Interactive Design-of-Experiments: Optimizing a Cooling System

Rainer Splechtna, Majid Behravan, Mario Jelović, Denis Gračanin, Helwig Hauser, and Krešimir Matković

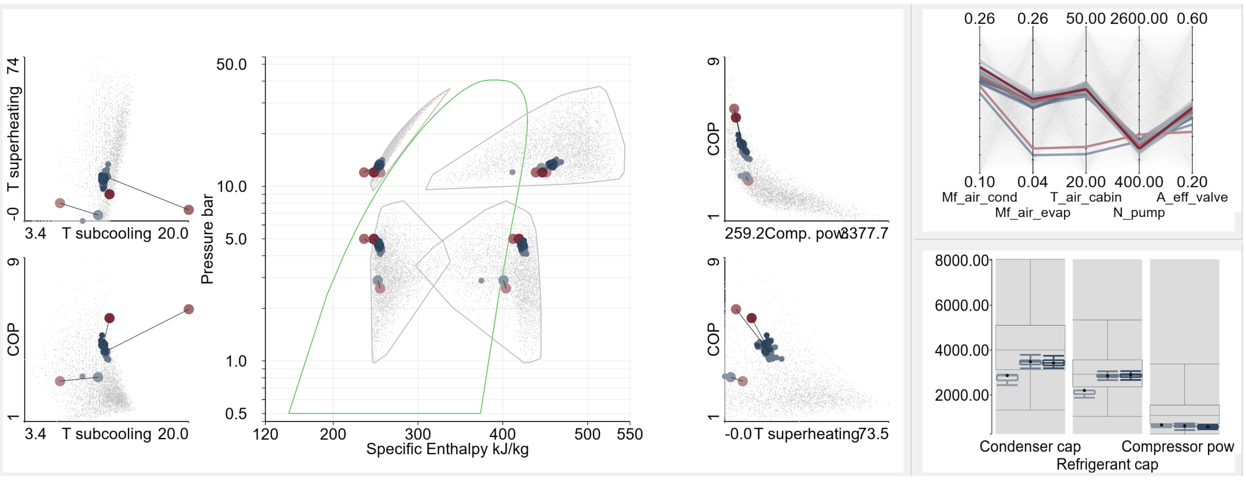


Fig. 1: Key views of the visualization component in iterative design of experiments. The interactive p-h diagram, central to cooling system design, presents multiple layers of information: user-defined desired points (in shades of red), simulated points generated by parameters predicted through deep learning (shades of blue), and scatterplots offering a dual data perspective (with lines connecting Deep Learning prediction and simulation for the same parameters). Parallel coordinates show control parameters, while box plots offer insights into numerical aggregated values. The colors in all views are consistent. Darker colors show recent iterations.

Abstract—The optimization of cooling systems is important in many cases, for example for cabin and battery cooling in electric cars. Such an optimization is governed by multiple, conflicting objectives and it is performed across a multi-dimensional parameter space. The extent of the parameter space, the complexity of the non-linear model of the system, as well as the time needed per simulation run and factors that are not modeled in the simulation necessitate an iterative, semi-automatic approach. We present an interactive visual optimization approach, where the user works with a p-h diagram to steer an iterative, guided optimization process. A deep learning (DL) model provides estimates for parameters, given a target characterization of the system, while numerical simulation is used to compute system characteristics for an ensemble of parameter sets. Since the DL model only serves as an approximation of the inverse of the cooling system and since target characteristics can be chosen according to different, competing objectives, an iterative optimization process is realized, developing multiple sets of intermediate solutions, which are visually related to each other. The standard p-h diagram, integrated interactively in this approach, is complemented by a dual, also interactive visual representation of additional expressive measures representing the system characteristics. We show how the known four-points semantic of the p-h diagram meaningfully transfers to the dual data representation. When evaluating this approach in the automotive domain, we found that our solution helped with the overall comprehension of the cooling system and that it lead to a faster convergence during optimization.

Index Terms—Parameter space exploration

1 Introduction

In today's world, the importance of cooling systems has surged notably owing to the urgent imperative to curtail CO_2 emissions and to pioneer more efficient housing and transportation. Electric vehicles are an important example, where the necessity for cooling extends beyond passenger comfort to also ensure the crucial cooling of batteries.

Several factors exacerbate the challenge of optimizing the design of a modern cooling systems. The optimization has to deal with ***multi-dimensional control and output parameter spaces*** of the simulation models of such thermodynamic systems. These models are characterized by their ***non-linear behavior*** and can only be linearized locally with a small range of sufficient validity. To be close to the physical reality, the simulation models are usually formulated as systems of ordinary differential equations or as differential algebraic equation systems. Due to the computational complexity of the necessary iterative numerical simulation, it is not feasible to densely sample the multi-dimensional parameter space and ***inverting the simulation model is not possible*** to derive the system's configuration directly from a certain, desired behavior. Accordingly, a ***design of experiments approach is needed*** to optimize across the multi-dimensional parameter space effectively and efficiently. This methodology is commonly applied to simulations utilized in the automotive domain, and also in the use case presented in this paper. It encompasses planning, conducting, analyzing, and

- *Rainer Splechtna and Krešimir Matković are with the VRVis Research Center in Vienna, Austria. E-mail: {Splechtna, Matkovic}@VRVis.at*
- *Majid Bahravan and Denis Gračanin are with Virginia Tech, Blacksburg, VA, USA. E-mail: {behravan, gracanin}@vt.edu*
- *Mario Jelović is with AVL-AST doo, Zagreb, Croatia. E-mail: mario.jelovic@avl.com.*
- *Helwig Hauser is with University of Bergen, Norway. eMail: Helwig.Hauser@UiB.no*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

interpreting controlled simulation runs with the goal of gaining insight into a system's behavior with a minimal set of cases, i.e., simulation runs, by applying statistical methods and, typically, varying multiple parameters simultaneously. Parameter sweeps are also used to explore parameter spaces by issuing simulation runs with varying parameters per case, but we adopt the term design of experiments due to its broader scope and because it is established in the automotive domain.

It should also be possible to ***consider context knowledge of the engineers***, for example, regarding the expected, limited accuracy of the simulation model, as well as other external factors such as the availability of system parts, safety concerns, policies, etc.

Identifying suitable control parameters that reliably lead to a certain, optimized behavior of the cooling system under a variety of possible operating conditions, while also respecting expectations with respect to other related aspects (costs, safety, policies, etc.), amounts to a complex ***exploratory task***. Supporting this task with a mixed-initiative solution, empowering the expert user while helping with computational optimization, based on numerical simulation as well as on machine learning, promises to limit the number of necessary exploratory iterations.

In the following, we present our approach to the interactive visual exploration and optimization of a complex systems, such as described above, that we call “Interactive Design of Experiments” (IDoE). This IDoE approach integrates the following components (Figure 2) to facilitate an improved, iterative, exploratory optimization workflow:

1. **Initial ensemble**: an early offline component, providing a starting sampling of the parameter space of the simulation (costly and still sparse, even though providing quite a number of simulation runs – sparse, because the parameter space is multi-dimensional).
2. **Interactive visualization**: provides guided interaction to facilitate the expert's exploration of possible improvements of the system configuration and establishes a visual relation between parameters and the simulated system behavior.
3. **Deep learning (DL)**: an approximate inversion model provides rapid responses for desired output parameters, estimating reasonable control parameters for regions of interest in the parameter space, specified interactively in the visualization.
4. **Numerical simulation**: provides iterative refinements of the initial ensemble based on regions of interest in the parameter space as explored iteratively in the visualization; new simulation runs are also used to further improve the DL model.

The visualization component serves as the central interface, enabling the user to interact with and to interpret the data. We exploit an initial ensemble, computed offline, and actively guide the refinement process by specifying desired outcomes in the interactive visualization. Using our DL inversion model, we estimate the corresponding control parameters for these outcomes. Numerical simulation then computes the actual outcomes for these estimated parameters. This guided, iterative process enables to progressively improve the sampling of the parameter space, focusing particularly on regions that likely produce desired outcomes, hence the term ***Interactive Design of Experiments*** (IDoE). This iterative process is usually structured by the multiple objectives or scenarios the system should be optimized for. Each scenario is individually explored through several iterations.

For our use case, i.e., the optimization of air conditioning (AC) systems, the main view in the visualization component is an interactive p-h diagram (Figure 3), which illustrates the relation between pressure and specific enthalpy of the refrigerant, corresponding to a pivotal standard diagram in AC system design. In our interactive realization of this diagram, we integrate guided interaction to aid the specification of desired output parameters. We also extend it by four scatterplots—one related to each characteristic point—realizing an informative, dual perspective on the data.

The system's design is the result of a collaboration with domain experts, one of whom is also a coauthor of this paper. We present a practical use case to demonstrate the effectiveness of our approach. The IDoE approach, however, likely has a much broader applicability to optimizing complex simulation models that cannot be inverted, i.e., a very common case. The main contributions of this application paper can be summarized as follows:

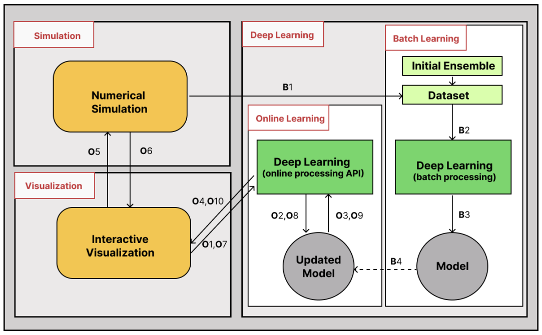


Fig. 2: The IDoE approach. Labels prefixed with a ‘B’ relate to batch learning, those with an ‘O’ correspond to online learning.

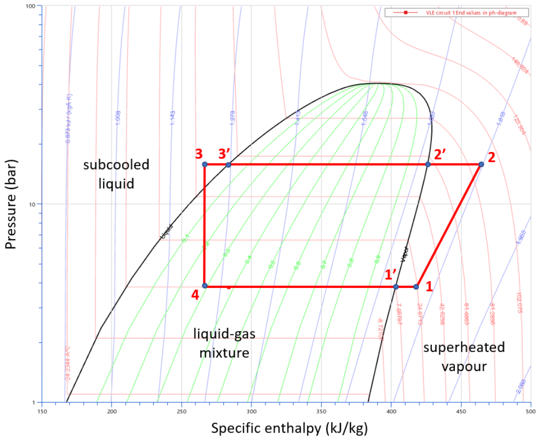


Fig. 3: An example of a p-h diagram showing the refrigeration cycle in counterclockwise direction. The characteristic points are shown in red. The connecting lines between the points correspond with state changes in the cycle: Compression (1→2), condensation (2→3), expansion (3→4) and evaporation (4→1). Condensation and evaporation pressures can also be identified in the diagram.

1. The Interactive Design of Experiments approach, enabling a rapid, guided optimization across a multi-dimensional parameter space, based on an initial simulation ensemble that is step-wise improved using two models—one for the numerical simulation of the system, as well as one for the approximate inversion of it.
2. Our visualization design of an interactive p-h diagram, enriched by a dual perspective on the data, and serving as the central interface for the guided, interactive optimization.
3. A use case, demonstrating the successful application of IDoE in tuning a cooling system in the automotive domain. The simulation model of the system in this use case has five control parameters, the initial ensemble consists of 5000 runs and a single run takes about ten seconds. Usually the analysis includes about ten scenarios, like "hot summer day", with up to a dozen iterations each. We showcase the exploration of one scenario.

2 Background

For decades, automotive air conditioning (AC) systems were primarily designed to cool the vehicle's interior. Their significance has recently surged, especially with the rise of electric vehicles, where AC systems play a crucial role in cooling batteries, inverters, and electric motors. The complexity of these systems demands meticulous thermal solutions, and their development process is time-intensive. Inefficient utilization can lead to substantial costs. Hence, it is imperative to analyze and predict the behavior of AC systems at the vehicle design stage. The early consideration of involved factors is essential for optimizing performance and minimizing potential drawbacks.

All AC systems operate according to fundamental thermodynamic principles, employing similar components and utilizing a refrigerant to achieve the desired cooling effect. The refrigerant undergoes temperature and pressure changes through the action of a compressor, transitioning between liquid and gaseous states in a closed system.

Typically, an AC system consists of four main components: the compressor, the condenser, an expansion valve, and the evaporator. While each component serves a specific function, they collaborate to realize the AC's functionality. The compressor pressurizes the gaseous refrigerant, directing it to the condenser, where heat is expelled, causing the gas to condense into a liquid under high pressure. Subsequently, the refrigerant passes through the expansion valve, where its pressure decreases, before entering the evaporator. Within the evaporator, the refrigerant evaporates, absorbing heat from the surrounding air (or another medium in the case of chillers), thereby cooling it. This heat exchange process facilitates the refrigerant's return to its gaseous state, and the cycle repeats.

Engineers represent this cycle in a pressure-enthalpy (p-h) diagram, which shows the thermodynamic properties of the refrigerant in the cooling system (Figure 3). The p-h diagram shows the refrigerant's pressure on the y-axis (logarithmic scale) and its specific enthalpy on the x-axis (linear scale). It provides a visual representation of the refrigerant's thermodynamic state as it traverses through the system's components throughout the cooling cycle. The refrigeration cycle shows up in a counterclockwise direction in the p-h diagram, with changes of state occurring at each of the four main components of the AC circuit:

1. **Compression** (1→2) of the gaseous refrigerant to condensing pressure at the compressor component.
2. *Vapor cooling* (2→2'), **condensation** (2'→3'), and *subcooling* (3'→3) of the liquid at the condenser component.
3. **Expansion** (3→4) of the liquid-gas mixture to evaporation pressure at the expansion valve component.
4. **Evaporation** (4→1') and *superheating* (1'→1) of the vapor at the evaporator component.

Points 1, 2, 3, and 4 in the p-h diagram (Figure 3) are indicators of the efficiency of the cooling cycle and are called characteristic points.

Typically, when designing a cooling system the engineers do not specify these points directly when describing the desired cooling cycle characteristics. They specify low (evaporating) and high (condensing) pressures and subcooling and superheating temperatures of the cycle instead. The characteristic points can be calculated from these values.

3 Related Work

AC systems consist of many components, including compressor, condenser, evaporator, tube, and thermostatic expansion valve. Hence, to simulate an AC system, a number of control parameters are used to specify its characteristics. The AC design space is very large and would benefit from approaches similar to ones shown for other domains, like the use of computational steering as shown by Atanasov et al. [3] or Mulder et al. [19] and parameter space exploration as described by Sedlmair et al. [23] to identify desirable system configurations.

Specifically, the functioning of AC systems heavily depends on using a refrigerant to exchange heat and produce a cooling effect. Knowing the thermophysical properties of refrigerants, including the relationship between pressure and enthalpy, is essential to identify the values of control parameters used to design and optimize AC systems. There are many software libraries that support high-accuracy simulation and evaluation of the thermophysical properties of fluids like CoolProp [4]. That also includes a number of machine learning based approaches as described by Abdul Jameel et al. [27], Folmsbee et al. [11] and Waxenegger-Wilfing [1].

Thermophysical properties can be visualized using various types of thermodynamic state diagrams, including pressure-enthalpy (ph), pressure-specific volume (pv), pressure-temperature (pT), temperature-enthalpy (Th), and temperature-entropy (Ts).

The p-h diagram (Figure 3) is an important graphical representation that shows the relationship of pressure and specific enthalpy. It is used frequently to determine the performance of AC systems and show how thermodynamic processes are working with changes of states as described by Aly et al. [2]. Pressure is shown on the y-axis (logarithmic scale) and enthalpy on the x-axis (linear scale) as described by Khennich et al. [13]. The p-h diagram shows how pressure and specific enthalpy relate to each other at certain boundary conditions.

There are many online and standalone general visualization tools, like MATLAB [25], maple [16] or Mathematica [28] and specialized visualization tools, like NIST [10] or CoolSelector [10] that support the exploration of p-h diagrams. Our approach differs, as it enables the user to specify wanted points and provides the corresponding parameters then. We also offer a dual perspective on the data.

Recently, DL methods have been used to analyze and control cooling systems as presented by Chen et al. [8], Chervonyi er al. [9] and Luo et al. [14]. In the exploration of Deep Neural Networks (DNNs) within an online setting, there are significant challenges and solutions, reflecting scenarios where data is not sequentially available, as opposed to traditional batch settings where DNNs are trained with the entire training data accessible from the outset, a method that becomes untenable in many real-world situations as shown by Sahoo et al. [22]. This is particularly true when data continuously accumulates over time or when its volume exceeds memory storage capabilities. This distinction underscores the necessity for adaptive approaches that accommodate the ongoing arrival of new information, thereby necessitating the exploration of the parameter space, its segmentation, and advanced techniques for interactive visual analysis in simulation environments.

Numerical simulations can produce extensive, multi-dimensional datasets, posing challenges for exploration. Innovative visualization approaches, like Bonneau et al. [7], are necessary to navigate and understand such complex data effectively. Typically, interactive visual analysis of simulation data employs customized views to illuminate various facets of multi-dimensional and spatio-temporal data as presented by Kehrer and Hauser [12]. However, not all visualization techniques, tools, and views are optimal for studying simulation data. Selecting suitable visualizations, tailored to the specific simulation domain and data characteristics, can benefit from a taxonomy of visualization solutions. Vernon-Bido et al. [26] outline such a taxonomy within the realm of modeling and simulation. In our case, we introduce the novel, interactive p-h view, integrating advanced visualization features while retaining the engineers' familiarity with their preferred view.

Pálenik et al. [20] introduce a mixed-initiative approach to visually guided modeling for atmospheric convection, integrating visual parameter space analysis with partial automatic parameter optimization. Although it relates to a different domain, the similarity of the IsoTrotter and our approach lies in depicting data in dual spaces. We use aggregated and computed data in addition to the p-h diagram, and they use the parameter space to constrain sampling.

Finally, steering of scientific simulations can be realized with a surrogate model as described by Bonneau et al. [7] or without a surrogate model as shown by Matković et al. [17]. Berger et al. [5] describe a system for conducting interactive, prediction-based local analysis of parameter spaces using methods from statistical learning. Splechtna et al. [24] use the output parameters of models of the same system at different complexity levels to map the different control parameters of these models and enable brushing and linking across models.

In our case we use a surrogate (DL) model to inverse the simulation, not to replace it, i.e., our approach is output parameter centric, in the sense, that in contrast to the aforementioned works, the start point or focal point [5] is not set in the control parameter space but in the output parameter space. In this way we can predict control parameters for the desired outputs of a given scenario, and steer the ensemble creation efficiently. Most approaches use similar visualization techniques, like coordinated multiple views with linking and brushing, and views, like scatter plots or parallel coordinates. We augment these techniques and views to highlight the data important for the current task at hand.

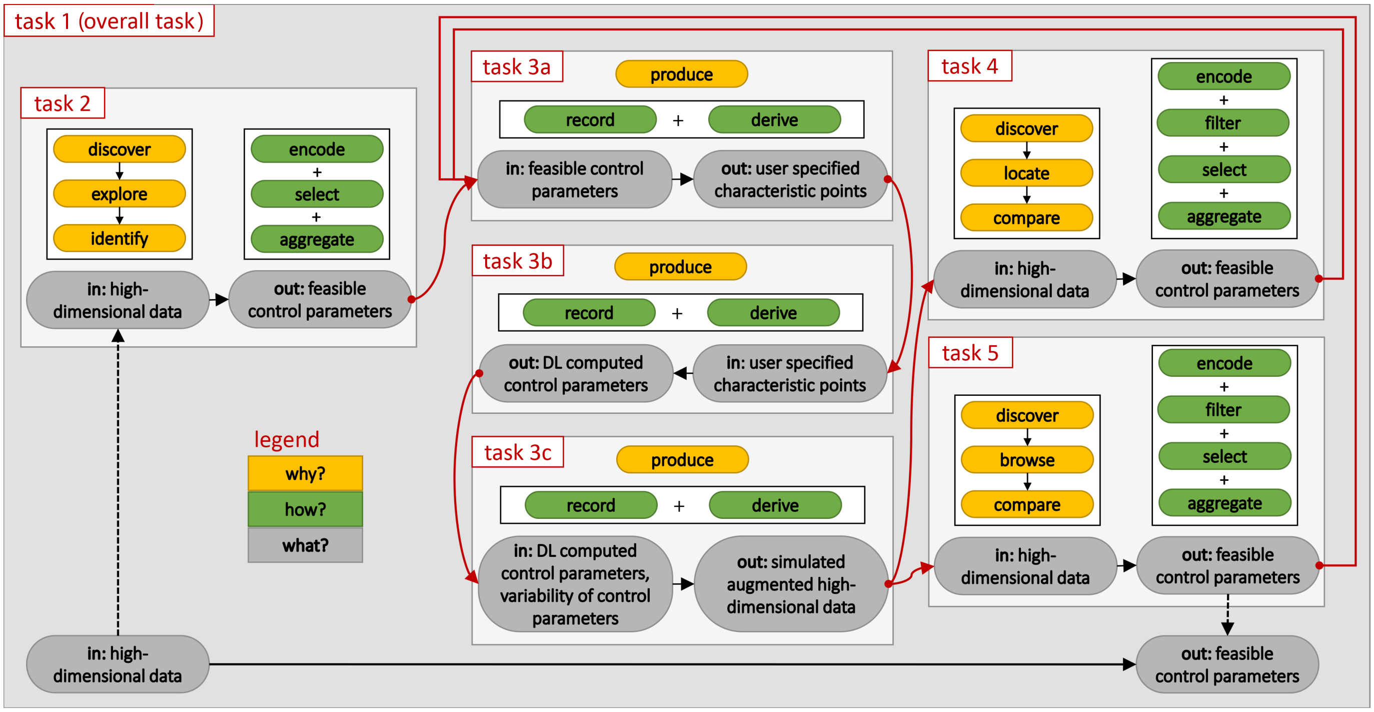

task 1 (overall task)
task 2
discover
explore
identify
encode
+
select
+
aggregate
in: high-dimensional data
out: feasible control parameters
task 3a
produce
record
+
derive
in: feasible control parameters
out: user specified characteristic points
task 3b
produce
record
+
derive
out: DL computed control parameters
in: user specified characteristic points
task 3c
produce
record
+
derive
in: DL computed control parameters, variability of control parameters
out: simulated augmented high-dimensional data
task 4
discover
locate
compare
encode
+
filter
+
select
+
aggregate
in: high-dimensional data
out: feasible control parameters
task 5
discover
browse
compare
encode
+
filter
+
select
+
aggregate
in: high-dimensional data
out: feasible control parameters
legend
why?
how?
what?
in: high-dimensional data
out: feasible control parameters

Fig. 4: Sequence of interdependent tasks in the IDoE workflow.

4 Tasks and Requirements Analysis

Designing an AC system, like many others, involves various constraints. Firstly, there are physical limitations: Depending on the selected refrigerant, only certain cooling cycles can be implemented. Additionally, manufacturing constraints, such as the maximum revolving speed of a compressor, come into play. If required component characteristics are unavailable due to manufacturing constraints, the system cannot be realized, neither. Furthermore, there are simulation constraints: The used simulation model usually only covers a subset of the actual physical system with sufficient accuracy.

The simulation model of the AC system is based on multiple control parameters that can be chosen by the user. These control parameters, together with other fixed values, determine the behavior of the system. The output parameters are then calculated by the simulation for the specified system. In the context of AC systems, a set of output parameters – called characteristic points – is of special interest since they are the essential indicators of the efficiency of the AC cycle (Figure 3).

The central challenge of cooling system design lies in identifying control parameters that yield a desired, optimized system behavior, with a special focus on the characteristic points, while also adhering to all other constraints. In principle, a system designer may wish to “go backwards”, deriving suitable control parameters from the specification of a certain, desired system behavior for a variety of scenarios. A scenario can be a specific operating condition, like "hot summer day", "Overheated cabin, cold motor start" or another design goal, like "Improve coefficient of performance". Each scenario is explored and optimized in an iterative fashion.

To inform our design, we engaged in a six-months collaboration of visualization and engineering experts. Given the established prevalence of the p-h diagram in cooling system design, we considered it necessary to base our visualization design on this standard. Based on numerous meetings and discussions, we introduced the IDoE approach described in section 1 and identified the following tasks. Following the approach by Brehmer and Munzner [6], we then analyzed these tasks in order to derive crucial requirements for our visualization design.

T1 Overall task: Comprehend and optimize the AC system, i.e., identify control parameters so that the system operates as efficient as possible for all relevant scenarios.

T2 Explore the p-h diagram, together with its dual representation, the derived values space, and the control parameters space, to maintain a holistic understanding of the system design.

T3 Iteration task: Identify necessary adaptations of the control parameters for an ensemble refinement based on desired cooling cycle characteristics for a specific scenario.

T3a Interactively specify desired cooling cycle characteristics.

T3b Estimate related control parameters through an attempted model inversion.

T3c Issue and integrate additional simulation results based on the inferred control parameters.

T4 Evaluate the ensemble refinements with respect to a specific scenario.

T5 Compare selected iterations of ensemble refinements with respect to all scenarios.

Analyzing the above outlined tasks and sub-tasks, we identified the following design requirements:

R1 Depict input and output spaces (control parameters and system outcomes) using linked views and support guided interaction.

R2 Use the standard p-h diagram, as known to the experts, to show and specify cooling cycle characteristics.

R3 Integrate a dual representation of characteristic points – keep the arrangement, but show additional, dual data.

R4 Based on the specified characteristic points, suggest related control parameters by attempting a model inversion in order to seed a new ensemble refinement iteration.

R5 Initiate a new ensemble refinement (additional simulation runs) in the vicinity of the suggested control parameters.

R6 Visualize all ensemble refinements and keep track of all iterations.

R7 Provide quantitative data for the iterations and for the individual runs to enable a comparison.

5 Interactive Design of Experiments

Our interactive design of experiments approach needs the following three main components to enable the engineer to perform the tasks as outlined in section 4, satisfying the requirements as imposed by the tasks: ***the simulation component***, which simulates the AC system; ***the deep learning component***, which attempts to invert the simulation model; and ***the interactive visualization component***, which serves as the central hub for visualization and interaction, as well as the main interface to all components. Figure 2 illustrates these three overall components and the data exchange between them. The IDoE workflow is based on an offline setup phase, followed by the iterative mixed-initiative optimization phase.

Setup phase (offline, not interactive, once):

1. Create an initial ensemble using the simulation model, providing a sparse, yet as-large-as-feasible sample of the parameter space, while considering manufacturing constraints.
2. Train a DL model to attempt an inversion of the simulation model, using the initial ensemble, suggesting likely responsible control parameters for a certain, desired system behavior.

Mixed-initiative optimization phase (interactive, iterative – see Figure 4 for our corresponding task analysis):

1. Visualize and explore the initial ensemble (T2).
2. Specify desired outputs, taking into account physical constraints; in our case, the user specifies four values to define the properties of a cooling cycle in the p-h diagram (T3a).
3. Estimate the control parameters for the specified outputs using the DL model (T3b).
4. Run the simulation for values at and near the estimated control parameters (T3c).
5. Compare the computed and predicted outputs, and sample the parameter space in the vicinity of the predicted parameters to assess the quality of the predicted solution (T4) and/or compare computed and predicted outputs of previous iterations (T5).
6. Iterate or conclude the exploration and export selected cases (T1).

5.1 The Interactive Visualization Component

The data we are working with exhibits a non-trivial structure. For a set of scalar control parameters, the simulation solver computes four points in the p-h diagram, along with several other scalar outputs. These four points are interconnected, forming a cycle within the diagram. Additionally, there are supplementary values, such as the coefficient of performance, which can be derived from the four points. These values offer a dual perspective on the cycle and represent a valuable source of information for engineers.

Given the complexity of both the data and the requirements, we have chosen to utilize the well-established Coordinated Multiple Views (CMV) paradigm (R1). This approach has demonstrated effectiveness in ensemble simulation analysis (as well as in many other applications). Attempting to address all requirements within a single integrated view would result in excessive complexity and confusion.

Our design revolves around the p-h diagram. This diagram holds paramount importance in AC system design, making its omission unacceptable. We have enhanced its usability by incorporating interactive features for ease of analysis. In order to display additional data, we augment it with four scatterplots. Each of them is strategically placed to convey information related to a certain point of the cycle. We also utilize parallel coordinates for visualizing control parameters and box plots for presenting quantitative data and facilitating comparisons. Special emphasis has been placed on interaction.

5.1.1 Interactive p-h Diagram

The p-h diagram illustrates the characteristic points of the AC cycle through an orthogonal plot, with the logarithm of pressure (P) represented on the y-axis and specific enthalpy values on the x-axis (linear scale). Alongside, it displays the saturation curve, iso-quality lines, iso-thermal lines, and iso-entropy lines. We utilize the p-h diagram to represent characteristic points of the initial simulation ensemble, specify desired cooling cycle characteristics, and depict both specified/calculated and simulated characteristic points (R2). Optionally, connecting quadrilaterals for these points, i.e., the cooling cycle, may also be included, as well as the convex hull of the initial points, which indicates the area resulting from plausible parameters.

While this diagram conveys a wealth of information, it risks becoming cluttered when all elements are displayed simultaneously. Clearly, there is a need for an efficient method to filter out data that is not currently the focus. For example when specifying the desired cooling cycle characteristics, the details of the initial points are not of interest but the convex hull of the initial points is, since it gives a good indication if the currently specified cooling cycle characteristics, that are represented by the four characteristic points, can be achieved within the system's plausible parameter range (see Figure 6a).

In the following, we emphasize the most noteworthy design considerations incorporated during the creation of the interactive p-h diagram.

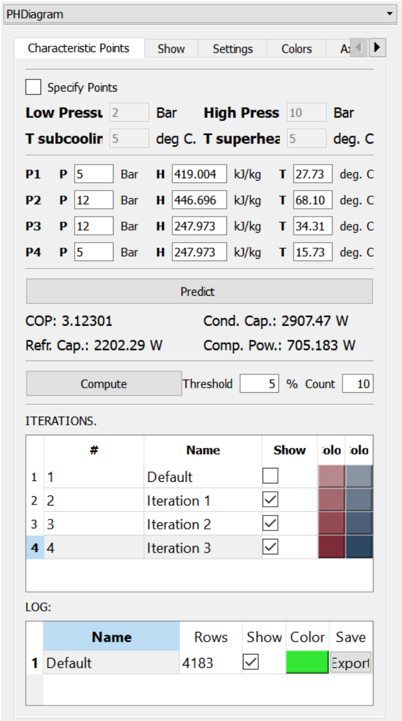
Guided Points Specification Since the four characteristic points of a cooling cycle are interdependent, they are typically indirectly specified by specifying four characteristic values of the desired cooling cycle, two pressures and two temperatures, as described in section 2.

Point 1 (refer, again, to Figure 3) is determined by the low pressure and the superheating temperature. Initially, point $1'$ is located (on the saturation curve at the liquid side for the given pressure), and the corresponding temperature is identified from the iso-thermal curves (red lines in Figure 3). Subsequently, the specified amount of superheating temperature is added, and the position of the resulting temperature with the specified pressure constitutes point 1. Interpolating from the iso-curves is not straightforward due to their non-linear gradient. We utilize the CoolProp library [4] to obtain the desired points.

The cycle then follows the iso-entropy line (ideally) until it reaches the high pressure. In practice, we account for the compressor efficiency and adjust point 2 accordingly. Subsequently, the cycle proceeds towards point $3'$, following the high pressure, and point 3 is computed similarly to the computation of point 1. Finally, point 4 shares the same enthalpy value as point 3 and has the low pressure. The user also has the option to move the points using the mouse, and all of the aforementioned constraints will be enforced. Without automatic guidance, specifying point coordinates directly for the desired cooling cycle would be rather impractical. During our exploration sessions, engineers consistently entered numeric values for the four parameters, finding it more efficient than using mouse selections.

The GUI elements for point specification are depicted in Figure 5. Once the four parameters are entered, pressure, enthalpy, and temperature are computed for each point. When a prediction by DL model is requested (R4), a new iteration is created. The corresponding control parameters are predicted along with the four values crucial for assessing the quality of the selected points: COP (coefficient of performance), condenser and refrigerant capacity, as well as compressor power. These four values are displayed in the GUI controls (Figure 5). Refinements for an iteration can also be triggered from the same control (R5).

Providing Dual Perspective The p-h diagram explicitly depicts pressure and specific enthalpy, while temperature, quality, and entropy for each characteristic point can be inferred from nonlinear iso-lines. These values are utilized to compute various aggregated data. However, interpreting values based on these non-linear iso-lines is challenging. To simplify analysis, we provide a dual representation of the p-h diagram using four scatterplots (R3). Each scatterplot shows additional aggregated values closely related to its characteristic point, reflecting its position in the cycle. To aid association with the points, the scatterplots mirror the arrangement of the points themselves. We experimented with two alignment strategies: grouping the scatterplots together on one side of the p-h diagram or distributing them across two opposite sides (left-right or top-bottom) to align with the original points. The latter arrangement, with two plots on each side, was unanimously preferred by domain experts. Figure 1 illustrates the favored arrangement. The



□ Specify Points

Low Pressu	2	Bar	**High Press**	10	Bar
T subcoolir	5	deg C.	**T superhea**	5	deg. C

P1	**P**	5	Bar	**H**	419.004	kJ/kg	**T**	27.73	deg. C
P2	**P**	12	Bar	**H**	446.696	kJ/kg	**T**	68.10	deg. C
P3	**P**	12	Bar	**H**	247.973	kJ/kg	**T**	34.31	deg. C
P4	**P**	5	Bar	**H**	247.973	kJ/kg	**T**	15.73	deg. C

COP: 3.12301 Cond. Cap.: 2907.47 W
Refr. Cap.: 2202.29 W Comp. Pow.: 705.183 W

Threshold 5 % Count 10

ITERATIONS.

	#	Name	Show	olo	olo
1	1	Default	□		
2	2	Iteration 1	☑		
3	3	Iteration 2	☑		
4	4	Iteration 3	☑		

LOG:

	Name	Rows	Show	Color	Save
1	Default	4183	☑	🟩	Export

Fig. 5: The control used to specify desired values (four values in the top), to provide feedback on aggregated data (the second block of values), to trigger ensemble refinements, and to keep track of iterations and findings.

points in the scatterplots are colored using the same colors as those in the p-h diagram. Additionally, to aid in identifying corresponding cases, predicted and computed cases for the same parameter setting are connected by a line, assisting in relating them to the corresponding points in the p-h diagram.

5.1.2 Additional Views and Integrated System

Besides the interactive p-h diagram, we have integrated additional views into a coordinated multiple view setup: the parallel coordinates view, the box plots view, and the details view. The details view presents detailed information for a selected subset in a tabular format. This view can be used to inspect individual runs after a drill-down or to export the values for use in other applications. The additional views provide information on control parameters, as well as different perspectives and statistics on some output values (R1). Composite brushing, which combines individual brushes across various views using Boolean operators, is supported. The chosen color maps are used consistently across all views. Also the visibility state of an iteration, chosen in the p-h diagram, is propagated to all other views, ensuring a consistent perspective on the data. This makes it easy to link cases and iterations across different views (R6).

Parallel Coordinates We utilize parallel coordinates to depict the simulation control parameters (Figure 1 top right). They provide a proven method for displaying a multidimensional parameter space, and our domain expert is familiar with them. Therefore, they were the natural choice for visualizing the five control parameters. Still, some adaptations were needed. We use the parallel coordinates to show the initial ensemble, the values predicted by the DL model, and the refinements by the simulation solver. Each of these lines have different settings in order to be easily distinguished. In addition, the rendering order is fixed, so that the initial runs are rendered first, the refinements then, and the predicted values on top.

Box Plots Seeing all computed values in scatter plots provides a good overview of distributions but lacks additional quantitative details of the output parameters (R7). To quickly compare different iterations, the box plots depict basic statistics for each one (Figure 1 bottom right). This provides an informative overview of the development of basic descriptive statistical values across iterations. Besides the statistics for each iteration, the box plots also show the overall statistics of the initial ensemble, which serves as context. Again, the rendering order ensures maximum visibility.

5.1.3 Visual Settings.

We follow an iterative workflow, resulting in different outcomes for each iteration. Alongside the initially computed points, we have predicted parameters and computed cases based on these predictions, as well as several refinements at each iteration. To visualize these variations, we utilize point size and color, distinguishing between different cases and iterations (R6).

Given that analysis sessions typically involve a limited number of iterations, we opted for a sequential color map with seven levels for both predicted and computed cases (one map for each). The darkest color represents the most recent iteration, while progressively lighter shades indicate earlier iterations. When there are only a few iterations, we avoid using neighboring color map levels to enhance the distinction between iterations. If more than seven iterations are present, older ones share the same color. Figure 6d displays two color maps, red and blue, alongside colors for three and five iterations.

To facilitate the association of predicted and simulated values, we connect corresponding points with a line. Furthermore, additional cases computed by the simulation solver are depicted using the simulation color map, albeit with a smaller point size to differentiate them from cases computed using parameters predicted by the DL model.

Figure 6b showcases a detail of the p-h diagram, depicting initial points (gray), predicted points (red), and simulated points (blue) across two iterations. The convex hull of the initial points (gray), along with iso-thermal lines (blue) and iso-entropy lines (orange), are also depicted. Additionally, gray quadrilaterals represent individual cooling cycles by connecting characteristic points of a single run. Although the display is cluttered with all this information, each layer is essential at specific workflow stages. Figure 6a shows the configuration used for specifying desired values, while Figure 6c shows a configuration when an engineer examines two iterations and focuses on the characteristic points only. The reference curve (green) and the convex hulls of initial point values corresponding to plausible parameters are included for context.

5.1.4 Iterations and Findings Bookkeeping

To facilitate an overview of iterations and manage them effectively, we provide a GUI element as depicted in the lower part of the Figure 5. For each iteration, it offers the option to change the name, displays information on the colors used for predicted and simulated cases, and includes a checkbox to toggle the visibility of that iteration (R6).

In line with the strategy proposed by Yang et al. [29] and in response to users' need to keep track of significant findings, we also offer a logging feature for certain cases. When users isolate a single case or a small set of related cases using interactive drill-down, they can be stored in a log of findings. Users can customize the name, toggle the visibility, adjust the color, or export cases of a finding to a CSV file for reporting or further analysis in a different tool.

5.2 The Deep Learning Component

The DL component is used to estimate an inverse simulation model based on the available simulation runs. In reverse to the simulation model, the DL model expects four characteristic points as input, and provides the corresponding control parameters, five in our case, for the simulation model as outputs (R4).

In the Batch Learning Process of our study, initially, simulation data generated from the AC system runs were collected and compiled into a comprehensive dataset. This dataset was crucial for the subsequent batch learning phase, where it was processed by a specifically designed DL module dedicated to batch processing. Embracing a traditional yet strategic approach, the DL model, identified to tackle the inherent non-linear regression problem of our study, was initially approached with simplicity, setting a baseline for further enhancement.

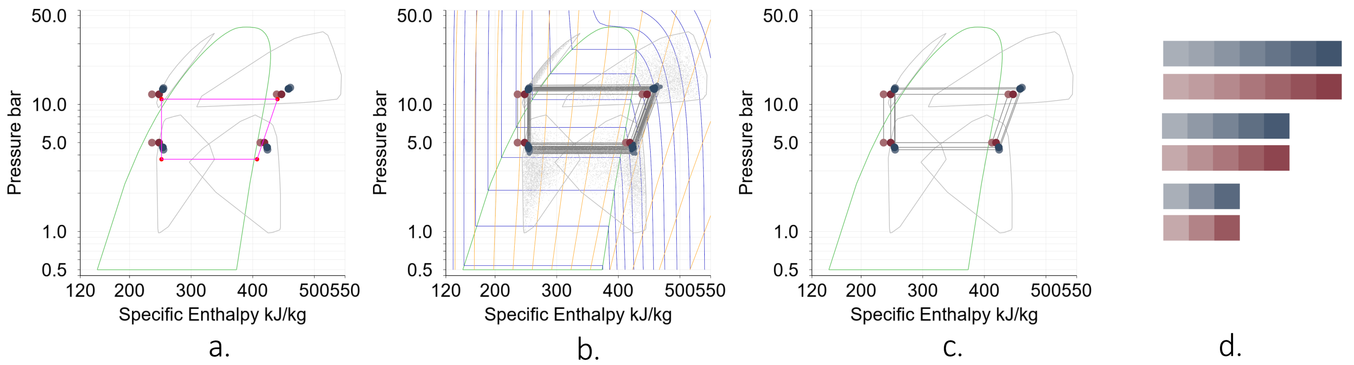
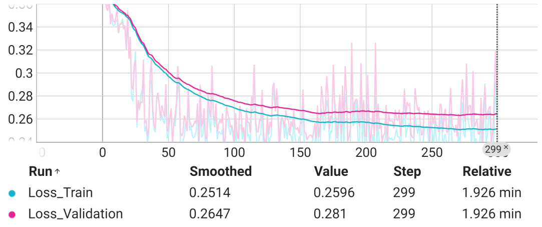


Fig. 6: Interactive p-h diagram showing different information at different workflow stages. **a.** When specifying desired points (bright red), the reference curve (green), convex hulls of the initial points (gray), and previous iterations of interest are shown (red and blue). **b.** A multitude of information can be shown: quadrilaterals of cooling cycles for simulation runs (gray), initial points (gray), iso-thermal lines (blue), iso-enthorpic lines (orange). Showing all of the layers at once results in a cluttered view. **c.** The engineer focuses on two iterations and hides many layers. Convex hulls and the reference curve provide context. **d.** We use seven-level color maps. The current iteration is mapped consistently to the darkest shade. The step in the color map is adjusted based on the number of iterations to enhance the contrast between iterations.



Run ↑	Smoothed	Value	Step	Relative
● Loss_Train	0.2514	0.2596	299	1.926 min
● Loss_Validation	0.2647	0.281	299	1.926 min

Fig. 7: Train and validation loss.

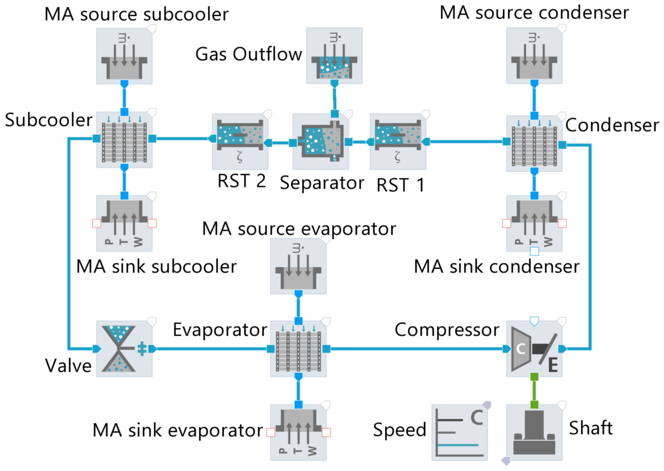


Fig. 8: The simulation model of an AC system.

We implemented a deep neural network comprised entirely of fully connected layers. Our analysis revealed that increasing model depth beyond three hidden layers did not enhance accuracy but instead increased complexity, parameters, and training time. Therefore, we opted for a model architecture featuring three hidden layers, each employing a tangent hyperbolic (Tanh) activation function, except for the output layer. Hyperparameter optimization was conducted using an exhaustive grid and basin search methodology, ensuring the model's robustness and efficiency.

The training of this model was a meticulous process, spanning 300 epochs, employing the Mean Squared Error Loss function, and leveraging the AdamW optimizer. This was complemented by precise adjustments to learning rate and weight decay, reflecting our commitment to refined model tuning. The fruit of this labor was evident in the model's performance, which registered an impressive average Mean Squared Error (MSE) of 0.25 on the test dataset—a testament to the model's robustness and the efficacy of our approach.

The dataset, an aggregation from 4182 simulation runs, was strategically partitioned, allocating 80% for training, and 10% for validation and 10% for testing. The structure enabled batch learning, updating model weights iteratively after processing each batch. This approach ensured continuous refinement across epochs and consistent evaluation, validating effective learning. This model architecture was supported by empirical findings detailed in Figure 7 and showcasing comparisons of training/validation loss.

Our study employs an Online Learning Enhancement strategy, enabling continuous model adaptation through incremental learning. This dynamic interaction ensures the model's relevance and accuracy over time, reflecting the evolving conditions of the simulated AC system (see workflow in Figure 2).

Further, our Iterative Feedback and Model Updating process establishes a robust feedback loop. Here, predictions from the deep learning component guide control parameters in the simulator, fostering a cycle of prediction, simulation, and data re-acquisition. This iterative approach, integrating batch and online learning methodologies, continuously refines the 'Updated Model', enhancing its precision and adaptability to real-world data dynamics.

Overall, our methodology harmonizes simulation and learning components, leveraging historical data for foundational learning while adapting to new insights. This integrated approach ensures the model's accuracy and flexibility in addressing the complex and dynamic demands of AC system simulation.

5.3 The Simulation Component

We use the multidisciplinary vehicle system simulation tool AVL CRUISE™M, which is commercially available and can simulate AC systems. The user defines the simulation model using basic building blocks (Figure 8), sets the simulation control parameters, and triggers the numerical simulation. The simulation computes the four characteristic points in the p-h diagram, and a lot of additional output values.

In our model the condenser, subcooler and evaporator are vapor, liquid and equilibrium (VLE) multi port extruded tubes (MPET) heat exchanger components. The MPET heat exchanger is a crucial component of a VLE circuit, used to simulate the heat transfer between moist air and a VLE fluid as well as the pressure-drop phenomena through the heat exchanger of a user-defined geometry. The simulation solver for such complex thermodynamic cycles are characterized by their non-linear behavior. The physical system models are often solved numerically in form of an ordinary differential equation system or differential algebraic equation system.

It is impossible to mathematically invert such a simulation model, i.e., it is not possible to specify the four desired characteristic points and to get the corresponding simulation control parameters. Due to the complexities of some elements in the model the computation time for a single simulation run is approximately ten seconds. Hence the computation for our initial ensemble consisting of 5000 runs took about 14 hours. Since the solver did not produce valid results for all runs we end up with 4182 runs in the initial ensemble.

6 Use Case

In our use case, we begin with a fine-tuned simulation model that we use to explore diverse scenarios and observe the system's behavior under various conditions. This yields valuable insights into the system's performance and highlights potential areas for improvement. However, it is important to note that interpreting the results of simulation models is not a straightforward task. The behavior of the AC system is influenced by a multitude of factors, and the interactions between these factors can be highly complex. The engineers must have a deep understanding of the system's behavior in order to accurately interpret the results of simulations.

We are dealing with an AC system designed to cool the cabin of an electric vehicle, where the cabin temperature is set to fluctuate between 20 and 50 degrees Celsius, with an external temperature of 30 degrees Celsius, representative of a typical summer day.

Our objective is to delve into the system and analyze the values requiring control parameter adjustments to achieve the desired functionality. The challenge involves pushing the system beyond current boundaries, exploring the potential for expansion while facilitating straightforward upgrades to avoid extensive interventions.

We seek parameter settings that maintain stability within the parameter space. While a highly efficient but overly sensitive solution is undesirable due to its susceptibility to minor parameter changes, it's crucial to ensure that the parameters remain feasible and adaptable to real-world fluctuations during system operation. We use the simulation model as shown in Figure 8 and we vary five parameters to create the initial ensemble. We set the physically plausible boundaries and use Latin Hypercube Sampling [18] to create individual cases. The five varied control parameters are:

1. N_pump – speed of the compressor pump.
2. MF_air_cond – mass flow of ambient air through which heat is expelled within the condenser.
3. T_air_cabin – temperature of cabin air to be cooled in evaporator.
4. MF_air_evap – mass flow of cabin air cooled in the evaporator.
5. A_eff_valve – effective flow area of the valve, used to calculate the actual flow rate of the valve orifice.

In addition to the four characteristic points, the simulation solver also computes the following output values relevant for the analysis:

1. COP – coefficient of performance, indicates how effective an AC is at transferring heat vs. power consumed by the compressor.
2. $T_{subcooling}$ – subcooling temperature (see section 2).
3. $T_{superheating}$ – superheating temperature (see section 2).
4. dh_e – condenser capacity, the ability of the condenser to transfer heat from the hot vapor refrigerant to the condensing medium.
5. dh_c – refrigerant capacity, the ability of the evaporator to transfer heat from the hot outside media to the refrigerant.
6. W – electrical power consumption of the compressor.

A previous exploration of a scenario confirmed that the model functions properly for the following values of pressures and temperatures: $P_{low} = 2.5\ bar$; $P_{high} = 12\ bar$; $T_{subcooling} = 7°C$, $T_{superheating} = 8°C$. In the current scenario, which we describe now, the goal is then to increase the coefficient of performance (COP).

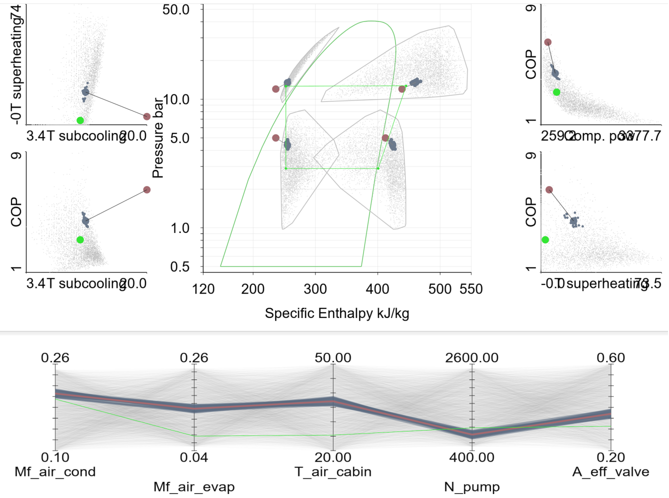


Fig. 9: First iteration: The user specified values are shown in red. They lie outside the plausible range. The DL suggests parameters which result in the blue points when simulated. They do not match the desired target, but they are plausible. The light green point in the scatterplots shows the starting scenario. We achieved the goal of increasing COP.

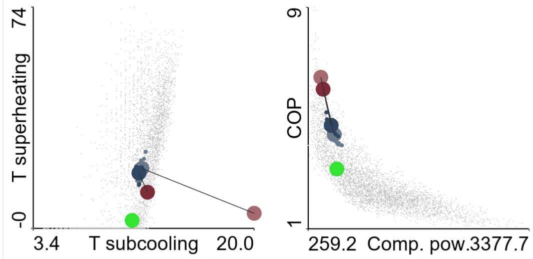


Fig. 10: **Left:** Temperatures for the predicted (red) and simulated values (blue). The first desired subcooling temperature (light red point) was set unrealistically high, and the DL model pushed it back into plausible range. **Right:** The COP values for the simulated refinements are significantly higher than for the starting scenario (light green point).

Iteration 1. The desired value for P_{low} is increased to $5\ bar$, while P_{high} remains at $12\ bar$. To enhance the refrigerant capacity, the subcooling temperature is elevated to $20°C$, and we set the superheating temperature to $5°C$. We do not lower it below $5°C$ to mitigate the risk of harmful compressor temperatures.

A prediction of the control parameters is requested from the DL model, and the results are immediately available. Subsequently, we commence the computation of twenty runs with the simulation solver, varying parameters by up to 5% relative to the predicted values. The results are explored in the visualization.

We use the three main views, the augmented p-h diagram, parallel coordinates, and boxplots, as shown in Figure 1. The scatterplots around the p-h diagram show data relevant for the use case (note that COP is shown in several plots). The parallel coordinates show the control parameters, and the box plots show the three aggregated values which indicate how much energy flows through the system. It is necessary to immediately see if the energy flow is too high.

The parallel coordinates in the Figure 9 show that the control parameters for the first iteration fall quite central within the original ranges (the DL model does not generate solutions outside plausible boundaries). In the p-h diagram we can see that requested point 3 (red point) is quite far from the plausible area (gray convex hull). The simulated refinements (blue points), however, fall within the original available area. The connected line between desired and simulated point indicates this difference. We proceed to examine other characteristics of the simulated cases, particularly focusing on COP and $T_{superheating}$. Notably, the desired $T_{superheating}$ lies significantly outside the plausible area. However, the simulated cases maintain $T_{superheating}$ within the acceptable range, albeit with several different values observed. Thus, we anticipate finding a suitable solution.

The COP values are satisfactory, prompting us to proceed to the next iteration. We show the starting scenario using bright green color. Computed cases have a higher COP values. We opt to reduce $T_{subcooling}$ in the subsequent iteration, aiming to achieve an optimal solution.

Iteration 2. We adjust $T_{subcooling}$ to align it closer to real values, setting it to $12°C$. The results indicate that we remain within a similar range as in the initial iteration, with $T_{subcooling}$ values around $11°C$. Consequently, we decide to retain this value and proceed to explore other parameters. The scatterplot in Figure 10 left shows the wanted values of the temperatures in red for the first two iterations, and corresponding simulated values in blue. The first subcooling temperature was excessively high, rendering it unattainable through simulation. The second desired value proved to be more realistic, although it also required correction. The superheating temperature was set closer to the computed values, the shift in the y-direction is much smaller.

Since we aim to maintain consistent pressures (which correspond to ambient temperatures), we refrain from altering them. Instead, we focus on increasing $T_{superheating}$ to observe its effects. However, caution is warranted to avoid excessively high values or those below $0°C$, as such extremes could potentially damage the compressor.

Iteration 3. We adjust $T_{superheating}$ to $12°C$ and repeat the procedure. After computing the simulation runs, we obtain a potential configuration for our simulation model. The coefficient of performance (COP) shows a reasonable increase compared to the original scenario (Figure 10 right). Additional control over cabin temperature indicates that the system can maintain the desired characteristics even with temperature variations. Cases in the vicinity of the selected case exhibit similar temperatures (see, for example, the parallel coordinates in Figure 1). Ensuring close proximity of cabin temperatures avoids the need for large adjustments in the system settings for minor temperature changes.

This brief use case demonstrates our ability to significantly enhance the COP in just three iterations. We swiftly corrected the initial setting and pinpointed the promising area for improvement. The fine-tuning process, coupled with sensitivity analysis of the chosen solution, proved highly efficient. Achieving such an increase in COP through conventional methods would typically entail extensive trial and error, resulting in a more laborious workflow. The exploration and tuning continues by examining further scenarios in the same manner.

7 Discussion

Designing a visual analysis system for domain experts poses significant challenges to both the visualization designers and the domain experts. The IDoE approach emerged from a participatory design and development process, which was lengthy and involved numerous discussions, clarifications, and iterations of the design, including various color scales, view configurations, and more. Reflecting on the process, we cannot overstate the importance of repeatedly discussing visual analytics principles and learning the basic principles of the domain.

In our case, we were fortunate that the domain expert already had experience with interactive visual analysis. On the other hand, we had to learn extensively about AC systems. Familiarity with the domain is crucial for success. It became evident to us that the p-h diagram had to be the central view. Given our five-dimensional parameter space, we suggested using parallel coordinates. Interestingly, with visualization tools being more widely used in the past decade or two, there are more experts familiar with parallel coordinates. Our main collaborator was aware of them, and his colleagues, to whom we presented the system, quickly understood them. Despite some drawbacks, we could not find a better way to depict the 5D parameter space.

The decision to use a consistent color scale across all views proved to be correct and very important. We experimented with many color scales and, following also the reviewers' suggestions, finally selected red and blue colors.

The time of domain experts is usually very valuable as they are engaged in various projects within their companies. Finding an expert willing to engage in developing a new approach was crucial. To maintain the expert's enthusiasm and prevent them from leaving the project early, the proposed system had to offer significant advantages compared to the current workflow. After initial discussions with the domain expert, it was essential to present a prototype that captured the expert's attention and clearly demonstrated its usefulness. Of course, the first prototype was far from a fully operational system, but it had to clearly show its potential benefits.

General design guidelines suggest not offering too much flexibility. John Maeda [15], for example, explores the concept of simplicity in design and how it can lead to better user experiences. One of the laws discussed is "Reduce", which suggests that removing unnecessary complexity can improve usability. At the same time, Jenny Preece et al. [21], e.g., discuss the importance of flexibility in interfaces to support diverse user needs and workflows, especially for expert users. Since we cater to experts, and different requirements may arise depending on the current workflow state, we offer a highly flexible system where individual layers can be easily toggled on or off. We organize all customization options into different semantic tabs (e.g., Show tab, Characteristic Points tab, etc.) to facilitate quick access to individual settings. After a brief customization period, experts were satisfied with the organization, finding it easy to locate the required settings promptly.

When reflecting back on the feedback of the domain users, one of the most appreciated aspects of the method is the tight integration of all three components: deep learning, simulation, and visualization. Tuning an AC system, as described in the paper, would otherwise necessitate multiple separate tools and cumbersome data exchange between them. The integration of various aspects within a single workflow, along with diagrams, that are familiar to the domain experts and are augmented for enhanced understanding, notably boosts the efficiency of exploration and prediction processes, as reported by the domain experts. They further attested that the IDoE approach accelerates their cooling system design tasks by at least an order of magnitude. An analysis that took several days can now be accomplished in few hours, at most.

The IDoE approach, designed for AC Systems, can also be applied to other problems where inverting a simulation model is not possible. Limitations include the number of control parameters and the speed of simulation. Using parallel coordinates, the practical limit is about a dozen control parameters. Simulations need to be relatively fast to generate refinements; our engineers were willing to wait a few minutes for a refinement with twenty runs. Longer waits may be impractical, though parallelization can help. The definition of "quickly" varies by domain, as some users are accustomed to longer waits.

8 Conclusion

Whenever a simulation model cannot be inverted or inversion proves overly complex, interactive design of experiments (IDoE) emerges as a potent approach for exploration and analysis. This method ensures that additional simulation runs are strategically computed in areas of the control parameter space likely to yield desired outputs. The general concept is applicable across various domains.

We acknowledge the complexity of the user interface and that its usability could undoubtedly be improved. However, mechanical engineers who utilized the system quickly became acquainted with it and utilized it seamlessly. Their familiarity with complex simulation software and their expectation of a learning curve likely contributed to its favorable reception.

Looking ahead, future research should capitalize on advancements such as deep learning to further expedite simulation solver speed and augment the tool's capabilities. This includes incorporating additional constraints in the specification of the design goals in the DL prediction, while also expanding the interface support accordingly, enabling the exploration of scenarios outside of the default control parameters space, and implementing an adaptive interface that flexibly adjusts to the user's needs. By continuously refining and expanding upon these advancements, we can empower experts to navigate the complexities of diverse engineering systems with enhanced efficacy and precision.

Acknowledgments

The authors would like to thank Damjan Rafaj for the first version of the DL model. VRVis is funded by BMK, BMDW, Styria, SFG, Tyrol, and Vienna Business Agency in the scope of COMET – Competence Centers for Excellent Technologies (879730), which is managed by FFG. Parts of this work have been done in the context of CEDAS, i.e., the Center for Data Science at the University of Bergen, Norway.

References

[1] A. G. Abdul Jameel, A. Al-Muslem, N. Ahmad, A. B. S. Alquaity, U. Zahid, and U. Ahmed. Predicting enthalpy of combustion using machine learning. *Processes*, 10(11):2384:1–2384:18, 2022. doi: 10.3390/pr10112384 3

[2] F. A. Aly and L. L. Lee. Self-consistent equations for calculating the ideal gas specific heat capacity, enthalpy, and entropy. *Fluid Phase Equilibria*, 6(3–4):169–179, 1981. doi: 10.1016/0378-3812(81)85002-9 3

[3] A. Atanasov, H.-J. Bungartz, J. Frisch, M. Mehl, R.-P. Mundani, E. Rank, and C. van Treeck. Computational steering of complex flow simulations. In S. Wagner, M. Steinmetz, A. Bode, and M. M. Müller, eds., *High Performance Computing in Science and Engineering, Garching/Munich 2009*, pp. 63–74. Springer, Berlin, Heidelberg, 2010. doi: 10.1007/978-3-642-13872-0_6 3

[4] I. H. Bell, J. Wronski, S. Quoilin, and V. Lemort. Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library coolprop. *Industrial & Engineering Chemistry Research*, 53(6):2498–2508, 2014. doi: 10.1021/ie4033999 3, 5

[5] W. Berger, H. Piringer, P. Filzmoser, and E. Gröller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Computer Graphics Forum*, 30(3):911–920, 2011. doi: 10.1111/j.1467-8659.2011.01940.x 3

[6] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, 2013. doi: 10.1109/TVCG.2013.124 4

[7] D. Butnaru, G. Buse, and D. Pflüger. A parallel and distributed surrogate model implementation for computational steering. In *Proceedings of the 2012 11th International Symposium on Parallel and Distributed Computing*, pp. 203–210. ACM, New York, 2012. doi: 10.1109/ISPDC.2012.35 3

[8] B. Chen, Z. Cai, and M. Bergés. Gnu-RL: A precocial reinforcement learning solution for building HVAC control using a differentiable mpc policy. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pp. 316–325. ACM, New York, 2019. doi: 10.1145/3360322.3360849 3

[9] Y. Chervonyi, P. Dutta, P. Trochim, O. Voicu, C. Paduraru, C. Qian, E. Karagozler, J. Q. Davis, R. Chippendale, G. Bajaj, S. Witherspoon, and J. Luo. Semi-analytical industrial cooling system model for reinforcement learning. *arXiv:2207.13131 [cs.AI]*, 2022. doi: 10.48550/arXiv.2207.13131 3

[10] Danfoss. Coolselector 2. `https://www.danfoss.com/en/service-and-support/downloads/dcs/coolselector-2/`. Last accessed 1 July 2024. 3

[11] D. L. Folmsbee, D. R. Koes, and G. R. Hutchison. Evaluation of thermochemical machine learning for potential energy curves and geometry optimization. *The Journal of Physical Chemistry A*, 125(9):1987–1993, 2021. PMID: 33630611. doi: 10.1021/acs.jpca.0c10147 3

[12] J. Kehrer and H. Hauser. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):495–513, 2013. doi: 10.1109/TVCG.2012.110 3

[13] M. Khennich, M. Sorin, and N. Galanis. Equivalent temperature-enthalpy diagram for the study of ejector refrigeration systems. *Entropy*, 16(5):2669–2685, 2014. doi: 10.3390/e16052669 3

[14] J. Luo, C. Paduraru, O. Voicu, Y. Chervonyi, S. Munns, J. Li, C. Qian, P. Dutta, J. Q. Davis, N. Wu, X. Yang, C.-M. Chang, T. Li, R. Rose, M. Fan, H. Nakhost, T. Liu, B. Kirkman, F. Altamura, L. Cline, P. Tonker, J. Gouker, D. Uden, W. B. Bryan, J. Law, D. Fatiha, N. Satra, J. Rothenberg, M. Waraich, M. Carlin, S. Tallapaka, S. Witherspoon, D. Parish, P. Dolan, C. Zhao, and D. J. Mankowitz. Controlling commercial cooling systems using reinforcement learning. *arXiv:2211.07357 [cs.LG]*, 2022. doi: 10.48550/arXiv.2211.07357 3

[15] J. Maeda. *The Laws of Simplicity*. The MIT Press, Cambridge, MA, 2006. 9

[16] Maplesoft. Maple. `https://www.maplesoft.com`, 2024. Last accessed 1 July 2024. 3

[17] K. Matković, D. Gračanin, M. Jelović, and H. Hauser. Interactive visual steering — rapid visual prototyping of a common rail injection system. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1699–1706, 2008. doi: 10.1109/TVCG.2008.145 3

[18] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979. doi: 10.2307/1268522 8

[19] J. D. Mulder, J. J. van Wijk, and R. van Liere. A survey of computational steering environments. *Future Generation Computer Systems*, 15(1):119–129, 1999. doi: 10.1016/S0167-739X(98)00047-8 3

[20] J. Palenik, T. Spengler, and H. Hauser. Isotrotter: Visually guided empirical modelling of atmospheric convection. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):775–784, 2021. doi: 10.1109/TVCG.2020.3030389 3

[21] J. Preece, Y. Rogers, and H. Sharp. *Interaction Design: Beyond human-computer interaction*. John Wiley & Sons, Chichester, UK, 4th ed., 2015. 9

[22] D. Sahoo, Q. Pham, J. Lu, and S. Hoi. Online deep learning: Learning deep neural networks on the fly. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 2660–2666, 2018. doi: 10.24963/ijcai.2018/369 3

[23] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2161–2170, 2014. doi: 10.1109/TVCG.2014.2346321 3

[24] R. Splechtna, K. Matković, D. Gracanin, M. Jelovic, and H. Hauser. Interactive visual steering of hierarchical simulation ensembles. In *Proceedings of the 2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 89–96, 2015. doi: 10.1109/VAST.2015.7347635 3

[25] The MathWorks Inc. MATLAB version: 9.13.0 (r2022b). `https://www.mathworks.com`, 2022. Last accessed 1 July 2024. 3

[26] D. Vernon-Bido, A. Collins, and J. Sokolowski. Effective visualization in modeling & simulation. In *Proceedings of the 48th Annual Simulation Symposium*, pp. 33–40. Society for Computer Simulation International, San Diego, 2015. doi: doi/10.5555/2876341.2876346 3

[27] G. Waxenegger-Wilfing, K. Dresia, J. Deeken, and M. Oschwald. Machine learning methods for the design and operation of liquid rocket engines — research activities at the DLR institute of space propulsion. *arXiv:2102.07109 [cs.LG]*, 2021. doi: 10.48550/arXiv.2102.07109 3

[28] Wolfram Research, Inc. Mathematica, Version 13.2. `https://www.wolfram.com/mathematica`, 2024. Last accessed 1 July 2024. 3

[29] D. Yang, E. A. Rundensteiner, and M. O. Ward. Nugget discovery in visual exploration environments by query consolidation. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, pp. 603–612. ACM, New York, 2007. doi: 10.1145/1321440.1321525 6