solution convergence parameters, and optimization setup. The input file is then parsed, and the correctness of input commands is checked and validated. The parsing process also converts the user commands to commands that are understandable for the framework. Depending on the selected solution, the required solution class is called, which contains references to required analysis modules. When the solution is completed, and depending on the save and export options defined in the input file, the job data are saved to binary files, the summary of the solution is exported to text files, and the generated plots are printed to image files.

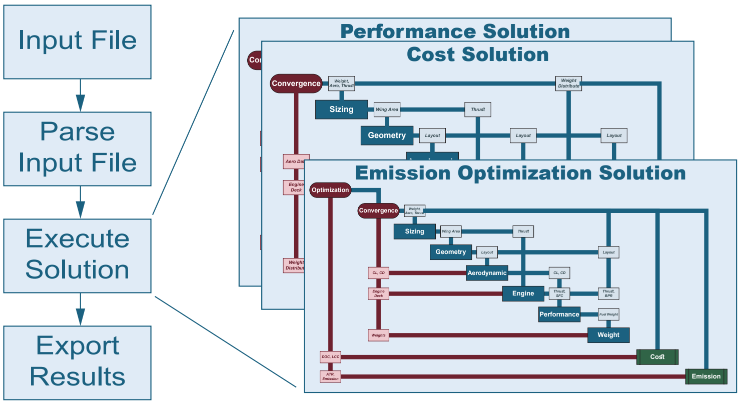


Figure 3. Framework execution sequence.

2.8. Open Architecture

The LAMBDA features an open architecture, which enables the addition and integration of new methods and modules easily by both developer and user. The developer can add new classes and methods to the program directory, and it would be possible to access the `Aircraft` object within that class and update the properties. More importantly, the user can develop their own scripts and request the framework to use those scripts for the job. The call to these customized scripts is made through the input file, and the framework will use the user scripts instead of built-in functions and methods.

2.9. Dependency Management

From the literature review, it is clear that two approaches exist for the integration of external tools:

1. **Low-Dependency**: In this approach, the disciplines are developed within the framework, which means analyses (such as geometry, structure, and aerodynamics) are conducted using codes that are part of the framework. For instance, for the implementation of the high-fidelity geometry module, a set of codes is needed to model all the required geometrical shapes (points, curves, surfaces, etc.) and geometrical operations (intersection, multisection, sweep, etc.) within MATLAB. Then, it would be possible to export the resulting geometry in standard formats, such as STEP or IGES. Additionally, at the end of the development phase, extensive testing and validation of this tool is needed.

2. **High-Dependency**: In the second approach, proven high-fidelity external tools are used, and the data are transferred between the framework and the tool. For example, for the geometry module, the modeling is conducted via a validated external tool, such as CATIA, and only the interface protocol is developed within the framework. Though many programming languages may be used, a considerably lower amount of effort would be needed when compared to the development of a new high-fidelity tool.

Since the current framework is intended to be used for high-fidelity optimizations, and to reduce the development and testing time, the second approach is selected. In the process of selecting the external tools, extreme care was taken to select those that have suitable automation capabilities (with available documentation) and provide the required functionalities with a high level of stability.

2.10. Coding Style

Using a consistent coding style throughout the whole framework has been tried, which includes strict variable naming conventions, class description, code commenting, file naming conventions, and providing examples in classes and functions. Each module, class, or function has three sections: input data, calculations, and output data. In the input data section, input variables are validated, default values are assigned (if no input is provided for that variable), and the required data are assigned to local variables.

3. Framework Modules

Two types of modules are defined in LAMBDA: Analysis modules and Interface modules. In the following subsection, a brief introduction to the modules is presented in the follwing subsections.

- **Analysis Modules:** These modules are used to perform the analysis and computation. The results of these computations and analyses are used to update the `Aircraft` properties. Core analysis modules of the LAMBDA are `Requirement`, `Weight`, `Sizing`, `Geometry`, `Aerodynamic`, `Engine`, `Performance`, `Cost`, `Emission`, and `Optimization`.
- **Interface Modules:** Interface modules are developed to send commands and receive information from external tools and software that are used by the framework. The development of these interfaces, which are mainly implemented in MATLAB, requires knowledge of the automation interfaces of the target software. The interface modules are called by the analysis modules to perform specific types of analysis (most of the time for high-fidelity methods, such as Nastran for structure analysis), and the outputs from interface modules are used by analysis modules.

3.1. Requirement

Since all required information, inputs, and options are defined inside the input file before the execution of the code, there would be no interaction with the user during the code processing. The requirement module processes the top-level aircraft requirements, which are defined by the user, and develops the requirements' data structure to be used inside the solution. This module parses the input file, detects user commands, and validates the inputs of the command.

3.2. Solution

The `Solution` module controls the sequence and order of the execution of different disciplines and the exchange of data among them. As an example, the sequence of execution for `Cost Analysis` is presented in Figure 4. Many solutions have a convergence loop that is required for solutions where some of the input data to one or many analysis modules are updated in the downstream modules. The converge loop is executed until there is no change in key parameters, and a threshold of convergence is defined by the user for this aim. For the first iteration, initial values are assigned to the required variables. For example, in solution `Performance Sizing`, the weight information is needed in the `Sizing` module, while this information is updated in the `Weight` module, which is called after the `Sizing` module.

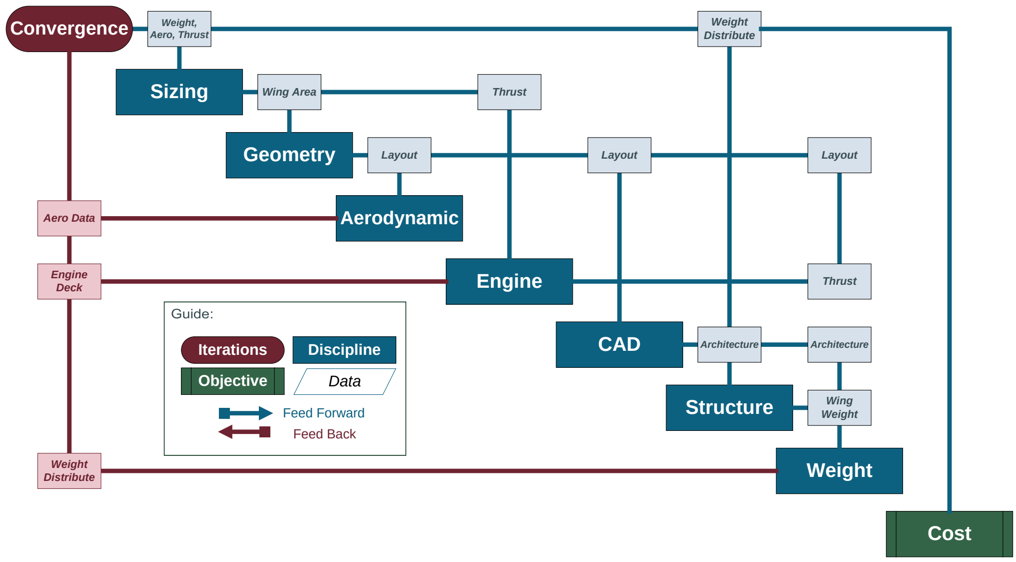


Figure 4. Cost analysis solution.

3.3. Sizing

The `Sizing` module calculates the required engine thrust and wing area to fulfill the performance requirements, which are based on the mission and Part-25 requirements. The static sea-level required engine thrust, T_{sls}, and the wing area, S, are computed using the matching diagram. In the first iteration of the sizing process, since the aircraft layout is not defined yet, and no aerodynamic information is available, a guess of the required thrust and wing area is made, which is simply a constant value of T/W and W/s obtained from similar aircraft. In subsequent iterations, the classical formulation of the matching diagram presented by Roskam [11] is used. The user can select which of these requirements should be considered in the sizing process and can adjust the requirements' values and settings. The overall process for the sizing module is presented in Figure 5.

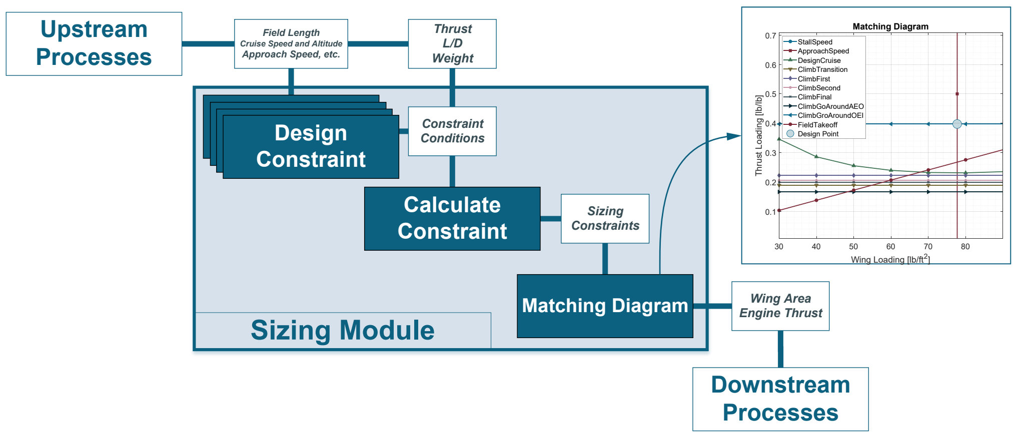


Figure 5. Sizing module process.

3.4. Geometry

The design of the geometrical components (wing, fuselage, nacelle, horizontal tail, and vertical tail) layout is carried out using statistical and engineering methods, such as [51]. The `Geometry` module defines and calculates the aircraft geometrical parameters, which can later be used for high-fidelity geometry modeling in the `CAD` module (see Section 3.7) and any other modules that need geometrical properties, e.g., the `Aerodynamic` (see Section 3.6) module needs the wetted area for drag calculations. The aircraft components are modeled using a few classes, such as `Wing` class for wings and tails, and `Fuselage` class for fuselage and nacelles.

3.4.1. Wing

This class of geometries is used to define the shape of the wing, horizontal tail, vertical tail, and canard. Each wing consists of many sections and a segment between each pair of consecutive sections. The wing section parameters are the longitudinal, lateral, and vertical location of a reference point along the chord, chord length, incidence angle, and the airfoil of the section (see Figure 6). The user can define symmetrical control surfaces (such as elevators), asymmetrical control surfaces (such as ailerons), and high-lift devices (such as flaps and slats) on each of the wing-like surfaces. If the wing is equipped with truss and struts, the user can define the location and planform of these struts. Based on the input planform, and if wing area sizing is requested by the user, the wing geometry is scaled to match the required wing area. In addition, the design of the wing planform can be linked to the cruise lift coefficient and Mach number, which is useful for optimization jobs where the cruise Mach number is a design variable. The developed method for wing design is presented in Appendix A.

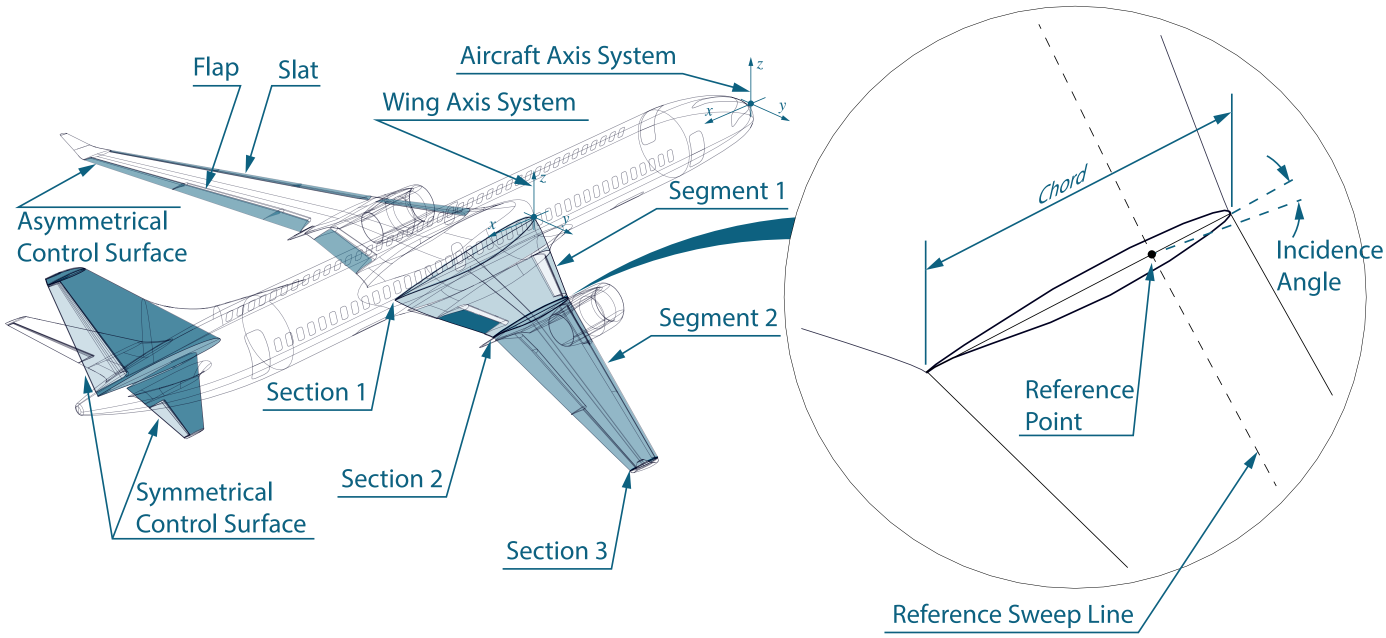


Figure 6. Wing geometry parametrization.

3.4.2. Fuselage

This class of geometry can be used to define the external shape of fuselage and nacelles. The geometry of the fuselage consists of sections placed longitudinally and segments between each pair of consecutive sections. The parameters that define the fuselage section are the longitudinal, lateral, and vertical location of the center point, width, and height of the section (see Figure 7). The above methodology allows for the calculation of the wetted area of each segment and the cross-sectional area of each section, which are required in many empirical and analytical aerodynamic methods. Similar to the wing, it is possible to define the fuselage planform manually by user inputs or to design based on the number of passengers.

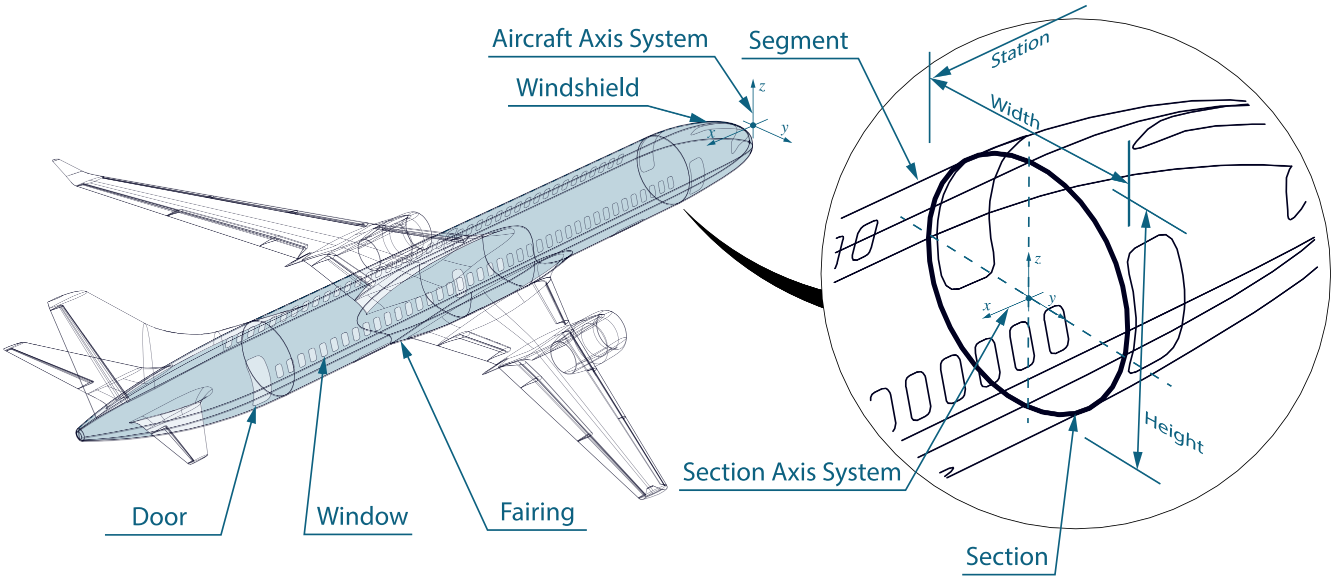


Figure 7. Fuselage geometry parametrization.

3.4.3. Tail

As mentioned in Section 3.4.1, tails are instances of the wing class, and they inherit the geometrical properties and methods. However, the horizontal tail and the vertical tail area are dependent on the wing area and are sized using the tail volume coefficient. The values of the tail volumes are selected based on the aircraft database. The tail location is dependent on the fuselage layout and is placed in the most aft possible position. The vertical tail is placed slightly in front of the horizontal tail to avoid rudder blanketing by the horizontal tail. The required CG (Center of Gravity) limits are calculated from the weight module (see Section 3.10.2). By default, the planform design is carried out using the method provided in [12], while the user can integrate custom functions with the framework for tail sizing and planform design.

3.4.4. Cabin

Based on the defined number of passengers, the cabin layout is defined, and the cabin length will be used to design the fuselage planform.

3.4.5. Aircraft

A transformation, i.e., translation, rotation, reflection, or dilation, can be applied to each of the wing or fuselage components. Theoretical formulas are used to calculate the aircraft reference dimensions, i.e., reference area, chord, and span.

3.5. Engine

The `Engine` module prepares the thrust and SFC (Specific Fuel Consumption) values in the whole flight envelope. Three approaches are available for generating the engine performance charts: (1) **Direct Input**: The engine data are tabulated in text files, and the addresses of these files are added to the input file. (2) **Engine Scaling**: The "rubber engine" concept [52] is incorporated, and the thrust and SFC are scaled from a baseline engine. (3) **Engine Sizing**: The engine cycle is sized to the required thrust and the pre-defined top-level technological requirements of the engine, such as BPR (Bypass Ratio), OPR (Overall Pressure Ratio), and TIT (Turbine Inlet Temperature) [53,54]. The results of engine sizing are used for the off-design analysis, and the variation of SFC and thrust versus Mach number and altitude are evaluated [55]; see Figure 8.

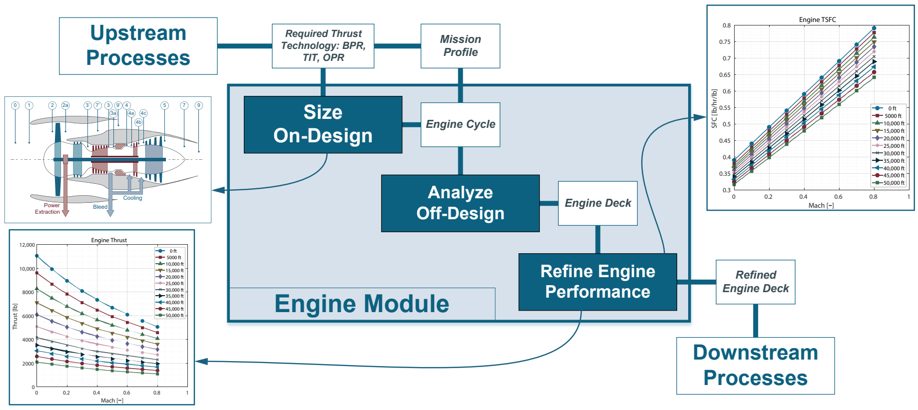


Figure 8. Engine module process.

3.6. Aerodynamic

The `Aerodynamic` module calculates the aerodynamic coefficients, which include lift and drag coefficients at different AOAs (Angle of Attacks) and Mach numbers. In this module, initially, the baseline aerodynamic characteristics are calculated, and the engineering corrections are added to account for the additional features; see Figure 9. Currently, many different methods are available for the baseline aerodynamic analysis: (1) **Direct Input**: The user provides the aerodynamic coefficients in a tabular format to the framework, and those data will be used for the analysis. This method is helpful when the source of the aerodynamic data is external, i.e., the data are the result of high-fidelity CFD simulations or wind tunnel experiments. (2) **Engineering Methods**: A set of engineering formulas, which are available in the literature, are used to calculate the aerodynamic coefficients. Since DATCOM [56] can provide a relatively complete set of aerodynamic data using engineering methods, a dedicated interface module is developed to prepare the input and post-process the output of DATCOM. For items not included in DATCOM, such as trusses, empirical methods provided in [57] are used to estimate the drag. (3) **Low-Fidelity Numerical Methods**: These types of methods, such as VLM (Vortex Lattice Method), are suitable for the estimation of subsonic lift and induced drag characteristics. An interface module is developed to calculate the aerodynamic characteristics using one of the widely-used examples in this field, namely, AVL (Athena Vortex Lattice) [58]. Since VLM methods cannot compute the parasite drag, engineering methods, such as DATCOM, are used to compute the parasite drag. (4) **High-Fidelity Numerical Methods**: Works are in progress to add a CFD method as a means to calculate the aerodynamic characteristics.

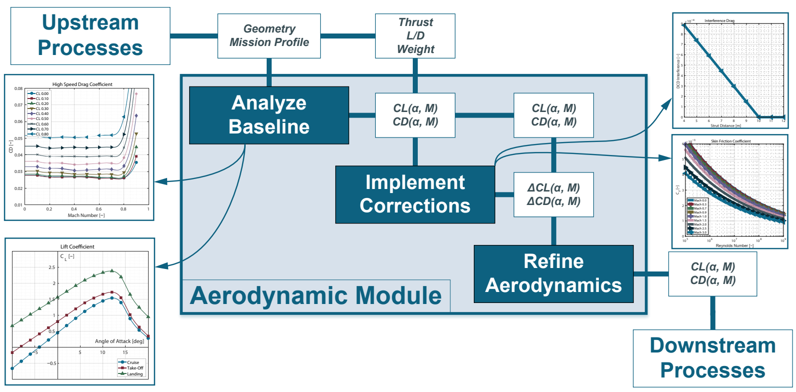


Figure 9. Aerodynamic module process.

3.7. CAD

The CAD module is used to update the external surfaces, generate the internal structure architecture, and extract the geometrical information of the aircraft. The overall process for the CAD modules is presented in Figure 10. The geometrical architecture is used for the finite element modeling (see Section 3.8.2), and the extracted geometrical information is primarily used for the load analysis (see Section 3.8.1). In the present research, the CATIA [59] modeling tool is used in an automated process to achieve this goal [60]. The CAD module incorporates an interface module for CATIA through the COM (Component Object Model) interface, to update the aircraft geometry, and extract the numerical measurement from the model. The interaction through this interface is conducted via code commands, which are based on VBA (Visual Basic for Applications) language. This interface can generate CATIA documents, update the geometry in CATIA according to aircraft geometry in MATLAB, and extract geometrical measurements from the CATIA file, such as fuel tank volumes and their CG.

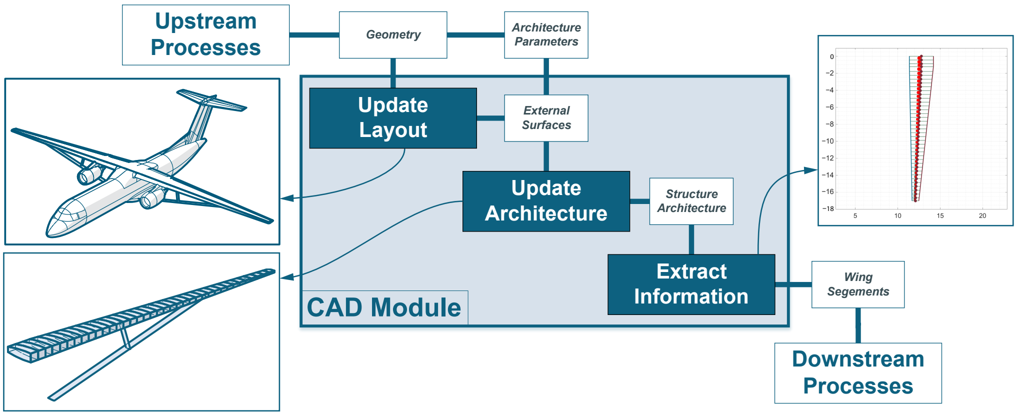


Figure 10. CAD module process.

3.8. Structure

The `Structure` module is mainly responsible for the estimation of structural weight using high-fidelity methods. As can be seen from Figure 11, this module first employs the `Load` module (see Section 3.8.1) and FEM module (see Section 3.8.2) to prepare the initial finite element model. By having this information, and depending on the requested result, the structure is sized according to strength (Section 3.8.3), stiffness (Section 3.8.4), and stability (Section 3.8.5) requirements.

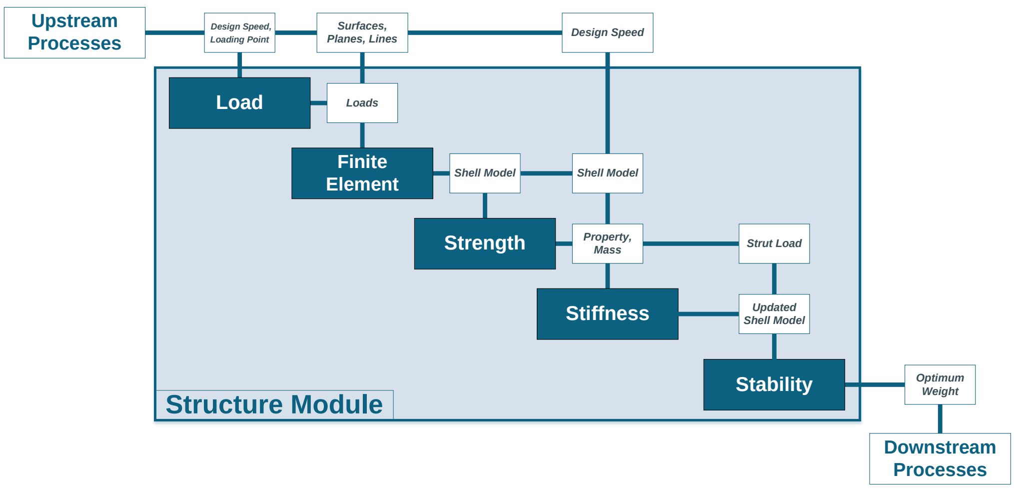


Figure 11. Structure module process.

Nastran [61] is used as the core structure analysis tool. In the developed structure process, many structure variants of the original FEM are required, which need an interface module to read the Nastran input database and prepare the input file for Nastran execution. To this aim, a comprehensive interface module for Nastran is developed in MATLAB. This module can interpret Nastran input files (BDF) and modify the grids, coordinate systems, properties, materials, loads, constraints, and load cases. Also, it can read the Nastran output file (H5) and extract grid displacements, balance forces, constraint forces, and element stresses.

3.8.1. Load Analysis

To calculate the load distribution over the wing, an automated process is developed to provide the required load data for other processes. This module interacts with an existing and validated tool, namely, AVL, for aerodynamic trimming and load distribution. The developed code is capable of calculating hundreds of load cases in just a few minutes with conventional home computers. The overall process of the load analysis (see Figure 12) includes load cases, load analysis, load discretization, and load filtration. To restrict the computational time, only those load cases that are critical for the wing sizing are considered using a two-dimensional load filtration technique. Interested readers are referred to [62] for further details.

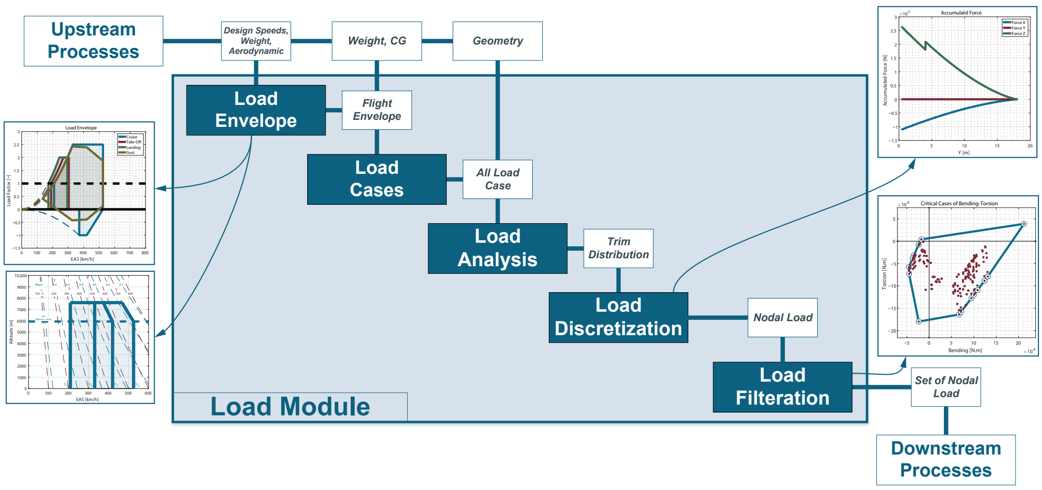


Figure 12. Load module process.

3.8.2. Finite Element Model

If the high-fidelity FEA (Finite Element Analysis) is selected for the wing weight estimation, the FEM of the wing is needed, which should be generated from the CAD model. The modeled wing main box structure is presented in Figure 13 and consists of panels, ribs, and spars. In order to reduce the number of variables and the computational time during optimization, the skin panels are idealized using smeared stringers [63]. To achieve this goal, a set of scripts in VBA language within the "LMS Virtual.Lab" tool version 11-SL2 [64], which is a plug-in to CATIA, is developed. These scripts can create the grid, properties, materials, loads, constraints, and cases of the FEM. The whole process of creating the FEM is commanded by MATLAB. The critical load cases, which are calculated in the load analysis, are applied to the loading points of the finite element model. In the last step of this process, the Nastran input file is generated in BDF format.

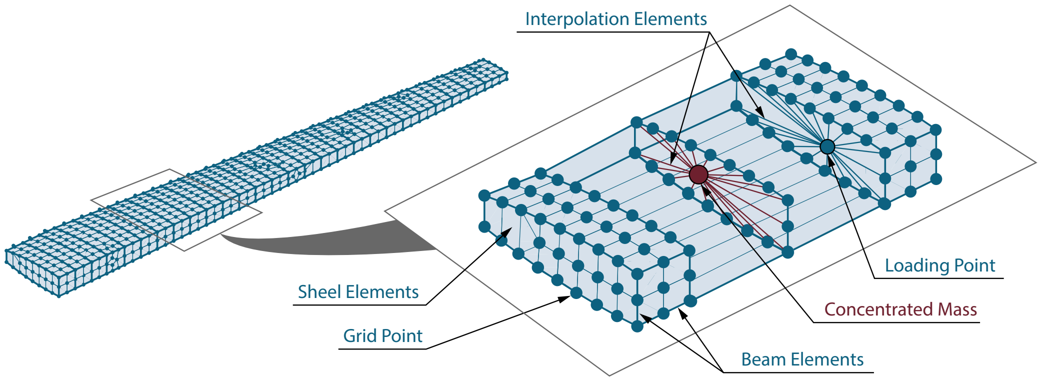


Figure 13. Definition of the finite element model.

3.8.3. Strength Sizing

The structure sizing is employed to calculate the minimum required material mass for a given wing planform when subjected to the previously calculated external loads. The process of structure strength sizing is presented in Figure 14. In this process, the created FEM, which included a static solution, is converted to an optimization solution. The design variables are the thicknesses of the upper and lower panel's skin, the thicknesses of the front and rear spar's web, and the diameters of the spars' caps. To avoid complexity within the framework, the bounds of the design variables are independent of the aircraft and are suitable for both large and small aircraft. The optimization process employs a gradient-based algorithm by calling Nastran solution 200 (`SOL 200`). Gradient-based algorithms are suitable for large design spaces and provide rapid convergence [65]. Specifically, the MSCADS algorithm, which is a customized iteration of the publicly accessible Automated Design Synthesis (ADS) tool [66], is selected from the range of available options for this purpose. The updated properties are extracted from the optimization results' files and are used to update the properties in the original model, and the final model is saved to disk.

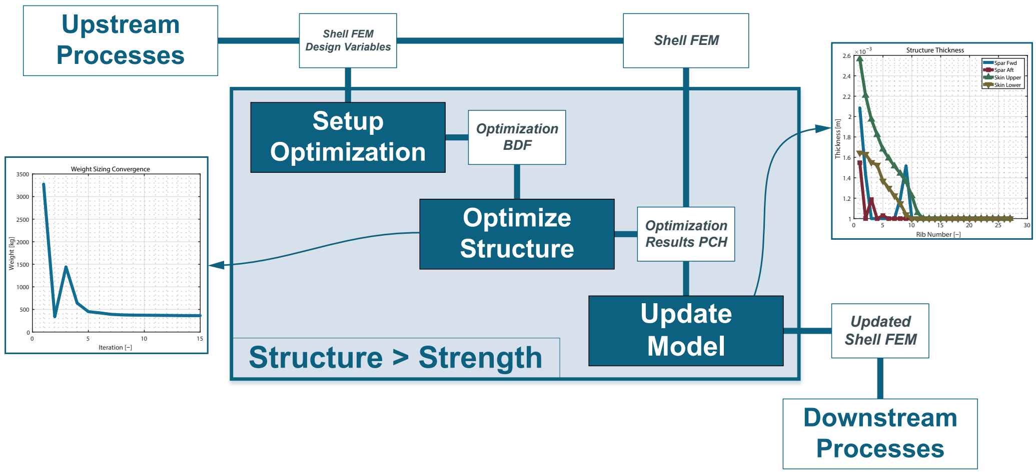


Figure 14. Strength sizing process.

3.8.4. Stiffness Sizing

In some analyses, the wing structure needs to be sized according to the flutter requirements. For this purpose, two modeling approaches can be used: the beam elements [67] or shell elements [68]. Generally, beam models have simpler geometry and fewer degrees of freedom (usually four orders of magnitude), which reduces the computational time and makes them suitable for conceptual design frameworks [69]. For this reason, the beam model is used for sizing according to the stiffness requirements. For employing the beam model, the shell model should be converted to an equivalent beam model, and this process is called condensation; see Figure 15. To condense the wing box shell finite element model, a method consisting of three sequential steps is employed: geometry condensation, stiffness condensation, and mass condensation. The condensation process was validated by comparing the deflections of the beam model against the deflections of the shell model under equal loading conditions. The resulting beam model is used for stiffness sizing and optimization.

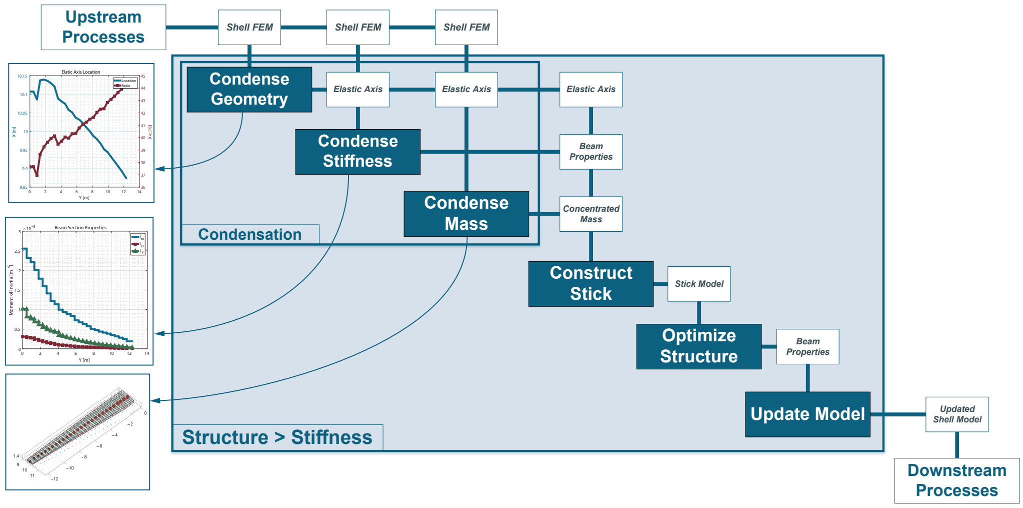


Figure 15. Stiffness sizing process.

In the optimization setup, and for each segment (between two ribs), a design variable is defined. This variable represents the scale factor for the segment skin thicknesses, i.e., spar web and panel skin thicknesses. The lower bound of this design variable is 1.0, which means the thickness will not be less than the previously designed thickness. The constraint is to have no flutter below 1.15 of the dive speed (15% margin) [70]. Similar to the strength sizing, the objective is to minimize the weight. Nastran's Design Sensitivity and Optimization solution sequence (SOL 200) in conjunction with Nastran's Flutter solution sequence (SOL 145) are used for optimization and sizing purposes. Since in hollow thin-walled structural sections the second moment of inertia is a linear function of the thickness [71], the relation between the design variables and the second moment of inertia of the section is also linear. After the flutter sizing, all thicknesses for each segment are multiplied by the respective design variable, and the shell finite model is updated.

3.8.5. Stability Sizing

In the current implementation, the Euler buckling method is used to estimate the required structure thickness for the strut and jury strut of the TBW configuration, and the work is in progress to include the shell buckling in the sizing of wing panels. To size the strut according to buckling, the grid point loads are extracted from the FEA results for each truss member at each load case; see Figure 16. The maximum value of the compressive loads is used to calculate the required second moment of area of the section. Based on the required moment of area, the required thickness of the strut is calculated, and the finite element model is updated.

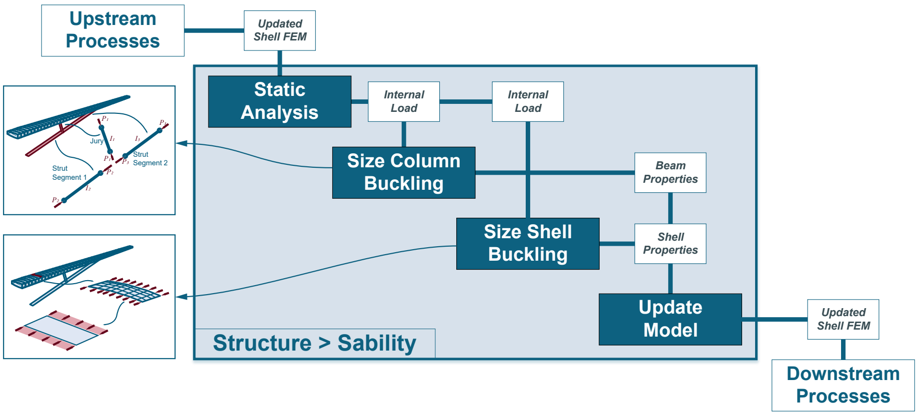


Figure 16. Stability sizing process.

3.9. Performance

The `Performance` module calculates the performance characteristics of the aircraft, as presented in Figure 17. This module includes submodules for: (1) **Mission Analysis**: The required fuel to complete a custom mission, which is defined by the user, or the achievable range with a fixed amount of fuel is calculated. To this aim, the aircraft trajectory is discretized in each phase, and the 3-DOF (Degree of freedom) performance equations are solved to obtain the trim conditions, i.e., AOA and throttle. By having the throttle and the engine performance chart, the fuel flow is computed at each segment, which can be numerically integrated to obtain the consumed fuel. (2) **Payload-Range Diagram**: The aircraft mission performance is evaluated at corner points of the payload-range diagram, and the payload-range plot is generated. (3) **Flight Envelope**: The aircraft performance limitations in terms of altitude and speed are calculated, and the flight envelope ($V - H$) is plotted. To this aim, the whole range of speed (V) and altitude (H) is discretized, and the 3-DOF performance equations are solved at every combination of speed and altitude to obtain the trim conditions. At each point, the specific excess power (P_s) is computed, and the limits of the flight envelope are the locus of points where the specific excess power is zero.

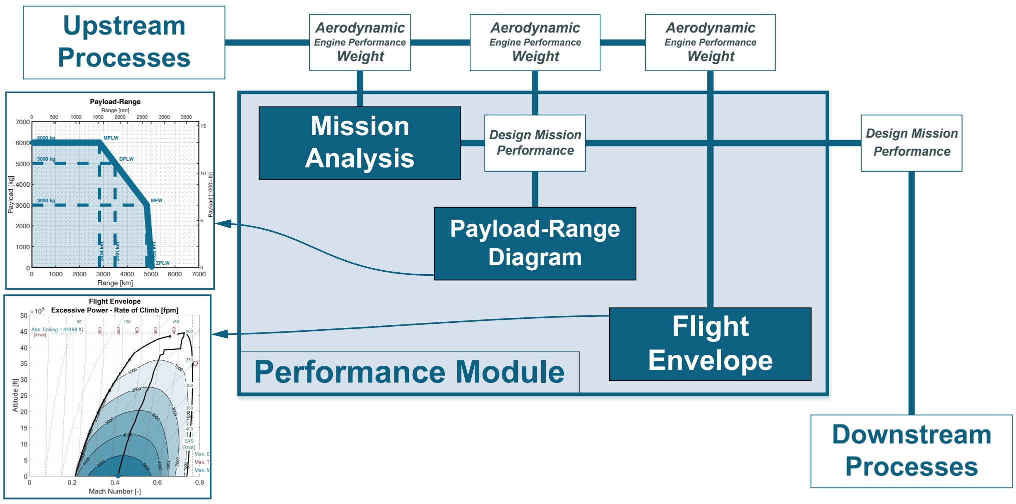


Figure 17. Performance module process.

3.10. Weight

The `Weight` module has many functionalities, which can be requested by the user: Weight Breakdown (Section 3.10.1), Weight Limitation (Section 3.10.2), and Weight Distribution (Section 3.10.3). The process for weight analysis is presented in Figure 18.

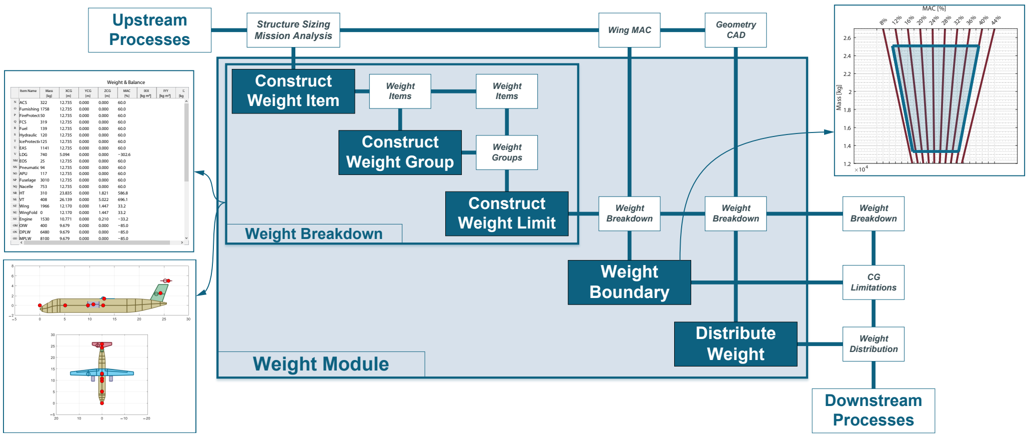


Figure 18. Weight module process.

3.10.1. Weight Breakdown

The purpose of the weight breakdown analysis is to calculate the buildup of the MTOW (Maximum Take-Off Weight). One of the challenges encountered during the implementation of the weight buildup approach was the definition of weight hierarchy and breakdown tree. The methodologies provided in textbooks and references (such as from Roskam [72] or Torenbeek [12]) may use different weight breakdowns based on the application, and if only one or a few of these methods are implemented in the framework, this will limit the application of the framework for future cases. To overcome this issue, it was decided to implement a fully customizable and flexible weight hierarchy definition. In this approach, three classes of weight are defined (see Figure 19): (1) **Weight Items** are the most basic category in the weight hierarchy, and it is possible to calculate their weight and CG using different methods, such as statistical (weight fraction), empirical (based on geometry and mission), and numerical (FEA and simulations) independent of the method used for other weight items. (2) **Weight Groups** are collections of weight items or groups (a weight group can contain other weight groups as well), and their mass and CG are calculated from their constituting weight items and groups. (3) **Weight Limits** are independent of the weight hierarchy and can be used for all types of aircraft; typical examples are MTOW and MZFW (Maximum Zero-Fuel Weight). This approach will make it possible to use different levels of fidelity for different weight items, e.g., to use Torenbeek's [12] formula for fuselage weight and use FEA for wing weight.

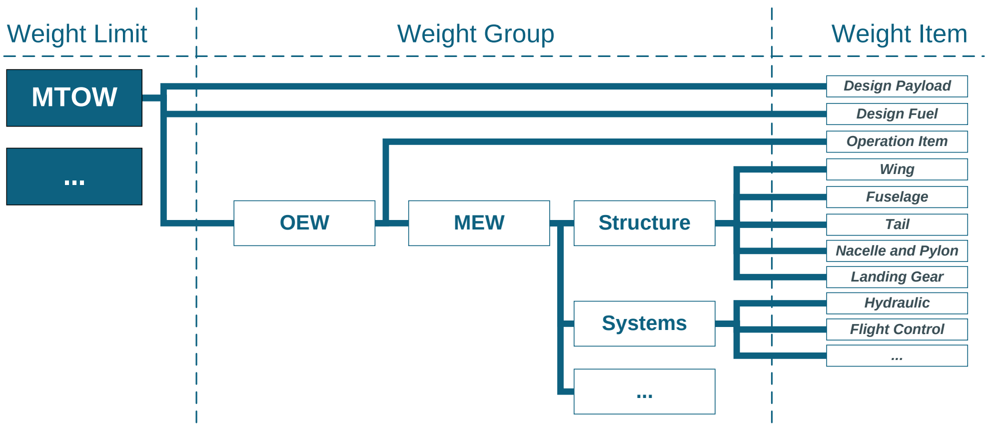


Figure 19. Weight breakdown architecture.

The OEW (Operating Empty Weight) is the sum of the structure, powerplant, systems, interiors, and operational items' weight. Though it is possible to use any customized method to estimate the weight and center of gravity of weight items, by default, a combined method of General Dynamics and the Torenbeek method from Roskam [72] is used. The wing structural weight can be computed from high-fidelity structure analysis. The effect of bypass ratio on engine weight was modeled using the method provided by Jenkinson [73].

3.10.2. Center of Gravity Limitations

This function calculates the CG limitations, which can be used later for other processes, such as tail sizing. The input to this module can be a manual input, in which the corner points of the CG envelope the chart. Alternatively, the CG envelope can be calculated by analyzing different passenger/payload/fuel loading scenarios and finding the outermost CG positions. The work for implementing the second approach is in progress.

3.10.3. Weight Distribution

If the wing load analysis is requested to perform wing weight sizing using FEA, the weight distribution across the span for the wing is required. The weight distribution is the mass and CG of each zone between each two consecutive ribs. This function extracts the position of wing ribs from the CAD module, and based on the calculated wing weight, the mass and CG of each wing bay are calculated.

3.11. Emission

The `Emission` module calculates the emissions of aircraft in terms of emission mass and temperature response (see Figure 20), and these parameters can be used in the optimization process as objective functions. The accounted species are Carbon Dioxide (CO_2), Water vapor (H_2O), Sulfate (SO_4), Nitrogen Oxides (NO_x), Methane (CH_4), long-lived Ozone (O_{3L}), short-lived Ozone (O_{3S}), Soot, and contrails. The engine emission is characterized using the parameter "Emission Index", which is the mass of produced emission per 1 kg of consumed fuel. By having the emission index at each point during the flight, it is possible to calculate the produced mass of emissions in each phase and, subsequently, in the whole mission. The emission of the aircraft throughout its life cycle can be characterized using the temperature response. The temperature response is represented by ATR (Average Temperature Response) [74,75], which has units of temperature (such as mK) and shows the average increment in the surface temperature of the earth due to pollution of the aircraft over several years. The details of the implemented formulations to calculate the emission index and temperature response are presented by the authors in a previous publication [55].

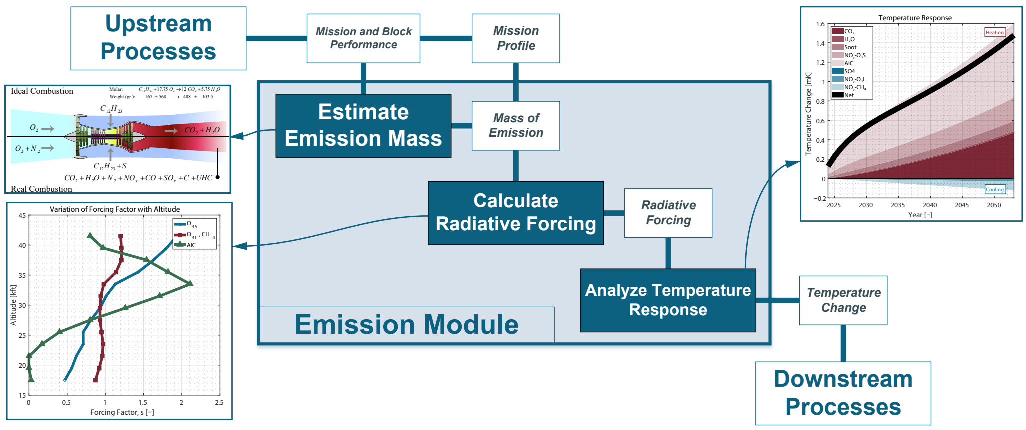


Figure 20. Emission module process.

3.12. Cost

The `Cost` module is performed to calculate the operating and life cycle costs. These costs can be used as objectives in the optimization problem. The aircraft cost analysis is mainly based on the method presented by Roskam [76], and minor modifications are implemented. In this method, the aircraft life cycle cost is divided into development, acquisition, and operation costs. Based on the operation cost, the DOC (Direct Operating Cost) can be calculated. The DOC is the sum of operational costs: flight (fuel cost and crew cost), maintenance, depreciation, fees, and finance costs. The cost module is capable of distributing the cost over years of operation and production, which would make it possible to find the break-even point.

3.13. Optimization

A surrogate optimization methodology incorporating DoE (Design of Experiment), ANN (Artificial Neural Network), and GA (Genetic Algorithm) is implemented in the `Optimization` module. The overall scheme of this module is presented in Figure 21. When high-fidelity analyses, such as weight sizing using FEA, are included, the most time-consuming part of the optimization process is the simulation step. The implemented approach provides the capability of optimizing the aircraft with respect to different single objectives or multiple objectives using a single batch of simulations. Furthermore, it is possible to save the simulation results and, at a later time, to conduct an optimization job with a new objective, without the need for simulating the models. The optimization algorithm, which is called "Metamodel-Assisted Optimization" [77], is suitable for problems where evaluating the objective function is computationally demanding.

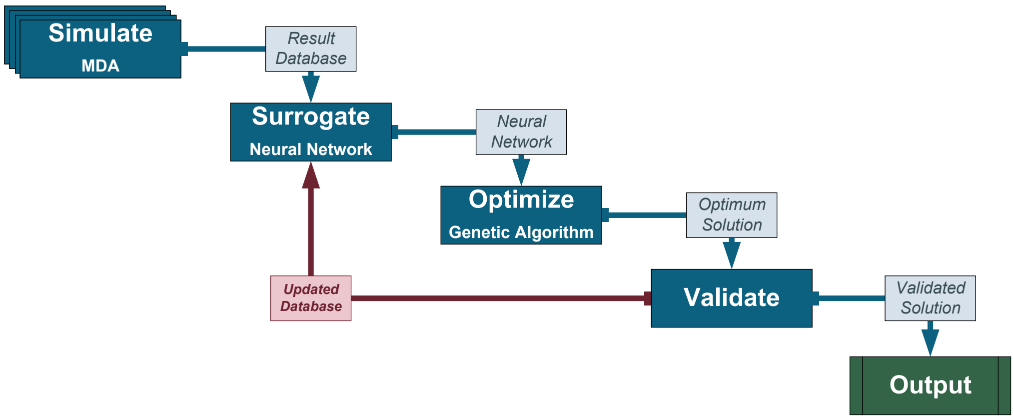


Figure 21. Optimization process.

The DoE is employed to generate the required number of design samples based on the selected design variables and their upper and lower limits. To create the samples, initially, a population of samples is randomly generated using the Latin Hypercube method, and to capture the extreme conditions of the design variables, the four-level full factorial design experiments are also considered. Then, the MDA (Multidisciplinary Analysis) is iterated for each of these samples, and this step is the most time-consuming process. Using these results, an ANN is trained for each of the objective functions. These networks are then used as the evaluating functions in the GA optimization algorithm to find the optimum design point for each objective. Since these optimum points are based on the surrogate model, the optimum points are validated by conducting the MDA simulation at these points. The MDA results are compared with the ANN results, and if the disparity exceeds a predefined threshold, these design points are added to the DoE, and surrogate models are re-trained. This iterative process continues until the ANN and MDA results converge and demonstrate consistency. The implemented algorithm is presented in Algorithm 1. The optimization module can be used for sensitivity analysis, in which the variation of objective functions with respect to the design variables is investigated.

Algorithm 1: Implemented Metamodel-Assisted Optimization

Data: Initial sample size N, maximum number of iterations $maxIter$, convergence tolerance ϵ

Result: Optimal design $\mathbf{x}^*$

Generate N initial sample points $\mathbf{x}$ using DoE;
Simulate sample points $\mathbf{x}$ to evaluate objective functions;
for *each objective* **do**
    while *not converged and iteration* $< maxIter$ **do**
        Train surrogate model $f(\mathbf{x})$ using ANN;
        Find candidate solution $\mathbf{x}_{\text{cand}}$ using GA applied to surrogate model $f(\mathbf{x})$;
        Simulate candidate solution $\mathbf{x}_{\text{cand}}$ to evaluate objective function;
        if *convergence criteria met* **then**
            Optimal design $\mathbf{x}^*$ is $\mathbf{x}_{\text{cand}}$;
            Terminate optimization;
        else
            Update sample points $\mathbf{x}$ with $\mathbf{x}_{\text{cand}}$;
        end
    end
end

4. Framework Application

The developed framework is applied to many civil aircraft design, analysis, and optimization problems. In this section, a few examples of these applications and key results are presented. These applications are:

1. Design of a New Conventional Aircraft;
2. Development of an Affordable Conventional Aircraft;
3. Development of an Affordable TBW Aircraft; and,
4. Optimization of the TBW Aircraft.

In the current version of the framework, novel propulsion architectures are not supported yet, as the implementation of such systems affects the cores of the performance module. In addition to that, currently, it is not possible to design or optimize AAM (Advanced Air Mobility) configurations, as they have additional modes of flight, such as hovering or transition phases. Other modules provide the flexibility for defining novel fixed-wing configurations, such as the C-Wing. In this case, the aerodynamic and weight estimation methods should be provided by the user as add-on scripts, which is possible due to the architecture openness of the framework; see Section 2.8.

4.1. Design of a New Conventional Aircraft

The primary goal of this problem is to design a 92-passenger regional aircraft and calculate the operation costs. The aircraft is a twin-engine aircraft with under-the-wing mounted engines, low wings, 4-abreast fuselage, and conventional tails. The aircraft layout is presented in Figure 22.

For cost analysis, the aircraft mission performance is calculated, for which engine performance and aerodynamic analysis are required. The aircraft weight is calculated depending on the geometry, and engineering methods are used. The aircraft development cost is calculated, which takes into account the aircraft weight and standard assumptions from [76]. The results of the cost analysis are presented in Figure 23.

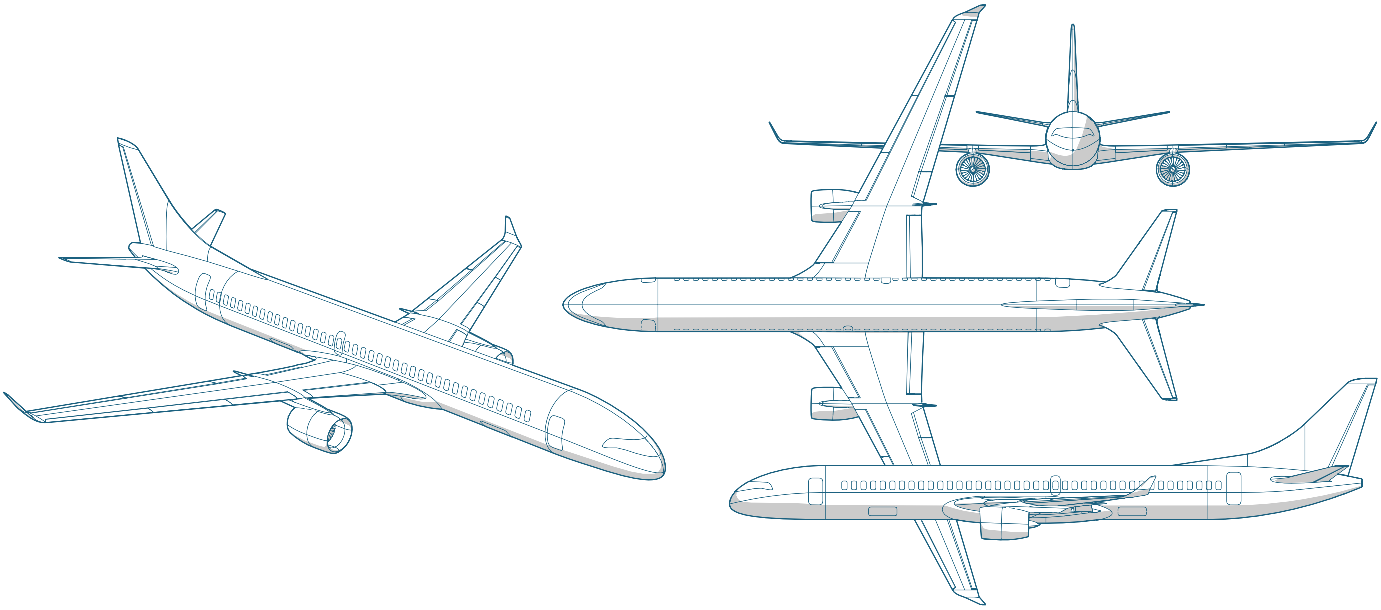
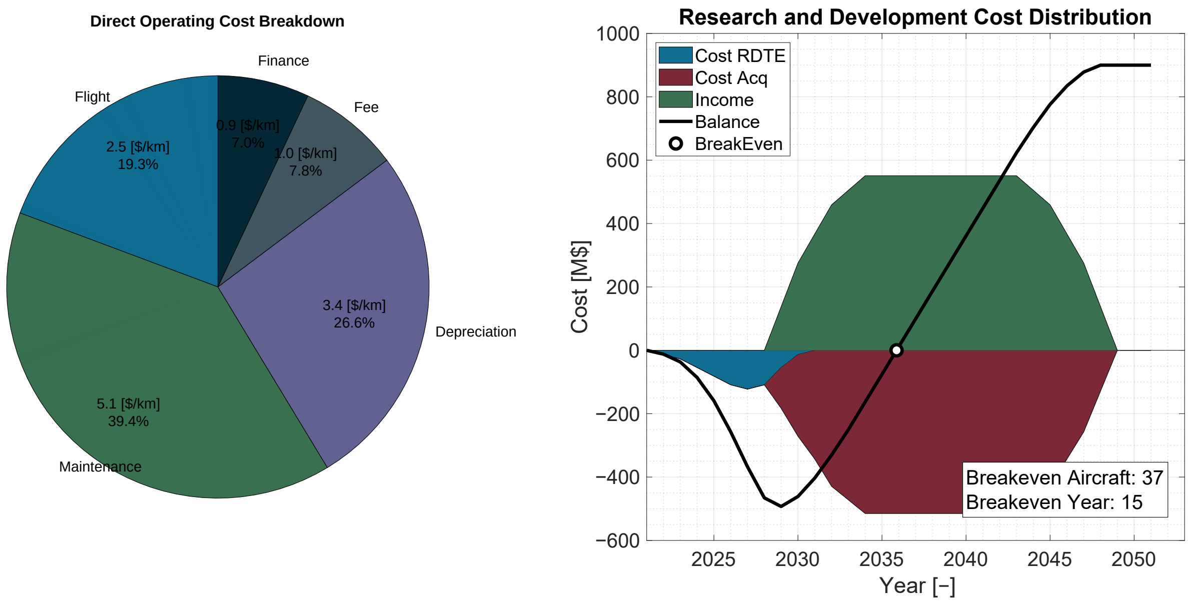


Figure 22. Aircraft layout of a new conventional regional aircraft.

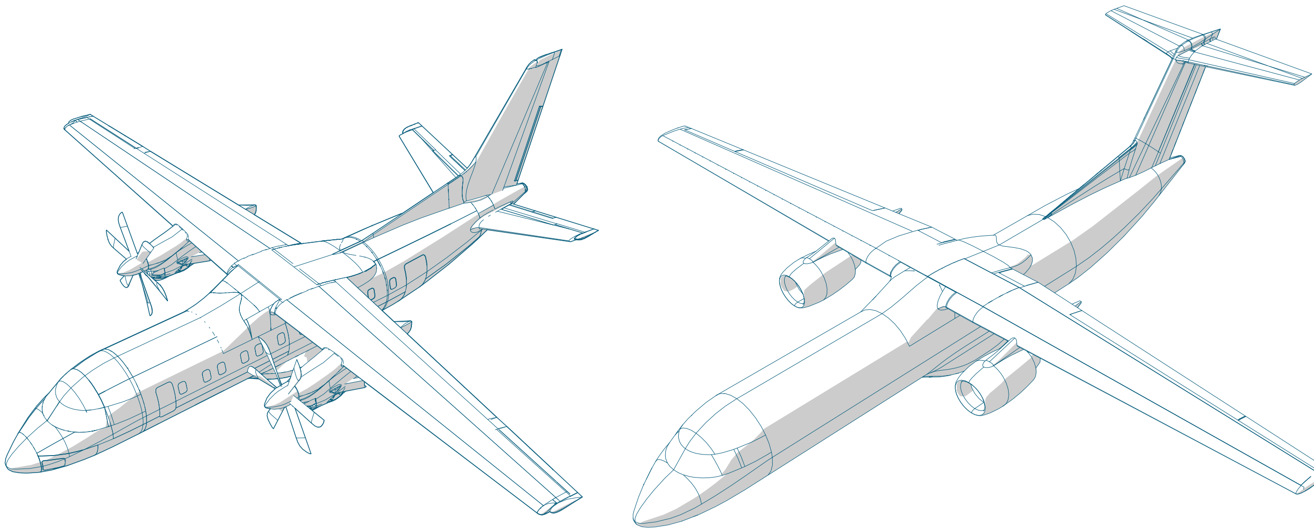


(**a**) DOC breakdown (**b**) Cost distribution over years

Figure 23. Results of cost module for new 92-passenger conventional aircraft.

4.2. Development of an Affordable Conventional Aircraft

In this application, a conventional turbofan-powered aircraft is developed based on a conventional turboprop aircraft. In contrast to the previous application (see Section 4.1), this problem demonstrates the benefits and challenges associated with variant development. In this process, many modules of the base aircraft are reused in the development of a partially new aircraft. The primary motivation behind this approach is to reduce the development cost and time by maximizing the usage of already-certified modules and systems. By using this strategy, the benefits and challenges of an affordable 72-passenger jet-powered aircraft derived from an existing 52-passenger prop-powered regional aircraft (see Figure 24a) are investigated. The engine replacement was carried out in order to achieve better climb performance, higher cruise speed, and the capability to operate in hot and high conditions. The increase in the number of passengers is to achieve a reduction in operating costs. The resulting configuration is presented in Figure 24b, and the sized wing area is 64.5 m^2, each engine thrust is 13,085 lbf, and the MTOW has increased from 21,500 kg in the base aircraft to 26,188 kg in the developed aircraft. In this configuration, the engines have a BPR of 6, and the configuration is named "CLW-06".



(**a**) Base Aircraft (**b**) Affordable CLW with HBPR engines (CLW-06)

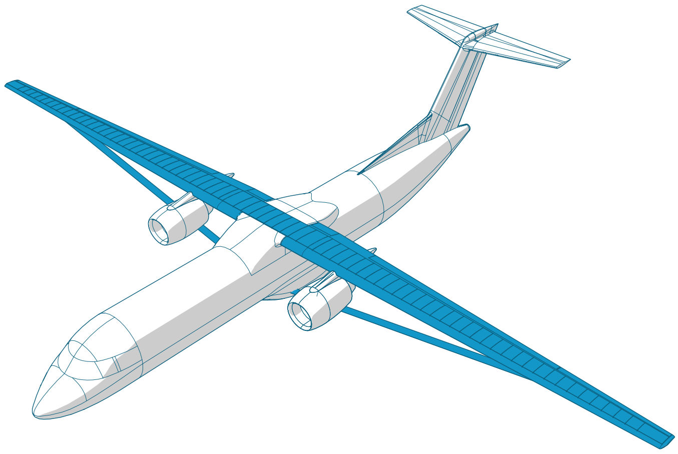
Figure 24. Application of the framework for the development of affordable conventional aircraft.

4.3. Development of an Affordable TBW Aircraft

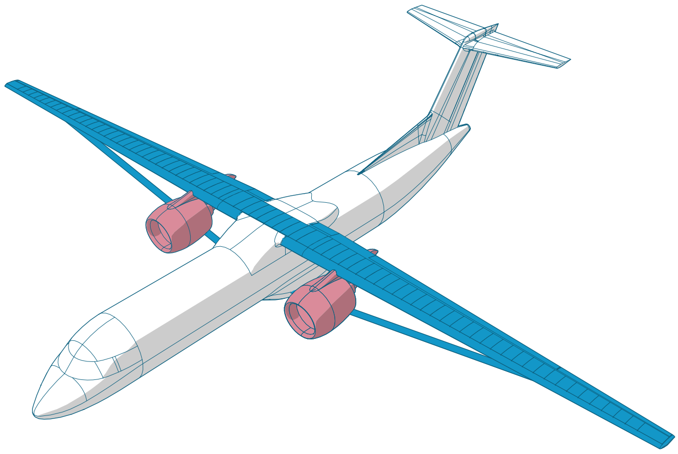
Based on the affordable CLW (Cantilever Wing) configuration (Section 4.2), an affordable TBW configuration is developed. This affordable methodology has been proposed in many aircraft-level technology development projects. Airbus, in collaboration with Siemens, has proposed to use BAe 146 as the base platform for the development of a hybrid-electric propulsion system in the project E-Fan X [78]. Similarly, Boeing has begun the modification of an MD-90 aircraft into the TBW aircraft, named X-66A, as a platform for demonstrating the TBW with major modifications to the wing and engine location, and minor modifications to the fuselage and the tail [79]. In this application of the developed framework, the outer wing is replaced with a trapezoidal high aspect ratio wing, and a truss is added. No changes in the fuselage, nacelle, pylons, landing gears, and tail installation are incorporated. Since the mission profile, including the cruise speed, is kept similar to the base aircraft, no major change in thickness ratio distribution and sweep angle is implemented. The resulting configuration, which is named "TBW-06", is presented in Figure 25a. By applying these changes, the aircraft MTOW has reduced to 24,883 kg, and the fuel weight is reduced from 5144 kg in the CLW to 4558 kg in the TBW. Additionally, the DOC has reduced from 235.4 $\$/pax/trip$ in the CLW to 224.1 $\$/pax/trip$ in the TBW.

In the next iteration of the design, the HBPR (High Bypass Ratio) engines are replaced with VHBPR (Very High Bypass Ratio) ones to achieve more reduction in fuel consumption. The resulting configuration is named "TBW-12" and is presented in Figure 25b. This change affects the geometry, aerodynamics, propulsion, and weight characteristics. Geometrically, the engine replacement will entail a growth in engine diameter, which increases the drag and decreases the lift over the pylon region. Moreover, the VHBPR engines contribute to more windmilling drag in the OEI (One Engine Inoperative) conditions and increase the required engine thrust. Furthermore, the increment in windmilling drag increases the yawing moment in this condition, and it may increase the required vertical tail or rudder area. The application of VHBPR engines also have consequences on the weight of the nacelle, pylon, and the engine itself. Since it is possible to change the default method in the framework without altering the source code, the effects of the BPR are incorporated using:

1. **Geometry**: The equation relating the nacelle diameter to engine thrust developed by Svoboda [80] is expanded to cover the effect of the bypass ratio and subsequently used in this framework.
2. **Aerodynamics**: The method presented by Roskam [81], which uses the wetted area, is used to calculate the drag.
3. **Propulsion**: The variation of SFC with respect to BPR is extracted from Torenbeek [12].
4. **Weight**: The method provided by Jenkinson [73], which considers thrust and BPR, is calibrated and used.



(**a**) Affordable TBW with HBPR engines (TBW-06)



(**b**) Affordable TBW with VHBPR engines (TBW-12)

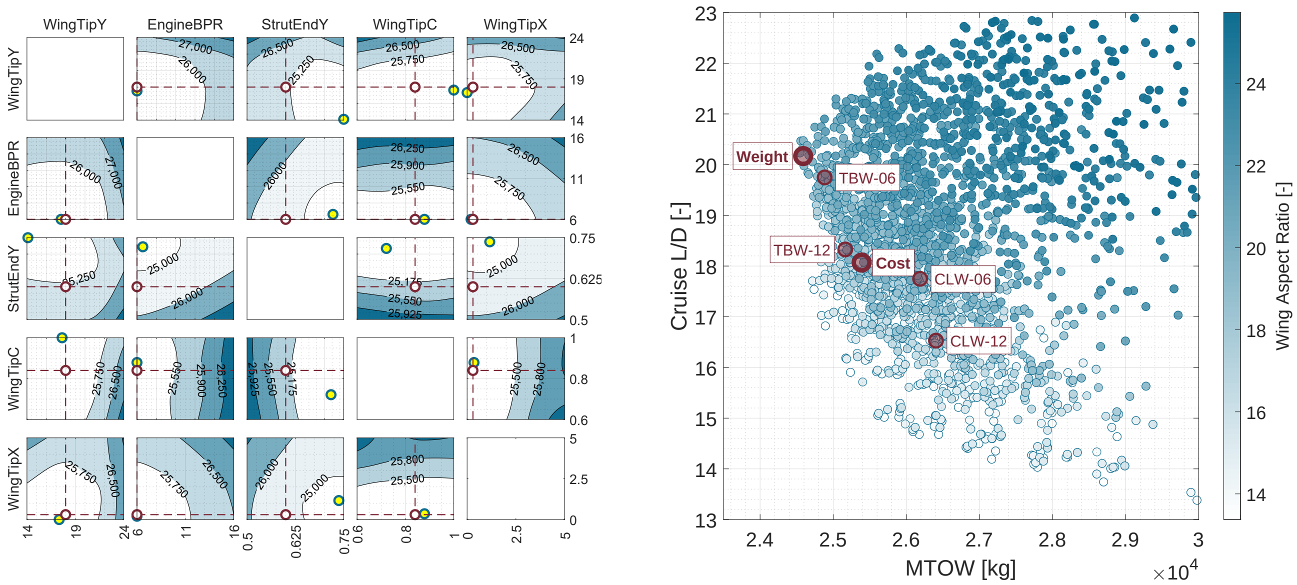
Figure 25. Application of the framework.

The fuel weight is reduced by 1296 kg (25.2%) in this aircraft relative to the CLW version, which is the result of both increased aerodynamic performance due to a higher aspect ratio and lowered fuel consumption due to a higher bypass ratio. Also, this configuration has a lower wing weight (2128 kg) compared to the CLW configurations (2436 kg) as the result of two factors, mainly due to the strut and secondly from reduced take-off weight.

For this application, the aerodynamic analysis was performed using a hybrid combination of VLM and empirical methods. The parasite drag was computed using DATCOM, and the strut drag and interference drag were added using the wetted area method. Though this method may be suitable for the conceptual design phase, high-fidelity CFD methods are required to capture the true nature of the interference effects around the wing-strut and fuselage-strut junction points. In the structure model as well, although high-fidelity methods were used for the structure sizing, the wing deflection effect on the load distribution is not considered. The consideration of the wing flexibility requires the coupling between the structural and aerodynamic model and will increase the computational time. Furthermore, the effect of the aileron reversal phenomena is not considered, which can induce weight penalties in the TBW configuration.

4.4. Optimization of Truss-Braced Wing Regional Jet

The developed framework is employed for the optimization of the regional TBW aircraft to minimize multiple objectives. In the optimization process, the independent design variables are wing span, strut-wing spanwise location, engine bypass ratio, wing sweep, and wing tip chord. The optimization objectives are cost in terms of DOC and weight in terms of MTOW. The framework is capable of performing a 2-D sensitivity analysis, in which the variation of each objective to changes in each pair of design variables is investigated, while other design variables are constant. Each subplot in Figure 26a represents the variation of the MTOW with respect to a pair of design variables, e.g., the subplot on the fourth row and third column represents the effect of wing tip chord and strut spanwise location. The plot helps to understand the sensitivity of the objective to the variables, find the points on each subplot where the objective will be minimum (yellow circles), and the closeness of the base design point (red circles) to the optimum point.



(**a**) Sensitivity Analysis (**b**) Pareto Front

Figure 26. Results of the optimization module.

For optimization purposes, an initial population of samples is created, which is used to train the surrogate models and develop the Pareto front. For example, in Figure 26b, the variation of cruise aerodynamic performance and wing weight is investigated. In this figure, the randomly generated DoE samples are in blue, the affordable design cases are in simple red (TBW-06, TBW-12, CLW-06, and CLW-12), and the weight-optimum and cost-optimum design points are in bold red. As can be seen, the take-off weight and cruise L/D ratio drive the optimum design in contradicting directions. Generally, as the wing span increases, the L/D increases, but not uniformly due to different nacelle and strut drag. As the L/D increases, the mission fuel is reduced, which reduces the MTOW. The reduction continues until a point at which the weight penalties of the high aspect ratio wing and the wing folding are introduced and will result in higher take-off weights. In the cost-optimum design, the operating cost is reduced by 7.7% with respect to the base CLW configuration. The optimum aircraft layouts are plotted in Figure 27. As can be seen, the cost objective has driven the problem toward higher aspect ratios and higher engine bypass ratios (Figure 27 left). On the other hand, the weight optimization results in a lower aspect ratio and lower bypass ratio. This clearly shows the importance of considering cost objective function in the conceptual design phase.

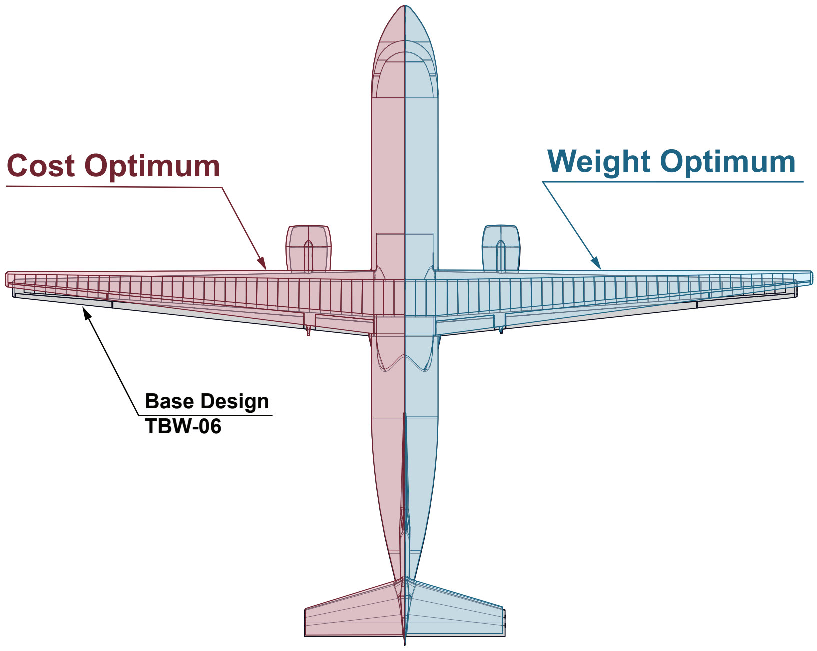


Figure 27. Aircraft optimum layout.

5. Conclusions

In this paper, a new aircraft design, analysis, and optimization framework, which are developed in AUT (Amirkabir University of Technology (Tehran Polytechnic)) for the MDO (Multidisciplinary Design Optimization) of civil aircraft, are introduced. To this aim, the architecture, functionalities, and modules are presented. This framework, which is developed using MATLAB, has an open architecture that enables the extension of the built-in analysis tools from low-fidelity to high-fidelity. Currently, the framework covers the essential modules that are needed for the optimization of a civil aircraft, and it has been successfully applied for the optimization of a TBW (Truss-Braced Wing) configuration. The developed framework is applied to different aircraft design problems, including design from scratch, design of conventional variants, and design of novel configurations.

Integration of high-fidelity external tools has enabled the framework to design and optimize a novel configuration, while no empirical data were available. Many low-fidelity codes (such as AVL and DATCOM) and high-fidelity software (such as CATIA and Nastran) were successfully integrated into the framework, and thanks to interface modules, the data were exchanged fluently. The results have encouraged the authors to further develop the framework to include novel propulsion architecture and high-fidelity aerodynamic methods into the framework.

The work is in progress to increase the computational speed, particularly in the structure analysis module, in which the third-party 3-D structure meshing tool will be replaced with a MATLAB or Python meshing code. On the aerodynamic side, the high-fidelity aerodynamic analysis using CFD (Computational Fluid Dynamics) is being implemented and integrated into the framework, which can enable analysis of the aerodynamic characteristics using high-fidelity tools, and performing aerodynamic shape optimization and wing high-fidelity aero-structure optimization. In the current implementation, the developed framework can handle only gas-powered turbofan and turboprop engines, and the work is planned for the integration of novel propulsion technologies, such as hybrid-electric, turboelectric, and hhydrogen-powered propulsions. Currently, as the tails are sized using the tail volume method, the static margin and the effectiveness of control surfaces are not enforced explicitly, which are required to ensure aircraft stability and controllability. The application of more accurate methods, such as the X-plot, is planned for the next versions of the framework.

In parallel with the development of technical modules, the work for the preparation of a detailed and comprehensive manual is planned. This documentation is an essential part of the framework release, as the framework highly relies on external tools for high-fidelity analyses. Furthermore, the input file is textual (compared to interactive or graphical inputs), which requires proper documentation of all input commands. For these reasons, though the developed framework features an open architecture, the code is not open source yet.

Author Contributions: Conceptualization, S.H. and M.A.V.-Z.; methodology, S.H., M.A.V.-Z. and H.R.O.; software, S.H.; validation, S.H.; investigation, S.H.; writing—original draft preparation, S.H.; writing—review and editing, S.H., M.A.V.-Z. and H.R.O.; visualization, S.H.; supervision, M.A.V.-Z. and H.R.O.; project administration, M.A.V.-Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data generated and developed in this study are available from the corresponding author, M.A.V.-Z., upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

AAA	Advanced Aircraft Analysis
AAM	Advanced Air Mobility
ACSYNT	Aircraft Synthesis
ADDAM	Aircraft Design DAta Model
ADEBO	Aircraft DEsign BOx
AI	Artificial Intelligence
ANN	Artificial Neural Network
AOA	Angle of Attack
ARC	Ames Research Center
ATR	Average Temperature Response
AUT	Amirkabir University of Technology (Tehran Polytechnic)
AVL	Athena Vortex Lattice
BLI	Boundary Layer Ingestion
BPR	Bypass Ratio
BWB	Blended Wing Body
CAD	Computer-Aided Design
CEASIOM	Computerized Environment for Aircraft Synthesis and Integrated Optimization Methods
CFD	Computational Fluid Dynamics
CG	Center of Gravity
CLI	Command Line Interface
CLW	Cantilever Wing
COM	Component Object Model
COTS	Commercial Off-The Shelf
CPACS	Common Parametric Aircraft Configuration Schema
DEE	Design and Engineering Engine
DELWARX	Distributed Design Optimization of Large Aspect Ratio Wing Aircraft with Rapid Transonic Flutter Analysis in Linux
DLR	German Aerospace Center
DOC	Direct Operating Cost
DoE	Design of Experiment
DOF	Degree of freedom

EU	European Commission
FEA	Finite Element Analysis
FEM	Finite Element Model
FLOPS	Flight Optimization System
GA	Genetic Algorithm
GUI	Graphical User Interface
HBPR	High Bypass Ratio
HEP	Hybrid Electric Propulsion
JPAD	Java toolchain of Programs for Aircraft Design
LAMBDA	Laboratory of Aircraft Multidisciplinary Knowledge-Based Design and Analysis
LaRC	Langley Research Center
LEAPS	Layered and Extensible Aircraft Performance System
MDA	Multidisciplinary Analysis
MDO	Multidisciplinary Design Optimization
MDOPT	Multidisciplinary Design Optimization
MIT	Massachusetts Institute of Technology
MTOW	Maximum Take-Off Weight
MZFW	Maximum Zero-Fuel Weight
NASA	National Aeronautics and Space Administration
OEI	One Engine Inoperative
OEW	Operating Empty Weight
OOP	Object-Oriented Programming
OPR	Overall Pressure Ratio
PASS	Program for Aircraft Synthesis Studies
PrADO	Preliminary Aircraft Design and Optimization Program
PreSTo	Preliminary Sizing Tool
PyPAD	Python module for Preliminary Aircraft Design
QCARD	Quick Conceptual Aircraft Research and Development
RADE	Rapid Airframe Design Environment
RAPID	Robust Aircraft Parametric Interactive Design
RDS	Raymer Design Software
SFC	Specific Fuel Consumption
SUAVE	Stanford University Aerospace Vehicle Environment
TASOPT	Transport Aircraft System Optimization
TAW	Tube-and-Wing
TBW	Truss-Braced Wing
TEP	Turboelectric Propulsion
TIT	Turbine Inlet Temperature
TUI	Textual User Interface
TUM	Technical University of Munich
VBA	Visual Basic for Applications
VHBPR	Very High Bypass Ratio
VLM	Vortex Lattice Method

Symbols

b	Wing Span
c_k	Kink Chord
c_r	Root Chord
c_t	Tip Chord

H	Altitude
P_s	Specific Excess Power
L/D	Lift to Drag Ratio
M_{cr}	Cruise Mach Number
S	Wing Area
T_{sls}	Sea/Level Static Thrust
T/W	Thrust Loading
V	Velocity
W/s	Wing Loading
y_k	Kink Lateral Position
y_t	Tip Lateral Position
η_k	Kink Span Ratio
λ	Taper Ratio
Λ_{c4}	Sweep Angle of Quarter-Chord Line
Λ_{le}	Sweep Angle of Leading Edge Line
Λ_{te}	Sweep Angle of Trailing Edge Line

Appendix A. Wing Planform Design

The design of the wing planform can be linked to the cruise Mach number. The developed method, which applies to kinked wings, is derived from the method presented in [82], which itself was based on the data provided in [12,83]. In the implemented approach, the wing area, wing span, and kink spanwise location are input values (design variables), and the chord of sections and sweep angle of segments are computed. A similar approach is presented in [84] for cases where the Airbus definition of the reference wing area is used. The wing parameters are depicted in Figure A1.

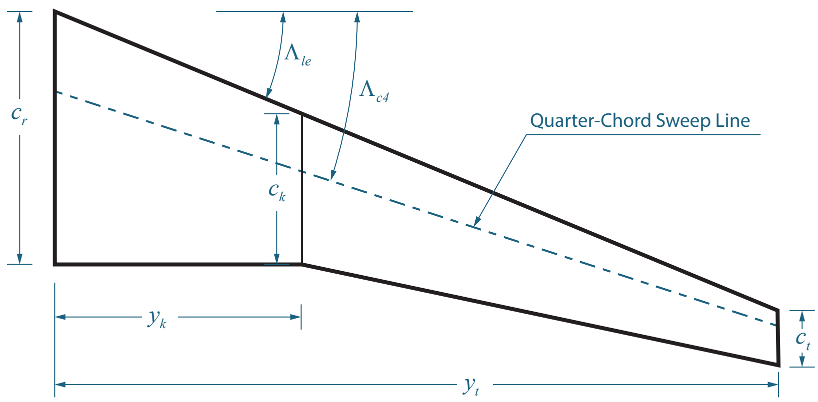


Figure A1. Kined wing parameters.

The wing quarter-chord sweep angle (Λ_{c4}) is computed from:

$$\Lambda_{c4} = \begin{cases} 0 & M_{cr} < 0.66 \\ \arccos\left(\dfrac{1.16}{M_{cr}+0.5}\right) & M_{cr} \geqslant 0.66 \end{cases} \tag{A1}$$

in which M_{cr} is the cruise Mach number. The taper ratio (λ) is computed from:

$$\lambda = -0.0083\Lambda_{c4} + 0.4597 \tag{A2}$$

where Λ_{c4} is in degrees. The kink and tip spanwise position, y_k and y_t, respectively, are computed from:

$$y_t = \frac{b}{2} \tag{A3a}$$

$$y_k = \eta_k y_t \tag{A3b}$$

The leading edge sweep angle and chords of root, kink, and tip are the unknown parameters, and four geometrical equations are established to calculate these parameters. The first equation relates the wing area to wing chords:

$$\begin{aligned} S &= 2\left[\left(\frac{c_r + c_k}{2}\right)y_k + \left(\frac{c_k + c_t}{2}\right)(y_t - y_k)\right] \\ &= y_k c_r + y_t c_k + (y_t - y_k)c_t \end{aligned} \tag{A4}$$

In the second equation, the root chord and tip chord are related to each other using the taper ratio:

$$c_t = \lambda c_r \tag{A5}$$

The third equation states the sweep angle of the trailing edge (Λ_{te}) of the inner segment of the wing is zero:

$$c_r - c_t = y_k \tan(\Lambda_{le}) \tag{A6}$$

The fourth equation relates the wing quarter-chord sweep angles to the leading edge sweep angle:

$$y_t \tan(\Lambda_{c4}) = \left(y_t \tan(\Lambda_{le}) + \frac{c_t}{4}\right) - \frac{c_r}{4} \tag{A7}$$

By combining Equations (A4)–(A7), a system of linear equations is developed:

$$\begin{bmatrix} y_k & y_t & y_t - y_k & 0 \\ \lambda & 0 & -1 & 0 \\ 1 & -1 & 0 & -y_k \\ -1/4 & 0 & 1/4 & y_t \end{bmatrix} \begin{bmatrix} c_r \\ c_k \\ c_t \\ \tan(\Lambda_{le}) \end{bmatrix} = \begin{bmatrix} S \\ 0 \\ 0 \\ y_t \tan(\Lambda_{c4}) \end{bmatrix} \tag{A8}$$

By solving Equation (A8), the leading edge sweep angle and chords of root, kink, and tip are calculated, which can be used to establish the wing planform.

References

1. Airbus. Global Market Forecast 2022–2041. Report, Airbus. 2022. Available online: https://bit.ly/3TuwJhT (accessed on 21 March 2024).
2. ACARE. Flightpath 2050, Europ's Vision for Aviation. Report, ACARE. 2011. Available online: https://bit.ly/49Znm0H (accessed on 21 March 2024).
3. Li, L.; Bai, J.; Qu, F. Multipoint Aerodynamic Shape Optimization of a Truss-Braced-Wing Aircraft. *J. Aircr.* **2022**, *59*, 1179–1194 . [CrossRef]
4. Lyu, Z.; Martins, J.R.R.A. Aerodynamic Design Optimization Studies of a Blended-Wing-Body Aircraft. *J. Aircr.* **2014**, *51*, 1604–1617. [CrossRef]
5. Uranga, A.; Drela, M.; Hall, D.K.; Greitzer, E.M. Analysis of the Aerodynamic Benefit from Boundary Layer Ingestion for Transport Aircraft. *AIAA J.* **2018**, *56*, 4271–4281. [CrossRef]
6. Zamboni, J.; Vos, R.; Emeneth, M.; Schneegans, A. A Method for the Conceptual Design of Hybrid Electric Aircraft. In Proceedings of the AIAA Scitech 2019 Forum, San Diego, CA, USA, 7–11 January 2019. [CrossRef]
7. Connolly, J.W.; Chapman, J.W.; Stalcup, E.J.; Chicatelli, A.; Hunker, K.R. Modeling and Control Design for a Turboelectric Single Aisle Aircraft Propulsion System. In Proceedings of the 2018 AIAA/IEEE Electric Aircraft Technologies Symposium, Cincinnati, OH, USA, 9–11 July 2018. [CrossRef]
8. Karpuk, S.; Ma, Y.; Elham, A. Design Investigation of Potential Long-Range Hydrogen Combustion Blended Wing Body Aircraft with Future Technologies. *Aerospace* **2023**, *10*, 566. [CrossRef]
9. Andrews, S.A.; Perez, R.E. Comparison of Box-Wing and Conventional Aircraft Mission Performance using Multidisciplinary Analysis and Optimization. *Aerosp. Sci. Technol.* **2018**, *79*, 336–351. [CrossRef]
10. Hosseini, S.; Vaziri-Zanjani, M.A.; Ovesy, H.R. Conceptual Design and Analysis of an Affordable Truss-Braced Wing Regional Jet Aircraft. *Proc. Inst. Mech. Eng. Part J. Aerosp. Eng.* **2020**. [CrossRef]
11. Roskam, J. *Airplane Design, Part I: Preliminary Sizing of Aircraft*; Roskam Aviation and Engineering Corporation: Lawrence, KS, USA, 1985; p. 202. Available online: https://bit.ly/3TpT53P (accessed on 21 March 2024).
12. Torenbeek, E. *Synthesis of Subsonic Airplane Design*; Springer: Dordrecht, The Netherlands, 1982; p. 598. [CrossRef]

13. Raymer, D. *Aircraft Design: A Conceptual Approach*, 5th ed.; American Institute of Aeronautics and Astronautics, Inc.: Washington, DC, USA, 2012. [CrossRef]
14. Vanderplaats, G.N. Automated Optimization Techniques for Aircraft Synthesis. In Proceedings of the AIAA Aircraft Systems & Technology Meeting, Dallas, TX, USA, 27–29 September 1976. [CrossRef]
15. Mccullers, L.A. Aircraft Configuration Optimization Including Optimized Flight Profiles. Report, NASA. 1984. Available online: https://ntrs.nasa.gov/citations/19870002310 (accessed on 21 March 2024).
16. Horvath, B.L.; Wells, D.P. Comparison of Aircraft Conceptual Design Weight Estimation Methods to the Flight Optimization System. In Proceedings of the 2018 AIAA Aerospace Sciences Meeting, Kissimmee, FL, USA, 8–12 January 2018. [CrossRef]
17. Kroo, I.M.; Takai, M. A Quasi-Procedural, Knowledge-Based System for Aircraft Design. In Proceedings of the AIAA/AHS/ASEE Aircraft Design, Systems and Operations Meeting, Atlanta, GA, USA, 7–9 September 1988. [CrossRef]
18. Lissys. Piano 5 for Windows, Version 5.3. 2014. Available online: https://www.lissys.uk/Piano5.html (accessed on 21 March 2024).
19. Simos, D. Computer Methods for the Preliminary Design and Operational Optimisation of Twin Engine Propeller Driven AIrcraft. Ph.D. Thesis, Loughborough University, Loughborough, UK, 1984. Available online: https://dspace.lboro.ac.uk/2134/25116 (accessed on 21 March 2024).
20. Roskam, J.; Malaek, S.M.; Anemaat, W. AAA (Advanced Aircraft Analysis): A User-Friendly Approach to Preliminary Aircraft Design. In Proceedings of the 17th International Congress of the Aeronautical Sciences, Stockholm, Sweden, 9–14 September 1990. Available online: https://bit.ly/3IKMps20 (accessed on 21 March 2024).
21. Anemaat, W.; Kaushik, B. Geometry Design Assistant for Airplane Preliminary Design. In Proceedings of the 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition. American Institute of Aeronautics and Astronautics, Orlando, FL, USA, 4–7 January 2011. [CrossRef]
22. Nah, S.H. DESAID: The Development of an Expert System for Aircraft Initial Design. Ph.D. Thesis, Cranfield University, Bedford, UK, 1991. Available online: https://dspace.lib.cranfield.ac.uk/handle/1826/3593 (accessed on 21 March 2024).
23. Bardenhagen, A.; Kossira, H.; Heinze, W. Interdisciplinary Design of Modern Hypersonic Waveriders Using the Integrated Program PrADO-HY. In Proceedings of the 19th International Congress of the Aeronautical Sciences, Anaheim, CA, USA, 18–23 September 1994; pp. 1053–1063. Available online: https://bit.ly/4akFxgW (accessed on 21 March 2024).
24. Raymer, D.P. RDS—A PC-based aircraft design, sizing, and performance system. In Proceedings of the Guidance, Navigation and Control Conference, Head Island, SC, USA, 10–12 August 1992. [CrossRef]
25. Isikveren, A.T. Quasi-Analytical Modelling and Optimization Techniques for Transport Aircraft Design. Ph.D. Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2002. Available online: https://kth.diva-portal.org/smash/record.jsf?pid=diva2:9126 (accessed on 21 March 2024).
26. LeDoux, S.; Herling, W.; Fatta, G.; Ratcliff, R. MDOPT—A Multidisciplinary Design Optimization System Using Higher Order Analysis Codes. In Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, USA, 31 August–1 September 2004. [CrossRef]
27. La Rocca, G.; Van Tooren, M.J.L. Knowledge-Based Engineering Approach to Support Aircraft Multidisciplinary Design and Optimization. *J. Aircr.* **2009**, *46*, 1875–1885. [CrossRef]
28. Kaenel, R.V.; Rizzi, A.; Oppelstrup, J.; Ghoreyshi, M. CEASIOM: Simulating Stability and Control with CFD/CSM in Aircraft Conceptual Design. In Proceedings of the 26th International Congress of the Aeronautical Sciences, Anchorage, AK, USA, 14–19 September 2008; pp. 1–14. Available online: https://bit.ly/499MekS (accessed on 21 March 2024).
29. Seeckt, K.; Scholz, D. Application of the Aircraft Preliminary Sizing Tool PreSTo to Kerosene and Liquid Hydrogen Fueled Regional Freighter Aircraft. In Proceedings of the DGLR: Deutscher Luft- und Raumfahrtkongress 2010: Tagungsband—Ausgewählte Manuskripte, Hamburg, Germany, 31 August–2 September 2010. Available online: https://perma.cc/K94E-BW3Z (accessed on 21 March 2024).
30. Greitzer, E.M.; Bonnefoy, P.A.; DelaRosaBlanco, E.; Dorbian, C.S.; Drela, M.; Hall, D.K.; Hansman, R.J.; Hileman, J.I.; Liebeck, R.H.; Lovegren, J. N+3 Aircraft Concept Designs and Trade Studies. Volume 2. Appendices-Design Methodologies for Aerodynamics, Structures, Weight, and Thermodynamic Cycles. Report NASA/CR-2010-216794/VOL2, NASA. 2010. Available online: https://ntrs.nasa.gov/citations/20100042398 (accessed on 21 March 2024).
31. Böhnke, D.; Nagel, B.; Gollnick, V. An approach to multi-fidelity in conceptual aircraft design in distributed design environments. In Proceedings of the 2011 Aerospace Conference, Big Sky, MT, USA, 5–12 March 2011; IEEE: New York, NY, USA, 2011; pp. 1–10. [CrossRef]
32. Nagel, B.; Böhnke, D.; Gollnick, V.; Schmollgruber, P.; Rizzi, A.; La Rocca, G.; Alonso, J.J. Communication in Aircraft Design: Can We Establish a Common Language? In Proceedings of the ICAS, Brisbane, Australia, 23–28 September 2012. Available online: https://www.icas.org/ICAS_ARCHIVE/ICAS2012/ABSTRACTS/201 (accessed on 21 March 2024).
33. Munjulury, R.C.; Berry, P.; Krus, P. RAPID—Robust Aircraft Parametric Interactive Design: A Knowledge Based Aircraft Conceptual Design Tool. In Proceedings of the 4th CEAS Conference in Linköping, Linköping, Sweden, 16–19 September 2013; pp. 255–262. Available online: http://liu.diva-portal.org/smash/record.jsf?pid=diva2:687478 (accessed on 21 March 2024).
34. Munjulury, R.C.; Staack, I.; Berry, P.; Krus, P. A Knowledge-Based Integrated Aircraft Conceptual Design Framework. *Ceas Aeronaut. J.* **2016**, *7*, 95–105. [CrossRef]

35. Schumann, H.; Berres, A.; Escher, S.; Stehr, T.; Fricke, S. ParADISE for Pre-Designing Aircraft and Systems. In Proceedings of the Deutscher Luft- und Raumfahrtkongress 2013, Stuttgart, Germany, 10–12 September 2013. Available online: https://publikationen.dglr.de/?tx_dglrpublications_pi1[document_id]=301340 (accessed on 21 March 2024).
36. Herbst, S.; Hornung, M. ADDAM: An Object Oriented Data Model for an Aircraft Design Environment in MATLAB. In Proceedings of the AIAA Modeling and Simulation Technologies Conference, Kissimmee, FL, USA, 5–9 January 2015. [CrossRef]
37. Travaglini, L.; Ricci, S.; Bindolino, G.; Isikveren, A.T.; Stumpf, E. PyPAD: A multidisciplinary framework for preliminary airframe design. *Aircr. Eng. Aerosp. Technol.* **2016**, *88*. [CrossRef]
38. Lukaczyk, T.; Wendorff, A.D.; Botero, E.; MacDonald, T.; Momose, T.; Variyar, A.; Vegh, J.M.; Colonno, M.; Economon, T.D.; Alonso, J.J. SUAVE: An Open-Source Environment for Multi-Fidelity Conceptual Vehicle Design. In Proceedings of the 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Dallas, TX, USA, 22–26 June 2015. [CrossRef]
39. Nicolosi, F.; Marco, A.D.; Attanasio, L.; Vecchia, P.D. Development of a Java-Based Framework for Aircraft Preliminary Design and Optimization. *J. Aerosp. Inf. Syst.* **2016**, *13*, 234–242. [CrossRef]
40. Corman, J.A.; Weston, N.; Friedland, C.; Mavris, D.N.; Laughlin, T.W. Rapid Airframe Design Environment (RADE): A Parametric, Modular, and Multidisciplinary Framework for Conceptual Phase Airframe Design. In Proceedings of the 2018 AIAA Modeling and Simulation Technologies Conference, Reston, VA, USA, 8–12 January 2018. [CrossRef]
41. Smith, H.; Sziroczák, D.; Abbe, G.; Okonkwo, P. The GENUS aircraft conceptual design environment. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2019**, 233. [CrossRef]
42. Welstead, J.R.; Caldwell, D.; Condotta, R.; Monroe, N. An Overview of the Layered and Extensible Aircraft Performance System (LEAPS) Development. In Proceedings of the 2018 AIAA Aerospace Sciences Meeting, Kissimmee, FL, USA, 8–12 January 2018. [CrossRef]
43. Khan, K.H.; Mallik, W.; Kapania, R.K.; Schetz, J.A. Distributed Design Optimization of Large Aspect Ratio Wing Aircraft with Rapid Transonic Flutter Analysis in Linux. In Proceedings of the AIAA Scitech 2021 Forum, American Institute of Aeronautics and Astronautics, Virtual Event, USA, 11–21 January 2021. [CrossRef]
44. Liersch, C.M.; Hepperle, M. A Distributed Toolbox for Multidisciplinary Preliminary Aircraft Design. *Ceas Aeronaut. J.* **2011**, *2*, 57–68. [CrossRef]
45. Moerland, E.; Becker, R.G.; Nagel, B. Collaborative understanding of disciplinary correlations using a low-fidelity physics-based aerospace toolkit. *Ceas Aeronaut. J.* **2015**, *6*, 441–454. [CrossRef]
46. Moerland, E.; Pfeiffer, T.; Böhnke, D.; Jepsen, J.; Freund, S.; Liersch, C.M.; Pinho Chiozzotto, G.; Klein, C.; Scherer, J.; Hasan, Y.J.; et al. On the Design of a Strut-Braced Wing Configuration in a Collaborative Design Environment. In Proceedings of the 17th AIAA Aviation Technology, Integration, and Operations Conference, Denver, CO, USA, 5–9 June 2017. [CrossRef]
47. Hartmann, J.; Pfeiffer, T.; Breymann, B.; Silberhorn, D.; Moerland, E.; Weiss, M.; Nagel, B. Collaborative Conceptual Design of a Mid-Range Aircraft under Consideration of Advanced Methods for Technology Assessment. In Proceedings of the 31st International Congress of the Aeronautical Sciences, Belo Horizonte, Brazil, 9–14 September 2018. Available online: https://www.icas.org/ICAS_ARCHIVE/ICAS2018/data/preview/ICAS2018_0167.htm (accessed on 21 March 2024).
48. Woehler, S.; Atanasov, G.; Silberhorn, D.; Fröhler, B.; Zill, T. Preliminary Aircraft Design within a Multidisciplinary and Multifidelity Design Environment. In Proceedings of the Aerospace Europe Conference, Bordeaux, France, 25–28 February 2020. Available online: https://elib.dlr.de/185515 (accessed on 21 March 2024).
49. Fioriti, M.; Boggero, L.; Prakasha, P.S.; Mirzoyan, A.; Aigner, B.; Anisimov, K. Multidisciplinary Aircraft Integration within a Collaborative and Distributed Design Framework using the AGILE Paradigm. *Prog. Aerosp. Sci.* **2020**, *119*, 100648. [CrossRef]
50. The MathWorks Inc. MATLAB 9.12.0 (R2022a). 2022. Available online: https://bit.ly/497xLWO (accessed on 21 March 2024).
51. Roskam, J. *Airplane Design, Part II: Preliminary Configuration Design and Integration of the Propulsion System*; Roskam Aviation and Engineering Corporation: Lawrence, KS, USA, 1985. Available online: https://bit.ly/3IONFdM (accessed on 21 March 2024).
52. Guha, A.; Boylan, D.; Gallagher, P. Determination of optimum specific thrust for civil aero gas turbine engines: A multidisciplinary design synthesis and optimisation. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2013**, *227*, 502–527. [CrossRef]
53. Mattingly, J.D. *Aircraft Engine Design*; AIAA: Washington, DC, USA, 2002. [CrossRef]
54. Mattingly, J.D.; Boyer, K.M.; von Ohain, H. *Elements of Propulsion: Gas Turbines and Rockets*; AIAA: Washington, DC, USA, 2006. [CrossRef]
55. Hosseini, S.; Vaziry-Zanjany, M.A.; Ovesy, H.R. Multi-Objective Multidisciplinary Design Optimization of Regional Truss-Braced Wing Jet Aircraft. In Proceedings of the Aerospace Europe Conference 2023, Lausanne, Switzerland, 9–13 July 2023. [CrossRef]
56. Fink, R.D. USAF Stability and Control DATCOM. Report AFWAL-TR-83-3048, Air Force Wright Aeronautical Laboratories, USAF. 1978. Available online: https://apps.dtic.mil/sti/citations/ADB072483 (accessed on 21 March 2024).
57. Gudmundsson, S. *General Aviation Aircraft Design: Applied Methods and Procedures*; Butterworth-Heinemann: Oxford, UK, 2014. [CrossRef]
58. Drela, M. AVL (Athena Vortex Lattice). 2004. Available online: https://bit.ly/4afMbVW (accessed on 21 March 2024).
59. Dassault Systèmes. CATIA V5. 2010. Available online: https://bit.ly/4aGUkmv (accessed on 21 March 2024).
60. Hosseini, S.; Vaziri-Zanjani, M.A.; Ovesy, H.R. Aircraft Automated Layout and Architecture Conceptual Design Using Knowledge-Based Engineering. In Proceedings of the 19th International Conference of Iranian Aerospace Society, Tehran, Iran, 18–21 May 2021. Available online: https://bit.ly/3TvUCpd (accessed on 21 March 2024).
61. MSC Software. Nastran. 2020. Available online: https://bit.ly/3TW8qLf (accessed on 21 March 2024).

62. Hosseini, S.; Ovesy, H.R.; Vaziri-Zanjani, M.A. Wing Loading Process in Knowledge-Based Aircraft Design and Optimization Framework. In Proceedings of the ICAS 2022, 33th Congress of the International Council of the Aeronautical Sciences, Stockholm, Sweden, 4–9 September 2022. Available online: https://www.icas.org/ICAS_ARCHIVE/ICAS2022/data/preview/ICAS2022_0519.htm (accessed on 21 March 2024).
63. Walch, D.; Tetreault, S.; Dervault, F. Smeared Stiffeners in Panel for Mesh Simplification at Conceptual Design Phase. In Proceedings of the 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Boston, MA, USA, 8–11 April 2013. [CrossRef]
64. LMS Virtual.Lab Team. LMS Virtual.Lab 11-SL2. 2011. Available online: https://bit.ly/3vIxjjU (accessed on 21 March 2024).
65. Dababneh, O.; Kipouros, T.; Whidborne, J.F. Application of an Efficient Gradient-Based Optimization Strategy for Aircraft Wing Structures. *Aerospace* **2018**, *5*, 3. [CrossRef]
66. Vanderplaats, G.N. ADS: A FORTRAN Program for Automated Design Synthesis: Version 1.10. Report NASA-CR-177985, NASA. 1985. Available online: https://ntrs.nasa.gov/citations/19860002427 (accessed on 21 March 2024).
67. Sohst, M.; do Vale, J.L.; Afonso, F.; Suleman, A. Optimization and comparison of strut-braced and high aspect ratio wing aircraft configurations including flutter analysis with geometric non-linearities. *Aerosp. Sci. Technol.* **2022**, *124*. [CrossRef]
68. Ouyang, X.; Yu, X.; Wang, Y. Flutter Analysis for Wing Structure Using Finite Element Modeling with Equivalent Stiffness. *J. Vibroeng.* **2014**, *16*, 1483–1493. https://www.extrica.com/article/14973 (accessed on 21 March 2024).
69. Calderon, D.E.; Cooper, J.E.; Lowenberg, M.; Neild, S.A.; Coetzee, E.B. Sizing High-Aspect-Ratio Wings with a Geometrically Nonlinear Beam Model. *J. Aircr.* **2019**, *56*, 1455–1470. [CrossRef]
70. FAA. Advisory Circular: Aeroelastic Stability Substantiation of Transport Category Airplanes, 2014. Available online: https://bit.ly/3VrjjWg (accessed on 21 March 2024).
71. Megson, T.H.G. *Aircraft Structures for Engineering Students*, 7th ed.; Elsevier: Amsterdam, The Netherlands, 2021. [CrossRef]
72. Roskam, J. *Airplane Design, Part V: Component Weight Estimation*; Roskam Aviation and Engineering Corporation: Lawrence, KS, USA, 1985. Available online: https://bit.ly/4a2NqYK (accessed on 21 March 2024).
73. Jenkinson, L.R.; Simpkin, P.; Rhodes, D. *Civil Jet Aircraft Design*; AIAA: Washington, DC, USA, 1999. [CrossRef]
74. Kandlikar, M. Indices for comparing greenhouse gas emissions: integrating science and economics. *Energy Econ.* **1996**, *18*, 265–281. [CrossRef]
75. Dallara, E.S.; Kroo, I.M.; Waitz, I.A. Metric for Comparing Lifetime average Climate Impact of Aircraft. *AIAA J.* **2011**, *49*, 1600–1613. [CrossRef]
76. Roskam, J. *Airplane Design, Part VIII: Airplane Cost Estimation*; Roskam Aviation and Engineering Corporation: Lawrence, KS, USA, 1990. Available online: https://bit.ly/3TLpuTL (accessed on 21 March 2024).
77. Soares do Amaral, J.V.; Montevechi, J.A.B.; Miranda, R.d.C.; Junior, W.T.d.S. Metamodel-based Optimization: A Systematic Literature Review. *Simul. Model. Pract. Theory* **2022**, *114*. [CrossRef]
78. Benegas Jayme, D. Evaluation of the Hybrid-Electric Aircraft Project Airbus E-Fan X. Master's Thesis, Faculty of Engineering and Computer Science, Hamburg University of Applied Science, Hamburg, Germany, 2019. [CrossRef]
79. Boeing Media Relations. Airplane Arrives at Boeing Site for X-66A Modification. 2023. Available online: https://bit.ly/4afRepm (accessed on 21 March 2024).
80. Svoboda, C. Turbofan engine database as a preliminary design tool. *Aircr. Des.* **2000**, *3*, 17–31. [CrossRef]
81. Roskam, J. *Airplane Design, Part VI: Preliminary Calculation of Aerodynamic, Thrust and Power Characteristics*; Roskam Aviation and Engineering Corporation: Lawrence, KS, USA, 1987; p. 550. Available online: https://bit.ly/3vlYm4s (accessed on 21 March 2024).
82. Proesmans, P.J.; Vos, R. Airplane Design Optimization for Minimal Global Warming Impact. *J. Aircr.* **2022**, *59*, 1363–1381. [CrossRef]
83. Raymer, D.P. *Aircraft Design: A Conceptual Approach*, 6th ed.; AIAA: Washington, DC, USA, 2018. [CrossRef]
84. Ramachandran, K.; Scholz, D. PreSTo Wing Module Optimization for the Double Trapezoidal Wing. In Proceedings of the Innovative Design and Development Practices in Aerospace and Automotive Engineering: I-DAD, Avadi, Chennai, India, 22–24 February 2016; Bajpai, P.R., Chandrasekhar, U., Eds.; Springer: Singapore, 2016; pp. 337–346. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.