Author version

Published as:

Swaminathan, R., Sarojini, D., & Hwang, J. T. (2023). Integrating MBSE and MDO through an extended requirements-functional-logical-physical (RFLP) framework. In AIAA AVIATION 2023 Forum (Paper No. AIAA 2023-3908). https://doi.org/10.2514/6.2023-3908

BibTeX for this reference:

https://lsdolab.github.io/lsdo-publications/publication/swaminathan2023integrating/

```
@inproceedings{swaminathan2023integrating,
  author = {Rajashekar Swaminathan and Darshan Sarojini and John T. Hwang},
  title = {Integrating MBSE and MDO through an Extended
    Requirements-Functional-Logical-Physical (RFLP) Framework},
  booktitle = {AIAA AVIATION 2023 Forum},
  year = {2023},
  doi = {10.2514/6.2023-3908},
  url = {https://doi.org/10.2514/6.2023-3908}
}
```

Integrating MBSE and MDO through an Extended Requirements-Functional-Logical-Physical (RFLP) Framework

Conference Paper · June 2023

DOI: 10.2514/6.2023-3908

3 authors, including:

Darshan Sarojini
University of California, San Diego
45 PUBLICATIONS **206** CITATIONS

SEE PROFILE

John T. Hwang
University of California, San Diego
92 PUBLICATIONS **1,975** CITATIONS

SEE PROFILE

Integrating MBSE and MDO through an Extended Requirements-Functional-Logical-Physical (RFLP) Framework

Rajashekar Swaminathan* and Darshan Sarojini† and John T. Hwang‡
University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093

The design of mechanical systems is a complex process that is often associated with significant time and budget overruns. Model-Based Systems Engineering (MBSE) has emerged as a way to mitigate these issues. However, MBSE alone lacks the ability to simulate, analyze and design mechanical systems. Multidisciplinary Design Optimization (MDO) treats the parameters of the mechanical system as design variables and uses numerical optimization methods to find the parameters that optimize performance metrics while meeting constraints. However, it lacks the systematic traceability that MBSE offers. This paper proposes a methodology that builds on the Requirements Engineering, Functional Design, Logical Design, and Physical Design (RFLP) approach to integrate MBSE and MDO and overcome each other's limitations. The recently proposed Process RFLP is used to aid in the development, enumeration, and selection of design activities involving simulation and MDO. Guidelines for the implementation of RFLP are formulated and presented. The approach is demonstrated through a test case involving the design of a single-aisle transport category aircraft. Compliance with aircraft requirements is verified, and a change in requirements case is presented. The long-term goal is to automate the methodology to enable MBSE to drive the formulation and execution of MDO.

Acronyms

CAD Computer-Aided Design.
CSDL Computational System Design Language.
FAA Federal Aviation Administration.
FLOPS Flight Optimization System.
MBSE Model Based Systems Engineering.
MDO Multidisciplinary Design Optimization.
NASA National Aeronautics and Space Administration.
RFLP Requirements Engineering, Functional Design, Logical Design, Physical Design.
SE Systems Engineering.
SUAVE Stanford University's Aircraft Vehicle Environment.

I. Introduction

In recent times, the design process of mechanical systems, such as aircraft, automobiles, and robots, has become increasingly complex due to the involvement of multiple engineering disciplines, conflicting objectives, and the management of a substantial amount of design data, resulting in higher market entry costs and longer development times [1]. Mechanical systems refer to 'machines that manage power to perform tasks involving forces and motion'. Prior studies have suggested that incorporating later-stage activities into the product design cycle at an early stage can be beneficial as changes made during later stages can be both costly and time-consuming [2, 3]. However, in practice, design changes are inevitable due to emerging technologies that increase system complexity and requirements change over time [4, 5]. During product development, designers must consider numerous options and make multiple design decisions to satisfy specifications and accommodate stakeholder needs over time [6].

In particular, the complexity of modern aircraft products has increased significantly due to the interplay of various factors such as overall performance requirements, lowering emissions, cost, and schedule [7]. As a result, the development

*Masters student, Mechanical and Aerospace Engineering Department, AIAA student member
†Postdoctoral scholar, Mechanical and Aerospace Engineering Department, AIAA member
‡Assistant professor, Mechanical and Aerospace Engineering Department, AIAA senior member

of these products has become more challenging, causing difficulties in both product design and manufacturing. To remain competitive, aircraft companies are required to introduce innovative and complex products within shorter timeframes while maintaining high standards for risks and quality [8].

The field of Systems Engineering (SE) has emerged due to the increasing complexity of developing, testing and validating modern mechanical systems [9, 10]. According to the Gesellschaft für Systems Engineering (GfSE), SE is crucial for developing complex mechanical systems [11]. SE relies on technical models that evolve throughout the lifecycle and support design verification and trade studies. However, the current practice of using standalone models for each discipline and sharing information primarily through static documents has limitations.

To address this issue, SE has been transitioning towards model-based approaches, with Model Based Systems Engineering (MBSE) identified as a primary component of the SE vision for 2020 and 2025 by the International Council on Systems Engineering (INCOSE) [12, 13]. INCOSE defines Model Based Systems Engineering (MBSE) as "the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases."

MBSE includes the representation of what the system is required to do, how the system interacts with users, which functions are required to perform its requirements, and how user interactions and functions are implemented in physical parts of the mechanical system. These are represented by a *system model*. The system model is distinct from discipline-specific mathematical models that are a set of equations whose solution describes the predicted behavior (i.e., state) or performance (output/ quantity of interest) of a mechanical system. MBSE emphasizes the importance of all disciplines 'viewing' a consistent system model.

Before proceeding further, it is essential to define terms that will be used throughout the rest of the paper [12, 14]:

- **Process**: A logical sequence of tasks performed to achieve a particular objective. A process defines the 'what' is to be done, without specifying the 'how' each task is to be performed.
- **Method**: Consists of techniques for performing a task, the 'how' of each task.
- **Tool**: An instrument that, when applied to a particular method, can enhance the efficiency of a task. Thus, methods help bridge the gap between process and tools. The purpose of the tool should be to facilitate the accomplishment of the 'how'.
- **Methodology**: Defined as a collection of related processes, methods, and tools.
- **Executable Behavior**: The ability of a system model to be translated into code that can be run on a computer, and which accurately represents the behavior of the system being modeled.

Methodologies are employed to facilitate the design of system models. A MBSE methodology can be characterized as the collection of related processes, methods, and tools used to support the discipline of systems engineering in a "model-based" or "model-driven" context [15]. Designing and developing complex, multi-disciplinary systems requires a comprehensive system model that incorporates customer requirements, system functions, and operating principles of various disciplines. The Requirements Engineering, Functional Design, Logical Design, Physical Design (RFLP) approach, which was initially proposed by Kleiner et al. [16] and is currently implemented by Dassault Systemes, is one of the dominant methodologies used to design MBSE models across various industries today.

The RFLP acronym represents four distinct phases in the MBSE process: **R**equirements Engineering, **F**unctional Analysis, **L**ogical Design, and **P**hysical Design. An illustrative example highlights the benefits of employing MBSE using the RFLP approach. Specifically, the requirement "Aircraft shall be maneuverable" is considered as one type of requirement among many, such as corporate, end-use, financial, or design requirements. In order to satisfy this requirement, multiple functions are identified, including "Aircraft shall generate yawing moment of 1 deg/sec", "Aircraft shall generate pitching motion", and "Aircraft shall generate rolling motion". These functions represent the 'what' that must be performed to meet the requirement. Next, the logical options for the function "Aircraft shall generate yawing motion of 1 deg/sec" are analyzed, and two potential options are identified: 1) Rudder deflection, and 2) Differential Thrust. These logical options provide potential implementation choices for the function, describing the 'how' aspect of the design. If the rudder is selected, a physical design using a Computer-Aided Design (CAD) model would be created to execute the logical model. The schematic representation of this process is depicted on the left side of Figure 1.

The previous example serves as an illustration of the advantages and disadvantages of MBSE. The benefits of MBSE are its capacity for traceability and the ability to monitor assumptions made during the design process. On the other hand, limitations of MBSE include the difficulty of integrating analytical and numerical models into models, and the inability of MBSE to solve a large number of design variables, except through novel techniques [17]. When requirements are formulated at the system level, system design may yield numerous technical variants with different design variables. MBSE methods are typically used to design a limited number of variants, resulting in a subset that meets the requirements.

Due to the above limitations, there has been a search for a methodology that can carry out a multitude of analyses necessary for the design process to be effectively utilized, reviewed, traced, and used to make informed design decisions [18]. To address the limitations in MBSE, Multidisciplinary Design Optimization (MDO) is often employed[19].

An extension of the example provided above illustrates how simulations and MDO can be utilized to integrate and optimize the design process. We again define terms that will be used throughout this paper:

- **Mathematical model**: A mathematical model is a set of equations (that may include differential or integral operators) whose solution describes the predicted behavior (i.e., state) or performance (output/ quantity of interest) of a mechanical system.
- **Numerical model**: A numerical model is a finite set of algebraic equations that results from the discretization of the differential or integral operators in a mathematical model. Such discretization uses the geometric information stored in the Product Physical.
- **Computational model**: A computational model is a set of explicit computations that results from applying numerical solvers (e.g., nonlinear solvers, linear solvers, eigenvalue solvers) to numerical models.
- **Computer program:** A computer program is an implementation of a computational model in a computer language

In this case, the objective would be to obtain a minimum drag rudder for the aircraft such that the side slip angle is greater than one deg/sec. The simulation here involves computational models that describe the aerodynamic behavior, the loads, and the structural margins of the internal structure when critical loads are applied [20]. The MDO engineer picks computer programs to perform the simulations and uses an optimizer to design the rudder. The MDO engineer 'passes the design along' to another engineer who would generate the optimized CAD model of the rudder, stored in the 'physical' of the system model. However, this approach has certain limitations. For instance, the MDO engineer does not know the original requirements and functions. MDO also does not know that another option, differential thrust, was available to explore. Further, when coding the MDO model, the origin of the one deg/sec constraint may not be readily apparent. This may be added as a comment in the code, written down in a spreadsheet, or worse, not documented and is just known to the engineer. It is easy to see that such practices can lead to errors and delays, especially in large organizations spanning multiple geographical locations.

As with MBSE, MDO has both benefits and limitations. Among its advantages, MDO can efficiently handle a large number of design variables and identify variables with the most favorable objective values. However, one of its limitations as discussed above is that it cannot track requirements and assumptions, resulting in a lack of traceability. Interestingly, these benefits and limitations are the opposite of those of MBSE. Therefore, it is evident that MBSE and MDO can complement each other, allowing for the mitigation of each other's limitations. A visual representation of the aforementioned discussion is depicted in Figure 1.

The combined utilization of MDO with MBSE provides a comprehensive and systematic approach to system development. In recent years, this integration has gained significant attention from organizations seeking to enhance their engineering processes and improve product development outcomes [21]. MBSE offers a holistic framework for capturing system requirements, defining system architecture, and managing system models. MDO, on the other hand, enables efficient analysis and optimization of system performance across multiple disciplines. Through this integration, design decisions are based on a comprehensive understanding of system behavior and performance, leading to informed and optimized designs.

Scholarly literature on the integration of MBSE and MDO has progressed from early concerns about accurately representing the MBSE system model in the MDO problem, to the development of methodologies for the integration [22, 23]. The current focus in a majority of the literature has been to highlight the key features in integrating MBSE and MDO. One of these features is the ability to perform system-level optimization. This was highlighted by Nagel et al. [24], who proposed a framework for the integration of MBSE and MDO in the context of the design of aircraft systems. The framework involved the use of MBSE to develop a system model, which was then used as the basis for MDO analysis. Similarly, Wang et al. [25] presented a similar method for the integration of MBSE and MDO in the design of automotive powertrains.

Another area of interest is the development of processes that are specifically tailored to using design optimization algorithms along with MBSE models. Chang et al. [26] proposed an approach that combines a particle swarm optimization algorithm with an MBSE model to optimize the design of a thermal control system for a spacecraft. The authors demonstrated the effectiveness of the approach by comparing the results with those obtained using solely optimization without MBSE. Researchers are also investigating the use of existing MDO techniques in the context of MBSE models. Khodaparast et al. [27] used a response surface methodology to optimize an MBSE model of a space exploration rover. Investigation into the integration of MBSE and MDO with other design tools, such as simulation and

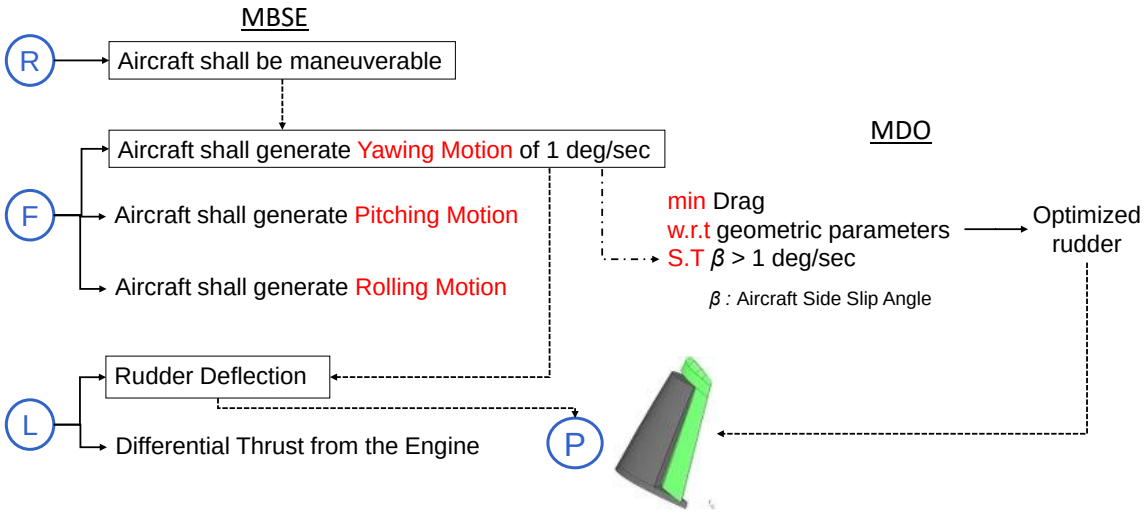

Fig. 1 Example illustrating MDO and MBSE

analysis software is well documented. Rasti et al. [28] proposed an approach that integrates MBSE, MDO, and Finite Element Analysis (FEA) to optimize the design of a gearbox for wind turbines.

While the above literature focus on highlighting key features of MDO and MBSE, very limited literature has been focused on developing a methodology for tightly integrating MBSE and MDO, where these features can be tested and implemented. Notably, Habermehl et al. [29] presented a comprehensive review of different approaches to integrating MBSE and MDO, and proposed a method for the integration of MDO and an MBSE system model. The proposed method involves identifying solutions for system functions in the system model. Next, the authors define processes that allow for the system to be designed. These design processes are termed 'design workflow' by the authors. Similarly, the authors' test of requirements was satisfied using a 'test workflow'. These two workflows, design, and test workflows are separate. To avoid the manual iteration of sequentially executing design and test workflows, an 'optimization workflow' is proposed, which is formalized in the MBSE system model. The limitations of the paper were that the executable behavior had to be manually implemented in the MBSE model, key MDO decisions relied on experience and logical reasoning, and the optimization workflow could only be applied to one type of MDO architecture. In a recent study, Bruggeman et al. [30] proposed an MBSE-based Requirement Verification Framework to facilitate the MDO process. The researchers integrated model-based requirements into the optimization problem and developed MDO workflows based on requirement verification methods. Nevertheless, the workflows were specific to the requirements and optimization framework, which implies that different MDO architectures necessitate the development of new optimization frameworks.

The current literature on integrating MBSE and MDO reveals a gap. Some studies have imposed requirements as constraints in the MDO problem while neglecting the need for a thorough integration between the two models. Other approaches have been tailored to specific MDO architectures, thus limiting their generalizability. Some studies have modeled the MDO process within the MBSE system, leading to the development of an executable behavior to run the MDO. However, this approach is time-consuming and requires modification of the representation within the MBSE when changes are made to the MDO process, resulting in a novel MBSE methodology for a particular MDO architecture.

This paper proposes a new methodology that addresses the challenges and limitations of integrating MDO and MBSE. The proposed approach builds on the RFLP approach [31]. The traditional RFLP that is widely used in the industry is retained. It incorporates product data from requirements, function, and physical designs [32] (represented as a CAD model), and is termed *Product RFLP*. The paper adapts a new RFLP, called *Process RFLP*[*], which focuses on the development, enumeration, and selection of design activities, such as simulations, analysis, MDO, design space

[*]Process RFLP was originally proposed by the Aerospace Systems Design Lab (ASDL) at the Georgia Institute of Technology

exploration, etc. In addition, the paper presents unambiguous rules for RFLP that can facilitate its implementation in software packages.

The proposed methodology involves a systematic approach to design that involves the decomposition of requirements into a hierarchical structure of derived requirements. The requirements are then mapped to functions that need to be performed in order to meet the requirements. Next, the options to perform the functions are enumerated: parts (of the mechanical system) in Product RFLP, and MDO in Process RFLP. The chosen option is then executed. To verify the system's compliance with the requirements, the results obtained from the execution of the chosen option are analyzed. The verification process checks whether the system meets the specified requirements and whether it aligns with its intended purpose. The primary objective of this process is to identify and correct any errors or inconsistencies in the model or system before its implementation, thus minimizing the risk of costly and time-consuming changes later in the design process [33].

In this work, the creation of the system model, verification of requirements, and evaluation of changes in requirements are accomplished using MATLAB. MATLAB provides a convenient option to implement RFLP. In particular, for Process RFLP, MATLAB allows system model objects to be converted into mathematical models using Simulink. Because Simulink can call external MDO models, the MBSE model is kept separate from the MDO model, and both models' original fidelity is preserved. Nonetheless, two-way information exchange exists between the two models.

The proposed methodology is demonstrated by designing a single-aisle transport category aircraft. Moreover, this methodology takes into account the analysis of changes in requirements, which is a fundamental aspect of MBSE that enables continuous improvement and refinement of the developed system. As requirements evolve, they can affect the system's design, behavior, and performance, thus impacting the entire system [34]. The test case will demonstrate how the methodology tracks the impact of requirement changes through the model.

The remainder of the paper is organized as follows: Section II covers the background on MBSE, MDO, RFLP and describes MATLAB's toolboxes used in this paper. Section III describes the method used in this work to integrate MBSE and MDO, the test case that will showcase the proposed method and presents the results. Finally, Section IV presents the conclusion and avenues for future work.

II. Background

This section provides an overview of Model Based Systems Engineering (MBSE) and Multidisciplinary Design Optimization (MDO), followed by a description of MATLAB's toolboxes used to implement the proposed RFLP framework.

A. Model Based Systems Engineering (MBSE)

MBSE presents a formalized approach that offers not only the ability to develop innovative and complex products within tight timeframes, but also provides a structured pathway for managing the inherent complexity of the processes involved, while incorporating a wide range of domain-specific environments[15]. These domain-specific environments are incorporated in MBSE models. Methodologies are used to design MBSE models. The collection of related processes, methods and tools used to support the discipline of system engineering in a "model-based" contest is termed as MBSE methodology[35].

The INCOSE MBSE Initiative conducted a comprehensive survey of candidate MBSE methodologies in 2007[15], which was subsequently formalized and published in 2008 as an INCOSE technical report[35]. The survey evaluated six candidate MBSE methodologies: INCOSE Object-Oriented Systems Engineering Method (OOSEM), IBM Rational Telelogic Harmony-SE, IBM Rational Unified Process for Systems Engineering (RUP-SE), Vitech MBSE Methodology, JPL State Analysis (SA), and Dori Object-Process Methodology (OPM).

One of the dominant methodologies used to design MBSE models in various industries today is: "The RFLP Approach". The RFLP approach has emerged as a prominent approach in the field of MBSE. In this paper, the RFLP approach will be introduced as the fundamental method for model-based design, with the aim of facilitating close interaction and collaboration among various engineering disciplines. The implementation of this approach is expected to enhance the efficiency of resources and processes, improve the quality of the product, and ensure that the final system meets the specified requirements, while reducing the design cycle time and engineering lead time. Effective articulation and integration of the diverse needs of clients, system functions, and operating principles from various disciplines are paramount in the design and development of complex, multidisciplinary products.

In the system analysis phase, the product under development is conceptually characterized through requirements definition, functional analysis, logical architecture design, and component specification. This phase results in product

models that provide initial details. Subsequently, in the system development phase, product development data such as 3D CAD models and behavior models are generated. Components are designed, simulated, and tested, followed by their integration into the system, and ongoing validation and verification activities are conducted. Finally, the system integration phase encompasses the comprehensive validation and verification of the integrated system.

Prior to the adoption of the RFLP approach, Systems Engineering (SE) in product development lacked sufficient support. However, the integration of an information model that spans all product development phases and disciplines has enabled the RFLP approach to provide holistic support for the design and development of projects based on systems engineering principles in various industries including aircraft's [36, 37].

The definition of RFLP provided by Dassault Systemes is presented below. This definition will be strictly followed throughout the paper.

- **Requirements Engineering and Management**: Define a set of criteria or specifications that must be met. These can either be corporate, end-user, financial, or design specifications. Requirements can take many different forms in MBSE, however in this paper, we stick to defining requirements by textual descriptions. Requirements can be used to inform the mechanical system's design, development, testing, and validation activities, and they can also Requirements serve as a basis for verifying that the mechanical system meets its intended purpose.
- **Functional Design**: Functional Design describes "what" functions must be performed to satisfy the requirements.
- **Logical Design**: Options for "how" to implement the functions while maintaining the overall mechanical system's requirements.
- **Physical Design**: The actual parts are specified to execute the logical model.

The benefits of MBSE system models are that it employs a centralized system model that continuously links development data. This model represents the target system or its subsystems, incorporating structures, artifacts, and domain-specific and domain-independent components, including simulation models [38]. A key characteristic of an efficient model-based system is the clear formalization of processes, methods, and tools. By integrating requirements and goals from multiple engineering disciplines, MBSE enables effective management of large-scale systems throughout their entire life cycles [39, 40]. MBSE has also been used in the aviation industry to identify regulatory gaps for electrified transport aircraft [41] and in airworthiness certification [42].

System design within MBSE can identify a wide range of technical alternatives with varying design parameters, depending on the specified needs at the system level. However, these technical alternatives may not always be compatible with each other, resulting in potential goal conflicts that need to be resolved. As MBSE techniques are often used to generate a limited number of variants, only a few viable options may satisfy the requirements. Determining the optimal strategy for resolving objective disagreements may be challenging, as it may be impossible to ascertain the best viable objective values. To address these limitations, MDO can be utilized.

B. Multidisciplinary Design Optimization(MDO)

It is clear that MBSE does not encompass analytical or numerical approaches, which are typically provided by additional tools and toolchains, leading to a potential lack of integration between system architecting and system analyses. A viable solution to the problem detailed above comes in the form of MDO, which has been widely recognized in the literature as an effective approach for addressing complex engineering challenges [43]. MDO is an engineering approach that involves the integration of multiple disciplines and optimization techniques to address complex design challenges [44].

MDO typically involves the use of computational models and simulations to capture the behavior of the system or product under consideration, and optimization algorithms to search for the best possible design solutions within the given design space [45, 46]. The objective is to identify design variants that achieve the best performance in terms of predefined criteria, such as minimizing weight, maximizing efficiency, or meeting specific performance requirements [47]. MDO provides a systematic and quantitative approach to decision-making in engineering design, enabling engineers to explore a wide range of design alternatives, evaluate their performance, and make informed decisions to achieve optimal design solutions [48, 49].

MDO has shown significant potential in improving the efficiency and effectiveness of the design process, reducing the need for costly physical prototypes and testing, and accelerating the development of innovative and optimized products [50]. By considering the interactions and trade-offs among different disciplinary components, MDO enables engineers to make informed decisions and achieve optimal design solutions that meet performance requirements while balancing competing design objectives [51].

Previously, a bottleneck in Multidisciplinary Design Optimization (MDO) was the challenge of obtaining derivatives

of the objective function and constraints in relation to design variables. However, PI Hwang has recently introduced the Computational System Design Language (CSDL) as a solution to this issue [52]. CSDL is a new modeling language that facilitates the automated computation of derivatives across tightly coupled disciplines. CSDL adopts a functional programming paradigm, which means code is written in a natural way with functions and declarative statements that are indistinguishable from regular Python code. Thus, practitioners can write near-Python code and CSDL automates computing the derivatives. CSDL is designed to facilitate vectorized implementations of models with minimal use of for-loops. Therefore, the computational graph is as compact as possible; however, it is decomposed to the greatest extent—to the level of arithmetic and other fundamental binary and unary operations. Various studies have demonstrated the efficacy of CSDL in large-scale MDO for diverse applications, including electric vertical takeoff and landing (eVTOL) vehicle concepts [53], trajectory optimization [54], spacecraft design [55], and concentric tube robots [56].

1. Early-Stage Aircraft Design Computational Models

The early stages of aircraft design are complex and present a large design space from which a good candidate design must be found that satisfies the top-level aircraft requirements (TLARs). The requirements arise from market, performance, and environmental considerations. A number of aircraft conceptual design methods and tools have been presented in the literature. Historically, design is done using regressions or using a limited set of parameters describing aerodynamics, propulsion, and flight conditions. The design methods by Raymer [57], Roskam [58], Torenbeek [59], and Loftin [60] work well for conventional tube-and-wing aircraft. Flight Optimization System (FLOPS) [61] is a tool developed by NASA for this conceptual aircraft design that contains regressions to predict key aircraft parameters such as the lift-to-drag ratio, the thrust produced by the engines, and the weight of the aircraft. The regressions are reliable due to the data accumulated over the last few decades and work well for conventional concepts. Other methods use physics-based models in conjunction with regressions. SUAVE [62] is a conceptual-level aircraft design environment that uses physics-based methods to size novel configurations and unconventional propulsion architectures. Chakraborty et al. [63, 64] developed an energy-based sizing framework named PEACE. All these methods are the *logical* options of Process RFLP.

In this work, we use the method presented by Loftin [60] to perform conceptual sizing of a single-aisle long-range transport category aircraft like the Airbus A320 and the Boeing B737.

Loftin's method is a set of empirical equations that relate various aircraft design parameters to the aircraft's performance characteristics. The equations were developed by Howard W. "Bud" Loftin, a noted aerospace engineer, and were first published in his book "Quest for Performance: The Evolution of Modern Aircraft" [60]. The Loftin Equations are used in the preliminary design of subsonic aircraft to estimate the aircraft's performance characteristics, such as maximum speed, range, and rate of climb, based on the design parameters of the aircraft. The equations are based on empirical data and simplifying assumptions and are not intended to provide accurate predictions of aircraft performance. Instead, they provide a useful tool for comparing different aircraft configurations and exploring the trade-offs between different design parameters. The method ensures the aircraft will be sized to meet the regulations under 14-CFR Part 25. The method is described as an XDSM in Fig. 2. It is setup as the following optimization problem

$$
\begin{aligned}
\text{minimize} \quad & W_{TO} & & \\
\text{with respect to} \quad & AR_{eff}, \Lambda_{25}, \lambda_{eff}, t/c, V/V_{md} & & \\
\text{subject to} \quad & L_{cr} > W_{TO} & & \text{Lift constraint} \\
& V_{cr} > V_{stall} & & \text{Stall constraint} \\
& V_T > V_F & & \text{Fuel tank volume constraint} \\
& m_L > m_{ZFW} + m_{F_{res}} & & \text{Landing weight constraint}
\end{aligned}
\tag{1}
$$

Here, the design variables are the wing's aspect ratio AR_{eff}, the wing's quarter chord sweep Λ_{25}, the wing's taper ratio λ_{eff}, and the wing's average thickness-to-chord ratio t/c. In addition, the parameter V/V_{md} serves as a surrogate for the lift-to-drag ratio L/D and is the design variable responsible for ensuring the aircraft is sized to meet the mission.

The user specifies the stakeholder requirements: the range of the aircraft R, the cruise Mach number M, the maximum take-off and landing field lengths s_{TOFL}, s_{LFL}, the number of passengers n_{pax}, and the mass of the cargo to be carried m_{cargo}. Other constants are specified based on historical values: the maximum lift coefficient at takeoff and landing $C_{L_{max,TO}}, C_{L_{max,L}}$, the fuselage outer diameter d_{F_o}, and the engine by-pass-ratio BPR.

The cruise aerodynamic block in Fig. 2 uses the aircraft geometry to determine the $(L/D)_{max}$, the lift coefficient at cruise $C_{L_{cr}}$. The block also imposes a constraint that ensures the cruise velocity V_{cr} is greater than the stall velocity

V_{stall}.

The constraint analysis block imposes point-performance on the aircraft. It considers the following mission segments: cruise, takeoff, climb, maneuver, landing, and mission approach. At each segment, the required thrust-to-weight ratio T/W and the wing loading W/S is computed. The maximum of these is used to size the aircraft.

The mission analysis block runs through the entire mission and determines the fuel needed for each segment of the mission. Additional fuel to complete safety-related portions of the mission (specified by 14-CFR regulations) include 200 nautical miles flight to an alternate airport, 5% extra fuel for long-range missions, and 1800 seconds of loiter time are taken into account. The block uses the information to compute the maximum take-off gross weight W_{TO}. The mission analysis block also imposes constraints to ensure the aircraft produces sufficient lift at cruise: $L_{cr} > W_{cr}$ and ensure the aircraft's allowable landing weight m_L is greater than the actual landing weight $m_{ZFW} + m_{F_{res}}$, where m_ZFW is the zero-fuel weight of the aircraft, and $m_{F_{res}}$ is the weight of reserve fuel.

With the two ratios W/T and W/S, and the weight W_{TO}, the aircraft can be sized to obtain the necessary wing area S_W and the sea-level static thrust at take-off T_{TO}.

Finally, the fuel tank volume block will ensure the aircraft's fuel tanks in the wing have sufficient volume capacity V_T to carry the volume of the necessary fuel for the mission V_F. This block takes into account 14-CFR 25.979(b) fuel tank corrections and accounts for unusable fuel.

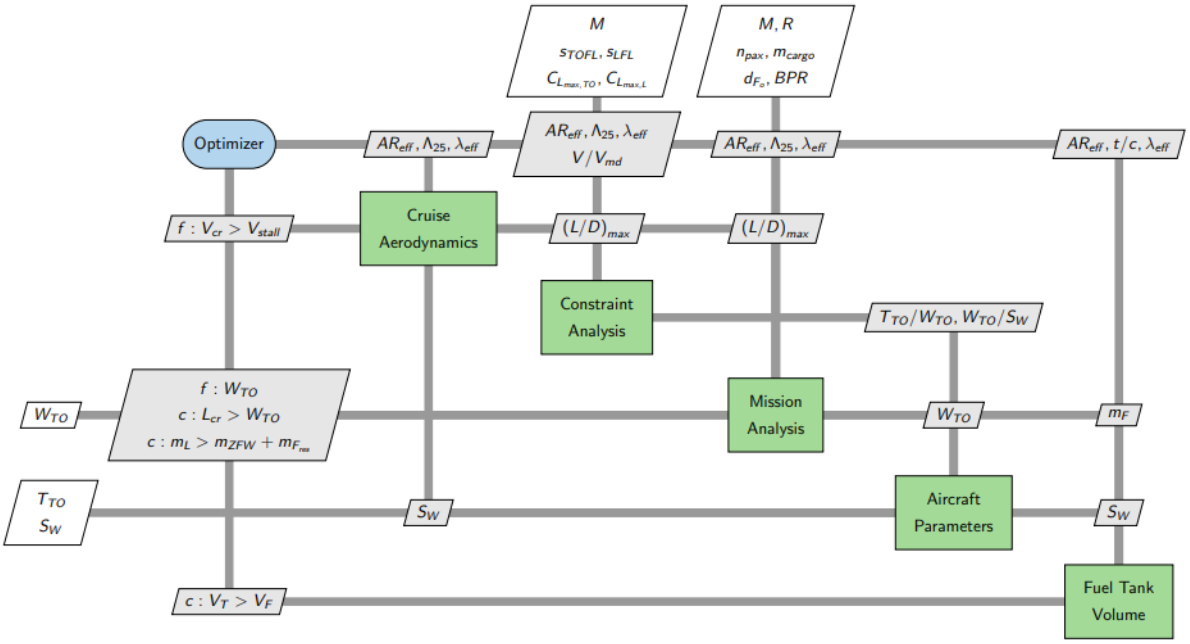

Fig. 2 XDSM diagram for aircraft sizing

Overall, the Loftin Equations are a useful tool for the conceptual design of transport category aircraft. While they have limitations and should be used with caution, they provide a quick and easy way to estimate the performance of a proposed aircraft configuration and explore the trade-offs between different design parameters.

C. MATLAB Toolboxes and Software Platforms Employed

1. Requirements Editor

A Requirements Editor implemented within the MATLAB environment is a sophisticated software tool that empowers users to systematically define, manage, and analyze requirements for software projects or systems. With its intuitive graphical interface, the Requirements Editor in MATLAB enables users to create, edit, and meticulously organize requirements in a hierarchical or modular fashion, adhering to established industry standards and best practices.

Within the Requirements Editor in MATLAB, users can meticulously specify diverse types of requirements, including functional, non-functional, and performance-related requirements, while meticulously attributing them with

key attributes such as priority, status, and traceability. In addition to its comprehensive requirements management capabilities, the Requirements Editor in MATLAB offers sophisticated analysis and verification tools. These tools facilitate meticulous checking for inconsistencies, conflicts, or gaps in the requirements documentation, while providing robust traceability management features to establish and track traceability links among requirements, design artifacts, and test cases.

2. System Composer

MATLAB System Composer is a powerful tool that offers comprehensive support for MBSE. It provides a graphical environment for designing, simulating, and analyzing complex systems, making it a valuable resource for engineers and researchers involved in system-level design and optimization. With its intuitive interface, MATLAB System Composer allows users to create and manipulate system models using a wide range of modeling notations, such as block diagrams, state machines, and activity diagrams, among others.

One of the key features of MATLAB System Composer is its ability to facilitate the integration of different system models across multiple disciplines, enabling interdisciplinary collaboration and system-level optimization. Furthermore, MATLAB System Composer provides powerful simulation and analysis capabilities that enable users to assess the behavior and performance of their system models. Additionally, MATLAB System Composer provides a rich set of analysis tools for system-level performance evaluation, such as sensitivity analysis, optimization, and trade-off analysis, which are crucial for making informed decisions during the system design and optimization process. The diverse range of features and constructs offered by System Composer is designed with a clear emphasis on prioritizing these overarching goals [65].

3. Simulink

MATLAB Simulink is a robust graphical simulation and modeling tool that finds widespread use in the fields of engineering and science. Simulink provides a comprehensive and intuitive environment for designing, simulating, and analyzing dynamic and static systems that enables users to harness MATLAB's extensive numerical computation, data analysis, and visualization capabilities. A notable feature of Simulink is its block diagram-based modeling paradigm, which facilitates visual representation of system components as interconnected blocks, accompanied by the specification of their behavior using built-in or custom-designed mathematical functions. This graphical approach to modeling empowers users with efficient and intuitive system design, offering real-time or accelerated time simulation capabilities to gain valuable insights into system performance, stability, and robustness. Moreover, Simulink offers an extensive library of pre-built blocks and toolboxes tailored to various domains, including aerospace, automotive, communications, and robotics, allowing users to leverage ready-to-use components and models for rapid prototyping and system development [66, 67].

4. Test Harness

A test harness in MATLAB is a robust tool used in testing to systematically validate the functionality and performance of code modules, functions, or systems. It provides a structured framework for creating, executing, and analyzing tests, ensuring that the software meets the desired specifications and requirements in a systematic and organized manner. One of the key features of a test harness in MATLAB is its ability to establish a well-defined and organized approach for designing and implementing tests. Test cases can be precisely defined within the harness, specifying inputs, expected outputs, and tolerances for comparison. Automated testing, which is facilitated by test harnesses, saves time and effort compared to manual testing, and enables swift identification and resolution of issues in the code.

5. Test Manager

MATLAB Test Manager is a robust and comprehensive tool provided by MathWorks and is designed to streamline the testing process and ensure the quality and reliability of code and systems developed using MATLAB. One of the prominent features of MATLAB Test Manager is that it provides comprehensive tools for analyzing test results. They allow users to easily compare actual outputs with expected outputs, generate detailed reports, and visualize test results through graphical plots and charts. This facilitates the identification of discrepancies and the understanding of code behavior under different test scenarios, aiding in debugging and improving the software's quality. Test results can also be exported in various formats, facilitating documentation and sharing of test results with team members or stakeholders [68].

III. Design Process based on the RFLP Approach

This section walks through the proposed methodology: a design process that leverages combinations of RFLP (Figure 3). Rules for implementing RFLP in a manner that facilitates easy and rigorous implementation in software packages are presented. The proposed methodology will be demonstrated through a test case whereby it is used to guide the conceptual level design of a single-aisle transport category aircraft.

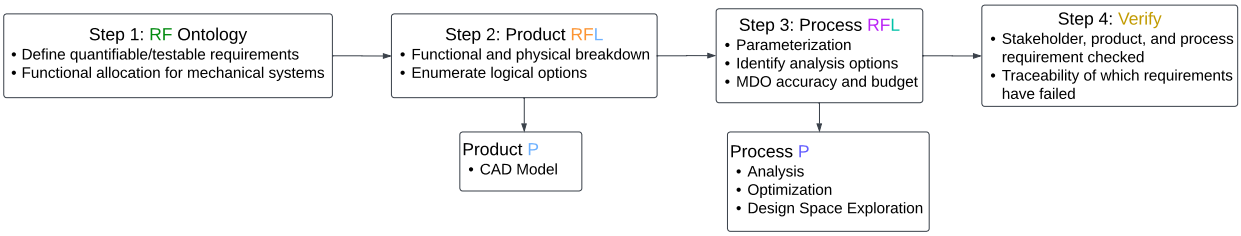

Fig. 3 Design process using product and process RFLP

Prior to delving into the proposed methodology, it is imperative to establish precise definitions of RFLP. The definitions used in this paper are illustrated in Section II.A. The conventional order is to commence with Requirements Management (R), proceed with Functional Analysis (F), followed by Logical Design (L), and culminate with Physical Design (P). This paper adheres to this prescribed order.

The example illustrated in Section I is examined once again. Consider the following example: a requirement specifies that the "Aircraft shall be maneuverable in the air". A function that enables this requirement is "Aircraft shall generate yawing moment." Notably, the function does not specify how the yawing moment should be generated; rather, this is the role of logical options. For instance, two options might include 1) rudder deflection or 2) differential thrust from the engines. If we choose to use the rudder deflection option, the physical model would be the CAD model of the rudder, which could be employed to manufacture the product. This example illustrates the conventional RFLP approach. In this work, a design process that uses combinations of RFLP is presented.

A. RF Ontology

The initial step of the proposed methodology involves identifying the requirements of the stakeholders. In this test case, two requirements of interest are that the "Aircraft shall carry passengers and cargo", and that the "Aircraft shall fly a mission". The focus of the test case will be on the latter requirement. In the proposed methodology, requirements are derived from other requirements to provide more detailed specifications. For instance, the requirement "Aircraft shall fly a mission" can be used to derive additional requirements that cater to specific geographical regions. For example, the derived requirement "Aircraft shall fly between any airports in the United States" is based on the intended use of the aircraft. Once the stakeholders' requirements have been determined, the next step is to identify the critical functions that need to be executed to fulfill these requirements. To satisfy the "Aircraft shall fly a mission" requirement, the identified functions must cover all potential scenarios, ranging from the aircraft being stationary at the airport to the aircraft flying to an alternate airport if necessary and ultimately landing. Figure 4 presents the requirements and functions employed in this test case. At this stage, the specifics of how the functions will be performed and the execution of the logical and physical models are not specified. This stage is referred to as the *RF Ontology*. The RF Ontology's requirements and functions are implemented using MATLAB's Requirements Editor. Figure 4 emphasizes the function "Aircraft shall land," which will be comprehensively explored in subsequent steps of the proposed methodology.

B. Product RFLP

The second step in the proposed methodology is referred to as *Product RFLP*. The first rule established in this paper is

Rule 1:
Product requirement is derived from the RF Ontology

Figure 5 provides a visual representation of this rule being implemented, where it can be observed that the function of the RF Ontology derives the Product Requirement.

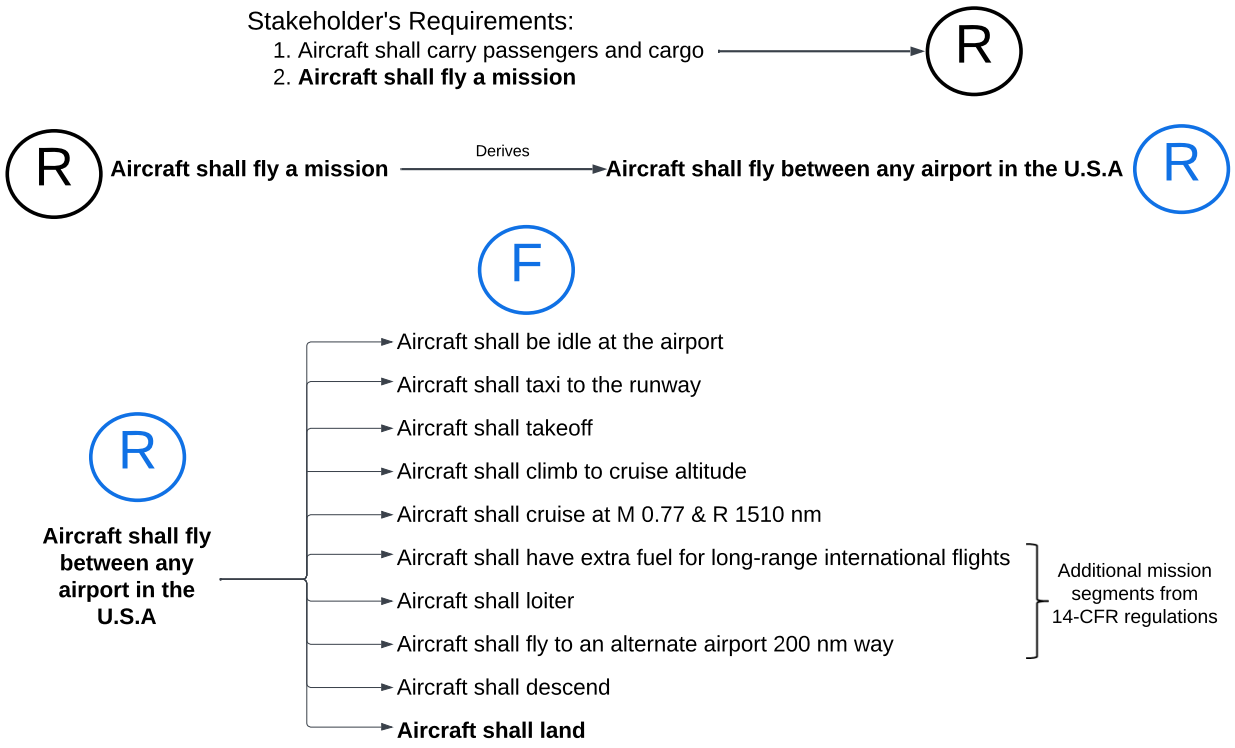

Fig. 4 Step 1: RF Ontology. Two requirements are shown (top). The functions necessary to satisfy the 2nd requirement are detailed (bottom). The functions include regulatory mission segments established by the FAA.

A product requirement can be derived from the function “Aircraft shall land” and the requirement “Aircraft shall fly between any airports in the United States”. The derived product requirement is ”Aircraft shall land at airports that have an elevation between 9934 feet and 210 feet above and below mean sea level respectively”, which enables the aircraft to land at any airport in the United States. This example is chosen such that the aircraft can land at the highest (Lake County Airport at Leadville, Colorado) and lowest (Furnace Creek Airport at Death Valley, California) airports in the United States. The relationship between a function and a product requirement is depicted in Figure 5. All figures shown are accurate representations of MATLAB figures.
A general rule can be established based on the above discussion:

Rule 2:
A requirement can be derived from another requirement, a function, or a logical option. This requirement is termed a *derived requirement*

After establishing the product requirement and function, the logical options are enumerated. For the function ”Aircraft shall land at airports that have an elevation between 9934 feet and 210 feet above and below mean sea level respectively”, two possible options are: 1) taxi-based landing, 2) or vertical takeoff or landing (VTOL). It is evident that the Logical Design is where different options for "how" to implement the function are identified. The physical design involves the actual execution of the logical model, which, in this case, will be implemented through the use of a CAD model representing the various logical design options. This leads to a rule:

Rule 3:
Physical Design in the Product RFLP should be represented by a CAD Model

The requirements and functions of the product step are implemented using MATLAB’s Requirements Editor, while the logical options are implemented in MATLAB’s System Composer. The physical design, on the other hand, is left to the user’s preferred CAD software. Figure 5 shows the RF Ontology in conjunction with the Product RFLP Diagram for the "Aircraft shall land" function.

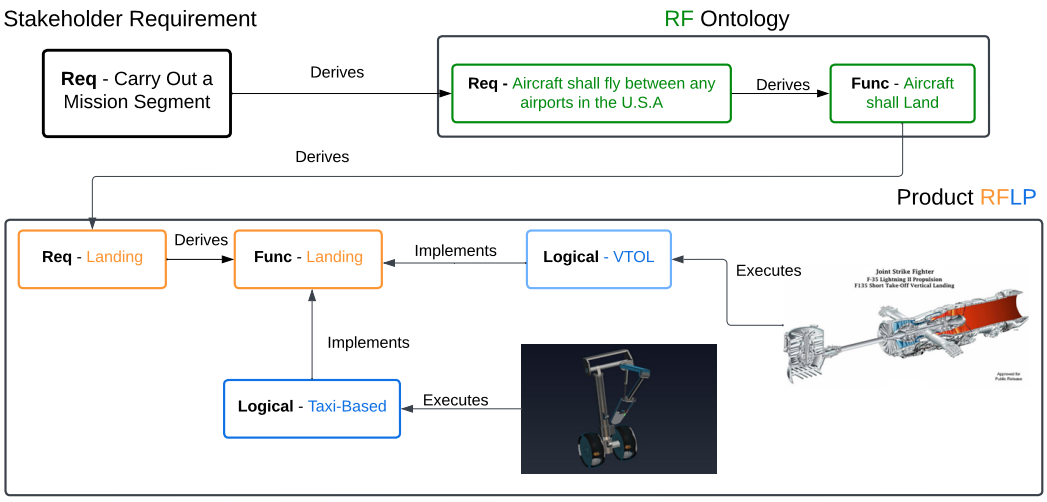

Fig. 5 RF Ontology and Product RFLP diagram for the "Aircraft shall land" function.

The above discourse demonstrates that, at this juncture, no analysis or simulation is carried out using any tools. Instead, essential characteristics of MBSE are determined. Nonetheless, when defining logical options, the relevant parameters can be stored, such as the numerical values of mass and drag of the landing gear. Geometric particulars of the logical option can be accessed via the CAD model. All of this information will be crucial when conducting an analysis of the logical option. Consequently, a fourth rule can be stated as

Rule 4:

All the parameters required to simulate the mechanical system should be stored in the Logical and Physical Design

C. Process RFLP

The RFLP approach has conventionally been applied to the product development process, as explained in the previous section. However, a novel concept known as *Process RFLP* has been proposed by the Aerospace Systems Design Lab at Georgia Tech. The Process RFLP involves the development, enumeration, and selection of simulations and is utilized in various design activities, such as sensitivity analysis, optimization, and design space exploration.

In the Product RFLP, it was established that the RF Ontology is responsible for deriving product requirements. However, in the Process RFLP, additional requirements must be taken into consideration. These requirements are dependent on various factors such as the available time, budget, computational resources, and the maturity and accuracy of simulation tools. Thus, the engineering team can impose process requirements to ensure that the design process meets their needs. An example of a process requirement can be “MDO shall provide a solution within 15 minutes” or “MDO shall optimize take-off gross weight” or “MDO shall provide a solution with a 95 percent confidence interval”. It is evident that process requirements can vary depending on the specific needs of the engineering team. Another rule is established here:

Rule 5:

The engineering team specifies additional Process Requirements

In this test case, the Process Requirement is determined as “Simulation shall compute the landing-field length”. Similar to the RF Ontology and the Product RFLP, the requirement determines the function. The process requirements and functions are implemented in MATLAB’s Requirements Editor.

At this point, it is important to address how simulation, analysis, and MDO fit into the design process. Consider Figure 5. Shown are two logical options: 1) taxi-based landing, 2) or vertical takeoff or landing (VTOL). Consider another requirement “Aircraft shall produce energy to complete the mission”, its function “Aircraft shall generate

energy", and its logical options: 1) fuel-based, 2) battery-based. The choice of logical options determines the required simulation tool.

For this test case, the chosen logical options are taxi-based landing and fuel-based energy, which offer various computational models such as SUAVE, FLOPS, and Loftin Equations (Section II.B.1). These options are part of the Process Logical, which enumerates the available computational models for conducting design. This establishes the next rule

Rule 6:

Chosen Product RFLP determines the Process Logical. The process logical enumerates options for computational models

At this stage, the user can select the optimal analysis method based on the process requirement, after determining the process logical options. In the current test case, the user chooses Loftin's equations as the design method for the aircraft. The logical options are defined using MATLAB's System Composer, which is then converted into a Simulink model via the so-called 'round-trip engineering', an inbuilt process within MATLAB. The Simulink model serves as the computational model.

In accordance with Rule 4, all the necessary parameters for simulating the mechanical system should be saved in both the Logical and Physical Designs. In the Process Logical, assumptions, empirical values, and metadata (such as the computational cost, runtime, and accuracy of the computational model) are stored.

Once the logical option is finalized, the physical design for the Process RFLP is established. In the Product RFLP, the physical design is represented by the CAD model as it executes the logical options of the product. Similarly, in the Process RFLP, the computer program utilized in the analysis is the physical design, as it executes the logical options of the process.

Rule 7:

The physical design in the Process RFLP is the computer program used to simulate the mechanical system

The Process Physical Model that implements Loftin's equations is done in CSDL. MATLAB has built-in support to execute computer models via the command line. This functionality is used to execute the computer program that implements Loftin's equations in CSDL.

To summarize the discussion up to this point, Figure 5 shows that the RF Ontology derives the Product RFLP. With this reference, Figure 6 proceeds to illustrate how the Process Logical options are derived from the chosen Product RFLP, and how the Process requirements influence the process logical options, revealing a complete Process RFLP.

D. Verify

After the execution of the product physical, the subsequent stage involves verifying the product requirements based on the MDO outputs. The verification of requirements holds significant importance in MBSE as it guarantees that the mechanical system conforms to the defined requirements, and the end product accomplishes its intended purpose. The verification process is depicted in Figure 7. MATLAB offers a convenient method for the management and engineering personnel to assess whether the requirements have been fulfilled by utilizing a visual representation as elucidated in Figure 8. The three constraints checked in this test case include the "Fuel Tank Volume", "Cruise", and "Landing".

Table 1 Comparison of the actual values used in an A320-200 aircraft with the designed MDO code using Loftin's equations

Parameter	Actual	Calculated	% difference
Takeoff Mass (kg)	73500	71099.83	3.265537415
Landing Mass (kg)	64500	62425.65	3.216046512
Total Fuel Mass (kg)	13000	12930.6	0.533846154
Operating Empty Mass (kg)	41244	39088.26	5.226796625
Wing Area (m^2)	122.4	118.49	3.194444444
Takeoff Thrust, all engines (N)	222400	214455.12	3.572338129

Executing the MDO sizes the aircraft. Table 1 compares the outputs from Loftin's method to those of the A320-200 aircraft. It is seen that the method is well-suited to design conventional fuel-based tube-and-wing transport category

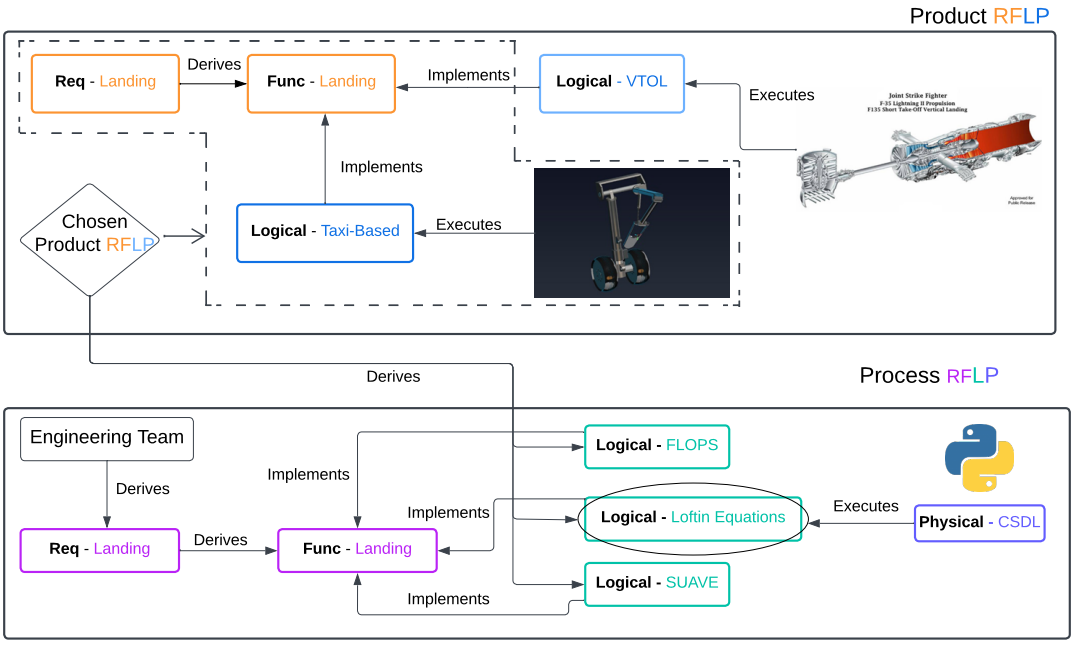

Fig. 6 Product RFLP and Process RFLP being shown.

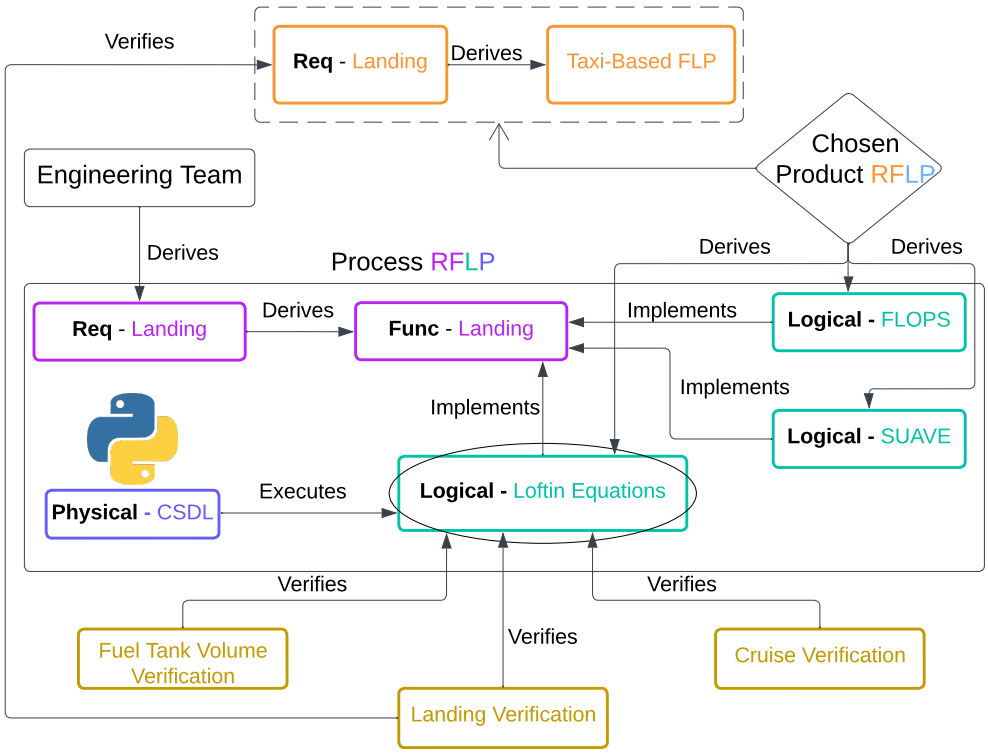

Fig. 7 Process RFLP and verification being shown.

aircraft. The obtained parameters are fed back into the parameters of the Product Logical. For example, the 'Aircraft'

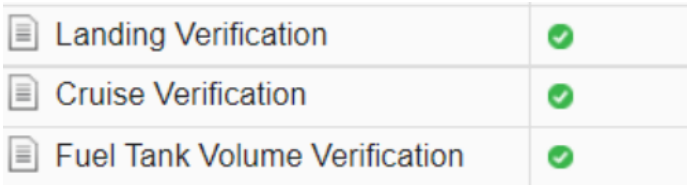

Fig. 8 Requirement verification.

logical component will store the aircraft-level parameters such as takeoff mass, landing mass, and operating empty mass. The 'Fuel Tank' logical component will store the total fuel mass and the fuel tank volume. Similarly, the 'Wing' component stores the wing area and the 'Engine' component stores the takeoff thrust. The geometric parameters are also used to update the Product Physical, i.e., the CAD model.

The proposed methodology also facilitates the ability of the engineering team or stakeholders to modify requirements and assess the resulting impacts. This can pertain to modifications in stakeholder, process, or product requirements.

For the specific test case, the user modifies two parameters specified by the stakeholder, specifically the Mach number and range. This modification is intended to affect the product requirements, although this approach can be extended to impact any requirement type as previously noted. The proposed approach identifies which requirements have not been met. Here, the fuel tank volume verification has not been satisfied. This is depicted in Figure 9.

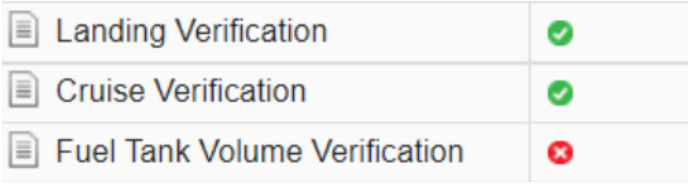

Fig. 9 Fuel Tank Volume verification failing

Once the engineering team recognizes which requirement has not been met, a *traceability diagram* incorporating the RF Ontology, Product RFLP, and Process RFLP can be generated. Traceability diagrams track changes in a system by allowing for the identification of the effects of modifications to requirements. By comprehending the consequences of modifications, it is easier to assess the feasibility and cost of implementing such changes and to make informed decisions about the design process. Figure 10 depicts the requirements and functions impacted by this failure.

IV. Conclusion

Designing mechanical systems is a complex task that involves meeting changing stakeholder needs, incorporating new technologies for increased efficiency, and meeting stringent regulatory requirements for safety standards. Additionally, accurately simulating and designing mechanical systems to maximize performance metrics poses a significant challenge. These factors contribute to the substantial cost and extended time-to-market of mechanical systems.

The methodology presented in this paper offers a new approach to designing mechanical systems by extending the well-known Requirements Engineering, Functional Design, Logical Design, Physical Design (RFLP), a systematic approach that is used to define requirements, create functional designs, and implement logical and physical designs of the mechanical system.

The proposed methodology uses combinations of RFLP and involves four steps: 1) RF Ontology, 2) Product RFLP, 3) Process RFLP, and 4) Verification.

In RF Ontology, the requirements analysis examines, evaluates, and translates stakeholder needs into a set of functional and performance requirements. The goal is to determine the needs that make up a system to satisfy an overall need. The functional analysis and allocation is a top-down process of translating system-level requirements into detailed functional and performance design criteria. The result of the process is allocated system requirements that are traceable to each system function

Next, the Product RFLP defines system specifications based on the RF ontology and establishes functional and logical design models. It is important to note that multiple functions can exist for a single requirement to ensure comprehensive coverage. A similar scenario can be replicated for any product or process requirement. For every

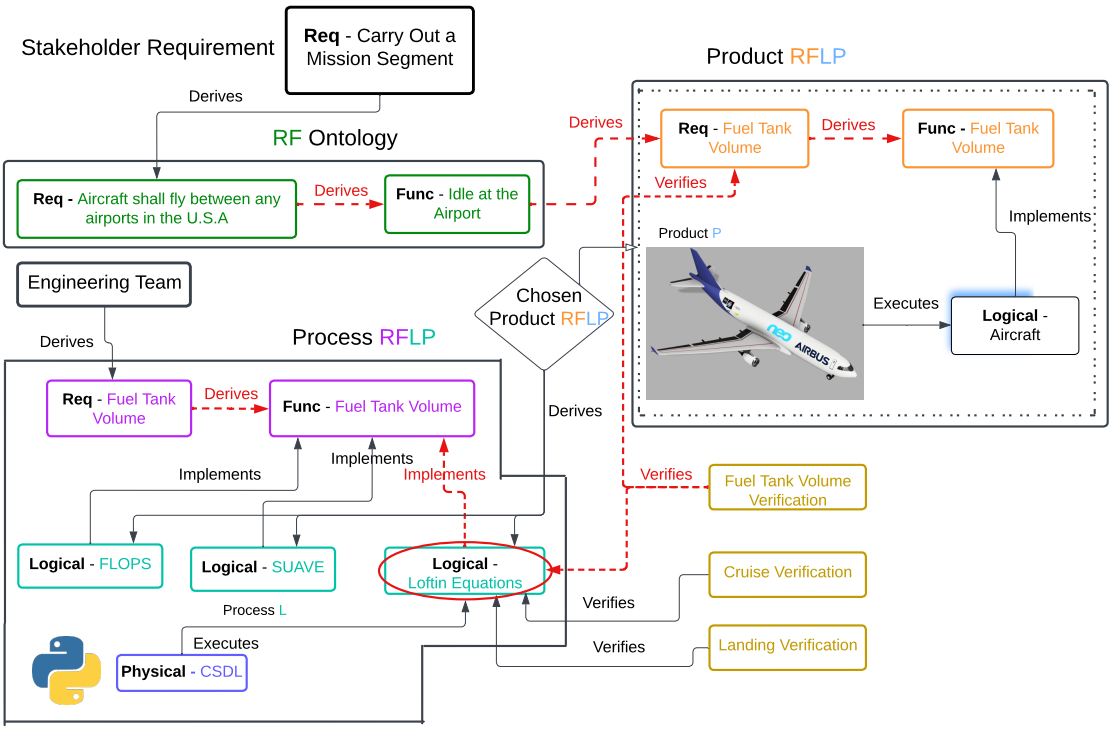

Fig. 10 Fuel Tank Volume Check traceability

function, logical options are enumerated to achieve the function. The physical design here is the CAD model of the logical option.

The next step is the Process RFLP, a novel method that retains the RFLP structure and applies it to design activities such as simulations, analysis, and Multidisciplinary Design Optimization (MDO). The key difference between product and process arises in the logical and physical designs. The Process logical involves options for computational models to simulate the mechanical system. The Process Physical is the computer program that executes the computational model.

An advantage of the proposed RFLP methodology becomes apparent in the last step, Verification, where requirements are evaluated and verified. The traceability diagram was shown to visually depict a means to trace and track the flow of information, requirements, and design decisions across various components, facilitating effective communication and coordination among stakeholders involved in the design process.

The benefits of the methodology are:

- The systematic approach provided by RFLP is extended. Previously RFLP was applied to only products, however, the Process RFLP is applied to include simulations and MDO.
- The approach presents unambiguous rules for RFLP that facilitate easy implementation in software packages and clearly delineates the MBSE model and MDO model, removing the need to model the details of MDO inside the system model.
- Unlike previous literature where specific MDO architectures are considered, this approach is agnostic to how the user wants to execute the MDO, making it more flexible and adaptable to different scenarios.

All the MBSE and MDO models used in this work have been made available open-source on Github: pyMbseMdo. Future work will consider the design of an electric air taxi concept, which involves a significant number of regulatory requirements, computational wall-time budget constraints, geometric changes, and physics-based models. The application of the proposed RFLP design process to this problem presents a challenging yet promising opportunity to evaluate the effectiveness of the methodology.

Acknowledgments

The authors would like to extend their sincere appreciation to Ruxandra Duca for her invaluable guidance and feedback on the RFLP approach during the development of this paper. Additionally, we express our gratitude to our colleagues at the Large Scale Design Optimization Lab for their valuable insights and discussions that contributed significantly to the final outcome of this work. In particular, we thank Marius Ruh for his contributions in formulating some of the definitions used in this paper.

References

[1] Sage, A. P., and Armstrong, J., J. E., "Introduction to Systems Engineering," *John Wiley & Sons, New York, NY*, 2000.

[2] Whitney, D. E., *Mechanical assemblies: their design, manufacture, and role in product development*, Vol. 1, Oxford university press New York, 2004. https://doi.org/10.1016/b0-08-043152-6/03027-8.

[3] Boothroyd, G., "Product design for manufacture and assembly," *Computer-Aided Design*, Vol. 26, No. 7, 1994, pp. 505–520. https://doi.org/10.1016/0010-4485(94)90097-3.

[4] Hammer, M., Maschotta, R., and Zimmermann, A., "Model-driven application development for evaluation and optimization of automotive E/E-architectures," *2021 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE)*, IEEE, 2021, pp. 1–8. https://doi.org/10.1109/RASSE53195.2021.9686943.

[5] Thomas, E., Thomas, O., Bianconi, R., Crespo, M., and Daumas, J., "Towards enhanced process and tools for aircraft systems assessments during very early design phase," *Proceedings of the 11th International Modelica Conference, Versailles, France, September 21-23, 2015*, Linköping University Electronic Press, 2015, pp. 831–843. https://doi.org/10.3384/ecp15118831.

[6] Fox, R. L., *Optimization methods for engineering design*, Addison-Wesley Publishing Company, 1971. https://doi.org/10.1115/1.3426572.

[7] Fakih, M., Klemp, O., Puch, S., and Grüttner, K., "A modeling methodology for collaborative evaluation of future automotive innovations," *Software and Systems Modeling*, Vol. 20, No. 5, 2021, pp. 1587–1608. https://doi.org/10.1007/s10270-021-00864-3.

[8] Hennig, A., Topcu, T. G., and Szajnfarber, Z., "So You Think Your System Is Complex?: Why and How Existing Complexity Measures Rarely Agree," *Journal of Mechanical Design*, Vol. 144, No. 4, 2022. https://doi.org/10.1115/1.4053291.

[9] Alt, O., *Modellbasierte Systementwicklung mit SysML*, Carl Hanser Verlag GmbH Co KG, 2012.

[10] Beier, G., Rothenburg, U., Woll, R., and Stark, R., "Modellbasiertes systems engineering: Durchgängige Entwicklung mit erlebbaren Prototypen," *Digital Engineering*, Vol. 3, 2012, pp. 14–17. https://doi.org/10.1007/s35763-012-0213-3.

[11] Graessler, I., Hentze, J., and Oleff, C., "Systems engineering competencies in academic education: An industrial survey about skills in systems engineering," *2018 13th annual conference on system of systems engineering (SoSE)*, IEEE, 2018, pp. 501–506. https://doi.org/10.1109/SYSOSE.2018.8428879.

[12] Friedenthal, S., Griego, R., and Sampson, M., "INCOSE model based systems engineering (MBSE) initiative," *INCOSE 2007 symposium*, Vol. 11, sn, 2007. https://doi.org/10.1002/j.2334-5837.2007.tb00839.x.

[13] INCOSE, A., "World in motion-systems engineering vision 2025, international council on systems engineering," , 2014. https://doi.org/10.1002/j.2334-5837.2014.tb01113.x.

[14] Martin, J. N., *Systems engineering guidebook: A process for developing systems and products*, Vol. 10, CRC press, 1996. https://doi.org/10.1201/9781420073620.

[15] Estefan, J. A., et al., "Survey of model-based systems engineering (MBSE) methodologies," *Incose MBSE Focus Group*, Vol. 25, No. 8, 2007, pp. 1–12. https://doi.org/10.1002/j.2334-5837.2007.tb00478.x.

[16] Kleiner, S., and Kramer, C., "Model based design with systems engineering based on RFLP using V6," *Smart Product Engineering*, Springer, 2013, pp. 93–102. https://doi.org/https://doi.org/10.1007/978-3-642-30817-8_10.

[17] Cordero, S. S., Fortin, C., and Vingerhoeds, R., "Concurrent conceptual design sequencing for mbse of complex systems through design structure matrices," *Proceedings of the Design Society: DESIGN Conference*, Vol. 1, Cambridge University Press, 2020, pp. 2375–2384. https://doi.org/10.1017/dsd.2020.214.

[18] Min, B. I., Kerzhner, A. A., and Paredis, C. J., "Process integration and design optimization for model-based systems engineering with SysML," *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 54792, 2011, pp. 1361–1369. https://doi.org/10.1115/DETC2011-48360.

[19] Simpson, T. W., and Martins, J. R., "Multidisciplinary design optimization for complex engineered systems: report from a national science foundation workshop," *Journal of Mechanical Design*, Vol. 133, No. 10, 2011. https://doi.org/10.1115/1.4005224.

[20] Xie, J., Cai, Y., Sarojini, D., Harrison, E. D., and Mavris, D. N., "Certification-Constrained Vertical Tail Sizing and Power Split Optimization for Distributed Electric Propulsion Aircraft," *Journal of Aircraft*, 2023, pp. 1–18. https://doi.org/10.2514/1.C034728.

[21] Bussemaker, J., Boggero, L., and Ciampa, P. D., "From system architecting to system design and optimization: A link between mbse and mdao," *INCOSE International Symposium*, Vol. 32, Wiley Online Library, 2022, pp. 343–359. https://doi.org/10.1002/j.2334-5837.2022.01420.x.

[22] Chaudemar, J.-C., and de Saqui-Sannes, P., "Mbse and mdao for early validation of design decisions: a bibliography survey," *2021 IEEE International Systems Conference (SysCon)*, IEEE, 2021, pp. 1–8. https://doi.org/10.1109/SysCon49348.2021.9474047.

[23] Ciampa, P. D., La Rocca, G., and Nagel, B., "A mbse approach to mdao systems for the development of complex products," *AIAA Aviation 2020 Forum*, 2020, p. 3150. https://doi.org/10.2514/6.2020-3150.

[24] Nagel, R., Zhang, W., Mahadevan, S., and Cutler, P., "Integration of Model-Based Systems Engineering and Multidisciplinary Design Optimization for Aircraft Systems," *AIAA Journal*, Vol. 57, No. 11, 2019, pp. 4641–4652. https://doi.org/10.2514/1.J057879.

[25] Wang, W., Liu, Z., Gu, X., Li, W., and Wang, H., "Research on Integration Method of MBSE and MDO in Automotive Powertrain Design," *SAE Technical Paper*, 2020. https://doi.org/10.4271/2020-01-0422.

[26] Chang, S., Kim, J., and Park, G., "Thermal Control System Design Optimization for Spacecraft Based on MBSE Model and PSO Algorithm," *Aerospace Science and Technology*, Vol. 87, 2019, pp. 176–184. https://doi.org/10.1016/j.ast.2019.01.028.

[27] Khodaparast, H. H., Siad, A. R., and Mirmohammad Sadeghi, S., "Multidisciplinary design optimization of a space exploration rover using Model-Based Systems Engineering approach," *Measurement*, Vol. 178, 2021, pp. 108–118. https://doi.org/10.1016/j.measurement.2021.109056.

[28] Rasti, H., Shahmohamadi, H., and Yazdi, A., "Integration of Model-Based Systems Engineering, Multidisciplinary Design Optimization and Finite Element Analysis to Optimize the Design of Wind Turbine Gearboxes," *Renewable Energy*, Vol. 174, 2021, pp. 1087–1098. https://doi.org/10.1016/j.renene.2021.04.016.

[29] Habermehl, C., Höpfner, G., Berroth, J., Neumann, S., and Jacobs, G., "Optimization Workflows for Linking Model-Based Systems Engineering (MBSE) and Multidisciplinary Analysis and Optimization (MDAO)," *Applied Sciences*, Vol. 12, No. 11, 2022, p. 5316. https://doi.org/10.3390/app12115316.

[30] Bruggeman, A.-L., van Manen, B., van der Laan, T., van den Berg, T., and La Rocca, G., "An MBSE-Based Requirement Verification Framework to support the MDAO Process," *AIAA AVIATION 2022 Forum*, 2022, p. 3722. https://doi.org/10.2514/6.2022-3722.

[31] Kleiner, S., and Kramer, C., "Smart Product Engineering," *Model Based Design with Systems Engineering Based on RFLP Using*, Vol. 6, 2013, pp. 93–102. https://doi.org/10.1260/1745-0855.6.2.93.

[32] Baughey, K., "Functional and logical structures: a systems engineering approach," Tech. rep., SAE Technical Paper, 2011. https://doi.org/10.4271/2011-01-2562.

[33] Xie, K., Zhang, L., Li, X., Gu, P., and Chen, Z., "SES-X: A MBSE Methodology Based on SES/MB and X Language," *Information*, Vol. 14, No. 1, 2022, p. 23. https://doi.org/10.3390/info14010023.

[34] Russell, M., "Using MBSE to enhance system design decision making," *Procedia Computer Science*, Vol. 8, 2012, pp. 188–193. https://doi.org/10.1016/j.procs.2012.01.031.

[35] Estefan, J., "Model-based systems engineering," *INCOSE International Symposium*, Vol. 18, No. 1, 2008, pp. 615–627. https://doi.org/10.1002/j.2334-5837.2008.tb00456.x.

[36] Kleiner, S., *Föderatives Informationsmodell zur Systemintegration für die Entwicklung mechatronischer Produkte*, Shaker, 2003.

[37] Li, T., Lockett, H., and Lawson, C., "Using requirement-functional-logical-physical models to support early assembly process planning for complex aircraft systems integration," *Journal of Manufacturing Systems*, Vol. 54, 2020, pp. 242–257. https://doi.org/10.1016/j.jmsy.2020.03.004.

[38] Gavsevic, D., Djuric, D., and Devedvzic, V., *Model driven engineering and ontology development*, Springer Science & Business Media, 2009. https://doi.org/10.1007/978-3-642-10871-6.

[39] Martin, J. N., *Systems engineering guidebook: A process for developing systems and products*, CRC press, 2020. https://doi.org/10.1201/9780429268781.

[40] Fuchs, J., "Multi-disciplinary MBSE Approach in industrial phases," *Infotech@ Aerospace 2012*, 2012, p. 2532. https://doi.org/10.2514/6.2012-2532.

[41] Glinski, S., Fazal, B., Harrison, E. D., Bendarkar, M. V., Fields, T., Garcia, E., and Mavris, D. N., "An MBSE Framework to Identify Regulatory Gaps for Electrified Transport Aircraft," *2022 IEEE Transportation Electrification Conference & Expo (ITEC)*, IEEE, 2022, pp. 772–777. https://doi.org/10.1109/ITEC51664.2022.308675.

[42] Bleu-Laine, M.-H., Bendarkar, M. V., Xie, J., Briceno, S. I., and Mavris, D. N., "A model-based system engineering approach to normal category airplane airworthiness certification," *AIAA Aviation 2019 Forum*, 2019, p. 3344. https://doi.org/10.2514/6.2019-3344.

[43] Sobieszczanski-Sobieski, J., *Multidisciplinary Design Optimization: An Emerging New Engineering Discipline*, Springer, 1995. https://doi.org/10.1007/978-1-4615-2107-4.

[44] Altus, S. S., Kroo, I. M., and Gage, P. J., "A genetic algorithm for scheduling and decomposition of multidisciplinary design problems," *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 17162, American Society of Mechanical Engineers, 1995, pp. 157–164. https://doi.org/10.1115/DETC1995-1579.

[45] Giesing, J., and Barthelemy, J.-F., "A summary of industry MDO applications and needs," *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1998, p. 4737. https://doi.org/https://doi.org/10.2514/6.1998-47370.

[46] Martins, J. R., and Hwang, J. T., "Review and unification of methods for computing derivatives of multidisciplinary computational models," *AIAA journal*, Vol. 51, No. 11, 2013, pp. 2582–2599. https://doi.org/10.2514/1.J051289.

[47] Agte, J., De Weck, O., Sobieszczanski-Sobieski, J., Arendsen, P., Morris, A., and Spieck, M., "MDO: assessment and direction for advancement—an opinion of one international group," *Structural and Multidisciplinary Optimization*, Vol. 40, 2010, pp. 17–33. https://doi.org/10.1007/s00158-009-0457-9.

[48] Bermell-Garcia, P., Verhagen, W. J., Astwood, S., Krishnamurthy, K., Johnson, J. L., Ruiz, D., Scott, G., and Curran, R., "A framework for management of Knowledge-Based Engineering applications as software services: Enabling personalization and codification," *Advanced Engineering Informatics*, Vol. 26, No. 2, 2012, pp. 219–230. https://doi.org/10.1016/j.aei.2011.09.005.

[49] Heath, C., and Gray, J., "OpenMDAO: framework for flexible multidisciplinary design, analysis and optimization methods," *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, 2012, p. 1673. https://doi.org/10.2514/6.2012-1673.

[50] Bartholomew, P., "The role of MDO within aerospace design and progress towards an MDO capability," *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1998, p. 4705. https://doi.org/https://doi.org/10.2514/6.1998-4705.

[51] Koch, P. N., Simpson, T. W., Allen, J. K., and Mistree, F., "Statistical approximations for multidisciplinary design optimization: the problem of size," *Journal of aircraft*, Vol. 36, No. 1, 1999, pp. 275–286. https://doi.org/10.2514/2.2546.

[52] Gandarillas, V., Joshy, A. J., Sperry, M. Z., Ivanov, A. K., and Hwang, J. T., "A graph-based methodology for constructing computational models that automates adjoint-based sensitivity analysis," *Structural and Multidisciplinary Optimization*, 2022. https://doi.org/10.1007/s00158-022-03139-w.

[53] Sarojini, D., Ruh, M. L., Joshy, A. J., Yan, J., Ivanov, A. K., Scotzniovsky, L., Fletcher, A. H., Orndorff, N. C., Sperry, M., Gandarillas, V. E., et al., "Large-Scale Multidisciplinary Design Optimization of an eVTOL Aircraft using Comprehensive Analysis," *AIAA SCITECH 2023 Forum*, 2023, p. 0146. https://doi.org/https://doi.org/10.2514/6.2023-0146.

[54] Orndorff, N. C., Sarojini, D., Scotzniovsky, L., Gill, H., Lee, S., Cheng, Z., Zhao, S., Mi, C., and Hwang, J. T., "Air-taxi transition trajectory optimization with physics-based models," *AIAA SCITECH 2023 Forum*, 2023, p. 0324. https://doi.org/10.2514/6.2023-0324.

[55] Gandarillas, V., and Hwang, J. T., "TALOS: A toolbox for spacecraft conceptual design," *arXiv preprint arXiv:2303.14936*, 2023. https://doi.org/10.1016/j.actaastro.2023.04.006.

[56] Lin, J.-T., Girerd, C., Yan, J., Hwang, J. T., and Morimoto, T. K., "A Generalized Framework for Concentric Tube Robot Design Using Gradient-Based Optimization," *IEEE Transactions on Robotics*, Vol. 38, No. 6, 2022, pp. 3774–3791. https://doi.org/10.1109/TRO.2022.3141616.

[57] Raymer, D., *Aircraft design: a conceptual approach*, American Institute of Aeronautics and Astronautics, Inc., 2012. https://doi.org/10.2514/4.862873.

[58] Roskam, J., *Airplane design: Part I: Preliminary sizing of airplanes*, Darcorporation, 2003. https://doi.org/10.1016/B978-1-932613-67-6.X5000-5.

[59] Torenbeek, E., *Synthesis of Subsonic Airplane Design: An Introduction to the Preliminary Design of Subsonic General Aviation and Transport Aircraft, with Emphasis on Layout, Aerodynamic Design, Propulsion and Performance*, Springer Science & Business Media, 2013. https://doi.org/10.1007/978-94-007-5599-9.

[60] Loftin, L. K., *Quest for performance: The evolution of modern aircraft*, 468, Scientific and Technical Information Branch, National Aeronautics and Space Administration, 1985. https://doi.org/10.1016/0010-4485(86)90110-1.

[61] NASA, "FLOPS Manual," , 1997. https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19970023805.pdf.

[62] Lukaczyk, T. W., Wendorff, A. D., Colonno, M., Economon, T. D., Alonso, J. J., Orra, T. H., and Ilario, C., "SUAVE: an open-source environment for multi-fidelity conceptual vehicle design," *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2015, p. 3087. https://doi.org/10.2514/6.2015-3087.

[63] Chakraborty, R., Roy, S., and Rao, K. D., "Generalized Uniform Sampling in Fourier Domain Optical Coherence Tomography," *IEEE Journal of Selected Topics in Quantum Electronics*, Vol. 27, No. 4, 2021, pp. 1–7. https://doi.org/10.1109/JSTQE.2021.3077606.

[64] Chakraborty, R., Gupta, K., Guha, S., and Raghunathan, A., "Design Peace: Towards a Unified Approach for Safety and Security Requirements in Cyber-Physical Systems," *arXiv preprint arXiv:2203.06590*, 2022.

[65] Watkins, C. B., Varghese, J., Knight, M., Petteys, B., and Ross, J., "System architecture modeling for electronic systems using mathworks system composer and simulink," *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, IEEE, 2020, pp. 1–10. https://doi.org/10.1109/DASC50956.2020.9268463.

[66] MathWorks, T., "Matlab," *The MathWorks, Natick, MA*, Vol. 176, 2004. https://doi.org/10.1109/MCSE.2004.1299632.

[67] Documentation, M., "The MathWorks Inc," *Natick, MA*, 2005. https://doi.org/10.1109/MCSE.2005.23.

[68] Matlab, M., "MATLAB documentation," , 2018. https://doi.org/10.1007/s13398-014-0173-7.