Machine learning for optimal flow control in an axial compressor

M. A. Elhawary[1,a], Francesco Romanò[1,b], Jean-Christophe Loiseau[2,c], and Antoine Dazin[1,d]

[1] Univ. Lille, CNRS, ONERA, Arts et Métiers Institute of Technology, Centrale Lille, UMR 9014, LMFL - Laboratoire de Mécanique des Fluides de Lille - Kampé de Fériet, 59000 Lille, France
[2] Arts et Métiers Institute of Technology, CNAM, DynFluid, HESAM Université, 75013 Paris, France

Abstract Air jets for active flow control have proved effective in postponing the onset of stall phenomenon in axial compressors. In this paper, we use a combination of machine learning and genetic algorithm to explore the optimal parameters of air jets to control rotating stall in the axial compressor CME2. Three control parameters are investigated: the absolute injection angle, the number of injector pairs and the injection velocity. Given an experimental dataset, the influence of the air jet parameters on the surge margin improvement and power balance is modeled using two shallow neural networks. Parameters of the air jets are then optimized using a genetic algorithm for three rotational velocities, i.e., $\Omega = 3200\,\text{RPM}, 4500\,\text{RPM}$ and $6000\,\text{RPM}$. First, surge margin improvement and power balance are being maximized independently. Then, a bi-objective optimization problem is posed to explore the trade-off between the two competing objectives. Based on the Pareto front, results suggest that a globally optimal set of parameters is obtained for a velocity ratio (defined as the ratio of the injection velocity to the rotor tip speed) ranging from 1.1 to 1.6 and an injection angle attack varying from $1°$to$11°$. These outcomes point out a potential generalization of the control strategy applicable to other compressors.

1 Introduction

Axial compressors are standard components in numerous industrial applications such as power generation or aircraft engines. One long-term goal is to improve their performances while maintaining safe and reliable operating conditions. Rotating stall [1] is a key problem hindering the stability and the progress of modern axial compressors. When it occurs, flow instabilities are generated, leading to degraded performances. In the worst-case scenario, rotating stall can eventually cause machine failure. Aiming at delaying the onset of this phenomenon, a lot of efforts have thus been devoted to designing and implementing effective flow control strategies, including both passive [2] and active [3] methods. In particular, active flow control methods have the added advantage of being switched on only when control is needed, thus leading to an overall reduced energy consumption. Previous studies [3,4] have demonstrated the effectiveness of air jets in postponing the onset of rotating stall in axial compressors. Such a control strategy relies on the injection of high-momentum fluid near the tip leading edge of the rotor blades by a discrete set of air jets positioned circumferentially upstream of the compressor. Despite the effectiveness of air jets for active flow control, achieving efficient and reliable control strategies remains challenging. One reason for that is the lack of efficient and accurate predictive models for control design.

A standard approach is thus to use *surrogate models*. Built on data, these models can approximate efficiently the relationship between the input parameters and the output of a given process. Examples of surrogate models include the *response surface methodology* [5], *Kriging* (also known as *Gaussian process regression*) [6], or *artificial neural networks* [7]. Such surrogate models can be used to tackle local optimization using gradient-based techniques, or global optimization with the Nelder–Mead's simplex method [8] or genetic algorithms [9]. In turbomachinery applications, surrogate model applications include aerodynamic design optimization [7], flow modeling [10], performance predictions [11] and flow control [12], among others [13].

For flow control, data-driven techniques have recently gained a lot of attention [14,15]. Two prominent avenues are evolutionary algorithms [16–19] and deep

T.I. : Quantitative AI in Complex Fluids and Complex Flows: Challenges and Benchmarks. Guest editors: Luca Biferale, Michele Buzzicotti, Massimo Cencini

[a] e-mail: mohamed.elhawary@ensam.eu
[b] e-mail: francesco.romano@ensam.eu
[c] e-mail: jean-christophe.loiseau@ensam.eu
[d] e-mail: antoine.dazin@ensam.eu (corresponding author)

reinforcement learning (DRL) [20–25]. Deep reinforcement learning relies on *deep neural networks* to approximate the optimal state–action combination from repeated queries of the experimental or numerical environment. Genetic algorithms on the other hand borrow ideas from natural selection and genetic principles to evolve a population of possible control strategies. Such evolutionary algorithms have been used experimentally by [26] to optimize the control parameters of DBD actuators to control the flow separation downstream of a backward-facing step. Likewise, genetic algorithms have been coupled to artificial neural networks to optimize the parameters of a suction/blowing strategy to control the separation over SD7003 airfoil by [27]. Intending for machine learning methods that automatically learn from data and improve over experience without being explicitly programmed, from the authors' point of view, GAs do not belong to them because GAs do not automatically learn, but they rather sort out, combine and compare sub-optimal solutions. On the other hand, DRL falls well within the category of ML methods. For an extensive overview, interested readers are referred to [15,28] or the book on *Machine Learning Control* [29].

To the best of our knowledge, limited references exist in the literature highlighting the potential of such control design strategies to mitigate rotating stall in axial compressors. This is the problem we tackle in this work. Given an experimental dataset [4], we train shallow neural networks to capture the relationship between our control parameters and two performance indices, i.e., the *power balance* and the *surge margin improvement*. The control parameters include the rotor rotational velocity and three air jet-related parameters: the absolute injection angle, the number of injectors pairs and the absolute injection velocity. First, the control parameters are optimized to maximize each of the performance indices independently. Then, a bi-objective optimization problem is formulated to explore the trade-off between these competing objectives. In both cases, we use shallow neural networks to approximate the cost function and genetic algorithms to solve the optimization problem. The paper is organized as follows: Sect. 2 presents a brief overview of the CME2 axial compressor, along with some definitions and an overview of experimental dataset used in the rest of this work. Then, details about our surrogate modeling and optimization procedures are given in Sect. 3. Finally, the key results are discussed in Sect. 4, while Sect. 5 provides the conclusion and some perspectives to this work.

2 Experimental flow configuration and dataset

2.1 Experimental configuration and definitions

The experimental setup considered in this work is the CME2 workbench, a single-stage low-pressure axial compressor with 30 and 40 blades for the rotor and stator, respectively. The main characteristics of this experimental platform can be found in [30]. The actuators consist in 20 equally spaced air jets pairs, each with two injectors. These injectors are mounted circumferentially at 10 mm upstream of the leading edge of the rotor blades. Each injector pair can be supplied by an external power source [3], or from the high-pressure region of the compressor [31]. The control process can use either 10 or 20 pairs of injectors. By using 10 such pairs, the injectors can generate approximately the same injection velocity as when all 20 pairs are used, albeit at half the total injection flow rate. These actuators leverage the Coandă effect by injecting momentum near the tip clearance of the rotor blades as illustrated in Fig. 1. Previous studies [3,4] have demonstrated the effectiveness of air jets to postpone the onset of rotating stall in axial compressors. Following [4], we focus our attention on three control parameters: the number of injectors pairs N_{inj}, the injection angle α_{inj} and the injection velocity V_{inj}. Three operating conditions, characterized by the rotational velocity Ω, are moreover considered: $\Omega = 3200$ RPM, 4500 RPM and 6000 RPM. The range of these different parameters is summarized in Table 1.

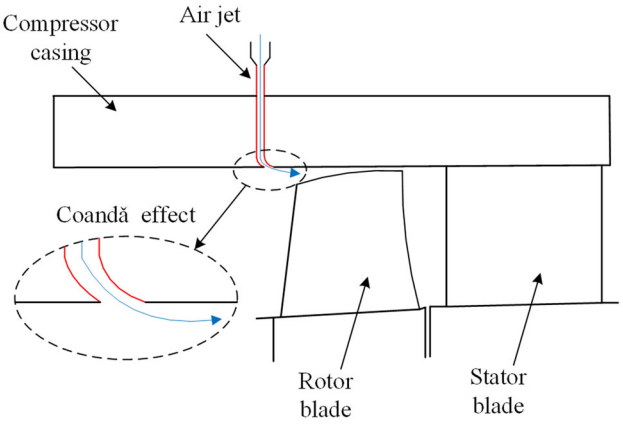


Fig. 1 Schematic view for air jet control system

Table 1 Range of experimental parameters considered

Rotor's rotational velocity Ω	3200 RPM, 4500RPM and 6000RPM
Number of injectors pairs N_{inj}	10 and 20
Absolute injection angle α_{inj}	$-45°, -30°, -15°, 0°, 15°, 30°$
Absolute injection velocity V_{inj}	46 m/s to 202 m/s

In turbomachinery applications, the effectiveness of a control strategy is often evaluated based on two performance indices: the surge margin improvement and the power balance. The surge margin improvement, hereafter denoted as SMI_{tot}, is defined as

$$SMI_{\text{tot}} = \frac{SM_{\text{c}} - SM_{\text{b}}}{SM_{\text{b}}} \times 100, \tag{1}$$

where SM_b and SM_c are the surge margins of the baseline and controlled configurations, respectively. These are given by

$$SM_{\mathrm{b}} = \left(\frac{p_{\mathrm{tt,b}}}{Q_{\mathrm{s,b}}} \times \frac{Q_{\mathrm{nom}}}{p_{\mathrm{nom}}} - 1 \right) \times 100,$$

and

$$SM_{\mathrm{c}} = \left(\frac{p_{\mathrm{tt,c}}}{Q_{\mathrm{s,c}}} \times \frac{Q_{\mathrm{nom}}}{p_{\mathrm{nom}}} - 1 \right) \times 100.$$

where $Q_{s,b}$ and $p_{\mathrm{tt,b}}$ are the flow rate and total-to-total pressure in the baseline configuration, while $Q_{s,c}$ and $p_{\mathrm{tt,c}}$ are the flow rate and total-to-total pressure in the controlled setup. Finally, Q_{nom} and P_{nom} are the nominal flow rate and power of the compressor. For a rotational velocity $\Omega = 3200RPM$, the nominal power and flow rate are $P_{\mathrm{nom}} = 2300W$ and $Q_{\mathrm{nom}} = 5.3\,kg/s$, respectively. This performance index characterizes the effectiveness of the control strategy in terms of the margins in the flow rate and the total pressure ratio between the nominal and the critical operating points.

The second performance index considered is the power balance, hereafter denoted as PB. It is defined as

$$PB = \left(\frac{(Q_{\mathrm{s,b}} + Q_{\mathrm{inj}}) \times p_{\mathrm{tt,c}}}{\rho_{\mathrm{c}}} - \frac{Q_{\mathrm{s,b}} \times p_{\mathrm{tt,b}}}{\rho_{\mathrm{b}}} \right) \times \frac{100}{P_{\mathrm{nom}}}, \tag{2}$$

where ρ denotes the density. The subscript s, c, b, inj and nom refers to the values of these different variables for the last stable point, the controlled cased, the baseline case, injection and nominal conditions, respectively. Compared to the surge margin improvement, the power balance performance index evaluates the cost of the control process by comparing the output power gain by the compressor and the input power paid to activate the actuators.

Both of these performance indices are schematically presented in Fig. 2. In Fig. 2a, two surge lines are plotted in red for the baseline case and the controlled case. The baseline configuration can be characterized by the stall pressure $p_{\mathrm{s,b}}$ and the stall mass flow rate $Q_{\mathrm{s,b}}$ at the last stable point LSP_b. Likewise, the controlled configuration is characterized by $p_{\mathrm{s,c}}$ and $Q_{\mathrm{s,c}}$ at the last stable point LSP_c. It should be emphasized finally that, compared to [4], the pressure rise p_{tt} across the compressor inlet for the controlled case is computed based on the mass average value in the jet and the mean flow.

2.2 Experimental dataset

In this study, we use data from 175 experiments to train and test the two neural network models. The experiments are conducted at LMFL laboratory by [4] in the framework of ACONIT project [32]. The specifications of the experimental parameters are summarized in Table 1. The dataset was split into 96% for training and 4% for testing, but five experiments are excluded due to missing values of the power balance PB. Additionally, five extra experiments were performed, outside the range of the training and testing dataset, to test the extrapolation capabilities of the two neural network models.

3 Surrogate model and optimization problem

We present in Sect. 3.1 a brief overview of the architecture and training procedure of the shallow neural networks used as surrogate models to approximate the input–output relationship between the control parameters and the two performance indices defined in the previous section. Then, Sect. 3.2 describes the formulation of the single- and multi-objective optimization problems and their resolution using genetic algorithms.

3.1 Shallow neural networks

Given the experimental dataset, we approximate the relationship between the control parameters (i.e., N_{inj}, α_{inj}, V_{inj} and Ω) and the two performance indices with a surrogate model built using a shallow feedforward neural network. Each of these two relationships is approximated using a different network, albeit with the same architecture. A schematic representation is shown in Fig. 3. The architecture is as follows:

- Each network takes as input a vector $\mathbf{x} \in \mathbb{R}^4$ defined as

$$x = [\alpha_{\mathrm{inj}}\ V_{\mathrm{inj}}\ N_{\mathrm{inj}}\ \Omega]\,.$$

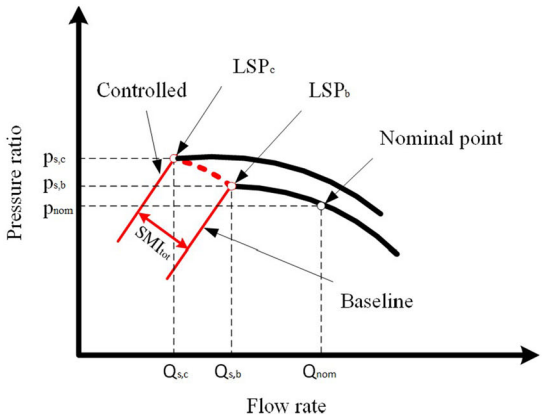
- The first hidden layer has 64 artificial neurons performing the following operations

$$z_i^{(1)} = \sigma\left(b_i^{(1)} + \sum_{j=1}^{4} W_{ij}^{(1)} x_j \right) \quad \text{for } i = 1, \cdots, 64,$$

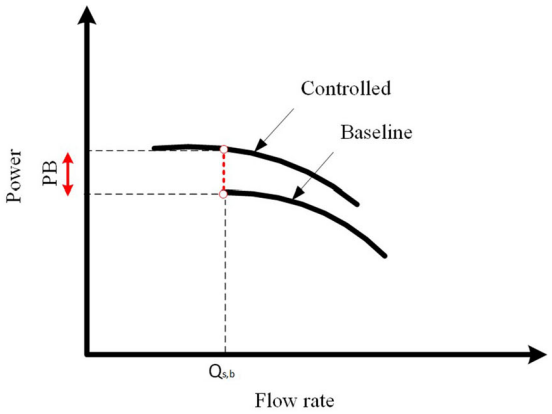
  where $z_i^{(1)}$ denotes the i$^{\mathrm{th}}$ output of this layer, and $\mathbf{b}^{(1)} \in \mathbb{R}^{64}$ (the bias vector) and $\mathbf{W}^{(1)} \in \mathbb{R}^{64\times 4}$ are learnable parameters. The activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is the ReLU defined as $\sigma(x) = \max(0, x)$.
- The second hidden layer, with 32 neurons, acts in a similar way on $\mathbf{z}^{(1)}$, i.e.,

$$z_i^{(2)} = \sigma\left(b_i^{(2)} + \sum_{j=1}^{64} W_{ij}^{(2)} z_j^{(1)} \right).) \quad \text{for } i = 1, \cdots, 32.$$

  Once again, the activation function σ is set to the ReLU function.
- Finally, the output layer takes as input $\mathbf{z}^{(2)}$ and maps it to the desired performance index. This map-



(a) Definition of the surge margin improvement



(b) Definition of the power balance

Fig. 2 Schematic definitions of **a** the *surge margin improvement* SMI_{tot} and **b** the *Power Balance* PB. Both panels depict an uncontrolled (i.e., baseline) and controlled surge line. In panel **a**, LSP_b denotes the *last stable point* in the uncontrolled case, while LSP_c is the corresponding last stable point when control is applied. Likewise, subscripts Q's and p's denote the corresponding flow rates and pressure ratios

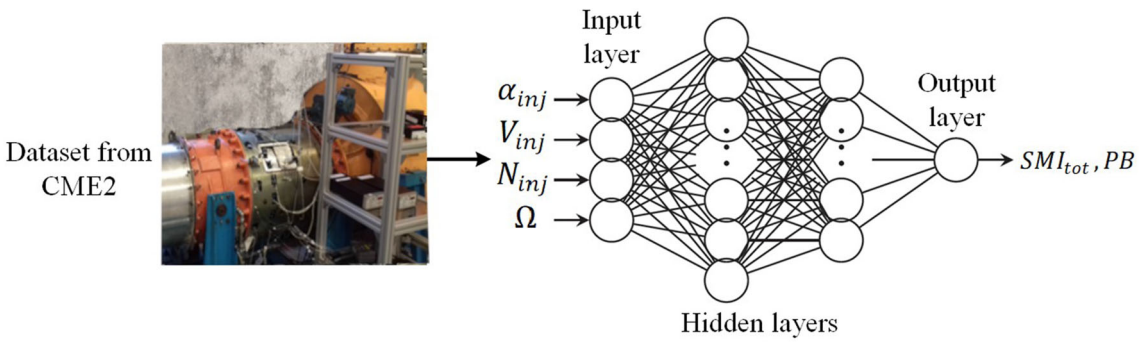


Fig. 3 Schematic description of the neural network model

ping simply reads 4

$$\hat{y} = b^{(o)} + \sum_{j=1}^{32} W_{1j}^{(o)} z_j^{(2)},$$

where $b^{(o)} \in \mathbb{R}$ (the bias vector) and $\mathbf{W}^{(o)} \in \mathbb{R}^{1\times 32}$ are the associated learnable parameters, and $\hat{y} \in \mathbb{R}$ is the approximation of either performance index.

Denoting by $y \in \mathbb{R}$ the ground truth value of either performance index, the learnable parameters of each layer are obtained by minimizing the following mean square error loss function

$$\mathscr{L}\left(\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)}, \mathbf{W}^{(o)}, b^{(o)}\right) = \frac{1}{M}\sum_{i=1}^{M}\left(y_i - \hat{y}_i\right)^2, \tag{3}$$

where $M = 168$ is the total number of data points in the training set. Both of the networks are implemented using Keras with the TensorFlow back end, while the ADAM optimizer [33] is used for the minimization of the cost function. The different parameters of the networks and optimizer are summarized in Table 2.

Table 2 Specification of shallow neural network's architecture and optimizer used for the learning process

Number of neurons per layer	$4 \to 64 \to 32 \to 1$
Activation function	$\text{ReLU}(x) = \max(0, x)$
Optimizer	ADAM
Learning rate	0.005
Epochs	550

3.2 Optimization problem and genetic algorithm

The aim of this study is to find the set of control parameters $\mathbf{x} \in \mathbb{R}^3$ maximizing either of the performance indices defined in Sect. 2.1, or a combination thereof. Letting $\mathscr{J}_1 : \mathbb{R}^3 \to \mathbb{R}$ be the surrogate model approximating the surge margin improvement SMI_{tot} and $\mathscr{J}_2 : \mathbb{R}^3 \to \mathbb{R}$ be the corresponding model for the power balance PB, we can thus formulate the following

multi-objective unconstrained optimization problem

$$\text{maximize} \quad (1-\lambda)\,\mathscr{J}_1(\mathbf{x}) + \lambda\,\mathscr{J}_2(\mathbf{x}), \tag{4}$$

where $\lambda \in [0,1]$ is a trade-off parameter. Note that setting $\lambda = 0$ or $\lambda = 1$ reduces Eq. 4 to a single-objective optimization problem. The maximizer $\mathbf{x}(\lambda)$ of Eq. 4 thus forms a parameterized family of optimal solutions forming a Pareto front. It cannot be guaranteed in general that the entries of the solution $\mathbf{x}$ will be in a given range when solving an unconstrained optimization problem. As such, we also consider the following constrained variation

$$\begin{aligned} \text{maximize} \quad & (1-\lambda)\,\mathscr{J}_1(\mathbf{x}) + \lambda\,\mathscr{J}_2(\mathbf{x}) \\ \text{subject to} \quad & -45^\circ \le \alpha_{\text{inj}} \le 30^\circ \\ & 46\text{m/s} \le V_{\text{inj}} \le 202\text{m/s} \\ & N_{\text{inj}} = 10 \text{ (or } 20). \end{aligned} \tag{5}$$

The benefit of using the constrained formulation of the optimization problem is that practitioners can force the solution (if it exists) to be in a given range of safe operating conditions. On the other hand, using an unconstrained formulation allows for the possibility of finding a global optimal solution outside of the range of the experimental data. It should be noted that our control system only supports N_{inj} values of 10 and 20 due to design constraints. Therefore, N_{inj} is set to either 10 or 20 in both constrained and unconstrained optimization processes. Additionally, compressors are typically designed to operate at a constant rotational velocity Ω; thus, the optimal control parameters are explored at $\Omega = 3200$, 4500 and 6000 RPM. The range of all experimental data is summarized in table creftab: set of configurations.

A genetic algorithm (GA) is employed to solve both problems Eqs. 4 and 5. The main search operators in GAs are crossover and mutation. These two operators can function as an exploration or exploitation operators [34]. In our genetic algorithm (1), random (uniform) mutation operation with a single-point crossover operator [34] is applied.

Algorithm 1 : Genetic algorithm

1: Population initialization;
2: Specify λ in $(1-\lambda)\,\mathscr{J}_1(\mathbf{x}) + \lambda\,\mathscr{J}_2(\mathbf{x})$;
3: Set constant value of Ω;
4: Set optimization boundaries for N_{inj}, α_{inj} and V_{inj};
5: **while** Stopping criteria is not met **do**
6: Objective function calculations;
7: Parents selection;
8: Crossover;
9: Mutation;
10: New-generation selection;
11: **end while**

3.3 Model selection and hyperparameter optimization

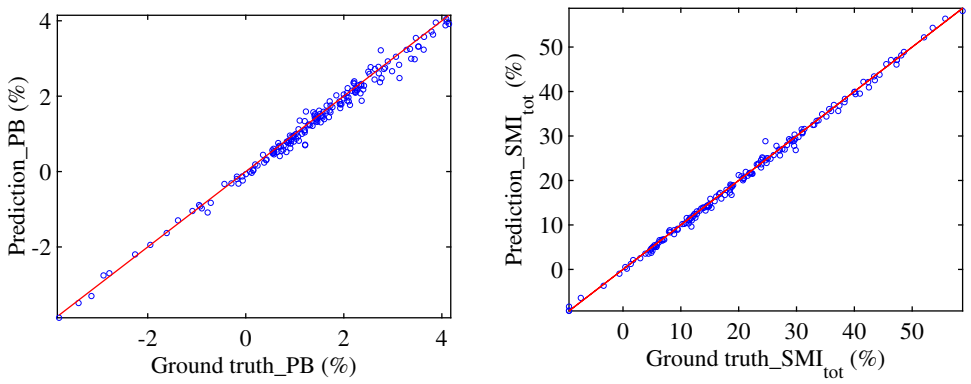
In this section, we discuss the methodology of tuning the neural network hyperparameters and the model selection criteria. Given a dataset in the domain $\mathscr{D}$, we split them into training and testing sets $\mathscr{D}_{\text{train}}$ and $\mathscr{D}_{\text{test}}$, respectively. In the process of splitting the original dataset $\mathscr{D}$, we use K-fold cross-validation method [35], which divides $\mathscr{D}$ into K equally sized samples. The K folds take the following form: $\mathscr{D}_{\text{train}} = \{D^{(1)}_{\text{train}}, \ldots, D^{(K)}_{\text{train}}\}$ for the training and $\mathscr{D}_{\text{test}} = \{D^{(1)}_{\text{test}}, \ldots, D^{(K)}_{\text{test}}\}$ for the testing datasets such that $D^{(i)}_{\text{train}} = \mathscr{D} \backslash D^{(i)}_{\text{test}}$ for $i = \{1, \ldots, K\}$ where K in the number of folds. Let $\mathscr{A} = \{A^{(1)}, \ldots, A^{(N)}\}$ denote a set of learning algorithms, and let the associated hyperparameters x of each algorithm $A^{(j)}$ having a domain $\Lambda^{(j)}$. Finally, let $\mathscr{L}({A_x}^{(j)}, {D^{(i)}}_{\text{train}}, {D^{(i)}}_{\text{test}})$ be the loss function of a certain learning algorithm $A^{(j)}$ trained on ${D^{(i)}}_{\text{train}}$ and tested on ${D^{(i)}}_{\text{test}}$ utilizing the hyperparameter x. This allows the model selection and the hyperparameter optimization equation to be written as

$$\text{minimize} \quad \frac{1}{K}\sum_{i=1}^{K} \mathscr{L}({A_x}^{(j)}, {D^{(i)}}_{\text{train}}, {D^{(i)}}_{\text{test}}). \tag{6}$$

The goal of Eq. 6 is to find the optimal neural network algorithm and hyperparameters, ${A_{x^*}}^{(\dagger)}$, that minimize the loss function, where x^* denotes the optimal hyperparameter and † refers to the optimal algorithm. Various methods are discussed in the literature [36–38] for tackling the optimization problem Eq. (6). In our study, we use a genetic algorithm for random search of the neural networks hyperparameters and 25 folds, i.e., $K = 25$, for our model selection. Furthermore, to simplify the hyperparameter optimization process, we use a neural network architecture with two hidden layers (encompassing 64 and 32 neurons, respectively). ReLU is employed as activation function and Adam as optimizer. Then we employ the GA for optimizing two parameters, namely the learning rate and the weight initialization.

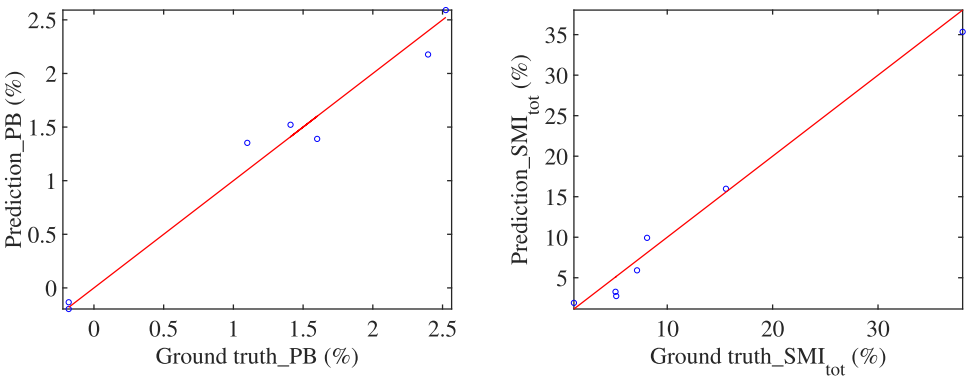
4 Results

This section summarizes the key results of our study. First, Sect. 4.1 describes the in-sample (training) and out-of-sample (testing and extrapolation) performances of the surrogate models. Then, the results of the optimization task pertaining to finding the optimal control parameters to delay rotating stall are being presented in Sect. 4.2.



(a) Power Balance PB (b) Surge margin improvement SMI$_{tot}$

Fig. 4 Comparison between the experimental results (ground truth) of the *training set* and the shallow network's predictions for **a** the power balance PB and **b** the surge margin improvement SMI_{tot}



(a) Power balance *PB* (b) Surge margin improvement *SMI*$_{tot}$

Fig. 5 Comparison between the experimental results (ground truth) of the *test set* and the shallow network's predictions for **a** the power balance PB and **b** the surge margin improvement SMI_{tot}

4.1 Surrogate models

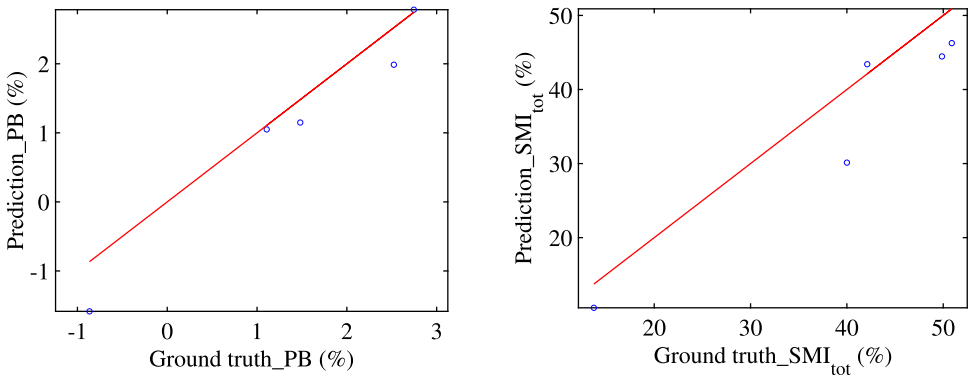
Given the training set, neural network surrogate models approximating the power balance PB and surge margin improvement SMI_{tot} have been trained until the loss function has stabilized. More precisely, the training procedures are performed for 550 epochs until a mean square error less than 10^{-3} is achieved. Figure 4a compares the prediction of the surrogate models with the ground truth data for the power balance PB, while Fig. 4b provides the same comparison for the surge margin improvement SMI_{tot}. The quality of this approximation can be quantified using the coefficient of determination R^2 defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{M} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{M} (y_i - \bar{y})^2},$$

where $\bar{y}$ denotes the sample mean of the performance index considered. In both cases, the surrogate models achieve a coefficient of determination $R^2 \geq 0.98$. The two surrogate models moreover exhibit similar performances when the test set is being considered, as shown in Fig. 5. Furthermore, Fig. 6 compares the predictions of the surrogate models with the actual data for the extrapolation dataset. The surrogate models achieve $R^2 = 0.89$ and 0.82 for PB and SMI_{tot}, respectively.

To better understand the capabilities and limitations of our model in extrapolating, we study the impact of splitting the dataset $\mathscr{D}$ between training $\mathscr{D}_{\text{train}}$ and testing $\mathscr{D}_{\text{test}}$; hence, we aim to investigate different combinations of $D^{(i)}_{\text{train}}$ and $D^{(i)}_{\text{test}}$. We found that the split of the input database between training and testing affects the robustness of the extrapolation prediction. To identify the cause of this sensitivity, we first obtained the optimal neural network architecture by employing a genetic algorithm for hyperparameter optimization as discussed in Sect. 3.3. Then, we trained the optimal neural network architecture on 2000 random data splits of the original database (i.e., different training/testing sets). Out of the analysis of our 2000 random data splits, we identified a narrow set of data that consistently resulted in poor predictions for both extrapolation and testing conditions. In order to confirm that these ill-conditioned testing points unconditionally undermines both testing and extrapolation predictions, we trained another 2000 different neural network architectures by forcing such data to belong to the testing dataset. As a result, none of these 2000 trials yielded a model that gives good predictions in terms of testing and extrapolation sets. In this process, we also utilized the GA for hyperparameter optimization, discussed in Sect. 3.3, but without cross-validation.

To further understand whether the ill-conditioned testing data could be correlated with the sensitivity of the underlying physics, we compute the gradients of the neural network outputs, i.e., PB and SMI_{tot}, with respect to the relevant inputs for the flow control system, i.e., the velocity ratio $V_{\text{R}} = V_{\text{inj}}/U$, where $U = \frac{\pi D \Omega}{60}$, and the angle of attack β_{att}, defined as the rela-



(a) Power balance PB (b) Surge margin improvement SMI_{tot}

Fig. 6 Comparison between the experimental results (ground truth) of the *extrapolation set* and the shallow network's predictions for **a** the power balance PB and **b** the surge margin improvement SMI_{tot}

tive injection angle β_{inj} minus the compressor stagger angle γ_{st}, which is fixed at 45°. We therefore computed $\partial_{V_{\text{R}}} SMI_{\text{tot}}$, $\partial_{\beta_{\text{att}}} SMI_{\text{tot}}$, $\partial_{V_{\text{R}}} PB$ and $\partial_{\beta_{\text{att}}} PB$ keeping constant Ω and N_{inj}. This protocol is therefore supposed to allow comparisons for different air jet control parameters (V_{inj} and α_{inj}) with the compressor parameters (U and γ_{st}), making it easier to compare the compressor's performance under different operating conditions (i.e., $\Omega = 3200RPM, 4500RPM \text{and} 6000RPM$).

Figure 7 shows $\partial_{V_{\text{R}}} PB$, and $\partial_{\beta_{\text{att}}} PB$, and similarly in Fig. 8 for the gradient components of SMI_{tot} at $\Omega =$ 3200 RPM and $N_{\text{inj}} = 20$. Figures 7 and 8 show that the ill-conditioned testing data (marked by red squares) are partially located in regions where there is a high sensitivity of the output gradients (sharp change of gradient components on the right end of the figures). We further stress, however, that most of the red squares are located well within the bulk region of the domain, where the outputs are not sensitive to the injection parameters V_{R} and β_{att}. Hence, we speculate that the physical features of the system are not responsible of the observed ill-conditioning of the debated testing inputs. A possible reason for the ill-conditioning of such testing data could be the limited dataset. In addition, the results suggest that data points far away from the interpolation/extrapolation boundary may severely affect the training of the models up to ill-condition the coherence of the model training. In fact, to have a reliable model, we need to include such ill-conditioned testing points in the training dataset.

We conducted two additional trials to test the significance of the ill-conditioned testing points. The first trial involved training 2000 neural network models using the same process discussed in Sect. 3.3, while intentionally including the ill-conditioned points in the training dataset. The goal was to decrease the K-fold number (10 folds instead of 25 folds), i.e., increase the testing dataset from 4% to 10%. Although the models converged rapidly for the same K folds (K= 25), we observed no significant benefits for lower K-fold tests (K= 10). In the second trial, we trained 2000 neural network models using the same process discussed in Sect. 3.3, after removing the ill-conditioned points from the dataset completely. The results of these tests showed that excluding the ill-conditioned points undermined the predictions for both testing and extrapolations.

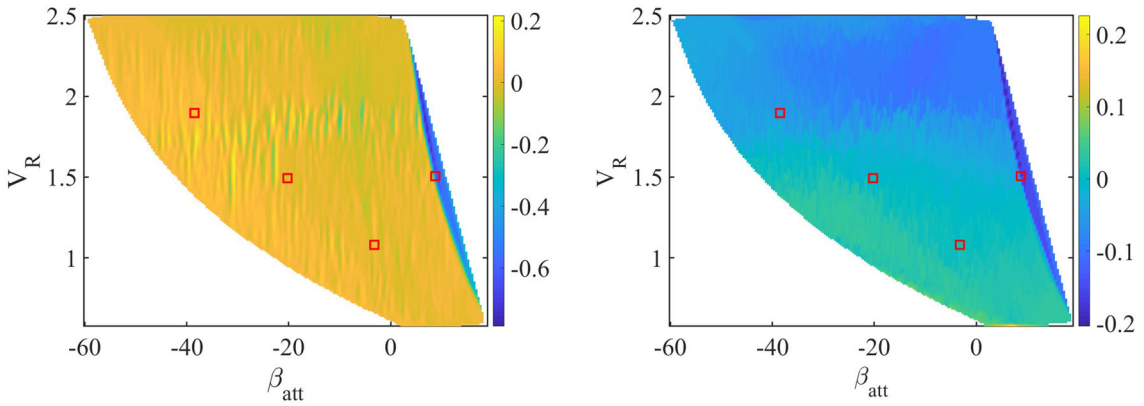
In conclusion, including the ill-conditioned points in the testing dataset or excluding them completely from the dataset unconditionally undermines the predictions of the testing and the extrapolation data. On the other hand, including them in the training dataset benefits the model training convergence, but it cannot be used to increase the testing percentage of the datasets. These results might indicate the deficiency of the limited number of dataset, but they also give an insight about the possible amount of data required to build accurate predictive models for similar problems.

4.2 Optimization results

Given the surrogate models presented in the previous section, we now turn our attention to the solutions of both optimization problems from Sect. 3.2, beginning with their single-objective variations in Sect. 4.2.1, while the results of the bi-objective formulation are presented in Sect. 4.2.2.

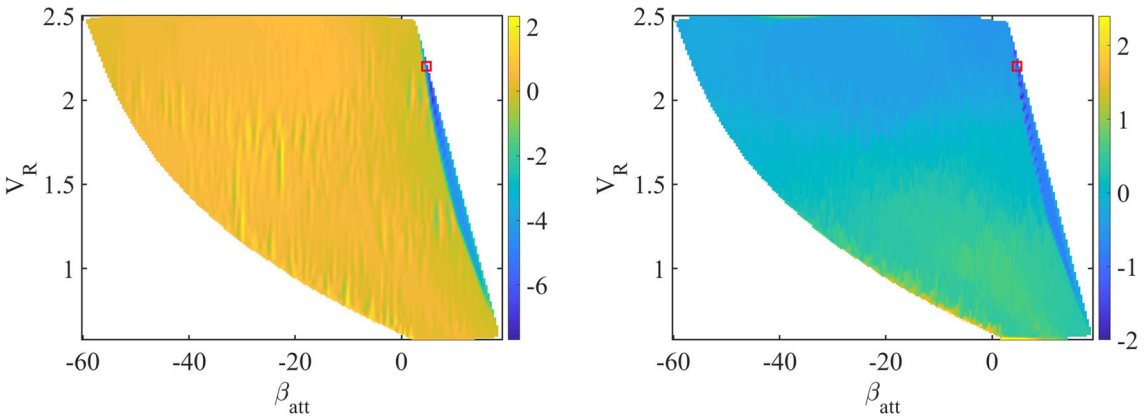
4.2.1 Single-objective optimization

Maximizing the power balance PB compared to the baseline configuration, optimizing the power balance performance index amounts to maximize the output power generated by the compressor for a given set of operating conditions (e.g., fix mass flow rate Q). Tables 3 and 4 summarize the results of the corresponding constrained and unconstrained maximization problems, respectively. Values highlighted in red correspond to optimized parameters which are at or beyond the extrema specified in the constrained case. These extrema correspond to those of the training set. It can be observed that the injection velocity V_{inj} increases almost linearly with the rotational velocity Ω, while the injection angle α_{inj} appears to decrease with increasing Ω. Note moreover that, for $\Omega =$ 3200 RPM and 4500 RPM, the injection angle is at or below the lower bound specified in the constrained problem. As discussed in Sect. 4.2.2, the Pareto-optimal points occur within a specific range of the velocity ratio V_R and the relative angle of attack β_{att}. The formu-



(a) $\partial_{\beta_{\text{att}}} PB$ (b) $\partial_{V_{\text{R}}} PB$

Fig. 7 Derivatives of **PB** with respect to β_{att} **a** and V_{R} **b** at $\Omega = 3200$ RPM and $N_{\text{inj}} = 20$. The ill-points are highlighted in red squares



(a) $\partial_{\beta_{\text{att}}} SMI_{\text{tot}}$ (b) $\partial_{V_{\text{R}}} SMI_{\text{tot}}$

Fig. 8 Derivatives of SMI_{tot} with respect to β_{att} **a** and V_{R} **b** at $\Omega = 3200$ RPM and $N_{\text{inj}} = 20$. The ill-points are highlighted in red squares

Table 3 Results of the constrained optimization procedure when aiming at maximizing the power balance performance index PB for the different operating conditions

Rotational velocity (RPM)	Injector pairs	Injection angle (°)	Injection velocity (m/s)
3200	20	**−45**	92
4500	20	**−45**	129
6000	20	**−19**	159

Note that for 3200RPM and 4500 RPM, the optimizer sets the injection angle at its maximum absolute value (see values in bold)

Table 4 Results of the unconstrained optimization procedure when aiming at maximizing the power balance performance index PB for the different operating conditions

Rotational velocity (RPM)	Injector pairs	Injection angle (°)	Injection velocity (m/s)
3200	20	**−65**	103
4500	20	**−59**	137
6000	20	-18	161

Note that for 3200RPM and 4500 RPM, the optimizer sets the injection angle (see values in bold)

lation of V_R and β_{att} is discussed in Sect. 4.1. As Ω increases, U increases, and thus, V_{inj} tends to increase in order to maintain the optimal V_R range. Similarly, as both V_{inj} and Ω increase, the injection angle α_{inj} tends to decrease to maintain the optimal β_{att} range.

Maximizing the surge margin improvement SMI_{tot} while we optimized for the power balance and let the surge margin improvement unchecked in the previous section, we now do the opposite: Optimize for the surge margin improvement and leave the power balanced unhampered. Optimizing for this performance index primarily aims at extending the range of conditions at which the compressor can safely operate. Tables 5 and 6 summarize the results of the corresponding constrained and unconstrained maximization problem, respectively.

Table 5 Results of the constrained optimization procedure when aiming at maximizing the surge marring improvement SMI_{tot} for the different operating conditions

Rotational velocity (RPM)	Injector pairs	Injection angle (°)	Injection velocity (m/s)
3200	20	–33	144
4500	20	–33	165
6000	20	**–45**	**202**

Note that for 6000 RPM, the optimizer sets the both the injection angle and the injection velocity to their maximum allowed value (see values in bold)

Table 6 Results of the unconstrained optimization procedure when aiming at maximizing the surge marring improvement SMI_{tot} for the different operating conditions

Rotational velocity (RPM)	Injector pairs	Injection angle (°)	Injection velocity (m/s)
3200	20	–32	144
4500	20	–31	166
6000	20	**–45**	**206**

Note that for 6000 RPM, the optimizer sets the injection velocity beyond its maximum allowed value (see values in red)

As before, values highlighted in red correspond to optimized parameters at or beyond the bounds specified in the constrained formulation. While the evolution of the injection velocity with respect to the rotational velocity of the rotor appears to follow a similar trend as what is observed when maximizing the power balance (i.e., increasing V_{inj} with increasing Ω), the optimized values are comparatively larger (as much as 40% larger for $\Omega = 3200$ RPM). More importantly, the evolution of the injection angle α_{inj} is different. While α_{inj} decreases with Ω when maximizing the power balance, it now appears to stay constant (if not increase) as Ω increases when the surge margin performance index is considered. As discussed in Sect. 4.2.2, the Pareto-optimal points are achieved within a specific range of V_R and β_{att}. This can help explaining the evolution trends of V_{inj}. Additionally, it can be observed that at a rotational velocity of 6000 RPM, the optimal value of V_{inj} for surge margin improvement is higher than that for power balance. This difference in V_{inj} could explain the reverse trend in the evolution of α_{inj}. In other words, as V_{inj} is higher in the case of SMI_{tot}, for the same rotational velocity, α_{inj} tends to increase in order to maintain the same relative injection angle β_{inj} and thus achieve the same optimal value of β_{att}.

4.2.2 Bi-objective optimization

As discussed in the previous section, maximizing the power balance PB and the surge margin improvement SMI_{tot} leads to conflicting evolution of the optimized parameters, most notably for the injection angle. Aiming at achieving the best of these two objectives, we now turn our attention to the bi-objective optimization Eq. 5. In particular, we look at how the solution of this optimization problem changes as the trade-off parameter λ evolves. The bi-objective optimization processes are performed based on constrained and unconstrained conditions.

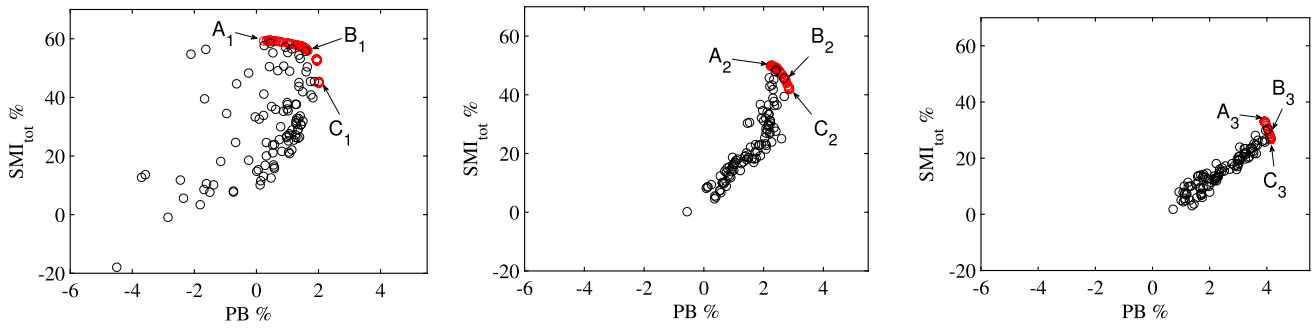
This process begins by analyzing the case where control parameters are constrained. Figure 9 shows the non-dominated optimal points (i.e Pareto front) at different rotational velocities (3200 RPM, 4500 RPM and 6000 RPM). Each Pareto front is labeled by three optimal points: A_i, C_i and B_i. Points A_i correspond to the maximization of surge margin improvement, points C_i correspond to the maximization of power balance, and points B_i represent the Pareto-optimal solutions. The specifications of points A_i, C_i and B_i are summarized in Table 7. It can be observed that as we pass from points A_i to C_i at each rotational velocity, the value of V_R increases linearly. The value of β_{att} follows a similar trend as V_R, except at $\Omega = 6000$ RPM. The physical interpretation of these observations is the subject of ongoing study.

Comparisons between the non-dominated optimal points for the constrained case are also shown in Fig. 10. A notable feature is the similarity in the range of V_R and β_{att} where all Pareto-optimal points (B_i) are achieved. In particular, as shown in Fig. 10 and Table 7, the Pareto-optimal points (B_i) can be obtained at V_R between 1.2 and 1.4 and β_{att} between 7° and 11°. These similarities in the range of V_R and β_{att} at different operating conditions may be due to the self-similar flow topology of the air jets, as discussed in [39]. This issue is also subjected to the ongoing research study.

For the case where control parameters are unconstrained, Fig. 11 shows similar trends compared to the constrained optimization Fig. 10, except for some intersections between rotational velocities of 3200RPM and 4500RPM. Additionally, small changes can be observed in the range of V_R and β_{att} where the Pareto-optimal points (B_i) are achieved. This range, in the unconstrained case, is between 1.1 and 1.6 and between 1° and 11° for V_R and β_{att}, respectively. The similarities in the range of V_R and β_{att} at all operating conditions suggest the possibility of developing non-dimensional forms of the control parameters. These dimensionless quantities aim to generalize the active flow control strategy and make it applicable to other compressors.

5 Conclusion

A machine learning optimization algorithm is used to identify the optimal air jet control parameters for active flow control in an axial compressor CME2. Two neural networks are first created to model the relationship between our control parameters and two performance metrics, namely power balance and surge mar-



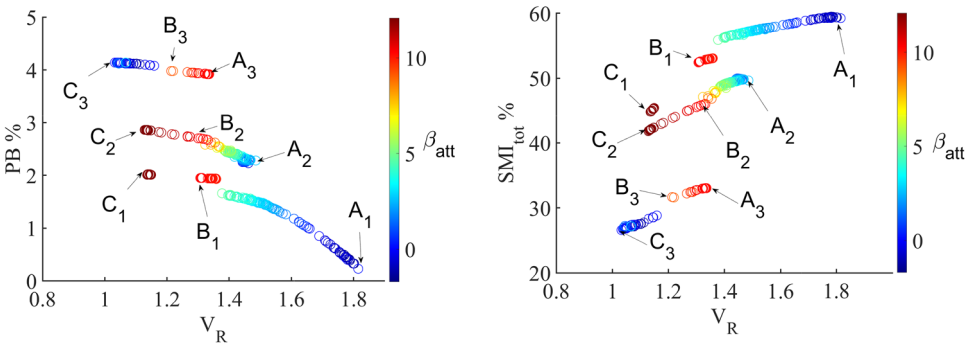
(a) $\Omega = 3200$RPM (b) $\Omega = 4500$RPM (c) $\Omega = 6000$RPM

Fig. 9 Results of the bi-objective optimization problem in the (PB-SMI_{tot}) plane for different rotational velocities Ω. Points in red highlight the Pareto front of this bi-objective optimization problem, *i.e.,* the set of optimal values obtained for varying trade-off λ. Points in black correspond to the values of the two objectives obtained for random set of input parameters. In each panel, points labeled A_i and C_i correspond to the maximization of the surge margin improvement or power balance, respectively. Points labeled B_i highlight the Pareto-optimal solutions. The set of optimized parameters for each of these points is summarized in Table 7

Table 7 Specifications of non-dominated optimal points resulting from the bi-objective optimization problem

Point	Ω (RPM)	N_{inj}	α_{inj} (°)	V_{inj} (m/s)	V_R	β_{att} (°)	SMI_{tot}	PB
A_1	3200	20	–33	144	1.81	0.2	59	0.2
B_1	3200	20	–45	105	1.31	10.4	52	1.9
C_1	3200	20	–45	92	1.14	11.9	45	2.0
A_2	4500	20	–31	165	1.45	0.4	50	2.3
B_2	4500	20	–45	151	1.33	10.1	46	2.7
C_2	4500	20	–45	129	1.13	12	42	2.9
A_3	6000	20	–45	202	1.34	10.1	33	3.9
B_3	6000	20	–41	185	1.22	8.7	32	3.9
C_3	6000	20	–19	159	1.04	–0.5	27	4.1

Points labeled A_i and C_i correspond to the maximization of the surge margin improvement or power balance, respectively. Points labeled B_i correspond to the best trade-off between the two conflicting objectives. The optimal values highlighted in red are the limits of the training dataset range



(a) (b)

Fig. 10 Comparison between the non-dominated optimal points of the Pareto fronts for **constrained** optimization for **a** the power balance PB and **b** the surge margin improvement SMI_{tot}. The points show non-dominated optimal points at $\Omega = 3200$ RPM, 4500 RPM and 6000 RPM. Each point is colored by β_{att}. Points labeled A_i and C_i correspond to the maximization of the surge margin improvement or power balance, respectively. Points labeled B_i correspond to the best trade-off between the two conflicting objectives

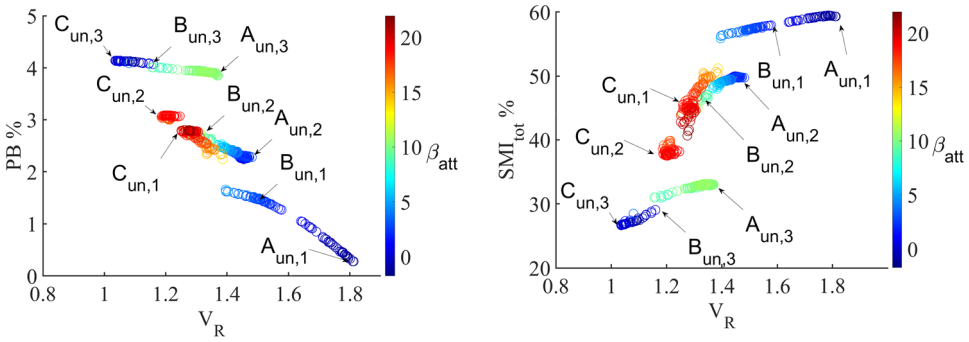


Fig. 11 Comparison between the non-dominated optimal points of the Pareto fronts for **unconstrained** optimization for **a** the power balance PB and **b** the surge margin improvement SMI_{tot}. The points show non-dominated optimal points at $\Omega = 3200$ RPM, 4500 RPM and 6000 RPM. Each point is colored by β_{att}. Points labeled A_i and C_i correspond to the maximization of the surge margin improvement or power balance, respectively. Points labeled B_i correspond to the best trade-off between the two conflicting objectives

gin improvement. A genetic algorithm is then used to search for the optimal control parameters. The study focuses on three air jet control parameters: absolute injection angle, number of injector pairs and absolute injection velocity, at three different rotational speeds (3200 RPM, 4500 RPM and 6000 RPM). The neural networks are trained and tested using a dataset from 175 experiments. Additionally, 5 experiments are conducted to evaluate the networks' extrapolation capabilities.

The two neural networks are trained for 550 epochs until a mean square error less than 10^{-3} is reached. All neural networks achieve a coefficient of determination of $R^2 \geq 0.98$ for the training and testing dataset. When extrapolating to new data, the networks had $R^2 = 0.89$ and 0.82 for the power balance and surge margin improvement performance metrics, respectively. A sensitivity analysis has been carried out by investigating the gradients of the neural network outputs with respect to the flow control parameters. We demonstrated that having limited dataset can allow to build a reliable neural network model. Special attention must be paid to the database splitting, as limited databases could require sensitive data to be part of the training rather than the testing dataset. Hence, if a study aims to produce a neural network model for applications with a parameter space comparable to ours and their model relies on a limited database, a sensitivity analysis is here advised. It is worth noting that, to the best of our knowledge, the dataset we utilized is one of the most comprehensive databases available in the literature for active flow control in axial compressors, given the large dimensionality of the control parameter space (i.e., injection velocity, injection angle, number of injectors and rotational velocity). This means that limited databases are a challenge to face in active flow control applied to axial compressors, which further proves the widespread importance of sensitivity analyses.

After this, single- and bi-objective optimization using these neural network models are performed under both constrained and unconstrained control parameter conditions. The results of the bi-objective optimization showed that the Pareto-optimal points were achieved at specific ranges of V_R and β_{att}. When control parameters were constrained, the optimal range for V_R was between 1.2 and 1.4, and for β_{att} was between 7 and 11 degrees. When control parameters were unconstrained, the optimal range for V_R was between 1.1 and 1.6, and for β_{att} was between 1 and 11 degrees. This similarity in the range of the optimal control parameters suggests the possibility of developing control laws that could be applied to other compressor geometries. Additionally, this work opens up opportunities for further research to explore the underlying physics behind the observations made in this study. For example, further study could be done to understand the reasons behind the reversed trend of the relative angle of attack of the non-dominated optimal points at 6000 RPM.

Author contribution statement

All the authors were involved in preparing the manuscript. All the authors have read and approved the final manuscript.

Data availability Statement The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Financial interests This paper is supported by Clean Sky European Union's Horizon 2020 research and innovation program under grant agreement No 886352, project ACONIT.

Conflict of interest The authors declare no conflict of interest

References

1. C. Tan, I. Day, S. Morris, A. Wadia, Spike-type compressor stall inception, detection, and control. Annu. Rev. Fluid Mech. **42**, 275–300 (2010)
2. M.D Hathaway, Passive endwall treatments for enhancing stability. Technical report (2007)
3. G. Margalida, P. Joseph, O. Roussette, A. Dazin, Active flow control in an axial compressor for stability improvement: on the effect of flow control on stall inception. Exp. Fluids **62**(1), 1–13 (2021)
4. J. Moubogha Moubogha, G. Margalida, P. Joseph, O. Roussette, A. Dazin, Stall margin improvement in an axial compressor by continuous and pulsed tip injection. Int. J. Turbomach. Propuls. Power **7**(1), 10 (2022)
5. C. Jang, K. Kim, Optimization of a stator blade using response surface method in a single-stage transonic axial compressor. Proc. Inst. Mech. Eng. Part A: J. Power Energy **219**(8), 595–603 (2005)
6. T.W. Simpson, T.M. Mauery, J.J. Korte, F. Mistree, Kriging models for global approximation in simulation-based multidisciplinary design optimization. AIAA J. **39**(12), 2233–2241 (2001)
7. T. Mengistu, W. Ghaly, Aerodynamic optimization of turbomachinery blades using evolutionary methods and ann-based surrogate models. Optim. Eng. **9**(3), 239–255 (2008)
8. J.A. Nelder, R. Mead, A simplex method for function minimization. Comput. J. **7**(4), 308–313 (1965). https://doi.org/10.1093/comjnl/7.4.308
9. T. Bäck, H.-P. Schwefel, An overview of evolutionary algorithms for parameter optimization. Evol. Comput. **1**(1), 1–23 (1993)
10. J. Li, X. Du, Martins, J.R.: Machine learning in aerodynamic shape optimization. arXiv preprint arXiv:2202.07141 (2022)
11. Z. Liu, I.A. Karimi, Gas turbine performance prediction via machine learning. Energy **192**, 116627 (2020)
12. J. Yu, Z. Wang, F. Chen, J. Yu, C. Wang, Kriging surrogate model applied in the mechanism study of tip leakage flow control in turbine cascade by multiple dbd plasma actuators. Aerosp. Sci. Technol. **85**, 216–228 (2019)
13. Z. Li, X. Zheng, Review of design optimization methods for turbomachinery aerodynamics. Prog. Aerosp. Sci. **93**, 1–23 (2017)
14. S.L. Brunton, B.R. Noack, Closed-loop turbulence control: progress and challenges. Appl. Mech. Rev. **67**(5), 1 (2015). https://doi.org/10.1115/1.4031175
15. F. Ren, H.-B. Hu, H. Tang, Active flow control using machine learning: a brief review. J. Hydrodyn. **32**(2), 247–253 (2020)
16. T. Bäck, H.-P. Schwefel, An overview of evolutionary algorithms for parameter optimization. Evol. Comput. **1**(1), 1–23 (1993)
17. A. Debien, K.A.F.F. von Krbek, N. Mazellier, T. Duriez, L. Cordier, B.R. Noack, M.W. Abel, A. Kourta, Closed-loop separation control over a sharp edge ramp using genetic programming. Exp. Fluids **57**(3), 40 (2016). https://doi.org/10.1007/s00348-016-2126-8
18. R. Li, B.R. Noack, L. Cordier, J. Borée, F. Harambat, Drag reduction of a car model by linear genetic programming control. Exp. Fluids **58**(8), 103 (2017). https://doi.org/10.1007/s00348-017-2382-2
19. Z. Wu, D. Fan, Y. Zhou, R. Li, B.R. Noack, Jet mixing enhancement using machine learning control. Experiments in Fluids (Accepted in print), 35 (2018). Springer Verlag (Germany). Accessed 8 Jan 2023
20. J. Rabault, M. Kuchta, A. Jensen, U. Réglade, N. Cerardi, Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. J. Fluid Mech. **865**, 281–302 (2019)
21. J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de las Casas, C. Donner, L. Fritz, C. Galperti , A. Huber, J. Keeling, M. Tsimpoukelli, J. Kay, A. Merle, J.-M. Moret, S. Noury, F. Pesamosca, D. Pfau, O. Sauter, C. Sommariva, S. Coda, B. Duval, A. Fasoli, P. Kohli, K. Kavukcuoglu, D. Hassabis, M. Riedmiller, Magnetic control of tokamak plasmas through deep reinforcement learning. Nature 602(7897), 414–419 (2022). https://doi.org/10.1038/s41586-021-04301-9. Accessed 2022-05-19
22. P. Garnier, J. Viquerat, J. Rabault, A. Larcher, A. Kuhnle, E. Hachem, A review on Deep Reinforcement Learning for Fluid Mechanics. arXiv:1908.04127 [physics] (2021)
23. R. Paris, S. Beneddine, J. Dandois, Robust flow control and optimal sensor placement using deep reinforcement learning. J. Fluid Mech. **913**, 25 (2021). https://doi.org/10.1017/jfm.2020.1170
24. H. Tang, J. Rabault, A. Kuhnle, Y. Wang, T. Wang, Robust active flow control over a range of Reynolds numbers using an artificial neural network trained through deep reinforcement learning. Phys. Fluids **32**(5), 053605 (2020). https://doi.org/10.1063/5.0006492
25. M. Elhawary, Deep Reinforcement Learning for Active Flow Control around a Circular Cylinder Using Unsteady-mode Plasma Actuators. arXiv preprint arXiv:2012.10165 (2020)
26. N. Benard, J. Pons-Prats, J. Periaux, G. Bugeda, P. Braud, J. Bonnet, E. Moreau, Turbulent separated shear flow control by surface plasma actuator: Experimental optimization by genetic algorithm approach. Exp. Fluids **57**(2), 1–17 (2016)
27. D. Kamari, M. Tadjfar, A. Madadi, Optimization of sd7003 airfoil performance using tbl and cbl at low reynolds numbers. Aerosp. Sci. Technol. **79**, 199–211 (2018)
28. S.L. Brunton, B.R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics. Annu. Rev. Fluid Mech. **52**, 477–508 (2020)
29. T. Duriez, S.L. Brunton, B.R. Noack, Machine Learning Control - Taming Nonlinear Dynamics And Turbulence. Fluid Mech. Appl. **116**, 1. https://doi.org/10.1007/978-3-319-40624-4
30. M. Veglio, A. Dazin, O. Roussette, G. Bois, Pressure measurements in an axial compressor: from design operating conditions to rotating stall inception. In: 49th Symposium of Applied Aerodynamics (2014)
31. S.B.-F.K.-J. Raub, R. Niehuis, Ejector tip injection system for active aerodynamic compressor stabilization part ii: Cfd investigations (2017)

32. A. Dazin, P. Joseph, F. Romano, Q. Gallas, J. Marty, G. Aigouy, M. Stôβel, R. Niehuis, The aconit project: an innovative design approach of active flow control for surge prevention in gas turbines. In: IOP Conference Series: Materials Science and Engineering, vol. 1024, p. 012068 (2021). IOP Publishing
33. D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs] (2017)
34. S.M. Lim, A.B.M. Sultan, M.N. Sulaiman, A. Mustapha, K.Y. Leong, Crossover and mutation operators of genetic algorithms. Int. J. Mach. Learn. Comput. **7**(1), 9–12 (2017)
35. L.A. Yates, Z. Aandahl, S.A. Richards, B.W. Brook, Cross validation for model selection: a review with examples from ecology. Ecol. Monogr. **1557**, 1 (2022)
36. T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2623–2631 (2019)
37. L. Kotthoff, C. Thornton, H.H. Hoos, F. Hutter, K. Leyton-Brown, Auto-weka: Automatic model selection and hyperparameter optimization in weka. Automated machine learning: methods, systems, challenges, 81–95 (2019)
38. M. Feurer, F. Hutter, Hyperparameter optimization. Automated machine learning: Methods, systems, challenges, 3–33 (2019)
39. O. El Mokkadem, X. Chen, C. Phan, J. Delva, P. Joseph, A. Dazin, F. Romanò, Small widths wall-attached coandă jets for flow control (2022)