engineering proceedings

MDPI

Proceeding Paper

Integration of Functional Mock-Up Units into Digital Twins of Aircraft Thermal Management Systems †

Tobias Reischl *, Corentin Lepais and Raphael Gebhart

German Aerospace Center (DLR), Münchener Straße 20, 82234 Wessling, Germany; corentin.lepais@dlr.de (C.L.)
* Correspondence: tobias.reischl@dlr.de

† Presented at the 15th EASN International Conference, Madrid, Spain, 14–17 October 2025.

Abstract

Hybrid-electric regional aircraft require detailed thermal management digital twins to assess performance and feasibility while reducing physical test effort. The Functional Mock-Up Interface (FMI) enables partners to exchange subsystem models as Functional Mock-Up Units (FMUs) for gate-to-gate simulation while preserving intellectual property. However, FMU integration introduces numerical coupling challenges, interface overhead, and potential loss of accuracy depending on the integration method. Benchmarking against a DLR Thermofluid Stream (TFS) reference model showed that FMU-based co-simulation can significantly increase computational effort, specifically from 8 min up to 2.5 h. Control-based integration further implicates transient deviations due to filtering, although steady-state accuracy generally remains unchanged. Therefore, it is mandatory to evaluate and compare FMU integration strategies to show that digital twin performance targets remain achievable when design, solver settings, and filtering are only applied selectively and systematically. The results show clear design guidance: employ native fluid libraries when possible for speed and accuracy, use FMU paired with adapters and without filters for accuracy, and reserve filtering for numerical stabilization only. Using a control approach to integrate the FMU improves simulation speed compared to adapters but introduces a small error, which in turn reduces simulation accuracy.

Keywords: digital twin; thermal management system; Functional Mock-Up Interface; Functional Mock-Up Unit; modelica; vapor cycle system; environmental control system

Academic Editors: Spiros Pantelakis, Andreas Strohmayer and Gustavo Alonso

Published: 20 April 2026

1. Introduction

Hybrid-electric regional aircraft must manage thermal loads in the megawatt range, far above today's tens-of-kilowatts levels [1,2], which arise mainly from high-power batteries, fuel cells, and power electronics. The European initiative TheMa4HERA develops and validates corresponding thermal management concepts up to TRL 5 within a system-level digital-twin framework [2,3], enabling early architectural feasibility checks by dynamic gate-to-gate mission assessments. The digital twin includes the complete thermal management system, including the environmental control system (ECS), enabling virtual demonstration before hardware availability [4]. Subsystem models are exchanged via the FMI standard [5] and provided as FMUs [6], which DLR integrates into the overall framework [4]. Integrating high-fidelity FMUs introduces numerical-coupling challenges, solver interactions, and potential accuracy deviations, especially for complex systems such as the ECS. This paper, therefore, compares alternative FMU-integration strategies regarding accuracy, CPU time, and robustness. Section 2 presents the vapor cycle system

 https://doi.org/10.3390/engproc2026133023

and FMU-generation approaches, Section 3 outlines the integration strategies and Section 4 provides the comparative evaluation.

2. Vapor Cycle System

This section introduces the VCS model, which serves as the baseline for comparing the different FMU integration methods. An overview of these methods is provided, along with a description of the specific modifications of the model required for FMU generation in each approach.

2.1. Model Description

The VCS forms part of the electrified ECS pack architecture investigated in TheMa4HERA and conditions the cabin supply air. It cools the fresh air delivered by an electric compressor and rejects the associated waste heat to the environment via a ram-air duct, which together define the subsystem's inlet and outlet boundary conditions. Heat transfer between fresh-air and ram-air streams is achieved through an integrated vapor-compression loop—including compressor, condenser, expansion valve, and evaporator, as used in recent ECS concepts with supplemental VCS cooling [7,8], see Figure 1. The subsystem provides thermodynamic states, pressure, temperature, and water mass fraction [9,10]. Following established VCS control strategies, the fresh-air outlet temperature is regulated by the compressor operating point, which sets the high- and low-pressure saturation levels. The expansion valve is controlled to maintain a prescribed refrigerant superheat at the evaporator outlet, ensuring stable operation and effective heat exchange [11]. The reference model, shown in Figure 1, is implemented using the TFS library [12–14]. It avoids adapters and FMUs, simplifies integration, and reduces computational cost, and it serves as the baseline against which all subsequent methods are compared.

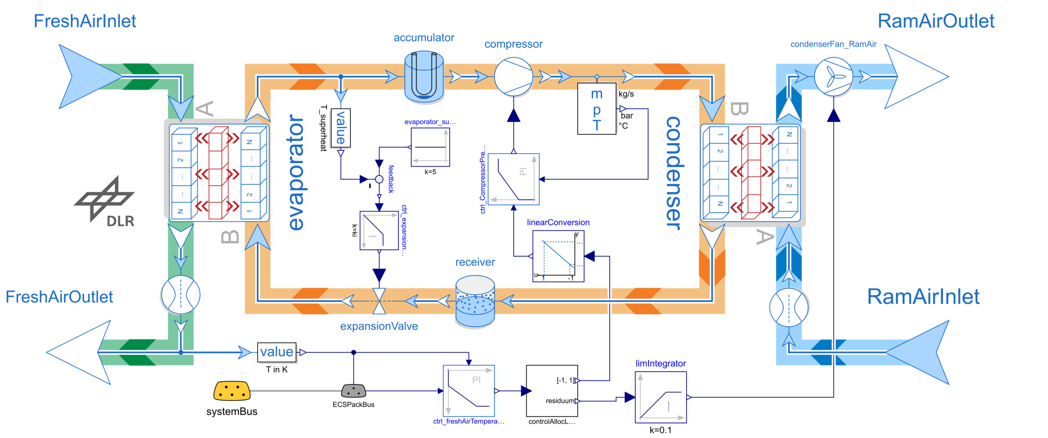


Figure 1. Reference VCS model using the DLR TFS library.

2.2. Methods Overview

Figure 2 illustrates three alternative methods for integrating a common reference model. All approaches are compatible with both Modelica models and FMUs. The first method depicted in the diagram is the control-like approach, detailed in Section 3.2 and whose FMU generation model is described in Section 2.4. The second method relies on adapter, as discussed in Section 3.1, with its FMU generation model presented in Section 2.3. An extension of this method incorporates a filter between the FMU and the adapter, as shown in Figure 3.

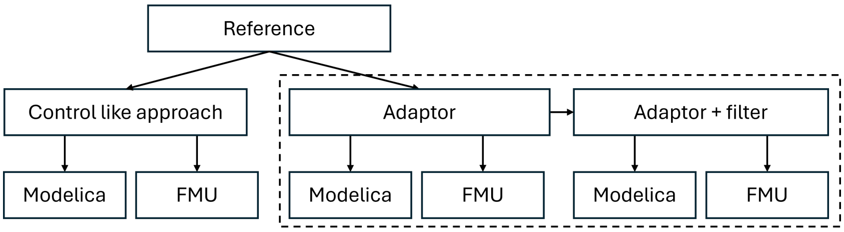


Figure 2. Overview of the different integration methods.

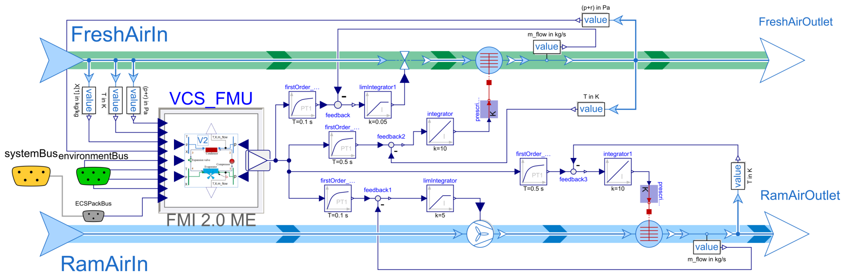


Figure 3. Integration of the VCS FMU with the control-like approach.

2.3. FMU Generation with Adapters

The VCS model shown in Figure 1 needs to be modified before being exported as a FMU. More precisely, acausal inlet/outlet TFS connectors must be replaced by causal real inputs/outputs. This conversion is done by the adapters, shown in Figure 4. On one side, the adapter can interface with the TFS connectors, and, on the other side, it provides flow, potential and thermodynamic state as real signals. The development of such an adapter for the TFS library is still ongoing and will not be further explained in detail in this paper. However, the key point is the tight coupling between the adapter, model development, and integration method, each step demanding a deep understanding of the final simulation environment and its requirements.

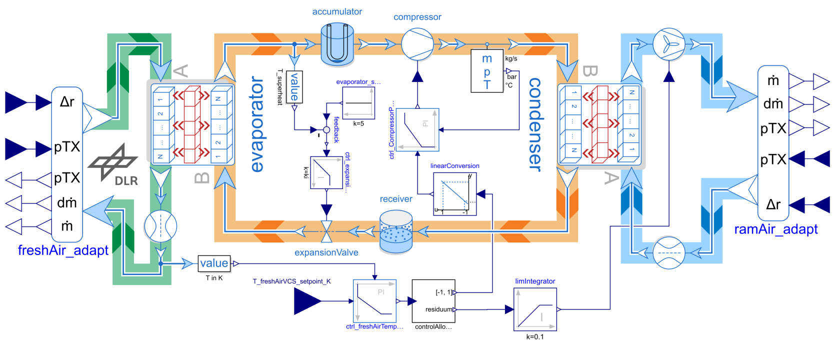


Figure 4. VCS model for FMU export with adapter as interface.

2.4. FMU Generation with Sources, Sinks and Sensors

Another approach to generate an FMU from Figure 1 is to use boundary components and sensors to define the interface of the subsystem. Components such as sources and sinks can be used to define the inlet conditions (temperature, pressure, mass fraction) and the outlet potential (pressure in the case of the TFS sink), while sensors provide the resulting behavior of the model under these conditions (mass flow, outlet temperature and outlet absolute humidity). This second approach is easier to implement than the solution with adapters because it does not require extensive knowledge about adapters and the simulation target, but only knowing which boundary conditions can be provided to the model. In this case, the pressure at the outlet of the fresh-air channel is defined by the cabin pressure and therefore should be an input for the VCS, which will provide the resulting mass flow. The resulting model is shown in Figure 5.

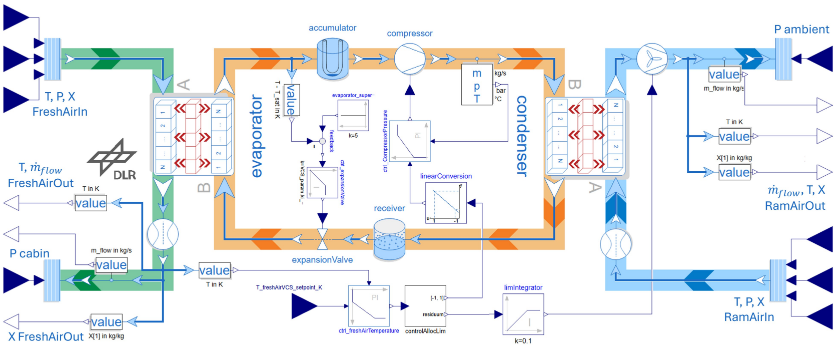


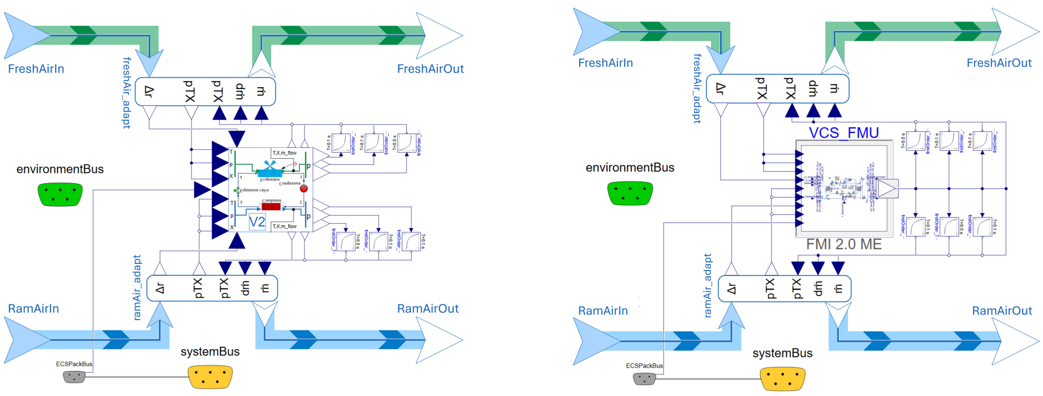
Figure 5. VCS model for FMU export with source/sink/sensor interface.

3. Integration Methods

The integration of FMUs into the digital twin reverses the adaptations described in Sections 2.3 and 2.4, i.e., the conversion of real inputs/outputs connectors into acausal TFS connectors, to be able to connect the imported models with the rest of the digital twin model. On one hand, the adapters, presented in Section 2.3, can be used as an integration tool, as described in Section 3.1. On the other hand, the export model presented in Section 2.4 can be integrated within a control-like framework, as described in Section 3.2.

3.1. Adapter Approach

The model from Figure 1 can be re-built using the model from Figure 4 and the TFS adapters as an interface. It yields the same results, yet the internal system is grouped into a single component block with defined input and output ports. These ports can be connected to a first-order filter before connecting to adapters. The filters are often not necessary, but can be used to break algebraic loops. The adapters ensure a physically consistent connection between the FMU and the surrounding models by the acausal connectors into flow and potential variables and then into acausal connectors inside the FMU itself. The resulting integration is shown in Figure 6 with the integration of the Modelica model from Figure 4 (Figure 6A) and with the corresponding exported FMU (Figure 6B). Figure 6A is used to assess the effect of the integration method itself, while Figure 6B validates that the method works when used with a real use case and highlights the effects of the FMU. Both are displayed with the first-order filter put in place, but will also be simulated without these first orders.



(**A**) Modelica model (exemplary use case) (**B**) FMU (real use case)

Figure 6. Integration of the VCS export model from Figure 4 using adapters.

3.2. Control-like Approach

Gebhart et al. [4] introduce a control-based method for interfacing an FMU with a Modelica library, the DLR TFS library. The FMU outputs are interpreted as setpoints driving the TFS components to reproduce the FMU behavior. Their example, an FMU of an outflow valve connected to a TFS valve with an I-controller, demonstrates the method while being simple: the actuator choice is clear, and the tuning of the single-input single-output (SISO) control loop remains straightforward. The VCS FMU, in contrast, represents a full subsystem that simultaneously influences temperature, mass flow, and, when a dehumidifier is present, absolute humidity. This results in a multiple-inputs multiple-outputs (MIMO) problem requiring multiple TFS components to reproduce the FMU outputs. The fully integrated FMU is shown in Figure 3. For each air stream, the FMU provides temperature and mass flow; hence, one flow-control actuator (valve, fan, compressor, . . .) and one temperature-control actuator (conduction element or heat pipe) are required. The choice of the flow-control actuator depends on whether the FMU's internal flow is actively generated or arises from passive pressure losses and the components available in the library. In the configuration in Figure 1, fresh air is driven only by pressure losses; therefore, a valve is most suitable while ram air is actively driven and can be mimicked with a fan. Temperature control uses a generic conduction element in both cases. Because these actuators are arranged in series, their order must take into account their physics. Active flow-control devices influence downstream temperature; therefore, temperature control is placed downstream to ensure correct outlet conditions. The MIMO control problem is split into parallel SISO loops, one for each actuator. The FMU output defines the setpoint, a filtered sensor provides the controlled variable, and an I-controller generates the actuator command (fan speed, valve position, or conduction-element temperature). Although feeding the FMU temperature directly into the conduction element is plausible, this open-loop strategy would introduce a steady-state offset due to the finite heat-transfer coefficient. Increasing the coefficient reduces the offset but destabilizes the simulation, making closed-loop control the best option, offering both robustness and accuracy. All I-controllers are tuned to minimize residual tracking error without inducing oscillations.

4. Results

To evaluate the architectures introduced in Section 3 against the defined benchmarks, the following sections present the resulting CPU times and errors of the simulations.

4.1. Description of the Simulation Test Case

The models introduced in Section 3 are assessed by simulating the complete digital twin of the ECS over a gate-to-gate flight mission. The corresponding altitude and speed profiles are provided in [3,4]. This mission profile enables evaluation of the integrated FMU models under both steady-state conditions (e.g., stabilized cruise) and transient phases (e.g., take-off or flight-phase transitions). Using this setup, it is possible to quantify the additional transient error introduced by the different integration methods, the increased model complexity in steady-state operation, and the impact on simulation time throughout the mission.

4.2. Simulation Time

Simulation time, often referred to as CPU time, is a critical limitation for high-fidelity simulations, as detailed dynamic models can lead to significant computational cost and prolong repetitive study. Therefore, reducing CPU time is an essential objective. Since the models presented in Section 3 are implemented in Modelica, the CPU time can be recorded automatically. Figure 7 presents the results for the different integration methods shown in Figure 2, with native Modelica implementations on the left and FMU-based implementations on the right. This provides a direct comparison of the computational effort associated with each approach. The x-axis represents the full gate-to-gate mission duration, from taxi-out to taxi-in, while the y-axis shows the corresponding CPU time.

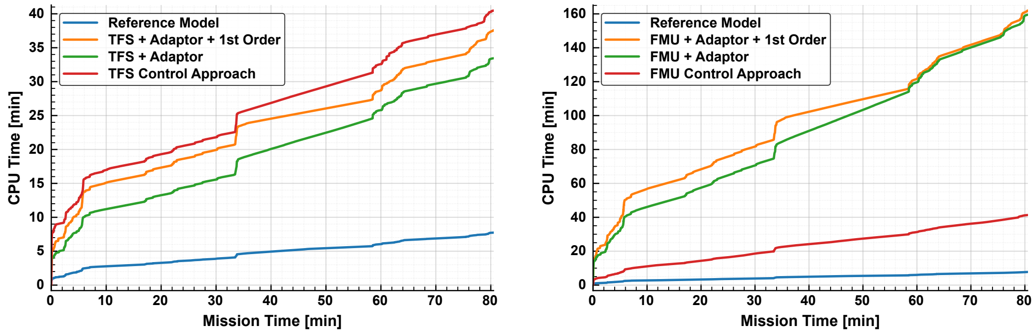


Figure 7. Comparison of CPU time for the TFS and FMU models.

The blue curve, representing the reference model shown in Figure 1, requires approximately eight minutes for a complete simulation run. On the TFS side, adding the adapter increases CPU time by roughly 30 min, and enabling the first-order filter increases it by an additional 15%. However, the filter also accelerates computation during quasi-steady regions, most notably during cruise around 35 min, where the slope flattens due to reduced output variability. On the FMU side, the control-based approach results in nearly the same CPU time as the corresponding TFS implementation, with both finishing in about 42 min. In contrast, combining the FMU with an adapter increases CPU time by up to two additional hours. Since the VCS is only one subsystem within a larger digital twin, such overhead poses a severe scalability constraint, particularly when multiple FMUs are integrated in the same simulation environment.

4.3. Error

The simulation error is defined as the deviation from the reference model. Figure 8 illustrates this deviation for each architecture. The x-axis again represents mission time, but is limited to the period from taxi-out to the beginning of cruise, highlighting transient behavior before reaching steady-state conditions. The y-axis shows the deviation in cabin temperature in Kelvin. The most noticeable result is that the control-based approach (green curve) exhibits the largest error, reaching up to approximately 0.5 K, which can be attributed to the additional dynamics introduced by the control loops. Furthermore, there is no significant difference between the TFS and FMU variants. As expected, applying a first-order filter introduces additional deviation, although this is mainly limited to transient phases.

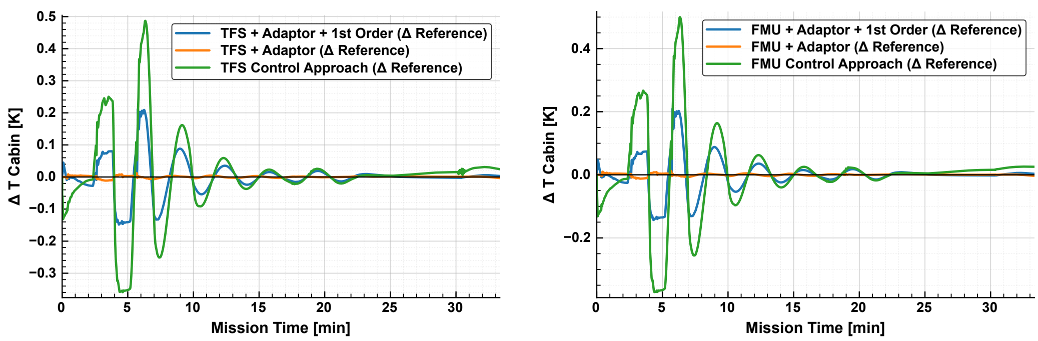


Figure 8. Comparison of cabin temperature deviations for the TFS and FMU models.

5. Discussion

Two strategies for integrating FMUs into the thermal management simulation framework were evaluated: the adapter-based approach and the control-like approach. The adapter method enables direct physical signal exchange and avoids the need for control actuators or explicit filtering. However, it requires adapter models that match the modeling format (e.g., pressure as a potential variable). Regarding initialization, the control-like approach requires consistent initialization between the controllers and the FMU, whereas the adapter approach relies solely on FMU initialization to ensure consistent start values. At the same time, adapter-based coupling may introduce algebraic loops, resulting in additional nonlinear equations that can make initialization more challenging for the solver. These loops can be broken using filters, which then themselves require consistent initialization. In contrast, the control-like approach inherently relies on signal filtering and closed-loop control. This introduces additional dynamics and may cause slight transient delays and deviations relative to the FMU output. By carefully tuning the controllers, these deviations can be minimized and remain limited to transient behavior. Steady-state accuracy can be achieved through the use of I-controllers.

During the study, it was also observed that the adapter-based approach can significantly increase computational time due to the additional implicit nonlinear equations generated by the current TFS library implementation. This effect should be investigated further, as it may explain the considerable increase in CPU time.

6. Conclusions

A comparison of the FMU integration strategies shows clear trade-offs between computational performance and accuracy. The adapter-based configuration resulted in the highest computational cost, with runtimes up to 150 min. Corresponding to approximately a nineteen-fold increase in CPU demand, compared to the reference model. Such overhead is critical in large-scale simulation frameworks, where iterative studies and design variations are required. In contrast, the control-like approach yielded nearly identical runtimes for the TFS and FMU versions. Adding signal filters increased runtime by approximately 15%, although it improved numerical behavior during steady-state operation. With respect to accuracy, the control-based method showed the largest transient deviations due to filtering and closed-loop dynamics, resulting in noticeable—but transient—temperature errors while maintaining negligible steady-state deviation.

In summary, the control-like approach was the fastest when FMUs were applied, whereas the adapter-based configuration without filtering provided the highest accuracy but at substantially increased computational cost.

Author Contributions: Conceptualization, T.R., C.L. and R.G.; methodology, T.R., C.L. and R.G.; software, T.R. and C.L.; validation, T.R.; formal analysis, T.R. and C.L.; investigation, T.R. and C.L.; resources, T.R. and C.L.; data curation, T.R. and C.L.; writing—original draft preparation, T.R. and C.L.; writing—review and editing, R.G.; visualization, T.R. and C.L.; supervision, R.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was carried out as part of the European project TheMa4HERA. The project is supported by the Clean Aviation Joint Undertaking and its members. This research was funded by the European Union under grant agreement No. 101102008.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Models and simulation results belong to the TheMa4HERA project under Grant Agreement No. 101102008 and cannot be shared without explicit consent of the consortium.

Acknowledgments: The authors thank the partners of the TheMa4HERA consortium for providing inputs and discussions that supported this work.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

ECS	Environmental Control System
FMI/FMU	Functional Mock-Up Interface/Unit
VCS	Vapor Cycle System
TFS	Thermofluid Stream Library

References

1. Sabet, S.; Gumprich, W.; Moeller, M.; Felgueroso, A.; González Nieves, I.; Díaz, M.; Mancin, S. Numerical Analysis of Potential Energy Recovery via a Thermoelectric Generator (TEG) for the Next-Generation Hybrid-Electric Regional Aircraft. *Eng. Proc.* **2025**, *90*, 64. [CrossRef]
2. Clean Aviation Joint Undertaking. TheMa4HERA: Thermal Management for the Hybrid Electric Regional Aircraft. Project Duration: 2023–2026; Grant Agreement No. 101102008; CORDIS ID 101102008. 2023. Available online: https://www.clean-aviation.eu/research-and-innovation/clean-aviation/clean-aviation-projects/thema4hera (accessed on 31 October 2025).
3. Lepais, C.; Zimmer, D. Prototypical Control for the Digital Twin of Aircraft Environmental Control System. In Proceedings of the Modelica Conferences, Lucerne, Switzerland, 8–10 September 2025; pp. 81–90.
4. Gebhart, R.; Lepais, C. Novel Control-like Approach for the Robust Integration of Functional Mock-Up Units into Digital Twins. *Eng. Proc.* **2025**, *90*, 12. [CrossRef]
5. Blochwitz, T.; Otter, M.; Åkesson, J.; Arnold, M.; Clauss, C.; Elmqvist, H.; Friedrich, M.; Junghanns, A.; Mauss, J.; Neumerkel, D.; et al. Functional mockup interface 2.0: The standard for tool independent exchange of simulation models. In Proceedings of the 9th International Modelica Conference, Munich, Germany, 3–5 September 2012; The Modelica Association: Linköping, Sweden 2012; pp. 173–184.
6. Fang, J.; Luo, M.; Wang, J.; Hu, Z. FMI-Based Multi-Domain Simulation for an Aero-Engine Control System. *Aerospace* **2021**, *8*, 180. [CrossRef]
7. Ablanque, N.; Torras, S.; Rigola, J.; Oliet, C. Dynamic Model of Vapor Compression Systems for Aircraft Environmental Control Systems. *J. Aircr.* **2024**, *61*, 1387–1396. [CrossRef]
8. Ablanque, N.; Satrio Loka, N.R.B.; Torras, S.; Gurumurthy, S.; Rigola, J.; Oliet, C.; Couckuyt, I.; Dhaene, T.; Monti, A. Vapor Compression System Data-Driven Surrogate Models for Aircraft Environmental Control Systems. *Int. J. Refrig.* **2025**, *178*, 336–346. [CrossRef]
9. Eschenbacher, P.W.; Zimmer, D.; Weber, N. First Principle Model of an Electric ECS Pack. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2022; Volume 1226, p. 012069. [CrossRef]
10. Jennions, I.; Ali, F.; Esperon Miguez, M.; Camacho Escobar, I. Simulation of an Aircraft Environmental Control System. *Appl. Therm. Eng.* **2020**, *172*, 114925. [CrossRef]
11. Michalak, T.; Emo, S.; Ervin, J. Control Strategy for Aircraft Vapor Compression System Operation. *Int. J. Refrig.* **2014**, *48*, 10–18. [CrossRef]
12. Zimmer, D.; Meißner, M.; Weber, N. The DLR ThermoFluid Stream Library. *Electronics* **2022**, *11*, 3790. [CrossRef]
13. Zimmer, D. Robust object-oriented formulation of directed thermofluid stream networks. *Math. Comput. Model. Dyn. Syst.* **2020**, *26*, 204–233. [CrossRef]
14. Gebhart, R.; Jordan, P.; Stein, P.; Junglas, P.; Weber, N.; Eschenbacher, P.; Zimmer, D. Community Updates to the DLR ThermoFluid Stream Library. In Proceedings of the Modelica Conferences, Lucerne, Switzerland, 8–10 September 2025; pp. 51–55.

https://doi.org/10.3390/engproc2026133023