European Union Aviation Safety Agency

This project has received funding from the European Union's Horizon Europe Programme

RESEARCH PROJECT EASA.2022.C25

D-3.1 REPORT ON THE MAIN CHANGES REQUIRED TO REGULATORY MATERIAL AND STANDARDS

MODEL-SI

Digital Transformation - Case Studies for Aviation Safety Standards - Modelling and Simulation

An Agency of the European Union

Disclaimer

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Union Aviation Safety Agency (EASA). Neither the European Union nor EASA can be held responsible for them.

DELIVERABLE NUMBER AND TITLE:	MODEL-SI, D-3.1 Report on the main changes required to regulatory materials and standards
CONTRACT NUMBER:	EASA.2022.C25
CONTRACTOR / AUTHOR:	ZHAW/Marcello Righi
IPR OWNER:	European Union Aviation Safety Agency
DISTRIBUTION:	Public

REV #	DATE	AUTHORS	REVIEWER
1	06.11.2024	Andrea Pedrioli, Andrea Vaiuso, Marcello Righi	Marcello Righi

DATE: 06 November 2024

SUMMARY

This report is a deliverable document labelled D-3.1 about the “Main changes required to regulatory materials and standards” of the research project number EASA.2022.C25 named MODEL-SI (Digital Transformation - Case Studies for Aviation Safety Standards - Modelling and Simulation).

Nowadays, Artificial Intelligence (AI) applications have great potential to be used in many areas of the aerospace industry. In addition to new applications, such as the ones related to computer vision, it is noted that in some more traditional areas, such as Computational Fluid Dynamics (CFD), it could possibly reduce the workload and computational costs. Independent of the specific application, the reliability and trustworthiness of AI outputs need to be carefully assessed, before being used in its operational domain. To achieve this, a thorough analysis of current regulatory frameworks is crucial. This will help us understand how AI systems can be reliably integrated into certification activities. In cases where existing regulations fall short, the development of new guidelines may be necessary to ensure the safe and responsible implementation of AI in aerospace. Traditional deterministic approaches, governed by aviation authorities Means of Compliance (MOC), face limitations in assessing AI systems due to AI’s inherent unpredictability, data dependency, and need for continuous lifecycle monitoring.

This study evaluates the challenges AI introduces in the aerospace domain, including transparency, data management, risk assessment, and validation, focusing on the existing regulatory gaps and offering solutions through test and validation frameworks. This work examines initiatives like the EU AI Act [1] and ISO standards [2,3,4,5], analysing their approaches and suggesting AI-specific development guidance. These include specialized testing protocols for Machine Learning (ML)-based simulations, designed to ensure that data-driven systems adhere to stringent safety requirements. In addition, we performed an analysis of the currently available regulatory materials, standards and MOC identified in our literature review [6]. Our investigation into new certification methods is also based on three key documents. NASA proposes a "Certification by Analysis" for airplanes and engines [7], in which they show a reduction in testing costs while maintaining safety through advanced numerical models and robust uncertainty analysis. Another remarkable project named "Rotorcraft Certification by Simulation" [8] aims to shorten the certification process for innovative vehicles using flight models and simulators. Lastly, the EASA Concept Paper on AI/ML applications [9] offers guidance on integrating these new technologies into the certification process. These documents provide valuable insights for developing new, efficient, and safe certification methods. In addition, the certification requirements related to novel electric Vertical Take-Off and Landing (eVTOL), namely EASA CS SC-VTOL, are reviewed in order to identify possible gaps and create specific use cases.

A Digital Twin (DT) use case illustrates the importance of AI-specific protocols for applications such as flight simulation, highlighting the need for an expanded MOC approach that ensures robust, reliable, and interpretable AI models. The findings advocate for updated standards to bridge current gaps, emphasizing the importance of transparency, interpretability, and data integrity in AI-aided aerospace applications.

CONTENTS

ABBREVIATIONS

ACRONYM	DESCRIPTION
AI	Artificial Intelligence
AOA	Angle Of Attack
AOS	Angle Of Sideslip
BNN	Bayesian Neural Networks
CFD	Computational Fluid Dynamics
ConOps	Concept of Operations
CS	Certification Specification
DT	Digital Twin
EASA	European Union Aviation Safety Agency
eVTOL	Electric Vertical Take-Off and Landing
FEM	Finite Element Method
FCC	Flight Control Computer
FCS	Flight Control System
FL	Front Left
FSM	Flight Simulation Model
FR	Front Right
GDPR	General Data Protection Regulation
GP	Gaussian Process
HF	High-Fidelity
KL	Kullback-Leibler
ISO	International Organization for Standardization
LF	Low-Fidelity
MAE	Mean Absolute Error
MF	Mid-Fidelity
MOC	Means of Compliance
MSE	Mean Squared Error
ML	Machine Learning
NN	Neural Network
NVC	Normalized Variability Coefficient
ODD	Operational Design Domain
PP	Pusher Propeller
RL	Rear Left

ROM	Reduced Order Model
RMSE	Root Mean Squared Error
RPM	Round Per Minute
RR	Rear Right
TL	Transfer Learning
VLM	Vortex Lattice Method
VTOL	Vertical Take-Off and Landing
WP	Work Package
W&B	Weight and Balance

1. Introduction

The increasing adoption of Artificial Intelligence (AI) across safety-critical industries, such as aerospace, has created a demand for revised regulatory frameworks capable of addressing AI-specific risks and operational characteristics. Unlike traditional software, AI systems utilize Machine Learning (ML) based models that introduce challenges including probabilistic behaviour, data dependency, and complex decision-making processes. These factors necessitate continuous model evaluation and adaptation, marking a divergence from the deterministic nature of traditional aerospace systems.

In the European aerospace regulatory landscape, EASA's MOCs provide a structured approach for assessing deterministic systems, ensuring operational predictability and compliance through stringent testing protocols. However, AI is based on a probabilistic approach, relies on vast datasets and needs for ongoing validation extend beyond MOCs current specifications. This paper analyses AI regulatory gaps within EASA frameworks, with a particular focus on transparency, explainability, and continuous data management, critical for flight safety and reliability. We explore initiatives such as the EU AI Act [1] and ISO standards [2,3,4,5], together with other remarkable works conducted by NASA on "Certification by Analysis" [7], European researchers on "Rotorcraft Certification by Simulation" [8] and EASA on a guidance on ML applications [9]. Comparing their methodologies and proposing AI-tailored lifecycle processes, including advanced testing protocols for ML-based simulations, to ensure AI-driven systems meet rigorous safety standards.

Chapter 2 explores the primary regulatory challenges presented by AI in safety-critical applications and outlines gaps in existing MOCs. It discusses how MOCs, originally designed for deterministic systems, are limited in addressing the unique demands of ML-based models with probabilistic behaviours. Key regulatory themes such as transparency, model robustness, and lifecycle monitoring are introduced, with identified gaps that restrict MOCs applicability to AI solutions in aerospace.

Chapter 3 examines the use of ML-based models, or so-called Digital Twins (DT) in aerospace, specifically when used for eVTOL flight simulations, and highlights the regulatory gaps for ML-integrated applications. Through a detailed gap analysis, it assesses requirements like data quality, explainability, model robustness, and continuous monitoring. The chapter demonstrates the challenges in certifying ML-based system and proposes a framework for AI development and validation tests in certification. By pinpointing these gaps, it suggests areas where new guidelines could better accommodate the innovative capabilities of DT in enhancing aviation safety and efficiency. It provides an example based on the ML-based model developed in the deliverable D-2.1 [13]. Classical Low-Fidelity (LF) model fails to capture complex aerodynamic phenomena which distinguish eVTOL operation from standard rotorcrafts and airplanes. The use of higher-fidelity approaches is complex and requires a considerable amount of computational power. Therefore, the goal was to develop an aerodynamic ML-based model of all five eVTOL rotors, capturing the interaction between rotor-wing-rotor at a reasonable cost.

Chapter 4 conducts a gap analysis for the SC-VTOL 2245 Aeroelasticity use case, assessing the readiness of existing regulations to accommodate AI-driven aeroelasticity analysis in VTOL aircraft. Aeroelasticity, a critical factor in aircraft design and safety, involves the interaction between aerodynamic forces and structural elasticity. This chapter highlights where current standards are inadequate in addressing the use of AI in MOC SC-VTOL 2245 and proposes areas for regulatory improvement. The analysis focuses on ensuring that AI applications meet safety and performance requirements, addressing any unique regulatory needs for SC-VTOL technologies.

2. Overview of AI Regulation and Comparison with EASA MOCs

The rise of AI technologies across various industries has led to a growing need for specific regulations to manage the risks and ensure the safe deployment of AI systems. Unlike traditional software systems, AI introduces challenges such as unpredictability, data dependency, and non-deterministic behaviour, which require tailored approaches for assessment and validation. These challenges have led regulatory bodies worldwide to develop frameworks and standards aimed at governing the development, deployment, and certification of AI systems.

In the aerospace sector, the safety-critical nature of operations means that the integration of AI must be carefully controlled. AI applications, including those used for simulation, control systems, and data analysis, must meet stringent safety, reliability, and performance requirements. The European Union, recognizing the unique risks posed by AI, has introduced initiatives like the EU AI Act [1], which aims to classify AI systems based on risk levels and impose appropriate regulatory requirements. At the international level, standards organizations such as ISO/IEC have also been active in developing guidelines for AI. ISO/IEC JTC 1/SC 42 [10] focuses on standardizing AI practices across different industries, covering aspects like data quality, model robustness, and transparency. For instance, ISO/IEC 24029 [3] provides guidelines for evaluating the robustness of ML models, which is crucial for safety-critical applications like those in aerospace. Such efforts highlight the need for regulations that go beyond traditional safety standards. While traditional certification methods, such as EASA's MOCs, are designed for deterministic systems where behaviours can be precisely predicted and validated, AI systems often involve probabilistic models and data-driven decision-making. This introduces new dimensions of risk, including bias, interpretability challenges, and the need for continuous monitoring throughout the system's lifecycle.

2.1 High-level Gaps Between AI Regulations and EASA MOCs

Traditional EASA MOCs are primarily designed for deterministic systems, where behaviour can be precisely predicted and verified through well-defined rules, equations, and physical principles. These systems allow for clear, repeatable testing procedures, such as physical tests and simulations, which yield consistent results under controlled conditions. This predictability ensures compliance with safety standards, making it possible to conduct thorough and reliable testing for conventional aircraft components.

In contrast, AI systems, particularly those involving ML, operate on a fundamentally different basis. AI models make decisions by identifying patterns in large datasets, leading to variability in their performance. The model's accuracy is influenced by the quality and diversity of the data on which it is trained, as well as changes in the operational environment. For instance, an AI model trained to predict flight dynamics may perform well under typical conditions but could struggle with scenarios that were not adequately represented in the training data. This non-deterministic behaviour introduces new challenges for assessment and certification, as traditional testing methods may not adequately address these variations. The current MOCs lack provisions for such assessments, indicating a need for expanded protocols that address the unique challenges of AI systems.

The need for specific AI regulations has become evident with the emergence of these challenges. AI regulations, such as those outlined in the EU AI Act and ISO standards, introduce requirements for transparency, explainability, data quality, and continuous monitoring—factors that are less emphasized in traditional MOCs. For example, AI systems must be designed to include mechanisms that explain their decision-making processes, ensuring that outputs are interpretable by human operators. This contrasts with deterministic systems, where the system's functioning is inherently understandable and does not require additional explainability provisions. This is due to the fact that AI models can function as "black boxes," producing decisions based on complex patterns that are not easily interpretable. This lack of transparency can pose significant safety concerns, especially in critical applications where unexpected behavior can have severe consequences. Current MOCs do not explicitly require explainability, creating a gap in how AI systems should be developed, assessed, and certified.

Another significant distinction is the emphasis on data management and lifecycle monitoring. AI regulations focus on rigorous data handling practices, including data integrity, bias mitigation, and ongoing updates to the model. This continuous adaptation is crucial for AI systems, which must remain effective as new data is integrated. For example, an AI-powered DT for flight simulation must be retrained and revalidated as new flight data becomes available. Requirements for data quality management and such an ongoing requirement are not covered by existing MOCs.

Conventional MOCs, however, generally focus on validating system performance at a specific point in time, without accounting for changes that may occur throughout the system's operational lifecycle. Consequently, AI systems need ongoing validation and verification processes, unlike the one-time certification events typically sufficient for deterministic systems.

Furthermore, AI regulations often require robust risk assessment and mitigation strategies tailored to the specific characteristics of AI. Traditional risk assessments mainly focus on physical and mechanical risks, such as system failures due to hardware defects. In contrast, AI introduces new categories of risk, including algorithmic bias, data vulnerabilities, and unpredictable behaviour under certain conditions. Regulatory frameworks like ISO SC 42 [10] provide guidelines for thorough risk assessments that include these factors, emphasizing the need for safeguards to mitigate potential harm. However, current MOCs do not fully encompass these aspects, underscoring the need for expanded risk assessment protocols when certifying AI systems.

To recap, there is a need for clear, sector-specific guidelines that address AI in safety-critical sectors like aerospace. Initiatives like ISO PAS 8800 [11] for road vehicles illustrate the benefits of tailored standards for guiding the development and deployment of AI technologies. Similar efforts are necessary for aerospace, where specific standards can help integrate AI solutions seamlessly into existing certification processes, ensuring that safety, reliability, and accountability measures are met from the outset.

2.2 Bridging the Gap

The integration of AI technologies into safety-critical sectors like aerospace requires a shift in how certification frameworks approach software-based systems, particularly those utilizing ML. Unlike traditional hardware, which undergoes static testing and certification, AI models are dynamic software components that evolve with new data and learning. Addressing this difference calls for the adoption of robust lifecycle management processes, which are currently underdeveloped in existing EASA MOCs.

This fundamental difference necessitates a standard lifecycle that emphasizes continuous development, validation, and adaptation. ISO standards, such as ISO/IE 5338 [12], stress the importance of having established and standardized processes for the development, deployment, and maintenance of AI models. These processes need to be documented rigorously to ensure transparency, reliability, and safety across the entire AI lifecycle.

Currently, EASA's MOC does not provide explicit guidelines for technical documentation of the AI/ML lifecycle. This creates a gap, compared to the UE AI Act, where technical documentation is one of the most emphasized requirements and objectives for high-risk systems.

To bridge this gap, there is an opportunity for EASA to integrate the AI lifecycle management principles laid out by international standards bodies. In particular, EASA Concept Paper: Guidance for Level 1 & 2 Machine Learning Applications [9] provides very interesting objectives that are also well-aligned with the standard AI/ML lifecycle. Several processes and their corresponding documentation should be thus presented in an MOC to close this gap. Section 3.1 details the process needed throughout the AI lifecycle along with their corresponding documentation.

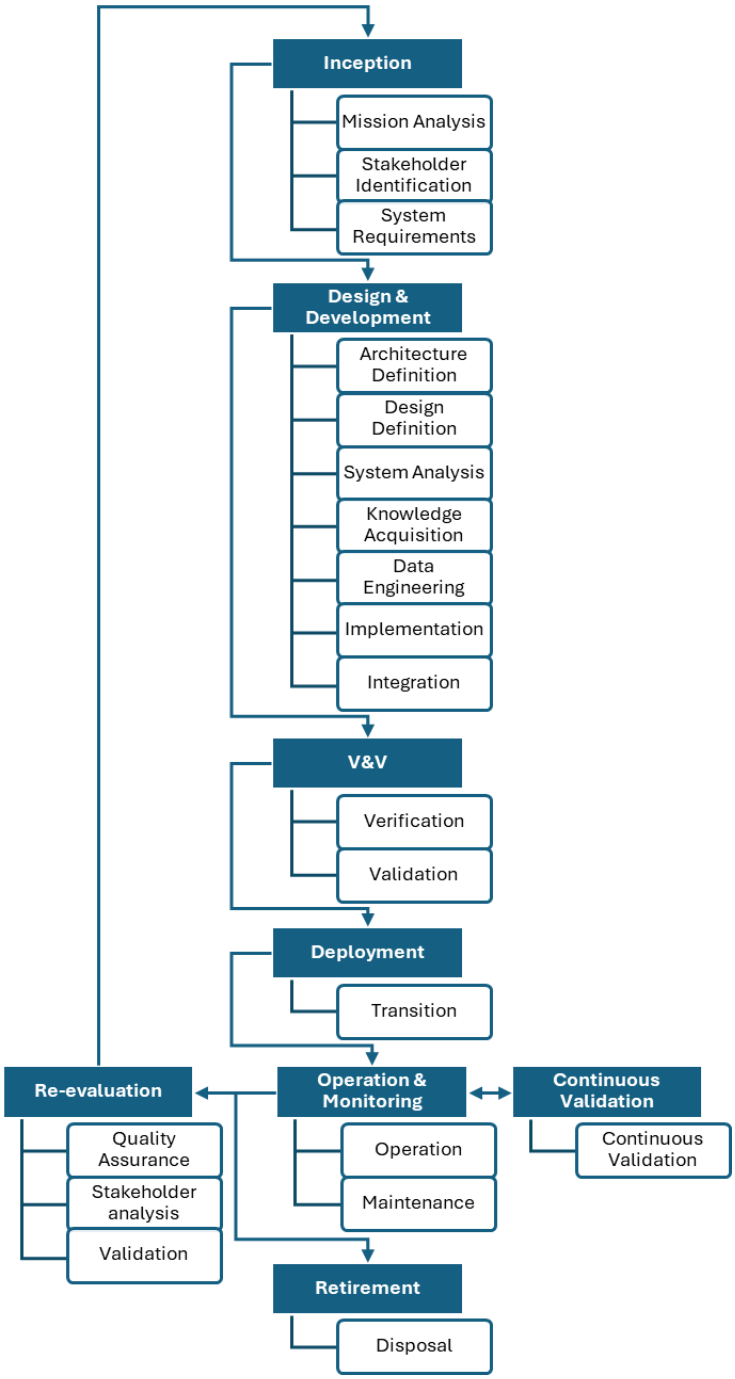

Figure 1:AI lifecycle from ISO/IEC 5338 [12].

Another crucial area for bridging the gap between AI systems and existing MOCs is the adaptation of testing methodologies. Traditional tests in EASA MOCs are designed for deterministic software and hardware-based systems, where performance is predictable and repeatable. In contrast, ML models require a different set of tests, focused on aspects like data dependency and robustness. The following are key generic tests to be considered in the MOCs outlining the overall Verification and Validation (V&V) plan. Such a plan can be considered during the implementation, assessment, and certification of ML-based solutions. While items specified in the V&V plan do not directly correspond to any specific parts of the current EASA MOCs, their role in verifying the safety, robustness, and reliability of AI systems is critical. Such verifications and validations are not requested by the current MOCs. By adding the following, we can address the existing gap:

- **Bias and Variance Analysis**: Iterative learning process evaluations are necessary to demonstrate consistent performance across various subsets of training data, indicating that the model is not overly reliant on specific data segments. These segments can be related to different scenarios and conditions.

The model's bias and variance should align with defined learning process management requirements, ensuring that performance stays within acceptable limits.

- **Performance Evaluation on Test Data**: A dedicated dataset, never used during model training and validation, should be used to evaluate the model. This ensures that the model can generalize its learning to unseen data, a key factor for reliable performance in diverse operational environments.
- **Requirements-Based Verification**: The model's behaviour should be verified against specified requirements (e.g., accuracy and robustness KPIs), ensuring it meets the operational criteria necessary for safe deployment. This involves documenting behavioural conditions, such as distributions of absolute errors across sequences of data frames, in detailed verification reports.
- **Learning Algorithm Stability Analysis**: Performance metrics should be analyzed over the course of training to detect and mitigate unwanted behaviours, such as overfitting or large oscillations, which could hinder the model's ability to generalize effectively to new data.
- **Model Stability and Robustness Verification**: The Model's stability should be evaluated under standardized conditions, ensuring it consistently retains learned information. Robustness information should be verified when the trained model is exposed to adverse conditions, such as environmental variability, to guarantee dependable performance under diverse operational settings.
- **Sensitivity Analysis for Error Propagation**: Sensitivity analysis is necessary to understand how errors at the model level might affect other system components. This analysis is crucial for quantifying the impact of potential errors, enabling proactive measures to mitigate adverse effects.
- **Generalization Boundaries Verification**: The model's ability to generalize should be verified by evaluating it on various types of data, including training, validation, and test sets, as well as scenarios with previously unseen conditions. This ensures that the model can accurately adapt to both familiar and novel situations, a critical factor for safety-critical applications like flight simulation.

Although all AI/ML systems from inception to retirement go through the same lifecycle, the V&V should include application-specific tests, measures, and metrics. At the application level, the current MOCs specify tests required for certification of a VTOL. For instance, the hovering test at different weights, altitudes, and temperatures as well as unsteady responses and maneuverability of the test are elaborated in the current MOCs. However, these tests in the ML context and data-driven models require different setups. Especially, concerning the DT use case, creating scenarios corresponding to different test conditions needs to be addressed. Section 3.2 goes through application-level tests. Implementation guidance is then further detailed in Annex B .

In addition to these tests that should be addressed by EASA, measures and metrics required to interpret them should also be clarified. As a result of the probabilistic and stochastic nature of ML models, specific measures and metrics are required to validate the test results rather than the more conventional ones. For instance, the result of ML testing can be demonstrated by Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared, etc. These statistical performance metrics are not mentioned by the MOCs.

3. Gap Analysis on Digital Twin Use case

A gap analysis was conducted on the CS SC-VTOL [14] and MOC SC-VTOL [15] standards to assess their applicability to AI-powered solutions, specifically the Flight Simulation Model (FSM) as a DT for eVTOL simulation. This analysis highlights that, while MOCs provide requirements at the aircraft level, there are currently no specific requirements for implementing AI/ML solutions within simulation models. The MOCs lack both a formalized set of guidelines and a defined test list for evaluating AI-driven solutions, especially those based on ML.

Although MOCs offer some insights into testing protocols, these are predominantly at the aircraft level, focusing on physical components and deterministic systems. They do not extend to the AI and ML lifecycle, reliability, or explainability requirements like the ones in the present DT use case. This results in a significant gap. Hence, the applicants wishing to leverage AI systems, such as a DT, must rely on additional protocols and testing strategies that go beyond traditional MOC requirements. The proposed DT guidance and test lists in this section provide a structured approach for bridging this gap, establishing a reliable framework for AI-driven FSM assessment and certification. In other words, this section outlines a comprehensive testing and analysis plan that should be performed on the DT system, i.e., the FSM developed in WP2. The plan focuses on two key areas:

- DT guidance requirements: this section defines the minimum requirements needed to describe the AI implementation into an FSM. This documentation will serve as a vital reference for understanding the model design, development process, and operational characteristics. It will also provide a clear review for regulatory purposes.
- DT tests list: this part defines a comprehensive list of tests that the FSM needs to undergo. These tests will rigorously evaluate its functionality, performance, and robustness across its intended operational domain. The test list is thought to ensure all critical safety requirements.

3.1 Bridging the Gap with Digital Twin Process Requirements

This section defines the minimum requirements required to describe the design, development, implementation, and deployment of AI systems in the FSM. It can serve as a compliant document, providing a clear understanding of the AI system components, operational process, and limitations. It is based mainly on the objectives defined in the EASA Concept Paper: Guidance for Level 1 & 2 Machine Learning Applications [9]. Similarly to our reference document, the requirements are divided into AI trustworthiness analysis, AI assurance, and AI risk analysis.

3.1.1 AI trustworthiness analysis

The AI trustworthiness analysis outlines the high-level but key steps to ensure the safe and reliable operation of an AI-based system. It begins by identifying the stakeholders and end-users, their roles, responsibilities, and expertise, followed by defining the tasks these users will perform with the system. The AI system itself must be clearly defined, with a documented Concept of Operations (ConOps) that details how tasks are shared between users and the AI, while also noting operational limitations.

A functional analysis is then required to define the system's purpose, high-level functions, and their breakdown into sub-functions, which are then analysed and validated. The system must be classified according to its level of autonomy, ranging from human assistance (Level 1) to full autonomy (Level 3), based on human-AI interaction. Such classification is important as it leads to further specific requirements.

The applicant must ensure compliance with data protection regulations such as General Data Protection Regulation (GDPR) and conduct a transparency analysis to assess how the system explains its outputs, ensuring clarity, consistency, and relevance. This analysis also addresses the timing and rationale for how and when the AI system communicates its decision-making processes.

In what follows, we provide a detailed review of these processes, and on a best-effort basis, we address each for the DT use case.

1.1 End-user identification:

The applicant should identify the list of end-users that are intended to interact with the AI-based system, together with their roles, their responsibilities, and their expected expertise.

Table 1: End-user identification table.

End-user	Role	Responsibilities	Expected Expertise
Chief Engineer	Make overall decisions based on surrogate model predictions	Correctly interpret the AI predictions to make final design, sizing, FCS decisions	Expert on the overall aircraft design
Structural Engineer	Develop the overall aircraft structure	Define and analyze the overall aircraft loads from various sources (aerodynamic, weight, landing gear impact, …). Develop the structure and select materials	Expert on overall aircraft structure development
Flight Control System Engineer	Develop the aircraft FCS	Pilot controls, route planning and autopilot	Expert on the aircraft control system. How the aircraft is controlled by the pilot, which tasks are autonomous, autopilot
Aeroelastic Engineer	Analise the flexible parts of the aircraft	Implement the interaction between the structural, aerodynamic and FCS part	Expert in the modelling interaction between structural, aerodynamic and FCS to avoid harmful behavior like flutter
Compliance Verification Engineer	Independent verification of aircraft certification documents	Ensure compliance with certification regulations and standard	Experts in the specific field and of the aviation regulations and standards to effectively assess compliance

1.2 End-user task

For each end-user, the applicant should identify which high-level task(s) are intended to be performed in interaction with the AI-based system.

Table 2: End-User Task identification.

End-user	High-level task

Chief Engineer	Develop the overall aircraft design, architecture, systems integration, …
Structural Engineer	Perform some simulations to check when the aircraft structure will fail
Flight Control System Engineer	Use the surrogate model to test a new type of FCS or high-level controller
Aeroelastic Engineer	Perform simulation to analyze to analyze flutter, LCO, buffeting, …
Compliance Verification Engineer	Verify that the aircraft (or one of its parts) meets the safety and regulatory requirements before submission for certification approval

1.3 AI system identification

The applicant should determine the AI-based system while considering domain-specific definitions of 'system'.

The system is identified as a FSM which is composed of standard physics-based methods and data-driven models. The AI system is only a part of the FSM. More details are available on D-2.1 [13].

1.4 Operational Design Domain (ODD)

a. Documentation should describe the application domain and clearly define ODD, including environmental conditions, operational scenarios, and system limitations.

b. Documentation should include risk analysis associated with each identified ODD scenario, along with a description of mitigation strategies. The documentation should also include ODD formalization including specific conditions, scenarios, or constraints of the AI/ML system design. Disturbance identification and grading, including their impact on the system's performance, should be part of the documentation.

a. Application Domain and ODD definition

The application domain of the FSM with its AI/ML-based rotor thrust and torque model lies within the context of simulating flight dynamics for eVTOL aircraft across various operational scenarios. The AI/ML model serves to predict rotor thrust and torque values using multi-fidelity data as training and validation datasets, and its predictions feed into the overall flight simulation.

We focus the ODD for an aeroelastic analysis. In this case, the ODD aims on predicting and analysing the interaction between aerodynamic forces and structural dynamics, such as flutter, under various environmental conditions.

The operational environmental conditions can be classified according to:

- Flight conditions: The FSM simulates a range of operating conditions, including:
  - Nominal operational scenarios such take-off and landing, ascent, descent and cruise in HC mode and ascent, descent and cruise in AP mode.
  - Non-nominal conditions such as in presence of wind gusts and turbulence. In these conditions, the FSM also outputs dynamic responses due to unsteady aerodynamics, such as transient

responses to wind gusts and rapid rotor changes, which affect rotor thrust and torque predictions in nominal cases.

- Atmospheric external conditions: The AI/ML model is designed to function within a defined flight envelope, which accounts for variations in altitude, temperature, air pressure, and turbulence.
- Dynamic Response: The model also covers unsteady aerodynamic conditions, such as transient responses to gusts and rapid rotor changes, which affect rotor thrust and torque predictions.

The FSM system's ODD encompasses a variety of operational scenarios, such as:

- Trimmed flight conditions (steady state): such as hover, climb, descent and cruise.
- Transition between flight HC and AP modes and vice versa
- Different manoeuvres, including rapid accelerations and decelerations, turns, etc.
- Non-nominal flight conditions due to atmospheric turbulence, wind gusts, or air density variations.

As described in point "1.6 – Functional Analysis", the Aeroelastic Module performs linear stability analysis and provides the aircraft structure dynamic response. The former analysis employs linearized state-space models to evaluate the aircraft stability margins across the studied condition, while the dynamic response shows comprehensive insights into the structural deformation when subjected to time-varying aerodynamic forces (such in the presence of a wind gust).

Some system limitations are summarized and collected as follow:

- Data Fidelity: The accuracy of the AI/ML predictions relies heavily on the fidelity of the input data. Low-fidelity (LF) data or not sufficient Mid-Fidelity data may introduce regions with greater uncertainties into thrust and torque predictions.
- Operational Boundaries: The AI/ML system is limited to the range of data it has been trained on. Some extreme conditions (e.g., high-speed manoeuvres beyond normal operational limits or unanticipated failure modes) may lead to erroneous results. Such false predictions can be spotted thanks to the uncertainty quantification.
- Rotor and aerodynamic transients: The modelisation of certain time lags associated in particular to high-frequency gusts which directly affects rotor RPM changes is not trivial and more effort should be spent on this topic.

b. Risk analysis and mitigation strategies

The most important risk and mitigation strategies associated to our AI/ML system are tabulated in Table 3.

Table 3: Risk and mitigation strategies associated to our AI/ML system.

Risk identified	Mitigation Strategy
Insufficient training data or data bias: The AI/ML model performance depends on the diversity and quality of the input data. If the training data lacks coverage of critical scenarios or flight conditions, the model may not generalize well to real-world situations.	Comprehensive training dataset: Ensure the AI model is trained on a comprehensive dataset that includes varying fidelity levels and a wide range of operational scenarios. Incorporate training data that represents nominal and extreme flight conditions (e.g., rapid transitions, high-altitude turbulence).
Model Drift: As the AI/ML model is trained on historical simulation data, there is a risk of model drift if new, unrepresented conditions arise (e.g., new aerodynamic configurations or extreme weather conditions), potentially reducing prediction accuracy.	Dataset update and re-training: Periodically update the AI model with new data, ensuring it adapts to changing operational environments or new aircraft configurations. This can help mitigate model drift and maintain performance accuracy.
Overfitting: If the model is overfitted to HF simulations, it may perform poorly on LF inputs or in scenarios it has not been explicitly trained for.	Cross-validation: Implement cross-validation techniques during model training to prevent overfitting.

Data augmentation: it is performed to improve the quality and diversity of the training dataset for better model learning and generalization. For example, this included adding new data points with near-zero conditions, interpolating thrust and torque values using cubic functions, all aimed at enhancing model accuracy and stability in predictions.

1.5 Concept of Operations

The applicant should define and document the Concept of Operations (ConOps) for the AI-based system, including the task allocation pattern between the end-user(s) and the AI-based system. A focus should be put on defining the operational design domain (ODD) and capturing specific operational limitations and assumptions.

The ConOps is related to predictions performed by the FSM for certification purposes: an example could be to show compliance with the requirement SC-VTOL.2245 Aeroelasticity [14]. The proposed MOC refers to MOC VTOL.2245 Aeroelasticity [15].

Here the end-user could be an Aeroelastic Engineer who would provide FSM results to comply with the certification documentation.

For example, in subpoint (b), he should provide some results to show that the *aircraft should be designed to be free from aeroelastic instability for all configurations and design conditions within the aeroelastic stability envelopes.* This will be done with the prediction calculated by our FSM.

To pursue this goal, the end-user will perform the following tasks:

- Initialization of the FSM: initialize all the simulation parameters
- Define the flight simulation test matrix for the determination of the aeroelastic flight envelope
- Trim the aircraft at such conditions
- Linearize the aircraft
- Check the stability by means of eigenvalues analysis
- Determine if the aircraft is flutter-free

1.6 Functional analysis

The applicant should perform a functional analysis of the system:

a. Define the system's purpose
b. Identify high-level functions
c. Break down high-level functions into sub-functions
d. Analyse each sub-function
e. Validate the functions

The functional analysis of our DT model is presented here.

a. System purpose:
   The FSM model is used to simulate flight dynamics, internal and external loads, and operational conditions to assist in certification processes and optimize aircraft design and performance using DT technology.

The Flight Mechanics Module purpose is to provide stability-envelopes, rigid-body stability and control analysis and to test and simulate the aircraft behaviour in several manoeuvres, such as transition from Helicopter to Airplane mode.

The Aeroelastic Module purpose is to identify in which regions the eVTOL can fly without encountering any aero-structural instability, such as flutter.

b. Identification of High-Level Functions

The FSM is split into Flight Mechanics and Aeroelastic Module, here represented in Figure 2.

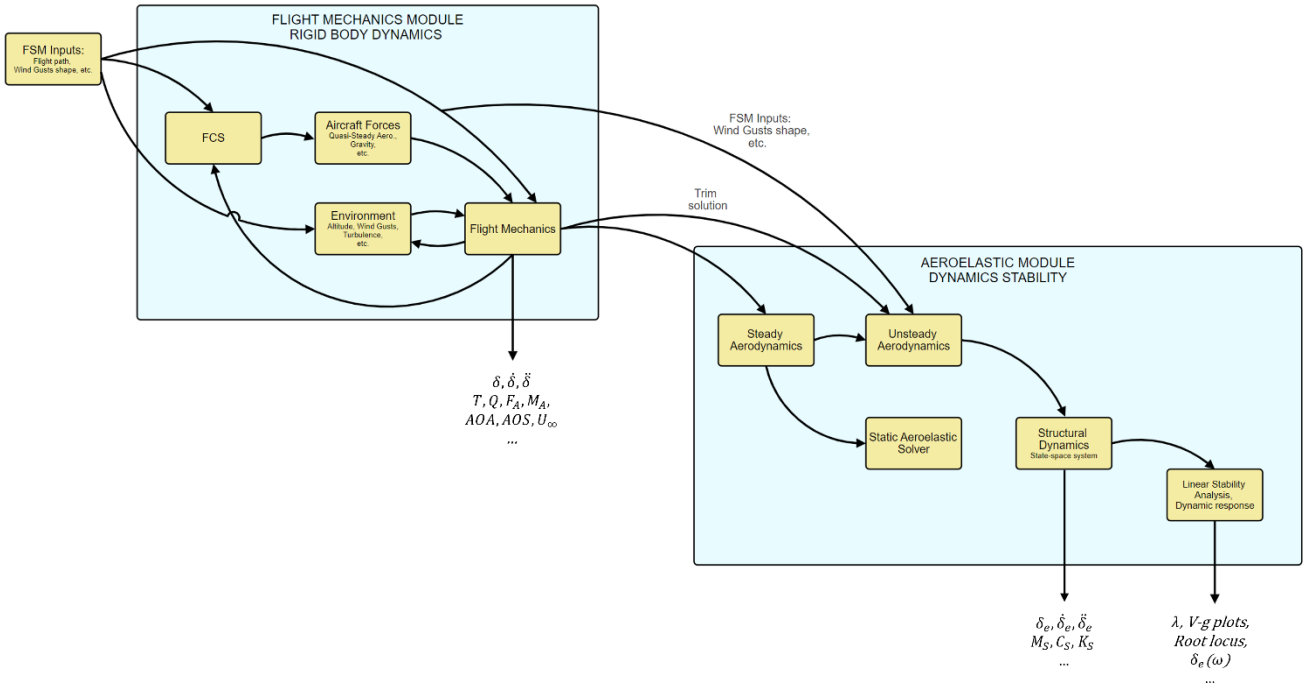

Figure 2: FSM: the main high-level functions are "Flight Mechanics" and "Aeroelastic" modules

The Flight Mechanics high-level functions are depicted in Figure 3 and here reported:

- Flight Control Computer (FCC) function
- Actuators function
- Aircraft function
- Sensor function

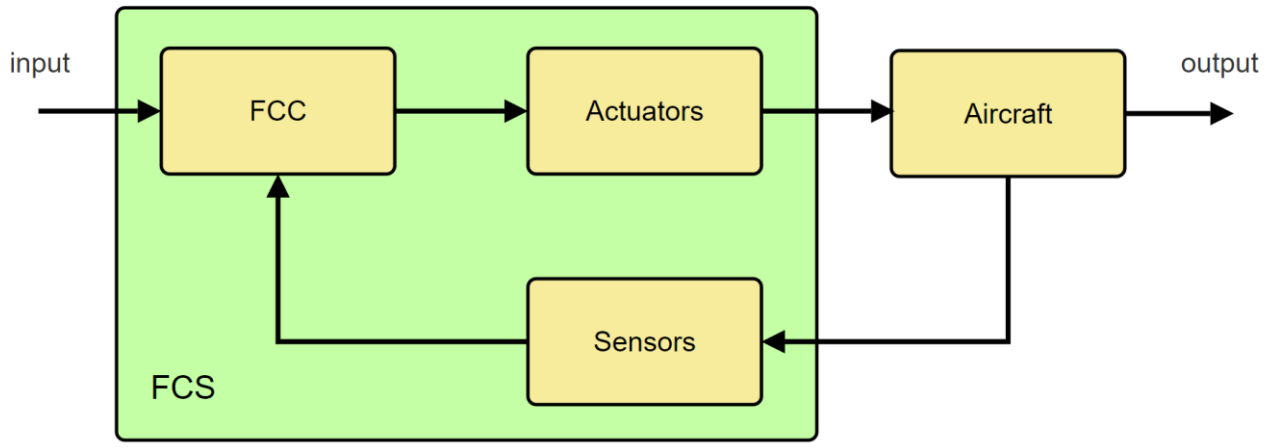

Figure 3: Flight Mechanics Module detailed representation.

The Aeroelastic Module high-level functions are depicted on the right side of Figure 2 and here reported:

- Steady Aerodynamics
- Unsteady Aerodynamics
- Static Aeroelastic Solver
- Structural Dynamics
- Linear Stability Analysis and Dynamic response

c. Break-down of functions into sub-functions

The Flight Mechanics high-level functions are broke-down in the following way (see Figure 2 and Figure 3):

- FCC function:
  - Airplane FCC sub-function
  - Helicopter FCC sub-function
  - Actuators function
  - Ailerons sub-function
- Ruddervators sub-function
- Sensor function:
  - General sensors sub-functions
- Aircraft model:
  - Weight and Balance (W&B) sub-function
  - Gravity sub-function
  - Propulsion sub-function
  - Aerodynamics sub-function
  - Equation of Motion sub-function
  - Environment sub-function

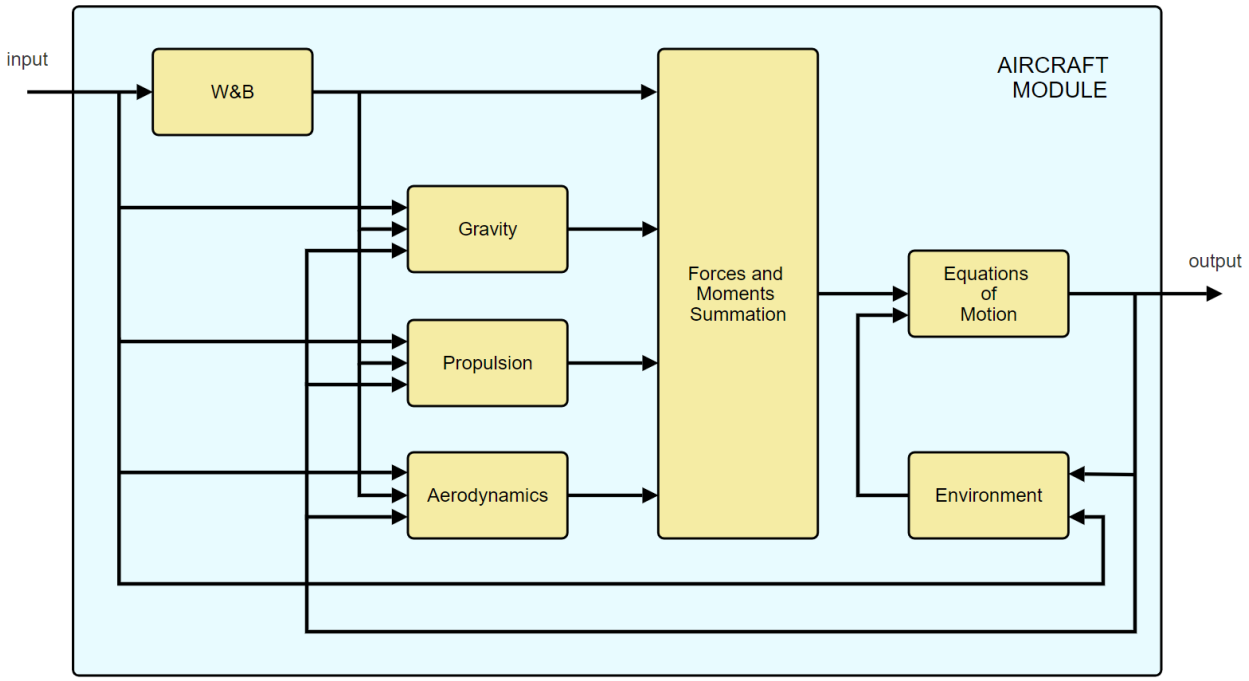

Figure 4: Aircraft model.

The Aeroelastic high-level functions are broke-down in the following way (see Figure 5 and Figure 6):

- Steady Aerodynamics function:
  - LF rotor thrust sub-function
  - MF rotor thrust sub-function
  - HF rotor thrust sub-function

    - Multi-fidelity rotor thrust sub-function
- Unsteady Aerodynamics function
    - Analytical unsteady function
    - Multi-fidelity unsteady response function
    - Unsteady aerodynamic Reduced Order Model (ROM) function
    - Unsteady wing function
- Static Aeroelastic function
- Structural Dynamics function
    - FEM Structure sub-function
    - Model condensation sub-function
    - FEM blades and rotor sub-function
    - Multi-blade coordinates sub-function
    - Structural Dynamic Solver sub-function
- Linear Stability Analysis and Dynamic response

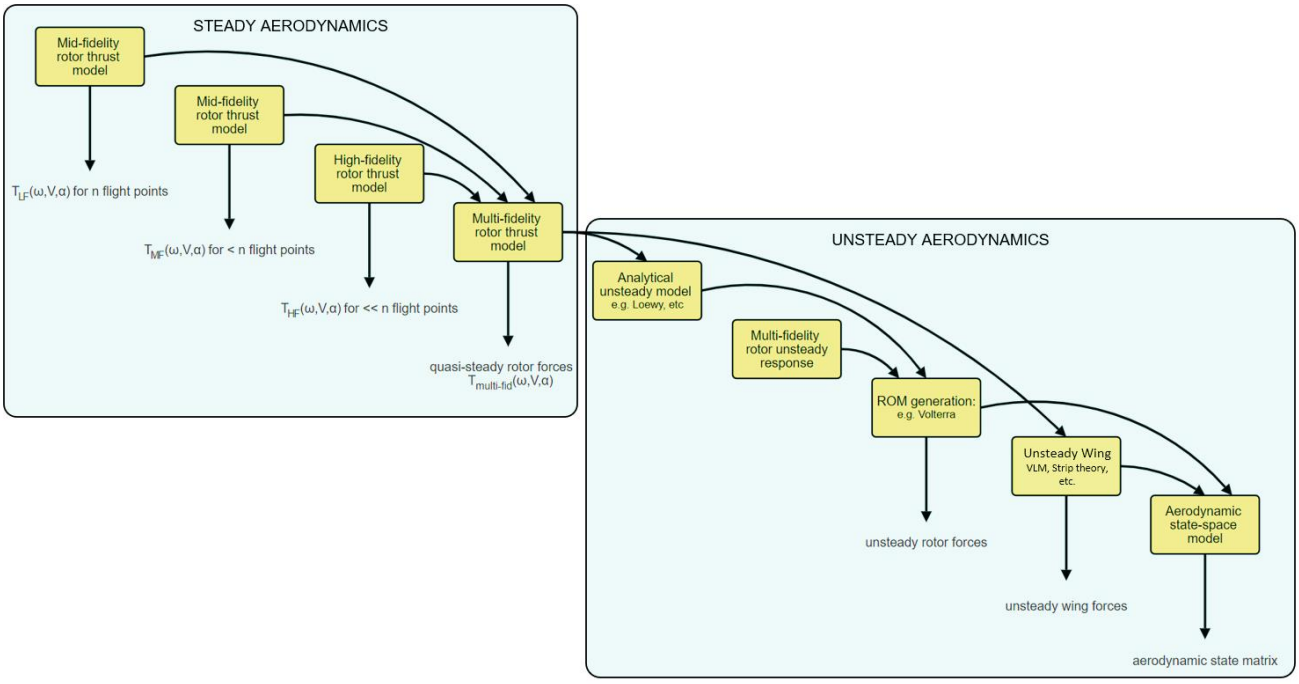

Figure 5: Steady and Unsteady Aerodynamics models.

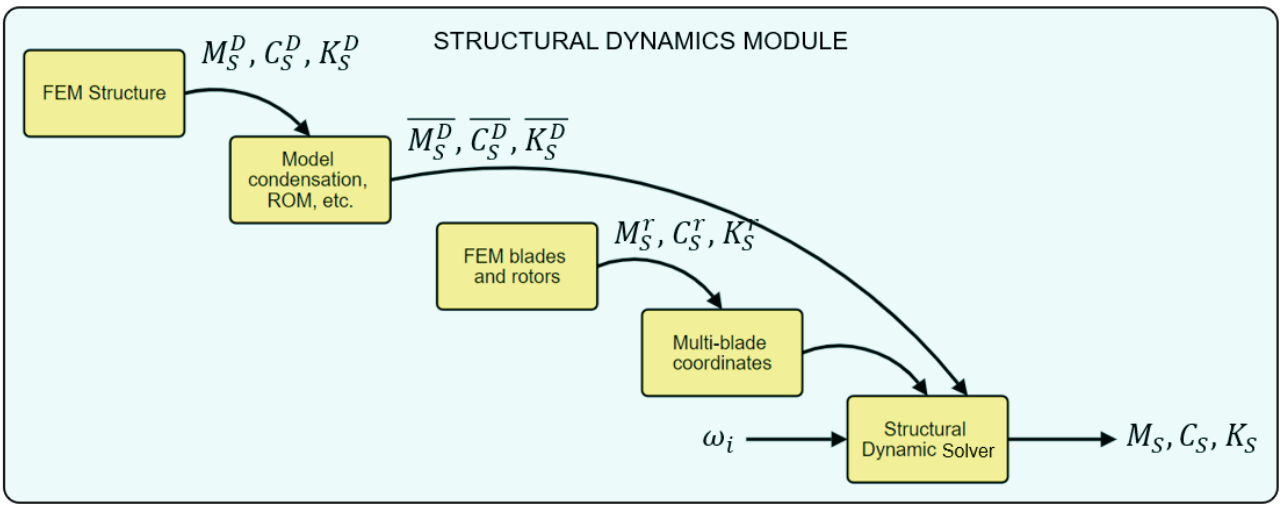

Figure 6: Structural Dynamics Module.

d. Analysis of the sub-functions

The Flight Mechanics high-level function can be analysed by looking at the Aircraft model (see Figure 4). Here is the list of all relevant sub-functions:

- FCC function:
  - Airplane FCC sub-function: cascaded PID control loops responsible to control the eVTOL in airplane model. The sub-function primarily uses the pusher propeller and the aerodynamic surfaces.
  - Helicopter FCC sub-function: cascaded PID control loops responsible to control the eVTOL in helicopter model. The sub-function primarily uses the four lift propellers.
  - Transition FCC sub-function: set of control laws to be used to control the eVTOL during the transition phase from helicopter to airplane and vice versa.
- Actuators function:
  - Ailerons sub-function: modelled as simple low-pass filters. Their bandwidth and rate limits have been chosen to represent the typical characteristics of a scale model actuator.
  - Ruddervators sub-function: modelled as simple low-pass filters. Their bandwidth and rate limits have been chosen to represent the typical characteristics of a scale model actuator.
- Sensor function:
  - General sensors sub-functions: all sensors are modelled as low-pass filters with transport delays. The filter characteristics have been selected to reflect the typical performance of sensors commonly used in this type of application.
- Aircraft function: It represents the core flight mechanics of the vehicle. It simulates key elements such as the equations of motion, aerodynamics, propulsion system, structural behaviour, and atmospheric turbulence. In the LF implementation.
  - Weight and Balance (W&B) sub-function: based on the input data (mainly mass and inertia distribution), it calculates the eVTOL center of gravity and moments of inertia. Depending on the type of aircraft, in particular when moving parts and fuel are present, the overall mass and inertia can be subjected to change, which could change the flight dynamics behaviour the aircraft during flight
  - Gravity sub-function: based on the eVTOL attitude, it calculates the forces and moments generated by the gravity
  - Propulsion sub-function: it provides the forces and moments generated by all the five propellers. The LF model does not account for complex airflow interactions between components. It is eventually replaced by the AI/ML based rotors data-driven model.
  - Aerodynamics sub-function: it provides the forces and moments generated by all the aerodynamic surfaces (wing and tail including control surfaces) and the fuselage. It is modelled using a vortex lattice method (VLM). However, this approach is limited to small angles of attack and sideslip. To account for higher angles of attack, the aerodynamic coefficients were extrapolated using techniques.
  - Forces and moments Summation sub-function: all forces and moments are collected and summarized here.
  - Equation of Motion sub-function: It computes the aircraft's 6-degree-of-freedom equations, controlling both its translational and rotational movements. This sub-function determines the eVTOL position, velocity, and orientation by integrating the forces and moments applied to it, and it updates them throughout the flight simulation.
  - Environment sub-function: It calculates and accounts for all the changes due to altitude, pressure and temperature variations. In addition, it can provide external disturbances due to turbulence (Dryden turbulence model) and wing gusts.

The Aeroelastic functions is presented here below.

- Steady Aerodynamics function:
  - LF rotor thrust sub-function: It calculates the rotors thrust and torque by means of LF methods such BEM and actuator disk model. A simple inflow model is also used.

    - Mid-fidelity (MF) rotor thrust sub-function: It determines the rotors thrust and torque with Vortex Particle Method.
    - HF rotor thrust sub-function: A few experimental and numerical (CFD) points are available to approximate the rotors thrust and torque.
    - Multi-fidelity rotor thrust sub-function: The low-, mid-, and HF sub-functions progressively increase in accuracy and computational complexity, allowing the system to balance computational cost with precision. The multi-fidelity approach integrates these levels, providing predictions in both nominal flight conditions and edge cases, such as high angle of attack (AOA).
- Unsteady Aerodynamics function
    - Analytical unsteady sub-function: it takes advantages from theoretical models like Theodorsen's theory, while multi-fidelity and ROM sub-functions combine low- and HF data for dynamic conditions.
    - Multi-fidelity rotor unsteady response sub-function
    - Unsteady aerodynamic ROM sub-function
    - Unsteady wing sub-function
- Static Aeroelastic function: it provides nonlinear static aeroelastic displacement, calculated from a trimmed condition and subjected to steady aerodynamic forces.
- Structural Dynamics function
    - FEM Structure sub-function: it describes the aircraft structure using FEM technique
    - Model condensation sub-function: it reduces the complexity of the structural model by condensing it into a smaller and most representative number of dynamic modes, rather than considering all the degrees of freedom. This function maintains the accuracy of the simulation by focusing on the most critical vibration modes within the frequency range of interest, thus improving computational efficiency without losing key dynamic behaviour.
    - FEM blades and rotor sub-function: it describes the blades structure using FEM technique
    - Multi-blade coordinates sub-function: it models the periodic motion of individual blades into a fixed reference frame. By transforming the cyclic behaviour of the blades into a steady-state representation, this technique reduces computational complexity, allowing for more efficient analysis of aeroelastic stability and the dynamic response of a rotor.
    - Structural Dynamic Solver sub-function: it solves the standard 2nd order system equation if motions based on the mass, damping and stiffness matrices.
- Linear Stability Analysis and Dynamic response: Stability analysis uses the linearized state-space models to assess the stability margins of the aircraft under various conditions, with the dynamic response function providing detailed feedback on how the aircraft behaves under time-varying aerodynamic forces.

e. Validation of the subfunctions

The validation of the sub-functions is provided only at high-level on D-2.1 [13], where the FSM results are compared against the flight test data or other HF data.

1.7 AI classification

The applicant should classify the AI-based system, based on the levels (1A, 1B, 2A, 2B, 3A, 3B) AI typology and definitions, with adequate justifications.

- **Level 1: Assistance to human**
  - **Level 1A: Human augmentation**
  - **Level 1B: Human cognitive assistance in decision and action selection**
- **Level 2: Human/machine teaming**
  - **Level 2A: Human and AI-based system cooperation**
  - **Level 2B: Human and AI-based system collaboration**
- **Level 3: More autonomous machine**
  - **The AI-based system performs decisions and actions, overridable by a human.**
  - **The AI-based system performs non-overridable decisions and actions.**

The FSM is an AI-based system that leverages AI/ML through regression methods. Our system falls under Level 1, "Assistance to Human," as outlined in the EASA Concept Paper [9]. The distinction between levels is based on how decisions are ultimately made. In this case, humans retain full control over final decisions, with AI providing support by quantifying uncertainty to assist in the decision-making process.

1.8 Compliance with national regulations

The applicant should comply with national and EU data protection regulations (e.g., GDPR), i.e., involve their Data Protection Officer (DPO), consult with their National Data Protection Authority, etc. The applicant may explain why the data protection regulations do not apply.

The comprehensive FSM includes the rotors' data-driven modes, which is an AI/ML system trained for rotor thrust and torque predictions. It relies on numerical simulations and does not involve the use of personal or sensitive data. As such, the system does not process or store any personal data, meaning that compliance with national and EU data protection regulations, such as the GDPR, is not applicable in this case.

1.9 Transparency analysis

- **The analysis should include the assessment of each output w.r.t. the need for an explanation along with the specification of such explanations.**
- **The analysis should include rationales for clarity, relevance, consistency, and completeness of the qualitative/quantitative criteria.**
- **The analysis should outline the cases in which the AI/ML system communicates the rationale behind its decisions.**
- **The analysis should outline the considerations related to temporality of explainability, including the rationale for the selected timing, implementation details, user guidance, and testing results.**

The transparency analysis of the DT FSM is focused on the rotor thrust and torque module based on the AI/ML system:

- Output explanation: The AI/ML model, using regression methods based on datasets of varying fidelities, produces thrust and torque predictions along with their uncertainty standard deviations σ_i. Each thrust and torque predictions are well explained by being reported in SI units in $[N]$ and $[N \cdot m]$, respectively.

On the other side, the standard deviations σ_i are indicators for the trustworthiness of the simulation itself, in our simulation is given in Normalized Variability Coefficient (NVC) form.

- Clarity, Relevance, Consistency, and Completeness: The regression models have been designed with clear and well-documented methodologies to ensure consistency and relevance. The selection of fidelity levels was based on the computational costs of each fidelity level, while datasets boundaries were chosen from the eVTOL flight envelope limits ensuring they fully capture the required flight scenarios.

3.1.2 AI assurance

The AI Assurance process involves a rigorous documentation task to ensure the reliability and safety of AI/ML systems. As highlighted in [9], under current regulations, system safety relies primarily on the classical requirements-based "Development Assurance" approach, which works well for traditional system designs. However, with ML-based systems, this approach is limited. At the design level, ML systems require attention to the quality, representativeness and correctness of the dataset used for training, validation, and verification, rather than solely on requirements. The primary challenge is ensuring that ML models trained on datasets can generalize well to unseen operational scenarios.

To address this, the author of [9] proposed a new "Learning Assurance" definition, aiming to validate ML systems and provide confidence in their intended functionality. This approach emphasizes transparency and reliability in ML, moving beyond the "AI black box." To guide this process, EASA has outlined a W-shaped Learning Assurance process adapted from the traditional V-cycle, which aligns with ML principles and provides a framework for future regulatory guidance. Here, only the most important objectives are selected and applied to our system, provide an example from the novel “Learning Assurance” concept.

The AI Assurance framework provides a comprehensive approach to validating AI/ML components, aligning with EASA W-shaped process [9]. This framework emphasizes rigorous documentation and iterative verification, ensuring that AI systems meet safety, security, and operational standards within critical aerospace applications. Similarly to the W-shaped process, the assurance framework initiates by detailing system requirements for AI constituents, such as safety, information security, functionality, operational constraints, and non-functional requirements, and thus setting the foundation for robust lifecycle development.

Our proposed process is conducted thorough documentation of the AI/ML model architecture, specifying all constituent elements, such as classifiers and regressors, and describing interactions within the system to facilitate consistent integration across stages. With an emphasis on iterative learning and evaluation, this process defines performance metrics that assess the ML-based model. Continuous monitoring of the learning process occurs at each phase, covering data requirements, training methods, optimization techniques, and reproducibility measures to maintain the model fidelity.

Model verification aligns as well with the final W-shaped process validation stages, employing cross-validation and robustness analysis to confirm the ML-based model stability and responsiveness under diverse scenarios. Additionally, model conversions during implementation shall undergo validation testing to be compared with the actual trained model. It is verified thanks to an evaluation of inference model performance in real-environment testing to ensure that the ML-based was integrated correctly. Finally, a vulnerability assessment secures the model against potential risks, reinforcing the model’s resilience, security, and alignment with aviation’s safety-critical standards.

In what follows, we detail these processes and on a best-effort basis, we address them for the DT use case.

2.1 System requirements (AI/ML constituent requirements)

Documents should be prepared to encompass the capture of the following minimum requirements:

a. **Safety Requirements Allocated to the AI/ML Constituent.**
b. **Information Security Requirements Allocated to the AI/ML Constituent: These would detail how the system should protect data privacy and integrity.**
c. **Functional Requirements Allocated to the AI/ML Constituent: These would outline the functions the AI system needs to perform.**
d. **Operational Requirements Allocated to the AI/ML Constituent: These would state the conditions under which the system should operate (ODD) and how its performance should be monitored and recorded.**
e. **Non-Functional Requirements Allocated to the AI/ML Constituent: These would detail characteristics such**
f. **Interface Requirements: These would describe how the AI system should interact with other systems and users.**

a. Safety Requirements

The AI/ML system must ensure that it operates safely in all its defined operational domains, avoiding any failure that could lead to unsafe conditions. This includes the ability to handle failures in a controlled manner, providing fallback mechanisms such as human override or fail-safe states in the event of unexpected behaviour. If deployed in a real-time system (e.g., flight control), the model should guarantee bounded response times, ensuring timely decisions in critical situations.

b. Security Requirements

When the AI component is deployed as a server, the library must ensure that the service is secure, with protections against unauthorized access or data breaches. Sensitive data handling should be implemented with care, ensuring that no personal or proprietary data is exposed during the model training or prediction processes. This includes, when needed, the usage of encryption protocols (e.g. TLS) to secure communication channels and ensure that data cannot be intercepted during transmission. Additionally, all data used for model training or predictions should be anonymized where possible, especially if personal or identifiable data is involved. Any AI model that relies on third-party data should include a thorough data provenance check to ensure compliance with data protection standards.

Moreover, regular security patches and updates must be applied to the server to protect against new vulnerabilities. The system should include access controls that allow only authorized users to modify, update, or delete the model or its underlying datasets. This can include multifactor authentication (MFA), role-based access control (RBAC), and regular auditing of access logs to ensure compliance.

Additionally, data isolation should be enforced, ensuring that different client data is processed separately unless explicitly authorized for aggregation or sharing. In the event of a data breach or security incident, the system must have predefined protocols for immediate response, containment, and notification to affected parties in accordance with regulatory guidelines.

c. Functional Requirements

- Data Handling and Preprocessing: The library must support loading datasets from various formats and perform preprocessing steps such as normalization, augmentation, and feature selection. The system must provide the ability to split data into training, validation, and test sets with configurable ratios.
- Model Training: The library must allow for training Bayesian Neural Networks (BNNs) across different fidelity levels. It must support Transfer Learning (TL) to fine-tune models from one fidelity level using data from another.

- Prediction and Uncertainty Quantification: The system must enable predictions with uncertainty estimates, allowing for multiple stochastic forward passes to quantify predictive uncertainty. The library must allow for predictions using both trained BNN models or other custom models, integrating the error results from various fidelities.
- Model Testing and Evaluation: The library must provide functions to evaluate trained models on test datasets, outputting metrics such as error rates and uncertainty measures. It must support the comparison of different models trained on varying fidelity data.
- Model Deployment: The library must support deploying trained models as a service, accessible via a server to perform real-time predictions on incoming data. The system must provide configuration settings for server deployment, such as host address, port, and device selection (CPU/GPU).
- Interpretable Results: The library must provide utilities for visualizing results, including correlation matrices and prediction vs. actual output plots. It must allow for saving and exporting these visualizations for further analysis.

d. Non-Functional Requirements

- Performance: The library should be optimized to handle large datasets and complex model architectures efficiently, minimizing training and prediction time, especially when using GPU acceleration.
- Scalability: The system should be scalable, allowing users to easily extend it with new model types or additional data preprocessing and augmentation techniques.
- Usability: The library should have a clear and intuitive API, with comprehensive documentation, including examples, to facilitate easy integration into existing workflows. Configuration files should be user-friendly and well-documented to allow for easy customization of model training, testing, and deployment parameters.
- Robustness: The library should handle edge cases gracefully, such as missing or malformed data, and provide meaningful error messages to aid in debugging. It should include mechanisms for checking and validating input/output data and model configurations to prevent common errors during runtime.
- Portability: The system should be platform-independent, capable of running on various operating systems, and should not rely on any specific proprietary software.
- Maintainability: The codebase should be well-structured and modular, allowing for easy maintenance, updates, and extensions by future developers. The system should include automated tests for key functionalities to ensure that changes do not introduce bugs.

e. Interface Requirements

The AI system must define how it interacts with other systems and users, including data input/output formats, protocols, and APIs. The interface should also support real-time or near-real-time data exchanges, with well-defined communication protocols.

2.2 AI/ML constituents and model architecture

Documentation should describe the main AI/ML constituents that make up the system, including any classifiers, regressors, etc. along with their purpose. The interactions between the constituents should be explained. The model architecture should be described including model type and structure.

The AI module consists of a regression model based on the BNN with TL, which can be queried on-demand through the invocation of a dedicated process, that returns the desired/predicted values based on the provided input. This module can be seen as a component of the architecture that can interact with other system components independently.

Specifically, the AI model is trained to provide predictions based on the fluid dynamics simulation of various physical components of an aircraft. These predictions are then used to determine the whole physical state of the aircraft in different flight phases. All values are used to constitute a so-called DT of the aircraft, i.e., a digitized model capable of interacting with a virtual space.

In this specific case, the BNN is trained to do a regression task, starting from the input data described in Table 4.

Table 4: AI system input variables.

Input data label	Type	Range	Description
aoa	Float32	-180, 180	Angle Of Attack (AOA)
aos	Float32	-90, 90	Angle Of Sideslip (AOS)
u_inf	Float32	0, 40	Freestream Velocity
PP	Int32	0, 4000	Pusher Propeller (PP) RPM value
FR	Int32	0, 4000	Lift Front Right (FR) Propeller RPM value
FL	Int32	-4000, 0	Lift Front Left (FL) Propeller RPM value
RR	Int32	-4000, 0	Lift Rear Right (RR) Propeller RPM value
RL	Int32	0, 4000	Lift Rear Left (RL) Propeller RPM value

The values to be predicted from the BNN are the output of the regression task and are represented by Table 5.

Table 5: AI system output.

Output data label	Type	Range	Description
T_PP	Float32	0, 250	Thrust value of Pusher Propeller
Q_PP	Float32	-5, 0	Torque value of Pusher Propeller
T_FR	Float32	0, 250	Thrust value of Front Right Propeller
Q_FR	Float32	-8, 0	Torque value of Front Right Propeller
T_FL	Float32	0, 250	Thrust value of Front Left Propeller
Q_FL	Float32	0, 8	Torque value of Front Left Propeller
T_RR	Float32	0, 250	Thrust value of Rear Right Propeller
Q_RR	Float32	0, 8	Torque value of Rear Right Propeller
T_RL	Float32	0, 250	Thrust value of Rear Left Propeller
Q_RL	Float32	-8, 0	Torque value of Rear Left Propeller

The choosing of these parameters relies principally on physics-based considerations. In fact, the chosen input variables - Angle Of Attack (aoa), Angle Of Sideslip (aos), freestream velocity (u_inf), and propeller RPMs (PP, FR, FL, RR, RL)—are critical parameters that directly influence the aerodynamic forces and moments of the physical drone. These parameters are fundamental in CFD simulations and experimental aerodynamics. In particular, these inputs cover the essential aspects of an aircraft's aerodynamic state, including orientation (aoa, aos), airflow speed (u_inf), and the thrust generation by propellers (PP, FR, FL, RR, RL). Together, they define the complete aerodynamic and propulsion states, which are critical for accurately predicting the outputs.

The outputs—thrust (T_PP, T_FR, T_FL, T_RR, T_RL) and torque (Q_PP, Q_FR, Q_FL, Q_RR, Q_RL)—are the direct responses to the input conditions. These outputs are sufficient for modeling the DT, because they

represent the forces and moments the model is designed to predict, covering all critical aspects of the aircraft's aerodynamic performance.

A feature selection process was carried out to identify and eliminate redundant or highly correlated features, thereby mitigating the curse of dimensionality and model polarization. Through analysis using covariance matrices and scatter plots, it was confirmed that the input parameters "aoa" and "aos" provided unique contributions to the model's predictions. The parameter "u_inf" maintained a correlation index below 0.3 with rotor RPMs, which is still considered acceptable. However, the rotor RPMs in MF dataset exhibited high correlation among themselves due to constraints in the simulation inputs (e.g., significantly varying the RPM of a single rotor often resulted in non-converging outcomes). Since the four rotors collectively represent the input of the entire system, it is not feasible to arbitrarily exclude any of them. Instead, a data augmentation technique was applied to reduce this correlation (as explained in the next sections).

Ultimately, all the parameters were retained for the training process, as the overall number of features was considered small enough to avoid the effects of the curse of dimensionality.

Model Description

The BNN regression model is a type of Neural Network (NN) that uses probability distributions, instead of simple values, for each weight of every neuron in the network. This particular supervised ML model allows not only to estimate the desired values of prediction but also provides an estimation of epistemic uncertainty (related to model consistency) and aleatoric uncertainty (related to the inherent randomness of the data) of the generated predictions.

Hyperparameters

The hyperparameters of the model are as follows:

- Number of units per layer: the number of artificial neurons implemented for each layer.
- Number of total layers in the model.
- Activation functions: nonlinear functions that are applied to the output of each neuron to introduce nonlinearities.
- Optimization function: the function that determines how the model should be optimized by guiding the model's parameter updates to minimize the error during training.
- Number of Batch: the size of subsets into which the training data is divided during the training process.
- Number of Epochs: the total number of iterations over the entire training set of the model.
- Learning Rate: the sensitivity of updating the model from one iteration to the next.
- Prior distribution: initial probability distribution for each weight, determined by the mean and variance values of a Gaussian function.
- Number of layers to freeze during TL: The process involves fixing part of the network parameters (usually one or more layers from the input layer).

Finally, the model parameters, which are obtained at the end of the training, are represented by the mean and variance values of the probability distributions of each neuron in the BNN model, along with bias values of the activation functions. The parameters are then optimized based on the dataset during the optimization phase.

Final architecture

At the end of the optimization process, the final architecture is obtained:

- Input dimension: 8
- Number of layers: 5
- Units per layer: 224, 144, 160, 112, 96

- Optimization Function: LeakyReLU
- Prior Standard Deviation: 0.0351
- Prior Mu: 0
- Output dimension: 10
- Learning rate: 0.0016
- Learning rate during TL: 0.0081

2.3 Performance Metrics

Documentation should provide rationales for metrics selection and their target intervals or values.

Metrics should be chosen based on their ability to quantify the accuracy, reliability, and uncertainty of the AI/ML model across all relevant operational domains. These metrics are critical in ensuring the model's predictions meet safety and performance standards, especially in safety-critical applications such as flight simulations. The following key metrics have been selected to evaluate the system:

- **Mean Absolute Error (MAE):** The MAE is chosen as a primary metric for regression tasks, providing a direct measure of prediction accuracy by calculating the average absolute difference between predicted values and ground truth values. The MAE is a straightforward and interpretable measure, making it suitable for evaluating the overall performance of the BNN. The formula for MAE ensures that errors are weighted equally, allowing the metric to be intuitive and easy to communicate.
- **Percentage Error:** The percentage error allows for a normalized evaluation of model performance across different output dimensions, accounting for variations in the range of the target values. By scaling the error using the maximum and minimum values of the dataset, this metric provides a more balanced perspective, especially when working with outputs that have different ranges. This ensures the model's predictive performance is consistent across all predicted quantities.
- **Overall Error Coefficient:** This metric consolidates the errors across multiple output dimensions into a single scalar value, providing an aggregate measure of the model's overall performance. By squaring and averaging the percentage errors of each predicted output dimension, this metric gives more weight to larger errors, allowing the evaluation process to highlight critical performance issues in specific dimensions.
- **Normalized Variability Coefficient (NVC):** This coefficient estimates the uncertainty associated with the model's predictions, particularly useful in quantifying how much the predictions deviate from their expected values. NVC is normalized with respect to the range of the dataset, ensuring that uncertainty is expressed as a percentage, making it easier to compare across different output quantities. This is crucial for identifying areas where the model may exhibit high variability or uncertainty, helping to target further improvements or recalibrations.

The evaluation process is conducted on the test set, which is composed of examples extracted from the highest fidelity dataset. To avoid introducing bias from critical points present in the training data, the test set is designed by subdividing the dataset into several parts, with each part evaluated separately. This helps ensure that the model generalizes well across unseen data. The errors are averaged across multiple subdivisions of the HF dataset to provide a more robust estimate of the model’s true performance.

These metrics collectively provide a comprehensive evaluation framework for assessing the predictive capabilities, reliability, and robustness of the AI/ML component. By monitoring both accuracy and uncertainty, the system ensures that it not only meets performance expectations but also maintains the level of confidence required for regulatory compliance in safety-critical applications.

2.4 Learning process management and training process

a. **Documentation should include data requirements, training process, training infrastructure, and model selection and evaluation process.**
b. **Documentation should include the rationale for loss function selection, techniques/algorithms used for optimization, and their target intervals or values.**
c. **Documentation includes training loss and accuracy, validation loss and accuracy, and learning curves.**
d. **Documentation should include a list of optimizations performed, and their rationales.**
e. **Documentation should model complexity, model selection strategy to provide such a trade-off, and description of any techniques used (e.g., regularization).**
f. **Documentation should outline measures taken to ensure reproducibility including data handling, training configuration, hardware and software used, and model versioning.**

The training process makes use of several datasets to train the model, validate it, and evaluate the results. These data must be structured in such a way that they are compatible with a regression task; therefore, all the data will be arranged in a table by rows and columns, where the rows represent the samples, and the columns represent all the features of each sample. In a regression task, one or more columns are defined "label", that, in a supervised training, represent the desired output of the model, or rather the ground truth value for each sample. The training process involves the following steps:

1. All available data are grouped into a csv file. The data represent the results of various simulations at various levels of fidelity. In our case, the data are gathered from simulation with two levels of fidelities. The data can be divided into low and HF. All results of the simulations are checked before being entered into the dataset. The choice of points on which to run the simulations is established through the Latin Hypercube method, which allows for optimally distributed samples. In this case, only converged simulations are used as samples, to make sure that the dataset does not contain erroneous values.
2. The dataset is normalized by column using the MinMax method. The use of normalization generally allows for faster and more stable convergence of the AI model.
3. The dataset is divided into two subsets: LF/HF data, identified by the fidelity of the samples. Each sub-dataset is again divided into other two subsets: training and validation sets with a 70/30 percentage proportion. Finally, the validation set with higher fidelity is divided in half, and one of the two sub-parts is used as a test set. So, finally, the dataset with higher fidelity data will be divided with a percentage proportion of 70/15/15 between training, validation, and test sets. Table 6 represent the dataset subdivision.

Table 6: Dataset composition.

Dataset	Size
Train LF	1404
Validation LF	602
Train HF	108
Validation HF	23
Test HF	24

4. The lowest fidelity training set is used at the beginning for training the model, and the validation error calculated on the validation set is used for early stopping. To improve the performance of the trained model on a dataset with a limited number of data, and to get a more precise value of the model generalization error, the k-fold validation technique is used. For the k-fold validation, k=4 is used.

5. After the model is trained, TL is performed on a dataset with increasing fidelity. The weights of a limited number of initial layers are frozen, while subsequent layers are updated during higher fidelity training. In this way, the model corrects predictions on a limited number of higher-fidelity datasets. This technique improves the accuracy of the model. In fact, it's been shown that using this data fusion technique leads to better accuracy compared to a model trained only on the highest fidelity data, especially when the number of these data is small.
6. Finally, the performance of the model is evaluated on the higher fidelity test set.

Loss function

The loss function is indeed not a variable hyperparameter of the model; rather, it's an essential component that measures the discrepancy between predicted and actual outputs during training. In the context of a BNN-based regression model, the MSE function is commonly used to quantify this difference, calculated using Kullback-Leibler (KL) divergence when dealing with probability distributions. The choice of MSE as the loss function is a standard practice in BNN models for regression tasks.

For each data point in the training set, the BNN computes the predicted output based on the current model parameters. The MSE loss for that data point is then calculated as the squared difference between the predicted output and the actual target output. Mathematically, it can be expressed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\tilde{y}_i - y_i)^2$$

Where $\tilde{y}_i$ is the value predicted by the model, and y_i is the truth value. KL divergence is used in variational inference to quantifies how one probability distribution diverges from another and is used when training probabilistic models to match the predicted distribution with the true distribution of the data. KL divergence between two probability distributions P and Q is defined as:

$$KL(P(x)||Q(x)) = \sum_{x\in D} P(x)\log\frac{P(x)}{Q(x)}$$

Where D is the sample space. This divergence is incorporated into the loss function by adding KL weighted divergence to MSE, to ensure that the model's predictions align with the underlying probability distribution of the data. The final equation for the loss function is defined as:

$$\mathcal{L}(x) = MSE + KL(Q(x)||P(x))$$

Dataset Augmentation

Dataset augmentation operations were conducted with the objective of enhancing the quality and diversity of the dataset for training purposes. A series of basic data analysis operations were performed, including the calculation of the average, maximum, and minimum values for each feature. Additionally, outliers were removed to enhance the precision of the predictions. The following section will provide a detailed explanation of each data augmentation operation:

1. New rows are added in order to augment zero values, explicitly setting some thrust (T) and torque (Q) values to zero where the input (PP, FR, FL, RR, RL) is zero. Additionally, small random variations are introduced to certain RPM values in rows where they are initially zero. This simulates near-zero conditions, expanding the dataset with subtle variations. Moreover, expands the dataset with near-zero conditions to help the model learn about small perturbations around zero, which is crucial for modeling stability and low-power operations.
2. Interpolated data points are generated using cubic interpolation functions to estimate thrust and torque values at different RPMs. This augments the dataset with more finely spaced RPM values between existing data points.
3. A pre-trained LF BNN model is used to predict new outputs by adding small random perturbations to input RPM values and recalculating the corresponding thrust and torque. This utilizes the model to generate realistic but unseen data points based on slight variations in input conditions.

The objective of each augmentation step is to create a more comprehensive and realistic dataset, thereby enabling the BNN model to more effectively learn and generalize from the training data. Furthermore, covariation matrices are calculated before and after data augmentation operations. As illustrated in the next figure, the correlated variables at the outset have become more independent and representative.

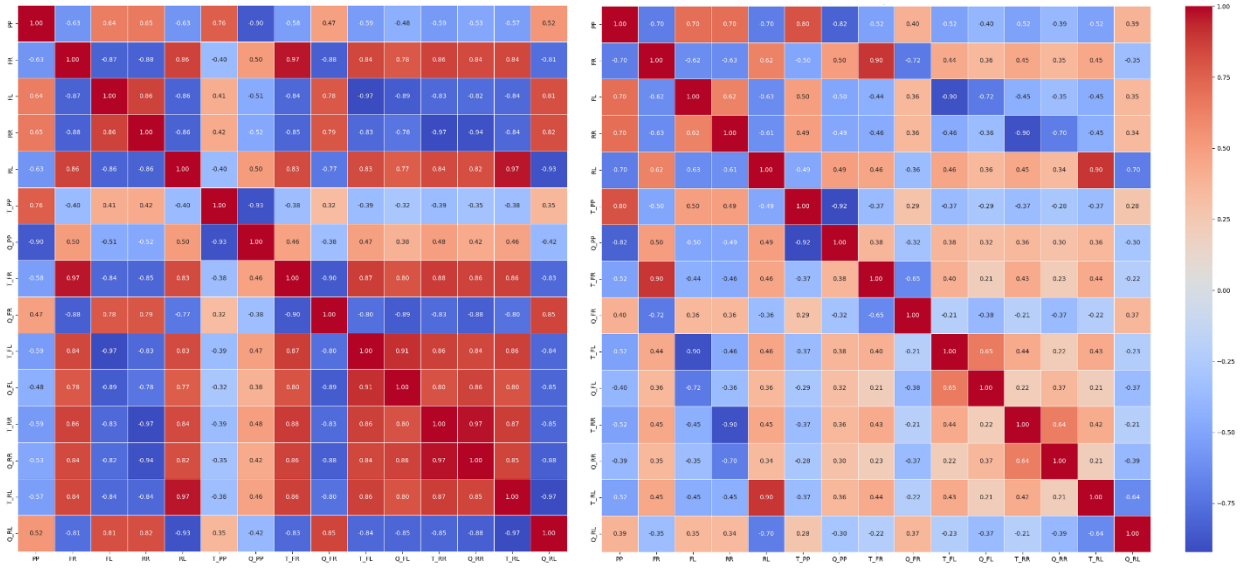

Figure 7: Heatmap before and after dataset augmentation

This helps in improving the robustness, accuracy, and reliability of the model.

Training curves

During the training phase, the model is first trained on LF data. Figure 8 shows the trend of training and validation loss during training. The red segmented line represents the early stopping epoch.

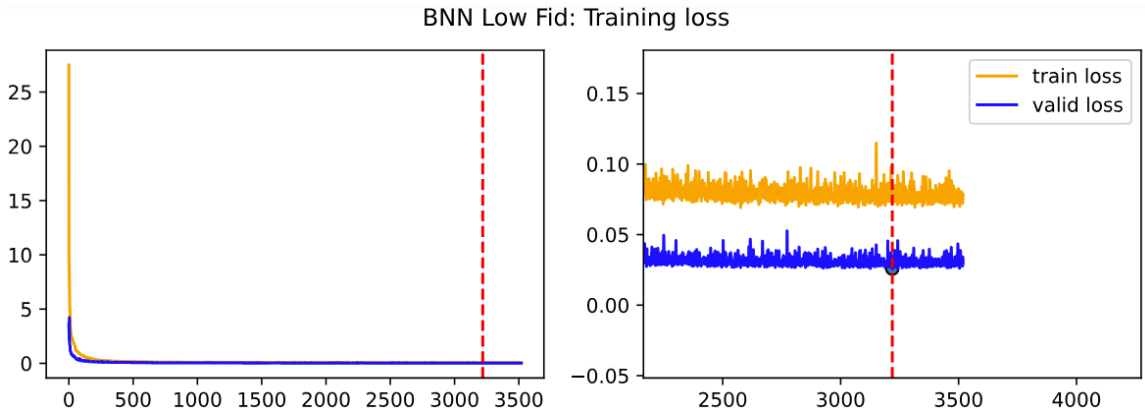

Figure 8: Trend of training and validation loss during training.

After this phase, the 3 latter layers are then frozen, and the training has been executed again, with the following trend of train and validation loss:

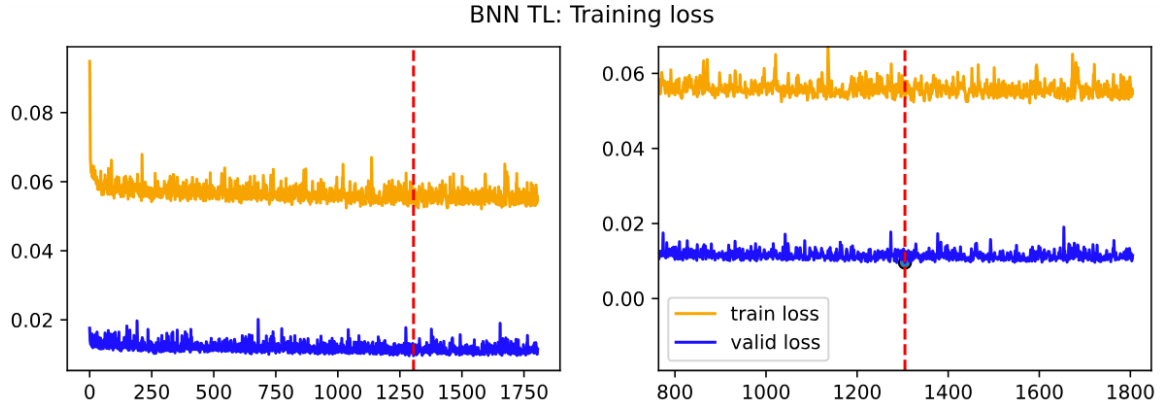

Figure 9: Trend of training and validation loss after the TL.

In conclusion, the use of early stopping combined with the observed stability in training and validation curves helps safeguard against overfitting, as the model training halts once performance on the validation set ceases to improve. This approach minimizes the chance of fitting to noise in the training data. Furthermore, the use of k-fold cross-validation reinforces this robustness by training and validating the model across multiple data splits, ensuring that the model's performance is consistent across different subsets of the data. Together, these techniques offer strong assurance that the model generalizes well rather than overfitting to specific patterns in the training set.

Hyperparameters optimization

The optimization of hyperparameters is done through the Bayesian optimization technique, which utilizes a probabilistic model to guide the search for hyperparameters, reducing the total number of iterations needed compared to exhaustive search methods such as random or grid search. Optimization was performed using the parameters in Table 7.

Hyperparameter	Min value	Max value	Step	Best value
Number of units per layer	16	256	16	[224, 144, 160, 112, 96]
Number of layers.	2	7	1	5
Activation functions	[ReLU, PReLU, LeakyReLU, Tanh]		-	LeakyReLU
Prior distribution's mean	0	0.1	0.0005	0
Prior distribution's std	0.0001	0.0996	0.0005	0.0351
Learning rate during pre-training	0.0001	0.0996	0.0005	0.0016
Learning rate during TL	0.0001	0.0996	0.0005	0.0081
Number of layers to freeze during TL	1	6	1	2

Table 7: Hyperparameter range during Bayesian optimization and best value obtained

The optimization function was not optimized because Adam is a common choice, which automatically adjusts the learning rate for each network parameter, ensuring computational efficiency and adaptability to different types of data.

The number of Batch was not optimized as the network tends to perform better always with a small number of batches.

The number of epochs is also not optimized as it is determined by the early stopping method.

Dropout and batch normalization were not implemented because, in all cases, they generally decrease the accuracy of the model.

Comparison with other models

The performance of the BNN model with TL was compared with one of the few alternative representative methods that allows Data Fusion and calculating uncertainty simultaneously, namely CoKriging. This technique uses the Gaussian Process (GP) model trained on HF data, where LF prediction is added for each point. The CoKriging method obtains good prediction estimates around the points in the training set, but it does not have the same generalization capabilities as BNN.

Framework

The framework used to implement the model is PyTorch. Specifically, PyTorch version 2.0.1 was used with support of the torchbnn package version 1.2 for the BNN implementation. Both packages were installed in a Python 3.8.18 environment.

2.5 Learning Process Verification

a. **Performance evaluation on test data: Documentation should describe the test dataset, evaluation metrics, evaluation methodology (e.g., Cross-Validation), and the results.**
b. **Requirements-based verification of the trained model behaviour: Documentation should include the verification methods, and a coverage assessment evaluating the extent to which these methods provide sufficient coverage of the requirements. Any limitations and assumptions made should be stated.**
c. **Robustness optimization during training and developing: Documentation should describe how development and training increase the robustness of the AI component.**
d. **Stability analysis: Documentation should provide a stability analysis of the algorithms and the trained model including sensitivity and robustness analysis along with the results.**

a. Performance evaluation on test data: Documentation should describe the test dataset, evaluation metrics, evaluation methodology (e.g., Cross-Validation), and the results.

Metrics

The evaluation is done on the test set consisting of examples taken from the highest fidelity dataset. To avoid that the test set contains important points for training, the evaluation is done on several models trained away on different subdivisions of the HF dataset and the error is averaged. To evaluate the performance of the regressor and the amount of uncertainty generated by the BNN, a percentage estimate of the error is calculated. The error is calculated as the average of the MAE on each sample of the test set. The value is scaled by the maximum offset of the values in the entire dataset and then multiplied by 100 to obtain a percentage evaluation. Finally, this error is then squared averaged for all the 10 output dimensions predicted to determine a univocal value Err_{coeff} that determines the overall error of the model. In this way is possible to better consider high error values on single dimensions. The error formula is then depicted:

$$Err_q = \frac{\sum_{y \in \mathbb{D}_{test}} \left(MAE(y, \tilde{y}) \right)}{n_{\mathbb{D}_{test}} \cdot (\max(\mathbb{D}) - \min(\mathbb{D}))} \cdot 100$$

$$Err_{coeff} = \sqrt{\sum_q \frac{\left(Err_q \right)^2}{n_q}}$$

Where Err_q is the percentage error referred to the quantity q. $MAE(y, \tilde{y})$ is the Mean Absolute Error between y that is the Ground Truth in the test set, and $\tilde{y}$ that is the value predicted by the model. n_q represents the number of quantities, which in our case are 10 different output quantities, and $n_{\mathbb{D}_{test}}$ is the number of samples in the test set. Finally, $\mathbb{D}$ represent the whole dataset and $\mathbb{D}_{test}$ is the test set.

To give an estimate of uncertainty, a percentage coefficient is again used, which is the NVC. This coefficient is based on the value of standard deviation σ produced by the prediction of the BNN. For each quantity predicted, the coefficient is calculated as the value of σ_q scaled by the maximum offset of the values in the entire dataset and then multiplied by 100 to obtain a percentage evaluation:

$$NVC = \frac{\sigma_q}{\max(\mathbb{D}) - \min(\mathbb{D})} \cdot 100$$

Where σ_q is the standard deviation referred to as the output quantity q. This coefficient provides a normalized measure of the variability of the signal relative to its range, expressed as a percentage. This can be particularly useful in comparing the variability of different signals with varying ranges.

Evaluation of the method

Using the value of Err_q and Err_{coeff}, it's possible to evaluate the models on the test set to verify the effectiveness of the method. *Table 8* shows the multi-fidelity BNN-TL comparison with BNN models trained separately on low and HF dataset. All the training results are calculated and averaged in a k-fold cross validation process, with k=5.

Table 8: Comparison between LF, HF and BNN-TL errors.

Model	Err_{T_PP}	Err_{Q_PP}	Err_{T_FR}	Err_{Q_FR}	Err_{T_FL}	Err_{Q_FL}	Err_{T_RR}	Err_{Q_RR}	Err_{T_RL}	Err_{Q_RL}	Err_{coeff}
BNN LF	1.86	3.17	4.55	6.38	3.97	7.69	3.98	5.8	4.21	3.98	**4.88**
BNN HF	2.73	4.24	9.84	15.11	8.8	14.14	8.36	15.15	9.29	11.8	**10.8**
BNN TL	1.95	2.13	3.41	3.91	3.35	3.54	2.59	2.52	3.36	3.18	**3.06**
CoKrig.	2.19	2.56	3.24	5	3.22	5.13	2.73	3.18	3.12	3.32	**3.5**

Additionally, deviations are calculated for the error values to assess oscillations across different measurements. A mean standard deviation of 0.1% is observed across all model error calculations, indicating strong overall stability and reliability of the error values using k-fold cross validation method.

From the results, it's clear that using the Data Fusion based method based on BNN-TL increased the prediction accuracy, with an overall decrease in the error calculated in every quantity, and a lower error on coefficient Err_{coeff}.

b. Requirements-based verification of the trained model behaviour: Documentation should include the verification methods, and a coverage assessment evaluating the extent to which these methods provide sufficient coverage of the requirements. Any limitations and assumptions made should be stated.

The AI/ML-based rotor thrust and torque model in the FSM is subject to several key requirements, as outlined in the initial system design documentation. These include accuracy, performance under various flight conditions, and reliable integration into the overall flight dynamics simulation. To verify that the trained model meets these requirements, the following methods are proposed and commented:

- **Comparison with ground-truth data:** the rotor thrust and torque predictions from the AI/ML model are compared against ground-truth data, which could be HF CFD simulations and/or flight test data. This verifies that the model predictions are in line with state-of-the-art physical models.
- **Error Metrics:** The error metrics such as the Err_{coeff} defined in point 2.5.a can be used to quantify the deviation of the AI/ML model predictions from the validation dataset. The goal would be to verify the model accuracy and minimize prediction error across different operational scenarios.
- **Model testing on edge cases:** The AI/ML model is tested to flight envelope boundaries, such as high-speed manoeuvres, rapid altitude changes, and heavy gusts, to assess its robustness. Those tests can be useful to analyse the model under non-nominal conditions, verifying it maintains accuracy and stability outside normal operational ranges.

- **Cross-Validation:** To verify that the model performs consistently across the dataset and is not overfitted to specific conditions, training and validation using k-fold cross-validation should be used. This process ensures the model's generalizability across various rotor speeds, flight conditions and fidelity levels.

c. Robustness optimization during training and developing: Documentation should describe how development and training increase the robustness of the AI component.

The robustness of the AI module, based on the BNN-TL, is optimized throughout the development and training phases to ensure it performs reliably across diverse operational conditions, including varying input data and environmental conditions typical of fluid dynamics simulations in aircraft systems. This section outlines the key techniques and strategies used to enhance the robustness of the AI module discussed in previous sections (for a full overview of technical details, please refer to the whole section 2).

- **TL for Multi-Fidelity Data**: The model employs TL to realize Data Fusion and improve robustness when handling both low- and HF datasets. Initially trained on a larger set of LF data, the BNN is later fine-tuned with a smaller set of HF data. During this phase, certain layers of the network are frozen to retain knowledge from the LF data, while other layers are adjusted to fit the higher-fidelity inputs. This approach ensures that the model is capable of generating reliable predictions, even when trained on a limited amount of high-quality data, while leveraging the broader patterns learned from LF data.
- **Uncertainty Quantification**: The BNN architecture inherently incorporates uncertainty quantification, providing not only predictions but also estimates of the model's epistemic and aleatoric uncertainties. This ability to quantify uncertainty directly contributes to robustness, as it allows the model to assess its confidence in its predictions. By identifying scenarios where the model exhibits high uncertainty, further training or adjustments can be applied to reduce variability and improve reliability in critical flight phases.
- **Data Augmentation**: A series of data augmentation techniques are used to reduce correlations among input variables and expand the training dataset. For instance, small random variations are introduced to RPM values, and near-zero conditions are simulated to help the model handle low-thrust and low-torque situations. This augmentation ensures that the BNN is robust when exposed to a wider range of input conditions and edge cases, ultimately improving its generalization capabilities. For further details on data augmentation, please refer to the related section.
- **Regularization and Hyperparameter Tuning**: To prevent overfitting and to increase model robustness, regularization techniques, such as Bayesian optimization of hyperparameters, are employed. This includes tuning key hyperparameters like the number of units per layer, learning rates, and prior distributions. Additionally, early stopping mechanisms are used during training to avoid overfitting and ensure that the model generalizes well to unseen data.
- **Cross-Validation**: The use of k-fold cross-validation during training helps ensure that the model is robust across different partitions of the dataset. By rotating through different subsets of the data, the model's performance is validated across various configurations, reducing the likelihood of overfitting to a particular subset and improving the overall generalization of the model across all inputs.

These techniques collectively ensure that the AI module, as part of the larger simulation framework, can reliably predict the physical states of the aircraft and its components under various flight conditions. By continuously optimizing robustness through the development and training process, the BNN-based model is better equipped to handle real-world applications and operational edge cases, providing accurate and stable predictions necessary for the creation of a DT of the aircraft.

d. Stability analysis

Documentation should provide a stability analysis of the algorithms and the trained model including sensitivity and robustness analysis along with the results.

The stability analysis of a model is made to validate its robustness and reliability. This process involves conducting many predictions across the entire flight envelope, representing all possible flight conditions within the input space. The goal is to demonstrate that the model consistently produces accurate and stable results across all possible inputs.

Additionally, to ensure stability BNN uncertainty quantification is used. BNNs provide not only predictions but also estimates of uncertainty associated with those predictions. This uncertainty quantification allows us to examine the dispersion or variability of the model's outputs across different flight conditions.

By calculating the NVC for the model's predictions, the average dispersion of results throughout the flight envelope can be assessed. A low NVC indicates that the predictions exhibit minimal variability or dispersion across various areas of the input space. This low variability signifies stability in the model's performance, indicating that it consistently produces reliable predictions regardless of the specific flight conditions.

In addition, representative points of the flight envelope predicted by the BNN model are subsequently reviewed by an expert of the field to determine if the model's behavior is realistic and does not produce inconsistent values, especially for points that were not included in the training/validation set. This evaluation can also be done visually by examining the output curves of the model in each section of the input space. An example is shown in Figure 10.

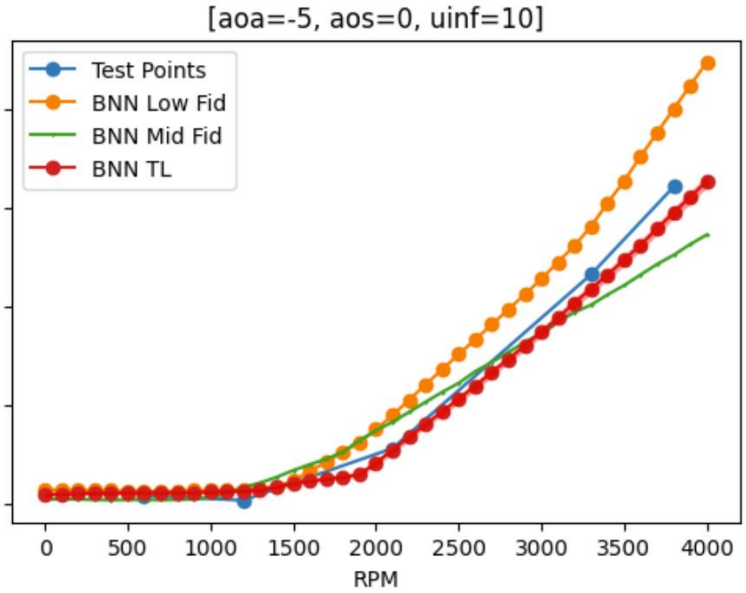

Figure 10: Comparison between BNN predictions and test points (Thrust over propeller rotational speed).

In this example, the red curves represent the thrust output of the pusher propeller provided by the multi-fidelity BNN-DF, which corresponds to the blue test points.

2.6 Model implementation

Identify and validate all model transformations, including conversion and optimization steps, ensuring that each change maintains model behaviour and performance when deployed in the software environment.

The rotor's ML-based model was developed and trained within a Python environment using dedicated libraries development for the BNN-TL method. After a successful Learning Process Verification step, the model was implemented into the two main modules:

- Flight Mechanics Module: The ML-based rotors model replaces the Propulsion sub-function within the Aircraft Module. The Flight Mechanics Module is built in MATLAB/Simulink, so an interface was developed to enable continuous communication between the two environments. The rotors ML-based

model operates in a dedicated Python environment and exchanges data with the Simulink model via an internal localhost server. At each timestep, Simulink sends the above-described inputs (aoa, aos, free-stream velocity, rotor rotational speeds) to the ML-based model through this server. The ML model processes these inputs to predict rotor thrust and torque outputs, along with their associated uncertainty estimates. These outputs are then transmitted back to Simulink via the server, allowing the simulation to proceed with updated rotor data.

- Aeroelastic Module: The ML model serves as a substitute for the Blade Element Momentum (BEM) model originally used in both Steady and Unsteady Aerodynamics sub-functions. Since the Aeroelastic Module is entirely written in Python, integrating the ML-based rotor model is straightforward. The necessary Python libraries for the ML model are imported directly within the Aeroelastic Module's environment.

2.7 Evaluation of the performance of the inference model

Documentation should include a description of the test environment (setup, conditions, etc.), test methodology (cases, metrics, execution procedure), and test results (real environment testing) of the inference model compared to the trained model.

The test environment is a dedicated MatLab/Simulink and python framework in which the physics-based Propulsion sub-function of the Flight Mechanics Module (implemented in Simulink) is replaced by the rotors ML-based model (deployed in a python environment). The input/output is managed through a server to allow efficient communication between the Simulink and python environments. Exactly the same test dataset is taken as test cases to be evaluated against the trained model. The errors of both the trained and inference models are compared in Table 9. Even though the test set is exactly the same, the non-deterministic nature of the BNN results us slightly different error values. However, the outputs are very similar, confirming the good performance of the inference model and a low variance between error values.

Table 9: Trained and inference model errors comparison.

Model	Err_{T_PP}	Err_{Q_PP}	Err_{T_FR}	Err_{Q_FR}	Err_{T_FL}	Err_{Q_FL}	Err_{T_RR}	Err_{Q_RR}	Err_{T_RL}	Err_{Q_RL}	Err_{coeff}
Trained BNN TL	1.95	2.13	3.41	3.91	3.35	3.54	2.59	2.52	3.36	3.18	**3.06**
Inference BNN TL	1.91	2.05	3.32	3.80	3.55	3.71	2.83	2.85	3.26	3.11	**3.12**

In addition and as reported in deliverable D-2.1 [13], another confirmation of the performance of our inference is model is given by the actual deployment on the FSM. The integrated ML-based model demonstrates physically plausible results, making it a valuable addition to the FSM by improving upon certain limitations of the original LF physics-based FSM.

First, in comparing the model to trimmed points, FSM-BNN-TL predictions align closely with both the original FSM-LF and flight tests, such as in comparison to elevator deflection and wing root bending moments across various airspeeds.

During flight manoeuvres, especially transitions from helicopter to airplane mode, the ML-based model captures complex aerodynamic interactions, such as rotor-wake and wing downwash effects. For instance, in forward ascent and descent manoeuvres, the BNN-TL model requires slight adjustments to rotor speeds, which enhances its accuracy compared to the LF model by reflecting real-world dynamics, such as increased rotational speeds needed to maintain thrust.

Additionally, the ML-based model predicts aeroelastic responses in the same range as flight tests, verified by comparing wing root bending moments and center-of-gravity accelerations under gust and turbulence conditions.

2.8 ML Requirements Verification

The requirements verification addresses the verification of the AI/ML component fully integrated in the overall system.

The system requirements for the AI/ML constituent are verified as follows:

a. Safety Requirements

The system operates in a safe, controlled closed environment, reducing potential safety risks associated with deploying AI in a real-world aircraft or simulator context. As a Level 1 AI system, it cannot make autonomous decisions, further mitigating any safety concerns as all decisions require user input or verification. Therefore, the operational environment and AI level ensure that safety requirements are adequately met.

b. Security Requirements

The AI component is hosted securely on an internal university server, minimizing the risk of unauthorized access or data breaches. Given that no personal or sensitive data is involved, there are no sensitive data protection requirements, and encryption protocols such as TLS are deemed unnecessary for this setup. Access controls, such as login credentials for authorized users, are in place, satisfying the minimum security requirements for its deployment.

c. Functional Requirements

- Data Handling and Preprocessing: The library supports essential data preprocessing capabilities, including dataset loading, normalization, augmentation, and feature selection. It provides configurable data splits for training, validation, and testing, which aligns with functional requirements for data handling.
- Model Training: The library allows BNN and GP training across various fidelity levels and includes data-fusion techniques (TL and Co-Kriging) to leverage multi-fidelity models, fulfilling model training requirements.
- Prediction and Uncertainty Quantification: The rotor ML-based system generates predictions with uncertainty estimates by performing stochastic forward passes, providing a measurable output of the model's predictive uncertainty.
- Model Testing and Evaluation: Functions for testing models on diverse datasets, outputting error rates and uncertainty metrics, are available. The library allows for comparison across fidelity levels, meeting the evaluation requirements.
- Model Deployment: The model can be deployed on a server to handle real-time predictions, with settings for host address, port, and hardware configurations, satisfying deployment requirements.
- Interpretable Results: Utilities for visualization, including correlation matrices and prediction-actual comparison plots, are provided. The ability to save and export these results allows for further analysis, ensuring interpretability requirements are fulfilled.

d. Non-Functional Requirements

- Performance: While the BNN-TL model is slightly slower on typical setups, it remains functional within the FSM environment. It is, however, not suited for real-time applications on standard laptops, meeting minimum performance requirements for non-real-time applications.
- Scalability: The system design allows for easy extension with new models and data processing methods, ensuring it meets scalability requirements.

- Usability: Documentation, including a GitHub repository, provides ample guidance on usage and integration, making the library accessible and user-friendly.
- Robustness: The library can handle edge cases, such as missing or malformed data, by providing clear error messages. This ensures reliability under various operational conditions, fulfilling robustness requirements.
- Portability: The library is platform-independent, supporting deployment on UNIX, Windows, and MacOS systems, meeting portability requirements.
- Maintainability: Well-structured documentation and code comments enable future modifications and updates, ensuring the system is maintainable for further development.

e. Interface Requirements

The system supports interaction with other applications and users via APIs and defined input/output formats. It allows for both server-based use and integration with MATLAB through direct Python function calls, supporting both real-time and near-real-time data exchanges. This design ensures the interface requirements are fully met.

2.9 Vulnerability assessment

Documentation should describe the methodology and the result of the assessment.

The vulnerability assessment is an important component in ensuring the robustness and security of the AI/ML system, particularly in safety-critical applications such as those involving flight simulation and aircraft certification. This section outlines the methodology used to identify potential vulnerabilities in the system and provides the results of this analysis.

A vulnerability analysis was conducted to identify potential security threats that could affect the AI/ML system. This involves analyzing the system's architecture, identifying critical components (e.g., data pipelines, model inference processes, and external interfaces), and mapping possible attack vectors. The threat modelling also considers both internal and external threats, including those posed by malicious actors, accidental misuse, and system malfunctions. Specific vulnerabilities inherent to ML models, such as adversarial attacks and data poisoning, are also considered. Given the sensitivity of the data used for training and inference, particular attention is given to the integrity and confidentiality of the data.

The vulnerability assessment revealed that, although the AI/ML system used for simulations does not typically involve handling critical or personal data, there are still potential security risks that need to be addressed to ensure system robustness. The following points summarize the key findings:

- **Limited Risk of Data Sensitivity Breach**: Given that the system primarily processes simulation data and does not directly handle sensitive or personal information, the risk of breaching data privacy regulations (e.g. GDPR) is minimal. However, data integrity remains a critical concern, particularly when the AI model relies on external, or private data sources for simulations. Ensuring that data flows remain untampered and confidential during training and prediction phases is essential for maintaining the accuracy and reliability of simulations. This could be reached by implementing internal security protocols to avoid data breaches.
- **Vulnerability to Malicious Data Manipulation:** While sensitive data is not involved, the system could still be exposed to malicious data manipulation, especially through adversarial attacks or data poisoning. These could alter the input data used in simulations, leading to inaccurate or potentially unsafe predictions. To mitigate this, securing access to the system through robust authentication and authorization mechanisms is recommended. Role-based access control (RBAC) and multifactor authentication (MFA) have been identified as key strategies to limit unauthorized access.
- **API Interaction Security**: The system's API, used for interacting with external systems components, poses a potential attack vector. Ensuring the security of API communications through encrypted protocols (e.g.,

TLS) is crucial to prevent unauthorized data injections or manipulation of simulation requests. The secure connection functionality is not directly implemented in the server provided in the deployment of the application; rather, it must be applied during network configuration depending on the specific system in which the application is used. Rate limiting and access controls have also been recommended to prevent potential denial-of-service (DoS) attacks targeting the API. For deploying the application in a shared network environment, it is highly recommended to seek consultation from a network security expert for proper network configuration.

- **Model Robustness to Adversarial Inputs**: The assessment highlighted that while adversarial testing did not reveal critical weaknesses in the AI model's performance under normal conditions, it showed some susceptibility to adversarial inputs where small data perturbations could result in incorrect predictions. Improving the robustness of the model through adversarial training was recommended as a precaution, depending on the assessment of the actual verification of this risk, ensuring that the AI model remains reliable even in the presence of manipulated input data.

Database Integrity and Encryption: Since the model processes and stores simulation data, ensuring data integrity and confidentiality is essential, when needed, to prevent corruption or tampering. Regular updates to encryption protocols are advised to ensure ongoing compliance with best practices in data security.

3.1.3 AI risk analysis

AI risk analysis involves an examination of potential risks associated with the AI system. This includes documenting the objectives, and methodologies such as Failure Modes and Effects Analysis (FMEA) or Fault Tree Analysis (FTA), risk identification, mitigation strategies, and ongoing monitoring. It also requires identifying stakeholders, addressing transparency challenges, and assessing risks related to system components, datasets, and the overall robustness of the AI system.

In the Annex A, a section describes the documentation required to achieve a comprehensive risk analysis, with a focus on the AI/ML components.

3.2 Bridging the Gap with Digital Twin Tests List

The DT needs to be analysed from several different points of view. Here, four main categories have been defined:

- Applications-specific tests: use-case-specific tests tailored to the model's application (e.g., DT for eVTOL simulation)
- Generic test plan for the ML model: general lifecycle stages (data, training, validation)
- Performance evaluation and reliability assessment: compare the DT results with HF data, test the DT stability and reliability
- Operational testing: test the DT under realistic comprehensive applications

In this section, only a general explanation of the category is outlined, while in the Annex B , the entire list is available. While Section 2.2 described the generic test plan for the ML models, the following subsections demonstrate the tests mainly based on CS 27 and partly from CS 23 together with the standards on AI/ML best practices/requirements and guidelines [9].

In order to test the ML model functionality with respect to different requirements outlined by CS 27 and CS 23, it is vital to build a sufficiently large dataset composed of enough samples resulting from the HF models or collected from field tests. This data should be used in both training and testing of the ML model. Note that the same samples cannot be used in both phases. In addition, to guarantee the ML-based DT reflects the real-world eVTOL throughout its operational lifespan and maintains certification standards with a predefined percentage accuracy. In general, setting this percentage is not a straightforward decision and it depends on several factors,

such as whether the ML-application is safety-critical or not. In any case, a percentage of at least 90% is recommended. To improve the model accuracy, a structured process for the continual ingestion of operational flight data into the ML training pipeline shall be implemented. By having established KPIs (Key Performance Indicators) to quantitatively assess the DT's accuracy against real flight data and HF simulations, it is possible to maintain the percentage accuracy requirement. In addition, periodic cross-validation exercises where DT predictions are benchmarked against empirical measurements are necessary for model continuity and accuracy. Finally, monitoring systems and feedback mechanisms shall be integrated to reinforce the model's predictive capabilities and detect early signs of deviation, prompting pre-emptive recalibration to ensure model reliability and validity for certification purposes.

3.2.1 Application-specific tests

When an ML model is put into a specific context, more specific tests will be applied. The contexts include the tasks performed by the ML model (e.g., computer visions vs natural language processing or regression vs classification) as well as, more importantly, the use-case application (i.e., in the scope of this project, DT for eVTOL simulation). These tests may refer to a specific metric, measure, or quality as a requirement By referring to CS 27, the following tests should be carried on to validate the DT solution along with the ML model:

- Hovering Performance at Different Weights, Altitudes, and Temperatures
- Performance Over Full Flight Conditions
- Unsteady Responses and Manoeuvrability
- Gust Load Handling
- Maximum Safe Hovering
- Take-Off Performance
- Dynamic Stability
- Static Longitudinal Stability in Diverse Conditions
- Manoeuvrability and Control
- Transition and Manoeuvring Capability
- Controllability through Different Flight Conditions

The DT solution along with the ML model should support for creating these scenarios and there should be sufficient data available for verification. A full guideline of how these tests should be conducted are described in Annex B

3.2.2 Generic test plan for the ML model

All ML models live the same lifecycle, from conceptualization to data management, learning process, verification and validation, deployment, and retirement. The set of tests applicable considering the general lifecycle of ML models is identical for every application and model type. While they do not point to any specific clause in CS 27 or similar, they play a key role in establishing a trustworthy framework of ML development and deployment for such a critical use case.

3.2.3 Performance evaluation and reliability assessment

This section focuses on a multi-faceted evaluation of the DT’s performance ensuring its reliability for real-world applications. To achieve this comprehensive assessment, we will explore four key areas:

- Statistical accuracy and reliability: We will leverage established statistical measures to quantify the DT's accuracy and reliability in predicting relevant outcomes.

- Predictive capability across flight conditions: A rigorous evaluation will be conducted to assess the DT's ability to make accurate predictions under a diverse range of flight conditions. This ensures the model's generalizability and applicability in various operational scenarios.
- Outlier detection and edge case handling: We will examine the model's capability to identify and handle outliers within the data. Additionally, its performance in unconventional or extreme situations (edge cases) will be scrutinized.
- Robustness under perturbations and variabilities: The DT robustness will be assessed against potential model perturbations, such as slight changes in input parameters. This testing also encompasses operational variabilities that might occur during real-world use. Evaluating robustness ensures the model's stability and its ability to deliver reliable predictions even in the presence of uncertainties.

3.2.4 Operational testing

The simulation of realistic mission profiles and operational scenarios is crucial for the development, validation, and certification of the DT for eVTOLs. This operational testing encompasses the following key aspects:

- Simulation of realistic mission profiles and operational scenarios
    - Flight Simulation Requirement Specification
    - Realistic Mission Profiles
    - Operational Scenario Testing
- Real-time monitoring and adaptive learning considerations
    - Context-Sensitive Mechanisms:
    - Timeliness of Explainability
    - Continual Learning and Model Evolution
    - Monitoring of Model Performance
    - Issue Detection and Resolution
    - Safety Margin Preservation
- Pilot-in-the-loop and human factors integration
    - Virtual Pilot Models and Abuse-Case Testing
    - Handling Qualities and Human-Factors Assessments
    - Realism in Pilot Interactions
    - Selection of Simulation Type Based on Requirements

As stated above, and detailed in Annex B , the certification of the ML-based DT solution for eVTOL simulation involves several different test categories. These tests are not addressed in any of the certification specifications nor the VTOL MOCs by EASA. Adding such a test list will significantly contribute to closing the gap when AI and ML are used in the context of aerospace. On a best-effort basis, throughout the project, we have conducted several of the aforementioned tests to close the gaps, as shown in the previous section about the learning process verification.

4. Gap Analysis on SC-VTOL 2245 Aeroelasticity Use case

In this section, we provide a more detailed gap analysis on a particular MOC namely, *MOC VTOL.2245 Aeroelasticity* [15] to demonstrate why the current MOC cannot ensure the safety and reliability of an ML model specialized in aeroelasticity modelling. Throughout this analysis, we try to close the gaps by suggesting new elements that should be paid attention to in this MOC. While the analysis mainly focuses on the model stability evaluation, it can be further extended to the entire AI lifecycle.

4.1 Analysis and Recommendations on MOC VTOL.2245: Stability

The following first describes the *MOC VTOL.2245* [15] as appeared in the reference text. Then, we detail how these points do not address the stability analysis of an ML model.

MOC VTOL.2245 Aeroelasticity

(a) General. The aeroelastic stability evaluations referred to in this MOC include flutter, divergence, control reversal and any undue loss of stability and control as a result of structural deformation. The aeroelastic evaluation should include whirl modes associated with any lift/thrust unit or other rotating device that contributes significant dynamic forces. Compliance with this paragraph should be shown by analyses, tests, or some combination thereof.

(b) Aeroelastic stability envelopes. The aircraft should be designed to be free from aeroelastic instability for all configurations and design conditions within the aeroelastic stability envelopes as follows:

(1) For normal conditions without failures, malfunctions, or adverse conditions, all combinations of altitudes and speeds encompassed by the VD versus altitude envelope, enlarged at all points by an increase of 20 percent in equivalent airspeed at constant altitude, should be considered. In addition, a proper margin of stability should exist at all speeds up to VD and there should be no large and rapid reduction in stability as VD is approached.

(2) For the conditions described in (c) below, for all approved altitudes, any airspeed up to VD should be considered.

(3) Failure conditions of certain systems should be treated in accordance with VTOL.2205. For these failure conditions, the speed clearances defined in MOC VTOL.2205 Figure 4 apply.

(c) Failures, malfunctions, and adverse conditions. The failures, malfunctions, and adverse conditions which should be considered are:

(1) For aircraft with disposable fuel: critical fuel loading conditions not shown to be extremely

improbable which may result from mismanagement of fuel

(2) Single failures, malfunctions, or disconnections, and any combination of these that is not extremely improbable, of elements in the primary flight control system, tab control system, or flutter damper

(3) Failure of any single element of the structure supporting any engine, lift/thrust unit, shaft, or large externally mounted aerodynamic body

(4) Failures of any single element of the lift/thrust unit structure that would affect the aeroelastic characteristics of the aircraft

(5) Any lift/thrust unit or rotating device capable of significant dynamic forces rotating at the highest likely overspeed

(6) Any damage or failure conditions, required or selected for investigation by VTOL.2240 (a) and (b)

(7) Any other combination of failures, malfunctions, or adverse conditions not shown to be extremely improbable.

(d) Flight flutter tests should be made to show that the aircraft is free from flutter, control reversal and divergence and to show by these tests that:

(1) Proper and adequate attempts to induce flutter have been made within the speed range up to VD;

(2) The vibratory response of the structure during the test indicates freedom from flutter;

(3) A proper margin of damping exists at VD; and

(4) There is no large and rapid reduction in damping as VD is approached.

(e) For modifications of the type design which could affect the flutter characteristics, compliance with (a) should be shown, except that analysis alone, which is based on previously approved data, may be used to show freedom from flutter, control reversal and divergence for all speeds up to the speed specified for the selected method.

a) For DT solutions, stability analysis must also extend to the stability of ML models involved. Stability is one of the most important properties of robustness, where robustness is the ability of an AI system to maintain its level of performance under any circumstances [3]. According to the standard terminology of AI concepts, stability is the extent to which the output of a NN remains the same when its inputs are changed. According to this definition, a stabler ML model is less likely to change its output when input changes are noise. When analysing the stability of the ML model, it still should address different scenarios, conditions, and issues such as flutter, divergence, control reversal, and any potential loss of control due to structural deformation. However, the stability of ML models should be evaluated, especially regarding edge and corner cases, outliers, anomalies, noise, and perturbations. These critical factors represent extreme or unusual scenarios that may not be well-represented in the training data but can still occur in real-world applications. Evaluating stability in these conditions ensures that the model not only performs well on typical data but also maintains reliable behaviours and makes safe, robust predictions when faced with irregular or challenging inputs. Additionally, stability properties and criteria must be defined for the ML model considering its specific role within the system. The stability criteria and properties should be documented along with the rationales for their choice. The stability analysis can be statistical, empirical, formal, or a combination of these. In any case, a rationale behind the choice of methodology should be provided. To understand the stability of the ML model, the analysis should consider all stability factors under comprehensive scenarios, where input perturbations, severe operating conditions, and all the possible corner cases, occur simultaneously. Stability should also be a high-level objective considered during all AI lifecycles besides the verification and validation. For example, techniques such as ensemble learning, adversarial training, data augmentation, regularization, and hyperparameter optimization can be explored during the design and development phase to enhance stability. Finally, the impact of ML model stability on other components and the overall system should be thoroughly evaluated and documented.
b) Stability of the ML model, besides the conditions mentioned in (1), should include variation in the input data features based on which the ML model was trained. These variations may differ for each feature, and the extent of permissible variation should be clearly specified and documented, along with the corresponding rationale for these limits.
c) Data used to train, validate, and test the ML model should include all failure, malfunction, and adverse conditions as specified. Performance (accuracy, and any other relevant KPI) in such scenarios should be traceable. The evaluations should specifically address each of the mentioned scenarios to facilitate a conclusion on the model stability. The input data must accurately represent the scenarios outlined in this MOC. The input features and labels (for supervised learning algorithms) of the ML model may

not directly include failures, malfunctions, and adverse conditions mentioned in this section of the MOC. However, these failures, malfunctions, and adverse conditions should be identifiable and traceable. For instance, while the shaft condition (failure or normal operation) might not be an explicit data feature or label, the effects of the shaft status should still be detectable within the input data.

d) The dataset used for testing should cover the entire speed range and other conditions outlined in this MOC. Given the inherent stochastic and probabilistic behaviour of ML models, test results must be statistically significant, ensuring that the model's performance across different conditions is reliable and not due to random variation.

In addition, the following are strongly recommended:

- Robustness assessment is key to ensuring an ML model can be deployed in safety-critical applications. Stability is only one of the properties of robustness and more evaluation is necessary. In particular, sensitivity, relevance, and reachability [3] (mainly when reinforcement learning is used) should be also analysed. It is strongly recommended to ask for a complete robustness analysis of the ML models. Such analysis can be statistical, empirical, formal, or a blend of them, depending on the context and models used.
- Due to the nature of ML models, the ML outputs should be accompanied by uncertainty quantification or confidence level. For interpolation systems, it is strongly recommended to calculate the uncertainty of the ML model as the stability analysis. The method to provide the uncertainty values and the accepted confidence level should be identified and documented along with the corresponding rationales. This can include confidence intervals, prediction intervals, or variance estimates for the stability metrics provided by the ML model. In addition, how the ML model handles uncertainty and how this affects the overall system stability should be evaluated and documented.
- Furthermore, clear processes for human control and means for human intervention, especially in cases where the uncertainty values may go above the accepted levels should be in place. In cases where human involvement is not feasible in a timely manner, the ML model should be automatically replaced by a deterministic approach that offers safe actions (such as fail-safe).
- Once deployed, the ML model output and relevant KPIs, like uncertainty or stability criterion should be continuously monitored.
- Once deployed, the input to the ML model should be validated prior to inference. If the input data falls outside the defined ODD, the output should not be used, regardless of the stability or other KPIs. Additionally, there must be a clear process for timely human expert intervention. In scenarios where timely human involvement is not feasible, the ML model should automatically switch to a deterministic approach that ensures safe actions (such as fail-safe).

The MOC relies heavily on physical tests and well-validated simulations. With AI/ML being involved, validation against physical phenomena becomes more challenging due to the uncertainty and interpretability issues of ML-based models. In addition, ML models can perform as well as the data on which they were trained. As a result, the training data should include all possible flight conditions, structural deformations, whirl modes, etc. The data used for testing should also incorporate similar representation. Both training and testing data should include well-validated data points, which are based on HF physical models or actual field tests. However, in practice, collecting sufficient data on structural deformations through field tests may be challenging.

Capturing the dynamics of transition phases through AI/ML might be also challenging due to the complexity of these manoeuvres and the potential lack of granular data representing these conditions accurately. In such cases, the limitations of the developed models must be well documented and communicated to the relevant stakeholders.

Simulating failure conditions with ML could be challenging due to the fact that these conditions are usually underrepresented in the training and testing data. A rigorous V&V plan should be in place to ensure that corner cases such as failure conditions are available during both the training and testing of the ML models. In

particular, if these conditions are not well presented, the model will be likely biased to normal conditions. Moreover, AI/ML models must effectively simulate lift/thrust units or rotating devices operating at overspeed conditions, which are critical for understanding aeroelastic instabilities. Such scenarios should be well-represented in the dataset to ensure accurate predictions.

4.2 Bridging the Gap

The focus of the above analysis was the stability of ML models and related concepts. However, to ensure an ML model can be deployed in high-risk domains and safety-critical applications, it is strongly recommended to include detailed requirements and MOCs for the entire AI lifecycle. High-level requirements can refer to existing norms from the ISO SC 42: Artificial Intelligence, as they are not application-specific and are horizontally applicable to all AI systems and ML models. For instance, ISO/IEC 5259 [2] is a 6-part norm for ensuring data quality for ML models. On the contrary, the current MOC offers very little on how to collect, process, and use data for training, validating, testing, and deploying an ML model. It is strongly recommended that the MOC includes:

- Data collection protocols
- Data preprocessing practices such as outlier and anomaly detection and handling, error detection and handling, missing data detection and handling, as well as measures and metrics for representativeness, completeness, timeliness, and other notable properties.
- Data validation processes during the operation (once the ML model is deployed)

In addition, although documentation is promoted throughout the MOC, technical documentation of ML development and deployment has its own intricacies. It is strongly recommended to include the following for technical documentation:

- data quality measures and assessment
- architecture and model selection and optimization
- performance metrics
- test plan and verification methodologies
- model explainability
- monitoring tools
- human control and oversight

Obviously, none of the above is addressed in the current MOC. In order to address this gap, MOC VTOL.2245 should be extended first with a reference to generic ML requirements and MOCs to cover the standard AI/ML lifecycle processes. Second, the aeroelasticity verification and validation, in particular, should focus on the key and standard terminology of AI and ML (for instance, robustness), and outline specific ML-related tests, measures, and metrics according to the items already enumerated in MOC VTOL.2245.

5. Conclusion

This document has presented a thorough gap analysis and recommendations for modifying EASA's MOC framework to better accommodate the unique requirements of ML-based FSM in aerospace applications. The analysis has shown that while MOCs are currently focused on physical testing and deterministic simulation models, they lack specific guidelines or requirements suited for AI/ML-driven models. This is particularly evident in the areas of validation, lifecycle management, and data quality—key aspects of AI-based systems that involve continuous updates, non-deterministic outputs, and dependency on diverse data sources. Bridging these gaps will be essential for ensuring that AI/ML technologies in safety-critical sectors meet stringent safety, reliability, and performance standards.

To address these challenges, we propose a multi-step approach for EASA. Initially, a generic Certification Specification (CS) for AI/ML in aerospace should be established, providing broad requirements applicable to all ML applications in this sector. Following this, the CSs should be amended with context specific MOCs that are tailored based on the application type, scope, and operational environment. This dual-level approach will not only support the unique demands of different applications but will also encompass the entire AI/ML lifecycle—ranging from model development, testing, and deployment to ongoing monitoring and lifecycle management.

Bibliography

[1] European Union. EU AI Act: first regulation on artificial intelligence. Brussels, 2023.

[2] ISO/IEC 5259 Artificial Intelligence - Data quality for analytics and machine learning (ML), 2024. International Organization for Standardization, Geneva, Switzerland.

[3] ISO/IEC TR 24029 Artificial Intelligence - Assessment of the robustness of neural networks, 2024. International Organization for Standardization, Geneva, Switzerland.

[4] ISO/IEC TR 5469 Artificial Intelligence - Functional safety and AI systems, 2024. International Organization for Standardization, Geneva, Switzerland.

[5] ISO/IEC TS 8200 Artificial Intelligence - Controllability of automated artificial intelligence systems, 2024. International Organization for Standardization, Geneva, Switzerland.

[6] Andrea Pedrioli, Marcello Righi. MODEL-SI, D-1.1 Report literature and digital solutions review. Research Project EASA.2022.C25, 2023.

[7] Timothy Mauery, Juan Alonso, Andrew Cary, Vincent Lee, Robert Malecki, Dimitri Mavriplis, Gorazd Medic, John Schaefer, and Jeffrey Slotnick. A guide for aircraft certification by analysis. Technical report, 2021.

[8] Linghai Lu, Gareth Padfield, Philipp Podzus, Mark White, and Giuseppe Quaranta. Preliminary guidelines for a requirements-based approach to certification by simulation for rotorcraft. 2022.

[9] Guillaume Soudain, Francois Triboulet, and Alain Leroy. EASA Concept Paper: Guidance for level 1 and 2 machine learning applications (Issue 02), 2024.

[10] ISO/IEC JTC 1/SC 42 Artificial intelligence - Standardization in the area of Artificial Intelligence, 2017. International Organization for Standardization, Geneva, Switzerland.

[11] ISO/PAS 8800 Road vehicles - Safety and artificial intelligence, 2024. International Organization for Standardization, Geneva, Switzerland.

[12] ISO/IEC 5338:2023 Information technology — Artificial intelligence — AI system life cycle processes, 2023. International Organization for Standardization, Geneva, Switzerland.

[13] Andrea Pedrioli, Marcello Righi. MODEL-SI, D-2.1 Case Study report. Research Project EASA.2022.C25, 2024.

[14] CS SC-VTOL - Special Condition for VTOL and Means of Compliance (Issue 02), 2024. EASA, Cologne, Germany.

[15] MOC SC-VTOL - Fourth Publication of Proposed Means of Compliance with the Special Condition VTOL – MOC-4 SC-VTOL (Issue 1), 2024. EASA, Cologne, Germany.

[16] CS-27 Small Rotorcraft. EASA, Cologne, Germany.

[17] CS-23 Normal, Utility, Aerobatic and Commuter Aeroplanes. EASA, Cologne, Germany.

Annex A Digital Twin Process Requirements

A.1 AI trustworthiness analysis

1.1 End-user identification:
The applicant should identify the list of end-users that are intended to interact with the AI-based system, together with their roles, their responsibilities, and their expected expertise.

1.2 End-user task
For each end-user, the applicant should identify which high-level task(s) are intended to be performed in interaction with the AI-based system.

1.3 AI system identification
The applicant should determine the AI-based system while considering domain-specific definitions of 'system'.

1.4 Operational Design Domain (ODD)

a. Documentation should describe the application domain and clearly define ODD, including environmental conditions, operational scenarios, and system limitations.
b. Documentation should include risk analysis associated with each identified ODD scenario, along with a description of mitigation strategies. The documentation should also include ODD formalization including specific conditions, scenarios, or constraints of the AI/ML system design. Disturbance identification and grading, including their impact on the system's performance, should be part of the documentation.

1.5 Concept of Operations
The applicant should define and document the Concept of Operations (ConOps) for the AI-based system, including the task allocation pattern between the end-user(s) and the AI-based system. A focus should be put on defining the operational design domain (ODD) and capturing specific operational limitations and assumptions.

1.6 Functional analysis
The applicant should perform a functional analysis of the system:

a. Define the system's purpose
b. Identify high-level functions
c. Break down high-level functions into sub-functions
d. Analyse each sub-function
e. Validate the functions

1.7 AI classification
The applicant should classify the AI-based system, based on the levels (1A, 1B, 2A, 2B, 3A, 3B) AI typology and definitions, with adequate justifications.

- Level 1: Assistance to human
  - Level 1A: Human augmentation
  - Level 1B: Human cognitive assistance in decision and action selection
- Level 2: Human/machine teaming
  - Level 2A: Human and AI-based system cooperation
  - Level 2B: Human and AI-based system collaboration
- Level 3: More autonomous machine
  - The AI-based system performs decisions and actions, overridable by a human.
  - The AI-based system performs non-overridable decisions and actions

1.8 Compliance with national regulations

The applicant should comply with national and EU data protection regulations (e.g., GDPR), i.e., involve their Data Protection Officer (DPO), consult with their National Data Protection Authority, etc. The applicant may explain why the data protection regulations do not apply.

1.9 Transparency analysis

- The analysis should include the assessment of each output w.r.t. the need for an explanation along with the specification of such explanations.
- The analysis should include rationales for clarity, relevance, consistency, and completeness of the qualitative/quantitative criteria.
- The analysis should outline the cases in which the AI/ML system communicates the rationale behind its decisions.
- The analysis should outline the considerations related to temporality of explainability, including the rationale for the selected timing, implementation details, user guidance, and testing results.

A.2 AI Assurance

2.1 System requirements (AI/ML constituent requirements)

Documents should be prepared to encompass the capture of the following minimum requirements:

a. Safety Requirements Allocated to the AI/ML Constituent.
b. Information Security Requirements Allocated to the AI/ML Constituent: These would detail how the system should protect data privacy and integrity.
c. Functional Requirements Allocated to the AI/ML Constituent: These would outline the functions the AI system needs to perform.
d. Operational Requirements Allocated to the AI/ML Constituent: These would state the conditions under which the system should operate (ODD) and how its performance should be monitored and recorded.
e. Non-Functional Requirements Allocated to the AI/ML Constituent: These would detail characteristics such as performance, scalability, reliability, and resilience.
f. Interface Requirements: These would describe how the AI system should interact with other systems and users.

2.2 AI/ML constituents and model architecture

Documentation should describe the main AI/ML constituents that make up the system, including any classifiers, regressors, etc. along with their purpose. The interactions between the constituents should be explained. The model architecture should be described including model type and structure.

2.3 Performance Metrics

Documentation should provide rationales for metrics selection and their target intervals or values.

2.4 Learning process management and model training

a. Documentation should include data requirements, training process, training infrastructure, and model selection and evaluation process.
b. Documentation should include the rationale for loss function selection, techniques/algorithms used for optimization, and their target intervals or values.
c. Documentation includes training loss and accuracy, validation loss and accuracy, and learning curves.
d. Documentation should include a list of optimizations performed, and their rationales.
e. Documentation should model complexity, model selection strategy to provide such a trade-off, and description of any techniques used (e.g., regularization).
f. Documentation should outline measures taken to ensure reproducibility including data handling, training configuration, hardware and software used, and model versioning.

2.5 Learning Process Verification

a. Performance evaluation on test data: Documentation should describe the test dataset, evaluation metrics, evaluation methodology (e.g., Cross-Validation), and the results.
b. Requirements-based verification of the trained model behaviour: Documentation should include the verification methods, and a coverage assessment evaluating the extent to which these methods provide sufficient coverage of the requirements. Any limitations and assumptions made should be stated.
c. Robustness optimization during training and developing: Documentation should describe how development and training increase the robustness of the AI component.
d. Stability analysis: Documentation should provide a stability analysis of the algorithms and the trained model including sensitivity and robustness analysis along with the results.

2.6 Model implementation
Identify and validate all model transformations, including conversion and optimization steps, ensuring that each change maintains model behaviour and performance when deployed in the software environment.

2.7 Evaluation of the performance of the inference model
Documentation should include a description of the test environment (setup, conditions, etc.), test methodology (cases, metrics, execution procedure), and test results (real environment testing) compared to the trained model.
2.8 ML Requirements Verification

The requirements verification addresses the verification of the AI/ML component fully integrated in the overall system.
2.9 Vulnerability assessment
Documentation should describe the methodology and the result of the assessment.

A.3 AI Risk analysis

3.1 Documentation should include objective, methodology (e.g., FMEA, FTA), assumption, risk identification, risk assessment, risk mitigation, residual risk assessment, and monitoring and review.

3.2 Documentation should include stakeholder identification, transparency challenges associated with each stakeholder, risk assessment, and mitigation strategies.

3.3 Documentation should include system decomposition, dataset identification, and risk analysis associated with the robustness of each component.

Annex B DT solution test list

In order to ensure the reliability and safety of an ML model, 3 separate, yet interconnected domains of testing shall be well thought out. All ML models live the same lifecycle, from conceptualization to data management, learning process, verification and validation, deployment, and retirement. The set of tests applicable considering the general lifecycle of ML models is identical for every application and model type. On the other hand, when the ML model is put into a specific context, more specific tests will be applied. The contexts include the tasks performed by the ML model (e.g., computer visions vs natural language processing or regression vs classification) as well as, more importantly, the use-case application (i.e., in the scope of this project, DT for eVTOL simulation). In what follows we detail the application-specific tests.

1. Application-specific Tests

1.1 Model Validation Tests:

The efficacy of these tests depends on the precision of the model training, the quality and variety of the training data, and the robustness of the validation process. In the context of a DT, it is essential to establish a feedback loop where the ML model's predictions are continuously compared with actual flight data as it becomes available, allowing for ongoing model enhancement and tuning.

1.1.1 Hovering Performance at Different Weights, Altitudes, and Temperatures

1.1.1.1 Hovering Ceiling Determination:

To validate the ML model's capability to simulate the eVTOL's hovering performance at maximum weight and standard atmospheric conditions:

- Develop an ML simulation scenario that implies maximum gross weight conditions over varying altitudes, considering a standard atmosphere and the rotorcraft in ground effect.
- Confirm that the ML model receives variables such as altitude, weight, and temperature inputs accurately and that it returns hovering ceiling metrics that align with expected physics-based outputs.
- Cross-validate the model's predictions with HF model outputs and actual test flight data (if available) for alignment with CS27 requirements.
- Analyse hovering ceiling determinations across the operational range, focusing on minimum required hovering ceilings at specified altitudes and temperatures.

1.1.1.2 Out-of-Ground Effect Hovering:

To test the ML model's ability to predict hovering performance without the benefits of ground effect at different operational weights, altitudes, and temperatures:

- Train and test the ML model with datasets capturing out-of-ground effect scenarios with various weight and environmental conditions, using LF simulations as a baseline.
- Set up test cases in the DT where the eVTOL hovers outside of ground effect using take-off power across the desired ranges of operational variables.

- Validate the ML model's output against HF models and real-world measurements, focusing on the consistency and accuracy of the predictions.

1.1.1.3 Flight Dynamics in Different Modes:

To evaluate the DT's capability to accurately simulate the eVTOL's flight dynamics across helicopter, transition, airplane, and landing modes, and during both nominal and failure states:

- Develop a comprehensive suite of simulation scenarios that encompass the full range of operational modes, focusing on predicting flight dynamics in a trimmed state.
- Incorporate failure mode scenarios in the DT to test the eVTOL's behaviour under different system malfunctions, such as engine shutdown or control surface lock-up.
- Train the ML model with datasets inclusive of nominal and adverse operating conditions, including manoeuvres and transitions across various flight configurations.
- Conduct DT validation against HF models, ensuring that the loads, forces, moments, and trimmed flight responses are accurately captured and align with real-world performance metrics for both steady-state and transient dynamics.

1.1.2 Performance Over Full Flight Conditions

The ML model within the DT must be meticulously tested to account for the intricacies of the eVTOL's performance in full flight conditions, from vertical operations to forward flight. It is crucial to balance the learning from LF and HF models, ensuring the ML algorithms are exposed to wide-ranging training data that includes edge cases and failures to meet the CS 27 performance requirements. Continuous validation against HF models and actual flight data (where available) is essential.

1.1.2.1 Vertical Climb/Descent:

To test the ML model's precision in simulating the eVTOL's rate of climb and descent over varying conditions:

- Collect or generate comprehensive training data sets that represent the eVTOL's climb and descent performance over a range of weight, altitude, and temperature conditions.
- Run simulation scenarios in the DT to assess the ML model's ability to predict steady rates of climb and descent at different minimum operating speeds, verifying with take-off power and landing gear settings.
- Compare the simulation results against HF model outputs and real-world (field test) data to validate accuracy.

1.1.2.2 Transition in Helicopter Mode:

To validate the ML model's capability to accurately simulate the eVTOL's transitional performance from hover to forward flight in helicopter mode:

- Train and test the ML model with diverse transitional flight scenarios that reflect the complexities of transition in helicopter mode[1].

[1] Reinforcement learning techniques or other dynamic models that can navigate the transitional flight envelope effectively may be used for training purposes.

- Execute simulations within the DT and observe the model's proficiency in managing power settings, aerodynamic changes, and control requirements during transition.
- Benchmark the DT predictions against HF simulations to identify any significant deviation and refine the ML model accordingly.

1.1.2.3 Transition to Cruise:

To ensure smooth and safe transitions from vertical flight modes to cruise using the ML model:

- Prepare the ML model with extensive transitional flight patterns between helicopter mode and cruise, including atypical conditions.
- Test the eVTOL's transition to cruise in the DT under varied operational conditions, mirroring **CS 27.141** guidance on transitions between any two flight states.
- Validate the simulation results against HF empirical data.

1.1.2.4 Cruise:

To evaluate the ML model's ability to predict the eVTOL's performance during steady-state cruise flight:

- Train and test the ML model with flight data encompassing a wide range of cruise conditions, including power requirements for level flight at cruise speeds[2].
- Simulate cruise conditions in the DT, comparing ML outputs with validated models and known cruise performance metrics.
- Confirm the model's capabilities to maintain required flight conditions as stipulated in **CS 27.141**, ensuring no exceptional piloting skill, alertness, or strength is required.

1.1.2.5 Cruise Climb/Descent:

To assess the ML model's fidelity in capturing the performance of the eVTOL during climbing and descending phases within the cruise envelope:

- Generate a dataset that reflects the variations in altitude and power settings the eVTOL may encounter during cruise climb and descend scenarios.
- Execute DT simulations to analyse the accuracy of the ML model in predicting the rate of climb/descent during cruise under varying conditions.
- Use the results to calibrate the model performance against HF simulations and actual flight data.

1.1.3 Unsteady Responses and Manoeuvrability

1.1.3.1 Unsteady Responses to Flight Control System Inputs

To ensure accurate simulation of the eVTOL's unsteady responses to dynamic inputs, such as Flight Control System (FCS) commands and varying environmental factors:

- Create an extensive range of FCS input scenarios and environmental conditions to train the ML model, capturing the complex interplay between control inputs and the eVTOL’s response.

---

[2] ML techniques such as GPs or Bayesian Models may be used to optimize for stable performance prediction, in order to capture steady flight dynamics.

- Conduct dynamic simulations within the DT to validate the ML model's ability to predict transient behaviours, including rapid changes in altitude, airspeed, and direction during manoeuvres.
- Compare the DT simulation results to HF models focusing on unsteady aerodynamic forces, moments, and changes in rotor and nacelle dynamics to ensure the predictions uphold necessary agility and safety margins.
- Implement a rigorous validation process to verify that the ML model facilitates stable control without excessive pilot workload across all phases of flight, encompassing the complete spectrum of manoeuvrability and control requirements outlined in CS regulations.

1.1.4 Gust Load Handling

1.1.4.1 Exposure to Vertical Gust of 9.1 m/s (30 ft/s)

To ensure the ML model's proficiency in predicting the eVTOL's structural integrity and performance when subjected to a vertical gust of 9.1 m/s (30 ft/s) as per **CS 27.341**:

- Procure or simulate datasets that capture the eVTOL's response to sudden vertical gusts, including both aerodynamic and structural responses.
- Train and test the ML model with both nominal and extreme gust encounter scenarios to ensure a robust understanding of the eVTOL's behaviour under gust load conditions.
- Develop a suite of gust load scenarios within the DT, ensuring they reflect **CS27.341** standards for vertical gust intensity.
- Simultaneously simulate lift, drag, and moment changes along with the eVTOL's corrective actions through the DT's control systems, validating the ML model's ability to dynamically predict load factors and stress distributions.
- Contrast the ML model's predictions against HF CFD models that account for gust load interaction, verifying that model outputs are within acceptable error margins of these higher-order simulations. If available, match findings with actual flight test data recorded during gust encounters to further confirm the model's fidelity.
- Analyse the eVTOL's load paths and structural behaviour as predicted by the ML model for alignment with the airframe's design tolerances and ensure no critical load exceedance occurs.
- Test not only for immediate responsiveness but also for the potential for performance degradation or required pilot intervention in the aftermath of a gust encounter.

1.1.4.2 Turbulence and Gust Handling

To confirm the ML model's competence in simulating the eVTOL's interaction with turbulence and background environmental flow fields:

- Integrate training data that includes various turbulence models and gust profiles to replicate realistic environmental conditions encountered during all flight modes.
- Evaluate the ML model's ability to predict the aircraft’s aerodynamic responses and structural integrity when exposed to turbulent air masses and gusts, ensuring alignment with CS 27.341 and other relevant regulations.

- Perform extensive DT simulations to rigorously test the eVTOL's airframe and control system reactions under the influence of turbulence intensity levels typically experienced during flight.
- Validate the ML model's predicted responses against higher-fidelity aerodynamic and structural models, considering both immediate effects on flight stability and potential longer-term material fatigue considerations.

1.2 Operational Envelope Tests:

1.2.1 Maximum Safe Hovering

1.2.1.1 Wind Conditions Testing up to 31 km/h (17 knots)

To ensure the ML model can reliably simulate the eVTOL's hover stability and performance within acceptable limits in wind speeds up to 31 km/h (17 knots):

- Feed the ML model with enriched datasets that simulate wind interactions at different velocities, directions, and operational scenarios presented during the hover phase.
- Conduct a comprehensive set of DT simulations subjecting the eVTOL to varying directions and intensities of wind while in hover, both in-field effect and out-of-field effect, to test the ML model's hover predictions.
- Evaluate the ML model's accuracy in the context of maximum wind resistance during hovering by comparing simulation results with those from HF models and any existing test data.
- Refine and adjust the ML algorithm as necessary to ensure safe operation within the operating envelope under wind conditions stated in **CS27.1587**.

1.2.2 Take-Off Performance Verification

1.2.2.1 Take-Off with Critical Center of Gravity and varying altitudes

To evaluate the ML model's predictions of take-off performance at varying weights, center of gravity (CG) positions, and altitude conditions to ensure compliance with **CS 27.51** requirements:

- Generate or collect detailed take-off performance data under a variety of conditions, emphasizing the eVTOL's most critical CG configurations and altitude effects.
- Ensure the dataset encompasses variations from sea-level to the maximum altitude for which take-off certification is requested.
- Train and test the ML model extensively using the collected data, focusing on the take-off phase's nuances, especially in relation to power usage, lift, drag, and control surface effectiveness.
- In the DT, set up take-off simulations that challenge the eVTOL across the range of possible CG positions and altitudes.
- The ML model should predict the required take-off distance, climb rate, and power settings and flag potential hazards or conditions that may lead to unsafe take-off scenarios.
- Validate the DT simulation outputs by comparing them with known physics-based HF models and, if available, actual field test data.

- Ensure that the simulated take-off performance adheres to the safety margins and does not require exceptional piloting skill or exceptionally favourable conditions, as per **CS 27.51**.

1.2.2.2 Safe Landing Post Engine Failure During Take-Off

To test the ML model's ability to simulate and predict the eVTOL's capability to make a safe landing at any point along the flight path post engine failure after take-off:

- Develop engine failure scenarios to occur at various stages during take-off and initial climb as per **CS 27.51(b)** requirements.
- Train and test the ML model using data that incorporates engine-out aerodynamics and performance, specifically focusing on single-engine operation and its effect on the asymmetry of thrust and control responsiveness.
- Using the DT, simulate engine failure immediately after take-off to assess the ML model's predictions for safe return and landing trajectories.
- Analyse the required pilot actions, the potential for control issues, and the adequacy of safety margins during an engine failure situation.

1.3 Dynamic Stability Tests:

1.3.1 Static Longitudinal Stability in Diverse Conditions

1.3.1.1 Climb Condition Stability at VY ± 19 km/h (10 knots):

To validate the ML model's capability to accurately simulate the eVTOL's static longitudinal stability during climb conditions across a range of airspeeds:

- Use datasets that portray the effect of airspeed variations on static longitudinal stability during the climb with critical weight and center of gravity.
- Simulate scenarios where the eVTOL is climbing at VY ± 19 km/h and verify the model's precision in maintaining stability without demanding pilot correction.
- Confirm the results against HF simulations and, if available, flight test data to ensure the accuracy of the ML model within the specified stability envelope.

1.3.1.2 Cruise Condition Stability at 0.8 VNE or VH ± 19 km/h (10 knots):

To ensure the ML model can reliably predict the stability of the eVTOL in cruise conditions around a central cruise velocity:

- Train the model with data covering a variety of cruise conditions, particularly focusing on speed ranges around 0.8 VNE or VH ± 19 km/h.
- Use validation techniques to compare the ML predictions with the behaviour outlined in **CS 27.175**, ensuring the eVTOL exhibits expected stability characteristics.
- Assess the ability of the Digital Twin, through the ML model, to accurately reflect the necessary power settings and trim configurations for stable flight.

1.3.1.3 VNE Condition Stability at VNE ± 28 km/h (20 knots):

To test the ML model's predictions of the eVTOL's stability near and at the never-exceed speed (VNE):

- Develop a training set capturing the eVTOL's performance at speeds approaching and reaching the VNE.

- Simulate various VNE ± 28 km/h scenarios to test static longitudinal stability, with attention paid to control force requirements and load factors.
- Validate the simulated outcomes with existing aerodynamic models and ensure that the eVTOL, as per the ML model's prediction, does not surpass critical stability limits or control loads.

1.3.1.4 Cruise Speed Range Stability between 0.7 to 1.1 VH or VNEI:

To examine the eVTOL's stability across the wider speed spectrum during cruise:

- Incorporate a broad range of speed stability data into the ML model's training set.
- Evaluate the eVTOL's ability to maintain stability as the speed varies from 0.7 to 1.1 times VH or VNEI within the DT environment.
- Analyse trim and control positions and compare predicted stability margins against those outlined in regulations, adjusting the model as required.

1.3.1.5 Slow Cruise Stability Range from 0.9 VMINI to 1.3 VMINI or 37 km/h above trim speed:

To assess the eVTOL's stability during slow cruise conditions at defined speed ranges:

- Tailor the ML model to account for flight dynamics characteristic of slow cruise conditions.
- Verify that varying speeds within the 0.9 VMINI to 1.3 VMINI range result in predictable and stable flight behaviours as per regulatory stipulations.
- Validate and refine the model using extensive simulation results to ensure regulatory compliance.

1.3.1.6 Descent Stability Range at 37 km/h either side of trim:

To validate stability during descent at airspeeds within 37 km/h of the trimmed descent speed:

- Build a descent scenario data set for training that includes varying speeds and power configurations.
- Analyse the ML model's accuracy in simulating the descent stability range and the associated control requirements.
- Ensure that results from the DT fall within the safety and performance constraints put forth by regulatory bodies.

1.3.1.7 Approach Stability Range from 0.7 times minimum recommended approach speed to 37 km/h above maximum:

To confirm the eVTOL's stability during approaches across a range of speeds from conservative to aggressive:

- Use approach-specific training data that reflects the spectrum of approach speeds.
- Validate the ML model's predictiveness for different approach profiles within the DT to ensure safe and stable landing approaches.
- Ensure that the model aligns with approach stability requirements and advice for flight manual operations.

1.4 Manoeuvrability and Control Tests:

1.4.1 Transition and Manoeuvring Capability

1.4.1.1 Sudden Engine Failure Scenarios

To ensure the ML model can accurately simulate the eVTOL's manoeuvrability and controllability following a sudden engine failure:

- Develop detailed simulation scenarios that represent sudden engine failure at various phases of flight, including take-off, climb, cruise, and approach.
- Train the ML model with data capturing the eVTOL's aerodynamic and control behaviour post-engine failure, considering the asymmetric thrust, and altered flight dynamics.
- Utilize robust ML techniques capable of predicting complex, time-critical responses to achieve safe flight behaviour with single or no engine power.
- Execute DT simulations and validate the eVTOL's transition capabilities when one engine fails, ensuring the aircraft can be controlled and landed safely without exceeding structural load limits.

1.4.2 Controllability through Different Flight Conditions

1.4.2.1 Transitions between Flight Conditions without Exceeding Load Factor

To establish that the ML model can accurately simulate the eVTOL's ability to maintain controllability and manoeuvrability through various flight conditions, such as turning, climbing, descending, and accelerating, without surpassing the limit load factor:

- Collect extensive flight data representing a wide range of transitions between flight conditions. This data should encompass turns, climbs, descents, accelerations, and decelerations under different load conditions.
- Design and run DT simulations that put the eVTOL through sequences of flight transitions to examine how the control forces, load factors, and aircraft's inertia affect the flight path and whether these remain within safe operational limits.
- Monitor the aircraft's performance closely for any signs that might lead to exceedance of the limit load factor.
- Compare and match simulation results with expected results from physics-based aerodynamic and structural models, focusing particularly on transitions that stress the airframe like aggressive turns or rapid pitch changes.
- Confirm that the model adequately simulates the eVTOL behaviour so that it performs within the load factor boundaries established by the aircraft's structural design criteria.

.

2. Generic Test Plan for the ML Model

The following tests are related to ML safety and reliability in general but adapted to the use case where possible. While they do not point to any specific clause in CS 27 or similar, they play a key role in establishing a trustworthy framework of ML development and deployment for such a critical use case.

2.1 Verification and Validation (V&V) strategies

2.1.1 Bias and Variance Analysis:

- Conduct a series of learning process iterations to demonstrate that the selected model's performance is consistent across various subsets of the training data, thus indicating that it is not overly dependent on particular data segments.
- Estimate and verify that the bias and variance of the model align with the learning process management requirements, ensuring that the model performance is within acceptable limits.

2.1.2 Performance Evaluation on Test Data:

- Evaluate the trained model's performance using a dedicated test data set separate from the training and validation sets.

2.1.3 Requirements-based Verification:

- Perform a verification of the trained model's behaviour against specified requirements to ensure the model meets the necessary operational criteria.
- Document behavioural conditions, such as distributions of absolute errors across sequences of data frames, in verification reports.

2.1.4 Learning Algorithm Stability Analysis:

- Analyse the performance metrics (e.g., loss functions) over the course of training via methods such as gradient descent to detect and mitigate unwanted behaviours like overfitting, underfitting, or large oscillations that could jeopardize the model's ability to generalize well to unseen data.

2.1.5 Model Stability and Robustness Verification:

- Verify the stability of the trained model by documenting its performance consistency under standardized conditions, ensuring it can reliably retain what it has learned.
- Assess and document the robustness of the model when exposed to adverse conditions to ensure dependable performance under a variety of environmental and operational settings.

2.1.6 Sensitivity Analysis for Error Propagation:

- Determine how model-level errors may affect other components within the system by performing sensitivity analysis and quantifying the impact of such errors, for instance, on pose estimates.

2.1.7 Generalization Boundaries Verification:

- Confirm the predicted generalization boundaries of the trained model using the test data set to guarantee that the model performs accurately on both seen and unseen conditions.
- This type of analysis may include evaluating the model on different types of data such as training, validation, and test sets, including scenarios with previously unseen runways.

2.2 Testing against low fidelity and high-fidelity models

For the thorough validation of the Flight Simulation Model (FSM) used in a DT, it is crucial to test the ML model against both LF and HF models. The steps outlined below constitute the test processes for ensuring that the resulting FSM can stand up to rigorous assessment of its predictive capabilities and fidelity within the eVTOL's flight envelope:

2.2.1 Component-Level Testing:

- Adopt a hierarchical approach, beginning with component-level testing and analyses before progressing to the validation of whole-aircraft behaviour against flight test data.
- Validate complex manoeuvres by preceding them with tests and analyses under relevant steady-state conditions and simpler manoeuvres.
- Component-level tests allow for the identification and remediation of modelling deficiencies before they escalate into overarching system-level issues.

2.2.2 Parallel Validation and Development:

- Proceed with the validation process in parallel with the development of the FSM, ensuring that model refinement is informed by validation findings.
- Utilize a prototype model to efficiently design experiments, select critical testing points, and measure variables vital for FSM validation.

2.2.3 Validation with Limited Flight Test Data:

- When flight test data for specific conditions of interest are not available or impractical to collect, use interpolation methods for validation within the Domain of Validity (DoV).
- In the Domain of Experimentation (DoE), where validation is intrinsically more complex, rely on a prespecified plan for credibility assessment and associated uncertainty analysis.

2.3 Data-driven tests using operational data

Operational data is essential for validating ML models, as it represents the real-world scenarios in which the eVTOL will operate. The following criteria outline the approach for conducting data-driven tests using operational data:

2.3.1 Data Accuracy and Resolution:

- Assess the accuracy and resolution of the data to ensure that it accurately reflects the eVTOL's operating parameters and is precise enough to capture the nuances of its performance.

2.3.2 Annotated Data Quality:

- Verify the quality of the annotated or labelled data since high-quality annotations are crucial for the training and validation of ML models.

2.3.3 Data Integrity Assurance:

- Ensure confidence that the data has not been corrupted while stored, processed, or transmitted over communication networks, which could significantly affect the model's training and validation process.

2.3.4 Traceability of Data Origin:

- Establish the ability to determine the origin of the data, which is necessary not only for troubleshooting and refining the model but also for compliance with regulatory and certification requirements.

2.3.5 Completeness and Representativeness:

- Confirm that the data datasets are complete and representative of the eVTOL's operational domain, encompassing all necessary conditions and scenarios for an accurate model.

2.3.6 Appropriate Data Format:

- Ensure the data is in a format suitable for processing by the model and is conducive to efficient learning and validation processes

2.4 Sensitivity analysis and uncertainty quantification tests

The eVTOL's DT development must incorporate sensitivity analysis and uncertainty quantification tests to ensure the ML model's operational effectiveness and safety. The following methodologies outline the techniques for conducting these essential tests:

2.4.1 Uncertainty Analysis and Quantification:

- Identify the major sources of uncertainty, including data accuracy, model assumptions, and environmental variables.
- Characterize uncertainties by establishing their mathematical descriptions and understanding their potential impact on the model's predictions.
- Determine how uncertainties propagate through the model and aggregate in the analysis results, affecting overall model performance and predictive quality.
- Analyse the implications of these uncertainties, including potential impacts on safety-critical decisions and the reliability of the model's outputs.

2.4.2 Physical Interpretation of Results:

- Ensure that the numerical outputs from the sensitivity analysis and uncertainty quantification are contextualized through physical interpretation, enabling meaningful conclusions to be drawn about the ML model's performance and reliably accounting for all sources of uncertainty.

3. Performance Evaluation and Reliability Assessment

3.1 Statistical measures for assessing ML model accuracy and reliability

Employ an array of statistical metrics to evaluate the accuracy and reliability of the ML model within the eVTOL Digital Twin (DT) context.

3.2 Evaluation of the ML model's predictive capability under diverse flight conditions

Evaluating the ML model's predictive capability under diverse flight conditions is vital in ensuring the safety and efficacy. The following methodologies will be used to achieve a comprehensive assessment:

3.2.1 Stability of Trained Model:

- Verify and document the stability of the trained ML model in response to fluctuations in operational data inputs such as sensor noise, which could impact the trained model output.
- Conduct verification of the model stability across the full Operational Design Domain (ODD), accounting for all expected operating scenarios and edge cases. This includes consideration of:
    - Nominal cases for all equivalence classes, ensuring that the model provides stable and accurate predictions for normal operating conditions.
    - Boundary cases for all singular points, verifying that the model's response at the extreme limits of its functional envelope is still predictable and safe.

3.2.2 Cross-Comparison with High-Fidelity Models and Flight Test Data:

- Benchmark the ML model against HF simulation models and compare predictions with actual flight test data to validate its predictive accuracy across a broad range of flight conditions.
- Evaluate the ML model predictions for various flight manoeuvres, operational parameters, and environmental conditions to ensure its capability to cope with the full spectrum of in-flight scenarios.

3.2.3 Statistical Validation Techniques:

- Apply statistical validation techniques such as k-fold cross-validation to assess the ML model's performance across different subsets of data, ensuring that the evaluation is not biased by any particular segment of data.
- Utilize performance metrics such as precision, recall, and ROC-AUC to gain insights into the model's ability to correctly predict flight conditions and system responses, and appropriately classify operational risks.

3.3 Detection of outliers and handling of edge cases

The ability to detect outliers and handle edge cases is a critical component of ensuring the reliability and accuracy of the ML model within the eVTOL's DT solution. To achieve this, the following strategies are implemented:

3.3.1 Data Cleaning:

- Perform data cleaning operations to improve the quality of the training data for the ML model. This involves the removal of anomalous data points that can pollute the dataset and skew the model's learning process—activities such as the elimination of "yo-yo flights" that disrupt model accuracy.
- Identify and remove outliers, which are data points that significantly deviate from the rest. For instance, in one instance, climb rates exceeding 1,000 feet per minute that were identified as not physically realistic or were related to inaccuracies in the measurement equipment (e.g., radar plots) were removed from the dataset.

3.3.2 Robust Training and Validation:

- Train the ML model using robust methods that can handle edge cases effectively. These include techniques that can generalize well from the core of the data distribution to its tails where less frequent operational scenarios (edge cases) reside.
- Design validation procedures to specifically test model performance on edge cases, ensuring that the model can handle rare scenarios with the same level of accuracy and predictability as more common ones.

3.3.3 Uncertainty Quantification:

- Quantify the uncertainty associated with model predictions, particularly for edge cases, to understand the level of confidence in the model's outputs. This can involve probabilistic modelling and Bayesian techniques that provide not only predictions but also confidence intervals around those predictions.

3.3.4 Adversarial Testing:

- Conduct adversarial tests to challenge the model with inputs designed to confuse or trick it, ensuring the model's resilience to potential "attack" scenarios that could lead to significant prediction errors. This is particularly relevant for any ML system that may be subject to evasion attacks where inputs are crafted to generate incorrect outputs deliberately.

3.3.5 Monitoring and Feedback Mechanisms:

- Implement real-time monitoring and feedback mechanisms to continually track the model's performance once deployed. By detecting unexpected behaviours or outliers in operational data, these systems can flag issues for further investigation or adaptation.

3.4 Robustness checks against model perturbations and operational variabilities

To ensure the reliability of the ML model within the eVTOL DT, robustness checks must be performed to evaluate the model's performance in response to perturbations and operational variabilities. These tests are designed to ensure that the ML model can maintain its stability and provide reliable outputs even when faced with disturbances that may occur during normal operational phases. The approach includes:

3.4.1 Addressing Input Fluctuations:

- Develop verification cases that represent reasonable deviations in operational parameters, considering the impact of such variations on model output.
- Evaluate the model's resilience when exposed to input fluctuations to assess whether the control strategies and system responses remain within acceptable safety margins.

3.4.2 Operational Variability:

- Account for different types of operational variabilities that could affect the model's performance. These could be aspects like changes in environmental conditions (e.g., temperature, humidity, wind), component wear and tear, or unanticipated user inputs.
- Create scenarios that simulate these variabilities to assess their influence on the model output and the associated safety implications.

3.4.3 Scenario Testing:

- Perform scenario-based testing that includes a broad spectrum of operational conditions, including extreme but plausible situations (such as harsh weather or near-failure states), to test the boundaries of the model's robustness.
- Incorporate feedback from these scenario tests into the model's development loop to refine its predictive capability and enhance its robustness.

4. Operational Testing

4.1 Simulation of realistic mission profiles and operational scenarios

The simulation of realistic mission profiles and operational scenarios is crucial for the development, validation, and certification of the Digital Twin (DT) for eVTOLs. This operational testing encompasses the following key aspects:

4.1.1 Flight Simulation Requirement Specification:

- Establish the types of flight simulations required, which may include desktop 'off-line' simulation, pilot-in-the-loop simulation, or hardware-in-the-loop simulation, each providing unique insights into the eVTOL's operational capabilities.
- Determine the characteristics required for the Flight Simulation Models (FSMs) and Flight Simulations (FS) in terms of predictive and perceptual fidelity, ensuring that the virtual environment accurately reflects the real-world behaviour of the eVTOL.
- Define the flight test data needed to support the validation of the FSM and the associated fidelity and credibility assessments, which will prove critical in demonstrating certification compliance through flight simulation.

4.1.2 Realistic Mission Profiles:

- Develop mission profiles that represent a wide spectrum of the eVTOL's operational environment, including manoeuvres, turbulence, and gusts, as well as different eVTOL statuses (nominal/failure) and modes (helicopter/transition/airplane).
- Create a detailed analysis of steady-state loads on the aircraft's wings, tail, and fuselage, as well as the aerodynamic forces and moments and flight dynamics during trim conditions, for each scenario within these mission profiles.

4.1.3 Operational Scenario Testing:

- Test the DT against operational scenarios that the eVTOL may encounter, from standard operating procedures to emergency responses, to ensure the model's response is both accurate and reliable.
- Evaluate the DT's predictions during manoeuvres and when the aircraft is exposed to wind gusts or under the influence of the Flight Control System (FCS) that could affect the aircraft's ability to maintain altitude or attitude.

4.2 Real-time monitoring and adaptive learning considerations

These considerations include:

4.2.1 Context-Sensitive Mechanisms:

- Implement adaptive explainability mechanisms within the AI system that enable it to adapt to its environment either by design or through learned experiences. Adaptive explainability helps the system to understand and react to various operational contexts intelligently.

4.2.2 Timeliness of Explainability:

- Determine the timing of when explainability features will be available to the end-user, considering factors such as the time-critical nature of the situation, user needs, and operational impact. For real-time monitoring to be effective, users must have timely access to understandable explanations of the AI system’s behaviours and decisions.

4.2.3 Continual Learning and Model Evolution:

- Establish a framework for continual learning where the model can update itself with new operational data to improve its predictive accuracy and resilience. This involves setting up processes to ingest new data, re-train or fine-tune the AI models, and validate these updates before deployment.

European Union Aviation Safety Agency

Konrad-Adenauer-Ufer 3

50668 Cologne

Germany

Mail EASA.research@easa.europa.eu
Web www.easa.europa.eu

An Agency of the European Union