International Journal of Refrigeration 178 (2025) 336–346

Contents lists available at ScienceDirect

International Journal of Refrigeration

journal homepage: www.elsevier.com/locate/ijrefrig

Vapor compression system data-driven surrogate models for aircraft Environmental Control Systems

Nicolás Ablanque [a],*, Nasrulloh Loka [b], Santiago Torras [a], Sriram Gurumurthy [c], Joaquim Rigola [a], Carles Oliet [a], Ivo Couckuyt [b], Tom Dhaene [b], Antonello Monti [c]

[a] Universitat Politècnica de Catalunya - Barcelona Tech (UPC), Heat and Mass Transfer Technological Center (CTTC), Spain
[b] IDLab, Ghent University - IMEC, Belgium
[c] ACS, EONERC, RWTH Aachen University, Germany

ARTICLE INFO

Keywords:
Complex system simulation
Environmental control system
Vapor compression system
Surrogate model
Data-driven model
Gaussian Process
Multi layer perceptron

ABSTRACT

Commercial aircraft manufacturers are actively working to develop more efficient Environmental Control Systems (ECS) in line with the More Electric Aircraft (MEA) concept. Novel ECS architectures may integrate supplementary cooling units based on Vapor Compression Systems (VCS), which provide more efficient cooling compared to traditional Air Cycle Machines (ACM). The design, optimization, and evaluation of these new ECS configurations rely on numerical simulations. Given the inherent complexity and computational demands of VCS, this study explores the use of data-driven surrogate models for efficient simulation of these systems. The main objective was to develop surrogate models for the VCS cooling unit, designed as steady-state equivalents of physics-based models, which have provided the training and testing data. To achieve this, we developed data-driven surrogate models using Gaussian Processes (GP) and Multi-Layer Perceptrons (MLP), both trained on outputs from high-fidelity simulations. The results show that GP models perform best when the output data exhibits smooth, continuous behavior, while MLPs are better suited to capturing complex, nonlinear behavior. This complementary performance highlights the strengths of each approach depending on the nature of the output data. Once model accuracy is verified, the VCS data-driven surrogate models were seamlessly adapted to be used under the Modelica/Dymola environment to comprehensively assess their accuracy and computational performance. The results demonstrated a significantly lower CPU time consumption when comparing physics-based models to data-driven twins across various scenarios, including VCS steady-state conditions, ECS steady-state conditions, and ECS dynamic conditions.

1. Introduction

Apart from propulsion, the largest energy consumer in commercial aircraft is the Environmental Control System (ECS) (Yuanchao and Zichen, 2019). The primary function of the ECS is to ensure the comfort of both crew and passengers by maintaining the appropriate temperature, pressure, and humidity in the cabin. Additionally, it handles other cooling requirements for food and electronic devices (Pérez-Grande and Leo, 2002). Traditionally, medium to large civil aircraft have relied on an Air Cycle Machine (ACM) for their major cooling needs, using bleed pressurized air from the engines (Smith et al., 2018). These systems have proven reliable but have a lower coefficient of performance compared to vapor compression systems, which is particularly disadvantageous under high cooling demands, such as on hot days at low altitudes (Santos et al., 2014). Recently, new ECS configurations aligned with the More Electric Aircraft (MEA) concept have been explored and implemented to achieve better efficiency (Sarlioglu and Morris, 2015). In this context, incorporating supplementary electrically driven Vapor Compression Systems (VCSs) into the ECS stands out as a favorable option to increase the global efficiency due to their better performance compared to ACMs (Merzvinskas et al., 2019).

The contributions of numerical simulations are crucial for the design and optimization of these novel ECS architectures. Analyzing these architectures is challenging due to the complexity introduced by the numerous systems involved, their interactions, and their varied physical characteristics, including thermal, electrical, pneumatic, and mechanical aspects. Extensive research has been devoted to simulating and investigating ECS focused on ACM and single-phase working fluids (Pérez-Grande and Leo, 2002; Jiang et al., 2020; Pollock, 2018), while more recent studies have shifted attention towards ECS incorporating VCS (Golle and Hesse, 2019).

* Corresponding author.
E-mail address: nicolas.ablanque@upc.edu (N. Ablanque).

https://doi.org/10.1016/j.ijrefrig.2025.06.035
Received 3 March 2025; Received in revised form 30 May 2025; Accepted 30 June 2025
Available online 15 July 2025

The simulation of aircraft supplementary cooling systems integrated with vapor compression systems (VCSs) has recently been explored by several authors. Hu et al. (2020) developed a steady-state simulation model incorporating a graph theory-based description (to represent various configurations of refrigerant cycles and cooling fluid loops), a decoding method (to identify all components and connections), a mathematical formulation (to integrate global governing equations and component models), and a tailor-made iterative algorithm (to solve the mathematical model). The model was experimentally validated, with deviations in cooling capacity limited to 8%. A cooling system configuration with two VCSs connected in series was compared to an equivalent parallel connection, showing slightly higher input power and energy efficiency in the former.

Dynamic simulations of VCSs as supplementary cooling systems in aircraft have also been explored. Erkinaci et al. (2024) conducted an experimental and numerical study of VCSs for avionics cooling. They developed a dynamic, system-level model of the VCS using the Simcenter Amesim simulation environment, based on geometric parameterization and performance calibration of system components. The model was validated against experimental data under varying heat loads and ambient temperatures, demonstrating different levels of predictive accuracy—with higher confidence in pressure predictions compared to those of superheating and subcooling temperatures. Using the Simulink platform, Li et al. (2024) developed a dynamic model of an aircraft auxiliary unit to address non-cabin cooling requirements, which included a VCS model. Dynamic component models for the compressor and heat exchangers were created using dynamic decoupling and explicit solution methods. The authors reported simulation time ratios of 5.6% for the VCS alone and 21.9% for the VCS integrated into the full cooling system. Ablanque et al. (2024) developed a physics-based numerical model, supplemented with empirical data, to dynamically simulate VCSs within a Modelica simulation environment. The model was designed to meet the numerical requirements necessary for integration into larger ECS models and to perform both real-time calculations and extensive parametric simulations. These requirements included strong numerical robustness (to effectively handle the complexities of large multi-physics architectures), high prediction accuracy, and minimal computational time. Simulation time ratios below 1% were reported for a simplified ECS model incorporating a VCS across complete flight missions.

The study presented herein aims to develop a twin VCS model using machine learning techniques, integrate it into the Modelica simulation environment, and evaluate its accuracy and computational performance when embedded within an ECS model.

The literature shows that various neural network and deep learning techniques have been employed to investigate and model different facets of VCSs. For instance, Wang et al. (2024) reviewed machine learning models used to predict the performance of Variable Refrigerant Flow (VRF) systems. They highlighted the strengths of these models, including their capability to provide accurate predictions without requiring an in-depth understanding of the underlying physics. However, they also noted limitations, such as the reliance on adequate and extensive data availability and the inability to address unexpected conditions. Yousaf et al. (2023) investigated the air and refrigerant critical features influencing the cooling capacity of an air-conditioning equipment. The selection of inputs was based on an Elastic Net methodology while an Artificial Neural Network (ANN) architecture was used to compare the cooling capacity prediction between different sets of features. Shao et al. (2012) proposed an hybrid steady-state modeling approach for air conditioning systems using ANN models at the component level but with first principles applied at the system level. They realized that the hybrid model could be up to 100 times faster than the fully physics-based model. However, they also underlined the large amount of data required for the models training and the difficulty to completely avoid the over-fitting risk. Ma et al. (2024) employed deep learning techniques to create a VCS system solver, using feedforward neural networks to simulate static components such as the compressor, and recurrent neural networks for the dynamic simulation of heat exchangers.

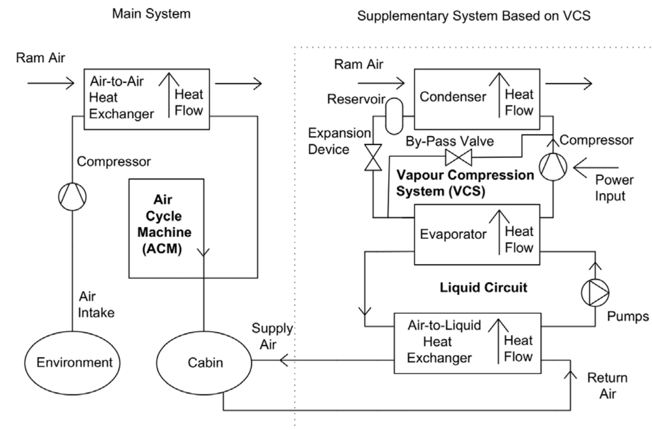

Fig. 1. Simplified scheme of ECS with supplementary cooling system based on VCS.

In this work, the entire VCS is modeled using machine learning techniques and fully incorporated into a comprehensive ECS architecture developed in the Modelica language. The main objective is to explore the data-driven approach capabilities compared to the optimized physics-based model presented in Ablanque et al. (2024). Specifically, a data-driven surrogate model was created for each target variable within the VCS. For target variables displaying smooth behavior, Gaussian Process (GP) regression models were constructed due to their data efficiency and strong interpolation capabilities for smooth functions. For more complex target variables with highly non-linear behavior, a Multi-Layer Perceptron (MLP) Neural Network model was utilized. Additionally, these two models are straightforward to translate into Modelica and demonstrate strong performance on comparable tasks (Loka et al., 2022, 2023)

Initially, the studied environmental control system is briefly presented in section two of this document. Then, the VCS physics-based model used to generate data is introduced in the third section, while the process to develop data-driven surrogate models for each VCS parameter considered is detailed in section four. Finally, section five is devoted to assess the computational performance of both physics-based model and surrogate twin, focusing on different scenarios including a dynamic simulation for a typical commercial airplane flight under hot day conditions.

2. Aircraft environmental control system

The schematic layout examined in this study for an aircraft ECS, which is based on an ACM unit with an additional VCS cooling unit, is shown in Fig. 1. On the one hand, the main system consists of an ACM unit, which is aimed to provide appropriate pressure, temperature and humidity conditions for the cabin. In this MEA-based configuration, the air drawn from the environment to supply the ACM is pressurized by dedicated electrically driven compressors instead of engine compressors. On the other hand, the supplementary cooling system consists of a VCS, whose purpose is to provide additional cooling to the cabin when demand is high (typically at low altitude or ground level during hot day conditions). This unit can be activated or deactivated at any point during the flight.

The VCS located at the core of the supplementary cooling unit is the focus of this work. It employs R-1234ze as the working fluid and follows the standard single-stage cycle configuration. It incorporates four main components: a centrifugal compressor, a moist air vs. refrigerant condenser, an expansion valve serving as the mass flow regulator,

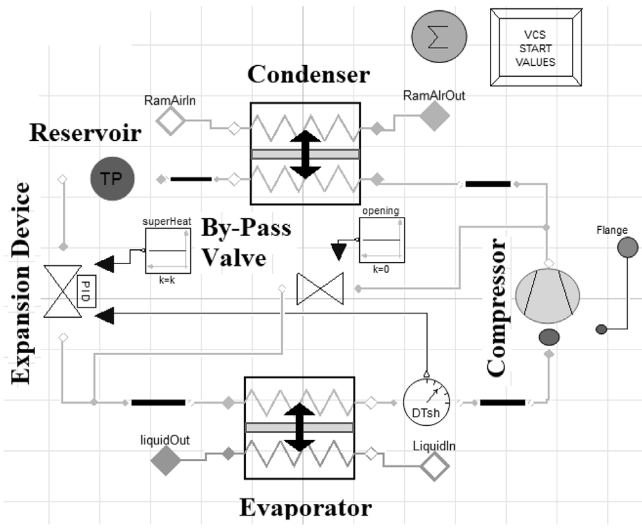

Fig. 2. VCS model graphical scheme.

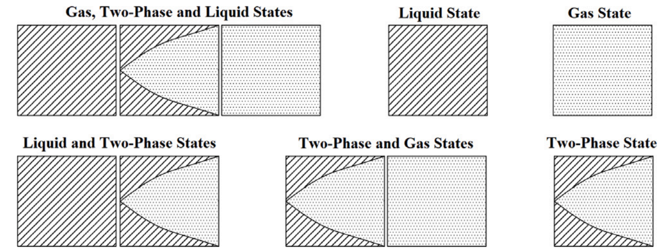

Fig. 3. Discretization depending on fluid states for refrigerant flow during evaporation.

and a liquid vs. refrigerant evaporator. Additionally, it features minor components such as a reservoir positioned between the condenser and the expansion valve, and a bypass valve primarily used to control the overall cooling capacity generated by the VCS (see Fig. 1).

3. VCS: Physics-based model

The physics-based model for the VCS and its components was developed using the Modelica/Dymola platform and its simplified layout is shown in Fig. 2. The model has been numerically tested and validated in previous studies (more detailed information can be found in Ablanque et al., 2024). It is important to emphasize that the model is partly based on physical principles, such as conservation equations, while also incorporating empirical data in the form of efficiencies or performance maps. The following sections provide summarized explanations of both components and system modeling.

3.1. Components

The circuit is primarily composed of two key components: the compressor and the heat exchangers. Among these, the condenser and evaporator pose the greatest challenges for modeling due to the complex phenomena they involve.

The Vapor Compression System under study is powered by a centrifugal compressor. This compressor is modeled using two distinct subcomponents: one responsible for calculating the compression process, which is assumed to occur instantaneously, and another for determining the refrigerant mass balance within the compressor's internal volume to capture the flow dynamics. The compression model utilizes performance maps derived either from experimental data or from detailed physics-based numerical models. These performance maps allow to calculate the compressor's pressure ratio and its power consumption from two parameters: the corrected compressor rotational speed and the corrected mass flow rate (Shen et al., 2019):

$$N_{corr} = N \frac{1}{\sqrt{T_{in}/T_{ref}}} \tag{1}$$

$$\dot{m}_{corr} = \dot{m} \frac{\sqrt{T_{in}/T_{ref}}}{p_{in}/p_{ref}} \tag{2}$$

Here, N denotes the rotational speed, $\dot{m}$ the mass flow rate, T_{in} the suction temperature, T_{ref} the suction temperature under nominal conditions, p_{in} the suction pressure, and p_{ref} the suction pressure under nominal conditions. Additionally, the compressor model includes the calculation of the critical mass flow rate at which surging occurs. The calculation is based on a fitted curve that expresses this threshold as a function of the compressor's normalized speed. This curve enables the implementation of control actions to prevent surge throughout the simulation. When the mass flow rate falls below the critical limit, a bypass line is opened to redirect part of the compressor discharge flow back to the suction side, shifting the operating point away from the surge threshold.

The heat exchangers are modeled with three sub-components: one for the secondary fluid, another for the refrigerant, and a third for the solid material that transfers heat between the fluids. The solid material is solved dynamically to capture the thermal inertia of the entire heat exchanger. The secondary fluid flow is calculated using a steady-state approach based on the $\varepsilon - NTU$ method, which enhances computational efficiency and ensures that temperatures remain within realistic limits. The refrigerant flow is solved similarly to the secondary fluid but with additional consideration of three distinct zones, which account for the gas, liquid and two-phase refrigerant's thermodynamic states. These zones are activated or deactivated according to the flow and heat conditions giving six possible combinations as shown in Fig. 3 for the case of an evaporator.

In addition to these primary components, the system also includes several minor elements. For example, the connecting pipes account for refrigerant pressure drops, and the reservoir employs a two-phase equilibrium formulation, where the amount of liquid is proportional to its height.

3.2. System

The VCS model must account for both the control of superheating and the management of refrigerant charge within the system.

Superheating in the VCS is regulated by a thermostatic valve placed between the reservoir and the evaporator (refer to Fig. 2). These self-regulating systems can introduce numerical instabilities. To mitigate this, the current VCS model employs an adjustable valve controlled by a PI (Proportional–Integral) controller. This controller compares the desired superheating degree with the measured value, adjusting accordingly. The system's response is influenced by the chosen PI parameters, which can be fine-tuned using reference or experimental transient data.

The VCS refrigerant charge is a constant value defined at the start of the simulation that must remain unchanged. To track the refrigerant amount during the simulation, the total charge is calculated at each time step by summing the refrigerant mass across all components. Due to the lumped approach used in the two-phase flow heat exchanger models, small charge deviations can occur during dynamic simulations (Laughman and Qiao, 2016), as the mass is estimated using empirical parameters such as the mean void fraction. To address this, the model allows minor corrections at each time step: if the calculated total refrigerant mass deviates by more than 1% from the reference value, mass is artificially injected into or extracted from the reservoir to restore correct total charge value.

4. VCS data-driven surrogate model

4.1. Surrogate model

A surrogate model (Gramacy, 2020) is a computationally efficient model designed to approximate the behavior of complex systems or processes that are expensive or time-consuming to simulate directly. This compact surrogate, can be streamlined for various tasks such as prediction, uncertainty quantification, and optimization. This work focuses on regression surrogate models for prediction, where the output of the surrogates are real numbers.

In a regression setting, given observed input–output pairs $\{X, Y\}$ and a true underlying function f, the goal is to predict the value $y_* := f(\mathbf{x}_*)$ for an unobserved input $\mathbf{x}_*$. The input set is defined as $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$, where each input $\mathbf{x}_i \in \mathbb{R}^d$ and d represents the dimensionality of the input space. Correspondingly, the output set is $Y = [y_1, y_2, \ldots, y_N]$, where each output $y_i := f(\mathbf{x}_i) \in \mathbb{R}$. In our case, the true function to be modeled, f, is the VCS physics-based model.

To this end, Gaussian Processes and Multi-Layer Perceptrons are utilized to build surrogate models for the VCS. These techniques are applied to develop predictive models that approximate the system's behavior, creating a computationally efficient twin of the direct simulations. We selected these two models based on the behavior of our target outputs. Outputs exhibiting high non-linearity are better suited for modeling using MLP, whereas smoother outputs are more effectively captured using GPs, which are also inherently data-efficient. Both models have demonstrated strong performance on comparable tasks (Loka et al., 2022, 2023). Another advantage of these models is their easy integration with another framework (in this case Modelica), as their prediction code is straightforward to implement, as detailed in Section 4.4.

4.1.1. Gaussian process regression

A Gaussian Process (GP) (Rasmussen and Williams, 2018) is a widely used and data-efficient surrogate model for solving regression tasks. Compared to other models, GPs provide flexible, non-parametric modeling, making them well-suited for small data sets by incorporating prior knowledge through the kernel to handle complex, non-linear patterns with limited data.

A GP is defined by a mean function $m(\mathbf{x})$ and a covariance (kernel) function $k(\mathbf{x}, \mathbf{x}')$. The mean function represents the expected output, while the kernel function defines the relationship between input points, capturing the data's structure. Another way to view the kernel is as a prior, determining the smoothness and shape of the unknown function. Given input data X and its evaluations, the GP prior is expressed as $f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$. A zero mean function is used in all GP models, as it is a common choice.

For the kernel function, the Matérn 5/2 (Handcock and Stein, 1993) is chosen because it allows for flexible modeling without enforcing strong smoothness assumptions on the unknown function to be modeled (Genton, 2002). The Matérn 5/2 kernel is defined as:

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \gamma\left(1 + \sqrt{5}r + \frac{5}{3}r^2\right)\exp(-\sqrt{5}r) \tag{3}$$

$$r = \sqrt{\sum_{m=1}^{d} \frac{\left(x_m - x'_m\right)^2}{\ell_m^2}} \tag{4}$$

where γ is the scale parameter and $\boldsymbol{\ell} := (\ell_1, \ell_2, \ldots, \ell_d)$ is the length scales parameter, both influencing the smoothness and spread of the GP model.

In GP regression, we assume that the observed outputs $\mathbf{y} := f(X)$ and the test output $f(X_\star)$ follow a joint Gaussian distribution. The joint distribution is given by:

$$\begin{bmatrix} f(X) \\ f(X_\star) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_X \\ \mu\left(X_\star\right) \end{bmatrix}, \begin{bmatrix} K_{\mathrm{xx}} & K_{\star\mathrm{x}} \\ K_{\star\mathrm{x}}^T & K_{\star\star} \end{bmatrix}\right) \tag{5}$$

where μ_{x} is the mean of the observed points outputs, $\mu\left(X_\star\right)$ is the mean of the test outputs, $K_{\mathrm{xx}} = k(X, X)$, $K_{\star\mathrm{x}} = k(X_\star, X)$, and $K_{\star\star} = k(X_\star, X_\star)$.

Then using the standard rules of conditioning Gaussian formula, the prediction on a new data point X_* can be done by computing the predictive mean $\mu\left(X_\star\right)$ and variance $\sigma^2\left(X_\star\right)$ as follows:

$$\mu\left(X_\star\right) = K_{\star\mathrm{x}} K_{\mathrm{xx}}^{-1} Y \tag{6}$$

$$\sigma^2\left(X_\star\right) = K_{\star\star} - K_{\star\mathrm{x}} K_{\mathrm{xx}}^{-1} K_{\star\mathrm{x}}^T \tag{7}$$

To train the GP (fit the model on a dataset), hyperparameters of the kernel $\theta = (\gamma, \boldsymbol{\ell})$ are estimated using Maximum Likelihood Estimation (MLE) (Rasmussen and Williams, 2018):

$$\hat{\theta} = \arg\max_{\theta} \log p(\mathbf{f} \mid X, \theta) \tag{8}$$

$$\hat{\theta} = \arg\max_{\theta} -\frac{1}{2}\left(\log\left|2\pi K_{\mathbf{xx}}\right| + \mathbf{f}^T K_{\mathbf{xx}}^{-1}\mathbf{f}\right) \tag{9}$$

For building the GP models, the GPFlow library (Matthews et al., 2017) is used, and Eq. (9) is optimized with the LBFGS-B algorithm (Zhu et al., 1997).

4.1.2. Multi-layer perceptron

The Multi-Layer Perceptron (MLP) (Hinton, 1989), a popular deep neural network architecture, is commonly used for regression tasks due to its ability to capture complex, non-linear relationships between input and output variables (Bishop, 1995). Although MLPs are typically less data-efficient than GPs, they are more scalable and computationally efficient with larger data sets. This scalability makes MLPs a practical choice for surrogate modeling when larger data sets are available, allowing them to approximate challenging functional dependencies across a wide input space (Hornik et al., 1989).

An MLP consists of multiple layers. First is an Input Layer that receives the input data, followed by $L-1$ Hidden Layers that transform the output of the Input Layer through a sequence of weighted sums and activation functions, and finally, an Output Layer that produces the network's prediction. The Input Layer corresponds to the number of features in the input data, while the Output Layer aligns with the dimensions specified by the target function. Typically, an MLP (Prince, 2023) is represented by the following general formula:

$$\begin{aligned}
\mathbf{h}_0 &= \left(b_0 + \mathbf{W}_0\mathbf{x}\right) \\
\mathbf{h}_1 &= \mathbf{a}\left(b_1 + \mathbf{W}_1\mathbf{h}_0\right) \\
\mathbf{h}_2 &= \mathbf{a}\left(b_2 + \mathbf{W}_2\mathbf{h}_1\right) \\
&\vdots \\
\mathbf{h}_{L-1} &= \mathbf{a}\left(b_{L-1} + \mathbf{W}_{L-1}\mathbf{h}_{L-2}\right) \\
\hat{\mathbf{y}} &= b_L + \mathbf{W}_L\mathbf{h}_{L-1}.
\end{aligned} \tag{10}$$

where $\mathbf{h}_0$ represents the input layer, $(\mathbf{h}_1, \ldots, \mathbf{h}_{L-1})$ denote the hidden layers, and $\hat{\mathbf{y}}$ is the output layer. In each layer, b_l and $\mathbf{W}_l$ are the bias term and weight parameters at the lth layer, and $\mathbf{a}(x)$ is the activation function. The Rectified Linear Unit (ReLU) is used as the activation function, defined as $\mathbf{a}(x) = \max(0, x)$, due to its simplicity and effectiveness in mitigating the vanishing gradient problem (Fukushima, 1969).

For this study, the base architecture of the MLP consists of three hidden layers with 150, 100, and 50 neurons, respectively, as shown in Fig. 4. This architecture configuration was chosen as it performs well on a similar task (Loka et al., 2023). The MLP surrogates were trained using the Scikit-learn machine learning library (Pedregosa et al., 2011). To determine the optimal values for b_l and $\mathbf{W}_l$ across all layers l from 0 to L, the ADAM optimizer (Kingma and Ba, 2015) is used.

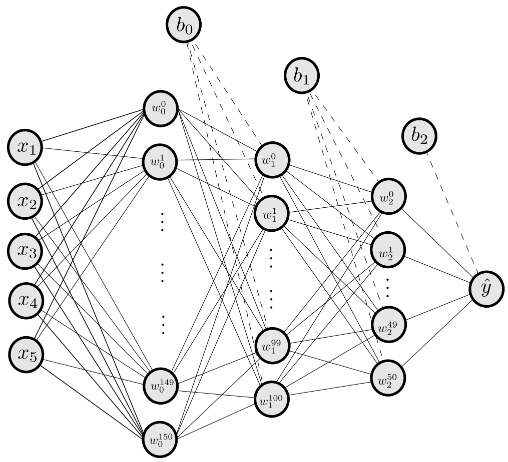

Fig. 4. Architecture of the MLP used in this paper.

Table 1
Specification of the vapor compression system surrogate: Input domain.

Variable	Description	Minimum	Maximum	Unit
$\dot{m}_{air}$	air mass flow rate[a]	$0.18\dot{m}_{nom}$	$1.05\dot{m}_{nom}$	kg/s
$T_{air,in}$	air inlet temperature	−20	60	°C
$p_{air,out}$	air outlet pressure	15,000	100,000	Pa
$T_{liq,in}$	liquid inlet temperature	10	30	°C
ω	compressor rotational speed[a]	$0.28\dot{\omega}_{nom}$	$1.18\dot{\omega}_{nom}$	rad/s

[a] Range derived from VCS reference nominal values.

Table 2
Specification of the vapor compression system surrogate: Outputs domain.

Variable	Description	Unit
$\dot{W}$	compressor power	W
$\dot{Q}_{cond}$	condenser heat transfer rate	W
$\dot{Q}_{evap}$	evaporator heat transfer rate	W
$\dot{m}$	refrigerant mass flow rate	kg/s
$T_{air,out}$	air outlet temperature	°C
$T_{liq,out}$	liquid outlet temperature	°C
$p_{air,in}$	air inlet pressure	Pa
$p_{liq,in}$	liquid inlet pressure	Pa
ΔT_{sh}	degree of super-heating	°C

4.2. Data generation

Data collection for the VCS surrogate models was carried out by evaluating the physical model developed in the Modelica framework. Three data sets were prepared from the input variables and ranges detailed in Table 1.

The first two data sets, containing 150 and 80,000 points, were generated using the Halton random sequence (Owen, 2017). The third data set, consisting of 100,000 random points, was used to test and select the surrogate model (Gramacy, 2020). A larger test dataset was employed compared to the training data because sampling from the simulator is not very expensive, allowing for a more thorough assessment of the surrogate model's performance. An example of a dataset generated with the Halton sequence is shown in Fig. 5.

4.3. Surrogate model validations and selection

Multiple sets of data-driven surrogate models were constructed and trained using the generated data discussed in the previous section. Table 2 provides a detailed list of all surrogate models developed in this study, along with the corresponding outputs of the physics-based simulator that these surrogate models represent.

Model selection was then conducted using a validation dataset of 100,000 points across the following surrogate model scenarios:

Table 3
Testing Relative Root Mean Squared Error (RRMSE) of all compared methods.

Output	RRMSE			
	MLP-80K	MLP-150	GP-150	RF-150
$\dot{W}$	**1.8345e − 02**	8.9119e−02	4.4013e−02	1.6675e−01
$\dot{Q}_{cond}$	**1.1366e − 02**	7.2061e−02	4.6298e−02	1.5313e−01
$\dot{Q}_{evap}$	**1.2132e − 02**	7.0830e−02	4.8576e−02	1.4034e−01
$\dot{m}$	**1.0329e − 02**	7.1366e−02	4.9600e−02	1.2651e−01
$\lvert T_{air,in} - T_{air,out}\rvert$	**1.6397e − 03**	1.0356e−02	6.1467e−03	2.0929e−02
$\lvert T_{liq,in} - T_{liq,out}\rvert$	**9.3428e − 04**	5.0575e−03	3.1052e−03	1.1474e−02
$p_{air,in}$	3.4875e−03	2.5127e−02	**5.6714e − 05**	9.2031e−03
$p_{liq,in}$	5.1681e−06	5.0932e−05	**6.7169e − 07**	1.3703e−05
ΔT_{sh}	**8.0247e − 02**	3.1487e−01	3.3221e−01	4.2126e−01

- MLP trained on 80,000 points (MLP-80K).
- MLP trained on 150 points (MLP-150).
- GP trained on 150 points (GP-150).
- Random Forest (RF) trained on 150 points (RF-150).

The Root Mean Squared Error (RMSE), and Relative Root Mean Squared Error (RRMSE) as defined in Eqs. (11) and (12), were used to evaluate the performance of the surrogate models. Table 3 shows the complete benchmark results.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{11}$$

$$\text{RRMSE} = \frac{\text{RMSE}}{\bar{y}}, \quad \text{where } \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{12}$$

The surrogate model with the lowest RRMSE was selected as the final model for each output. In most cases, the MLP-80K model achieved the lowest RRMSE compared to the other methods. However, for the $p_{air,in}$ and $p_{liq,in}$ outputs, the GP model outperformed the MLP. This is likely because these outputs exhibit less non-linear behavior compared to the others, making them more suitable for modeling with a GP than with MLPs. Consequently, all outputs are modeled using the MLP-80K except for $p_{air,in}$ and $p_{liq,in}$, where the GP model is preferred.

4.4. Models integration into modelica

The resulting data-driven surrogate models, originally implemented in Python, have been integrated into the Modelica framework. In this work Python was used for prototyping due to its flexibility, ease of use, and support for rapid development and testing. For integration with Modelica, the optimal parameters obtained during training are stored as matrices on MATLAB files, as they are compatible with the Modelica framework. For the GP model, these files store the training data, kernel parameters, and precomputed terms that are independent of newly observed data (i.e., $K_{XX}^{-1}Y$). This speeds up the computation of predictions since the expensive matrix inverse operation does not need to be recalculated every time a new point needs to be predicted. Additionally, for the MLP model, the resulting $\mathbf{W}_l$ and b_l terms are saved, along with the activation functions at each layer.

The integration process utilizes specific functions from the Modelica Standard Library (Fritzson, 2012) to import matrices and retrieve their dimensions. The surrogate model prediction routine was implemented in Modelica using Eqs. (6) and (10) for GP and MLP, respectively. The calculations conducted with the implemented routine in Modelica have been compared against the original code developed with Python showing a perfect match.

5. Physics-based vs. Surrogate models

The computing performance of the VCS physics-based model presented in Section 3 is compared against the VCS surrogate model

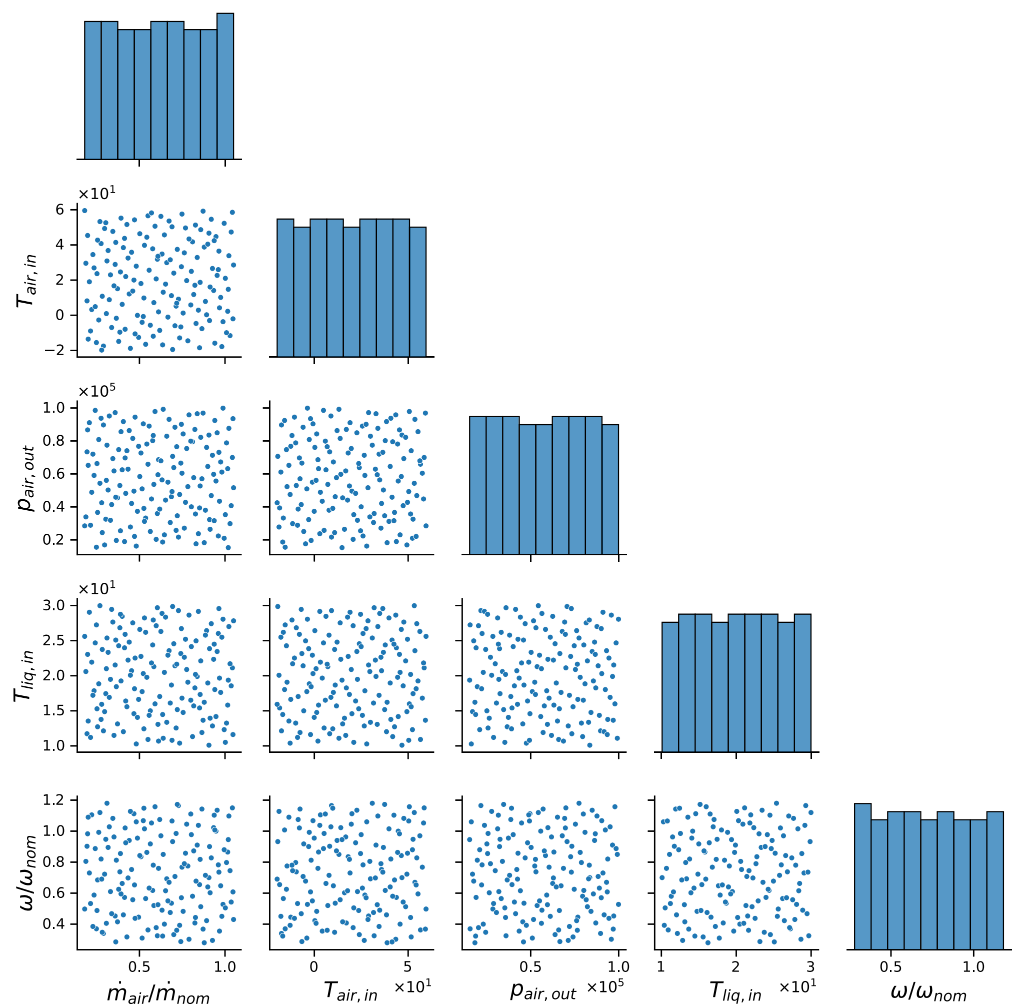

Fig. 5. An example of a 150-point dataset generated using a quasi-random Halton sequence, ensuring the points are random yet space-filling in the input domain. The lower triangular boxes display pair plots for variable pairs, while the diagonal boxes show normalized histograms representing the distribution of input values for each variable along the x-axis.

presented in Section 4, both running under the Modelica/Dymola environment. The default integration algorithm DASSL has been used for all the simulations conducted in this work. It was chosen as the reference solver for several reasons. It is an implicit, higher-order, multi-step solver with step-size control, making it stable across a wide range of models. Furthermore, it has a mature and well-tested codebase. The specifications of the PC used for all simulations are: Intel(R) Core (TM) i7-1165G7 CPU @ 2.80 GHz with 16384MB RAM.

5.1. VCS steady-state simulation

This initial test examines the computational speed for calculating steady-state conditions of the VCS. On the one hand, the physics-based model achieves the steady-state dynamically by setting an output interval of 1 s and applying specific convergence criteria to relevant system derivatives ($d\phi/dt < \epsilon$). The simulation is conducted dynamically and stops when the aforementioned criteria are met. On the other hand, the surrogate model reaches the steady-state condition in the very first time step calculation as no dynamic terms are computed.

A dataset of 90 different testing cases was created, based on various combinations of the VCS boundary condition values such as condenser air inlet conditions, evaporator liquid inlet conditions, and compressor operating speed. All conditions in the dataset are within the physically feasible operating range. For instance, the condenser air inlet temperature varies from −20 to 60 °C, while the evaporator liquid inlet temperature ranges from 10 to 30 °C.

The discrepancy of predicted values by the physics-based and surrogate models is evaluated. Fig. 6 shows the comparison between some representative variables of the VCS, namely, evaporation heat flow rate, refrigerant mass flow rate and condenser heat flow rate. The parameter used for the comparisons is the Prediction Discrepancy (PD) which characterizes the difference between the physics-based and the surrogate model predicted values, evaluated as follows:

$$\mathrm{PD} = \frac{|\theta_{surrogate} - \theta_{physics-based}|}{\theta_{physics-based}} \times 100 \tag{13}$$

To assess the comparison for the whole data, the Mean Prediction Discrepancy (MPD) is calculated:

$$\mathrm{MPD} = \frac{1}{N}\sum_{i=1}^{i=N} \mathrm{PD}_i \tag{14}$$

The MPDs are detailed in Table 4 for representative variables predicted by both VCS models. Regarding the assessment of predicted temperatures, a temperature difference value is used $|T_{surrogate} - T_{physics-based}|$ (TDiff) instead of the PD. The mean temperature difference (MTDiff) is calculated using the same method as the MPD. The models show good agreement as both MPD and MTDiff values are relatively low.

Fig. 7 illustrates the CPU time consumption results for each individual case (the index value corresponds to the numbering of test cases). The average CPU time and standard deviation are 2.30 and 0.65 s for

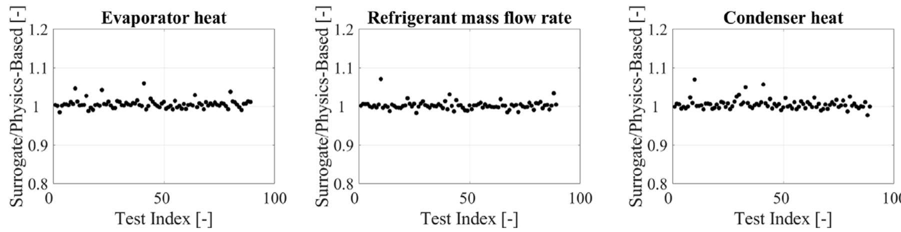

Fig. 6. Graphical representation of discrepancies between VCS physics-based and surrogate models predictions for VCS steady-state tests.

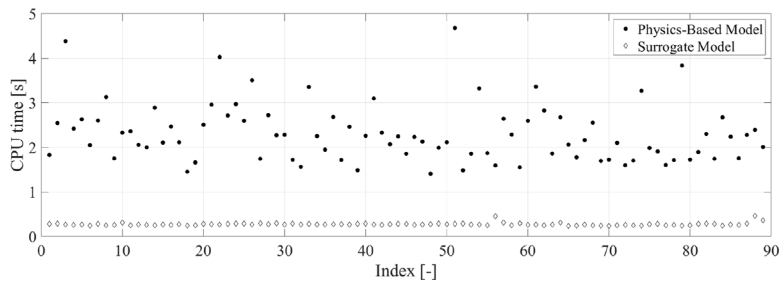

Fig. 7. Time consumption for steady-state VCS simulations.

Table 4
Numerical assessment of discrepancies between VCS physics-based and surrogate models predictions for VCS steady-state tests.

Parameter	MPD [%]	MTDiff [K]
Refrigerant mass flow rate	0.66	–
Evaporator heat flow rate	0.86	–
Condenser heat flow rate	0.89	–
Compressor power	2.85	–
Air outlet temperature	–	0.34
Liquid outlet temperature	–	0.14

the physics-based model, and 0.27 and 0.03 s for the surrogate model. The surrogate model is significantly more efficient, requiring an order of magnitude less time than the physics-based model. It also shows greater consistency with similar processing times across all cases.

5.2. ECS steady-state simulation

The performance of the VCS physics-based and surrogate models has also been evaluated as subsystems interacting with an ECS architecture, rather than as standalone systems. The present test focuses on assessing the accuracy and computational efficiency of calculating steady-state conditions for an ECS model that incorporates a VCS. To conduct the test, a simplified ECS model was built under the Modelica/Dymola environment, and a proper integration of the VCS surrogate model into the ECS model was required.

5.2.1. ECS model

A model for the ECS configuration depicted in Fig. 1 has been developed, focusing on the thermal section where the VCS operates. The scheme of the ECS model is shown in Fig. 8 for both cases, namely, using the VCS physics-based model (left), and using the corresponding VCS surrogate model (right).

The ECS model developed for this study is a simplified representation of a full environmental control system, and its primary objective is to serve as a platform for comparing the simulation performance of the physics-based VCS model with that of its surrogate counterpart. The ACS is not explicitly simulated in the current model; instead, its influence is incorporated into the cabin model in the form of heat gains and losses. The ECS model consists of three fluid loops: the air recirculation unit, the liquid circuit, and the VCS. Heat is extracted from the cabin via the air recirculation loop, transferred to the VCS evaporator through the liquid circuit, and then dissipated to the environment by the VCS condenser. The cabin is represented using simplified heat balance equations and performance maps based on data supplied by an industrial partner. Cabin temperature is controlled by adjusting the VCS compressor frequency with a PI controller.

5.2.2. Computing performance

The steady-state analysis of the ECS model reveals notable differences compared to standalone VCS simulation. First, it includes additional components and systems beyond the VCS. Second, the criteria to reach the steady-state are applied to a broader set of variables (e.g., $dT_{cabin}/dt < \epsilon$). Third, although the VCS can be solved using either steady-state surrogate models or transient physics-based models, the overall process remains dynamic due to the presence of dynamic terms in the ECS components and the continuous adjustment of the compressor speed by a PI controller to regulate the cabin temperature.

A dataset of 90 test cases was generated, based on various flight conditions with differing cooling requirements and altitudes ranging from 0 to 40,000 ft. The ECS boundary conditions are derived from the airplane altitude as follows:

- The ambient temperature T_{amb} is determined based on the altitude, using the standard relationship for a hot day as defined in ASHRAE (2019), which results in values within −20 and 60 °C.
- The ram air temperature T_{ram} is adjusted according to the aircraft speed (ranging from Mach 0 to 0.8) using the following equation based on the Mach number M and the heat capacity ratio γ:

$$T_{ram} = T_{amb}\left(1 + M^2\left(\frac{\gamma - 1}{2}\right)\right) \tag{15}$$

- The cabin heat gains and losses are obtained from performance maps provided by the manufacturer, which are based on altitude, aircraft speed, number of passengers, and other cooling needs.

The discrepancies between the VCS physics-based and surrogate models predictions were investigated. Fig. 9 provides a graphical comparison of key VCS variables, including the evaporator heat flow rate, refrigerant mass flow rate, and condenser heat flow rate. Perfect agreement is observed for the evaporator heat flow rate, as expected, since the ECS steady-state for any given condition demands a specific heat flow to be removed by the VCS. Both the physics-based and surrogate

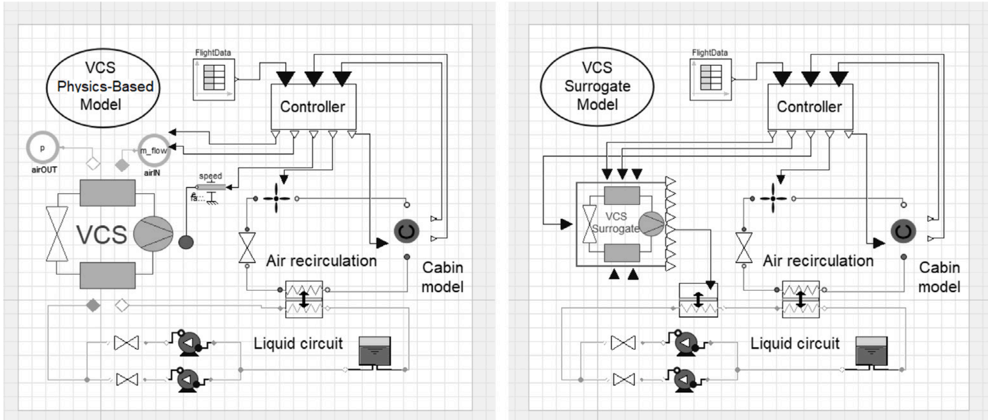

Fig. 8. ECS graphical scheme: VCS physics-based model (left) and VCS surrogate model (right).

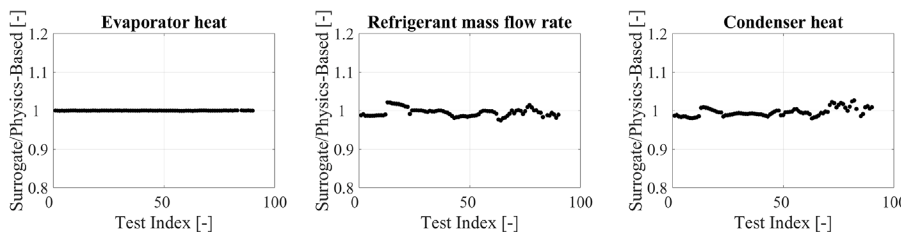

Fig. 9. Graphical representation of discrepancies between VCS physics-based and surrogate models predictions for ECS steady-state tests.

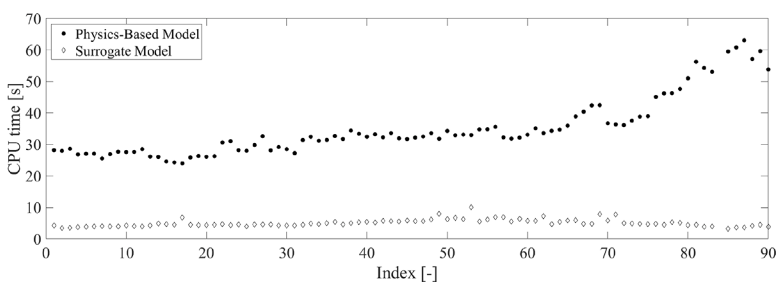

Fig. 10. Time consumption for steady-state ECS simulations.

Table 5
Numerical assessment of discrepancies between VCS physics-based and surrogate models predictions for ECS steady-state tests.

Parameter	MPD [%]	MTDiff [K]
Refrigerant mass flow rate	0.91	–
Evaporator heat flow rate	0.02	–
Condenser heat flow rate	0.97	–
Compressor power	2.13	–
Air outlet temperature	–	0.33
Liquid outlet temperature	–	0.18

models converge on the same evaporator heat flow rate, with only slight discrepancies in the predictions of other VCS parameters. The MPD and MTDiff values are relatively low, indicating good agreement between the two models as detailed in Table 5.

Fig. 10 shows the CPU time consumption for each individual case. The case based on the VCS physics-based model has an average CPU time of 35.05 s with a standard deviation of 9.33 s, while the case based on the surrogate model averages 5.04 s with a standard deviation of 1.15 s. The ECS incorporating the surrogate model is not only more consistent in CPU time consumption but also significantly more efficient, requiring nearly seven times less computational effort.

5.3. ECS dynamic simulation

The ECS model described in the previous section was also tested under transient conditions to compare the performance of the VCS steady-state surrogate model against the dynamic physics-based model. The objective was to evaluate the computational efficiency of the VCS surrogate model while examining the impact of ignoring time-dependent factors when using a VCS steady-state approach in a dynamic ECS simulation. For this purpose, a complete dynamic simulation of the ECS was conducted based on a flight scenario with high cooling demands.

5.3.1. Flight journey

The studied flight mission profile is depicted in Fig. 11 and has been derived from public available information as detailed in Ablanque et al. (2024). It focuses on a scenario with high cabin cooling requirements and an atmospheric temperature profile representing a hot day. The aircraft altitude versus time curve is based on an Airbus A320 flight,

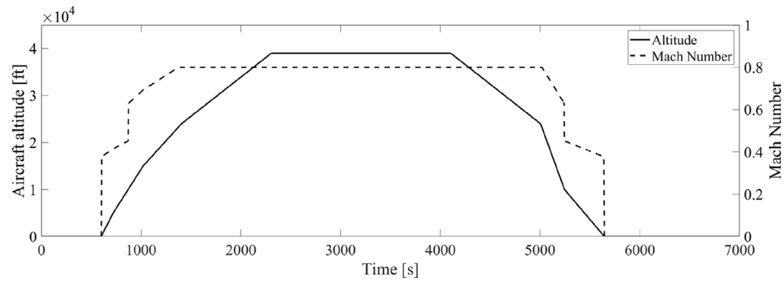

Fig. 11. Flight mission altitude and speed profiles.

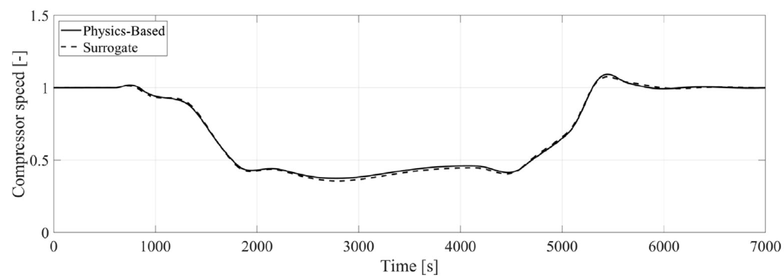

Fig. 12. ECS: VCS compressor speed changes during flight (dimensionless).

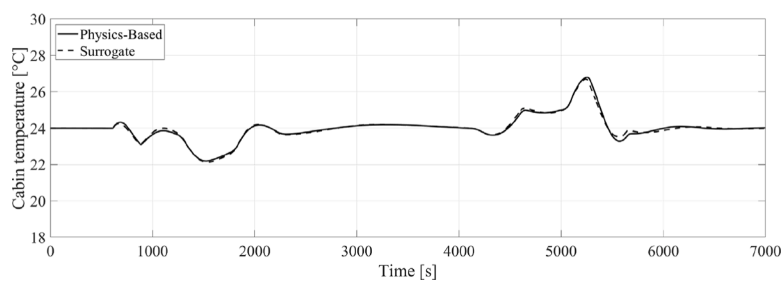

Fig. 13. ECS: cabin temperature evolution during flight.

and the statistical speed data adheres to air traffic regulations. The total mission time is 106 min, including 30 min of cruising, and 10 min of taxiing at both the beginning and ending of the flight.

5.3.2. Computing performance

The dynamic response of the ECS model was compared between the two approaches: one using the VCS physics-based model and the other using the VCS surrogate model. Figs. 12 and 13 display the compressor speed required by the PI controller and the cabin temperature throughout the flight, respectively (the compressor speed is reported as a dimensionless value based on the operating value at ground). The overall dynamic response of the ECS model during the flight mission is observed to be very similar for both approaches. This indicates that, for this particular case, the steady-state surrogate model is reasonably appropriate, as the absence of VCS internal dynamics leads to minimal overall discrepancies.

Fig. 14 illustrates the CPU time consumption for both approaches. When the ECS operates with the physics-based VCS model, it requires 32.30 s to simulate the entire flight. In contrast, the computational effort for the ECS operating with the VCS surrogate model varies based on the number of outputs calculated. If only the evaporator heat flow output is considered, the CPU time drops to 6.99 s, representing a 78% reduction compared to the physics-based model. However, including all outputs in the calculation (see Table 2) increases the time to 72.10 s, which is more than 123% higher than that of the physics-based model.

During the dynamic simulation, the VCS model is subjected to varying boundary conditions, including, condenser and/or evaporator secondary flow characteristics, and compressor speed. Throughout the considered flight journey, there are several instances where any of these conditions change rapidly, resulting in different responses between the VCS physics-based and surrogate models (the physics-based model will present a dynamic response while the surrogate model will present a more abrupt response due to its steady-state nature). As shown in Figs. 12 and 13, there is no significant impact at the ECS level; however, the differences can be seen at the internal level of the VCS. To highlight this, both responses are detailed in Fig. 15, focusing on the dimensionless value of the evaporator heat flow rate. Two specific moments during the flight, where condenser air conditions undergo rapid changes, are analyzed: (i) at 870 s when the aircraft is climbing and accelerates, and (ii) at 5450 s when the aircraft is descending and decelerates as observed in Fig. 11. If these boundary condition changes were substantial enough, the impact on the overall ECS results could be significant, making the surrogate steady-state approach less suitable (i.e., VCS start-up and shut-down operations).

6. Conclusions

This paper explores the development and implementation of VCS surrogate models from a computational perspective using simulations in the Dymola/Modelica framework. The focus is on simulating VCS models within larger architectures, such as ECS. Several key conclusions and findings are highlighted:

- Data-driven surrogate models have been created to calculate key VCS outputs based on training data generated by a physics-based model. Two approaches were employed: Gaussian Process

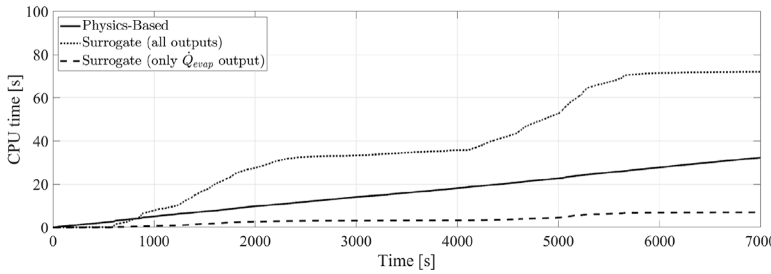

Fig. 14. CPU time consumption for flight transient simulation.

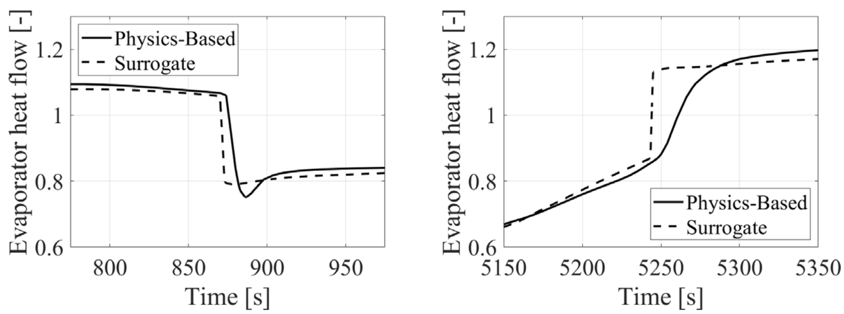

Fig. 15. VCS dimensionless evaporator heat flow at two different moments of the flight where condenser secondary flow conditions change rapidly.

(more suitable for outputs exhibiting more linear behavior) and Multi-Layer Perceptron (more effective for outputs with greater nonlinearities).

- The surrogate models have been successfully integrated into the Dymola/Modelica environment, enabling their interoperability with conventional Modelica-structured models.
- The computational performance of the data-driven steady-state surrogate models has been assessed in various scenarios against the physics-based model, demonstrating significantly lower time consumption: approximately 10, 7, and 5 times less computational effort for single VCS steady-state simulations, ECS steady-state simulations, and ECS dynamic simulations, respectively.
- The VCS surrogate model possesses certain drawbacks, including the need for additional computational resources during the training process and its inability to handle physical conditions not represented in the training data. Additionally, the computational time needed for the ECS dynamic simulations based on the VCS steady-state surrogate model can increase significantly depending on the number of outputs required. When numerous outputs are needed, the physics-based VCS model may offer a more efficient alternative.

This study investigates the application of steady-state VCS surrogate models for ECS simulations as an initial step. Future work will focus on developing and implementing dynamic VCS surrogate models to effectively address substantial dynamic variations.

CRediT authorship contribution statement

Nicolás Ablanque: Writing – original draft, Conceptualization, Writing – review & editing, Data curation, Formal analysis. **Nasrulloh Loka:** Formal analysis, Writing – review & editing, Writing – original draft, Data curation. **Santiago Torras:** Data curation. **Sriram Gurumurthy:** Software, Data curation. **Joaquim Rigola:** Funding acquisition, Project administration. **Carles Oliet:** Supervision, Validation. **Ivo Couckuyt:** Supervision, Validation. **Tom Dhaene:** Funding acquisition, Project administration. **Antonello Monti:** Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This project has received funding from the Clean Sky 2 Joint Undertaking (JU) under grant agreement No 886533. The JU receives support from the European Union's Horizon 2020 research and innovation program and the Clean Sky 2 JU members other than the Union.

Horizon 2020 European Union Funding for Research & Innovation

Clean Sky 2 JOINT UNDERTAKING

This work has also been supported by the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" and the "Fonds Wetenschappelijk Onderzoek (FWO)" programs.

Carles Oliet, as a Serra Húnter Associate Professor, acknowledges the Catalan Government for the support through this Programme.

Disclaimer: The contents presented in this article reflect only the author's point of view: the authors and Clean Sky JU are not responsible for any use that may be made of the information it contains.

References

Ablanque, N., Torras, S., Rigola, J., Oliet, C., 2024. Dynamic model of vapor compression systems for aircraft environmental control systems. J. Aircr. 61 (5), 1387–1396. http://dx.doi.org/10.2514/1.C037711.

ASHRAE, 2019. ASHRAE Handbook - HVAC Applications. Chapter 13 - Aircraft. American Society of Heating Refrigerating and Air-Conditioning Engineers.

Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Oxford University Press.

Erkinaci, T., Coskun, F., Cotur, A., Onler, R., Serincan, M.F., 2024. Experimental and numerical investigation of a compact vapor compression refrigeration system for cooling of avionics in harsh environments. Appl. Therm. Eng. 236, http://dx.doi.org/10.1016/j.applthermaleng.2023.121663.

Fritzson, P., 2012. The modelica standard library. pp. 157–168. http://dx.doi.org/10.1002/9781118094259.ch5.

Fukushima, K., 1969. Visual feature extraction by a multilayered network of analog threshold elements. IEEE Trans. Syst. Sci. Cybern. 5 (4), 322–333. http://dx.doi.org/10.1109/TSSC.1969.300225.

Genton, M.G., 2002. Classes of kernels for machine learning: A statistics perspective. J. Mach. Learn. Res. 2, 299–312. http://dx.doi.org/10.5555/944790.944815.

Golle, S., Hesse, U., 2019. Comparison of novel environmental control systems in aircraft based on vapor compression cycle. In: 25th IIR International Congress of Refrigeration. Montréal, Canada, http://dx.doi.org/10.13140/RG.2.2.30731.64804.

Gramacy, R.B., 2020. Surrogates: Gaussian process modeling, design, and optimization for the applied sciences. Chapman and Hall/CRC.

Handcock, M.S., Stein, M.L., 1993. A Bayesian analysis of kriging. Technometrics 35 (4), 403–410. http://dx.doi.org/10.2307/1270273.

Hinton, G.E., 1989. Connectionist learning procedures. Artificial Intelligence 40 (1), 185–234. http://dx.doi.org/10.1016/0004-3702(89)90049-0.

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Netw. 2 (5), 359–366.

Hu, H., Sun, H., Wu, C., Wang, X., Lv, Z., 2020. A steady-state simulation model of supplemental cooling system integrated with vapor compression refrigeration cycles for commercial airplanes. Appl. Therm. Eng. 166, http://dx.doi.org/10.1016/j.applthermaleng.2019.114692.

Jiang, H., Dong, S., Zhang, H., Ai, F., Zhang, Z., Wang, J., 2020. Optimization on conventional and electric air-cycle refrigeration systems of aircraft: A short-cut method and analysis. Chin. J. Aeronaut. 33 (7), 1877–1888. http://dx.doi.org/10.1016/j.cja.2020.02.021.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (Eds.), ICLR (Poster).

Laughman, C.R., Qiao, H., 2016. Mass conserving models of vapor compression cycles. In: 11th International Modelica Conference. Versailles, France, http://dx.doi.org/10.3384/ecp15118759.

Li, Y., Hu, H., Lei, R., 2024. Dynamic modelling of supplemental cooling systems with vapor compression refrigeration units in civil aircraft. Appl. Therm. Eng. 248, http://dx.doi.org/10.1016/j.applthermaleng.2024.123285.

Loka, N.R.B.S., Ablanque, N., Torras, S., Gurumurthy, S.K., Monti, A., Rigola, J., Oliet, C., Couckuyt, I., Dhaene, T., 2023. Heat exchanger surrogates for a vapor compression system. In: Modelica Conferences. pp. 599–606.

Loka, N.R.B.S., Gurumurthy, S.K., Amevor, B., Monti, A., Dhaene, T., Couckuyt, I., 2022. Surrogate modelling of dynamic phasor simulations of electrical drives. In: IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society. IEEE, pp. 1–6.

Ma, J., Dong, Y., Qiao, H., Laughman, C.R., 2024. A physics-constrained deep learning framework for dynamic modeling of vapor compression systems. Appl. Therm. Eng. 254 (5), http://dx.doi.org/10.1016/j.applthermaleng.2024.123734.

Matthews, A.G.G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., Hensman, J., 2017. GPflow: A Gaussian process library using TensorFlow. J. Mach. Learn. Res. 18 (40), 1–6.

Merzvinskas, M., Bringhenti, C., Tomita, J., Andrade, C.R., 2019. Air conditioning systems for aeronautical applications: a review. Aeronaut. J. 124, 1–34. http://dx.doi.org/10.1017/aer.2019.159.

Owen, A.B., 2017. A randomized Halton algorithm in R. http://dx.doi.org/10.48550/arXiv.1706.02808.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830. http://dx.doi.org/10.5555/1953048.2078195.

Pérez-Grande, I., Leo, T.J., 2002. Optimization of a commercial aircraft environmental control system. Appl. Therm. Eng. 22 (17), 1885–1904. http://dx.doi.org/10.1016/S1359-4311(02)00130-8.

Pollock, A., 2018. Modeling and Control of Aircraft Environmental Systems (Ph.D. thesis). Polythecnic University of Milan.

Prince, S.J., 2023. Understanding Deep Learning. MIT Press, URL https://udlbook.github.io/udlbook/.

Rasmussen, C.E., Williams, C.K.I., 2018. Gaussian Processes for Machine Learning. The MIT Press, Cambridge, Massachusetts, http://dx.doi.org/10.7551/mitpress/3206.001.0001.

Santos, A.P.P., Andrade, C.R., Zaparoli, E.L., 2014. A thermodynamic study of air cycle machine for aeronautical applications. Int. J. Thermodyn. 17, 117–126. http://dx.doi.org/10.5541/ijot.538.

Sarlioglu, B., Morris, T., 2015. More electric aircraft: Review, challenges, and opportunities for commercial transport aircraft. IEEE Trans. Transp. Electrification 1 (1), 54–64. http://dx.doi.org/10.1109/TTE.2015.2426499.

Shao, L.-L., Yang, L., Zhao, L.-X., Zhang, C.-L., 2012. Hybrid steady-state modeling of a residential air-conditioner system using neural network component models. Energy Build. 50, 189–195. http://dx.doi.org/10.1016/j.enbuild.2012.03.036.

Shen, H., Zhang, C., Zhang, J., Yang, B., Jia, B., 2019. Applicable and comparative research of compressor mass flow rate and isentropic efficiency empirical models to marine large-scale compressor. Energies 13, 47. http://dx.doi.org/10.3390/en13010047.

Smith, A., Childs, T., Chen, R., 2018. Study into electrically shaft driven air cycle machines. In: 1st International Conference on Advances in Aerospace Structures, Systems and Technology. London, United Kingdom.

Wang, J., Lu, X., Adetola, V., Louie, E., 2024. Modeling variable refrigerant flow (VRF) systems in building applications: A comprehensive review. Energy Build. 311, http://dx.doi.org/10.1016/j.enbuild.2024.114128.

Yousaf, S., Bradshaw, C.R., Kamalapurkar, R., San, O., 2023. Investigating critical model input features for unitary air conditioning equipment. Energy Build. 284, http://dx.doi.org/10.1016/j.enbuild.2023.112823.

Yuanchao, Y., Zichen, G., 2019. Power optimization of the environmental control system for the civil more electric aircraft. Energy 172, 196–206. http://dx.doi.org/10.1016/j.energy.2019.01.115.

Zhu, C., Byrd, R.H., Lu, P., Nocedal, J., 1997. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. ACM Trans. Math. Software 23 (4), 550–560. http://dx.doi.org/10.1145/279232.279236.