Preprints.org

Article

Not peer-reviewed version

A SysML Modeling Strategy to Digitalize FAR 23 Substantiation Methods

Claudio Mirabella *, Elena Garcia, Adam T. Baker, Pierluigi Della Vecchia

Posted Date: 17 November 2026

Keywords: MBSE; certification; SysML; substantiation; Python; airworthiness

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC, OpenAlex.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

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.

Article

A SysML Modeling Strategy to Digitalize FAR 23 Substantiation Methods

Claudio Mirabella [1,*], Elena Garcia [2], Adam T. Baker [3] and Pierluigi Della Vecchia [1]

[1] University of Naples Federico II, Dept. of Industrial Engineering, Via Claudio 21, Palazzina 11, I piano, Naples, Italy
[2] Georgia Institute of Technology, Aerospace Systems Design Laboratory, Chief Advanced Methods Division, 275 Ferst Drive NW, Paul Weber Space Science & Technology Building II / III (SST3), Atlanta, Georgia, United States of America
[3] SpaceWorks Enterprises, Inc., Systems Analysis Division, 2960 Brandywine Rd, Suite 200, Atlanta, GA 30341
* Correspondence: claudio.mirabella@unina.it; Tel.: +39-338-471-3099

Abstract

This study proposes a methodology for digitalizing aircraft certification processes under FAR 23, Subpart C, of the Federal Aviation Administration (FAA), through a model-based systems engineering (MBSE) approach utilizing the Systems Modeling Language (SysML). The proposed framework encodes regulatory requirements and substantiation methods. Consensus standards documents, such as the ones published and maintained by ASTM International, provide the main source for these methods. The encoded requirements and methods are exported and parsed into structured data formats and used to inform a Python code suite that performs the substantiation calculations of the modeled paragraphs. The methodology is demonstrated across three representative aircraft architectures. Key outputs include automated generation of flight envelopes, balancing loads, and powerplant load calculations. The framework is evaluated against four research questions: encoding, correctness, consistency with independent data, and behavior with faulty input. Verification against the definitions of the standard and the equilibrium equations revealed and corrected three defects in the balancing-load calculation; the engine torque, side load, and normal load agree within 1% with non-public manufacturer data, and the larger differences are traced to differing assumptions. The study is a proof of concept whose practical benefit remains to be evaluated by certification practitioners.

Keywords: MBSE; certification; SysML; substantiation; Python; airworthiness

1. Introduction

Type certification is the process through which an authority, such as the Federal Aviation Administration (FAA) in the United States and the European Union Aviation Safety Agency (EASA) in the European Union, issues a type certificate or amends an existing one [1,2]. The applicant must undertake a series of intricate, highly codified operations that require complex interactions among the design and manufacturing organizations involved in the project, as well as with the regulators. The process requires the applicant to provide evidence of compliance with all high-level requirements established by the regulatory authorities [3,4].

Although aircraft certification is traditionally viewed through a technical lens, it is increasingly recognized as a domain where engineering, regulatory policy, and organizational dynamics converge. The complexity of substantiation workflows, the multiplicity of stakeholders, and the evolving landscape of international standards call for interdisciplinary approaches [5]. Some of these challenges can be traced back to the institutional arrangements and their embedded characteristics, as discussed by Downer [6] and Correia [5]. EASA rulemaking is itself subject to the influence of interest groups and national agencies [7]. These factors motivate work on digital certification, which remains at an early stage of implementation [8].

Typically, the applicant demonstrates compliance with these high-level requirements by producing documents containing certification-related information [9]. These substantiation requirements often

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

consist of low-level technical criteria not explicitly included in the materials published by the FAA or EASA. To assist the applicant in determining which additional documents should be considered and adopted to demonstrate compliance, regulators frequently issue Advisory Circulars (ACs) and other complementary guidance materials [10].

This work develops a framework for substantiating high-level requirements in the context of general aviation certification, according to the airworthiness standards of Federal Aviation Regulations (FAR) Part 23 [11], with the scope limited to Subpart C. Subpart C has been selected because many of its requirements involve mathematical formulas, so a large part of compliance can be shown by calculation and documented in reports [12,13]. This makes the subpart well suited to illustrating a digital process.

The methods presented in this article extract data from a digital Systems Modeling Language (SysML) model of the airworthiness rules, including the selected substantiation documents. These data are subsequently used to inform the Python code that performs the Subpart C calculations of the modeled paragraphs in sequence. A mapping study of SysML research found that the language has to be customized for domain-specific aspects [14], which motivates the stereotype introduced in Section 2.1. Reported applications of model-based systems engineering (MBSE) include human space exploration at NASA [15], spacecraft functional avionics at Airbus, where interviews with 25 engineers identified application areas and adoption hurdles [16], the U.S. nuclear industry [17], and U.S. defense, where the effort of moving an existing system to MBSE has been measured [18]. Madni and Purohit [19] relate the case for MBSE investment to system complexity, regulatory constraints, and system lifespan, all of which are characteristic of aircraft certification.

Bleu-Laine et al. [20] collected FAR 23 regulations and their means of compliance in an integrated SysML model, with mappings that generate compliance checklists. Bendarkar et al. [21] extended this work with custom stereotypes and a capability that extracts the certification basis and means of compliance for a given aircraft. Fazal et al. [22] adapted the framework to Part 25 and reported the judgment needed to separate requirements from context in the regulatory text, and Glinski et al. [23] used it to identify regulatory gaps for electrified aircraft. Cartile et al. [8] derived requirements for modeling regulatory documentation and compared process mapping, UML, and ontologies against them. These studies concern the representation and management of regulatory content. Within the sources examined, the calculations that substantiate compliance remain outside the published scope of these FAR 23 models.

Earlier work published by the authors executed such calculations for EASA Certification Specification (CS-23) [24] from models built for that purpose, using MATLAB [25,26] and comparing a Papyrus/MATLAB toolchain with an MSoSA/C++ toolchain [27]. The present study differs in three ways: it adopts FAR 23 and the ASTM International consensus standards as the regulatory basis; it extends an existing FAR 23 model developed independently of the calculations, rather than one built for them; it extends the calculations to flight and powerplant loads.

This study therefore asks whether the calculation content of the standard can be added to that model, so that the substantiation calculations take their formulas and coefficients from the model rather than from the analysis code. The methodology may be extended to other subparts, particularly to requirements that call for substantiation through a calculation, a verification and validation test, or another procedure that can be encoded in a computer program; this extension has not been attempted here.

Finally, this work does not present an industry-grade framework or advocate the adoption of specific methods, techniques, or tools for the digital management of certification processes. Rather, it proposes research-based solutions that require evaluation by practitioners in representative industrial settings. Accordingly, this study should be regarded as a proof of concept for the digital generation and management of certification data for complex products, including aeronautical systems. Although the presented framework does not address the full set of certification requirements, it provides a

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

foundation for further assessment and development by domain experts, certification engineers, and regulatory authorities.

The structure of this article is as follows.

- Section 1 (Introduction) sets the scene for the complex certification landscape applicable to a general aviation aircraft. A limited set of substantiation documents is considered, issued by ASTM International, formerly the American Society for Testing and Materials (ASTM). The FAA accepts these standards as means of compliance with FAR 23, and EASA also refers to them for CS-23, Amendment 6 [24]; in this study, CS-23 is mentioned only for comparison.
- Section 2 (Tools, rationale, and methodology) explains the rationale behind the modeling strategy and the introduction of user-defined stereotypes capable of capturing the complex information contained in paragraphs and subparagraphs of the substantiation documents. The modeling environment selected for this work is Dassault Systèmes CATIA Magic Systems of Systems Architect (MSoSA). To extract information from the model and enable the compliance demonstration calculations, external Python code has been developed. This section also explains why the authors chose to develop the code in Python 3.9.23.
- Section 3 (Results) presents a selection of the results obtained using the digital certification pipeline, together with the verification of the framework. The framework is applied to three aircraft certifiable under FAR 23, which differ in size, certification level, and number of engines. The selected aircraft include a two-seat single-engine aircraft (inspired by the Tecnam P92), a four-seat twin-engine light aircraft (inspired by the Tecnam P2006), and an aircraft representative of a small commuter type capable of transporting eleven occupants (inspired by the Tecnam P2012). The purpose of this section is to provide a broad set of applications of the digital certification pipeline.
- Section 4 (Discussion) discusses the results and compares the robustness of the framework with that of related approaches.
- Section 5 (Conclusions) summarizes the present implementation and its growth potential. In particular, it provides the rationale for expanding the Python enabling code to include other substantiation methods.
- Appendix A includes Python code excerpts that illustrate the general approach adopted by the authors to digitalize the substantiation methods directly from the SysML model.
- Appendix B gives the numerical results in tabular form: geometry and mass data, flight envelope points, balancing loads, and powerplant loads.

2. Tools, Rationale, and Methodology

This section describes the modeling strategy. The modeling environment used is CATIA MSoSA, a software platform developed by Dassault Systèmes. The SysML model is maintained and expanded within this environment.

Subsection 2.1 discusses the general characteristics of the FAR 23 SysML model and the user-defined stereotypes applied to digitalize the ASTM International standards. Subsection 2.2 summarizes the Python code implemented to perform the substantiation calculations, with a focus on the data extraction phase and the techniques used to directly inform the functions with the extracted substantiation data, formulas, and general information. This approach is intended to let the code follow changes made to the substantiation content in the model without changes to the calculation code; this holds only for changes that respect the encoding conventions described below.

2.1. Substantiation Documents Modeling

The FAA suggests applicable substantiation documents to applicants through AC 23.2010-1 [10], FAA Accepted Means of Compliance Process for 14 CFR Part 23. In this article, Means of Compliance (MoC) denotes the methods accepted by an authority for showing compliance with a requirement.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

This AC lists a series of consensus standard documents, that is, standards developed and recognized by aviation industry stakeholders, which the FAA accepts as means of compliance [12]. The use of consensus standards as means of compliance has also been criticized on legal grounds [28].

Applicants may choose to adopt one or more of these standards. In the context of this work, the authors have selected the ASTM International standards as the reference substantiation documents.

These documents are referenced in FAA and EASA guidance [10,24], although the two regulatory frameworks retain differences in approach [29,30].

Figure 1 shows a SysML Block Definition Diagram (BDD) concerning requirement FAR 23.2210, *Structural Design Loads*. This requirement is an example of the modeling logic adopted in this work. AC 23.2010-1 connects the main paragraph to the substantiation documents selected by the authors. The BDD also includes the blocks corresponding to the paragraphs of the substantiation standards that explicitly refer to the limit and ultimate load factor requirements and definitions. To avoid losing generality in the applicability of the substantiation methods, one document has been selected: ASTM International F3116/F3116M-24 [31], *Standard Specification for Design Loads and Conditions*, which covers the design loads and conditions of a broad range of aircraft certifiable under FAR 23.

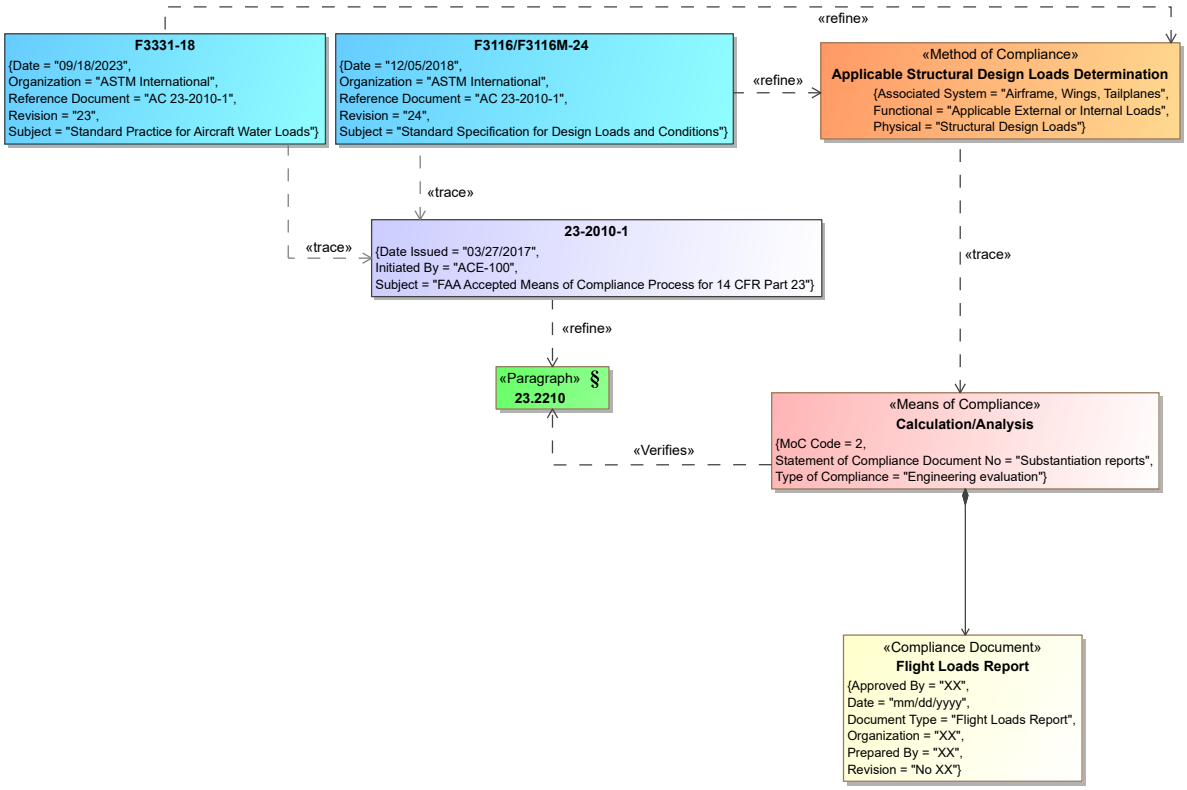


Figure 1. FAR 23.2210, modeled in CATIA MSoSA. The paragraph and its subparagraphs are highlighted in green, while the applicable substantiation documents are linked through AC 23.2010-1, which identifies the accepted means of compliance.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

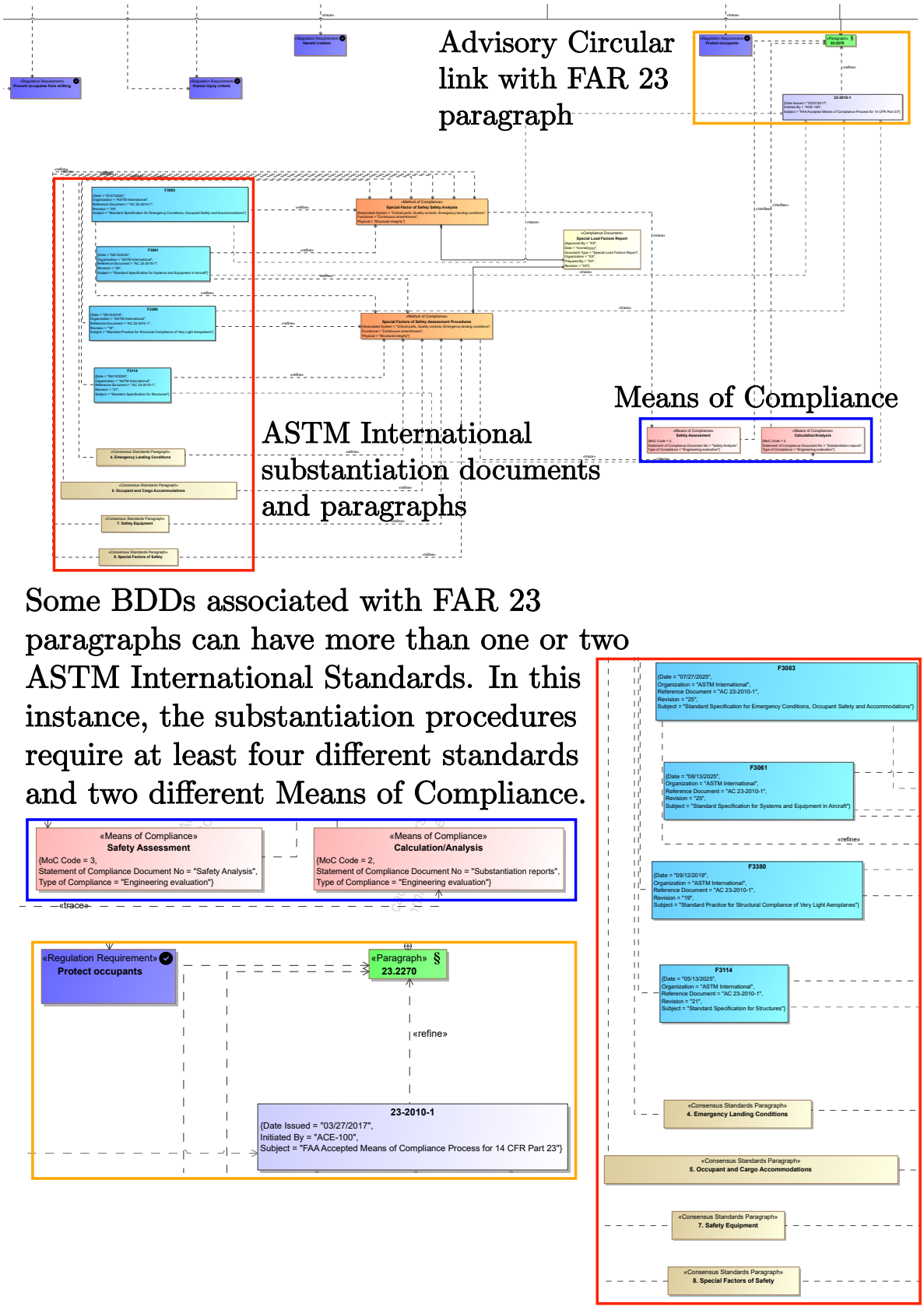


Figure 2. Partial view of the Block Definition Diagram (BDD) for FAR 23.2270, *Emergency Conditions*, showing the links between the requirement and the four ASTM standards needed to demonstrate compliance.

Figure 3 presents a graphical representation of the substantiation performed by the applicants, who search for applicable MoC in the regulatory framework, shown at the top of the figure. Different design architectures lead to the Means of Compliance codes table; the selected code is indicated by a

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

different color. Finally, at the bottom right, the three colored blocks describe the application presented in this work. The framework addresses only code MC2, calculation and analysis, and document type (e), substantiation reports; the other codes, including tests, inspections, and simulations, are outside the scope of this work.

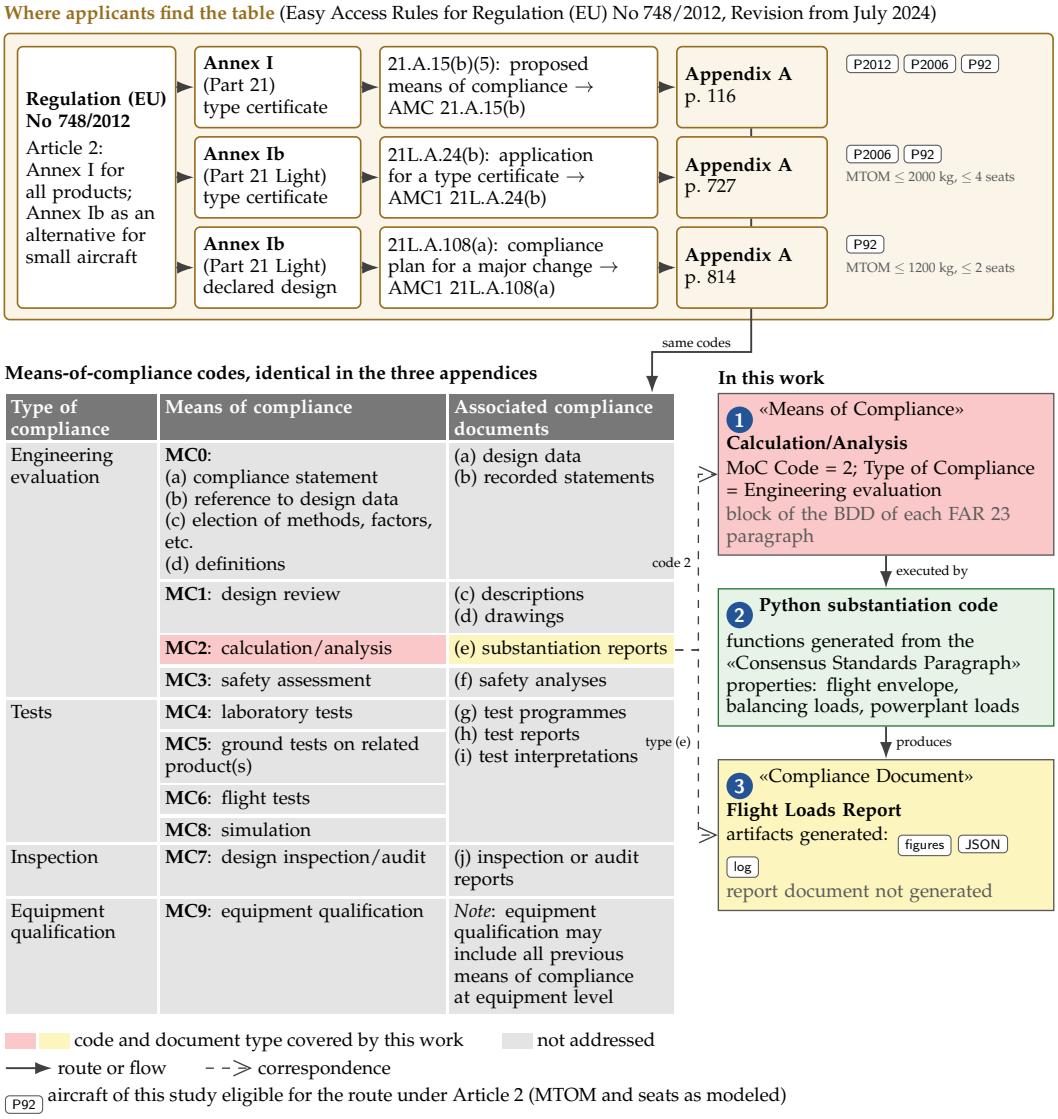


Figure 3. Graphical representation of how applicants reach the MoC table. The BDDs of the SysML create a direct link between airworthiness requirements and accepted MoC.

The ASTM International compendium includes additional documents applicable to the determination of limit and ultimate loads, for example, ASTM International F3396/F3396M [32], *Standard Practice for Aircraft Simplified Loads Criteria*. However, the scope and applicability of this document are limited, as it is only valid for Level 1 and Level 2 aircraft, which typically include low-performance, technologically less complex architectures. ASTM International F3331-18 [33], *Standard Practice for Aircraft Water Loads*, includes substantiation methods for determining limit and ultimate loads applicable to seaplane and amphibious aircraft, which are also outside the scope of the present work.

Other applicable ASTM documents are not included in this work, such as the previously mentioned F3396/F3396M, the F3331-18, and the F2245-23 [34]; these standards contain a less general set of methods. It could be of interest to compare certification outcomes using a different set of substantiation methods, extracted from alternative substantiation documents, particularly from a design and manufacturing perspective. However, such a comparison lies outside the scope of this study and may be addressed in future work.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

Figure 4 presents the logical steps covered by the Block Definition Diagrams reported in Figures 1 and 2; Figure 3 shows where the last steps of this chain sit within the EASA means of compliance codes, which are used here for illustration. Each FAR 23 paragraph contains requirements that need to be substantiated; a *Refine* relationship allows linking the paragraph to the guidance material, in this case the AC 23.2010-1 Advisory Circular, where applicants may find the most appropriate set of consensus standards documents and material. Given the certification tasks associated with Subpart C requirements, the reference standards selected are the ones issued by ASTM International. A *Trace* relationship is used to connect the substantiation to the correct standard. The substantiation is refined by a block that precisely states what must be provided to regulators to show compliance, tracing the execution of the tasks to the corresponding means of compliance code [35,36], issued by EASA under Part 21 [37–39].

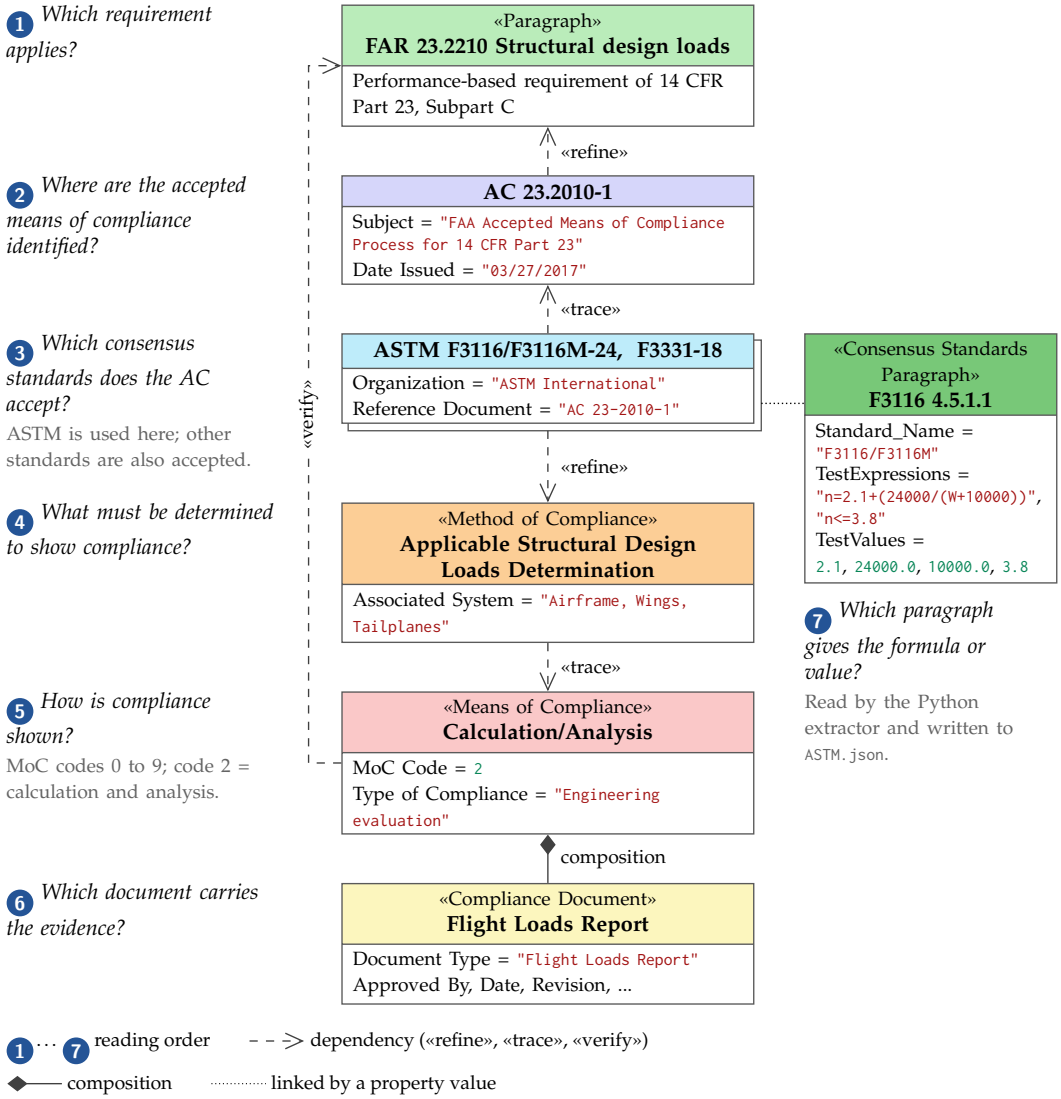


Figure 4. The general logical flow applied in drafting the BDDs presented in Figures 1 and 2.

In the FAA regulatory landscape, several documents offer procedures and accepted means for showing compliance and presenting the substantiation documentation correctly [40,41], without exactly corresponding to the accepted MoC table issued by EASA. Finally, the compliance must be presented in a certification item, typically a document; in this specific case, the *Flight Loads* compliance report is the desired outcome.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

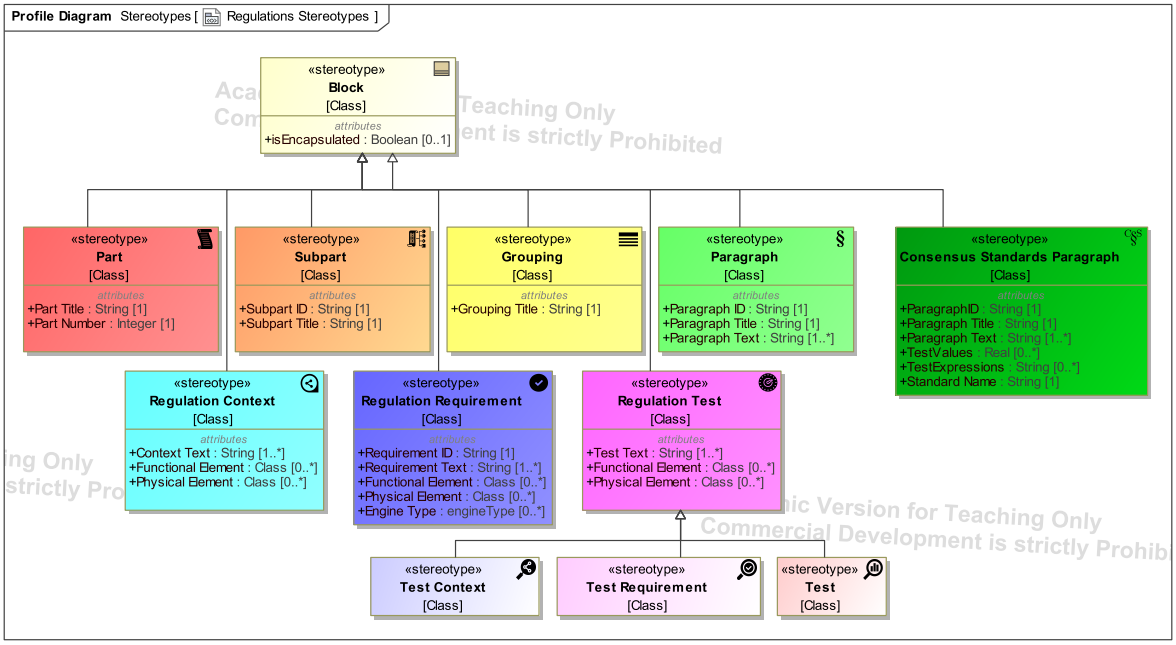


Figure 5. The Regulations Profile Diagram (PD). The stereotypes used in this work are an elaboration of the ones included in [23]. This diagram has been enriched with an additional user-defined stereotype, named Consensus Standards Paragraph, modeled upon the Paragraph stereotype, with two additional properties: *TestExpressions* and *TestValues*.

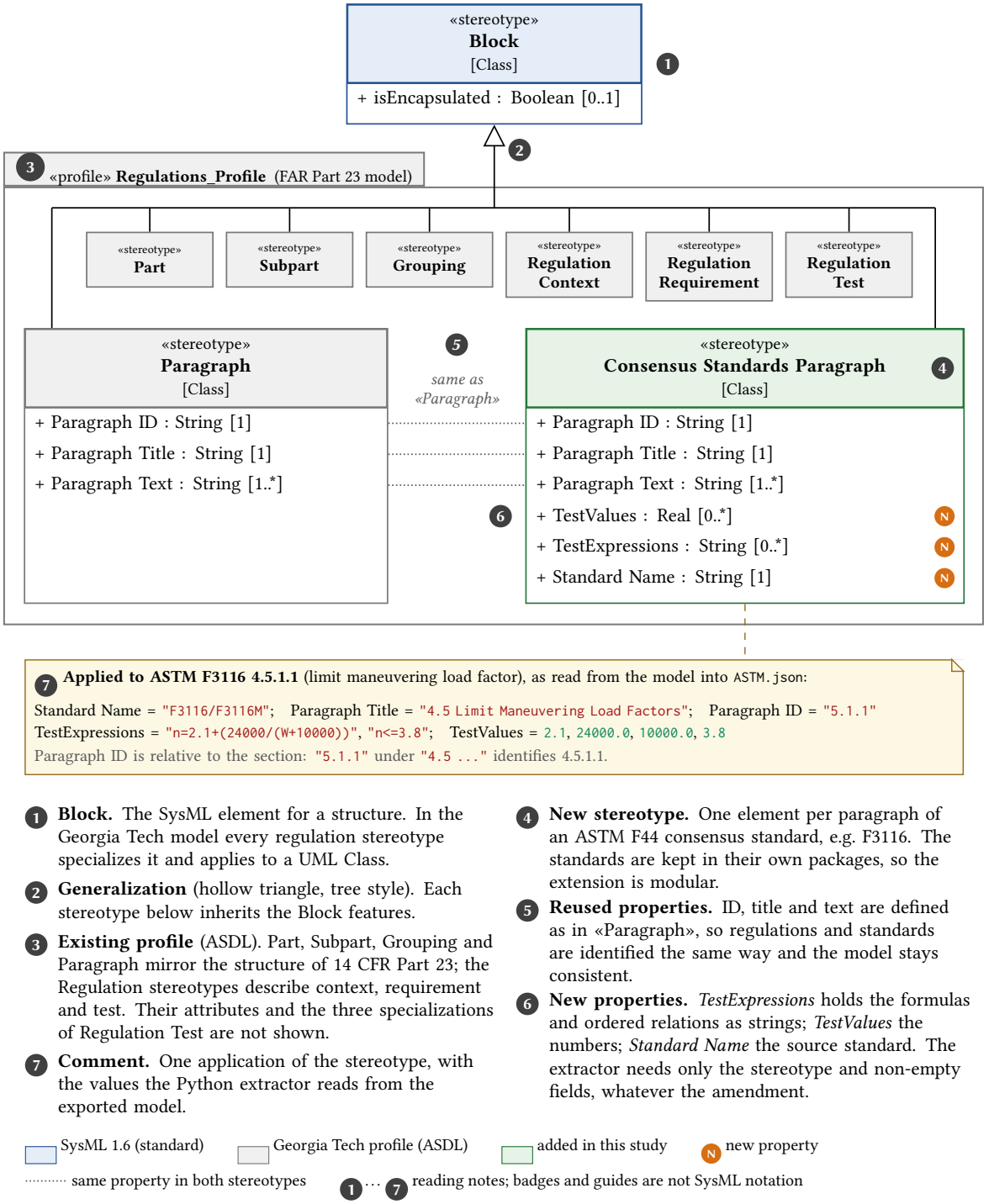
Establishing the connection between FAR 23 paragraphs and the corresponding substantiation standard documents can be cumbersome and, in some cases, particularly difficult, especially when the applicant does not retain within its organization the necessary knowledge, expertise, and know-how. Guidance issued by the authorities may not always provide a complete mapping between airworthiness requirements and the applicable ASTM International standards. In some cases, references may be ambiguous, incomplete, or outdated. Consequently, identifying the appropriate set of substantiation documents often depends on personnel with direct certification experience. Figure 2 illustrates this traceability challenge: to demonstrate compliance with airworthiness requirements, applicants may need to consult several ASTM standards and identify the specific paragraphs that contain the applicable substantiation methods.

The SysML formal language can be used to establish all relationships between FAR 23 paragraphs and ASTM International standards, improving traceability between high-level requirements and methods for showing compliance. The resulting model allows the applicant to initiate substantiation calculations and generate artifacts for the modeled requirements; in this work, a subset of Subpart C. The calculations are then consistent with the encoded content; errors made during encoding carry through to the results, so the encoding itself requires review.

To capture the ASTM substantiation requirements and methods, a new user-defined stereotype has been introduced into the SysML model. This stereotype, named Consensus Standards Paragraph (CSP), similar to the one used to model the FAR 23 paragraphs and subparagraphs, includes several properties that unambiguously define the paragraphs and subparagraphs of the modeled standards. Figure 6 presents all the features introduced to the existing Profile Diagram and how they relate to the previously defined stereotypes. The Figure includes the properties of the CSP, whether reused or newly defined, and their multiplicity.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2



1 **Block.** The SysML element for a structure. In the Georgia Tech model every regulation stereotype specializes it and applies to a UML Class.

2 **Generalization** (hollow triangle, tree style). Each stereotype below inherits the Block features.

3 **Existing profile** (ASDL). Part, Subpart, Grouping and Paragraph mirror the structure of 14 CFR Part 23; the Regulation stereotypes describe context, requirement and test. Their attributes and the three specializations of Regulation Test are not shown.

7 **Comment.** One application of the stereotype, with the values the Python extractor reads from the exported model.

4 **New stereotype.** One element per paragraph of an ASTM F44 consensus standard, e.g. F3116. The standards are kept in their own packages, so the extension is modular.

5 **Reused properties.** ID, title and text are defined as in «Paragraph», so regulations and standards are identified the same way and the model stays consistent.

6 **New properties.** *TestExpressions* holds the formulas and ordered relations as strings; *TestValues* the numbers; *Standard Name* the source standard. The extractor needs only the stereotype and non-empty fields, whatever the amendment.

SysML 1.6 (standard) · Georgia Tech profile (ASDL) · added in this study · N new property
same property in both stereotypes · 1 … 7 reading notes; badges and guides are not SysML notation

Figure 6. The Consensus Standards Paragraph stereotype added to the Profile Diagram of the Georgia Tech FAR 23 model, compared with the existing Paragraph stereotype.

Figure 5 shows the PD used in the FAR 23 SysML model and illustrates the differences between the Paragraph and the CSP stereotypes. Most of the added properties provide clear identification of the standard document to which each element belongs by specifying the title, ID, and standard name of the paragraph. Two of these properties are important in the construction of a robust digital certification pipeline.

1. The standard documents include equations, calculation formulas, and order relations. To capture this information, a property named *TestExpressions* has been introduced. In CATIA MSoSA, this property is defined as a vector of strings with zero-to-many multiplicity, each containing the specific information the user intends to extract from the model.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

2. To ensure that all relevant information is extracted from the paragraphs and subparagraphs of the standard documents, a property named *TestValues* has been introduced. In CATIA MSoSA, this property is defined as a vector of floats with zero-to-many multiplicity, containing numerical values found in formulas, order relations, and equations. Conversion factors are generally excluded from the information extracted from the model.

When modeling the substantiation paragraphs and subparagraphs, the user compiles a list of relevant values, which are numbers included in the textual content of the requirement, within the property field *TestValues*, and a list of equations, formulas, and order relations within the property field *TestExpressions*.

Figure 7 gives an example of CSP stereotype application to paragraph 4.5.1.1 of ASTM F3116/F3116M, *Standard Specification for Design Loads and Conditions*. Paragraph 4.5.1.1 provides the applicant with the mathematical formula to establish the appropriate limit maneuvering load factor n as a function of the aircraft maximum takeoff weight W_{MTO} measured in pounds. It also provides an upper limit to the value of n calculated using the provided formula for a normal-category aircraft. Paragraph 4.5.1.2 provides the equivalent upper limit for an aerobatic category aircraft.

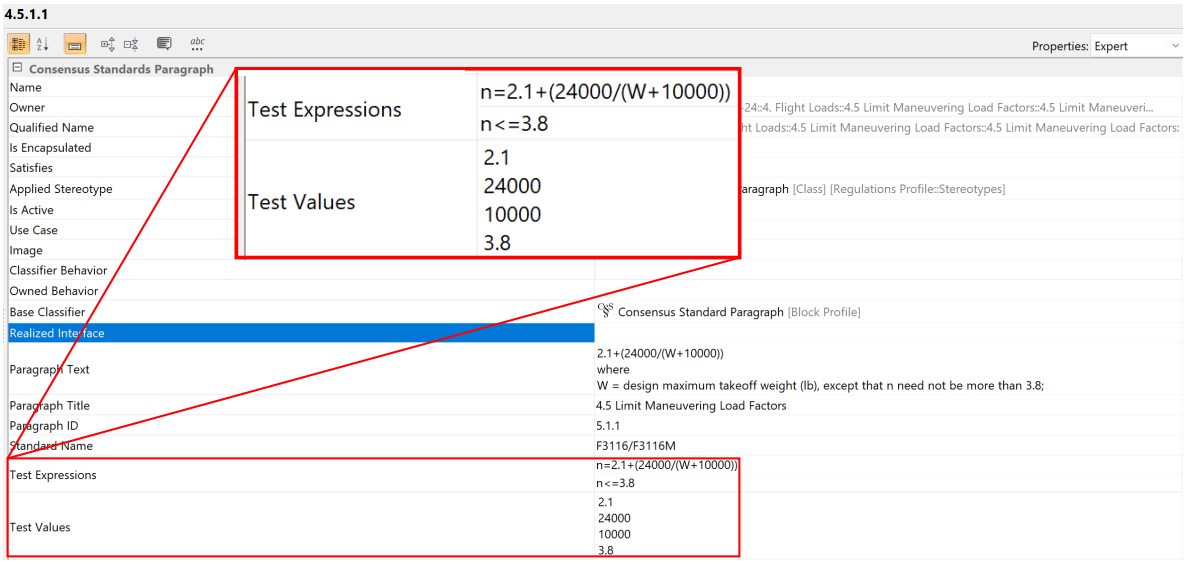


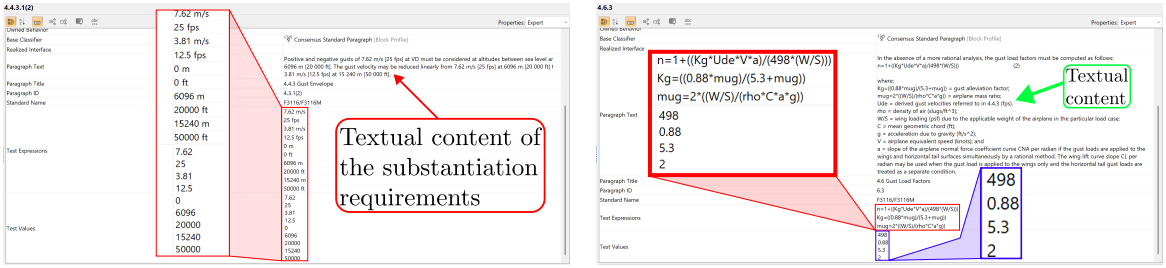
Figure 7. An example of a substantiation method modeled in the CATIA MSoSA environment, namely, Paragraph 4.5.1.1 of ASTM F3116/F3116M, *Standard Specification for Design Loads and Conditions*.

The applicant is instructed to evaluate the applicable limit maneuvering load factor n according to the mathematical formula included in the text of the substantiation method. This information is encoded in the *TestExpressions* property as a vector of strings, which includes both the equation for n and the order relation defining its upper limit. The *TestValues* property is a vector containing the numerical values of the constants referenced in the requirement text. External Python code reads these properties and generates the calculation functions from them, with fallback mechanisms for data extraction and function generation, so that the calculations remain directly linked to the FAR 23 SysML model. When formulas are available as strings, the code interprets them and generates the corresponding calculation functions. If no formula is available, it reconstructs the required expression from the stored numerical coefficients.

Figure 8 gives two additional examples extracted from the CATIA MSoSA modeling environment. Figure 8(**a**) illustrates how the textual content of ASTM F3116/F3116M, paragraph 4.4.3.1(2), has been condensed using the properties *TestExpressions* and *TestValues*, while Figure 8(**b**) shows the same information for ASTM F3116/F3116M, paragraph 4.6.3. In both cases, the textual content is in colored boxes for clarity.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2



(**a**) *ASTM F3116/F3116M, par. 4.4.3.1(2)* (**b**) *ASTM F3116/F3116M, par. 4.6.3*

Figure 8. Two examples of ASTM International substantiation requirements.

2.2. Substantiation Methods Encoding

To use the information encoded in the model, it is exported as a Unified Modeling Language (UML) standard file, which is functionally equivalent to an XMI/XML format. This file can be easily parsed by a computer and represents the most direct and convenient means of communication with external tools originating from CATIA MSoSA proprietary files, such as the Python code developed and described in this subsection. This approach is considered acceptable and consistent with the capabilities available in the CATIA MSoSA implementation examined in this study, which does not provide a native API-based export mechanism suitable for the proposed workflow.

To assist the user in understanding and monitoring the parsing process, the extracted data are first exported to a JSON file. The JSON file is a convenient format for structured data and is handled using well-established Python packages, such as the *pandas* package. At this stage, the data can be readily loaded and used. Figure 9 shows a schematic representation of the information extraction phase. The resulting SysML model is then exported as a standard UML file and parsed using an external Python routine, which searches for elements in the model that contain the *TestExpressions* and *TestValues* properties with non-empty fields. Once the routine has explored the entire model, it exports the data to a JSON file for convenience. All substantiation encoding was performed in Python. This choice is justified by the following considerations:

- Python offers a wide range of mature packages, such as *sympy* for symbolic computation, that facilitate rapid development.
- The authors of this work explored the use of other programming languages, such as C++. Although C++ was used in earlier work by the authors [27], it was excluded here due to time constraints and the complexity of implementing external software interfaces, for example, to evaluate spanwise load distribution along the main wing using tools such as OpenVSP.
- Previous applications developed by research engineers at the Aerospace Systems Design Laboratory of the Georgia Institute of Technology relied extensively on Python code; therefore, continuing development in Python was the most practical and consistent choice.

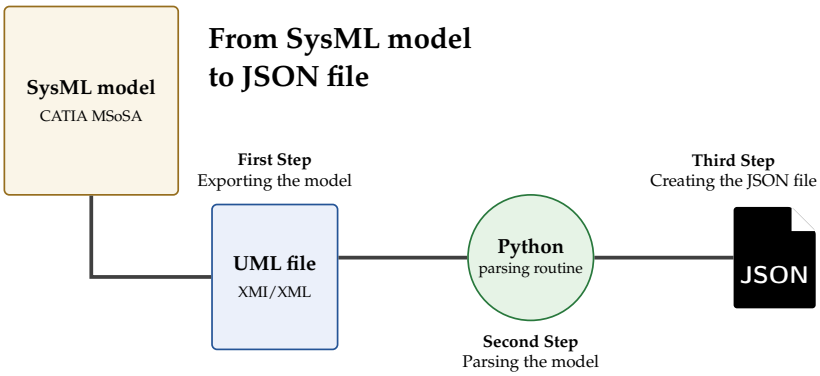


Figure 9. A schematic representation of the exporting phase. A UML standard file has been exported from the CATIA MSoSA environment. The UML file is parsed using an external Python routine. The program extracts the data and creates a JSON file.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

Figure 10 elaborates on the conceptualization phase, clarifying the underlying logic adopted to digitalize each single step, from regulatory documents to certification artifacts. It also schematically describes the conceptual approach used to establish a feasible digital certification path. Figure 11 presents the data exchange mechanism in detail. The symbol M indicates a step that must be performed manually, while the symbol A concerns an automated procedure. The workflow is articulated in 7 steps, from the single source of truth represented by the digitally encoded regulatory framework to the creation of the corresponding substantiation material. Listing 1, included in Appendix A, presents a brief excerpt from the ASTM JSON file, corresponding to the same paragraph shown in Figure 7. The JSON file serves as a structured medium for importing variables into the Python environment, typically as dictionary objects. The dictionary derived from the SysML model is named `ASTM`, and the corresponding JSON file is organized according to the model elements' name and ID nomenclature.

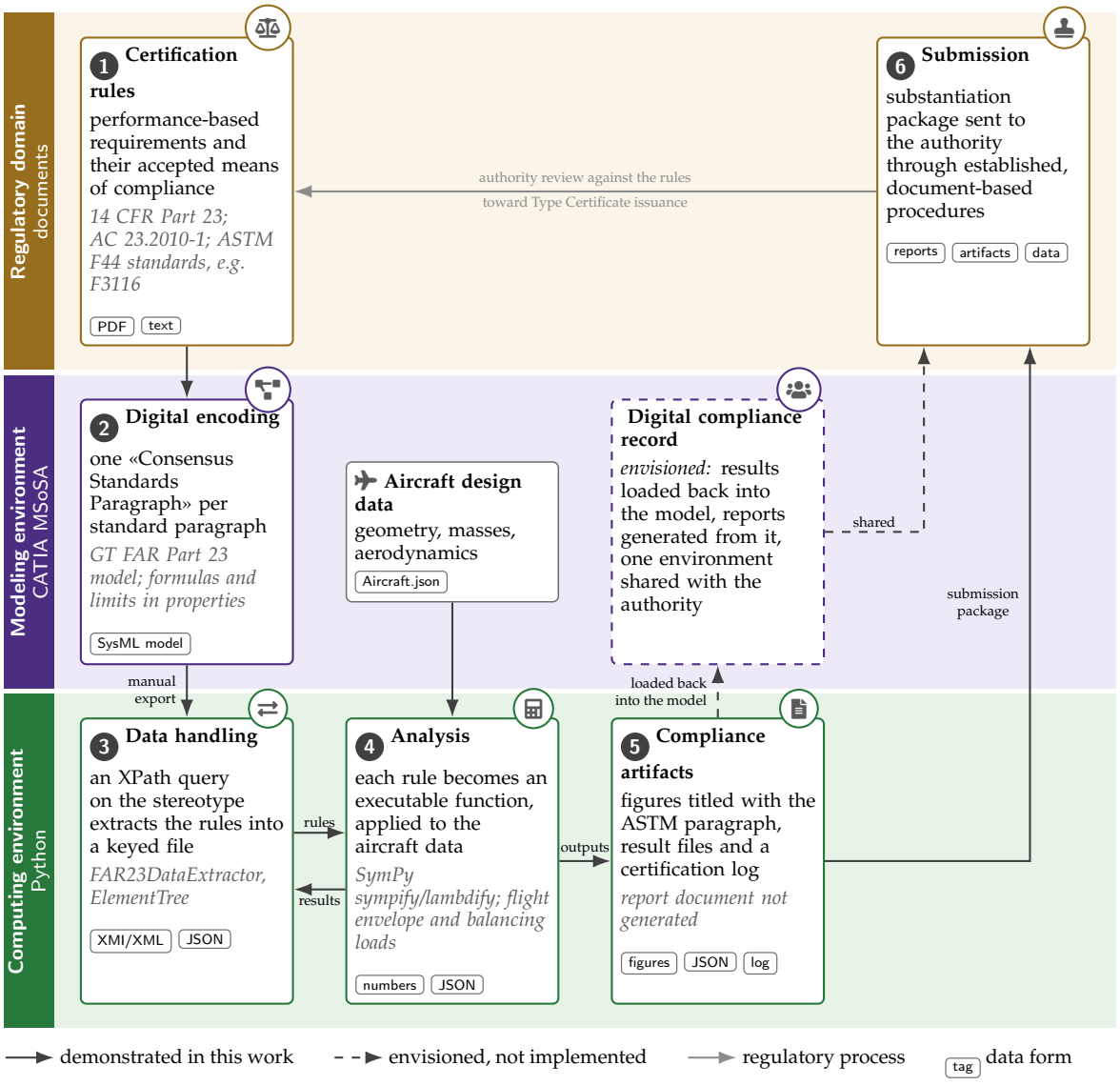


Figure 10. The general conceptualization of digital certification: six phases placed in the environment where they take place, with the data each phase produces. Solid arrows are demonstrated; dashed arrows are envisioned and not implemented.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

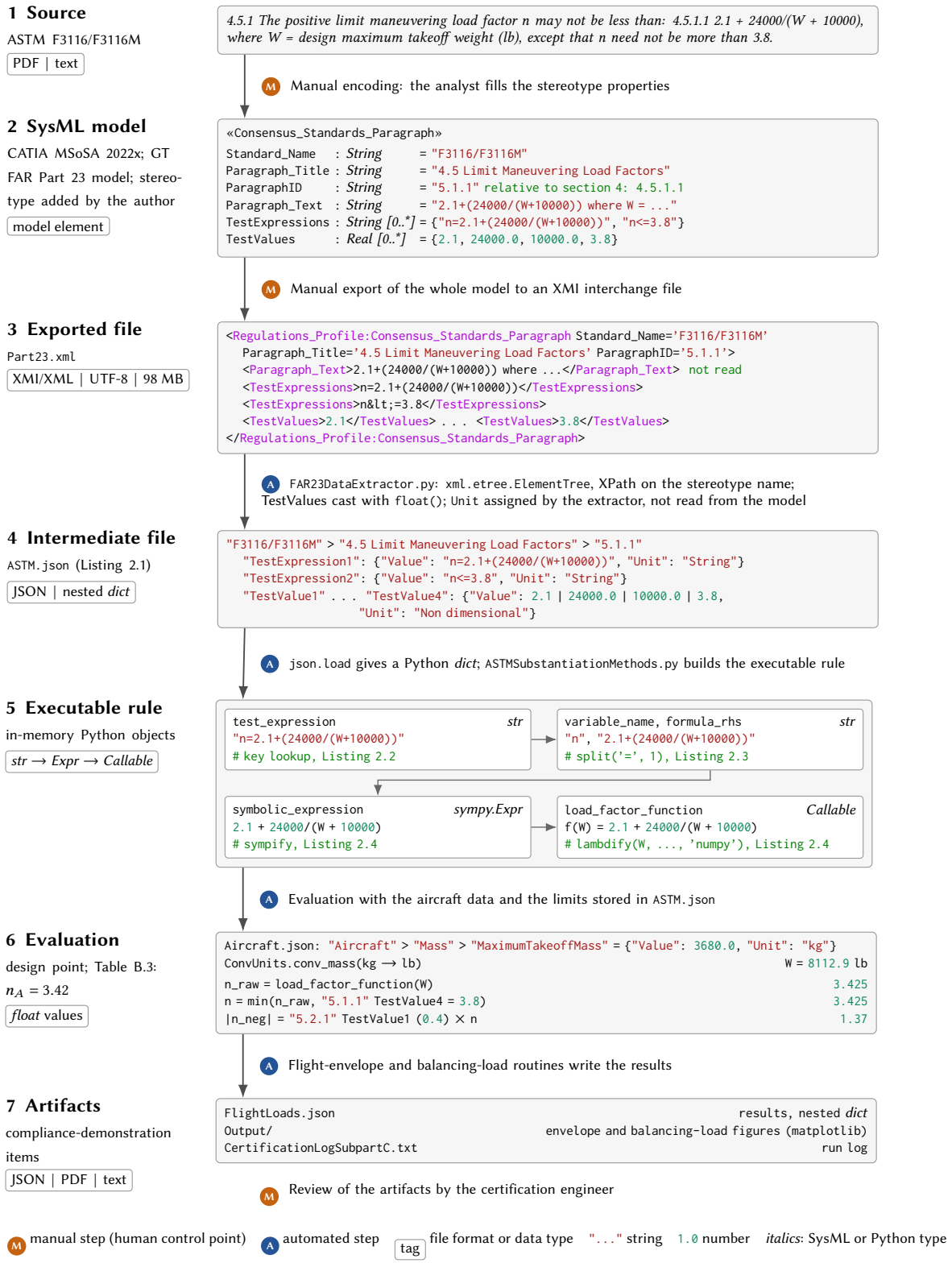


Figure 11. Data exchange in the CATIA MSoSA/Python framework, traced for ASTM F3116 4.5.1.1. Each stage shows its content and data type; M marks a manual step, A an automated one. Stages 4 and 5 correspond to Listings 1 to 4.

These dictionaries are then used to expand the aircraft database, producing results from codes that implement substantiation methods. The results include, but are not limited to, the preliminary calculations concerning geometrical, inertia, weight, and certification-related properties, the maneuvering and gust flight envelope branches, and the balancing loads data. During the code's normal execution, the data obtained throughout the calculation are included in a JSON file named `FlightLoads.json`,

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

while the points and the vectors representing the various components of diagrams, such as the flight envelope or the balancing loads diagram, are handled separately to generate the graphical outputs more easily.

The code is designed to extract each non-empty field from the *TestExpressions* and *TestValues* properties, leveraging the nested, hierarchical structure inherent to the JSON standard. Listings 2, 3, and 4 illustrate how the Python code operates on the data, using the same example described in Figure 7 and Listing 1. All the cited listings are included in Appendix A. The Python code automatically creates functions from the information stored in the JSON file using the symbolic computation package *sympy* to establish the input variables, in combination with the helper function *lambdify*, which enables the creation of executable functions from strings.

This strategy links the Python code directly to the substantiation methods contained in the model. When the model changes, the functions are regenerated from the new content, provided that variable names and expression format follow the conventions adopted in the model. The code includes error handling and fallback mechanisms. Specifically, if errors arise from the *TestExpressions* fields, the code reverts to the *TestValues* fields, defining the equations using numerical data. This fallback increases the likelihood of successful execution during this phase of the process.

The coding strategy illustrated in this section reflects the rationale and logic adopted by the authors in developing the substantiation methodologies proposed by ASTM International substantiation documents. The provided example shows the procedures and coding strategies implemented throughout the Python code structure. Although the code was developed in accordance with conventional Python best practices, the regulatory formulas used to generate the numerical and graphical outputs presented in this work were defined using information extracted from the SysML regulations model. This direct link between the encoded regulatory content and the calculations is the main contribution of the present work. Changes to airworthiness and substantiation rules can be reflected in the calculations by updating the model, within the encoding and exporting conventions established in this study.

2.3. Evaluation Design

The framework is evaluated against four research questions, summarized in Figure 12. RQ1 concerns the encoding: the regulatory formulas and coefficients of the modeled paragraphs should be read from the export of the SysML model rather than written into the analysis code. The engineering models that the standard does not prescribe, such as the trim equations, the pitch contribution to the inertial load, and the static thrust, are implemented in the code and stated as assumptions in Section 3. The evidence is the data-exchange pipeline described in this section and the code excerpts of Appendix A. RQ2 concerns correctness: the outputs must satisfy the definitions of ASTM F3116/F3116M and the equilibrium equations at every point of the flight envelope boundary. RQ3 concerns consistency with independent data: the results of the P2012 and P2006 cases are compared with non-public flight loads data of the aircraft manufacturer, and every difference must be traced to a stated difference in assumptions or data. RQ4 concerns the behavior with faulty input data: a fault must stop the run, be corrected with a warning, or be flagged in the results. The evidence for RQ2 to RQ4 is reported in Section 3.3. The practical benefit of the framework for applicants and authorities is not evaluated in this study.

Table 1 lists the paragraphs of ASTM F3116/F3116M that are addressed by the calculations reported in this study, together with those that are outside its scope.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

Table 1. Paragraphs of ASTM F3116/F3116M [31] addressed in this study.

Paragraph	Requirement	Treatment in this study	Status
4.2.1	Definition of the flight load factor	Vertical-force equation of the balancing loads, Eq. (1)	Covered
4.2.2	Compliance at each critical altitude, weight, and load distribution	One CG and mass combination per aircraft, Table A1	Partly covered
4.3.1, 4.16.1	Balancing tail load, in a rational or conservative manner	Force and moment equations solved together, Section 3.1	Covered
4.4.2	Maneuvering envelope	Figure 13, Table A2	Covered
4.4.3.1(1)–(2)	Gusts of 50 ft s^{-1} at V_C and 25 ft s^{-1} at V_D	Figure 13, Table A2	Covered
4.4.3.1(3)	Level 4 airplanes: rough-air gusts of 66 ft s^{-1} at V_B	–	Not covered
4.5.1, 4.5.2	Positive and negative limit maneuvering load factors	Table A2	Covered
4.6.3	Gust load factors	Figure 13, Table A2	Covered
4.8	High-lift devices	–	Not covered
4.16.2	Balancing loads on the limit maneuvering envelope and in the flap conditions	Envelope boundary, Figure 14, Tables A3 and A4; flap conditions not covered	Partly covered
5.1.1–5.1.3	Design airspeeds V_C, V_D, and V_A	Section 3.1, Table A2	Covered
5.1.4	Design speed for maximum gust intensity V_B	–	Not covered
6.1.1.1	Takeoff torque with 75 % of the limit loads of condition A	–	Not covered
6.1.1.2, 6.1.3	Limit engine torque with the limit loads of condition A	Eqs. (2) and (3), Table A5	Covered
6.1.1.3, 6.1.2	Turbopropeller and turbine engine torque	Piston engines only	Not applicable
6.2	Side load on the engine mount	Eq. (5), Table A6	Covered
6.3.1.2	Gyroscopic, inertial, and aerodynamic loads	Eqs. (6)–(8), Table A6	Covered
6.3.3	Level 4 airplanes: engine mount loads in the gust conditions of 4.6	Eq. (4), Table A6	Partly covered

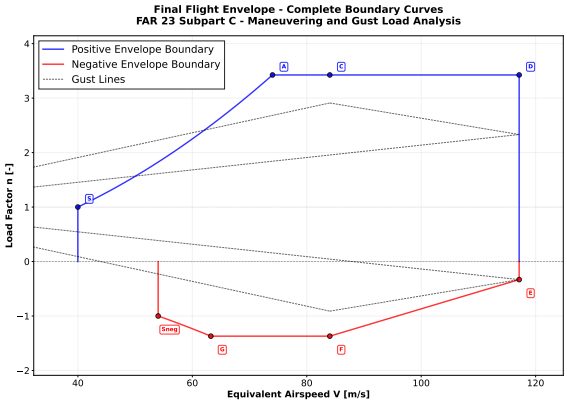
Research question	Criterion	Evidence	Where
RQ1 **Encoding.** Can the substantiation content of ASTM F3116 be held in the SysML model and supplied from there to the analysis code?	The regulatory formulas and coefficients of the modeled paragraphs are read from the model export.	Data-exchange pipeline; ASTM JSON file; code excerpts.	Section 2; Appendix A
RQ2 **Correctness.** Do the outputs satisfy the definitions of the standard and the equilibrium equations?	Load factor and balancing load as defined in F3116; force and moment residuals close to zero at every envelope point.	Definition checks; residuals at all trim points of the three aircraft; defects found and corrected.	Section 3.3
RQ3 **Consistency.** How do the outputs compare with independent flight loads data?	Each difference is explained by a stated difference in assumptions or data.	Agreement bands against non-public manufacturer data (P2012, P2006).	Section 3.3
RQ4 **Faulty input.** What does the framework do with wrong, missing, or inconsistent data?	The fault stops the run, is corrected with a warning, or is flagged; silent behavior counts as a defect.	Faults met while preparing the three input files.	Section 3.3

Not evaluated in this study: the practical benefit for applicants and authorities, which requires an evaluation by certification practitioners.

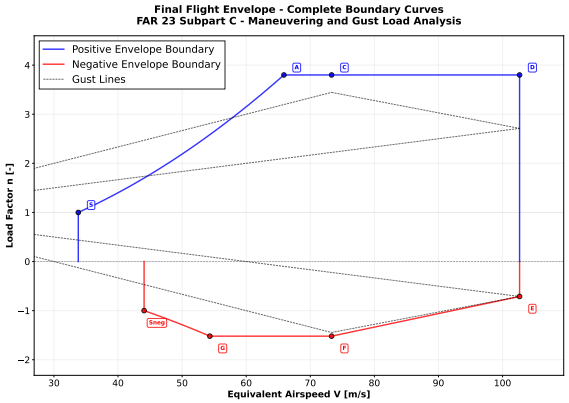
Figure 12. Evaluation design: research questions, criteria, evidence, and where the evidence is reported.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

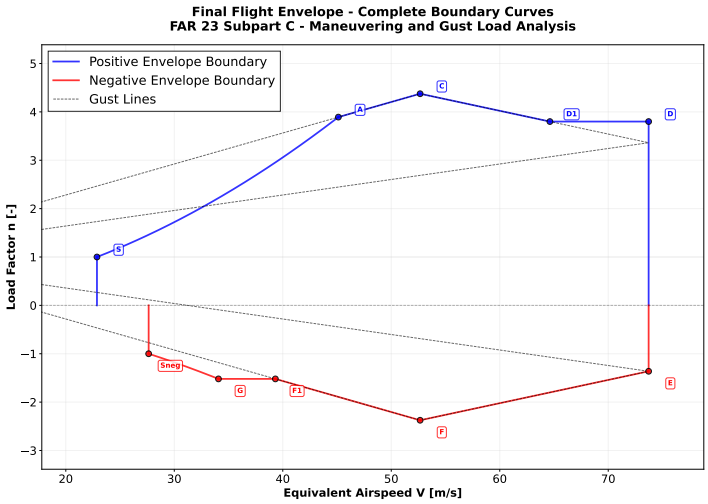
Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2



(**a**) *P2012 flight envelope.*



(**b**) *P2006 flight envelope.*



(**c**) *P92 flight envelope.*

Figure 13. The maneuvering and gust flight envelope for three different aircraft, generated by the Python code.

3. Results

3.1. Maneuvering and Gust Flight Envelope

This section presents a selection of results obtained by executing the code. The example architectures chosen for this study span the range of general aviation aircraft eligible for certification under the FAR 23 regulations.

1. The first aircraft selected is inspired by the Tecnam P2012, a Level 4, low-speed, normal-category, eleven-seat, twin-engine aircraft with a maximum takeoff mass of 3680 kg.
2. The second aircraft is inspired by the Tecnam P2006, a Level 2, low-speed, normal-category, four-seat, twin-engine aircraft with a maximum takeoff mass of 1400 kg.
3. The final aircraft used to test the digital certification framework is inspired by the Tecnam P92, a Level 2, small, low-speed, normal-category, two-seat, single-engine aircraft with a maximum takeoff mass of 600 kg.

The code can distinguish among different cases based on a variety of parameters, such as the aircraft weight, reference certification altitude, center-of-gravity (CG) position, and inertia properties. The most prominent example of this capability is the maneuvering and gust flight envelope, as defined by ASTM F3116/F3116M, *Standard Specification for Design Loads and Conditions*, which varies according to the aircraft's weight and the resulting impact of gusts on limit loads.

Figure 13 shows the maneuvering and gust flight envelope for the selected architectures. The design airspeeds that bound the envelope follow ASTM F3116/F3116M [31]. The design cruising

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

speed V_C, in knots, may not be less than $33\sqrt{W/S}$, with the wing loading W/S in lb ft^{-2} (the factor 33 decreases linearly for wing loadings above 20 lb ft^{-2}); however, V_C need not exceed 0.9 V_H, where V_H is the maximum speed in level flight with maximum continuous power. The code estimates V_H at sea level by equating the power required, computed with a parabolic drag polar, to the power available at maximum continuous power. For all three aircraft, 0.9 V_H is lower than the minimum derived from the wing loading, so it sets the design cruising speed: $V_C = 0.9\,V_H$ is 84.0 m s^{-1} (163.3 kn), 73.3 m s^{-1} (142.5 kn), and 52.7 m s^{-1} (102.4 kn) for the P2012, P2006, and P92, respectively, against wing-loading minimum values of 91.3 m s^{-1} (177.4 kn), 75.7 m s^{-1} (147.1 kn), and 53.1 m s^{-1} (103.2 kn). The design dive speed V_D is then obtained from this value of V_C. Figures 13(**a**) and 13(**b**) present the flight envelopes for the P2012 and P2006 aircraft, respectively. The gust load factors stay below the positive limit maneuvering load factor prescribed by ASTM F3116/F3116M; as a result, the gust lines do not change the positive side of the flight envelope diagram; on the negative side, the branch between points F and E ends at the value of the negative gust line at the design dive speed (Table A2).

Figure 13(**c**) illustrates a different scenario for the P92, the smallest aircraft tested. In this case, the gust load factors exceed the limit maneuvering load factors on both the positive and negative sides, becoming the dominant factor in sizing the aircraft structure. The code is designed to dynamically allocate the points and vectors that define the final flight envelope. The evaluation of complex intersections between the gust lines and the branches of the maneuvering and gust flight envelope is performed using the Python symbolic computation package *sympy*, which solves the corresponding mathematical problems to determine the intersection points. To ensure robustness, the code handles potential data type issues by converting symbolic floats into more suitable data types for subsequent processing. Users can analyze different aircraft architectures by preparing the input file accordingly and adhering to the established variable naming conventions. In Appendix B, Table A2 provides the numerical values of airspeed and limit load at the fundamental flight envelope points. These values show that the flight envelope, built using ASTM International substantiation methods, is adapted automatically to the aircraft weight, geometry, and aerodynamic characteristics.

The gust load factor increment scales inversely with wing loading [42,43]. This explains the ordering in Figure 13: the P92 has the lowest wing loading of the three (468 Pa, against 952 Pa for the P2006 and 1430 Pa for the P2012, Table A1), and it is the only one with gust-governed branches. The result does not depend on aircraft weight alone. The code selected these branches with no aircraft-specific change, which is the behavior this figure is intended to demonstrate.

Figure 14 shows the horizontal tailplane lift resulting from trimming the aircraft along the boundaries of the flight envelope presented in Figure 13. At every point of the envelope boundary, the code employs the wing-body aerodynamics to solve the vertical-force and pitching-moment equations together, with no pitching acceleration, as required for balancing loads by ASTM F3116/F3116M [31]. The vertical-force equation follows the definition of the flight load factor given in the same standard, that is, the ratio of the aerodynamic force component acting normal to the assumed longitudinal axis of the airplane to the weight of the airplane:

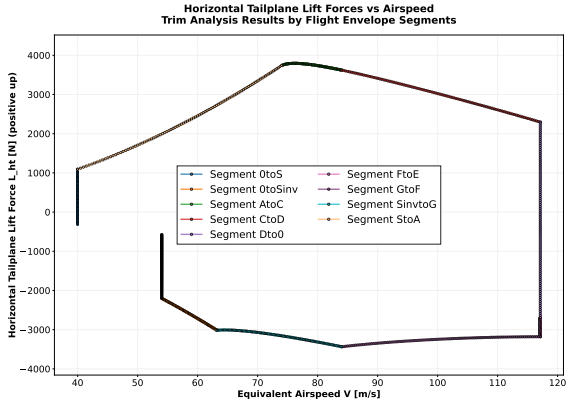
$$N = L_{\mathrm{wb}} \cos\alpha + D \sin\alpha + L_{\mathrm{ht}} = nW \tag{1}$$

where L_{wb} and D are the wing-body lift and drag, normal and parallel to the free stream, respectively, and L_{ht} is the horizontal tailplane force, normal to the longitudinal axis and positive upward. The thrust is assumed equal to the drag and parallel to the longitudinal axis, so it enters only the pitching-moment equation. Some manufacturer flight loads reports write $nW\cos\alpha$ on the right-hand side of Equation (1); with the regulatory definition adopted here, the wing-body lift is higher by up to about 3% of nW at the largest angles of attack. If the equilibrium required a wing-body lift coefficient beyond the maximum or minimum value of the wing-body lift curve, the code would hold it at that limit, balance the pitching moment, and report the load factor that the airplane can actually reach at that airspeed; this situation does not occur in the cases presented here. In this process, the code evaluates

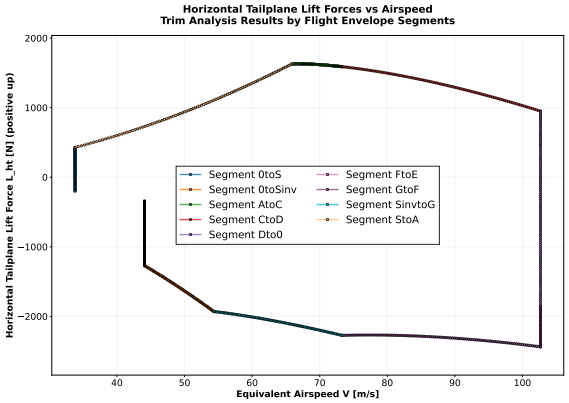
© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

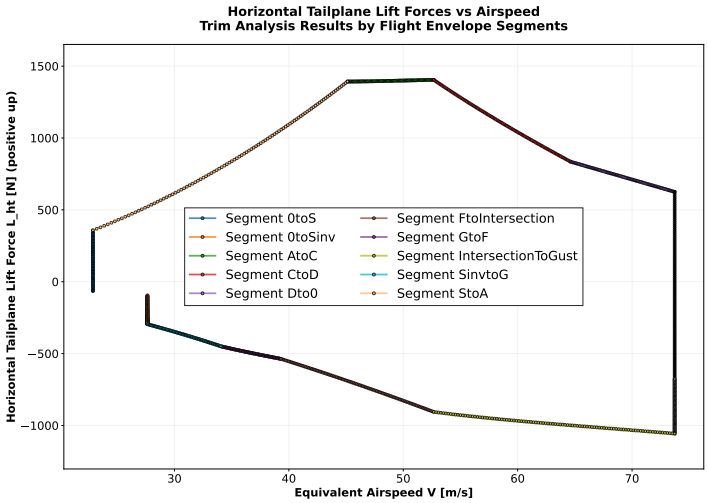
the wing-body lift, the full-vehicle lift $L_{\text{fv}} = N$, and the horizontal tailplane lift. All forces acting on the aircraft are expressed in newtons.



(**a**) *P2012 horizontal tailplane lift.*



(**b**) *P2006 horizontal tailplane lift.*



(**c**) *P92 horizontal tailplane lift.*

Figure 14. The horizontal tailplane lift for three different aircraft, generated by the code.

Figures 14(**a**) and 14(**b**) refer to the P2012 and P2006 aircraft, respectively. As expected, the gust has no significant effect on the horizontal tailplane loads; the primary influence arises from maneuvering.

In contrast, Figure 14(**c**) illustrates the impact of gust limit loads on the final shape of the diagram. The tail load is not proportional to the load factor, because the trim balance combines a term that scales with nW through the CG position and a term that scales with dynamic pressure through the wing-body pitching moment [44]. On the positive side, the largest tail load occurs between points A and C for the P2012 (3795 N at 76.1 m s^{-1}) and for the P2006 (1631 N at 66.7 m s^{-1}), and at point C for the P92 (1405 N). On the negative side, the largest magnitude occurs at point F for the P2012 (-3435 N), at point E for the P2006 (-2437 N), and at point E for the P92 (-1057 N). These loads come from a wing-body aerodynamic model with estimated data, so they demonstrate the procedure and do not estimate the loads on the actual aircraft.

Tables A3 and A4 included in Appendix B provide part of the numerical values used to draw these diagrams, including the dynamic pressure, wing-body, full-vehicle, and horizontal tailplane lift coefficients, the angle of attack, and the wing-body, full-vehicle, and horizontal tailplane lift.

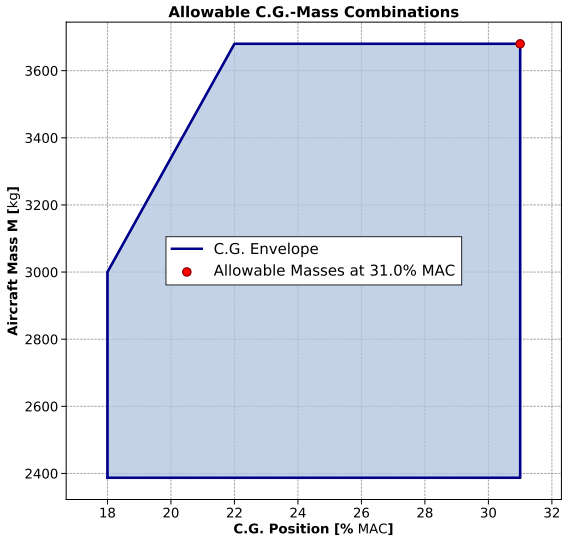
To achieve this flexibility, the code first evaluates the geometric and mass quantities required to calculate the balancing loads acting on the horizontal tailplane, including:

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

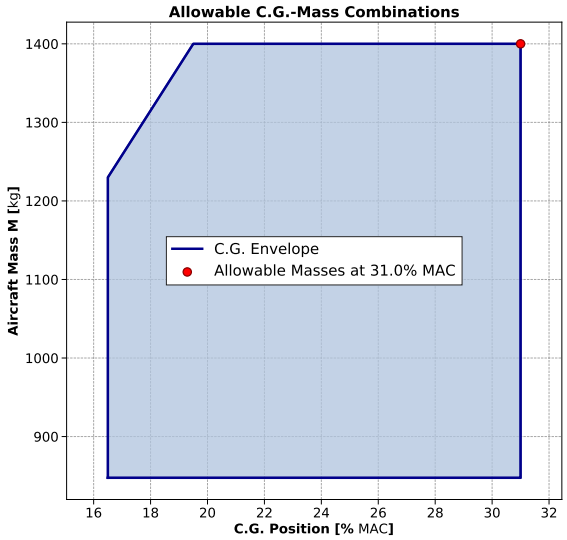
Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

- Allowable combinations of CG position and aircraft mass.
- Relevant moment arms.
- Vertical position of the CG.
- Moments of inertia.
- Position of the thrust axis.

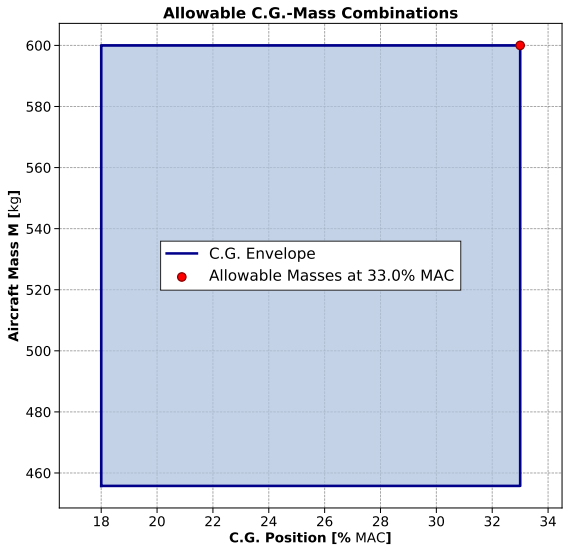
As a consistency check, the code plots the CG envelope, reported in Figure 15. The CG envelope also includes the analysis case selected by the user, as a red dot on the diagram. The results are conditional on this single CG and mass combination; a compliance demonstration would need to cover the critical combinations of the envelope.



(**a**) *P2012 allowable CG/mass envelope.*



(**b**) *P2006 allowable CG/mass envelope.*



(**c**) *P92 allowable CG/mass envelope.*

Figure 15. The allowable CG/mass envelope for the three selected aircraft architectures.

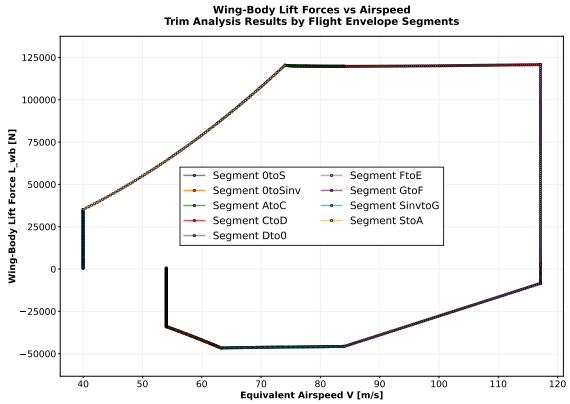
In Appendix B, Table A1 presents a selection of parameters evaluated during code execution. Most of the data pertain to the geometrical configuration of aircraft components and the CG and mass combinations selected for conducting the flight loads analysis. To enable these calculations, the user must provide input values, including the moments of inertia, the vertical position of the CG, and the

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

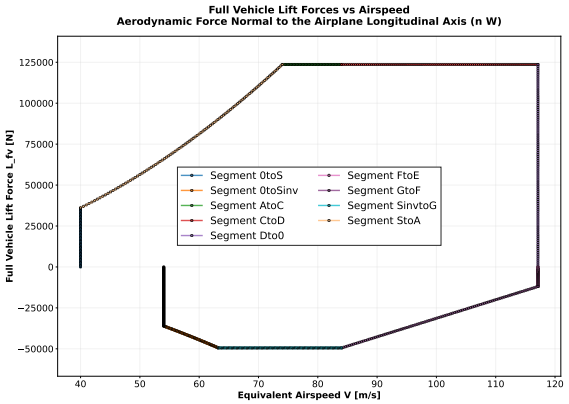
Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

location of the thrust axis relative to the CG across different CG and mass configurations. The code interpolates between the boundary conditions specified by the user.

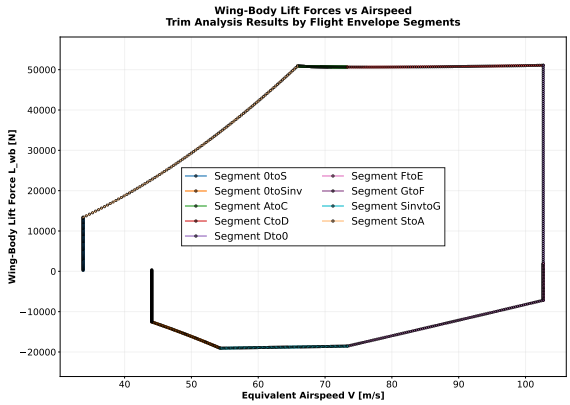
Finally, Figure 16 presents the wing-body and full-vehicle lift forces computed by the Python code for the three aircraft. These diagrams are included for completeness. Notably, Figures 16(**e**) and 16(**f**) once again highlight the significance of gust limit load factors in sizing the structures of small, light aircraft.



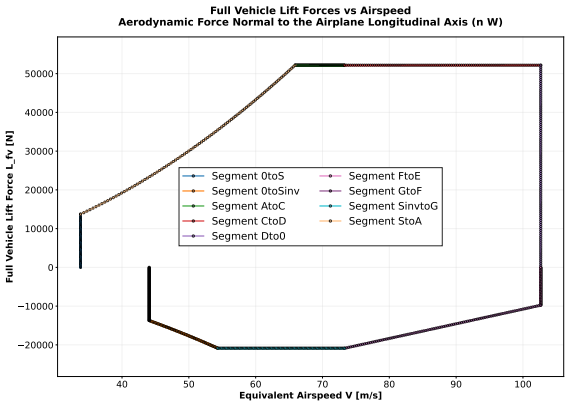
(**a**) *P2012 wing-body lift diagram.*



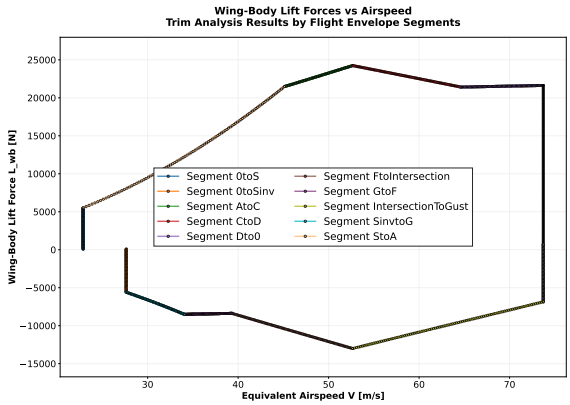
(**b**) *P2012 full-vehicle lift diagram.*



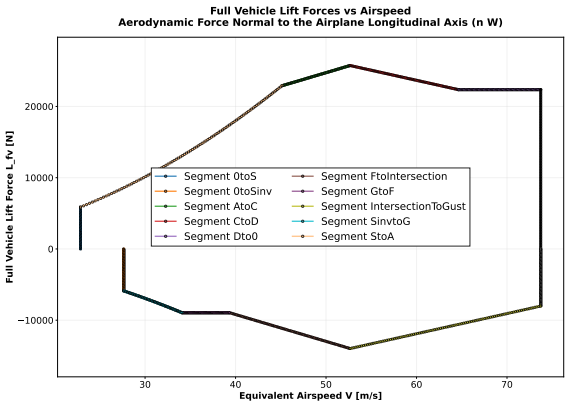
(**c**) *P2006 wing-body lift diagram.*



(**d**) *P2006 full-vehicle lift diagram.*



(**e**) *P92 wing-body lift diagram.*



(**f**) *P92 full-vehicle lift diagram.*

Figure 16. Wing-body and full-vehicle lift forces for the three aircraft architectures studied.

At every point, the full-vehicle lift of Equation (1) equals nW, and the wing-body and horizontal tailplane contributions close this balance; for example, at point A of the P2012 ($\alpha = 13.1309$ deg, $D = 11\,719.7$ N), $120\,348.5\,\text{N} \cdot \cos\alpha + 11\,719.7\,\text{N} \cdot \sin\alpha + 3739.2\,\text{N} = 117\,201.9\,\text{N} + 2662.4\,\text{N} + 3739.2\,\text{N} =$

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

123 603.5 N, which equals $n_A W = 3.425014 \times 36\,088.47$ N (Tables A1 and A3). This is an internal consistency check of the equilibrium solution; it does not validate the aerodynamic data.

In Appendix B, Tables A3 and A4 present the data concerning these diagrams at relevant points of the flight envelope, on the positive and negative sides, respectively. The full-vehicle lift diagrams therefore reproduce the shape of the flight envelope, scaled by the aircraft weight; the wing-body lift diagrams differ from them by the horizontal tailplane lift shown in Figure 14 and by the projection of the wing-body lift and drag onto the normal to the longitudinal axis.

This section illustrates the complexity of the substantiation process for aircraft certifiable under FAR 23 airworthiness rules. Variations in aircraft architecture lead to different branches of the substantiation methods defined by the accepted standards.

In the pipeline described in this work, the Python code selects the applicable methods from the aircraft data before initiating the substantiation calculations, so that the same selection logic is applied to each architecture.

Whether this reduces the need for direct human intervention, or improves transparency and oversight in an actual certification project, has not been evaluated in this study and would require assessment with practitioners and authorities.

3.2. Engine Loads

In Section 3.1, the Python calculation pipeline used the SysML model to evaluate the flight envelope and the balancing loads. Many of the calculations needed to substantiate FAR 23, Subpart C, rely on the evaluation of the maneuvering and gust flight envelope, specifically, on the values of the limit load factors calculated during the flight envelope analysis. In this section, another example of automated application of the FAR 23, Subpart C, substantiation methods is illustrated.

ASTM F3116/F3116M addresses engine load calculations in Section 6. The calculations are simple, but the multiplying factor depends on the engine type and on the number of cylinders. The Python routine that evaluates the engine loads extracts the multiplying factor that applies to each case. The calculations begin with the definition of a mean engine torque at maximum continuous power (MCP):

$$\text{Mean Engine Torque} = T_{\text{mean}} = \frac{P_{\text{mc}}}{\text{RPM}_{\text{mc}} \cdot \frac{2 \cdot \pi}{60}} \tag{2}$$

where P_{mc} is the maximum continuous power measured in W and RPM_{mc} is the engine rotation speed at maximum continuous power measured in rpm. The multiplying factor is indicated with the symbol MF. This value depends on the engine type installed on the aircraft. The limit engine torque is evaluated with the following formula:

$$\text{Limit Engine Torque} = \text{MF} \cdot T_{\text{mean}} \tag{3}$$

The torques from Equations (2) and (3) are used in subsequent calculations for the engine load conditions considered in this study.

Next, the inertial loads acting on the powerplant structural support are calculated as the product of a combined limit load factor—derived from the superposition of the gust limit load factor, evaluated at the design cruising speed, and the limit load factor associated with a pitch maneuver—and the overall system weight of the powerplant. The pitch contribution is the rotational load factor produced at the engine CG by the pitching acceleration that the incremental gust lift, acting at the wing-body aerodynamic center, induces about the airplane CG:

$$\dot{q} = \frac{W\,(n_{\text{gust}} - 1)\,\left|x_{\text{ac,wb}} - x_{\text{cg}}\right|}{I_{yy}}, \qquad n_{\text{rot}} = \frac{\dot{q}\,\left|x_{\text{cg}} - x_{\text{pp}}\right|}{g}, \qquad F_{\text{inertial}} = \left(n_{\text{gust}} + n_{\text{rot}}\right) W_{\text{engine}} \tag{4}$$

where x_{pp} is the longitudinal position of the powerplant group CG and I_{yy} is the pitch moment of inertia, scaled with the airplane mass. The code evaluates both the maximum takeoff mass and the

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

operating empty mass at a reference altitude selected by the user, with the gust load factor computed from the air density at that altitude; the larger inertial load governs.

Another calculation required by the substantiation methods is the side load acting on the structural support of the powerplant. The limit load factor must be evaluated as follows:

$$\text{Side Limit Load Factor} = n_{\text{side}} = \max\left(\frac{1}{3} \cdot n_A, 1.33\right) \tag{5}$$

where n_A is the limit load factor pertaining to the maneuvering and gust flight envelope at the design maneuvering speed. The resulting n_{side} obtained from Equation (5) is applied to the powerplant weight expressed in newtons.

Finally, a series of calculations addresses the gyroscopic and aerodynamic loads. ASTM F3116/F3116M offers two options for these loads; this study adopts paragraph 6.3.1.2, which requires all possible combinations of the following conditions, with the engine at maximum continuous rpm:

- A yaw rate of 2.5 rad s^{-1}.
- A pitch rate of 1.0 rad s^{-1}.
- A normal load factor of 2.5.
- Maximum continuous thrust (MCT).

The pitch and yaw gyroscopic couples may be calculated using the following formula:

$$T_{\text{gyro}} = I_{\text{polar}} \cdot \omega_1 \cdot \omega_2 \tag{6}$$

where I_{polar} is the polar moment of inertia of the propeller measured in kg m^2, and the angular velocities ω_1 and ω_2 represent the propeller rotation speed and the pitch or yaw rate, respectively, both expressed in rad s^{-1}. The possible combinations of loads include the evaluation of a normal load and the load associated with an MCP flight regime, defined by the following equations:

$$\text{Normal Load} = L_{\text{normal}} = n_{\text{normal}} \cdot W_{\text{engine}} \tag{7}$$

$$\text{MCT} = \left(\eta_{\text{average}} \cdot \text{MCP}\right)^{2/3} (2\,\rho_0\,A)^{1/3}, \qquad A = \frac{\pi D^2}{4} \tag{8}$$

where n_{normal} is defined by the substantiation methods, η_{average} is an average propeller efficiency, ρ_0 is the sea-level air density, and D is the propeller diameter. The maximum continuous thrust is estimated with the actuator-disc (momentum) theory in static conditions, which gives the largest thrust that the propeller can deliver at MCP and avoids the singularity of a power-over-speed formulation at low airspeed.

The data enabling these calculations are indicative of general aviation aircraft and were obtained by comparing similar aircraft architectures. The results may differ from those of the actual aircraft. They are considered adequate for the purpose of this work, which is to illustrate the digital certification pipeline. Appendix B presents the numerical results of these calculations in tabular form. The results were generated by the Python code suite using certification information extracted from the model.

3.3. Verification of the Framework

Figure 17 summarizes the verification of the framework. For RQ2, the balancing loads are computed by solving the vertical-force and pitching-moment equations together, with the flight load factor and the balancing load defined as in ASTM F3116/F3116M (Equation (1)). At every trim point of the three aircraft, the residual of the vertical-force equation is below 0.01 N, and the residual of the moment equation is at the level of round-off. The largest positive balancing load of the P2012 decreases from 8617 N to 3795 N (Figure 17b), and the full-vehicle lift equals nW at every point.

For RQ3, the powerplant loads and the flight envelope of the P2012 and P2006 were compared with non-public flight loads data of the aircraft manufacturer; because these data cannot be disclosed,

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

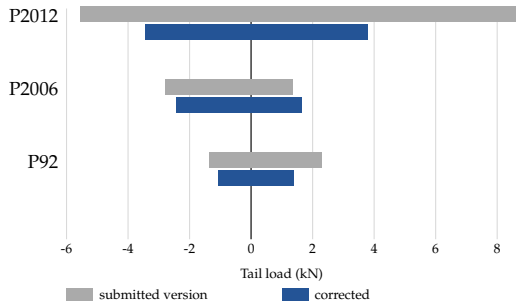
only the band of the absolute difference is reported (Figure 17c). The engine torque, side load, normal load, and limit maneuvering load factor agree within 1 %. The gust, total, and inertial load factors differ by 1 to 15 %, because the reference adopts a different design cruising speed, mass, and altitude. The larger differences have identified causes: a factor of two applied by the reference to the gyroscopic couple of two-blade propellers, a different definition of the pitch arm in the rotational load factor, the static thrust adopted by the code, and the negative-side lift model, which governs the negative stall and maneuvering speeds.

For RQ4, Figure 17a lists five faults encountered during this work. A wrong unit label stopped the run with an explicit error; a lift-curve point inconsistent with the linear branch was recomputed with a warning; a missing engine position left the inertial load reported as not calculated; and a lift coefficient beyond the maximum of the lift curve, which occurs for the P2012 at the most forward CG, is held at that limit and flagged together with the load factor that the airplane can reach. The code removes Output files from the corresponding folder at the start of each run; this avoids silent errors that are usually really hard to spot for the user.

(a) Checks and outcome

Check	Result	Outcome
Definitions of F3116 (load factor 3.2, 4.2.1; balancing load 4.16.1)	Applied in the trim solution	Corrected
Force and moment equilibrium at every trim point (3 aircraft)	Largest residual below 0.01 N	Passed
Defects found by the two checks above	Three defects in the trim model; see (b)	Corrected
Non-public manufacturer data (P2012, P2006)	Agreement bands; see (c)	Reported
Fault: unit label *hp* on a speed in rpm	Run stopped with an explicit error	Stopped
Fault: lift-curve point off the linear branch	Warning; point recomputed	Corrected, warned
Fault: engine position missing	Inertial load reported as not calculated	Flagged
Fault: C_L beyond $C_{L_{max}}$ (forward CG)	C_L held at the limit; reachable n reported	Flagged
Fault: result files of a previous run left in place	Silent; now removed at the start of each run	Corrected

(b) Horizontal-tail balancing load range



Defects corrected: wing-body aerodynamic center scaled twice by the chord; force and moment equations solved separately; full-vehicle lift sign. Vertical-force residual: up to the whole tail load before (8.6 kN, P2012), below 0.01 N after.

(c) Agreement with non-public manufacturer flight loads data: absolute difference band

Powerplant loads	P2012	P2006
Engine torque (MCP)	<1%	<1%
Side load	<1%	<1%
Normal load	<1%	<1%
Gyroscopic couple	<1%	>15% [a]
Gust load factor	1–15% [f]	1–15% [f]
Total load factor	1–15% [f]	1–15% [f]
Inertial load	1–15% [f]	1–15% [f]
Rotational load factor	>15% [b]	n/a [e]
Max. continuous thrust	>15% [c]	n/a

Flight envelope	P2012	P2006
Stall speed V_S	1–15%	1–15%
Maneuvering speed V_A	1–15%	1–15%
Limit load factor n_A	<1%	<1%
Cruise speed V_C	1–15%	1–15%
Dive speed V_D	1–15%	1–15%
Negative stall speed	>15% [d]	1–15%
Negative maneuvering speed	>15% [d]	1–15%
Negative limit load factor	<1%	1–15%

[a] The reference applies a factor of two to the gyroscopic couple of two-blade propellers, which the code does not. [b] The reference measures the pitch arm from the leading edge of the mean aerodynamic chord; the code from the wing-body aerodynamic center. [c] The code uses the static thrust, the largest value at maximum continuous power (conservative). [d] Negative-side wing-body lift model. [e] Term not included in the reference. [f] Different design cruising speed, mass, and reference altitude in the reference.

Figure 17. Verification of the framework: (a) checks performed and their outcome; (b) range of the horizontal-tail balancing load in the submitted and corrected versions; (c) agreement with non-public manufacturer flight loads data, reported as bands of absolute difference.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

4. Discussion

Section 3 presents the results graphically. Appendix B provides the corresponding numerical values in tabular form. These include two applications of the digital certification and substantiation pipeline:

- Construction and plotting of the maneuvering and gust flight envelope.
- Evaluation of the powerplant loads of Section 6 of ASTM F3116/F3116M.

This represents a step toward digitalizing certification processes for aeronautical products, including the substantiation calculations used in technical reports to demonstrate compliance with regulatory agencies. Manufacturing and design organizations may use the methodology presented in this study to manage certification data; whether it reduces the risk of inconsistencies remains to be evaluated in practice.

Sections 3.1 and 3.2 present selected results generated by the MBSE-based digital certification and substantiation framework developed in this study. Section 3.1 reports the results primarily in graphical form, whereas Section 3.2 presents mainly numerical results. The analysis considers three conventional aircraft configurations to cover different use cases, from a very light Level 2 aircraft to a larger Level 4 commuter aircraft.

Section 3.1 includes Figure 13, which presents the maneuvering and gust flight envelopes for three conventional aircraft configurations inspired by the Tecnam P2012, P2006, and P92. Specifically, Subfigures 13(**a**), 13(**b**), and 13(**c**) show the envelopes for the Level 4 twin-engine, Level 2 twin-engine, and Level 2 single-engine configurations, respectively. For the two twin-engine configurations, maneuvering loads govern the flight-envelope boundaries. By contrast, for the very light single-engine configuration, gust loads exceed maneuvering loads, particularly near the design cruising speed. The gust-induced load-factor increment increases as the wing loading decreases: the wing loading is 1430 Pa for the P2012, 952 Pa for the P2006, and 468 Pa for the P92. Similarly, a higher lift-curve slope increases the gust-induced load-factor increment. The Level 2 single-engine configuration illustrates the combined influence of these two factors.

Another important result presented in Section 3.1 is Figure 14, which presents the loads acting on the horizontal tailplane, better known as the balancing loads. ASTM F3116/F3116M defines a horizontal tailplane balancing load as the load necessary to maintain equilibrium in any specified flight condition with no pitching acceleration. The horizontal tailplane must be designed for the balancing loads occurring at any point on the limit maneuvering envelope and in the flap conditions; this study covers the envelope boundary only (Table 1). Subfigures 14(**a**), 14(**b**), and 14(**c**) present the corresponding graphical representation for the Level 4 twin-engine, Level 2 twin-engine, and Level 2 single-engine configurations, respectively. The balancing loads are evaluated on the maneuvering and gust flight envelope boundaries for each aircraft. The three different design architectures present different behaviors of balancing loads when the flight conditions are varied. On the positive side of the maneuvering and gust flight envelope, the highest balancing load occurs between the design maneuvering and design cruising speeds for the P2012 and the P2006, and at the design cruising speed for the P92. On the negative side, the highest magnitude occurs at the design cruising speed for the P2012 and at the design dive speed for the P2006 and the P92.

Figure 15 reproduces the allowable CG positions envelope as a function of the aircraft mass, between the reference operating empty mass and the maximum takeoff mass. The small red dot indicates the configuration selected for executing the analysis on the corresponding design architecture.

Figure 16 completes the series of graphical certification artifacts with the wing-body and full-vehicle lift for the three selected design architectures.

The robustness of the framework can be compared with that of related approaches. Formal requirement languages translate structured natural-language requirements into temporal-logic formulas whose semantics is defined and can be checked by analysis tools [45], and executable SysML models can be verified when a semantically well-defined subset of the language is adopted [46]. Model-driven compliance approaches capture the evidence required by safety standards in UML profiles and SysML

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

traceability models, with automated consistency checks and industrial case studies [47,48]. In construction, automated rule checking of building designs against codes has been developed for decades, and computable code representations are approaching commercial use [49,50]. Compared with these approaches, the framework presented here encodes the substantiation formulas as strings in the SysML model, which gives a weaker formal guarantee: an incorrect formula or coefficient in the model is executed as written. This is mitigated, but not removed, by the verification layer of Section 3.3: checks against the definitions of the standard, equilibrium residuals at every envelope point, and explicit handling of faulty input data. Compared with the authors' earlier MATLAB and C++ toolchains [25,27], the present framework extends an independently developed FAR 23 model rather than one built for the calculations.

5. Conclusions and Future Work

This study presents an application of a largely preexisting SysML model developed by research engineers at the Georgia Institute of Technology. The use case explored herein focuses on the extraction of certification and substantiation data from the SysML model, which are subsequently applied in accordance with the means of compliance accepted by the FAA. The scope of this study is confined to FAR 23, Subpart C (Structure) and adopts ASTM International consensus standard documents as the primary sources for substantiation. Relevant ASTM International documents pertaining to Subpart C have been modeled within the CATIA MSoSA environment. Among these, the most significant is ASTM F3116/F3116M, *Standard Specification for Design Loads and Conditions*, which applies to a broad range of general aviation airplanes certifiable under FAR 23, including airplanes approved for aerobatics.

However, the modeling logic employed and the data extraction phase can be extended to encompass additional subparts of FAR 23, as well as to incorporate a broader range of substantiation methods within a given subpart. For example, in the context of Subpart C, the selected substantiation methods may be compared with those outlined in ASTM F3396/F3396M, *Standard Practice for Aircraft Simplified Loads Criteria*. The stereotype used to encode the substantiation material within the model was sufficient for the paragraphs modeled in this study; its use for other subparts remains to be demonstrated. Future extensions of this framework may support comparative studies across jurisdictions.

Extending the logic underlying this study to other FAR 23 subparts, such as Subpart B, would require stereotype definitions capable of capturing the test cases specified by the applicable consensus standard material. With appropriate modeling effort, the framework could then support the assessment of takeoff, landing, climb, stall, and other performance or handling-qualities requirements, in normal operations and in emergency conditions such as one engine inoperative. The associated external code would need to distinguish data obtained from flight tests from data obtained from simulations and to produce the corresponding test outcomes. Flags, such as test identifiers *passed* or *not passed*, would be collected and compiled in a readable format to summarize the status of the compliance demonstration.

Automating this procedure is expected to improve the consistency of the certification process, the efficiency of information exchange among stakeholders, and the management of certification-related information; these expected benefits, which remain to be evaluated by certification practitioners, would apply both to regulatory authorities, whose primary objective is to ensure the safe operation of the aircraft, and to applicants, who seek to reduce certification time and costs without compromising compliance.

Acknowledgments: The first author gratefully acknowledges Dr. Elena Garcia and Dr. Adam T. Baker from the Aerospace Systems Design Laboratory at the Georgia Institute of Technology for generously providing access to their SysML model of the FAR 23 airworthiness regulations. Their foundational work in model-based systems engineering (MBSE) for aircraft certification has been instrumental in enabling this research. The novel application and extension of their model to digitalize ASTM International substantiation methods represents a new use case that would not have been possible without their collaborative spirit and commitment to advancing the field. The first author also wishes to thank the research team at the Georgia Institute of Technology, whose previous

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

publications on MBSE frameworks for regulatory modeling provided essential methodological guidance for this work. The first author also thanks Dr. Kader Engine Aydin for her valuable expertise and guidance in the development of the research methodology. The first author wishes to express sincere gratitude to Prof. Pierluigi Della Vecchia, Associate Professor of Flight Mechanics at the University of Naples Federico II, Department of Industrial Engineering, for his invaluable guidance, continuous support, and insightful discussions throughout the development of this research. His expertise in aircraft design and certification has been fundamental in shaping the methodology and direction of this work.

Appendix A. Selected Code Excerpts

Appendix A.1 contains the listings referenced in the article related to the Python code used for substantiation calculations. It offers a representative overview of the strategy employed to automatically capture airworthiness and substantiation information. The provided excerpts are not intended as a comprehensive illustration of the Python code developed in the context of this study.

Appendix A.1. The ASTM JSON File

Listing 1 presents the structure of the ASTM JSON file. The excerpt shows how the Python code extracts and organizes the material depicted in Figure 7 directly from the SysML model. The JSON file can include both numbers and strings to encompass all data required to initiate substantiation calculations and can be easily managed using well-established Python packages such as *pandas*. Once loaded into a Python environment, the resulting structured variable is essentially a complex dictionary object containing multiple data types. This variable can then be expanded and updated as calculations progress, and data can be conveniently exported to almost any file format, such as additional JSON files or simple text files. The structure and features of the code have been refined to improve stability and error handling, including several graceful degradation mechanisms. This approach links substantiation tasks to certification artifacts; this represents the main novelty of the present work. Furthermore, it can follow changes in airworthiness and substantiation rules, in accordance with the encoding and exporting strategies established in this study.

Listing 1. The ASTM JSON file structures after parsing operations. The substantiation requirement shown in this listing is the same as the one shown in Figure 7.

```
"4.5 Limit Maneuvering Load Factors": {
  "5.1.1": {
    "TestExpression1": {
      "Value": "n=2.1+(24000/(W+10000))",
      "Unit": "String"
    },
    "TestExpression2": {
      "Value": "n<=3.8",
      "Unit": "String"
    },
    "TestValue1": {
      "Value": 2.1,
      "Unit": "Non dimensional"
    },
    "TestValue2": {
      "Value": 24000.0,
      "Unit": "Non dimensional"
    },
    "TestValue3": {
      "Value": 10000.0,
      "Unit": "Non dimensional"
    },
    "TestValue4": {
      "Value": 3.8,
      "Unit": "Non dimensional"
    }
  },
```

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

Appendix A.2. Python Code Excerpts

Listing 2 and its algorithmic representation show how the code searches for data in the resulting JSON file.

Algorithm A1 Retrieving the limit maneuvering load factor expression.

Input: D, the ASTM data loaded from `ASTM.json` (nested dictionary)
Input: P, the key path of ASTM F3116 4.5.1.1 (Listing 2):
$P_1 =$ `"F3116/F3116M"`, $P_2 =$ `"4.5 Limit Maneuvering Load Factors"`,
$P_3 =$ `"5.1.1"`, $P_4 =$ `"TestExpression1"`, $P_5 =$ `"Value"`
Output: e, the requirement as an equation string, e.g. `"n=2.1+(24000/(W+10000))"`

LOG(`"Step 2: Extracting mathematical expression from ASTM data..."`)
try
  $e \leftarrow D[P_1][P_2][P_3][P_4][P_5]$ ▷ nested lookup, one level per key
  LOG(`"Extracted formula: "` $+ e$)
on error KEYERROR k **do** ▷ a key of P is missing
  LOG(error message)
  raise VALUEERROR(`"Required ASTM expression data not found: "` $+ k$)
end try
return e

Listing 2. In this Python code excerpt, the mathematical expression defined in the example illustrated in Figure 7 and Listing 1 is extracted and saved as a string.

```
# ================================================================
# STEP 2: EXTRACT MATHEMATICAL EXPRESSION FROM ASTM DATA
# ================================================================
print("[INFO] Step 2: Extracting mathematical expression from ASTM data...")

# Extract the positive load factor formula with error handling
try:
    testexpression = astmdata["F3116/F3116M"]
    ["4.5 Limit Maneuvering Load Factors"]["5.1.1"]["TestExpression1"]["Value"]
    print(f"[INFO]   Extracted formula: {testexpression}")

except KeyError as e:
    errormsg = f"..."
    print(f"[ERROR] {errormsg}")
    raise ValueError(f"Required ASTM expression data not found: {e}")
```

To ensure stability and safe execution, the code incorporates features such as error handling and mechanisms to prevent silent failures.

Listing 3 and the corresponding pseudocode demonstrate how the Python code processes strings to derive mathematical expressions, splitting them into the left- and right-hand sides of the equation.

Finally, Listing 4 and its associated algorithm show the key step, where the collected information is transformed into an inline (*lambda*) Python function. All strings and symbols are applied within the *sympify*/*lambdify* strategy, available thanks to well-known Python packages, such as *sympy*.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

Algorithm A2 Splitting the equation string into its left- and right-hand sides.

Input: e, the equation string from Algorithm A1
Output: v, the name of the dependent variable, e.g. `"n"`
Output: r, the right-hand side (the formula), e.g. `"2.1+(24000/(W+10000))"`

try
  if `"="` does not occur in e **then**
    raise ValueError(`"Invalid expression format - missing '=': "` + e)
  end if
  $(\ell, r) \leftarrow$ Split(e, `"="`, 1) ▷ split at the first `"="` only
  $v \leftarrow$ Strip(ℓ); $r \leftarrow$ Strip(r) ▷ remove surrounding blanks
  Log(`"Variable: "` + v)
  Log(`"Formula: "` + r)
on error IndexError or AttributeError **do** ▷ e is not a valid string
  Log(error message)
  raise ValueError(error message)
end try
return (v, r)

Listing 3. In this Python code excerpt, the equation shown in Figure 7 and Listing 1 is splitted into left and right hand side.

```
# Parse the mathematical expression to extract the formula
try:
    if '=' not in testexpression:
        raise ValueError(f"Invalid expression format -
                         missing '=': {testexpression}")

    # Split expression to get right-hand side (the actual formula)
    formulaparts = testexpression.split('=', 1)
    variablename = formulaparts[0].strip()
    formularhs   = formulaparts[1].strip()

    print(f"[INFO]   Variable: {variablename}")
    print(f"[INFO]   Formula: {formularhs}")

except (IndexError, AttributeError) as parseerror:
    errormsg = f"..."
    print(f"[ERROR] {errormsg}")
    raise ValueError(errormsg)
```

Algorithm A3 Creation of an executable function for n with *sympify* and *lambdify*.

Input: r, the formula string from Algorithm A2
Output: f, a callable with $f(W) = n$, e.g. $f(8113) = 3.425$

try
  $W \leftarrow$ Symbols(`"W"`) ▷ design maximum takeoff weight, lb
  $S \leftarrow$ Sympify(r) ▷ string → symbolic expression
  $f \leftarrow$ Lambdify(W, S, `"numpy"`) ▷ symbolic expression → function
  Log(`"Mathematical function created successfully"`)
on error SympifyError **do** ▷ r is not a valid expression
  Log(error message)
  raise ValueError(error message)
on error any other exception **do**
  Log(error message)
  raise RuntimeError(error message)
end try
return f

Listing 4. In this Python code excerpt, a new function using the formula illustrated in Figure 7 and Listing 1 is defined using helper functions *sympify* and *lambdify*.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2


```
# Create symbolic mathematical function from the formula
try:
    # Define weight symbol for symbolic mathematics
    weight_symbol = symbols('W')  # W = Design maximum takeoff weight in pounds

    # Convert string formula to symbolic expression
    symbolic_expression = sympify(formula_rhs)

    # Create executable lambda function for calculations
    load_factor_function = lambdify(weight_symbol, symbolic_expression, 'numpy')

    print(f"[INFO]   Mathematical function created successfully")

except SympifyError as sym_error:
    error_msg = f"..."
    print(f"[ERROR] {error_msg}")
    raise ValueError(error_msg)
except Exception as func_error:
    error_msg = f"..."
    print(f"[ERROR] {error_msg}")
    raise RuntimeError(error_msg)
```

Appendix B. Tables of Numerical Data

This appendix presents the numerical results generated by the Python code in tabular form. The comparison with the manufacturer data is summarized in Section 3.3. The examples are provided for illustrative purposes only and should not be interpreted as validated certification data.

Both the certification basis and the fidelity of the aerodynamic model significantly affect the accuracy of the certification calculations.

The results are intended to illustrate the procedure for aircraft certifiable under FAR 23.

Appendix B.1. Maneuvering and Gust Flight Envelope Data

Table A1 presents data related to the three aircraft architectures adopted for illustrative purposes in this study. The names Tecnam P2012, P2006, and P92 are intended to represent corresponding FAR 23 categories, as defined by the certification parameters currently established in the airworthiness regulations. The numerical data provided may differ from those of the actual aircraft and are included solely to support the initiation of the substantiation calculations.

Table A2 presents the numerical data corresponding to the graphs shown in Figure 13, and aligns with the definition of the maneuvering and gust flight envelope as established by the substantiation methods in ASTM F3116/F3116M, *Standard Specification for Design Loads and Conditions*. The data related to smaller, lighter aircraft represent the most complex scenario, in which the final boundaries of the flight envelope exhibit multiple branches due to the predominance of gust limit load factors over those resulting from aircraft maneuvering.

Tables A3 and A4 present the numerical data corresponding to Figure 14 for both the positive and the negative side of the flight envelope shown in Figure 13, which gives the lift generated by the wing-body, the full vehicle, and the horizontal tailplane.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

Table A1. Geometrical analysis performed by the code during the preliminary calculations phase. All three aircraft architectures examined are included. All values refer to the CG position and mass selected for the analysis.

	PARAMETERS	P2012	P2006	P92
Balancing and CG	Vertical CG position Z_{cg}	0.54 m	0.07 m	−0.07 m
	Thrust axis pos. Z_{thrust}	0.57 m	0.41 m	0.10 m
	I_{xx}	12 894.00 kg m^2	2879.88 kg m^2	735.27 kg m^2
	I_{yy}	17 122.00 kg m^2	2484.12 kg m^2	890.38 kg m^2
	I_{zz}	27 796.00 kg m^2	5037.16 kg m^2	1399.61 kg m^2
	CG position x_{CG}	31 % MAC	31 % MAC	33 % MAC
	Mass M_a	3680 kg	1400 kg	600 kg
	Weight W_a	36 088.47 N	13 729.31 N	5883.99 N
	Vert. tail arm l_{vt}	5.19 m	4.00 m	3.18 m
	Horiz. tail arm l_{ht}	5.97 m	4.52 m	4.04 m
Wing Loading	Wing surface S_{wing}	25.24 m^2	14.43 m^2	12.58 m^2
	Wing MAC	1.82 m	1.31 m	1.41 m
	MAC spanwise pos. y_{MAC}	3.30 m	2.62 m	2.23 m
	Wing Loading SI $\frac{W}{S}$	1430.09 Pa	951.53 Pa	467.84 Pa
	Wing Loading EN $\frac{W}{S}$	29.87 lb ft^{-2}	19.87 lb ft^{-2}	9.77 lb ft^{-2}

Table A2. Airspeeds and limit loads associated with the maneuvering and gust flight envelope diagrams for the proposed three aircraft architectures as shown in Figure 13.

POINTS	PARAMETERS	P2012	P2006	P92
Point S	Airspeed V	39.99 m s^{-1}	33.80 m s^{-1}	22.87 m s^{-1}
	Limit load n	1.00	1.00	1.00
Point A	Airspeed V	74.01 m s^{-1}	65.88 m s^{-1}	45.12 m s^{-1}
	Limit load n	3.43	3.80	3.89
Point C	Airspeed V	84.03 m s^{-1}	73.33 m s^{-1}	52.66 m s^{-1}
	Limit load n	3.43	3.80	4.38
Point D$_1$	Airspeed V	–	–	64.62 m s^{-1}
	Limit load n	–	–	3.80
Point D	Airspeed V	117.12 m s^{-1}	102.66 m s^{-1}	73.72 m s^{-1}
	Limit load n	3.43	3.80	3.80
Point S$_{neg}$	Airspeed V	54.02 m s^{-1}	44.07 m s^{-1}	27.64 m s^{-1}
	Limit load n	−1.00	−1.00	−1.00
Point G	Airspeed V	63.23 m s^{-1}	54.33 m s^{-1}	34.07 m s^{-1}
	Limit load n	−1.37	−1.52	−1.52
Point F$_1$	Airspeed V	–	–	39.32 m s^{-1}
	Limit load n	–	–	−1.52
Point F	Airspeed V	84.03 m s^{-1}	73.33 m s^{-1}	52.66 m s^{-1}
	Limit load n	−1.37	−1.52	−2.38
Point E	Airspeed V	117.12 m s^{-1}	102.66 m s^{-1}	73.72 m s^{-1}
	Limit load n	−0.33	−0.71	−1.36

The most significant calculation performed in this step is the evaluation of the horizontal tailplane lift, which serves as the basis for further assessment of the loading conditions acting on the horizontal tailplane and other aircraft components.

The data presented in this appendix are intended to further illustrate, through a practical example, the digital certification and substantiation workflow described in this study. While the results may differ significantly from those of the actual aircraft, they illustrate a typical substantiation calculation procedure for flight and powerplant loads in a conventional aircraft architecture.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

Table A3. Airspeeds and horizontal tailplane lift associated with balancing loads diagrams for the proposed three aircraft architectures as shown in Figure 14. The data in this table are relative to the case of a positive limit load factor. The full-vehicle lift is the aerodynamic force normal to the airplane longitudinal axis (Equation (1)); all lift coefficients are referred to the wing area.

POINTS	PARAMETERS	P2012	P2006	P92
Point S	Airspeed V	39.99 m s^{-1}	33.80 m s^{-1}	22.87 m s^{-1}
	Dyn. pressure q	979.51 Pa	699.66 Pa	320.44 Pa
	Wing-body lift coeff. $C_{L_{\text{wb}}}$	1.42	1.33	1.37
	Full-vehicle lift coeff. $C_{L_{\text{fv}}}$	1.46	1.36	1.46
	Horiz. tail lift coeff. $C_{L_{\text{ht}}}$	0.04	0.04	0.09
	Angle of attack α	13.13 deg	12.52 deg	12.12 deg
	Wing-body lift L_{wb}	35 138.11 N	13 394.77 N	5519.49 N
	Full-vehicle lift L_{fv}	36 088.47 N	13 729.31 N	5883.99 N
	Horiz. tail lift L_{ht}	1091.74 N	428.74 N	357.84 N
Point A	Airspeed V	74.01 m s^{-1}	65.88 m s^{-1}	45.12 m s^{-1}
	Dyn. pressure q	3354.85 Pa	2658.69 Pa	1247.18 Pa
	Wing-body lift coeff. $C_{L_{\text{wb}}}$	1.42	1.33	1.37
	Full-vehicle lift coeff. $C_{L_{\text{fv}}}$	1.46	1.36	1.46
	Horiz. tail lift coeff. $C_{L_{\text{ht}}}$	0.04	0.04	0.09
	Angle of attack α	13.13 deg	12.52 deg	12.12 deg
	Wing-body lift L_{wb}	120 348.53 N	50 900.12 N	21 482.46 N
	Full-vehicle lift L_{fv}	123 603.53 N	52 171.38 N	22 901.12 N
	Horiz. tail lift L_{ht}	3739.23 N	1629.22 N	1392.75 N
Point C	Airspeed V	84.03 m s^{-1}	73.33 m s^{-1}	52.66 m s^{-1}
	Dyn. pressure q	4324.75 Pa	3293.20 Pa	1698.50 Pa
	Wing-body lift coeff. $C_{L_{\text{wb}}}$	1.10	1.07	1.14
	Full-vehicle lift coeff. $C_{L_{\text{fv}}}$	1.13	1.10	1.21
	Horiz. tail lift coeff. $C_{L_{\text{ht}}}$	0.03	0.03	0.07
	Angle of attack α	8.85 deg	9.14 deg	9.23 deg
	Wing-body lift L_{wb}	119 745.56 N	50 655.35 N	24 246.65 N
	Full-vehicle lift L_{fv}	123 603.53 N	52 171.38 N	25 742.87 N
	Horiz. tail lift L_{ht}	3620.98 N	1589.02 N	1404.81 N
Point D$_1$	Airspeed V	–	–	64.62 m s^{-1}
	Dyn. pressure q	–	–	2557.90 Pa
	Wing-body lift coeff. $C_{L_{\text{wb}}}$	–	–	0.67
	Full-vehicle lift coeff. $C_{L_{\text{fv}}}$	–	–	0.70
	Horiz. tail lift coeff. $C_{L_{\text{ht}}}$	–	–	0.03
	Angle of attack α	–	–	4.78 deg
	Wing-body lift L_{wb}	–	–	21 416.05 N
	Full-vehicle lift L_{fv}	–	–	22 359.16 N
	Horiz. tail lift L_{ht}	–	–	835.09 N
Point D	Airspeed V	117.12 m s^{-1}	102.66 m s^{-1}	73.72 m s^{-1}
	Dyn. pressure q	8401.98 Pa	6454.67 Pa	3329.05 Pa
	Wing-body lift coeff. $C_{L_{\text{wb}}}$	0.57	0.55	0.52
	Full-vehicle lift coeff. $C_{L_{\text{fv}}}$	0.58	0.56	0.53
	Horiz. tail lift coeff. $C_{L_{\text{ht}}}$	0.01	0.01	0.01
	Angle of attack α	3.81 deg	3.87 deg	3.27 deg
	Wing-body lift L_{wb}	120 775.85 N	51 088.85 N	21 631.56 N
	Full-vehicle lift L_{fv}	123 603.53 N	52 171.38 N	22 359.16 N
	Horiz. tail lift L_{ht}	2296.39 N	950.81 N	625.19 N

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

Table A4. Airspeeds and horizontal tailplane lift associated with balancing loads diagrams for the proposed three aircraft architectures as shown in Figure 14. The data in this table are relative to the case of a negative limit load factor. The full-vehicle lift is the aerodynamic force normal to the airplane longitudinal axis (Equation (1)); all lift coefficients are referred to the wing area.

POINTS	PARAMETERS	P2012	P2006	P92
Point S_{neg}	Airspeed V	54.02 m s^{-1}	44.07 m s^{-1}	27.64 m s^{-1}
	Dyn. pressure q	1787.61 Pa	1189.42 Pa	467.84 Pa
	Wing-body lift coeff. $C_{L_{wb}}$	−0.75	−0.73	−0.95
	Full-vehicle lift coeff. $C_{L_{fv}}$	−0.80	−0.80	−1.00
	Horiz. tail lift coeff. $C_{L_{ht}}$	−0.05	−0.07	−0.05
	Angle of attack α	−12.11 deg	−11.69 deg	−12.77 deg
	Wing-body lift L_{wb}	−33 996.78 N	−12 547.33 N	−5602.55 N
	Full-vehicle lift L_{fv}	−36 088.47 N	−13 729.31 N	−5883.99 N
	Horiz. tail lift L_{ht}	−2201.27 N	−1269.72 N	−295.58 N
Point G	Airspeed V	63.23 m s^{-1}	54.33 m s^{-1}	34.07 m s^{-1}
	Dyn. pressure q	2449.04 Pa	1807.91 Pa	711.11 Pa
	Wing-body lift coeff. $C_{L_{wb}}$	−0.75	−0.73	−0.95
	Full-vehicle lift coeff. $C_{L_{fv}}$	−0.80	−0.80	−1.00
	Horiz. tail lift coeff. $C_{L_{ht}}$	−0.05	−0.07	−0.05
	Angle of attack α	−12.11 deg	−11.69 deg	−12.77 deg
	Wing-body lift L_{wb}	−46 575.78 N	−19 071.94 N	−8515.87 N
	Full-vehicle lift L_{fv}	−49 441.41 N	−20 868.55 N	−8943.66 N
	Horiz. tail lift L_{ht}	−3015.75 N	−1929.98 N	−449.28 N
Point F_1	Airspeed V	–	–	39.32 m s^{-1}
	Dyn. pressure q	–	–	946.89 Pa
	Wing-body lift coeff. $C_{L_{wb}}$	–	–	−0.70
	Full-vehicle lift coeff. $C_{L_{fv}}$	–	–	−0.75
	Horiz. tail lift coeff. $C_{L_{ht}}$	–	–	−0.05
	Angle of attack α	–	–	−9.14 deg
	Wing-body lift L_{wb}	–	–	−8378.47 N
	Full-vehicle lift L_{fv}	–	–	−8943.66 N
	Horiz. tail lift L_{ht}	–	–	−536.92 N
Point F	Airspeed V	84.03 m s^{-1}	73.33 m s^{-1}	52.66 m s^{-1}
	Dyn. pressure q	4324.75 Pa	3293.20 Pa	1698.50 Pa
	Wing-body lift coeff. $C_{L_{wb}}$	−0.42	−0.39	−0.61
	Full-vehicle lift coeff. $C_{L_{fv}}$	−0.45	−0.44	−0.65
	Horiz. tail lift coeff. $C_{L_{ht}}$	−0.03	−0.05	−0.04
	Angle of attack α	−6.45 deg	−7.51 deg	−8.11 deg
	Wing-body lift L_{wb}	−45 689.12 N	−18 549.01 N	−13 007.23 N
	Full-vehicle lift L_{fv}	−49 441.41 N	−20 868.55 N	−13 974.89 N
	Horiz. tail lift L_{ht}	−3435.23 N	−2273.25 N	−905.55 N
Point E	Airspeed V	117.12 m s^{-1}	102.66 m s^{-1}	73.72 m s^{-1}
	Dyn. pressure q	8401.98 Pa	6454.67 Pa	3329.05 Pa
	Wing-body lift coeff. $C_{L_{wb}}$	−0.04	−0.08	−0.16
	Full-vehicle lift coeff. $C_{L_{fv}}$	−0.06	−0.10	−0.19
	Horiz. tail lift coeff. $C_{L_{ht}}$	−0.02	−0.03	−0.03
	Angle of attack α	−2.41 deg	−3.86 deg	−3.67 deg
	Wing-body lift L_{wb}	−8432.33 N	−7183.12 N	−6858.62 N
	Full-vehicle lift L_{fv}	−11 969.60 N	−9771.65 N	−8017.22 N
	Horiz. tail lift L_{ht}	−3179.11 N	−2437.13 N	−1056.49 N

Appendix B.2. Powerplant Loads

Table A5 presents the numerical results from the powerplant loads calculations, performed in accordance with the methods outlined in ASTM F3116/F3116M, *Standard Specification for Design Loads and Conditions*. The substantiation provisions require the use of data derived from the construction of the aircraft's flight envelope. The mean and limit torques acting on the powerplant aerostructures have been computed and reported accordingly.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

Table A6 completes the sequence of powerplant load calculations. The data include inertial, side, gyroscopic, and normal loads acting on the powerplant group and associated aerostructures. As in previous steps, the calculations rely on information extracted directly from the model, for example, the yaw and pitch rates, r and q; the normal load factor applied to the engine weight, W_{engine}; and the formulas used for side load calculations. These computations also incorporate numerical data obtained from the flight envelope analysis.

Table A5. Powerplant loads: mean and limit engine torque.

	PARAMETERS	P2012	P2006	P92
Engine Torque	Max Cont. Power MCP_{hp}	375.00 hp	92.50 hp	92.50 hp
	Max Cont. Power MCP_{W}	279.64 kW	68.98 kW	68.98 kW
	Prop. rot. speed RPM_{MCP}	2575 rpm	2265 rpm	2263 rpm
	Prop. rot. speed Ω_{MCP}	269.65 rad s^{-1}	237.19 rad s^{-1}	236.98 rad s^{-1}
	Engine Type	Four stroke	Four stroke	Four stroke
	Number of cylinders	6	4	4
	Mult. factor	1.33	2.00	2.00
	Mean Eng. Torque	1037.03 N m	290.81 N m	291.07 N m
	Limit Eng. Torque	1379.24 N m	581.62 N m	582.13 N m

Table A6. Powerplant loads: inertial, side, gyroscopic, and aerodynamic loads.

	PARAMETERS	P2012	P2006	P92
Inertial Loads	Select. alt. h_{ft}	20 000 ft	15 000 ft	15 000 ft
	Select. alt. h_{m}	6096 m	4572 m	4572 m
	Gust speed U_{de}	50 ft s^{-1}	50 ft s^{-1}	50 ft s^{-1}
	Governing mass	OEM (2387.00 kg)	OEM (847.50 kg)	OEM (455.80 kg)
	Mass ratio μ_g	33.76	25.15	11.85
	Gust load fact. n_{gust}	4.03	5.01	5.73
	Pitch inertia I_{yy}	11 106.00 kg m^2	1503.80 kg m^2	676.30 kg m^2
	Pitch arm $\lvert x_{\text{ac,wb}} - x_{\text{cg}} \rvert$	0.11 m	0.08 m	0.11 m
	Engine lever $\lvert x_{\text{cg}} - x_{\text{pp}} \rvert$	1.03 m	0.84 m	1.37 m
	Pitch accel. $\dot{q}$	0.71 rad s^{-2}	1.76 rad s^{-2}	3.56 rad s^{-2}
	Rot. load fact. n_{rot}	0.07	0.15	0.50
	Tot. load fact. n_{tot}	4.11	5.16	6.22
	Inertial load	14 427.05 N	4709.70 N	5675.52 N
Side Loads	Engine mass M_{engine}	358.20 kg	93.00 kg	93.00 kg
	Engine weight W_{engine}	3512.74 N	912.02 N	912.02 N
	n_A	3.43	3.80	3.80
	$\frac{1}{3} \cdot n_A \cdot W_{\text{engine}}$	4010.40 N	1155.22 N	1155.22 N
	$1.33 \cdot W_{\text{engine}}$	4671.95 N	1212.98 N	1212.98 N
	Lim. side load L_{side}	4671.95 N	1212.98 N	1212.98 N
Gyroscopic	Prop. pol. mom. I_{polar}	1.76 kg m^2	0.37 kg m^2	0.37 kg m^2
	Yaw rate r	2.50 rad s^{-1}	2.50 rad s^{-1}	2.50 rad s^{-1}
	Pitch rate q	1.00 rad s^{-1}	1.00 rad s^{-1}	1.00 rad s^{-1}
	Yaw Gyro. couple T_{yaw}	1188.50 N m	219.40 N m	219.21 N m
	Pitch Gyro. couple T_{pitch}	475.40 N m	87.76 N m	87.68 N m
Normal	Normal load $L_{\text{normal}} = 2.5 \cdot W_{\text{engine}}$	8781.86 N	2280.05 N	2280.05 N
	Average prop. eff. η	0.90	0.90	0.90
	Prop. diameter D	2.08 m	1.78 m	1.78 m
	Prop. disc area A	3.41 m^2	2.49 m^2	2.49 m^2
	Max. cont. (static) thrust MCT	8086.68 N	2864.16 N	2864.16 N

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

The powerplant loads and the flight envelope of the P2012 and P2006 were also compared with non-public flight loads data of the aircraft manufacturer. Because these data cannot be disclosed, the comparison is reported only as agreement bands in Section 3.3. These comparisons are strongly influenced by the quality of the aerodynamic data, particularly in relation to drag, by the assumptions made regarding the certification basis, and by the maximum takeoff mass, which may differ from that of the actual aircraft.

The wing-body aerodynamic model affects the design speeds: the drag model governs the design dive speed, and the negative-side lift model governs the negative stall and maneuvering speeds, which are also difficult to characterize at low speed.

Author Contributions: Conceptualization, Claudio Mirabella; methodology, Claudio Mirabella; software, Claudio Mirabella; validation, Claudio Mirabella and Adam T. Baker; formal analysis, Claudio Mirabella, Adam T. Baker, and Elena Garcia; investigation, Claudio Mirabella; resources, Elena Garcia and Pierluigi Della Vecchia; data curation, Claudio Mirabella and Pierluigi Della Vecchia; writing—original draft preparation, Claudio Mirabella; writing—review and editing, Claudio Mirabella, Elena Garcia and Adam T. Baker; visualization, Claudio Mirabella; supervision, Elena Garcia and Pierluigi Della Vecchia; project administration, Elena Garcia. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. No external financial support was received for the preparation or publication of this manuscript.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

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

Abbreviations

The following abbreviations are used in this manuscript:

w	Side - displacement
M	Mach number
I_{zz}	Principal moment of Inertia along the z - body axes
g_0	Standard gravitational acceleration
W	Aircraft weight
q	Dynamic pressure
S_w	Wing reference Surface
b_w	Wingspan
α	Angle of attack referred to the x - body axes
C_L	Aircraft lift coefficient
ISA	International Standard Atmosphere
h	Above sea - level height
CS	Certification Specifications
CS-23	Certification Specifications for Normal-Category Aeroplanes
AC	Advisory Circular
BDD	Block Definition Diagram
PD	Profile Diagram
CSP	Consensus Standards Paragraph
CG	Center-of-Gravity
MAC	Mean Aerodynamic Chord
MCP	Maximum Continuous Power
MCT	Maximum Continuous Thrust
MF	Multiplying Factor
CFR	Code of Federal Regulations
XMI	Extensible Markup Language Metadata Interchange

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

CS-VLA	Certification Specifications for Very Light Aircraft
AMC	Acceptable Means of Compliance
MBSE	Model-Based Systems Engineering
SysML	Systems Modeling Language
UML	Unified Modeling Language
ASTM	American Society for Testing and Materials
TC	Type Certificate
MTOW/M	Maximum Takeoff Weight/Mass
INCOSE	International Council on Systems Engineering
JSON	JavaScript Object Notation
XML	Extensible Markup Language
FAR	Federal Aviation Regulations
FAA	Federal Aviation Administration
EASA	European Union Aviation Safety Agency
MSoSA	Magic Systems of Systems Architect
SI	Système international
MoC	Means of Compliance

References

1. National Research Council. *Improving the continued airworthiness of civil aircraft: A strategy for the FAA's aircraft certification service*; National Academies Press, 1998. https://doi.org/10.17226/6265.
2. De Florio, F. *Airworthiness: An introduction to aircraft certification and operations*; Butterworth-Heinemann, 2016. Available online: https://www.sciencedirect.com/book/9780081008881/airworthiness.
3. Joslin, R.E. Certification of Aircraft and Aircraft Systems. In *Handbook of Human Factors in Air Transportation Systems*; CRC Press, 2017; pp. 147–193. https://doi.org/10.1201/9781315116549-8.
4. Nair, S.; de la Vara, J.L.; Sabetzadeh, M.; Briand, L. An extended systematic literature review on provision of evidence for safety certification. *Information and Software Technology* **2014**, *56*, 689–717. https://doi.org/10.1016/j.infsof.2014.03.001.
5. Correia, V. Certification Issues Revealed by the 737 Max Crisis: A Comparative Approach from a European Perspective. *Air and Space Law* **2020**, *45*, 309–340. https://doi.org/10.54648/aila2020041.
6. Downer, J. Trust and technology: the social foundations of aviation regulation. *The British Journal of Sociology* **2010**, *61*, 83–106. https://doi.org/10.1111/j.1468-4446.2009.01303.x.
7. Joosen, R.; Haverland, M.; de Bruijn, E. Shaping EU agencies' rulemaking: Interest groups, national regulatory agencies and the European Union Aviation Safety Agency. *Comparative European Politics* **2022**, *20*, 411–442. https://doi.org/10.1057/s41295-021-00268-z.
8. Cartile, A.; Marsden, C.; Liscouët-Hanke, S. Digital Transformation in Aircraft Design and Certification: Developing Requirements for Modeling Regulatory Documentation. *Aerospace* **2025**, *12*, 724. https://doi.org/10.3390/aerospace12080724.
9. Federal Aviation Administration. The FAA and Industry Guide to Product Certification (CPI Guide), 3rd Edition. PDF document, 2017. Available online: https://www.faa.gov/sites/faa.gov/files/aircraft/air_cert/design_approvals/transport/CPI_guide.pdf.
10. Federal Aviation Administration. AC 23.2010-1 - FAA Accepted Means of Compliance Process for 14 CFR Part 23. Advisory Circular, 2017. Available online: https://www.faa.gov/regulations_policies/advisory_circulars/index.cfm/go/document.information/documentID/1030946.
11. Federal Aviation Administration. Title 14 CFR Part 23: Airworthiness Standards: Normal Category Airplanes. Available online: https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-23.
12. Federal Aviation Administration. Revision of Airworthiness Standards for Normal, Utility, Acrobatic, and Commuter Category Airplanes; Final Rule. Federal Register, Vol. 81, No. 251, pp. 96572 ff., Amdt. 23-64, 2016. Available online: https://www.govinfo.gov/content/pkg/FR-2016-12-30/pdf/2016-30246.pdf.
13. European Union Aviation Safety Agency. Easy Access Rules for Initial Airworthiness and Environmental Protection (Regulation (EU) No 748/2012). EASA eRules, Revision from July 2024, 2024. Available online: https://www.easa.europa.eu/en/document-library/easy-access-rules/easy-access-rules-initial-airworthiness-and-environmental.
14. Wolny, S.; Mazak, A.; Carpella, C.; Geist, V.; Wimmer, M. Thirteen years of SysML: a systematic mapping study. *Software and Systems Modeling* **2020**, *19*, 111–169. https://doi.org/10.1007/s10270-019-00735-y.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

15. Hill, T.R.; Carnevale, A.E.; Morris-Eckart, A.; Sundaram, V.; Farhaj, L.Z. NASA's Use of MBSE and SysML Modeling to Architect the Future of Human Exploration. *INCOSE International Symposium* **2024**, *34*, 527–543. https://doi.org/10.1002/iis2.13161.
16. Gregory, J.; Berthoud, L.; Tryfonas, T.; Rossignol, A.; Faure, L. The long and winding road: MBSE adoption for functional avionics of spacecraft. *Journal of Systems and Software* **2020**, *160*, 110453. https://doi.org/10.1016/j.jss.2019.110453.
17. Corrado, J.K. Model-based systems engineering adoption in the U.S. Nuclear industry. *Nuclear Engineering and Design* **2025**, *432*, 113752. https://doi.org/10.1016/j.nucengdes.2024.113752.
18. Assef, P.; Geiger, J. Adoption of Model-Based Systems Engineering in Traditional DoD Systems. *Defense Acquisition Research Journal* **2023**, *30*, 46–73. https://doi.org/10.22594/dau.22-892.30.01.
19. Madni, A.M.; Purohit, S. Economic Analysis of Model-Based Systems Engineering. *Systems* **2019**, *7*, 12. https://doi.org/10.3390/systems7010012.
20. Bleu-Laine, M.H.; Bendarkar, M.V.; Xie, J.; Briceno, S.I.; Mavris, D.N. A Model-Based System Engineering Approach to Normal Category Airplane Airworthiness Certification. In Proceedings of the AIAA Aviation 2019 Forum, 2019. https://doi.org/10.2514/6.2019-3344.
21. Bendarkar, M.V.; Xie, J.; Briceno, S.I.; Harrison, E.D.; Mavris, D. A Model-Based Aircraft Certification Framework for Normal Category Airplanes. In Proceedings of the AIAA AVIATION 2020 FORUM, 2020. https://doi.org/10.2514/6.2020-3096.
22. Fazal, B.; Glinski, S.; Harrison, E.; Fields, T.M.; Bendarkar, M.V.; Garcia, E.; Mavris, D.N. An MBSE Framework for Regulatory Modeling of Transport Category Airplanes. In Proceedings of the AIAA AVIATION 2022 Forum, 2022. https://doi.org/10.2514/6.2022-3256.
23. Glinski, S.; Fazal, B.; Harrison, E.D.; Bendarkar, M.V.; Fields, T.; Garcia, E.; Mavris, D.N. An MBSE Framework to Identify Regulatory Gaps for Electrified Transport Aircraft. In Proceedings of the 2022 IEEE Transportation Electrification Conference & Expo (ITEC), 2022, pp. 772–777. https://doi.org/10.1109/itec53557.2022.9813806.
24. European Union Aviation Safety Agency. Certification Specifications for Normal-Category Aeroplanes (CS-23), Amendment 6. Available online: https://www.easa.europa.eu/en/document-library/certification-specifications/group/cs-23-normal-utility-aerobatic-and-commuter-aeroplanes.
25. Mirabella, C.; Tuccillo, M.; Della Vecchia, P. A Model-Based Systems Engineering Digital Certification Framework for General Aviation Aircraft. *Aerotecnica Missili & Spazio* **2025**, *104*, 371–386. https://doi.org/10.1007/s42496-024-00241-5.
26. Della Vecchia, P.; Mirabella, C.; Tuccillo, M.; Riboldi, C.E.; Fioriti, M.; Shahini, F.; Roncolini, F. A SysML-based framework towards EASA CS-23 digitalization: An MBSE approach. *Transportation Research Interdisciplinary Perspectives* **2026**, *36*, 101810. https://doi.org/10.1016/j.trip.2025.101810.
27. Mirabella, C.; Tuccillo, M.; Della Vecchia, P. Evaluating MBSE Approaches and Tools for Aircraft Design and Certification: A Comparative Perspective. *Systems* **2026**, *14*, 482. https://doi.org/10.3390/systems14050482.
28. Griffin, J.A.; Takahashi, T.T. A Break from Tradition: The "New" 14 C.F.R. Part 23 and Consensus Standards. In Proceedings of the AIAA SCITECH 2023 Forum, 2023. https://doi.org/10.2514/6.2023-0116.
29. Cardoso, S.H.S.B.; Oliveira, M.V.R.d.; Godoy, J.R.S. eVTOL Certification in FAA and EASA Performance-Based Regulation Environments: A Bird Strike Study-Case. *Journal of Aerospace Technology and Management* **2022**, *14*. https://doi.org/10.1590/jatm.v14.1271.
30. European Commission. Commission Regulation (EU) No 748/2012 of 3 August 2012 laying down implementing rules for the airworthiness and environmental certification of aircraft and related products, parts and appliances, as well as for the certification of design and production organisations (recast). Official Journal of the European Union, L 224, 21.8.2012, pp. 1–85, 2012. Available online: https://eur-lex.europa.eu/eli/reg/2012/748/oj.
31. ASTM International. Standard Specification for Design Loads and Conditions, F3116/F3116M-24, 2024. https://doi.org/10.1520/F3116_F3116M-24.
32. ASTM International. Standard Practice for Aircraft Simplified Loads Criteria, F3396/F3396M-23a, 2023. https://doi.org/10.1520/F3396_F3396M-23A.
33. ASTM International. Standard Practice for Aircraft Water Loads, F3331-18, 2018. https://doi.org/10.1520/F3331-18.
34. ASTM International. Standard Specification for Design and Performance of a Light Sport Airplane, F2245-23, 2023. https://doi.org/10.1520/F2245-23.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 17 November 2026 https://doi.org/10.20944/preprints202609.2735.v2

35. European Union Aviation Safety Agency. ED Decision 2019/018/R: Amending the AMC and GM to Annex I (Part 21) to Commission Regulation (EU) No 748/2012, AMC and GM to Part 21, Issue 2, Amendment 9. Executive Director Decision, 2019. Available online: https://www.easa.europa.eu/en/document-library/agency-decisions/ed-decision-2019018r.
36. European Union Aviation Safety Agency. ED Decision 2023/013/R: Acceptable Means of Compliance and Guidance Material to Annex Ib (Part 21 Light) to Regulation (EU) No 748/2012 and to the Annexes to Regulation (EU) No 1321/2014. Executive Director Decision, 2023. Available online: https://www.easa.europa.eu/en/document-library/agency-decisions/ed-decision-2023013r.
37. European Union Aviation Safety Agency. Appendix A to AMC 21.A.15(b): Means of Compliance Codes. In Easy Access Rules for Initial Airworthiness and Environmental Protection (Regulation (EU) No 748/2012), Annex I (Part 21), Revision from July 2024, p. 116; ED Decision 2019/018/R, 2024. Available online: https://www.easa.europa.eu/en/document-library/easy-access-rules/easy-access-rules-initial-airworthiness-and-environmental.
38. European Union Aviation Safety Agency. Appendix A to AMC1 21L.A.24(b): Means-of-Compliance Codes. In Easy Access Rules for Initial Airworthiness and Environmental Protection (Regulation (EU) No 748/2012), Annex Ib (Part 21 Light), Revision from July 2024, p. 727; ED Decision 2023/013/R, 2024. Available online: https://www.easa.europa.eu/en/document-library/easy-access-rules/easy-access-rules-initial-airworthiness-and-environmental.
39. European Union Aviation Safety Agency. Appendix A to AMC1 21L.A.108(a): Compliance Activities for Declaring Compliance of a Major Change (Means-of-Compliance Codes). In Easy Access Rules for Initial Airworthiness and Environmental Protection (Regulation (EU) No 748/2012), Annex Ib (Part 21 Light), Revision from July 2024, p. 814; ED Decision 2023/013/R, 2024. Available online: https://www.easa.europa.eu/en/document-library/easy-access-rules/easy-access-rules-initial-airworthiness-and-environmental.
40. Federal Aviation Administration. Airworthiness Compliance Checklists Used to Substantiate Major Alterations for Small Airplanes. Initiated by ACE-100, 2005.
41. Federal Aviation Administration. Standardized Procedures for Obtaining Approval of Data Used in the Performance of Major Repairs and Major Alterations. Initiated by AFS-300, 2016.
42. Pratt, K.G.; Walker, W.G. A Revised Gust-Load Formula and a Re-evaluation of V-G Data Taken on Civil Transport Airplanes from 1933 to 1950. Technical Report NACA Report 1206, National Advisory Committee for Aeronautics, 1954.
43. Wright, J.R.; Cooper, J.E. *Introduction to Aircraft Aeroelasticity and Loads*; Wiley, 2014. https://doi.org/10.1002/9781118700440.
44. Lomax, T.L. *Structural Loads Analysis for Commercial Transport Aircraft: Theory and Practice*; American Institute of Aeronautics and Astronautics, 1996. https://doi.org/10.2514/4.862465.
45. Giannakopoulou, D.; Pressburger, T.; Mavridou, A.; Schumann, J. Automated formalization of structured natural language requirements. *Information and Software Technology* **2021**, *137*, 106590. https://doi.org/10.1016/j.infsof.2021.106590.
46. Horváth, B.; Molnár, V.; Graics, B.; Hajdu, Á.; Ráth, I.; Horváth, Á.; Karban, R.; Trancho, G.; Micskei, Z. Pragmatic verification and validation of industrial executable SysML models. *Systems Engineering* **2023**, *26*, 693–714. https://doi.org/10.1002/sys.21679.
47. Panesar-Walawege, R.K.; Sabetzadeh, M.; Briand, L. Supporting the verification of compliance to safety standards via model-driven engineering: Approach, tool-support and empirical validation. *Information and Software Technology* **2013**, *55*, 836–864. https://doi.org/10.1016/j.infsof.2012.11.009.
48. Nejati, S.; Sabetzadeh, M.; Falessi, D.; Briand, L.; Coq, T. A SysML-based approach to traceability management and design slicing in support of safety certification: Framework, tool support, and case studies. *Information and Software Technology* **2012**, *54*, 569–590. https://doi.org/10.1016/j.infsof.2012.01.005.
49. Eastman, C.; Lee, J.m.; Jeong, Y.s.; Lee, J.k. Automatic rule-based checking of building designs. *Automation in Construction* **2009**, *18*, 1011–1033. https://doi.org/10.1016/j.autcon.2009.07.002.
50. Amor, R.; Dimyadi, J. The promise of automated compliance checking. *Developments in the Built Environment* **2021**, *5*, 100039. https://doi.org/10.1016/j.dibe.2020.100039.

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.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.