Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 136 (2025) 552–557

www.elsevier.com/locate/procedia

35th CIRP Design 2025

Architectural design of structural health monitoring (SHM) based digital twin for the next-generation landing gear

Anapat Sarasin, Kailun Deng, Yifan Zhao, Agusmian Ompusunggu*

Centre for Life-cycle Engineering and Management (CLEM), Faculty of Engineering and Applied Sciences (FEAS), Cranfield University, College Road, Bedfordshire MK43 0AL, UK

* Corresponding author. Tel.: +44-1234-754120; E-mail address: agusmian.ompusunggu@cranfield.ac.uk

Abstract

Weight reduction on landing gear (LG) supported with a robust structural health monitoring (SHM) system is considered one of the promising approaches to achieving sustainable aircraft. In this research, the main requirements for establishing an SHM-based digital twin for the next-generation LG are formulated through the findings from a systematic literature review, as the inputs to design a scalable digital twin (DT) architecture comprising high-fidelity finite element (FE) models, sensor data, connectivity, AI, physics-informed-AI simulation models, etc. Given landing gear complexity, the model-based systems engineering (MBSE) paradigm was adopted using Capella and MATLAB to create, verify and validate the architecture.

Keywords: Landing Gear; Structural Health Monitoring (SHM); Digital Twin; Sustainability; Systems Engineering; Model-Based Systems Engineering (MBSE)

1. Introduction

In the aerospace industry, virtual testing tools like Finite Element Models (FEM), Computational Fluid Dynamics (CFD), and Multibody Dynamics (MBD) are essential for designing landing gear models and performing Time-Based Maintenance (TBM) to meet global certifications. However, TBM does not address unexpected incidents that may lead to maintenance schedule shifts and additional costs. To address this issue, U.S. organisations have advocated for Condition-Based Maintenance (CBM), which allows for predictive maintenance (PdM) through accurate health monitoring and forecasting [1]. CBM optimises product availability by improving monitoring, reducing downtime, and fulfilling customer needs [2]. Implementing CBM in aerospace applications, such as Health and Usage Monitoring Systems (HUMS) or Structural Health Monitoring (SHM), extends aircraft service life and lowers maintenance costs by detecting issues before failure, particularly for landing gear classified as "safe-life" components [3,4].

A Digital Twin (DT) is a practical tool for integrating virtual testing and high-precision technologies into a cohesive system. It enhances real-time monitoring and ensures product certifications, addressing the limitations of standalone virtual testing methods [2]. However, DT systems are often developed post-deployment, increasing development time and costs. The Model-Based Systems Engineering (MBSE) approach, using methods like the Architecture Analysis and Design Integrated Approach (ARCADIA), provides a structured way to define and manage DT complexity. At the same time, MATLAB aids in validating requirements and facilitating the integration of SHM systems [5,6].

Commercial landing gear systems are primarily designed using safe-life methodologies, which rely on historical data such as flight cycles, flight hours, and loading cycles to estimate component downtime. These methods focus on fatigue life but may not address unexpected maintenance

10.1016/j.procir.2025.08.095

needs or associated costs. To handle these limitations, damage tolerance design methodologies have been introduced for U.S. Navy carrier-based aircraft to extend service life and reduce downtime based on actual structural health conditions [4].

Digital Twin for Structural Health Monitoring (DT-SHM) is crucial for ensuring compliance with regulations while maximising product performance using advanced machine learning and virtual testing technologies. By integrating DT, landing gear components can shift from the *safe-life* to *damage tolerance* methodologies [12], allowing for the use of lighter materials that meet structural requirements without compromising in-service time. Additionally, DT can reduce the cost of SHM outsourcing by being developed alongside the prototype [2]. Implementing MBSE for DT development enhances landing gear reliability through CBM and allows for efficient adaptation to other aircraft components with minimal adjustments, improving profitability for all stakeholders in the aviation industry.

The ultimate goal of this research is to create a DT for the landing gear SHM system, which enables virtual testing during the landing gear design phase and can offer feedback to operators with compliance to EASA CS-25 regulations. This will be the assisting tool for reducing weight for the next-generation landing gear system in promoting the damage tolerance approach as opposed to the current safe-life practice. The SHM functions were identified based on the selected components, namely the *shock strut* and *drag strut*, which hold the majority of the landing force and the entire landing gear weight. The SHM functions include shock absorber monitoring, transient overload detection, and fatigue prediction. The research reported in this paper aims to create DT-SHM architecture specifically for fatigue analysis. To accomplish the research aim, a set of objectives are formulated as listed below:

1. To create the MBSE architecture of the DT-SHM in the fatigue analysis, which contains the specifications of general product validation features regarding recent CS-25 regulations through Capella software.
2. To validate and verify the generic model of MBSE architecture within MATLAB to ensure its practical and functional capability for the aerospace industry.

2. Related Work

The DT represents the most advanced version of digital modelling, enabling fully automatic data exchange between digital and physical assets [6]. A digital shadow automatically receives data from the physical world, whereas a digital model relies on manual data input [6]. The implementation of DT in product design and maintenance begins after the preliminary design phase, where conceptual designs are assessed using Failure Mode and Effect Analysis (FMEA). Once a final product design is selected, a detailed design and technological feasibility are concurrently evaluated [2]. If feasible, the DT system is integrated into physical prototypes for testing. Sensor data is collected and used in virtual tests and machine learning algorithms to refine the DT model until validation criteria are met. Financial feasibility is then assessed, followed by manufacturing, marketing, and distribution. DT data also supports the development of maintenance algorithms during the operation phase, feeding back into product design via enterprise applications [2]. The MBSE applied through SysML and Capella, facilitates Smart Product Design architecture, with Capella chosen over other tools such as Modelio, Papyrus, and Gaphor using the Analytic Hierarchy Process (AHP) [2].

As outlined in AIR6168, key SHM practices for landing gear include monitoring shock absorbers, corrosion, transient overloads, and fatigue [4]. Numerical simulations using MATLAB and ABAQUS accurately predict shock absorber performance during dynamic drop tests by modelling free fall, impact, and shock absorption phases [13, 5]. These models are crucial for optimising shock absorber designs and detecting fluid leakage, which affects performance. The Kriging model shows how nitrogen gas and oil leakage impact damping efficiency by monitoring changes in air spring stiffness and oil volume flow rate [10]. In normal practice, these parameters are proposed to be monitored on the landing gear systems during flights. Aircraft OEMs recommend a post-flight check two hours after the last operation to ensure fluid integrity during the least amount of gas dissolution for the A320 fleet [9, 11].

Additionally, transient overload detection is enabled by sensors that monitor hard landings undetectable by pilots, using pressure and load data [3]. For spacecraft components, fatigue monitoring employs a Dynamic Bayesian Network (DBN) within a DT framework to predict crack growth and estimate the remaining useful life (RUL). This model, which incorporates load and model uncertainties, has shown high accuracy, closely aligning with physical test results when relying on a damage tolerance basis [12].

In summary, while various technological approaches for DT-SHM offer significant innovations and support certification processes, a comprehensive architecture for DT-SHM for the next-generation lightweight landing gear is still lacking. There is a clear need for a fully validated and verified DT-SHM framework that integrates both the *safe-life* and *damage tolerance* methodologies during the transformation phase. This involves creating a clear, insightful representation of the system that promotes a shared understanding among all stakeholders, thereby preventing misconceptions and ensuring accurate communication about the product and service.

3. Methodology

To align with the MBSE concept, the V-model in Fig. 1 was introduced to encompass the design process from the system level to complete integration and testing [13]. The architecture of the digital twin for landing gear fatigue analysis covered the steps of system requirements and system-level design, including various technological selections.

3.1 Architecture Construction

To construct the system architecture, the Capella software was employed using the Arcadia method, an approach that, despite its similarities to SysML, provides a more detailed description of internal breakdowns and interconnections [14]. This methodology involves four analytical layers and facilitates simultaneous validation of each step. The approach followed the first three main phases of the ARCADIA method, namely (i) Operation Analysis, (ii) System Analysis, and (iii) System Logical Architecture.

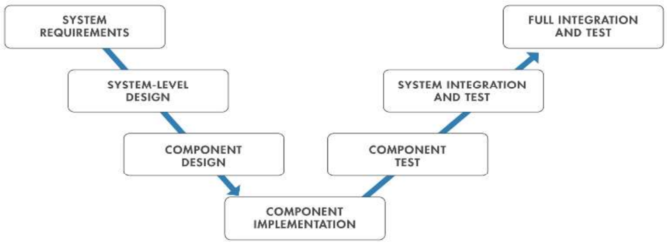

Fig. 1. V-Model for System Development [13]

(1) Operational Analysis

This phase focused on gathering requirements from stakeholders and relevant sources. The process began with identifying required capabilities using the Operational Capabilities Diagram (OCB). Capabilities were linked with arrows to show interactions or extensions. Subsequently, the Operational Architecture Diagram (OAB) details the high-level activities and interactions that the actors and entities are expected to perform as an operational flow, while the operational process can be created to highlight the important procedures.

(2) System Analysis

The second step focuses on a detailed analysis of the product's system, essential for completing the design process. This approach decomposes the system requirements into specific functions, validated through operational activities identified in the previous phase. Operational capabilities, actors, and entities were imported and developed into missions. These missions were broken down into system capabilities, visualised through the Missions and Capabilities Diagram (MCB). The system capabilities were then integrated into the System Architecture Diagram (SAB) to serve the stated missions, connected by functional exchange of essential data. To highlight key functional pathways, the functional chain feature was utilised for specific functional flows.

(3) System Logical Architecture

The SAB was refined to develop the final architecture by translating system functions into a logical structure. Initially, all system functions and exchanges were validated in the system logical architecture phase. The logical architecture was first drafted using **draw.io** for convenience and then finalised in Capella. The Logical Architecture Diagram (LAB) was created as the first deliverable, decomposing system functions into logical functions and linking them through functional exchanges to represent data flow. Operators like duplicate, split, gather, route, and select were used to manage pathways, with descriptions added in the properties tab.

3.2 Model Validation and Verification

To validate and verify the logical architecture, MATLAB's add-on software provides comprehensive tools designed for this purpose. Components within the LAB were selected for integration into the System Composer, which facilitates the creation of architecture models in MATLAB for simulation, validation, and verification. Simultaneously, the Requirements Toolbox supports the validation process by listing all requirements, mapping them to the corresponding components within the framework, and displaying the status of both validation and verification in a unified interface.

(1) Model Validation

All the specified requirements, including missions and capabilities, were integrated into MATLAB Requirement Editor, while each requirement was organised hierarchically

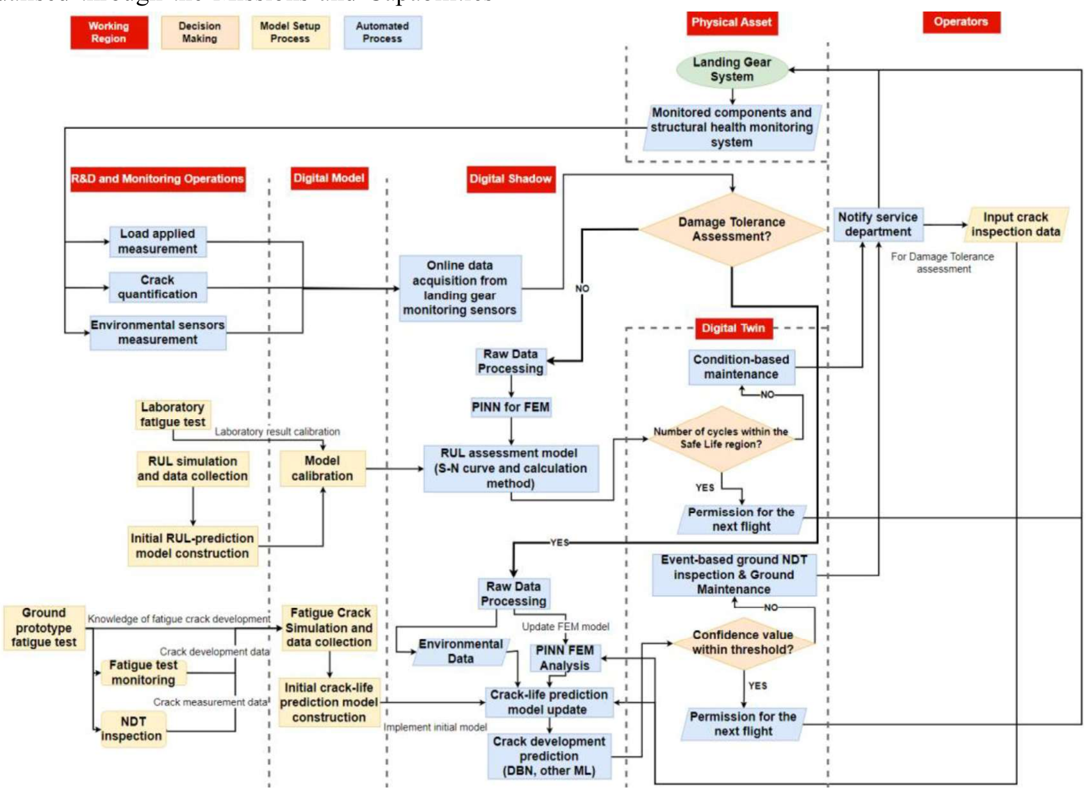

Fig. 2. System Logical Architecture Preliminary Diagram

to mirror the MCB. The Property Inspector tab enables detailed tracking of each requirement's implementation status in the System Composer architecture, indicated by the colour filled in blue when successfully integrated. The architecture was considered fully validated once all statuses were marked blue, confirming alignment with stakeholder needs.

(2) Model Verification

The final procedure validated the functional correctness of the architecture by generating Simulink block diagrams for each component and testing them with the Test Manager add-on for comprehensive test management. Each component used a test harness file to prevent structural changes during testing. Results indicate pass/fail status with green checkmarks for verified components and red crosses for failures. Verification status was tracked in the Requirement Editor, using *green for verified*, *red for failed*, and *yellow for pending tests*, ensuring the system model performed as expected before advancing in the V-model process.

4. Results and Discussion

4.1 MBSE Implementation Results

The framework was constructed from requirements and finalised in LAB consisting of the logical details embedded in one schematic. The preliminary diagram of the logical architecture, as depicted in Fig. 2, presents the logical flow chart of DT-SHM for fatigue analysis, organised into operational sections.

(i) Data acquisition

Initially, each landing gear component is monitored by SHM tools, which gather three key types of data:

1. **Load Data**: Captured by sensors like strain gauges and accelerometers for FEM analysis and structural change detection [15, 6].
2. **Crack Quantification Data**:
   - **Acoustic Emission (AE) Sensors**: Detect crack growth, delamination, and corrosion using piezoelectric, optical fibre, and MEMS materials [15].
   - **Radio Frequency (RF) Sensors**: Monitor crack initiation and quantify damages using passive RF identification technology [15].
3. **Environmental Data**: Parameters like temperature, pressure, humidity, and corrosive conditions, which influence crack growth and material behaviours are recorded. This data is vital for updating crack-life prediction models and ensuring accurate SHM sensor readings [21].

(ii) Data management and FEM

The DT-SHM system classifies assessment types based on data received from landing gear sensors. Load data alone triggers the safe-life assessment path, while full SHM data leads to damage tolerance analysis. The raw data is then transformed into a suitable format for analysis, often converting time-domain data to frequency-domain for structural change detection and FEM model updates using methods like Empirical Mode Decomposition (EMD) and CEEMDAN [12]. The processed data then proceed to FEM analysis on both assessment methodologies. As discussed in the system analysis phase, there are modified and unmodified FEM approaches tailored to specific fatigue assessments. The damage tolerance method frequently updates its FEM modal matrix when structural mode changes or any damages are detected, whereas the safe-life method relies on a consistent FEM model [12]. However, in routine operations, where real-time or near-real-time analysis is expected, surrogate models may replace FEM analysis. One plausible method among computing technologies is the Physics-Informed Neural Networks (PINNs), which aims to reduce analysis time by applying the principle of forward prediction and inverse estimation and coupling with the first principle models [16]. The FEM analysis generates stress distribution and other relevant values, which are then utilised in the subsequent

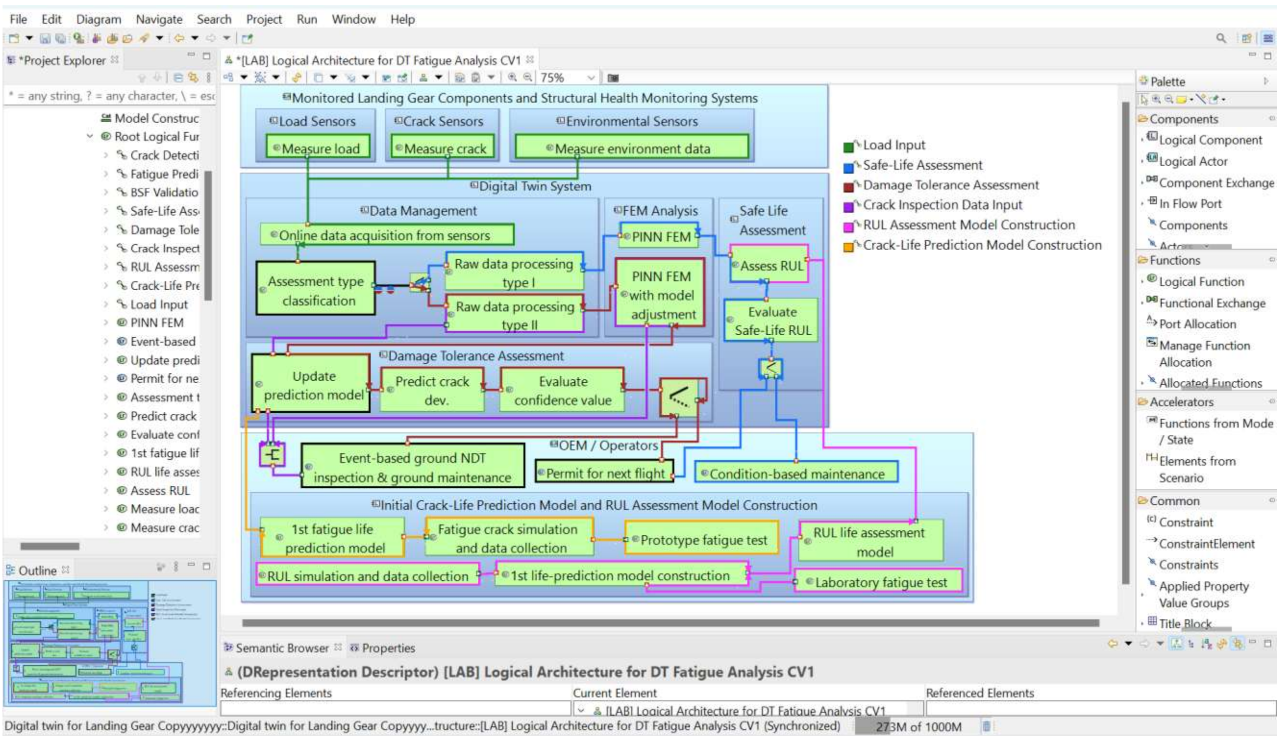

Fig. 3. System Logical Architecture of the DT-SHM with functional chains

fatigue assessments.

(iii) Fatigue analysis models

For the *safe-life* analysis, the RUL assessment is based on the S-N curve for each FEM result and evaluates the remaining number of load cycles, adjusting for a specified safe-life scatter factor. This factor varies for different components according to the CS-25.571 regulations [22]. Following the evaluation, the DT-SHM will assess the condition of the component. If the remaining safe life, expressed as the number of load cycles or flight hours, exceeds the average values required for a single flight, the system will grant permission for the next flight. If not, operators will be notified to replace the monitored components. Concerning the construction of the RUL assessment model, the upper set of yellow boxes in Fig. 2 must be successfully verified by the OEM company before the deployment of the DT-SHM.

The model construction begins with the repetitive RUL simulation in various scenarios and data collection, then the results are combined to create the initial trained RUL prediction model. The initial digital model is tested in the trial-and-error phase simultaneously with the laboratory physical fatigue test for model calibration purposes. Later, the final RUL assessment model is deployed into the DT-SHM system which is the end of the research and development phase.

Switching to the *damage tolerance* approach, as outlined in the MCB capabilities, it introduces additional complexity to the assessment process. Unlike the *safe-life* approach, the *damage tolerance* approach does not use a scatter factor, thus requiring more precise safety and prediction accuracy considerations. The crack-life prediction model is updated with each new data input, influenced by environmental factors, crack length, and FEM results. Once finalised, crack development predictions are generated using inference methods like DBN or other machine learning technologies, producing a curve depicting crack length versus loading time for future conditions [12]. Subsequently, reliability prognosis establishes a confidence value being monitored over time. If the confidence value exceeds a given threshold, a flight is permitted. If it falls below the threshold, operators are advised to perform a Non-Destructive Test (NDT), ground inspections, and necessary maintenance. For the post-inspection, updated crack length data must be manually input into FEM and crack-life prediction models to ensure the confidence value accurately reflects the landing gear's condition, as outlined by [12].

Regarding model construction, the initial crack-life prediction model is derived from ground prototype fatigue tests, which incorporate crack development, crack measurement, and the fatigue behaviour of each component to calibrate with the digital model simulation [23]. The data gathered from these simulations are used to construct the initial prediction model, after which the algorithms are integrated into the digital shadow as soon as the initial model is finalised.

The LAB final architecture, shown in Fig. 3, presents the data flow through six functional chains, focusing on two main streams, namely (i) the *safe-life* assessment (blue) and (ii) the *damage tolerance* assessment (red), both aligned with DT-SHM requirements from operational and system analysis procedures. Supporting chains are included for DT construction and data communication with OEMs/operators. The *damage tolerance* stream integrates crack-life prediction (orange) and manual crack inspection input during post-inspection (purple). The *safe-life* stream features the RUL assessment model (pink), crucial for initial state setup. Both main streams start after the "Assessment Type Classification" block.

(iv) MATLAB Implementation

The logical architecture of the DT-SHM was successfully validated through the process of defining requirements in Capella. A System Composer model was developed based on logical functions and functional exchanges, with requirements derived from missions and capabilities. These items were implemented in the System Composer architecture using a drag-and-drop approach, ensuring that all necessary missions and capabilities were included. The project was designed within a generic framework, allowing for technology options tailored to specific needs. The framework integrates sample Simulink behaviour linked to the requirements, facilitating traceability through the Property Inspector. The architecture, illustrated in the Simulink diagram, features a data management block that classifies assessment types and processes raw data. It also includes discrete and varying transfer functions for FEM representatives. The processed data is fed into a classified assessment process, where a sample fatigue accumulation model is applied through MATLAB function blocks [17]. Ten Simulink models were created and linked to their corresponding requirements in the Requirements Editor, and all test cases were successfully verified. The MATLAB Requirements Toolbox provided robust traceability and validation tools, finalising the system-level design before moving to the component phase, see Fig. 4.

Index	ID	Summary	Verified	Implemented
Digital_Twin_Land...				
Digital Twin LG Fa...				
1	#1	Provide Damage Tolerance An...		
1.1	#2	Predict Crack Development		
1.2	#3	Evaluate Confidence Value		
1.3	#4	Analyse FEM for Damage Toler...		
2	#5	Provide Safe-Life Analysis		
2.1	#6	Assess RUL		
2.2	#7	Evaluate RUL Safe-Life Factor		
2.3	#8	Analyse FEM for Safe-Life Anal...		
3	#9	Provide Feedback to Operators		
4	#10	Data Management		
4.1	#11	Acquire Online Data from Lan...		
4.2	#12	Assessment Type Classification		
4.3	#13	Raw Data Processing		

Fig. 4. Sample model validation and verification results

4.2 Discussion

The validation of the logical architecture in Capella was successful using the Arcadia method. However, the Simulink model validation in System Composer for the landing gear system is only partially functional. This is due to the incremental project's goal of creating a generic framework for DT-SHM, which introduces additional sub-requirements that vary the validation and verification processes in MATLAB. Different technology implementations lead to distinct subsystems and Simulink configurations. For instance, when manufacturers select DT-SHM technologies, the requirement table block can be integrated into Simulink components, allowing for more detailed logical requirements than those provided in the Requirements Editor. High-fidelity modelling technologies and data processing techniques can also be incorporated, such as using the Deep Learning Toolbox for PINNs [18] and machine learning principles for data processing with the Fast Fourier Transform method [19]. Meanwhile, the MATLAB function block allows for customisation when coding is necessary.

The use of Capella and MATLAB highlights the unique features of these MBSE tools. Capella excels in high-level architectural design, while System Composer and the Requirements Toolbox of MATLAB focus on detailed functions and performance testing. However, seamless integration between the two software platforms remains a challenge. Currently, Capella exports must be in XML format to be compatible with MATLAB, posing issues [13]. In the meantime, Thales, the developer of Capella, is developing a direct connection to MATLAB, but this is still in the development phase [20]. Therefore, the System Composer diagram for DT-SHM in this study was created manually in the MATLAB platform.

5. Conclusions and Future Work

This study successfully developed a DT-SHM framework for fatigue analysis using an MBSE architecture, supporting *safe-life* and *damage-tolerance* assessment methods while meeting regulatory requirements for all landing gear components. Validated and verified through MATLAB, the architecture provides a robust system-level model that facilitates technology integration in alignment with OEM specifications. According to the V-model, future work involves progressing to the component design phase, where the logical architecture is translated into a physical system design using Capella. First, physical components are assigned to logical subsystems, forming the integrated DT hardware system architecture. Then, the validation and verification processes are effectively conducted through MATLAB's Requirements Table feature, enhancing scenario testing details and completing the first half steps of the V-model. However, this only addresses one aspect of DT-SHM in CBM practice. To fully establish the DT-SHM framework, system and physical DT-SHM architectures for the shock absorber, corrosion, transient overload detection, and landing systems must be developed following the same methodology used for fatigue analysis to fulfil the definition of SHM for landing gear.

References

[1] Bickford J, Van Bossuyt DL, Beery P, Pollman A. Operationalizing digital twins through model-based systems engineering methods. Syst Eng. 2020;23(6):724-750. doi:10.1002/sys.21559.

[2] Shi T, Ompusunggu A. Shaping extended product design and development with predictive maintenance capability for digital servitisation. Procedia CIRP, Vol 128, pp 716-721, 2024, https://doi.org/10.1016/j.procir.2024.04.021.

[3] Forrest C, Forrest C, Wiser D. Landing gear structural health monitoring (SHM). Procedia Struct Integr. 2017;5:1153-1159. doi:10.1016/J.PROSTR.2017.07.025.

[4] Aerospace Landing Gear Systems Committee. Landing gear structural health monitoring. **Tech. Rep.** SAE International; 2019.

[5] What is MBSE?, https://mbse-capella.org/what_is_mbse.html, accessed: 2024-6-23.

[6] Leitenberger F, Gwosch T, Matthiesen S. Architecture of the digital twin in product validation for the application in virtual-physical testing to investigate system reliability. **2021**. doi:10.35199/dfx2021.10.

[7] R. Pecora, A rational numerical method for simulation of drop-impact dynamics of oleo-pneumatic landing gear, Appl. Sci. (Basel) 11 (9) (2021) 4136.

[8] S. Sivakumar, M. Syedhaleem, Non-linear vibration analysis of oleo pneumatic landing gear at touchdown impact, Math. Model. Eng. 4 (2) (2018) 89–97.

[9] W. Liu, Y. Wang, Y. Ji, Landing impact load analysis and validation of a civil aircraft nose landing gear, Aerospace 10 (11) (2023) 953.

[10] H. Yang, F. Wan, W. Cui, Failure simulation and identification of shock absorber in carrier-based aircraft landing gear, in: 2016 IEEE International Conference on Prognostics and Health Management (ICPHM), IEEE, 2016.

[11] Proper landing gear servicing for safe operations, https://safetyfirst.airbus.com/proper-landing-gear-servicing-for-safe-operations/, accessed: 2024-7-3 (Dec. 2022).

[12] Y. Ye, Q. Yang, J. Zhang, S. Meng, J. Wang, A dynamic data driven reliability prognosis method for structural digital twin and experimental validation, Reliability Engineering System Safety 240 (2023) 109543. doi:10.1016/J.RESS.2023.109543.

[13] Tutschka F. Creating a bridge between Capella and System Composer or Simulink. MathWorks. 2023. Available at: https://uk.mathworks.com/company/technical-articles/creating-a-bridge-between-capella-and-system-composer-or-simulink.html.

[14] Is Capella a SysML tool? MBSE Capella. Available at: https://mbse-capella.org/arcadia_capella_sysml_tool.html. Accessed August 11, 2024.

[15] Laflamme S, Ubertini F, Di Matteo A, Pirrotta A, Perry M, Fu Y, et al. Roadmap on measurement technologies for next generation structural health monitoring systems. Meas Sci Technol. 2023;34(9). doi:10.1088/1361-6501/acd135.

[16] New A, Gearhart A, Darragh R, Villafañe-Delgado M. Physics-informed neural networks for scientific modelling: uses, implementations, and directions. Proc SPIE. 2024;13051. doi:10.1117/12.3013520.

[17] Souto CD, Correia JA, De Jesus AM, Calçada RA. Fatigue damage tool (FDT) - a tool for fatigue damage assessment according to design codes. Procedia Struct Integr. 2019;22:376-385. doi:10.1016/j.prostr.2020.01.047.

[18] Raymond S. Physics-Informed Machine Learning: Using the Laws of Nature to Improve Generalized Deep Learning Models. MathWorks. 2021. Available at: https://uk.mathworks.com/videos/physics-informed-machine-learning-using-the-laws-1622124810574.html. Accessed August 23, 2024.

[19] Transform Time-Domain Data into Frequency Domain - MATLAB & Simulink. MathWorks. 2018. Available at: https://uk.mathworks.com/help/dsp/ug/transform-time-domain-data-into-the-frequency-domain-using-the-fft-block.html. Accessed August 23, 2024.

[20] Nowodzienski P. How to transform Capella model into Matlab's Simulink model. MBSE Capella Forum. 2022. Available at: https://forum.mbse-capella.org/t/how-to-transform-capella-model-into-matlabs-simulink-model/5333. Accessed August 31, 2024.

[21] Tuegel, E. J., Ingraffea, A. R., Eason, T. G., & Spottswood, S. M. (2011). Reengineering aircraft structural life prediction using a digital twin. Int. J. Aerosp. Eng. https://doi.org/10.1155/2011/154798

[22] European Union Aviation Safety Agency (EASA). (2023). CS 25.571 - Damage tolerance and fatigue evaluation of structure. In Easy Access Rules for Large Aeroplanes (CS-25). Retrieved from https://www.easa.europa.eu.

[23] Hoole, J. et al. (2020). A Framework to Implement Probabilistic Fatigue Design of Safe-Life Components. In: *ICAF 2019 – Structural Integrity in the Age of Additive Manufacturing*. Springer, Cham. https://doi.org/10.1007/978-3-030-21503-3_25