

Modelica Association Standards and Surrogate Modeling to Enable Multi-Fidelity Simulations

Olle Lindqvist¹ Robert Hällqvist^{2,4} Raghu Chaitanya Munjulury^{3,4}

¹FCS Verification & Validation, Saab Aeronautics, Sweden

²System Simulation and Concept Development, Saab Aeronautics, Sweden, robert.hallqvist@saabgroup.com

³Technical Management & Maintenance, Saab Aeronautics, Sweden

⁴Division of Fluid and Mechatronic Systems (FLUMES), Linköping University, Sweden

Abstract

System simulations are particularly useful when analyzing complex systems. Simulations are often cheaper and safer than physical tests of the actual system(s) of interest. Models can additionally be created for systems that do not exist to find solutions that are impossible to analyze experimentally in, for example, early life-cycle stages. Models used in system simulations require appropriate input data to give results with the required fidelity and, in the end, credibility. Integration is often challenging as each system commonly constitutes contributions from several engineering domains. Relying on relevant open standards for information exchange is seen as a means of mitigation. The results of the presented work encompass a developed methodology that allows [Computational Fluid Dynamics \(CFD\)](#) results to be integrated into a simulator using system identification and open standards. [Reduced Order Models \(ROMs\)](#) are generated based on results from a [CFD](#) analysis. These [ROMs](#) are coupled to lumped parameter system simulation models through the mechanisms of the [System Structure and Parameterization \(SSP\)](#) and [Functional Mock-up Interface \(FMI\)](#) standards. In addition, several important factors to consider before using the proposed methodology are presented. These include the intended use of the [ROMs](#), knowing the flow inside the system, what resources are available, and any potential licensing issues

Keywords: FMI, SSP, CATIA, CFD, System Identification, Neural Networks, Co-simulation

1 Introduction

Complex systems often need to be analyzed using models in order to exploit, understand, and manage emergent behavior. The use of models and simulations instead of physical experiments to analyze systems introduces a number of different benefits in relation to these needs. For instance, models can be created as cheaper and safer alternatives to testing the corresponding physical systems. It is also possible to create models, and multiple models coupled in simulators, of systems that do not exist yet,

for which experimentation would be impossible (Ljung and Glad 2003). Dynamic simulation models are models which show how modeled system properties change over time (Ellner and Guckenheimer 2011). Such models are typically described mathematically by differential and algebraic equations. Models in general are typically developed to fulfill some specified purpose. This purpose often requires information gathered from different engineering disciplines. To achieve this exchange efficiently, standardized means to communicate and share digital artifacts across modeling domain boundaries are necessary (Hällqvist, Munjulury, et al. 2021).

At Saab Aeronautics, dynamic models are used during the development of many aircraft systems and sub-systems, both on their own in local desktop environments but also in various co-simulation constellations at different levels of abstraction (from hereon referred to as simulators) (Hällqvist 2023; Steinkellner 2011; H. Andersson 2012). The sharing of digital artifacts between different engineering domains (be it models, simulators, or simulation results) is often challenging partly as a consequence of a need to use the best-suited software, deployed on the most suitable target, for each engineering domain to address different simulation purposes. The export and integration of models have typically, for example, the *Gripen E program* (Saab Group 2023), been performed through in-house developed standards. While these standards are used successfully, maintaining them is difficult and a time-consuming process. Native tool support for in-house standards, not adopted by the community as a whole, is challenging to motivate tool vendors as the tool vendors are bound by their overall customer needs. This challenge only becomes bigger if considering the long life cycles of aircraft as life-cycle information needs to be aggregated, with traceability links, and made available to the end user and decision-makers. Failure to address any of these highlighted aspects may, in the end, lead to sub-optimal designs founded on simulation results with in-accurate credibility (Hällqvist, Munjulury, et al. 2021).

1.1 Contributions

The presented work provides a summary of a master thesis project conducted at SAAB Aeronautics & Linköping University (Lindqvist 2022). A main result of the work is a proposed methodology for creating portable ROMs based on CFD results. In addition, several important factors to consider before using the proposed methodology are presented. These include the intended use of the ROM, a need to know the flow to be analyzed, what resources are available both during model development and end-use, licensing set-ups of tools to be used, etc. The results of the work demonstrate that steady-state data, of internal flow systems, can be used as intended for transient system simulations. The ROMs created using the method gave results that were generally close to the corresponding handbook equations available in the literature, see for example Miller (D. Miller 1990) for a comprehensive theoretical background on internal flow systems.

2 Theoretical Background

This paper incorporates techniques from several fields of research and industry: data-driven methods (system identification) for surrogate modeling purposes, physics-based lumped parameter modeling, CFD, and standardized co-simulation. The work strives to employ a set of standards, exemplified by a set of relevant tools, to address a need identified in the industry. This need is summarized through a development methodology where the best-suited domain-specific tool is used for each engineering task, where each contributing digital artifact and the model-based decision is fully traceable to the end result.

2.1 Utilized Standards

Three different standards jointly enable the interoperability presented herein: the FMI standard (FMI Development Group 2020), the SSP standard (Modelica Association 2019), and the ISO 10303-21 standard also known as *Standard for the Exchange of Product Data (STEP)* (International Standards Organization (ISO) 2016). All these three specifications strive to provide formats for a neutral, vendor-independent, and platform-independent information exchange and model-based decision-making.

The primary objective of the SSP standard is to establish standardized means for linking simulation models; in the end resulting in portable and executable set of coupled models. This standard offers a nested hierarchical definition of systems and subsystems included within the set. In contrast, the FMI standard focuses on facilitating the exchange of the constituent models, and their interfaces, specific to different engineering domains. The FMI standard additionally provides mechanisms for flexible management of *Intellectual Property (IP)* and simulation execution. The SSP standard provides a framework for standardized connection, configuration, and exchange of a connected group of models, which collectively form,

what here is referred to as, a simulator application. To convey this information, the SSP standard outlines various *Extensible Markup Language (XML)* schema. The composition of the simulator is stored in the *System Structure Description (SSD)* format, the parameter values in the *System Structure Parameter Values (SSV)* format, and the associations of these values with the individual executable models in the *System Structure Parameter Mapping (SSM)* format. All these different artifacts are exploited in the presented research. The ISO 10303-21 standard specifies a standardized file format for exchanging and representing product data across different *Computer Aided Design (CAD)* and *Computer-Aided Manufacturing (CAM)* systems, particularly focusing on data capturing three-dimensional geometric representations. STEP is, unlike the FMI and SSP standards, not an open standard available to all engineers. It is however widely adopted by tool vendors in the geometry and CFD domains. In this work, the STEP format is exploited to exchange geometry models between these two engineering domains. There are several additional features presented in the STEP specification that could contribute to achieving traceable and credible simulations, for example, the *Application protocol 209* for specification and exchange of solver related information (Lanza et al. 2018). These features are however not considered herein, and investigations of their usefulness within the presented context is left for future research.

2.2 Modeling and Simulation Tools

The work expands on a simulator that has been successively developed by Saab Aeronautics in order to capture and communicate industrial requirements on Modelica Association Standards, and the corresponding tool support, that jointly provide technology deduced as essential when developing complex systems (Hällqvist, Naeser, et al. 2022; Lind and H. Andersson 2011). The simulator in focus, see Hällqvist et al. for a detailed description (Hällqvist, Munjulury, et al. 2022), incorporates digital artifacts developed in Dymola (*Dymola User Manual* 2016), OpenModelica (Fritzson et al. 2005), Matlab/Simulink, and CATIA. This work adds high-fidelity information, obtained through CFD analysis, conducted with the Altair suite of tools. The CFD simulations were performed using the *Finite Element Method (FEM)*-based Altair AcuSolve CFD-solver (Altair 2023). Furthermore, meshing and post-processing were performed using the built-in tools of AcuSolve.

2.3 Systems simulation

In order to simulate complex systems using mathematical models, it is necessary to use some form of computer software. Modelica is one of the, at Saab Aeronautics, commonly used software languages for physics-based modeling and simulation. Physics-based system simulation models are constructed by connecting components, or blocks, in order to transfer information between them

(Modelica Association 2023). Components for use in creating the systems are contained in various modeling libraries.

The simulation of a system is performed by first converting the code of the different components into a system of differential equations. First, the equations are sorted based on the flow of information between them. Second, the system of equations are simplified in order to reduce the computational time. Finally, the equations are solved numerically (Fritzson 2003). Special care needs to be taken if the system contains both fast and slow dynamics. This can cause the solution of the differential equations to become inaccurate or unstable (Ljung and Glad 2003).

2.4 System Identification

System Identification is the process of taking some higher-order data and creating a ROM based on it. The data used could be experimental, from in-situ measurements, or virtual from some higher fidelity simulation. ROMs can be created through various different methods, many of them described in detail by Ljung in (Ljung 1999b). The system identification procedure has three main components (Ljung 1999a): a data set, one or more candidate models to describe the relationship between input and output, and some selected technique for evaluating which model best fits the data. As an example, consider some time-series data with recorded inputs $u(t)$ and outputs $y(t)$. One way of modeling the relationship between them is through a simple difference equation

$$y(t) + a_1y(t-1) + \dots + a_ny(t-n) = b_1u(t-1) + \dots + b_nu(t-n) \quad (1)$$

where a and b are some unknown parameters. In order to calculate $y(t)$, it can simply be isolated on the left-hand side resulting in

$$y(t) = \phi^T(t)\Theta \quad (2)$$

where $\Theta = [a_1 \dots a_n \quad b_1 \dots b_n]^T$ and $\phi(t) = [y(t-1) \dots y(t-n) \quad u(t-1) \dots u(t-n)]^T$. The goal of the system identification procedure is to find the values of the unknown parameters in Θ so that $y(t)$ fits with the recorded data. This can be done through, e.g., statistical methods or machine learning algorithms (Sjöberg et al. 1995). Thus $u(t)$ and $y(t)$ are the data set, Equation 1 is the candidate model, and the method used to calculate Θ is the final step in the above list.

2.5 Neural Networks to realize ROMs

One of the methods for identifying data-driven mathematical models is to train a neural network (Chen, Billings, and Grant 1990) on available data. Neural networks are machines or computer software that solve tasks by imitating the way a brain works. The neural network is built up of interconnected neurons, or nodes. A neural network

needs to be trained in order to gain the necessary knowledge to solve a problem. This is done by modifying the strength, or *weight*, of the connections between the neurons. Each neuron has an activation function that determines the output of the neuron based on the strength of the input signals entering it. Some common activation functions are the **Rectified Linear Unit (ReLU)**, logistic, and hyperbolic tangent functions (Haykin 1999). Neural networks can be useful for training on non-linear problems and for mapping the input and output signals of unknown systems. This makes them well suited for system identification tasks (Haykin 1999). There exist many examples in the literature of system identification performed using, for example, **Multi-Layer Preceptrons (MLPs)** (A. Parlos et al. 1991) (Fernandez, A. G. Parlos, and Tsai 1990).

2.6 Computational Fluid Dynamics

The Navier-Stokes equations are the partial differential equations that describe the motion of viscous fluid substances. Since there exist no known analytical solutions to the Navier-Stokes equations, they have to be solved numerically. This is known as **CFD**. In **CFD**, the continuity and Navier-Stokes equations are spatially, and sometimes temporally, discretized to allow for iterative, numerical solutions to be computed (Anderson, John D. 1995). There are several **CFD** methods available, with two of the most commonly used being the **Finite Volume Method (FVM)** and the **FEM**. The difference between these two different methods lies in how the governing equations are discretized.

For the **FEM**, the weak forms of the governing equations are discretized over the entire fluid domain through, for example, the Galerkin method (Donea, Jean and Huerta, Antonio 2003) (Fontes 2018). However, the convective term $(\nabla \cdot u)u$ is non-linear and gives non-symmetrical coefficient matrices, a problem that only gets bigger with increasing Reynolds numbers and turbulent flows. Thus, special techniques need to be used in order to stabilize the solution (Bathe, Klaus J. 2014). In contrast, the finite volume method discretizes the equations for each control volume (mesh element). This means that the solution will be stable because the flow will naturally be conserved for each element. From a practical point of view, **FEM** can achieve a higher order of accuracy for the discretization, however, this will also lead to a higher computational cost (Fontes 2018).

In order to spatially discretize the fluid domain, there are several mesh element types to choose from. 2D elements can be, for example, triangular or quadrilateral, while their 3D counterparts can be tetrahedral or hexahedral, etc. The choice of element type depends on the geometry of the domain. Triangular and tetrahedral elements are better at capturing curved and complex geometries, while quadrilaterals and hexahedrals can cover the same domain with fewer elements (Versteeg and Malalasekera 2007).

3 Method

This section describes how the work was carried out and how the previously described theory was tailored and applied to the use-case of this paper. The section covers the three main areas of the presented research: **CFD**, system identification, and integration of information from contributing disciplines. A description of the resulting application example simulator is also included.

3.1 Application example

The implementation of the targeted aircraft cooling system Modelica model is described in detail by Hällqvist et al. in (Hällqvist, Munjulury, et al. 2021). This particular model represents one essential part of a broader simulator that incorporates models and information from the engineering domains of hardware and physics-based modeling, control development and software modeling, architecture and requirements modeling, and geometry modeling using 3D **CAD**. The aircraft cooling system constituent piping and the corresponding internal flow is in focus here. The pipe component pressure drop is modeled as a function of the mass flow,

$$\Delta p = \frac{(z + c \cdot l/D)}{A^2 \cdot 2\rho} \dot{m}^2, \quad (3)$$

as described by, e.g., Miller in (D. S. Miller 1990). Here z is a parameter used to account for the pressure loss of pipe features such as bends and changes in the area, c is the friction coefficient, A is the pipe's cross-sectional area, and l is the pipe length. The pipe is connected to a heat exchanger at its outlet and a consumer of cooling power at the inlet. The inputs to the pipe inlet are pressure, mass flow, and enthalpy. Among the specified component outputs are pressure loss, fluid density, viscosity, etc. The parameters for the different system simulation components are automatically imported from the **CAD** geometry of the system. The Modelica model has been implemented in Dymola, using the in-house developed component library Modelica Fluid Lite (Eek, Gavel, and Ölvander 2017). The boundary conditions for the simulation consist of flight data (altitude and Mach number), and the heat load from the consumer. The atmosphere is modeled using the **International Standard Atmosphere (ISA)**. The atmospheric conditions impact the friction heating and heat transfer from the aircraft to the surroundings. The equations describing this are available in (Hällqvist, Munjulury, et al. 2021). The cooling system incorporated software then regulates the flow to keep the fluid temperature in the feed line at 20 °C.

3.2 Proposed Methodology

A methodology was developed based on the **CFD** and system identification work done during the thesis (Lindqvist 2022). **CFD**, System Identification, and **ROM** implementation make up its three primary stages. The methodology

is depicted in more detail in Figure 1, along with the steps that are part of each phase. There are, in addition, several crucial considerations that should be made both before and throughout the work, including the intended purpose of the finished **ROM**, how the system will function, and the resources that will be accessible.

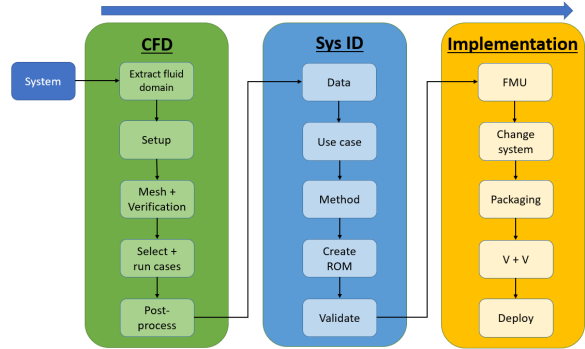


Figure 1. Detailed view of the methodology and its included steps.

The entire procedure is affected by the model's intended use. The appropriate **CFD** and system identification techniques, for instance, depend on whether the system's transient behavior needs to be accounted for or if steady-state characteristics are sufficient. For the **CFD** task, understanding the flow within the system that is being modeled is essential in order to, for example, choose the best turbulence model, meshing approach, and to determine whether the findings are plausible or not. In this context, the terms available resources and available software are used interchangeably. This will affect the number and sophistication of **CFD** simulations that can be performed, the system identification techniques that are accessible, etc. The following are the steps for each phase in Figure 1,

Stage 1: **CFD**

1. **Extract fluid domain:** Extraction of the fluid domain from the geometry of the system to be simulated is the first step in any **CFD** analysis. The domain could be divided into sections in order to focus on any interesting flow features while reducing the computational cost. Knowing the flow is crucial in this situation.
2. **Setup:** The setup of the **CFD** solver must be chosen in the next stage. This includes choosing a turbulence model, whether to execute steady-state or transient simulations, etc. Here, understanding the flow is equally crucial for choosing the modeling strategy that best depicts the anticipated flow characteristics.
3. **Mesh and Verification:** The mesh will need to meet different requirements depending on the turbulence model that is used, such as near-wall resolution. To make sure that it can handle any extreme scenarios, mesh verification should be done in the flow where the maximum turbulence is anticipated.

4. **Select + run cases:** If the system’s operational domain is known, an optimization method can be used to determine the ideal number of cases to cover the domain with the fewest number of simulations necessary. The operating domain can be expanded to accommodate system modifications, as was done for this study, to make the final **ROM** more adaptable.
5. **Post-process:** Depending on how much information the solution uses provides, post-processing the findings may be more or less challenging. When certain variables must be calculated using user-defined expressions, both the workload and the chance of error rise.

CFD can involve a lot of iterations. If, e.g., the findings of the post-processing indicate that a different turbulence model, is required, some processes might need to be repeated.

Stage 2: System Identification

The system identification steps largely follow the process outlined in Section 2.4.

1. **Data:** The data obtained from the **CFD** simulations need to be imported to the system identification tool used. For steady-state data, it may be necessary to add an “artificial” time-vector for each case, as many system identification techniques assume that such a vector is available.
2. **Method:** Here *method* denotes the decision on what candidate models and evaluation methods to use. The choice is likely dependent on what software that is chosen and available for the task at hand. In this scenario, knowledge about the system and flow can help with deciding, e.g., whether to use linear or non-linear models.
3. **Create ROM:** It is probable that the creation of the **ROM** will be highly iterative. It may be necessary to change the candidate models or evaluation in order to get a model that fits the **CFD** data.
4. **Validate:** The **ROM** should be validated against higher-order data that was not used to create it. In this study, the validation data consisted of **CFD** results for the same system at different flow cases.

Stage 3: Implementation:

1. **Functional Mock-up Unit (FMU):** In order to enable co-simulation, the **ROM** is exported as a **FMU**. This allows the **ROM** to be integrated into any **FMI** supporting system simulation environment.
2. **Change system:** In this study, the goal of the new **ROM** was to replace a component in a system model. Thus, the system model had to be modified to accept the new **ROM**. This step is not applicable when creating **ROM** for a completely new system model.

3. **Packaging:** The necessary model parameters, specifying the configuration or variant, for the **FMUs** are specified in an **SSV** file. If a system of several **FMUs** is to be simulated, it needs to be packaged as an **SSP**.
4. **Verification & Validation (V&V):** The new model or system needs to be verified, to make sure that it reflects its specification, and validated to make sure that it fulfills its intended use.
5. **Deploy:** The model can now be deployed and used for the intended system simulation.

4 Application example

The geometry modeling was performed in **CATIA** and the resulting geometry model serves as the foundation for the **CFD** analysis. The **CFD** simulations were performed using the **FEM**-based Altair AcuSolve **CFD**-solver. Meshing and post-processing were also performed using the built-in tools in AcuSolve. These tools were used for convenience, as the Altair tool suite also includes system identification and modeling tools.

4.1 Geometric Modeling

Figure 2 illustrates the feed and return lines of the aircraft cooling system. The starting point of the feed line is connected to a heat exchanger, while the endpoint is attached to a consumer of cooling power, such as a radar. To replicate the flow of the fluid within the pipeline, the **CAD** model was used to extract the fluid domain using **CATIA**. The **CAD** geometry of the fluid domain was then imported into AcuSolve to initiate the flow scenario configurations.

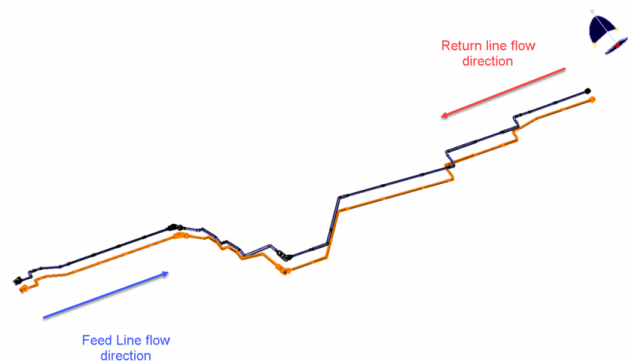


Figure 2. Pipes in the cooling system. The blue pipe is the feed line, while the return line is orange.

To decrease the computational cost of the simulations, the fluid domain of the return line was split into five distinct parts. These pipe sections were specifically selected due to their characteristics that could result in excessive pressure loss and turbulence, such as abrupt expansions and contractions, and acute bends. To ensure fully developed flow at the inlet and avoid reverse flow at the outlet, extensions were added to the inlets and outlets

of all pipe sections.

The initial section (here referred to as *Section 1*) focused on the inlet of the return line. It consisted of a 0.02m diameter pipe connected to a 0.008m diameter pipe through a fitting. The sudden contraction of the fitting was anticipated to cause separation and intensify the turbulence and pressure loss in this section. Since the section is symmetric along its axis, only a quarter of the fluid domain was simulated. The entire length of this section amounted to approximately 0.26m.

Section 2 and Section 3 (Section 2 is shown in Figures 3) exhibited resemblance. Each part comprised a pipe with a diameter of 0.008m and several closely located bends. Consequently, it was probable that turbulence would not disperse between the bends, which would result in an inaccurate estimation of pressure drop. Symmetry could be employed to simplify Section 3.

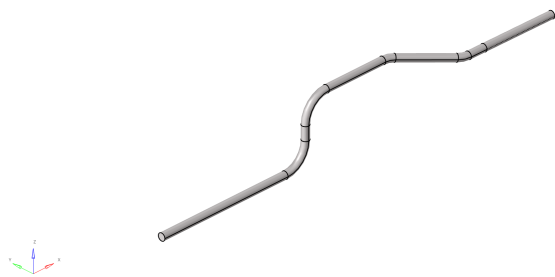


Figure 3. Section 2 of the return line, the flow direction is from right to left.

Section 4, depicted in Figure 4, constituted a complicated portion of the routing located approximately at the midpoint of the return line. A short pipe with numerous bends was linked to two additional linear sections using two fittings. Due to the intricate shape of this section, simplification through symmetry was infeasible. Furthermore, this was the most extensive section that was extracted and required the most computational resources to simulate. The complete length of this segment of the conduit was approximately 1.24m.

The last section dealt with the exit point of the return line (Figure 5). It showcased abrupt enlargements, reductions, and a 90-degree elbow. The inlet was elongated by 0.1m, whereas the outlet was lengthened by 0.2m in order to ensure fully developed flow at the inlet and outlet. This part could also be simplified using symmetry. The overall distance of the pipe in this segment was roughly 0.44m.

A test segment was additionally established to investigate the turbulence and pressure drop caused by the pipe fittings. This segment was comparable to Section 5, except for the appended outlet extension with a length of

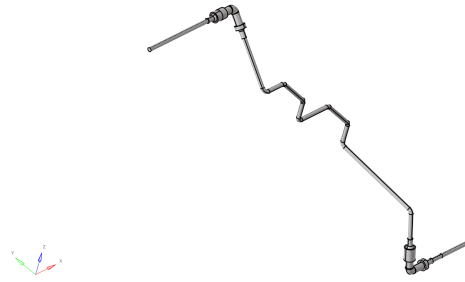


Figure 4. Section 4 of the return line, the flow direction is from right to left.

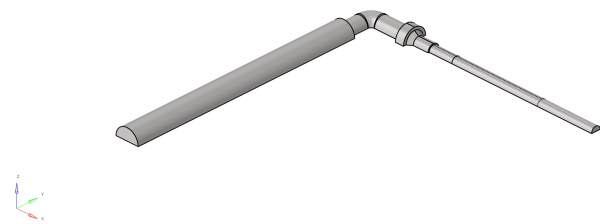


Figure 5. Section 5 (outlet) of the return line, the flow direction is from right to left.

50 pipe diameters, and the pipe diameter decreased back to 0.008m. The test segment is illustrated in Figure 6.

4.2 Meshing

Tetrahedral elements were primarily used to mesh the sections, ensuring the capture of their intricate geometries. However, on the inlet extensions where the flow was expected to be fully developed, hexcore elements were employed, effectively reducing the total element count. To obtain acceptable y^+ -values and capture the boundary layer, inflation layers were utilized. Furthermore, body of influence sizing was applied to refine the mesh in areas where significant gradients were anticipated, like sudden expansions. Figure 7 illustrates the general features common to all the meshes generated, including quad- and tri-surface meshes, the body of influence sizing around vital flow features, and inflation layers.

To verify the meshes, three different mesh sizes were evaluated for each section, and the solution results were compared. All mesh verification simulations were conducted using the fluid Dowcal 10 (Dow 2023) at 20°C with an inlet mass flow of 0.4kg/s. This resulted in a density of 1082 [kg/m³] and a dynamic viscosity of 0.005kg/ms, as shown in Figure 8. Table 1 displays the element numbers employed for each section. Additionally, to ensure a smooth **Eddy Viscosity Ratio (EVR)** gradient from the wall to the bulk flow, the near-wall mesh's resolution was also evaluated.

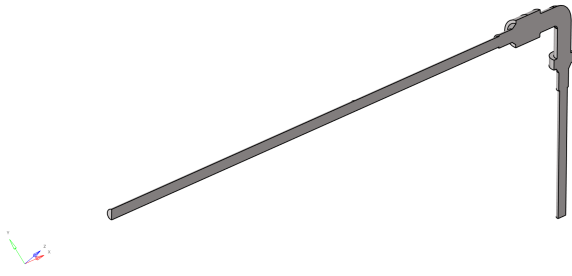


Figure 6. Test section with outlet extension with length of 50 pipe diameters, to investigate flow downstream of pipe fitting.

Section	No. of elements
1	204000
2	773000
3	489000
4	4180000
5	862000

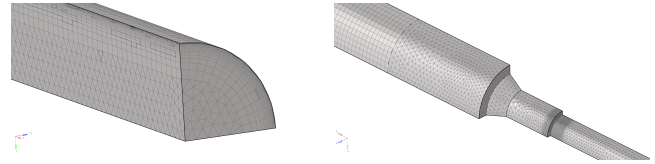
Table 1. Number of elements in the selected mesh for each section.

4.2.1 Solver Setup

Altair AcuSolve utilizes an implicit FEM-based solver with steady-state time-stepping. The turbulence model $k - \omega$ SST was chosen for the simulations, because of its good performance near walls, in adverse pressure gradients, and in separated flow. The boundary conditions for all sections were similar. At the inlet, the mass flow was specified, while the outlet was set to zero gauge pressure. The turbulence parameters at the inlet were automatically calculated. The pipe walls were given a wall roughness height of $2e - 5m$, the same as in the Dymola model. Symmetry was applied to the symmetry planes if available. Heat transfer through the walls of the pipe was neglected. The material was set to the water-glycol mixture Dowcal 10, the same fluid used in the Modelica model. The density and dynamic viscosity variations for Dowcal 10 as functions of temperature are shown in Figure 8. In order to ensure convergence of the solutions, residuals, and monitor points throughout the domains were checked. AcuSolve also uses another convergence metric called the *Solution Ratio*, which measures the difference in results between iterations (*AcuSolve Residual Computation* 2013).

4.2.2 Post-processing

The pressure drop Δp for the different sections was investigated by taking the difference in total pressure between the inlet and outlet. For routing Section 5, the pressure drop was defined as the difference between the point of lowest total pressure and the outlet. This is because the sudden expansion in the fitting causes the pressure to decrease rapidly to a minimum in this routing location. Thus, the pressure drop is the pressure required to get it back to zero at the outlet. Another example of calculating the pres-



(a) Mesh on section 1 showing inflation layers and bulk mesh. (b) Surface mesh on section 1 showing a transition from quad to tri elements, and body of influence sizing.

Figure 7. Examples of the generated meshes.

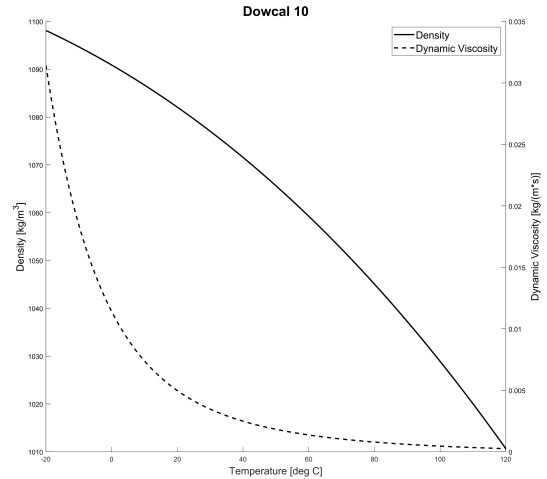


Figure 8. Density and dynamic viscosity as functions of temperature for Dowcal 10.

sure drop for a sudden expansion can be found in (Roul and Dash 2009).

4.3 System Simulation models - romAI & Lookup Table (LUT)

In order to integrate the romAI models in a system simulation context, a new pipe model had to be created. The purpose of this pipe model would be to combine the romAI models for the pipe sections of the pipe where CFD simulations had been run, with handbook equations for the rest of the pipe. This pipe model would then be exported as a FMU for integration using the FMI standard. The requirements for the FMUs were to use the FMI 2.0 standard for co-simulation and to be license free, enabling wide-spread use throughout the organization. The new model was created in Altair Activate. Figure 9 shows an overview of the romAI-based model.

The inputs provided by the surrounding modeled system are mass flow, enthalpy, and pressure entering the pipe model. The mass flow and enthalpy outputs from the pipe were also set to equal the input values, implying continuity and ideal thermal insulation. In order to calculate the pressure drop in the pipe, the input enthalpy first need to be converted into fluid density and viscosity, which are the inputs required by both the romAI model and the

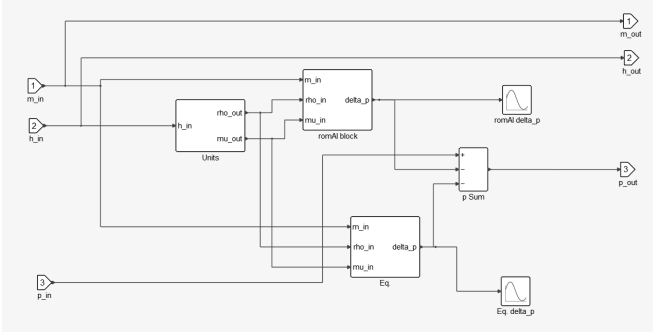


Figure 9. Overview of the romAI-based pipe model.

handbook equations. This conversion is conducted in block labelled *Units* in Figure 9. The equation block *Eq* receives the mass flow, density, and viscosity to calculate the pressure drop via Equation 3. The pressure drops calculated are then subtracted from the input pressure, to give the output pressure.

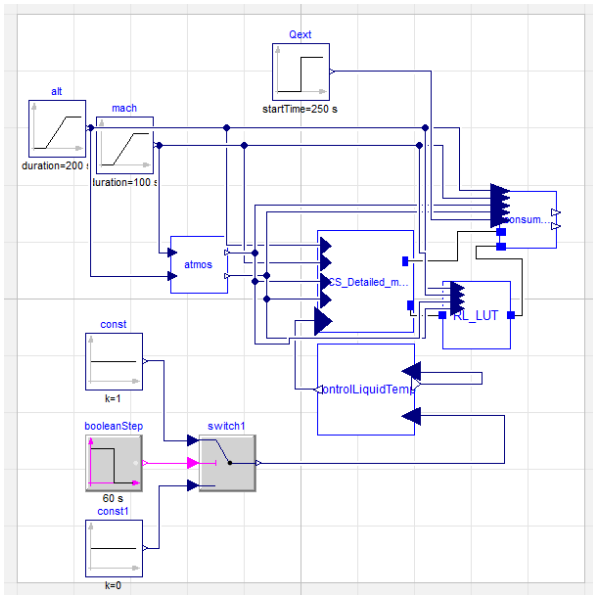


Figure 10. Overview of the new cooling system. *RL_LUT* is the new *LUT*-based pipe model (the romAI model was integrated in the same way).

The new *FMU* is integrated into the simulator by removing the *pipeB* component from the legacy model, and then inserting the new *FMU* as shown in Figure 10. The complete system is then re-packaged as a *SSP* file together with the necessary *SSV* data etc.

The simulation of the updated simulator was performed using the *OMSimulator* tool (Ochel et al. 2019), executed through dedicated Python scripts. The altitude, Mach number, and consumer heat load boundary conditions were varied over time to simulate hypothetical flight missions. The results from the updated simulator were then compared to the legacy version.

5 Results

This section presents the outcomes of applying the technique outlined in the preceding section. The produced *ROMs* and the output data from the *CFD* simulations are both included. The method for *CFD*-based system identification is offered as a last step.

5.1 Design of experiments

The flow cases were selected in order to give the maximum coverage of the pipe systems *Operational Domain (OD)*. The coverage was defined using the modified nearest-neighbor coverage metric described in (Atamturktur et al. 2015). A reduced metric value implies an increase in *OD* coverage. Figure 11 shows the operating

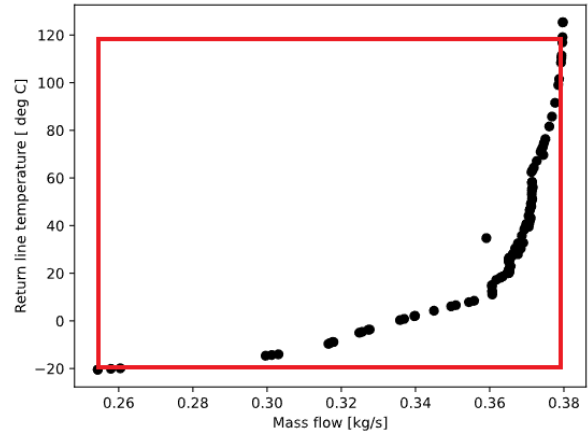


Figure 11. Operational domain of the pipe system. Black points indicate the operating points for a specific pump. The red rectangle is the created operational domain.

points of the pipe system with regard to mass flow and fluid temperature. These operating points were calculated while using a specific pump connected to the cooling system. The outlier at $\dot{m} = 0.36\text{kg/s}$ and $T = 40^\circ\text{C}$ is a result of the simulation not reaching steady-state. In order to make the model more generalized (e.g., be applicable for evaluating different pump alternatives), a rectangular operational domain was created by connecting the maximum and minimum values in Figure 11. This gave the *OD* the limits of $0.25 < \dot{m} < 0.38\text{kg/s}$, and $-20 < T < 120^\circ\text{C}$. Between one and 15 points (flow cases) were placed in the *OD* using Complex-RF optimization (Krus and J. Andersson 2003), striving to minimize the modified nearest-neighbor coverage metric. Figure 12 shows the change in coverage for an increasing number of points. A lower nearest neighbor coverage metric η_c indicates better coverage of the domain. Based on the result, ten cases were selected, see Figure 13. An additional ten randomly selected cases were run to use as validation for the created *FMU*. All 20 cases are shown in Table 2.

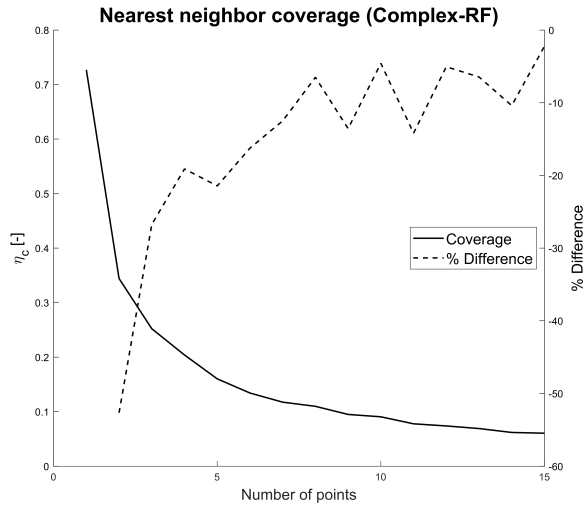


Figure 12. Nearest neighbor coverage for the operational domain. A lower η_c indicates better coverage.

	Case	\dot{m}_{norm} [-]	T_{norm} [-]	\dot{m} [kg/s]	T [degC]	ρ [kg/m ³]	μ [kg/ms]
Input	1	0.999	1.000	0.380	119.99	1010.66	0.000257
	2	0.663	1.000	0.252	119.94	1010.70	0.000257
	3	0.954	0.796	0.362	95.53	1032.69	0.000519
	4	0.658	0.316	0.250	37.93	1072.74	0.00266
	5	0.774	0.028	0.294	3.38	1089.52	0.00980
	6	0.786	0.611	0.299	73.30	1050.05	0.000948
	7	0.754	0.805	0.287	96.63	1031.76	0.000504
	8	0.674	-0.167	0.256	-19.98	1098.11	0.03141
	9	0.999	-0.166	0.380	-19.93	1098.09	0.03133
	10	0.917	0.319	0.348	38.26	1072.56	0.002633
Validation	11	-	-	0.309	116	1014.47	0.00029
	12	-	-	0.334	96	1032.30	0.000513
	13	-	-	0.268	80	1045.08	0.000791
	14	-	-	0.274	28	1078.09	0.003719
	15	-	-	0.295	42	1070.42	0.002337
	16	-	-	0.356	1	1090.50	0.010907
	17	-	-	0.378	-13	1095.77	0.021543
	18	-	-	0.256	31	1076.52	0.003351
	19	-	-	0.328	115	1015.41	0.000299
	20	-	-	0.340	56	1061.88	0.001534

Table 2. Selected flow cases. Cases 1-10 were selected to optimize coverage to be used as input data for the ROM creation. Cases 11-20 were selected randomly for validation purposes.

5.2 CFD Outcome

Table 3 shows the outcomes for all the steady-state CFD cases run for each pipe segment. Cases 1-10 are the cases that were utilized to make the ROMs, while cases 11-20 were utilized for validation. From the outcomes, it may be presumed that the cases with the lowest temperatures and highest mass flows give the highest pressure drop. This demonstrates that it is the friction along the line walls that contributes most to the losses. For more detailed of CFD results, see (Lindqvist 2022).

5.3 Created ROMs

Figure 14 shows the calculated pressure drops for the model of routing Section 1 created using romAI, compared to the validation CFD data. The blue line shows the results for the ROM, while the green line is the results from the validation data. Figure 15 shows the Δp output from the romAI and 8x8 lookup table run for a test case

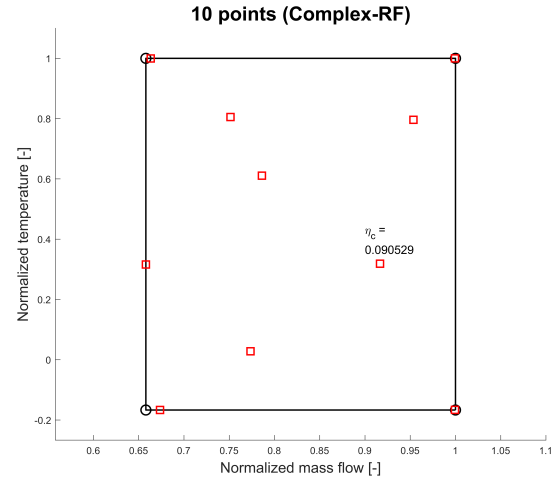


Figure 13. The 10 cases selected for the steady state model creation. The mass flow and temperature have been normalized with their maximum values (0.38 [kg/s] and 120 [°C] respectively).

Case	Pressure drop [Pa]				
	Section 1	Section 2	Section 3	Section 4	Section 5
1	13575	42584	32194	130447	15789
2	6589	19670	15101	59767	6970
3	14008	43059	32784	127623	14006
4	8615	28097	21591	78081	6441
5	15240	53233	40537	142794	8808
6	10555	32886	25349	94778	9353
7	9067	28032	21388	82144	8771
8	20130	73221	55077	181600	6854
9	34975	129844	98110	330255	15023
10	15748	50652	38909	143317	15023
11	9703	29390	22914	89460	10400
12	12025	37156	28283	110235	11907
13	8450	26382	20227	76801	7598
14	10816	35672	27361	98710	7709
15	11413	36773	28254	103944	8935
16	21884	75769	57738	204107	12985
17	29551	108363	82113	281642	14814
18	9362	30791	23622	85180	6683
19	10880	33029	25044	100190	11755
20	14126	44520	34192	128187	12058

Table 3. Steady-state CFD results for the pressure drop through each section.

with the mass flow and enthalpy inputs as sine curves. The lookup table result differs at most between 2 – 3kPa from the romAI model. Additional system identification results can be found in (Lindqvist 2022).

5.4 System Simulations

The pressure loss for the different return line pipe models over a range of mass flow and temperature values is shown in Figure 16. The models all have a similar pressure drop when the temperature is around 20 °C. As the temperature decreases, the pressure drops for the romAI and LUT models increase. The Modelica pressure drop decreases until $Re < 2000$, when it too starts to increase. For the low-temperature flows, the romAI and LUT models consistently give a higher pressure drop than the cur-

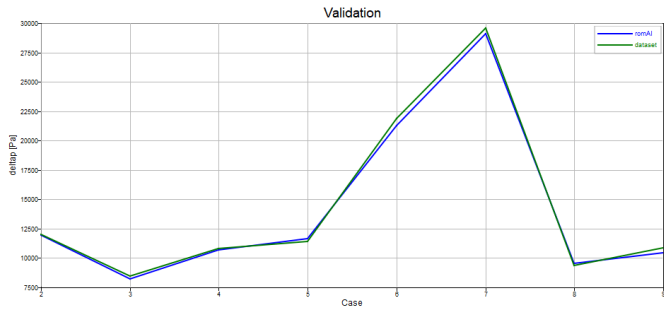


Figure 14. Comparison against validation data for Section 1.

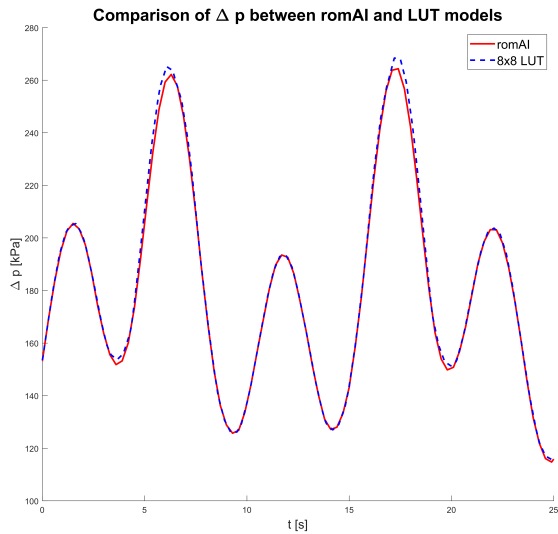


Figure 15. Comparison between 8x8 LUT and romAI model.

rent model. The difference in the romAI and LUT responses to the changing flow conditions is because the LUT model has been implemented using dynamical model components, while the romAI model contains no dynamics. Thus, the romAI model's response is instant. It should be noted that the temperature here goes lower than the lower limit of the operational domain of the created ROMs. Figure 16 mainly shows the similarity between the romAI and LUT results and the behavior of the Modelica pipe model at low Re.

6 Discussion and Conclusions

The work presented herein aims to fill a gap in the industry, enabling the adaption of model detail to the required level in model and simulator development. The presented use-case entails the incorporation of information of high-fidelity analysis using CFD in the systems' simulation domain efficiently. However, a number of different application areas exist. The work demonstrates an efficient means of producing Reduced Order Models from high-fidelity data, and such surrogates have a wide range of applications. One concrete example, to be exploited in the European Defense Fund Project-NEUMANN (Euro-

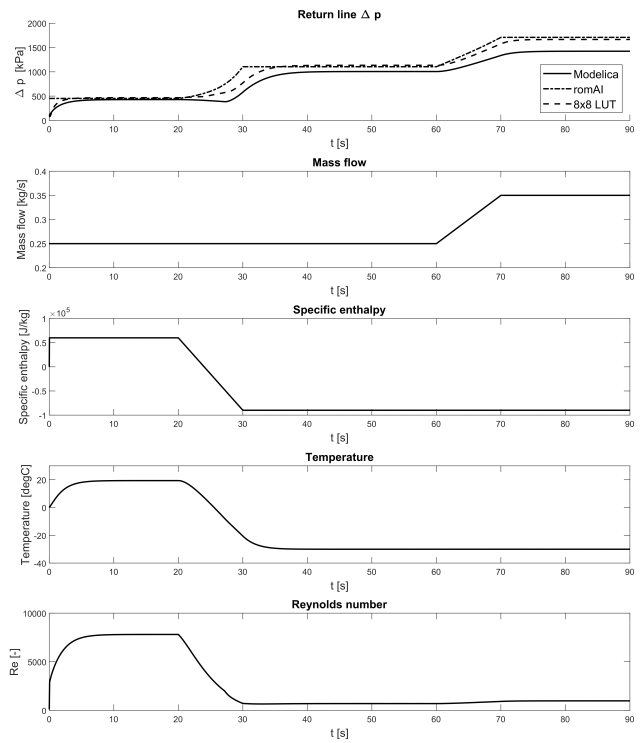


Figure 16. Comparison between current Modelica, romAI, and LUT models.

pean Defence Fund 2022), relates to adapting models for applicability in the model-based design of energy management control strategies in the aerospace domain.

References

- AcuSolve Residual Computation (2013). Altair University.
- Altair (2023-06-13). *Altair AcuSolve: General Purpose Fluids and Thermal Simulation*. URL: <https://www.altair.de/Product/AcuSolve/> (visited on 2023-06-13).
- Anderson, John D. (1995). *Computational Fluid Dynamics: The Basics with Applications*. 1st ed. McGraw-Hill.
- Andersson, Henric (2012). "Variability and Customization of Simulator Products : A Product Line Approach in Model Based Systems Engineering". PhD thesis. Linköping University, Machine Design, p. 83. ISBN: 978-91-7519-963-4.
- Atamturktur, Sez et al. (2015). "Defining coverage of an operational domain using a modified nearest-neighbor metric". In: *Mechanical Systems and Signal Processing* 50, pp. 349–361.
- Bathe, Klaus J. (2014). *Finite Element Procedures*. 2nd ed. K.J. Bathe, Watertown, MA.
- Chen, Sheng, Stephen A Billings, and PM Grant (1990). "Non-linear system identification using neural networks". In: *International journal of control* 51.6, pp. 1191–1214.
- Donea, Jean and Huerta, Antonio (2003). *Finite Element Methods for Flow Problems*. 1st ed. Wiley.
- Dow (2023-08-14). *Dowcal 100 Heat Transfer Fluids*. URL: <https://www.dow.com/en-us/pdp.dowcal-100-heat-transfer-fluid.244104z.html#overview> (visited on 2023-06-13).
- Dymola User Manual (2016-09). Dassault Systèmes AB.
- Eek, Magnus, Hampus Gavel, and Johan Ölvander (2017). "Definition and implementation of a method for uncertainty aggregation in component-based system simulation models". In:

- Journal of Verification, Validation and Uncertainty Quantification* 2.1.
- Ellner, Stephen P and John Guckenheimer (2011). *Dynamic models in biology*. Princeton University Press.
- European Defence Fund (2022). *NEUMANN-Novel Energy and propulsion systems for Air dominance, SELECTED PROJECTS EUROPEAN DEFENCE FUND (EDF) 2021*. URL: https://defence-industry-space.ec.europa.eu/system/files/2023-01/Factsheet_EDF21_NEUMANN_0.pdf (visited on 2023-08-03).
- Fernandez, Benito, Alexander G Parlos, and Wei Kang Tsai (1990). "Nonlinear dynamic system identification using artificial neural networks (ANNs)". In: *1990 IJCNN international joint conference on neural networks*. IEEE, pp. 133–141.
- FMI Development Group (2020-12-15). *Functional Mock-up Interface for Model Exchange and Co-Simulation*. Report 2.0.2.
- Fontes, Ed (2018-11). *FEM vs. FVM*.
- Fritzson, Peter (2003). *Principles of Object Oriented Modeling and Simulation with Modelica*. 1st ed. Wiley.
- Fritzson, Peter et al. (2005-12). "The OpenModelica Modeling, Simulation, and Software Development Environment". In: *Simulation News Europe* 44, pp. 8–16. ISSN: 0929-2268.
- Hällqvist, Robert (2023). *On the Realization of Credible Simulations in Aircraft Development: Efficient and Independent Validation Enabled by Automation*. Linköping Studies in Science and Technology. Dissertations: 2279. Department of Management and Engineering, Linköping University Electronic Press. ISBN: 9789179295981.
- Hällqvist, Robert, Raghu Chaitanya Munjulury, et al. (2021). "Engineering Domain Interoperability Using the System Structure and Parameterization (SSP) Standard". In: *Proceeding of the 14th International Modelica Conference*.
- Hällqvist, Robert, Raghu Chaitanya Munjulury, et al. (2022). "Realizing Interoperability between MBSE Domains in Aircraft System Development". In: *Electronics* 11.18. ISSN: 2079-9292. URL: <https://www.mdpi.com/2079-9292/11/18/2901>.
- Hällqvist, Robert, Johan Naeser, et al. (2022-09-04). "Heterogeneous System Modelling in Support of Incremental Development". In: *Proceedings of the 33rd Congress of the International Council of the Aeronautical Sciences*. Stockholm, Sweden.
- Haykin, Simon (1999). *Neural Networks: A Comprehensive Foundation*. 2nd ed. Prentice Hall.
- International Standards Organization (ISO) (2016-03). *Industrial automation systems and integration, Product data representation and exchange*. Part 21: Implementation methods: Clear text encoding of the exchange structure.
- Krus, Petter and Johan Andersson (2003). "Optimizing optimization for design optimization". In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 37009, pp. 951–960.
- Lanza, R. et al. (2018). "Relating structural test and FEA data with STEP AP209". In: *Advances in Engineering Software* 127.2019, pp. 96–105. DOI: [10.1016/j.advengsoft.2018.08.005](https://doi.org/10.1016/j.advengsoft.2018.08.005).
- Lind, Ingela and Henric Andersson (2011). "Model Based Systems Engineering for Aircraft Systems – How does Modelica Based Tools Fit?" In: *Proceedings of the 8th Modelica Conference*. Dresden, Germany.
- Lindqvist, Olle (2022). "A Method of CFD-based System Identification for Standardized Co-Simulation." In.
- Ljung, Lennart (1999a). *System Identification: Theory for the User*. 2nd ed. Prentice Hall. ISBN: 0-13-656695-2.
- Ljung, Lennart (1999b). *System identification: theory for the user*. 2nd ed. Prentice Hall. ISBN: 0136566952.
- Ljung, Lennart and Torkel Glad (2003). *Modellbygge och Simulering*. 2nd ed. Studentlitteratur. ISBN: 9789144024431.
- Miller, Donald (1990). *Internal Flow Systems*. Cranfield, Bedford: BHRA (Information Services. ISBN: 978-0956200204.
- Miller, Donald S. (1990). *Internal Flow Systems*. 2nd ed. Miller Innovations. ISBN: 978-0-9562002-0-4.
- Modelica Association (2019-03-05). *System Structure and Parameterization*. Report 1.0.
- Modelica Association (2023-03-09). *Modelica-A Unified Object-Oriented Language for Systems Modeling. Language Specification Version 3.6*. URL: <https://specification.modelica.org/maint/3.6/MLS.pdf>.
- Ochel, Lennart et al. (2019-03-04). "OMSimulator – Integrated FMI and TLM-based Co-simulation with Composite Model Editing and SSP". In: *Proceeding of the 13th International Modelica Conference*. DOI: [10.3384/ecp1915769](https://doi.org/10.3384/ecp1915769).
- Parlos, A et al. (1991). "Recurrent multilayer perceptron for nonlinear system identification". In: *IJCNN-91-Seattle International Joint Conference on Neural Networks*. Vol. 2. IEEE, pp. 537–540.
- Roul, Manmatha K and Sukanta K Dash (2009). "Pressure drop caused by two-phase flow of oil/water emulsions through sudden expansions and contractions: a computational approach". In: *International Journal of Numerical Methods for Heat & Fluid Flow*.
- Saab Group (2023). *Gripen E program Update 2021*. URL: <https://www.saab.com/newsroom/stories/2021/june/gripen-e-program-update-2021> (visited on 2023-08-14).
- Sjöberg, J. et al. (1995). "Nonlinear Black-Box Modeling in System Identification: a Unified Overview". In: *Automatica* 31, pp. 1691–1724.
- Steinkellner, Sören (2011). "Aircraft Vehicle Systems Modeling and Simulation under Uncertainty". Licentiate thesis. Linköping University, Division of Machine Design. ISBN: 9789173931366.
- Versteeg, H K and W Malalasekera (2007). *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. 2nd ed. Pearson. ISBN: 9780131274983.