Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Assisted Turbulence Modeling via Data Assimilation and Machine Learning for Turbomachinery Flows

Tom Moussie

tom.moussie@ensam.eu

Arts et Métiers

Pierric Joseph

Arts et Métiers

Marcello Meldi

Arts et Métiers

Research Article

Keywords: Data Assimilation, Machine Learning, Kalman Filter, Turbomachinery, Mixture of Experts

Posted Date: November 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11430624/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Assisted Turbulence Modeling via Data Assimilation and Machine Learning for Turbomachinery Flows

Tom Moussie[1*†], Pierric Joseph[1†] and Marcello Meldi[1†]

[1*]Univ. Lille, CNRS, ONERA, Arts et Métiers ParisTech, Centrale Lille, UMR 9014-LMFL- Laboratoire de Mécanique des fluides de Lille, Kampé de Feriet, F-59000, Lille, France.

*Corresponding author(s). E-mail(s): tom.moussie@ensam.eu;
†These authors contributed equally to this work.

Abstract

This work presents the development and assessment of a hybrid framework coupling numerical simulation, experimental samples, data assimilation (DA) and machine learning (ML) to investigate the flow within an axial compressor stage. The methodology first implements a Multi-Fidelity Ensemble Kalman Filter (MFEnKF) which integrates experimental PIV data with a two-fidelity numerical simulation setup, combining steady-state RANS on a fixed grid with unsteady hybrid RANS/LES simulations on a moving grid. This data assimilation step successfully reduces discrepancies between experiments and CFD realizations. Besides, it provides additional high fidelity samples when compared with the available PIV data. This enriched database is used to train machine learning (ML) tools, with the objective to obtain black-box forcing terms to be included in the discretized form of the Navier-Stokes equations. These models are expected to preserve the accuracy of the DA results while not longer requiring the input of experimental data. The neural network architecture used here maps local physical features such as the velocity, the velocity gradients and the turbulent viscosity, to obtain the volume forcing terms. The analysis of the performance of such ML models shows that global accuracy is preserved but localized increased errors in specific regions of the compressor, such as close to the blade tips, can exceed those observed from the baseline RANS simulations. To resolve this issue, a Mixture of Experts (MoE) method is employed, which classifies and combines the trained ML tools for different flow regions. The discrepancy with experimental data of this method is systematically lower than the one observed for the baseline RANS solution and very similar to the one obtained with the MFEnKF.

Keywords: Data Assimilation, Machine Learning, Kalman Filter, Turbomachinery, Mixture of Experts

1 Introduction

The improvement of the performance of energy devices in many industrial sectors is a key objective in current research activities in fluid mechanics. Among the objectives targeted, one can list increased efficiency as well as strict reduction of the carbon footprint associated with the industrial processes. Among the sectors that are strongly affected by the achievement of such objectives, aircraft engine manufacturing needs continuous advancement due to environmental regulation and fluctuating energy prices. Among all the components of aircraft engines, the axial compressors which equip the vast majority of the current engine generation are under severe scrutiny. In fact, they need to balance requirements for high energy efficiency (with respect to the Brayton-Joule thermodynamic cycle) with aircraft safety regulations due to the potential stability issues encountered at high pressure ratio and low mass flow rates. The importance of these components for the aeronautic industry resulted in extended investigations of several research teams since the 1940s, as comprehensively discussed in the review by Day [1]. Scientific advancement in recent decades highlights the global gains in the use of smaller and globally more compact engine cores to increase thermal efficiency [2]. This produces higher mechanical loads for the compressor stages, which has led to the development and use of unconventional design of the stages such as high turning stator blades [3] or rotors with tandem blades [4], possibly coupled with active flow control [5]. The development of these exotic rotor designs raise questions about the ability of the current industrial tools to correctly predict the global performance for such complex designs, in particular in terms of the prediction of the flow field around the blades. These flows are usually at high Reynolds regimes and they exhibit compressibility effects, therefore requiring advanced strategies for accurate prediction. These new research directions add to various existing or upcoming problems inherent to the implementation of such machines in aero engines. First, both military applications (with stealth requirement [6]) and civil ones (such as new architectures with boundary layer ingestion [7]) can require engines to be deeply buried inside the fuselage. This can lead to severe levels of inlet distortion [8] with, again, an urgent need to correctly assess the performance of the machines in such degraded conditions. Additionally, axial compressors are known to exhibit diverse and complicated patterns when transitioning to stall [9] and early detection of such unstable regimes is still an open topic to date [10]. This calls, here again, for fast and computationally efficient methods to predict flow across compressor stages. Experiments and high fidelity numerical simulations, such as Direct Numerical Simulation (DNS) [11] and Large Eddy Simulation (LES) [12], are the most reliable tools which can be used to assess and enhance the design strategies previously described. However, these investigative methods are expensive for turbomachinery flows and therefore comprehensive analyses considering several geometric features for multiple working configurations are virtually impossible. To this end, lower-fidelity strategies such as Reduced Order

Models (ROM) [13] or Reynolds-Averaged Navier-Stokes (RANS) simulation [14] can be used to investigate a large range of configurations. Computational costs are here reduced permitting a larger degree of freedom on the investigations thanks to the production of larger databases, at the expense of lower accuracy in the prediction of the results. Recent research has aimed to improve the accuracy of such models using tools from Machine Learning and data-driven architectures [15]. Several promising techniques have been developed in particular for RANS turbulence modeling [16, 17], which have shown increased accuracy for specific test cases. However, difficulties have been observed when applying models trained with specific datasets to different test cases, highlighting a specific challenge in extrapolation and generalizability of such models.
The difficulties arising from the different features and resources required for investigative tools can be reconciled using methods from Data Assimilation (DA) [18–20]. This branch of statistics [21] is devoted to obtaining accurate state estimation via data fusion from different sources of information. One notable feature of DA tools is the possibility to combine predictions from low-fidelity models, which can usually span very large parametric spaces producing quasi-continuous solutions in space and time, with limited measurements obtained from high-fidelity tools. For the latter, scarcity of data can be associated with limited availability of configurations for selected parameters of the setup of the problem investigated (due to computational cost requirement for DNS and LES) as well as to physical or temporal constraints in data acquisition, due for example to accessibility problems in experiments.
In the present work, an approach based on the Ensemble Kalman Filter (EnKF) [22, 23] is used to investigate the performance of the CME2 compressor test bench. The geometry of this test case is simpler than the ones observed in operative machines, but still permits to fully investigate the problematic aspects observed in realistic industrial and aerospace applications. Data from different sources are combined to obtain a robust state estimation of the flow. The methodology used to this purpose relies on a multi-fidelity approach [24] where several RANS realizations and fewer hybrid RANS/LES runs are combined with experimental measurements. The latter are available only on three two-dimensional planes within the compressor blades. Therefore they are localized on a small part of the domain of investigation. The DA state estimate thus obtained is then used to train corrective ML black box tools, which are generated using Neural Networks (NN), following the approach proposed by Villiers et al. [25]. Similar combinations of ensemble DA and ML have recently shown promising results for the state estimation of unsteady flows [26, 27]. These ML models are trained with the objective to reduce the discrepancy between the DA state estimation and the prediction of the low-fidelity RANS simulations, therefore targeting an increase of the predictive accuracy of the latter. The NN generative tools are then coupled using turbulent models and validated via a posteriori investigation and comparison with experimental data.
The manuscript is structured as follows. In section 2, the experimental setup and the data acquisition methods are presented. In section 3, the numerical ingredients, including the dynamic equations and the data assimilation procedure, are introduced. Section 4 provides an introductory numerical overview of the CME2 axial compressor. It is followed by section 5 in which the data assimilation algorithm is applied, thanks

to the experimental and numerical data respectively obtained in sections 2 and 4. Then, section 6 provides an overview of the machine learning tools used, presenting their architecture, training data and post-training metrics, as well as the results of integrating pre-trained tools into CFD solvers. Finally, concluding remarks are drawn in section 7.

2 Test case of investigation: the CME2

The test case investigated in the present study is the *Compresseur Mono-Étage, version 2* (CME2) which is a compressor test bench located at Arts et Métiers, Institute of Technology, in Lille, France. This rig, which is shown in Figure 1, consists of a single stage, low-speed, axial compressor operated using an electrical AC motor of 200 kW. The whole setup also includes an instrumented inlet section to measure the mass flow rate and to filter the incoming air, along with a throttling valve at the exhaust to control the operating point of the machine.

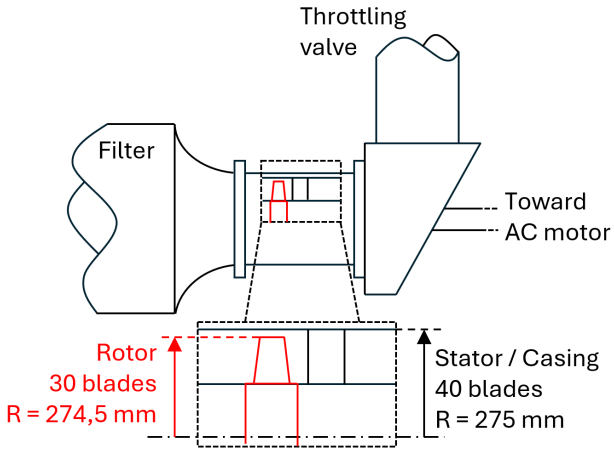
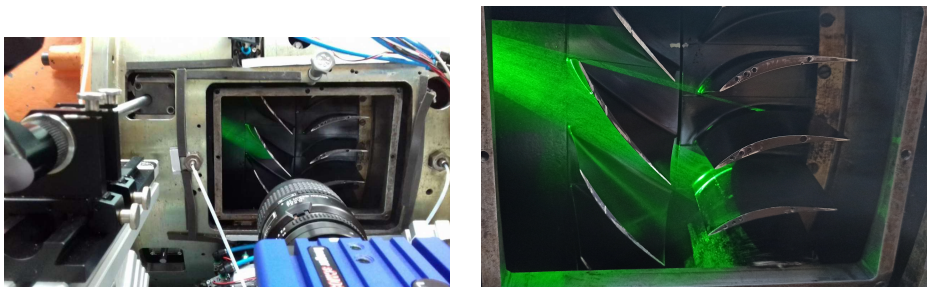


Fig. 1: Schematic description and main dimensions of the CME2 axial compressor.

The stage is composed of a 30-blade rotor and a 40-blade stator, positioned in a casing with a radius of 275 mm, as schematized in Figure 1. The rotor tip gap is 0.5 mm. The mass flow rate for 3 200 rpm is 5.3 kg $\cdot$ s^{-1} at the design point, which corresponds to a pressure ratio around 1.03. The Mach numbers at the inlet and at the rotor tip are $Ma_\infty = 0.12$ and $Ma_T = 0.28$, respectively.
The test bench in Lille has been used for several experimental campaigns to study problematic aspects observed in realistic configurations with different techniques and acquisition systems. Among the different tools used, one of interest for the present analysis is the Particle Image Velocimetry (PIV) which has been used to measure the velocity field inside the stage of the rotating machine. This system has been extensively described in previous articles from the research group (see [28–30]), but the main

characteristics are here described for the sake of completeness. This is a standard two-dimensional two-component (2D2C) equipment able to sample the velocity field over multiple planes located at a constant radius of the blades in the rotor section of the stage. The system itself is composed of a multi-pulse laser (Innolas Compact 400 PIV) coupled to an sCMOS camera from LaVision (Imager, 2 560 × 2 160 pixels). Optical access is performed via a transparent window on the side of the compressor. The projection of the laser beam into the compressor at the right location requires a sophisticated setup involving an articulated arm fitted with various mirrors and an endoscope at the end. Seeding is usually realized with a smoke generator installed at the compressor inlet. Additional pictures of the setup are provided in Figure 2 for illustration purposes.



(a) (b)

Fig. 2: (a) Experimental setup showing the laser beam and the camera; (b) Close-up of the laser beam in the compressor.

Measurements used here have been obtained on planes at 18 %, 51 % and 79 % of the blade height (see Figure 3), respectively referred to as **P1**, **P2** and **P3**, giving a field of view of approximately $64 \times 76\,\mathrm{mm}^2$. A small variation of the size of this area is reported between the three locations due to the change in blade shape with the radius, and the corresponding planes are shown in Figure 4.
The resulting pairs of images are post-processed using a code developed by the team based on a multi-pass correlation algorithm. In the present case, the final velocity fields have a spacing of 0.5 mm between adjacent vectors after using three interrogation windows (subsequently 64 × 64 pixels, 32 × 32 pixels and 64 × 64 pixels with 50 % overlap). Measurements were repeated 100 times at fixed intervals and averaged to produce the data used in this study. **P1**, **P2** and **P3** maps give access to 8 180, 6 944, and 8 090 data points where the velocity field is measured, respectively. For each point, two components of the velocity field are available. The first corresponds to the flow axial velocity (aligned with the compressor axis) and the second to a projection of the tangential velocity for the vertical component. The projection error (due to the casing curvature) is estimated to be less than 1 %, considering that the PIV observation region is small when compared with the global size of the compressor. The relative velocity can be computed using the local rotational velocity of the blade at each radius. The axial velocity mapped on the PIV point locations is displayed in Figure 5.

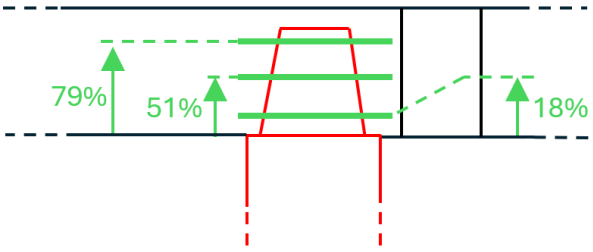
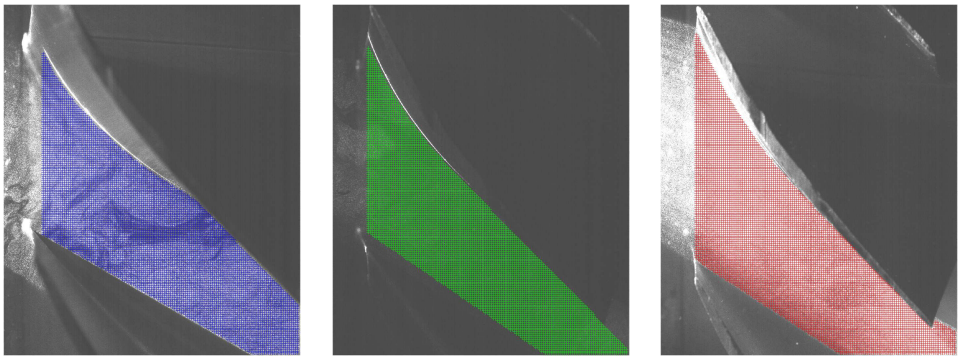


Fig. 3: Positioning of PIV planes in *CME2*, with respect to the blade height, close to the hub (**P1** at 18% of the blade's height), mid span (**P2** at 51% of the blade's height) and close to the tip (**P3** at 79% of the blade's height).



(a) (b) (c)

Fig. 4: PIV planes shown for (a) **P1**; (b) **P2**; (c) **P3**.

Globally, the measurement uncertainty of the PIV samples is estimated to be around 3 % (gathering together multiple error sources such as calibration, peak locking and other noise sources).
An additional uncertainty is related to the operating point of the compressor itself, as the working condition is set using a pair of pressure sensors (First Sensors BTE5000). First, the mass flow rate is measured using the static pressure drop of the converging section, assuming no velocity in the inlet settling chamber. Taking into account the various sources of error (temperature, etc.), the uncertainty on the mass flow is estimated to $\pm 0.06\,\mathrm{kg{\cdot}s^{-1}}$ (± 1 % of the design mass flow rate). Second, the pressure rise is directly logged by measuring the differential pressure between both sides of the stage. After calibration, the associated uncertainty is $\pm 12.5\,\mathrm{Pa}$, corresponding to ± 0.4 % of the design pressure rise.

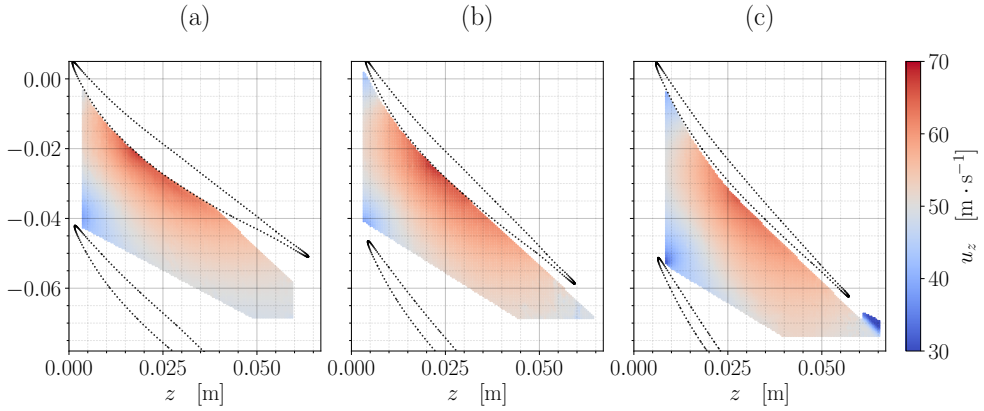


Fig. 5: Axial velocity u_z^{PIV} measurements obtained via PIV for (a) **P1**; (b) **P2**; (c) **P3**.

3 Numerical ingredients

This section includes the main numerical tools and methods used for this investigation. These include descriptions of the CFD solvers, of the DA strategies and of the ML models. Considering the large amount of information needed to comprehensively describe these tools, the main elements are presented in the core text of the article while detailed explanations are provided in Appendix A, Appendix B and Appendix C.

3.1 Dynamic equations

The current study uses finite-volume-based solvers [31] with discretization of the compressible Navier-Stokes equations for Newtonian fluids. The dynamic equations are written using index notation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_j)}{\partial x_j} = 0 \tag{1}$$

$$\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_i u_j + p\delta_{ij} - \tau_{ij}) = f_i \tag{2}$$

$$\frac{\partial (\rho E)}{\partial t} + \frac{\partial}{\partial x_j}(\rho E u_j + p u_j + q_j - u_i \tau_{ij}) = 0 \tag{3}$$

where Equation 1 represents the mass equation, Equation 2 is the momentum equation and finally Equation 3 is the energy equation. ρ represents the density, p is the pressure, $\mathbf{u}$ is the velocity, $\mathbf{f}$ is a volume forcing and E is the total energy. δ_{ij} is the Kronecker symbol. For a Newtonian fluid, the components of the viscous stress tensor $\bar{\bar{\boldsymbol{\tau}}}$ are expressed as:

$$\bar{\bar{\boldsymbol{\tau}}}_{ij} = 2\rho\nu_{\text{eff}}\bar{\bar{\mathbf{s}}}_{ij} \tag{4}$$

where ν_{eff} is the effective viscosity, which will be described in detail in the next paragraph. The components of the strain rate tensor $\bar{\bar{\mathbf{s}}}$ are defined as:

$$\bar{\bar{\mathbf{s}}}_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{1}{3}\frac{\partial u_k}{\partial x_k} \cdot \delta_{ij} \tag{5}$$

Finally, the components of the thermal flux $\mathbf{q}$ are expressed according to the Fourier law as $q_j = -\lambda\,(\partial T/\partial x_j)$ with λ the thermal conductivity coefficient and T the temperature.
The simulations in the present analysis are performed for Reynolds numbers corresponding to a fully turbulent behavior of the flow. For this reason, turbulence closures [11, 14] are employed to reduce computational costs. Using the Boussinesq approximation one has $\nu_{\text{eff}} = \nu + \nu_t$ where ν is the molecular kinematic viscosity and ν_t is the turbulent viscosity. For the simulations performed in this work, ν_t is obtained via the Spalart-Allmaras turbulence model [32]. This turbulence closure is a one-equation model formulated by the following transport equation:

$$\frac{\partial \tilde{\nu}}{\partial t} + \left(\frac{\partial \tilde{\nu}}{\partial x_j}\right) u_j = \underbrace{c_{b1}\,(1 - f_{t2})\,\tilde{S}\tilde{\nu}}_{\text{Production}} + \underbrace{\frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\nu + \tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right) + c_{b2}\left(\frac{\partial \tilde{\nu}}{\partial x_i}\right)^2\right]}_{\text{Diffusion}} - \underbrace{c_{w1} f_w \left(\frac{\tilde{\nu}}{\tilde{d}}\right)^2}_{\text{Destruction}} \tag{6}$$

where $\tilde{\nu}$ is a viscosity quantity used to compute the actual eddy kinematic viscosity ν_t:

$$\nu_t = \tilde{\nu}\, f_{v1} \tag{7}$$

The f_{v1} function is computed as:

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3} \qquad \chi = \frac{\tilde{\nu}}{\nu} \tag{8}$$

More details on the model constants prescribed for this analysis are provided in Appendix A. The previous numerical model is used to perform two distinct classes of simulations:

1. The first one includes classical steady RANS runs, in which the rotation of the rotor is accounted for using a Multiple Reference Frame (MRF). Details about this technique are reported in Wilhelm [33].
2. The simulations in the second class rely on a hybrid RANS-LES strategy referred to as Delayed Detached Eddy Simulations (DDES) [34]. These simulations use the same set of equations resolved for the RANS runs, but in this case the variable $\tilde{d}$ in Equation 6 is not constant and is calculated as:

$$\tilde{d} = d - \tilde{f}\,(\max(0, d - C_{\text{DES}}\Delta)) \tag{9}$$

where d is the distance from the closest wall, $C_{\text{DES}} = 0.65$ and $\Delta \equiv \max(\Delta x, \Delta y, \Delta z)$ is a function of the local grid resolution. Equation 9 requires the

calculation via empirical equations of two new variables $\tilde{f}$ and $\tilde{r}$:

$$\tilde{f} \equiv 1 - \tanh\left((8\tilde{r})^3\right) \; ; \; \tilde{r} = \frac{\nu_t + \nu}{\max\left(\sqrt{\left(\frac{\partial u_j}{\partial x_j}\right)^2}, 10^{-10}\right) \cdot \kappa^2 \tilde{d}^2} \tag{10}$$

The expression of the hybrid distance $\tilde{d}$ blends the model behavior between RANS and LES modeling. If the wall distance d increases, $\tilde{r} \to 0$ and $\tilde{f} \to 1$, thus the DDES model switches to LES mode. Conversely, if $d \to 0$, $\tilde{r} \to 1$ and $\tilde{f} \to 0$, which makes the DDES model fully RANS. DDES simulations are scale resolved in LES regions and are unsteady in time. For these cases, an Arbitrary Mesh Interface (AMI) method is used to control the interactions between the moving part of the computational grid (rotor region) and the static part of the grid (the rest of the physical domain). More details are provided in Carneiro et al. [35].

The different numerical simulations performed in the current study are realized using the C++ code OpenFOAM [36]. This open-source library is well known in academic studies thanks to its capacities in preprocessing, solving and postprocessing tasks. The current analysis requires the usage of two different solvers. A solver based on the SIMPLE algorithm [37] is used to solve steady RANS simulations, while a solver based on the PIMPLE algorithm [37, 38] is used to perform unsteady DDES runs. For the latter, time derivatives are discretized using a second-order backward scheme. Advection terms are discretized using a first-order upwind scheme for RANS simulations and a native LUST scheme for DDES calculations. The LUST method is a hybrid scheme combining second-order centered schemes (75%) with second-order upwind schemes (25%). For the viscous terms, second-order centered schemes are used for both families of simulations.

3.2 Data Assimilation and Ensemble Kalman Filter

Data Assimilation (DA) tools provide a mathematically rigorous framework to combine data from different sources of information, such as measurements from experiments and results of numerical simulations. One of the main advantages of these approaches is that, unlike several data-driven techniques, they are efficient in handling sparse data in space and time. Classical applications usually rely on *observation*, which is sampled using high-fidelity tools on limited parts of the domain for a restricted number of variables, and a *model*, which provides a quasi-continuous representation of the physics investigated but it is usually less accurate than the tools used to sample observation. The DA methods combine observation and model prediction to obtain a *state estimation*. Successful application provides physical states which are as accurate as observation while describing the whole domain of investigation. These applications have been extensively used in meteorological studies [18, 20] but their usage in fluid mechanics has seen a significant increase in the last two decades [19, 39–41]. Among notable applications, one can find works using variational techniques [42–47] as well as sequential methods [48–60]. The latter have shown very favorable features for the

investigation of unsteady, nonlinear problems, such as the high Reynolds flows here investigated.
Among the different sequential DA methods proposed in the literature, the stochastic Ensemble Kalman Filter (EnKF) [20] builds upon the fundamental principle of the classic Kalman Filter (KF) [61], but is tailored for high-dimensional and non-linear systems. Initially proposed by Evensen [22] for state estimation and later generalized for parameter optimization of the model [23], this methodology utilizes a population of n_e distinct realizations commonly referred to as *ensemble members*. The methodology, which is fully described in Appendix B, is developed in two steps:

1. The *forecast* step consists of the time advance of the numerical model between two available observations.
2. The *analysis* step includes numerical manipulations of the ensemble predicted state and the available observation to derive the state estimation.

While the generation of the state estimation for the EnKF requires computational resources that are significantly lower when compared with classical KF strategies, the generation of large databases using models describing millions of degrees of freedom (such as the solutions obtained via CFD) is still prohibitive for most realistic applications. A number of research works in the last years have offered solutions to reduce the computational burdens while keeping high accuracy, such as the Multigrid EnKF (MGEnKF) [60, 62, 63], the Multifidelity EnKF (MFEnKF) [24], or multi-level strategies enhanced by neural networks [27]. The MFEnKF relies on a higher-fidelity model, whose quantities are indicated via a suffix $\boldsymbol{\chi}$, and a computationally efficient lower-fidelity surrogate, identified via a suffix $\boldsymbol{\gamma}$. A variable $\boldsymbol{\gamma}$ is a control variate for a primary variable $\boldsymbol{\chi}$ if $\boldsymbol{\gamma}$ is strongly correlated with $\boldsymbol{\chi}$ and its expected value $\boldsymbol{\mu}_{\boldsymbol{\gamma}}$ is known *a priori*. By combining a few model realizations $\boldsymbol{\chi}$ and a larger number of runs $\boldsymbol{\gamma}$, the MFEnKF can obtain a state estimation whose accuracy is similar to the classical EnKF application. However, only a limited number of higher-fidelity model runs is required, therefore dramatically reducing the computational costs. The MFEnKF, which will be used in the present analysis, is comprehensively described in Appendix C.

4 Numerical investigation of the CME2 test case

Numerical simulations of the CME2 compressor are performed for the computational domain shown in Figure 6. One can see that z is the axial direction of the flow. The geometry is here simplified when compared with the case used in experiments, but previous analyses in the literature [64, 65] highlighted that these simplifications do not preclude the acquisition of physically meaningful results. Present analyses are performed for the operating condition presented in section 2, for which very small separation regions are observed. Therefore, the Spalart-Allmaras closure presented in subsection 3.1 has been selected for both RANS and DDES simulations. Two numerical grids have been prepared using the *Cadence AutoGrid* software, which is specifically designed to obtain exportable grids for the OpenFOAM format. The two grids, which encompass the physical domain shown in Figure 6 and are qualitatively shown in Figure 7, have the following features:

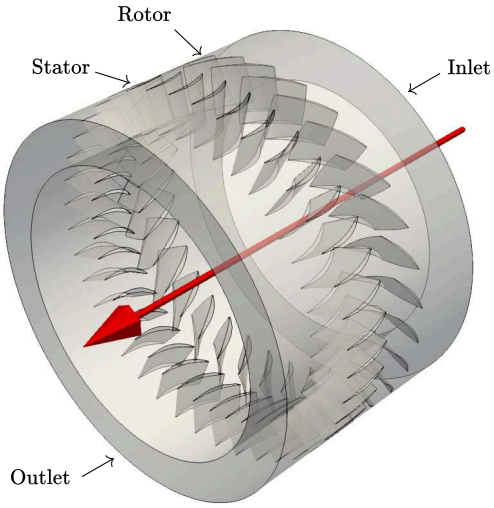
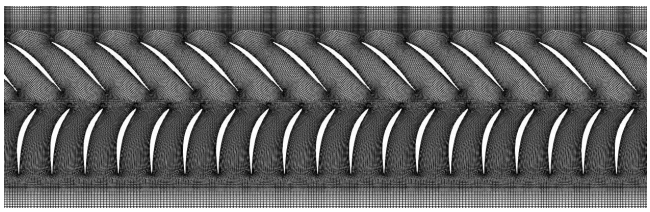
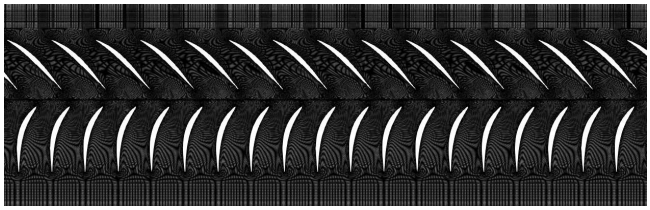


Fig. 6: Scheme of the computational domain used to perform numerical simulations of the flow inside the CME2 compressor. The red arrow represents the asymptotic axial direction of the flow.

1. The first grid **M1** is made of $\approx 5 \cdot 10^6$ cells. The resolution in the proximity of the wall is chosen to provide an adequate trade-off between computational costs and accuracy. An initial estimation of the size of the elements was performed via dimensional arguments and results were checked a posteriori after a few runs where modifications to boundary conditions were also performed. Calculations of the quantity $y^+ = y_{\text{cell}}/\delta_\nu$ showed that in the rotor region we have $y^+_{\text{rotor}} \in [1.67, 38]$. A part of the grid elements has its center in the viscous sublayer ($y^+ \leq 5$), providing an adequate resolution of the near-wall gradients. However, a maximum value of $y^+ = 38$ indicates that a part of the grid elements is within the buffer layer ($5 < y^+ < 30$). The mesh resolution is now briefly described, indicating the size of the grid elements in the axial direction Δ_z, in the radial direction Δ_r and in the azimuthal direction Δ_θ. The resolution of the grid elements in the rotor area is $\Delta_z \in [1.46 \cdot 10^{-2}, 3.7 \cdot 10^{-2}]\, c$, $\Delta_\theta \in [1.85 \cdot 10^{-2}, 3.35 \cdot 10^{-2}]\, c$ and $\Delta_r \in [1.55 \cdot 10^{-3}, 5 \cdot 10^{-2}]\, c$, where c is the average chord of the blade profiles. In the stator region, the wall resolution is $y^+_{\text{stator}} \in [7.6, 54]$. Here, since the minimum y^+ is 7.6, the grid does not fully resolve the viscous sublayer and the center of the near-wall cells is mainly located in the buffer layer, extending into the logarithmic zone for $y^+ = 54$. For the stator region, $\Delta_z \in [1.07 \cdot 10^{-2}, 6.7 \cdot 10^{-2}]\, c$, $\Delta_\theta \in [3.63 \cdot 10^{-2}, 4.87 \cdot 10^{-2}]\, c$ and $\Delta_r \in [3.43 \cdot 10^{-3}, 9.84 \cdot 10^{-2}]\, c$. The grid **M1** is specifically used for the RANS simulations and the MRF technique is here used to account for the movement of the rotor. Native OpenFOAM Spalding wall functions were used to improve near-wall accuracy of the calculations.
2. The second grid **M2** consists of $\approx 50 \cdot 10^6$ elements. For this second grid, we observe $y^+_{\text{rotor}} \in [0.5, 32]$, which indicates that the viscous sublayer is fully resolved over a part of the surface of the rotor blades. The upper bound at $y^+ = 32$ shows that



(a)



(b)

Fig. 7: Snapshot of the computational grids used for calculation. An azimuthal section for $r = 0.2475\,\mathrm{m}$ is shown for (a) the grid **M1** used for RANS calculations and (b) the grid **M2** used for DDES calculations.

the center of a part of the near-wall grid elements is located around the beginning of the logarithmic zone. The resolution of the grid elements in the rotor area is $\Delta_z \in [6.04 \cdot 10^{-3}, 2.14 \cdot 10^{-2}]\,c$, $\Delta_\theta \in [8 \cdot 10^{-3}, 1.28 \cdot 10^{-2}]\,c$ and $\Delta_r \in [1.49 \cdot 10^{-3}, 2.37 \cdot 10^{-2}]\,c$. In the stator region, $y^+_{\mathrm{stator}} \in [0.7, 21]$, therefore exhibiting similar to that of the rotor region. The size of the grid elements in the stator area is $\Delta_z \in [9.01 \cdot 10^{-3}, 1.92 \cdot 10^{-2}]\,c$, $\Delta_\theta = 9.1 \cdot 10^{-3}\,c$ and $\Delta_r \in [3.12 \cdot 10^{-3}, 3.32 \cdot 10^{-2}]\,c$. The grid **M2** is used to perform DDES simulations and the AMI tool permits for motion of the rotor region.

The boundary conditions, which are identical for both RANS and DDES simulations except for the rotor region, are now described. At the inlet, Dirichlet conditions for the total pressure $P_t = 101\,325\,\mathrm{Pa}$, the static temperature $T = 293.15\,\mathrm{K}$, and $\tilde{\nu} = 10^{-4}\,\mathrm{m}^2/\mathrm{s}$ are set using the native OpenFOAM boundary condition *pressureInletVelocity*. The shaft, casing, and stator blades are walls and they are represented using a *noSlip* condition for the velocity, a *zeroGradient* condition for the pressure and a *Spalding wall function* for the turbulent viscosity. For the steady RANS simulations using the MRF method, the rotor boundary condition for velocity is also set to *noSlip*. On the other hand, the unsteady DDES simulation using the AMI technique imposes a *movingWallVelocity* boundary condition to ensure mesh displacement in the rotor region. Finally, the outlet utilizes a custom static-pressure condition, referred to as *valve law*, which aims to replicate the mechanical behavior of the valve located at the outlet of the experimental setup (see section 2). This boundary condition has been previously used in the analyses by Rannou [66]. The static outlet pressure is defined

as:

$$P_{s_{\text{outlet}}} = P_{\text{ref}} + P_{i,0} \cdot \alpha \left(\frac{\dot{q}}{\dot{q}_{\text{ref}}} \right)^2 \tag{11}$$

where the stagnation pressure $P_{i,0} = 101\,325\,\text{Pa}$ and the reference pressure $P_{\text{ref}} = 0.8 \cdot P_{i,0}$. The reference mass flow rate is $\dot{q}_{\text{ref}} = 5.3\,\text{kg} \cdot \text{s}^{-1}$, and $\dot{q}$ is the current mass flow rate calculated at the outlet. Finally, the *valve law* is governed by the free coefficient α which defines operating conditions at a given mass flow rate. The nominal operating conditions are obtained when $(\dot{q}/\dot{q}_{\text{ref}}) = 1$ and $\alpha = 0.23$, which leads to $P_{s_{\text{outlet}}} = 104\,364.75\,\text{Pa}$.

RANS and DDES simulations have been initially run using the numerical setup previously described to assess their accuracy. First of all, a qualitative visualization of the axial velocity fields is shown in Figure 8. One can see that both RANS and DDES simulations are able to correctly represent the topology of the flow, which does not exhibit large separations as expected. For the RANS runs, one can see that the wake out of the rotor suddenly changes direction when entering the stator region. This effect is due to the MRF technique here employed, which is consistent with the steady flow representation. On the other hand, the AMI technique implemented for the DDES simulations complies with the scale-resolving features of the CFD solver. One can clearly see that the wake of the rotor is advected through the blades of the stator. Similar physical considerations can be drawn for the pressure p which is shown in Figure 8 (c) and (d), and for the effective viscosity normalized by the molecular viscosity ν_{eff}/ν fields displayed in Figure 8 (e) and (f).
The results obtained via RANS and DDES are now validated via the comparison of the predicted axial velocity u_z with the available experimental data in Figure 9 and Figure 10, respectively.
Figure 9 compares the time-averaged axial velocity $\bar{u}_z$ predicted by RANS with the PIV measurements. Columns (a), (b) and (c) respectively display the PIV data, the RANS prediction and the absolute value of the difference between the two, while rows (i), (ii) and (iii) correspond to **P1**, **P2** and **P3**, respectively. For the lower plane **P1** (row (i)), the discrepancy between the RANS prediction and the PIV data is mainly located in the bottom left-hand corner of the plane, which corresponds to the leading edge of the rotor blade located below the PIV plane. The maximum discrepancy reaches $12.65\,\text{m} \cdot \text{s}^{-1}$, with a mean absolute discrepancy of $3.52\,\text{m} \cdot \text{s}^{-1}$ over the plane. The RANS prediction on the mid-span plane **P2** (row (ii)) shows the best agreement with the PIV results: the maximum discrepancy, still located at the leading edge of the rotor blade, is $10.90\,\text{m} \cdot \text{s}^{-1}$, with a mean absolute discrepancy of $1.54\,\text{m} \cdot \text{s}^{-1}$. For the higher plane **P3** (row (iii)), the RANS/PIV discrepancy is mainly located at the leading edge of the lower rotor blade, but also at the trailing edge of the upper rotor blade, as observed in the right-hand corner of Figure 9 (iii,c). On this plane, the maximum discrepancy reaches $20.96\,\text{m} \cdot \text{s}^{-1}$, while the mean absolute discrepancy is $2.73\,\text{m} \cdot \text{s}^{-1}$. The discontinuity visible in Figure 9 (iii,b) and (iii,c) originates from the boundary condition, due to the height at which **P3** is located in the compressor.
Figure 10 presents the comparison between the mean axial velocity fields $\bar{u}_z$ obtained from DDES simulations and PIV measurements. Once again, columns (a), (b), and (c)

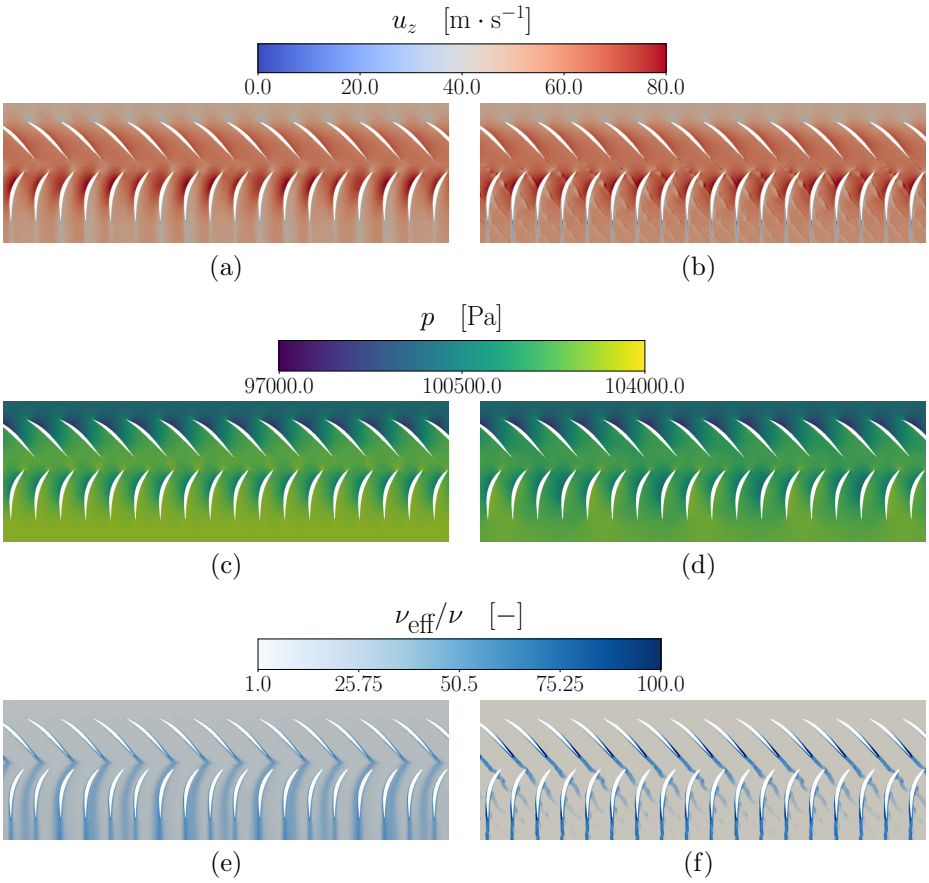


Fig. 8: Results for numerical simulations of the CME2 axial compressor. Top row: isocontours of the axial velocity u_z; Middle row: isocontours of the pressure p; Bottom row: isocontours of effective viscosity normalized by molecular viscosity ν_{eff}/ν. (a), (c) and (e) are reporting RANS solutions and (b), (d) and (f) are reporting DDES solutions. Visualizations are provided for a cylindrical surface for $r = 0.2475\,\text{m}$. Rotor is displayed at the top and stator at the bottom, flow is going from top to bottom.

illustrate the experimental PIV maps, the numerical DDES results, and the absolute difference between the two aforementioned approaches, respectively. Lines (i), (ii), and (iii) correspond to planes **P1**, **P2**, and **P3**. For **P1** on line (i), the major discrepancies between the DDES model and the experimental data are mainly concentrated in the lower-left area, as visible in sub-figure (i,c), which coincides with the leading edge of the rotor blading positioned below the plane. Locally, the maximum difference rises to $10.79\,\text{m} \cdot \text{s}^{-1}$, with an average discrepancy over the entire map (i,c) of $5.32\,\text{m} \cdot \text{s}^{-1}$. Regarding **P2**, which can be visualized on line (ii), the maximum error is once again located near the leading edge of the lower rotor blading with a value of $10.28\,\text{m} \cdot \text{s}^{-1}$, while the average error stands at $3.31\,\text{m} \cdot \text{s}^{-1}$. Finally, the analysis of **P3** on line (iii)

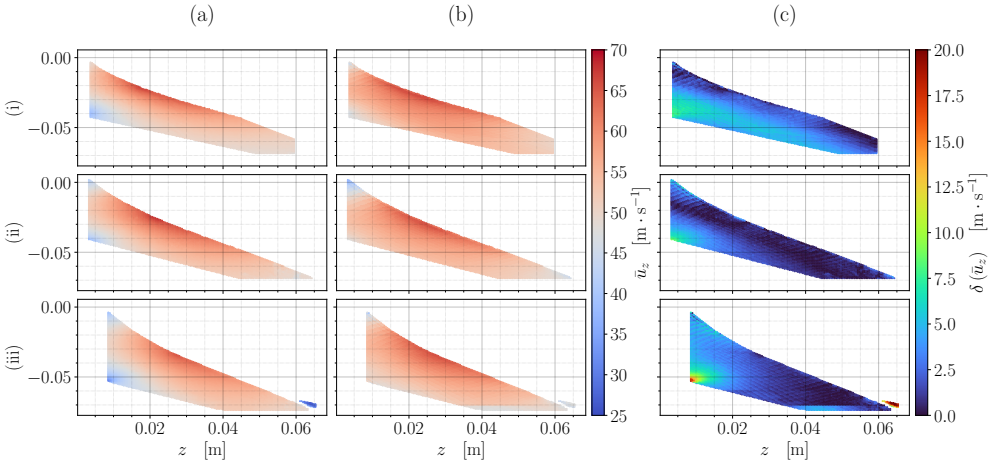


Fig. 9: Comparison of the RANS-predicted axial velocity $\bar{u}_z$ with the available experimental data. Column (a): PIV-measured axial velocity u_z^{PIV} data; column (b): RANS-predicted axial velocity $\bar{u}_z^{\mathrm{RANS}}$ data; column (c): absolute value of the difference between PIV and RANS axial velocities $\left|u_z^{\mathrm{PIV}} - u_z^{\mathrm{RANS}}\right|$. Row (i): results for **P1**; row (ii): results for **P2**; row (iii): results for **P3**.

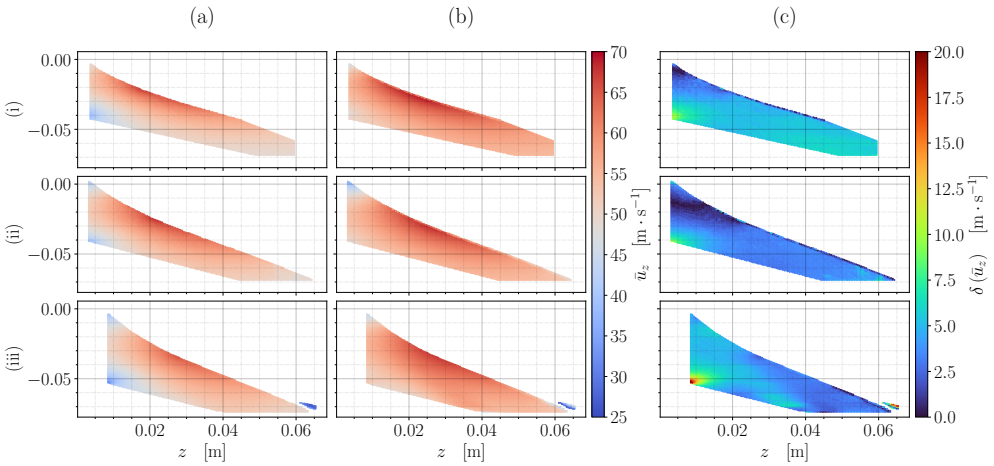


Fig. 10: Comparison of the DDES-predicted axial velocity $\bar{u}_z$ with the available experimental data. Column (a): PIV-measured axial velocity u_z^{PIV} data; column (b): time-averaged DDES-predicted axial velocity $\bar{u}_z^{\mathrm{DDES}}$ data; column (c): absolute value of the difference between PIV and DDES axial velocities $\left|u_z^{\mathrm{PIV}} - u_z^{\mathrm{DDES}}\right|$. Row (i): results for **P1**; row (ii): results for **P2**; row (iii): results for **P3**.

also reveals differences between the PIV and DDES fields, where less homogeneous disparities are noticeable in sub-figure (iii,c) compared to the other sub-figures in

column (c). With this configuration, the maximum discrepancy reaches a value of $21.75\,\mathrm{m}\cdot\mathrm{s}^{-1}$ at the leading edge of the lower profile, while maintaining an average discrepancy value of $4.66\,\mathrm{m}\cdot\mathrm{s}^{-1}$. This analysis shows that the discrepancy observed for the DDES and RANS results is similar and that on average the RANS run performs better in terms of pure accuracy. This finding is not completely surprising considering that the grid resolution for **M2** is on average two times more resolved than **M1** but the DDES simulation implements different techniques for the representation of the moving rotor and is unsteady, therefore introducing an additional level of complexity due to convergence of the time averages of the physical field. One interesting point is also that the cartography of the discrepancy with experimental data shows different patterns for the DDES and the RANS. This point will be exploited in the multifidelity DA application which is presented in the next section.

5 Data Assimilation: application of the EnKF to the analysis of the CME2 test case

The EnKF is here used to combine the flow prediction obtained with the CFD solvers and the available experimental data. More precisely, the multifidelity framework provided by the MFEnKF is used to obtain a state estimation combining results from several RANS runs, a smaller number of DDES simulations and the PIV measurements. The state estimation is here obtained via observation of the time-averaged axial velocity $\bar{u}_z$. This flow field is naturally obtained as a result of RANS simulation and it has been calculated performing phase average from instantaneous flow fields for both the DDES runs and the experiments. In the following, the architecture of the DA procedure is detailed.

5.1 Setup of MFEnKF strategy

The MFEnKF strategy relies on the following elements:

1. *Observations* are sampled from the PIV maps of the phase-averaged axial velocity $\bar{u}_z$ obtained during the experimental campaign. From the 8 180, 6 944, and 8 090 points respectively available in planes **P1**, **P2**, and **P3**, 1 000 points are randomly selected from each plane, as shown in Figure 11. This selection has been performed to test the capabilities of the DA algorithm to reconstruct an accurate state estimation even with limited available data. For each plane, we thus have:

$$\boldsymbol{y} = \left(\bar{u}_{z_0}, \bar{u}_{z_1}, \ldots, \bar{u}_{z_{n_o-1}}\right) \in \mathbb{R}^{n_o} \tag{12}$$

   Considering that a stochastic version of the filter is used, observation is perturbed following a truncated normal law defined as:

$$\mathcal{X} \sim \mathcal{N}\left(\mu, \sigma^2\right) : \mathcal{X} \in [\mu - 3\sigma, \mu + 3\sigma] \tag{13}$$

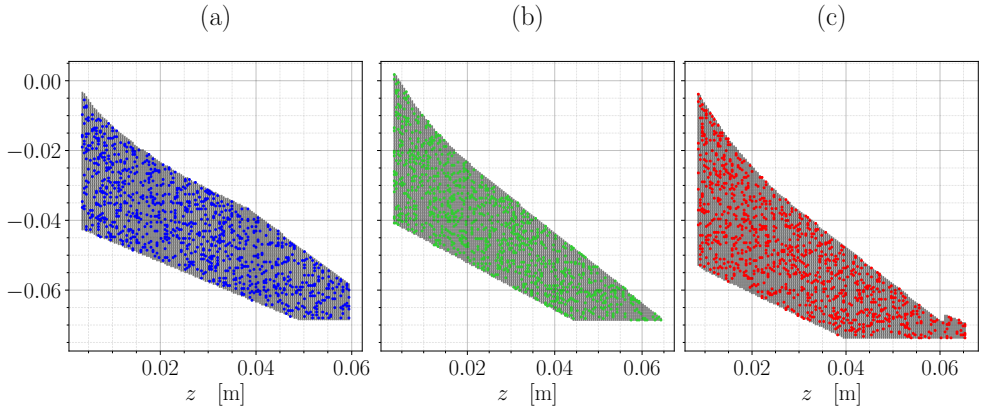


Fig. 11: Blue, green and red dots represent the selected observation points for MFEnKF among available data for (a) **P1**; (b) **P2**; (c) **P3**.

with the mean value $\mu = \boldsymbol{y}$ and its standard deviation defined as $\sigma = 0.03\,\boldsymbol{y}$. The selected value corresponds to the global measurement uncertainty experimentally quantified for the PIV samples.

2. An unsteady, computationally expensive *model* is used, which is based on DDES simulation using the AMI technique with the `rhoPimpleFoam` unsteady solver. Five DDES ensemble members are performed to this purpose. The variability of the prior states for these simulations is obtained by imposing slightly different outlet conditions. More precisely, the parameter α in Equation 11 is defined as $\alpha \sim \mathcal{N}\left(0.27, 0.02^2\right)$. This variability is essential to obtain different physical states to improve the convergence of the filter. Each DDES simulation has been initialized with an already-converged RANS field and run for a total solver time of $0.005\,t$, which corresponds to 50 000 iterations using a time step of $\delta t = 1 \cdot 10^{-7}\,t$, which represents 8/30 of a complete rotation of the rotor. The phase averages have been performed after a time of $0.0025\,t$, which was necessary to eliminate the effects of initial conditions.
3. A steady, computationally inexpensive *model* is run via RANS simulation with the MRF method. These simulations are performed using the steady-state solver `rhoSimpleFoam`. Thirty ensemble members are performed with this model. Five are the *control* members (i.e. the ones which are connected with the DDES realizations) while the other 25 are the *ancillary* members. Therefore, the total number of RANS+DDES ensemble members is $n_e = 35$. Each RANS simulation has been initialized with an already-converged RANS field, and run for a total of 6 250 iterations. The variability between the ensemble members is obtained via variation of the parameter α in Equation 11, defined as $\alpha \sim \mathcal{N}\left(0.27, 0.02^2\right)$.
4. The coupling between the model and observation is performed using the numerical library CONES [54, 67]. The hyperlocalization technique recently proposed by Villanueva et al. [57] for the EnKF is here employed. The idea here is that the state update produced by the EnKF is not calculated and applied on the complete physical domain of the simulations, but it is restricted to a determined physical region

around the sensors. This procedure allows filtering out sampling errors due to the limited number of ensemble members as well as to reduce the computational costs required. Therefore, the state estimation is performed only for three regions which are created around planes **P1**, **P2** and **P3**. A qualitative representation of such DA zones when compared with the experimental PIV planes is shown in Figure 12. In the zones determined for DA purposes, the state vector is built considering the three Cartesian components of the velocity field for the selected subset of grid elements:

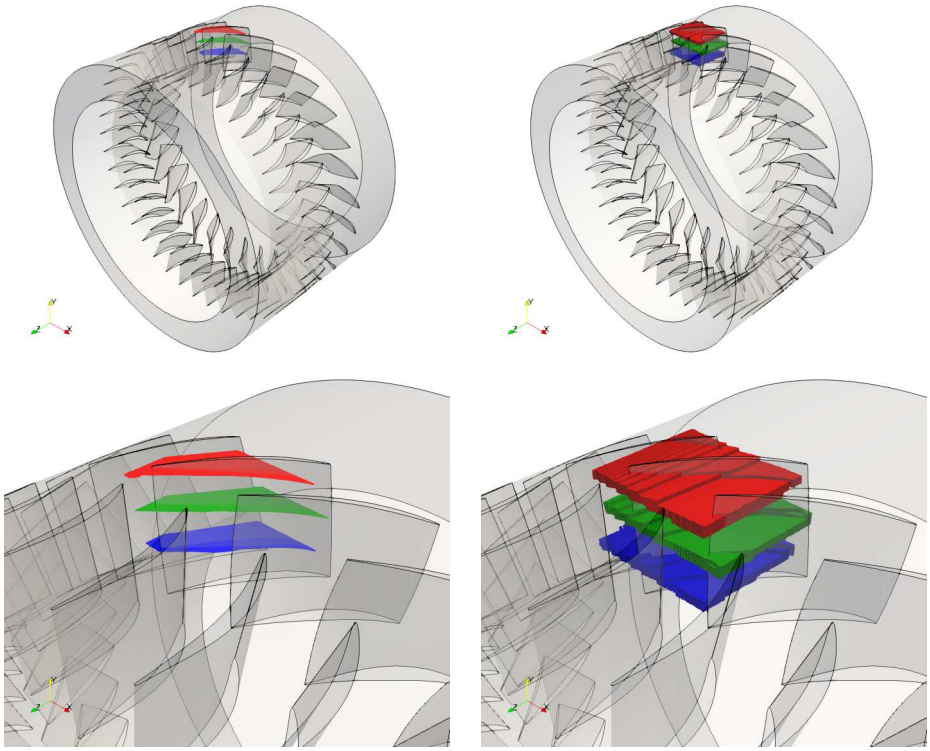
$$\boldsymbol{x} = \left(u_{x_1}, \ldots, u_{x_{n_\mathrm{C}}}, u_{y_1}, \ldots, u_{y_{n_\mathrm{C}}}, u_{z_1}, \ldots, u_{z_{n_\mathrm{C}}}\right) \in \mathbb{R}^{3 \times n_\mathrm{C}} \tag{14}$$

where n_C represents the number of cells in the aforementioned clipping boxes. It must be stressed that, for the grid M2 used for DDES calculations, $n_{\mathrm{C},\mathbf{P1}} = 38\,462$, $n_{\mathrm{C},\mathbf{P2}} = 44\,072$ and $n_{\mathrm{C},\mathbf{P3}} = 59\,113$. In this case, one can see that the total information produced by the DA state estimation is significantly larger than the availability of the experimental data, even if hyperlocalization is used. This point is important because it indicates that the DA approach generates a larger amount of high fidelity data starting from a limited amount of observation data, therefore providing more suitable conditions for the training of machine learning models.

Finally, the PIV, RANS and DDES data were obtained by phase, iteration and time averaging, respectively. For clarity, all three are hereafter referred to as *mean* data.

5.2 State estimation using the MFEnKF

Results obtained via the MFEnKF are now discussed. The results are presented for each plane and box showcased in Figure 12. After a single analysis phase, the MFEnKF results are displayed in Figure 13 for **P1**, corresponding to the blue plane and box in Figure 12. Figure 13 (a,i) shows the axial velocity u_z^{PIV} obtained from the PIV measurements, (a,ii) displays the DDES-predicted time-averaged axial velocity $\bar{u}_z^{\mathrm{DDES}}$, (a,iii) displays the RANS-predicted mean axial velocity $\bar{u}_z^{\mathrm{RANS}}$, while (a,iv) exhibits the ensemble average of the updated axial velocity from the MFEnKF analysis $\langle \bar{u}_z^a \rangle$, where the $\langle \cdot \rangle$ denotes the average of any quantity over the n_e ensemble members. It is easily observable that the velocity field in Figure 13 (a,i) is in better agreement with Figure 13 (a,iv) than with Figure 13 (a,ii) and (a,iii). More precisely, the improvement is qualitatively observed on the bottom left-hand corner of the (a,iv) field, which corresponds to the leading edge of the rotor blade below **P1**, where the axial velocity has been reduced to match the PIV data. Figure 13 (b,ii) displays the magnitude of the difference between the axial velocities obtained by PIV and DDES prediction $\left|u_z^{\mathrm{PIV}} - \bar{u}_z^{\mathrm{DDES}}\right|$, (b,iii) shows the magnitude of the difference between the axial velocities obtained by PIV and RANS prediction $\left|u_z^{\mathrm{PIV}} - \bar{u}_z^{\mathrm{RANS}}\right|$, while Figure 13 (b,iv) shows the magnitude of the difference between the axial velocities obtained by PIV and MFEnKF ensemble average update $\left|u_z^{\mathrm{PIV}} - \langle \bar{u}_z^a \rangle\right|$. The DA updated field shows a lower discrepancy with PIV data than the DDES and RANS-predicted fields, which



(a) (b)

Fig. 12: (a) column: position of **P1**, **P2** and **P3** PIV planes in the CME2 compressor, respectively colored in blue, green and red; (b) column: position of the DA hyperlocalization clipping boxes corresponding to the three PIV planes earlier mentioned, with identical color matching.

is measured via the L2 norm error $\bar{\varepsilon}$ which is defined as:

$$\bar{\varepsilon} = \frac{1}{n_\mathrm{p}} \sum \frac{|\boldsymbol{x}_\mathrm{PIV} - \boldsymbol{x}_\mathrm{CFD}|}{\boldsymbol{x}_\mathrm{PIV}} \tag{15}$$

in which n_p represents the total number of points in the PIV planes, $\boldsymbol{x}_\mathrm{CFD}$ represents data obtained using a CFD solver and $\boldsymbol{x}_\mathrm{PIV}$ data from PIV measurements. The results, provided in Table 1, show that $\bar{\varepsilon}_\mathrm{DDES} = 10.2\,\%$, $\bar{\varepsilon}_\mathrm{RANS} = 6.87\,\%$ and $\bar{\varepsilon}_\mathrm{MFEnKF} = 2.59\,\%$, therefore indicating that the DA procedure successfully improves flow prediction.
Similarly, the results of the MFEnKF analysis for **P2** are presented in Figure 14, which corresponds to the green plane and box in Figure 12. Figure 14 (a,i) shows the axial velocity u_z obtained from PIV measurements, (a,ii) displays the time-averaged axial velocity predicted by DDES $\bar{u}_z^\mathrm{DDES}$, (a,iii) displays the mean axial velocity predicted by RANS $\bar{u}_z^\mathrm{RANS}$, while (a,iv) presents the ensemble mean of the updated axial velocity

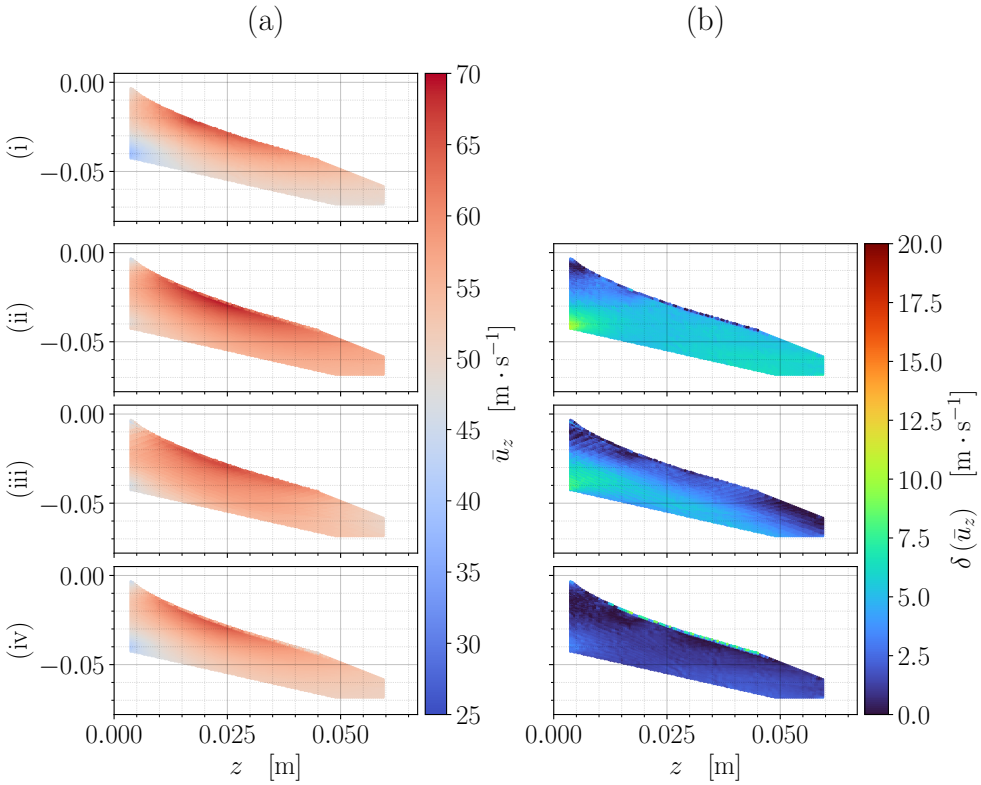


Fig. 13: Results obtained via DA using the MFEnKF for **P1**. Column (a): Maps of the time-averaged axial velocity $\bar{u}_z$. Line (i) Reference experimental data (PIV); line (ii) Initial numerical prediction (DDES); line (iii) Initial numerical prediction (RANS); line (iv) Updated field after data assimilation (ensemble mean of the MFEnKF update $\langle \bar{u}_z^a \rangle$); column (b): maps of the absolute discrepancy $\delta(\bar{u}_z) = \left| u_z^{\mathrm{PIV}} - \bar{u}_z \right|$ relative to the reference data. From top to bottom, the image displays the DDES model error $\delta(\bar{u}_z^{\mathrm{DDES}})$, the RANS model error $\delta(\bar{u}_z^{\mathrm{RANS}})$, and the bottom image displays the residual error of the model corrected by the MFEnKF algorithm ($\left| u_z^{\mathrm{PIV}} - \langle \bar{u}_z^a \rangle \right|$).

from the MFEnKF analysis $\langle \bar{u}_z^a \rangle$. Just as for **P1**, it is easily observed that the velocity field in Figure 14 (a,i) corresponding to PIV data is in much better agreement with the results from the MFEnKF in Figure 14 (a,iv) than with the results from baseline DDES and RANS in Figure 14 (a,ii) and (a,iii). More precisely, the improvement is qualitatively observable around the leading edge of the lower rotor blade profile, where the inaccuracies of the initial prediction have been corrected to fit the PIV data more accurately. From a quantitative standpoint, Figure 14 (b,ii) displays the magnitude of the difference between the axial velocities obtained by PIV and by the DDES prediction $\left| u_z^{\mathrm{PIV}} - \bar{u}_z^{\mathrm{DDES}} \right|$, (b,iii) shows the magnitude of the difference between the axial velocities obtained by PIV and by the RANS prediction $\left| u_z^{\mathrm{PIV}} - \bar{u}_z^{\mathrm{RANS}} \right|$, while Figure 14 (b,iv) shows the magnitude of the difference between the axial velocities

obtained by PIV and the updated ensemble mean from the MFEnKF analysis $\delta\left(\bar{u}_z\right) = \left|u_z^{\mathrm{PIV}} - \langle \bar{u}_z^a \rangle\right|$. It is then clearly noticeable that the updated field exhibits a lower residual error than the fields predicted by DDES and RANS, with the respective average relative difference $\bar{\varepsilon}$ decreasing from $\bar{\varepsilon}_{\mathrm{DDES}} = 6.21\,\%$ and $\bar{\varepsilon}_{\mathrm{RANS}} = 2.91\,\%$ down to $\bar{\varepsilon}_{\mathrm{MFEnKF}} = 1.81\,\%$.

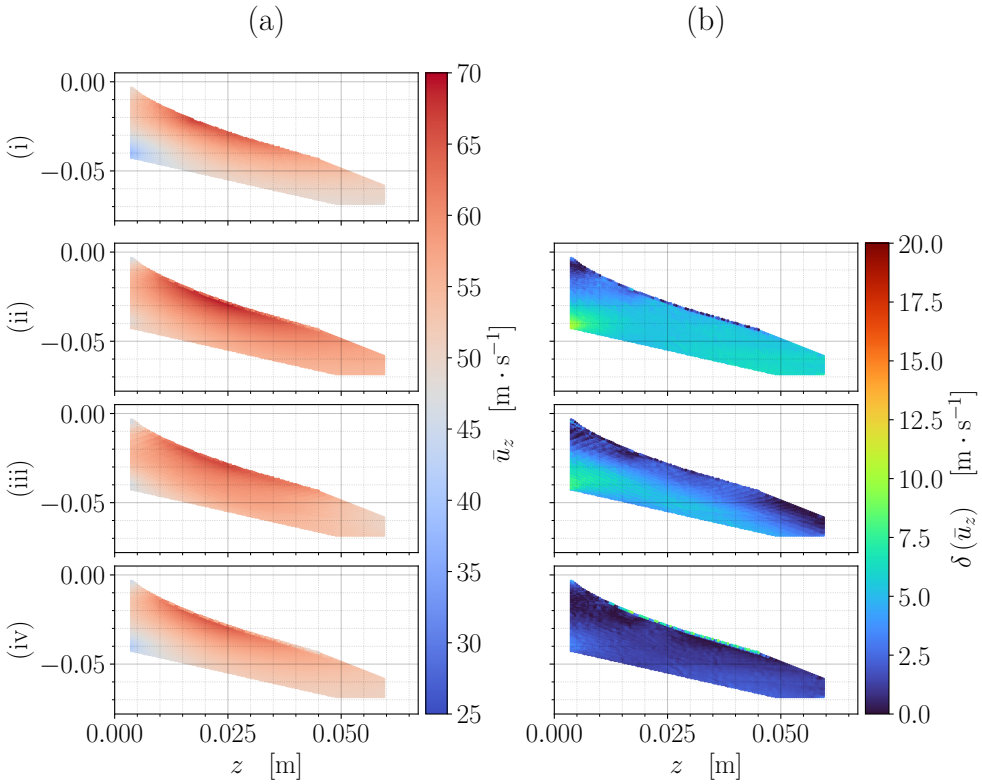


Fig. 14: Results obtained via DA using the MFEnKF for **P2**. Column (a): Maps of the time-averaged axial velocity $\bar{u}_z$. Line (i) Reference experimental data (PIV); line (ii) Initial numerical prediction (DDES); line (iii) Initial numerical prediction (RANS); line (iv) Updated field after data assimilation (ensemble mean of the MFEnKF update $\langle \bar{u}_z^a \rangle$); column (b): maps of the absolute discrepancy $\delta(\bar{u}_z) = \left|u_z^{\mathrm{PIV}} - \bar{u}_z\right|$ relative to the reference data. From top to bottom, the image displays the DDES model error $\delta(\bar{u}_z^{\mathrm{DDES}})$, the RANS model error $\delta(\bar{u}_z^{\mathrm{RANS}})$, and the bottom image displays the residual error of the model corrected by the MFEnKF algorithm ($\left|u_z^{\mathrm{PIV}} - \langle \bar{u}_z^a \rangle\right|$).

Finally, following the same procedure as for the previous planes, MFEnKF results for **P3** are presented in Figure 15, corresponding to the red plane and box in Figure 12. Here too, it is observed that the velocity field in Figure 15 (a,i) is in better agreement with Figure 15 (a,iv) than with Figure 15 (a,ii) and (a,iii). More precisely, the

improvement is qualitatively observable at the leading edge of the lower rotor blading, where the axial velocity has been corrected to adjust to the PIV data. It is clear that the updated field exhibits a lower residual error than the fields initially predicted by DDES and RANS, with the respective average relative difference $\bar{\varepsilon}$ dropping from 9.09 % and 5.55 % down to 3.81 %.

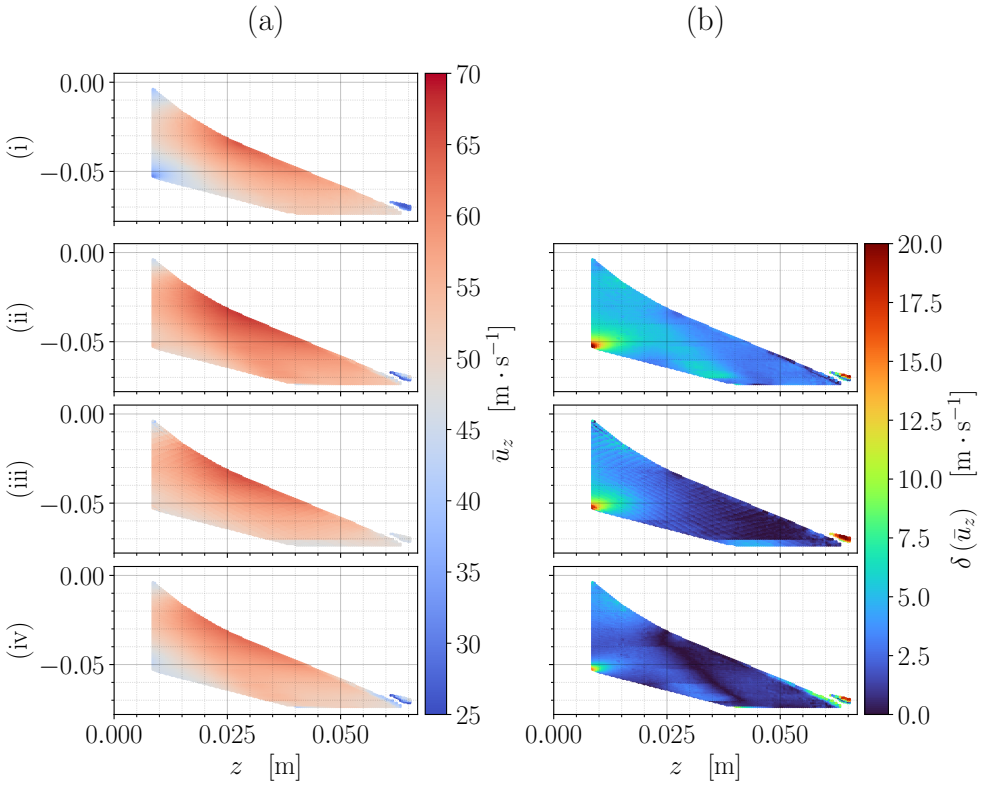


Fig. 15: Results obtained via DA using the MFEnKF for **P3**. Column (a): Maps of the time-averaged axial velocity $\bar{u}_z$. Line (i) Reference experimental data (PIV); line (ii) Initial numerical prediction (DDES); line (iii) Initial numerical prediction (RANS); line (iv) Updated field after data assimilation (ensemble mean of the MFEnKF update $\langle \bar{u}_z^a \rangle$); column (b): maps of the absolute discrepancy $\delta(\bar{u}_z) = \left| u_z^{\mathrm{PIV}} - \bar{u}_z \right|$ relative to the reference data. From top to bottom, the image displays the DDES model error $\delta(\bar{u}_z^{\mathrm{DDES}})$, the RANS model error $\delta(\bar{u}_z^{\mathrm{RANS}})$, and the bottom image displays the residual error of the model corrected by the MFEnKF algorithm ($\left| u_z^{\mathrm{PIV}} - \langle \bar{u}_z^a \rangle \right|$).

To summarize, the application of the MFEnKF data assimilation (DA) algorithm, coupled with the physical hyperlocalization technique, successfully and effectively corrected the errors of the DDES and RANS simulations. The results of the data

assimilation procedure are gathered in Table 1. By integrating numerical and experimental (PIV) data, a continuous flow field of near-experimental fidelity ($\langle \bar{u}_z^a \rangle$) was generated. However, the DA procedure requires available data for the calibration.

	P1	**P2**	**P3**
DDES	10.2 %	6.21 %	9.09 %
RANS	6.87 %	2.91 %	5.55 %
MFEnKF	2.59 %	1.81 %	3.81 %
ML-CR1	3.08 %	4.15 %	4.78 %
ML-CR2	4.81 %	2.65 %	5.17 %
ML-CR3	9.49 %	4.75 %	7.66 %
MoE	4.17 %	2.78 %	3.84 %

Table 1: Comparison of average discrepancies with PIV data obtained with the DDES and RANS simulations, the MFEnKF data assimilation, and the four machine learning models investigated. The average discrepancies are measured on the three planes where PIV measurements are available.

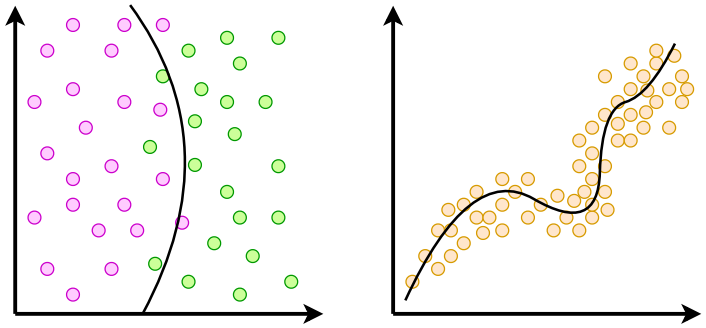
The next challenge to be overcome is to obtain turbulence models which can automatically include and use the experimental information that was available for the DA procedure. To this purpose, generative tools from Machine Learning will be used to locally correct the performance of the turbulence models. The expanded data produced by the MFEnKF from the initial PIV measurements provide efficient training conditions for a machine learning approach, therefore setting ideal conditions to transfer the DA improved prediction to black box tools. The following section will thus detail the development of an Artificial Neural Network (ANN) capable of learning and predicting this correction law, allowing it to be applied *a priori* and autonomously to future simulations.

6 Enhancement of turbulence models via Artificial Neural Networks

6.1 Introduction

In this section, the development of an artificial neural network is described, with the objective to mimic the corrections obtained with the Multi-Fidelity Ensemble Kalman Filter. The objective here is to obtain volume forcing source terms to be directly integrated in the Navier–Stokes equations, whose purpose is to locally correct the Spalart-Allmaras model used for turbulence closure.
Usually referred to as ANN, these models are sequences of operations designed to perform various tasks through machine learning [68], leveraging a function approximation $\mathcal{F}$ whose objective is to mathematically connect input and output data. This type of learning is ranked as *supervised learning*, as the input and output data are known *a*

priori [69]. Supervised learning is generally used for two main purposes: classification problems and regression problems.



(a) (b)

Fig. 16: Diagram illustrating an example of supervised learning methodology. (a) Classification principle; (b) Regression principle.

On the one hand, classification involves predicting a category, or a *label*, from a finite set of predefined categories, as illustrated in Figure 16 (a). There are many algorithms for solving a classification problem, including Support Vector Machines (SVMs), decision trees and neural networks.
On the other hand, regression involves predicting a continuous numerical value based on a set of input variables, as shown in Figure 16 (b). Unlike classification, regression establishes a mathematical relationship between the input variables and the expected output. Being quantitative in nature, this output is often referred to as the *target* and can therefore take on an infinite number of possible numerical values. The applications of regression are also numerous and span a wide range of fields. In industry, it can be used to estimate the lifespan of a mechanical component [70] or to predict energy consumption [71].
In the present section, the challenge of mimicking a corrective forcing term implies that ANNs are going to be used as **regression** methods rather than classification methods.

6.2 Basic neural network architecture and training

The architecture of the Neural Network is now described. The current network is composed of an input layer, followed by 4 hidden layers and finally an output layer. From Table 2, the number of trainable parameters is equal to 8 675.
Activation functions introduce non-linearity into the network. Without them, stacking multiple linear layers would be only able to represent a linear transformation of the input, regardless of the depth of the network. Thus, the introduction of activation functions allows the development of non-linear relationships between inputs and outputs, which is here essential for CFD applications. The ReLU activation function is defined as $f_{\mathrm{ReLU}}(x) = \max(0, x)$ [72].

Layers	Output shape	Number of parameters	Activation function
Input	6	0	–
Hidden 1	32	224	ReLU
Hidden 2	64	2 112	ReLU
Hidden 3	64	4 160	ReLU
Hidden 4	32	2 080	ReLU
Output	3	99	Linear

Table 2: Description of the structure of the regression ANN model

The utilized datasets are now described. For each of the clipping boxes displayed in Figure 12, **input** data can be described as a vector $\mathbf{z}_{\text{input}}$ such that:

$$\mathbf{z}_{\text{input}} = \left(\bar{u}_x, \bar{u}_y, \bar{u}_z, \mathcal{S}, \mathcal{W}, \tilde{\nu}\right)^\top \tag{16}$$

where $(\bar{u}_x, \bar{u}_y, \bar{u}_z)$ represents the three components of the velocity in a Cartesian framework, $\mathcal{S}$ is the magnitude of the symmetric part of the velocity gradient tensor defined as $\mathcal{S} = \left|\frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^\top\right)\right|$, $\mathcal{W}$ is the magnitude of the antisymmetric part of the velocity gradient tensor expressed as $\mathcal{W} = \left|\frac{1}{2}\left(\nabla\mathbf{u} - \nabla\mathbf{u}^\top\right)\right|$ and $\tilde{\nu}$ is the viscosity-like variable used in the Spalart-Allmaras turbulence model.
The **output** data is described as a vector $\mathbf{z}_{\text{output}}$ with three components such that:

$$\mathbf{z}_{\text{output}} = \mathbf{f}^{\mathcal{R}} = \left(f^{\mathcal{R}}_{\bar{u}_x}, f^{\mathcal{R}}_{\bar{u}_y}, f^{\mathcal{R}}_{\bar{u}_z}\right)^\top \tag{17}$$

The $\mathbf{f}^{\mathcal{R}}$ vector measures the discrepancy between the flow prediction obtained by the CFD model forecast and the DA state estimation. This term can be written as:

$$\mathbf{f}^{\mathcal{R}} = \mathbf{f}^{\mathcal{R}^a} - \mathbf{f}^{\mathcal{R}^f} \tag{18}$$

with

$$\mathbf{f}^{\mathcal{R}^f} = \rho\left(\bar{\mathbf{u}}^f \cdot \nabla\right)\bar{\mathbf{u}}^f + \nabla p - (\mu + \mu_t)\nabla^2\bar{\mathbf{u}}^f \tag{19}$$

$$\mathbf{f}^{\mathcal{R}^a} = \rho\left(\bar{\mathbf{u}}^a \cdot \nabla\right)\bar{\mathbf{u}}^a + \nabla p - (\mu + \mu_t)\nabla^2\bar{\mathbf{u}}^a \tag{20}$$

where the f and a superscripts respectively stand for the fields produced during the *forecast* and the *analysis* steps in the MFEnKF algorithm. The term $\mathbf{f}^{\mathcal{R}}$ can be seen as an additional volume source term to be integrated in the discretized form of the Navier-Stokes equations.

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho u_i u_j + p\delta_{ij} - \tau_{ij}\right) = f_i^{\mathcal{R}} \tag{21}$$

The idea, initially investigated by Villiers et al. [25], is that the inclusion of this term can lead the numerical model to naturally produce the flow field obtained via the DA

state estimation instead of the forecast result, therefore improving the accuracy of the underlying numerical model.
In order to proceed with an efficient training process and obtain a robust model, pre-processing is performed to eliminate outliers from the dataset. The InterQuartile Range method (IQR) [73] is used to prevent the training process from focusing on a small number of data points which could affect the network's parameter optimization. For any (i, j) entry from dataset z. The IQR method reads:

$$z_{i,j} \in \text{Clean Set} \iff \forall j : (Q_{1,j} - r \cdot IQR_j) \leq z_{i,j} \leq (Q_{3,j} + r \cdot IQR_j) \tag{22}$$

where the first and third quartiles, Q_1 and Q_3 , denote the values below which 25 % and 75 % of the target values fall. $IQR = Q_3 - Q_1$ is the interquartile range and j indexes the features from the dataset. r is the spread to which the outliers will be identified. For this work, Equation 22 is applied to the target vector $\mathbf{z}_{\text{output}}$, i.e. the output vector containing the forcing vector to be regressed, using a spread value equal to $r = 10$ to exclude strong outlying data. The IQR enforces target datasets to keep coherent values, ensuring a strong trained model. Data is split in three subsets, namely the train, validation and test set with a random split respectively corresponding to 80 %, 10 % and 10 % of the dataset. The validation set provides feedback during the training process, while the test set offers a final and independent performance assessment after the completed training. The model is compiled using the Mean Square Error (MSE) loss function and the Adam optimizer [74]. The initial learning rate is set to 10^{-4}, with an automatic reduction checkpoint down to 10^{-9} if the validation loss metric during training is not reducing for 10 consecutive epochs. In that same way, an automatic stop checkpoint is set to prevent training from continuing if relevant metrics do not improve anymore. In particular, the criterion here set to trigger the stop checkpoint is that the validation loss metric does not improve for more than 20 consecutive epochs. Finally, the whole procedure is implemented with the Python TensorFlow [75] and Keras [76] packages. The training algorithm is repeated three times, using data for each clipping box previously displayed in Figure 12. The obtained corrective models are referred to as CR1, CR2 and CR3 as they use DA data obtained with samples from the planes **P1**, **P2** and **P3**, respectively.
The accuracy of these models is first assessed via the analysis of the parity plots comparing the predicted forcing components $f^{\mathcal{R}}_{u_x}$, $f^{\mathcal{R}}_{u_y}$ and $f^{\mathcal{R}}_{u_z}$ against their ground truth values. CR1 training results obtained using data from **P1** are shown in Figure 17. The dashed red lines represent the $1:1$ identity, which indicates a perfect agreement between ground truth and predictions. The top row (blue dots) represents the validation data, while the bottom row (green squares) represents the test data. For each subplot, the coefficient of determination R^2 [77] is calculated to quantify the quality of the regression. This coefficient is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (z_i - \hat{z}_i)^2}{\sum_{i=1}^{N} (z_i - \overline{z}_i)^2} \tag{23}$$

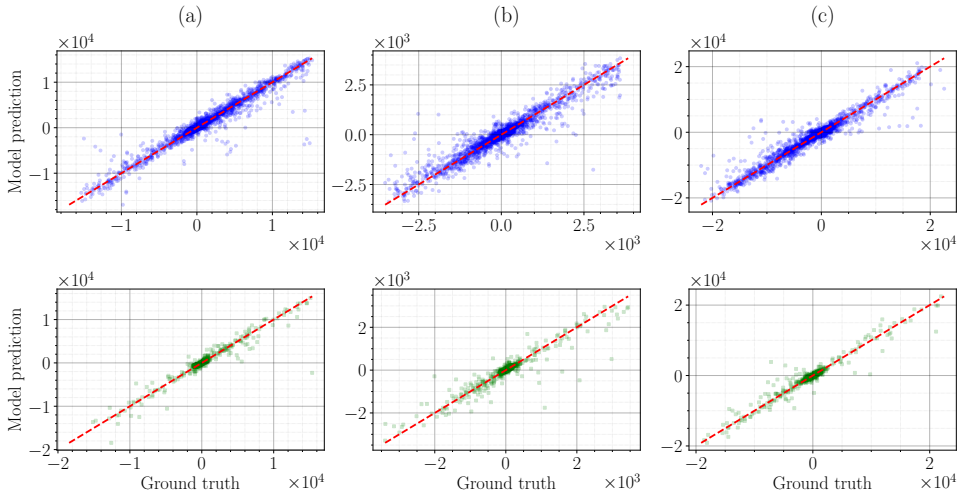


Fig. 17: Parity plots corresponding to the CR1 neural network training for **P1**. Column (a) corresponds to $f_{u_x}^{\mathcal{R}}$; column (b) corresponds to $f_{u_y}^{\mathcal{R}}$; column (c) corresponds to $f_{u_z}^{\mathcal{R}}$. Top row with blue dots represents the validation data; bottom row with green squares represents the test data.

where z_i are the ground-truth values, $\hat{z}_i$ are the predicted values and $\overline{z}_i$ is the average of all the ground-truth values. An R^1 of 1 score corresponds to a perfect regression fit, while an R^2 of 0 score indicates that the model is not more accurate than predicting the average value of the dataset.

The parity plots in Figure 17 show an encouraging level of convergence of the regression models for the three forcing components. More precisely, the model for $f_{u_z}^{\mathcal{R}}$ shown in Figure 17 column (c), achieves an R^2 score of 0.93 for the test set, indicating that the main flow features are captured with high accuracy. This component of the volume forcing is integrated in the streamwise component of the discretized equations. The component of the forcing vector $f_{u_x}^{\mathcal{R}}$ is also well predicted as an R^2 score of 0.91 is reached for the test dataset. A slight performance decrease is observed for the $f_{u_y}^{\mathcal{R}}$ component of the forcing vector, with an R^2 score of 0.87 for the test set. However, this score still represents a strong correlation between ground-truth and predictions. For all the forcing vector components, the close proximity between validation and test metrics confirms the generalization of the model on unseen data and it indicates that the model does not suffer from overfitting.

6.3 Integration of the Neural Network based corrections in the RANS solver

The usage of black-box correction terms based on neural networks is now described. The integration of the pre-trained models with OpenFOAM's solvers is realized through a Multiple Program, Multiple Data (MPMD) [78] architecture, which allows

the simultaneous execution of distinct computational tasks. Rather than directly embedding the Machine Learning libraries, which are developed with Python, into the C++ OpenFOAM solvers, the coupling strategy relies on the Message Passing Interface (MPI) [79] routines to bridge the two environments. The connection between the two environments is schematized in Figure 18.

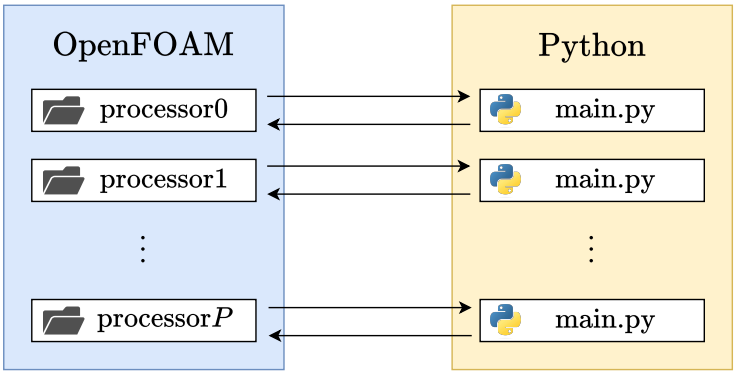


Fig. 18: Scheme of the coupling strategy between the CFD solver and the ML tools.

The machine learning volume forces are integrated within the resolution of the momentum equation of the RANS solvers. The reason is that the models are trained with mean data, which is a constraint associated with the available PIV samples. In order to develop a model suitable for time resolved simulations such as DDES one would have to perform DA state estimation using instantaneous snapshots for both CFD runs and experiments, which were not available for this analysis. However, the procedure would be exactly the same, therefore opening research opportunities for future investigations.
The flow prediction results obtained using RANS combined with ML tools are analyzed in the remainder of this section. The comparisons are performed considering the mean velocity fields on the planes **P1**, **P2** and **P3**, where PIV data is available. First, Figure 19 presents the results obtained on the plane **P1** (blue plane and box in Figure 12). As a reminder, Figure 19 (a,i) displays the axial velocity u_z obtained from PIV measurements, (a,ii) the mean axial velocity predicted by the prior DDES, (a,iii) the mean axial velocity $\bar{u}_z$ predicted by the prior RANS, while (a,iv) presents the ensemble mean of the updated axial velocity from the MFEnKF analysis $\langle \bar{u}_z^a \rangle$. In addition to these previously presented results, Figure 19 also provides the mean velocity fields obtained with the data-driven models RANS + CR1, RANS + CR2 and RANS + CR3. The second column reports the L2 norm of the discrepancy with PIV data $\left| u_z^{\mathrm{PIV}} - \bar{u}_z^i \right|$, where i indicates one of the numerical simulations previously listed. A first qualitative analysis reveals that the field predicted by the model RANS + CR1, shown in Figure 19 (a,v), shows strong similarities with the field $\langle \bar{u}_z^a \rangle$ obtained via DA in Figure 19 (a,iv), and therefore with the u_z^{PIV} measurements. Comparison with the RANS data in (a,iii) highlights an improvement in the prediction of the axial velocity

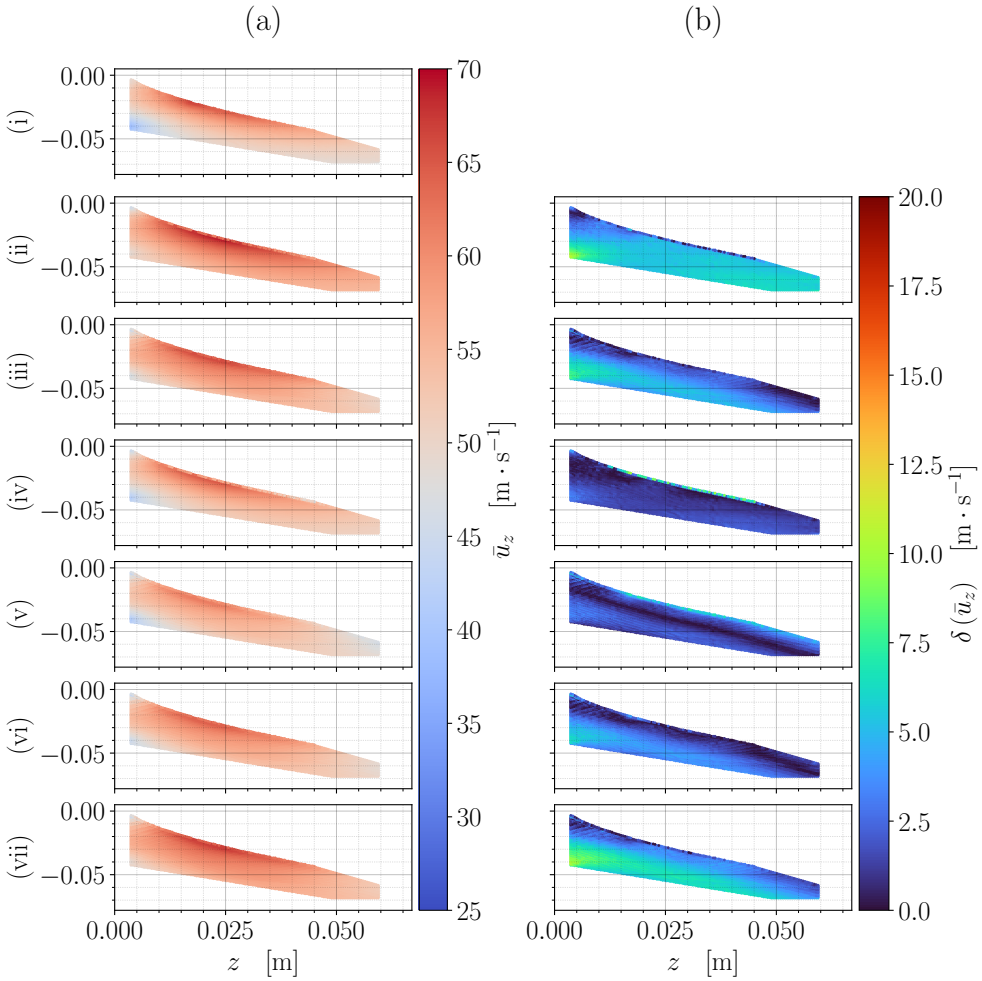


Fig. 19: Mean axial velocity $\bar{u}_z$ on **P1**. Column (a): isocontours of $\bar{u}_z$. Column (b): L2 norm measuring the discrepancy $\delta(\bar{u}_z) = \left|u_z^{\mathrm{PIV}} - \bar{u}_z\right|$ relative to the reference PIV data. Line (i) Reference experimental data (PIV); line (ii) DDES results; line (iii) RANS results; line (iv) DA state estimation; line (v) RANS + CR1 ML model; line (vi) RANS + CR2 ML model; line (vii) RANS + CR3 ML model.

at the leading edge of the lower profile, located in the lower-left corner of the plane. However, some discrepancies remain between the $\bar{u}_z^{\mathrm{PIV}}$ and $\bar{u}_z^{\mathrm{ML}}$ fields, particularly at the trailing edge of the upper profile where the ML approach underestimates the axial velocity. Discrepancies calculated via Equation 15 are reported in Table 1. From a quantitative standpoint, as a reminder, the DA analysis reduced the discrepancy for the velocity field from 6.87 % (RANS value) to 2.59 %. The integration of the CR1 ML model in the RANS run results in an L2 error norm of 3.08 % on the plane from which it received training data. This result proves to be satisfactory, as not only are these

errors lower than those of the original RANS simulation, but they are also very similar to those observed for the DA approach. For comparison, the velocity field obtained on **P1** by the models CR2 and CR3 is shown in lines (vi) and (vii). The L2 error norms are 4.81 % and 9.49 % respectively. These results are more in line with the prior RANS prediction, even if one can see that the run RANS + CR3 is the one exhibiting the largest discrepancy. This simulation relies on the model trained with data close to the blade tip.

The velocity fields predicted by the numerical simulations for the plane **P2** are now investigated. The results are shown in Figure 20 following the same structure previously used for the plane **P1** i.e. velocity fields and their discrepancy with respect to PIV data are shown. The combination of RANS and the ML model CR2 is first considered. This model is trained using DA data in the proximity of the plane **P2**, therefore it is expected to perform the best. The analysis of the results in Figure 20 (vi) confirms the accuracy of the prediction.

A qualitative analysis of the axial velocity isocontours in column (a) shows that the axial velocity field $\bar{u}_z^{\mathrm{ML}}$ predicted using the RANS + CR2 model (a,vi) well reproduces overall the topology of the flow resulting from the DA analysis, thus exhibiting a reduced discrepancy with the reference experimental measurements $\bar{u}_z^{\mathrm{PIV}}$. Examination of the L2 discrepancies $\delta(\bar{u}_z)$ in column (b) (Table 1) allows for quantification of these previous observations. The initial error of the RANS model was already relatively low (2.91 %) but the MFEnKF DA algorithm successfully reduced it to 1.81 %. The application of the RANS + CR2 model leads to an L2 error norm of 2.65 %, which is moderately reduced but still improved when compared with the baseline. The performance of the CR1 and CR3 ML models for the prediction of the flow on the plane **P2** is shown in Figure 20 (v) and (vii). Recall that the corrective models are here used far from the region where they were trained. Their average discrepancy shows an L2 error norm of 4.15 % and 4.75 %, respectively, which are higher than those observed for the solution via the CR2 model in **P2**, but also higher than the 2.91 % of the prior RANS solution.

The velocity fields predicted by the numerical simulations for the plane **P3** are finally investigated. The results are shown in Figure 21, following the same structure previously used for the planes **P1** and **P2**. As a reminder, **P3** is located at 79 % of the rotor blade height, in the proximity of the blade tip where complex and three-dimensional dynamics are observed. The combination of RANS and the ML model CR3 is first considered. This model is trained using the DA data in the proximity of the plane **P3**, so it is expected to perform the best. However, the analysis of the results in Figure 21 (vii) does not confirm that expectation. A qualitative analysis of the mean axial velocity field isocontours in column (a) shows that the axial velocity field $\bar{u}_z^{\mathrm{ML}}$ predicted using the RANS+CR3 model shown in (a,vii) is not able to match the accuracy of the MFEnKF prediction, which is shown in Figure 21 (a,iv). As a consequence, it exhibits a large discrepancy with the reference experimental measurements $\bar{u}_z^{\mathrm{PIV}}$. Using Equation 15, the error of the prior RANS is 5.55 % and the MFEnKF algorithm successfully reduces it to 3.81 % thanks to a coherent correction of the physical field. Nevertheless, the application of the RANS+CR3 model leads to an L2 error norm of 7.66 %, which represents a significant degradation of the accuracy

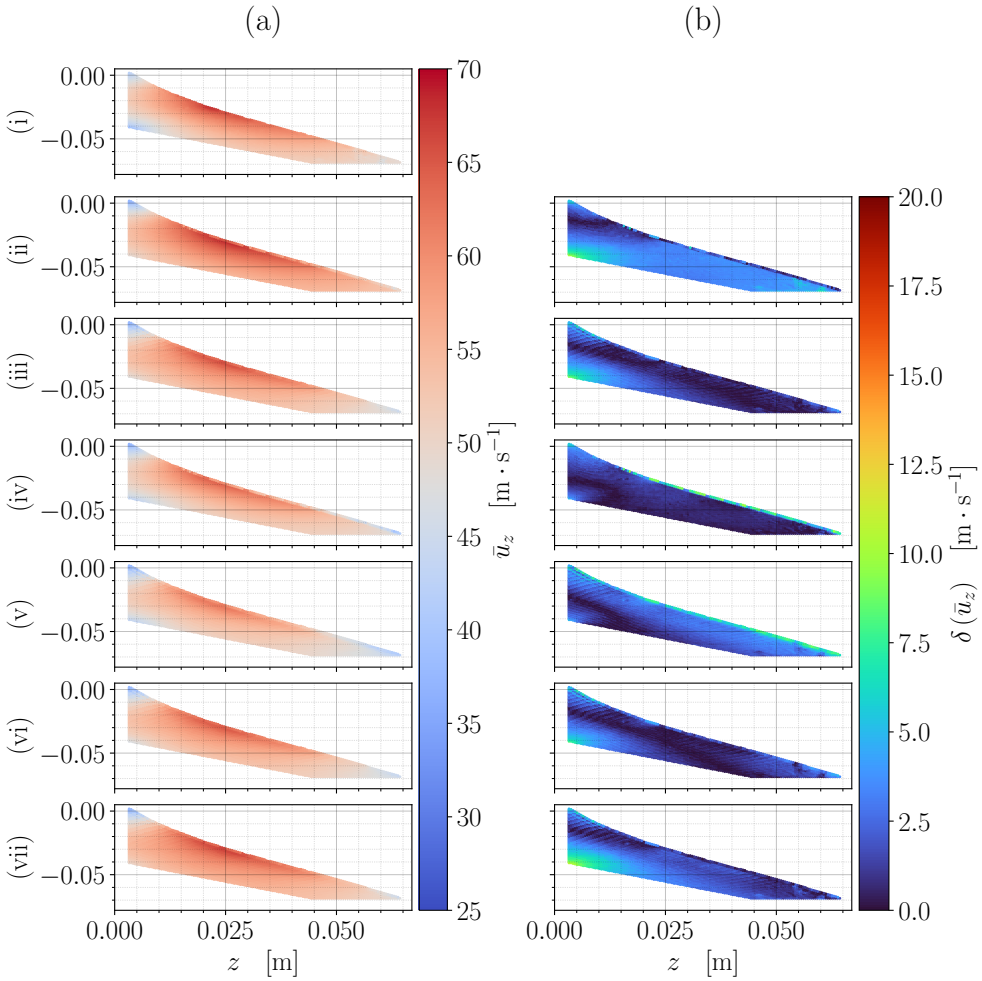


Fig. 20: Mean axial velocity $\bar{u}_z$ on **P2**. Column (a): isocontours of $\bar{u}_z$. Column (b): L2 norm measuring the discrepancy $\delta(\bar{u}_z) = \left|u_z^{\mathrm{PIV}} - \bar{u}_z\right|$ relative to the reference PIV data. Line (i) Reference experimental data (PIV); line (ii) DDES results; line (iii) RANS results; line (iv) DA state estimation; line (v) RANS + CR1 ML model; line (vi) RANS + CR2 ML model; line (vii) RANS + CR3 ML model.

with respect to the baseline results. This loss of accuracy can be explained by the local aerodynamic complexity of this region in the compressor, where the flow physics is governed by strong gradients, tip clearance flows and other vortical structures. Therefore, the accumulation of the physical phenomena involved represents a major hurdle for the application of ML tools, even in the proximity of the region where the CR3 training data was sampled. The performance of the CR1 and CR2 ML models for the prediction of the flow on the plane **P3** is shown in Figure 21 (v) and (vi) respectively. It is reminded that these corrective models are here evaluated far from the location

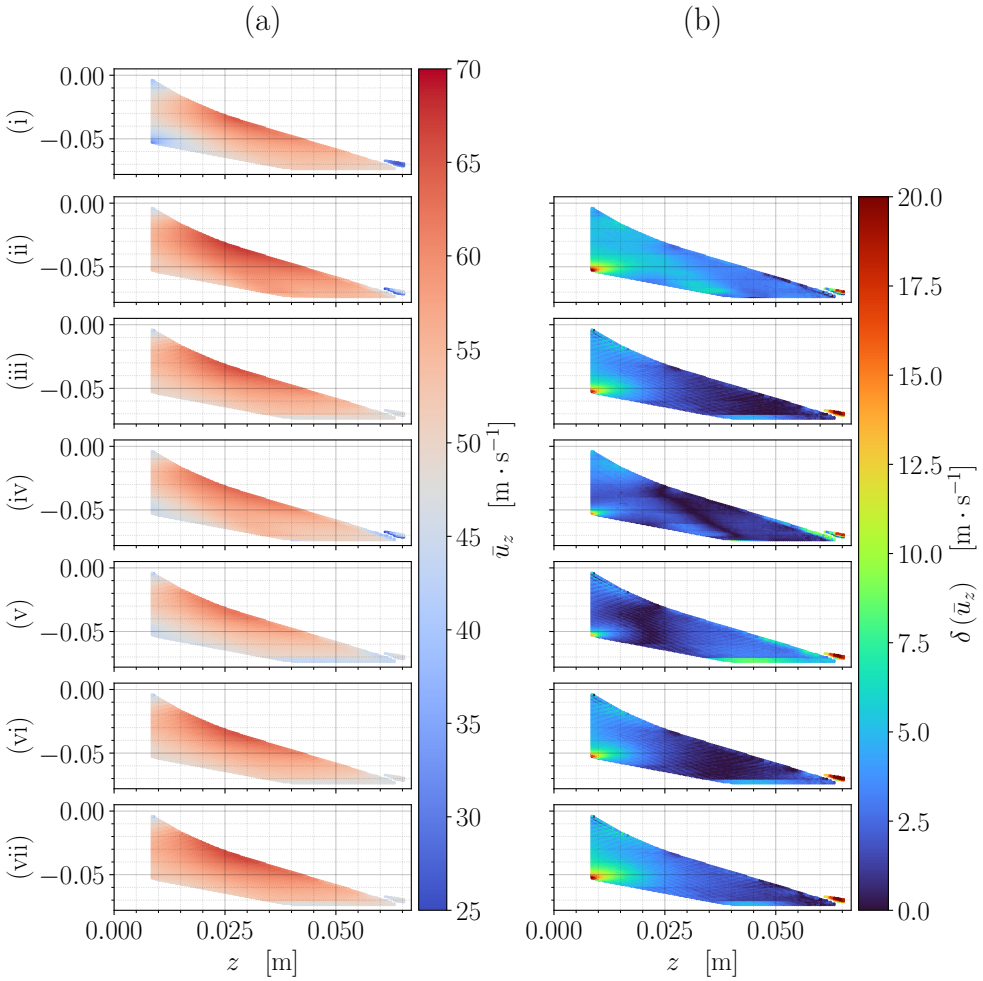


Fig. 21: Mean axial velocity $\bar{u}_z$ on **P3**. Column (a): isocontours of $\bar{u}_z$. Column (b): L2 norm measuring the discrepancy $\delta(\bar{u}_z) = \left|u_z^{\mathrm{PIV}} - \bar{u}_z\right|$ relative to the reference PIV data. Line (i) Reference experimental data (PIV); line (ii) DDES results; line (iii) RANS results; line (iv) DA state estimation; line (v) RANS + CR1 ML model; line (vi) RANS + CR2 ML model; line (vii) RANS + CR3 ML model.

where they received data for training. Their average discrepancy shows an L2 error norm of 4.78 % and 5.17 %. These values are higher than the one obtained with DA, but they are slightly lower than the 5.55 % of the prior RANS prediction. Therefore, on **P3**, the RANS+CR3 run not only fails to approach the MFEnKF results, but is also the least accurate of all the performed RANS simulations.
The analysis of the corrective models CR1, CR2 and CR3 has shown that, while an improvement of the prediction can be obtained, the results may be unreliable and also be responsible for a decrease in accuracy with respect to the baseline RANS prediction.

This is the case, for example, for the CR1 model applied on **P2**, as well as the model CR3 on all three planes. The next research question is whether this sensitivity of the models can be reduced by combining them in a common numerical framework, which is the mixture of experts model.

6.4 Mixture of experts development

In this section, the three corrective models CR1, CR2, and CR3 are blended with a single RANS simulation. This procedure exploits their higher accuracy when applied in physical regions which show similarities with their training data, therefore producing a more efficient global model. This procedure is performed with a mixture of experts (MoE) technique [80] which has been recently applied in fluid mechanics studies [81]. The procedure trains a master model which, via classification, decides which combination of the available model corrections is going to be employed. In the present work, for the sake of simplicity, the master model works with the geometric features of the compressor. While an algebraic law depending on the geometry could have been used in its place, the present work prepares a platform to include more sophisticated features in the selection of the model combination in future analyses. These features will include for example the local characteristics of the flow in terms of velocity, pressure and temperature.

6.4.1 Architecture setup

The architecture used for the present analysis is schematized in Figure 22. The three models CR1, CR2 and CR3, which are trained independently using different datasets, are referred to as *the experts*. They are now blended together via a classification network which routes each grid element to the appropriate combination of experts.
The training process of the classification network is now described. First, the **input** data is stored in the $\mathbf{c}_{\text{input}}$ vector with:

$$\mathbf{c}_{\text{input}} = (R, Z) \tag{24}$$

where (R, Z) corresponds to the cylindrical coordinates computed from the cartesian coordinates (x, y, z) as:

$$\begin{cases} R = \sqrt{x^2 + y^2} \\ Z = z \end{cases} \tag{25}$$

The architecture of the classification network, composed of 675 trainable parameters, is described in Table 3. The softmax activation function $\boldsymbol{\sigma}$ is used on the output layer, for its ability to convert a tuple of real values $\mathbf{c} = (c_1, \ldots, c_K) \in \mathbb{R}^K$ to a probability distribution $\boldsymbol{\sigma}(\mathbf{c}) \in (0, 1)^K$, thanks to the relation:

$$\boldsymbol{\sigma}(\mathbf{c})_i = \frac{\exp(c_i)}{\sum_{j=1}^{K} \exp(c_j)} \tag{26}$$

The **output** data is extracted from the state estimation boxes produced via MFEnKF in Figure 12 (b), where grid elements receive a label depending on their geometric

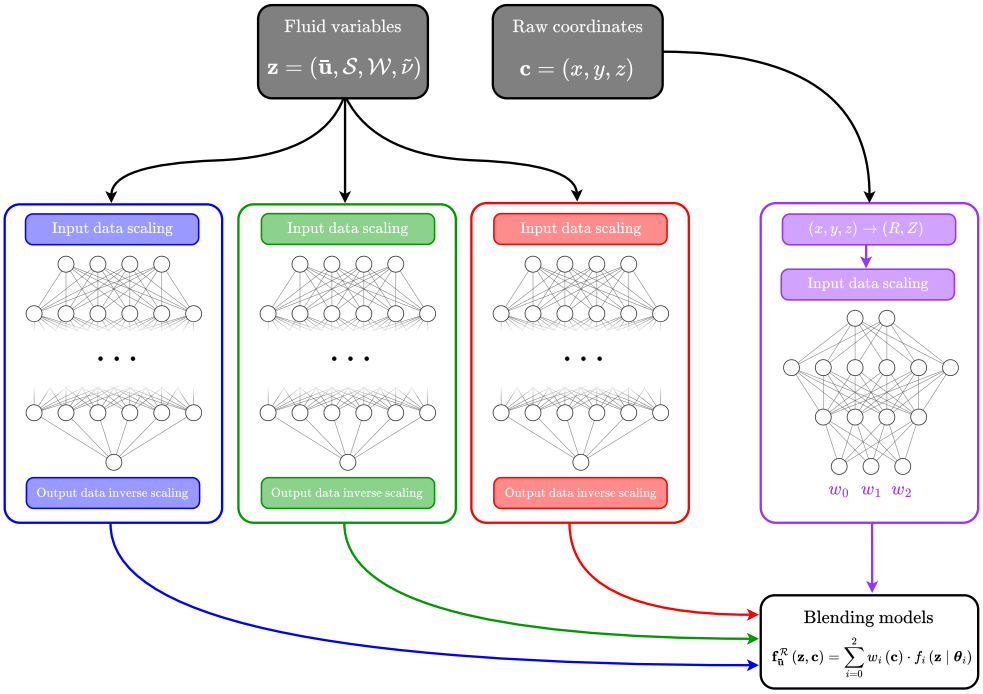


Fig. 22: Mixture of experts architecture.

Layers	Output shape	Number of parameters	Activation function
Input	2	0	–
Hidden 1	32	96	ReLU
Hidden 2	16	528	ReLU
Output	3	51	Softmax

Table 3: Hyperparameters used for the classification ANN used by the MoE

position and whether they are inside a state estimation region. We have the following encoding:

$$\begin{cases} \text{Elements in Box 1} \longrightarrow \text{Label 0} \longrightarrow \texttt{(1,0,0)} \\ \text{Elements in Box 2} \longrightarrow \text{Label 1} \longrightarrow \texttt{(0,1,0)} \\ \text{Elements in Box 3} \longrightarrow \text{Label 2} \longrightarrow \texttt{(0,0,1)} \end{cases} \tag{27}$$

These labels, which act as weights in the blending procedure, are stored in the $\mathbf{c}_{\text{output}} = (w_0, w_1, w_2)$ probability vector.
Data are split to a training and a validation set, respectively representing 80 % and 20 % of the full dataset. This classification model is compiled using the Categorical

Cross-Entropy (CCE) loss function $\mathcal{L}_{\text{CCE}}$, defined as:

$$\mathcal{L}_{\text{CCE}} = -\sum_{i=1}^{K} y_i \log\left(\hat{y}_i\right) \tag{28}$$

where y_i is the ground truth and $\hat{y}_i$ is the output predicted by the network.
Finally, the training process uses the Adam optimizer with a learning rate set to 10^{-3} and an automatic reduction based on the validation set accuracy if the metric does not improve for 10 consecutive epochs, down to a learning rate of 10^{-9}. An early stopping checkpoint is likewise implemented to halt training when performance stagnates. Specifically, training stops if the validation loss shows no improvement for 20 consecutive epochs.
Once the three experts and the classifier are trained, the models are blended into a single prediction $\mathbf{f}_{\tilde{\mathbf{u}}}^{\mathcal{R}}(\mathbf{z}, \mathbf{c})$:

$$\mathbf{f}_{\tilde{\mathbf{u}}}^{\mathcal{R}}(\mathbf{z}, \mathbf{c}) = \sum_{i=0}^{2} w_i(\mathbf{c}) \cdot f_i(\mathbf{z} \mid \boldsymbol{\theta}_i) \tag{29}$$

where $w_i(\mathbf{c})$ represents the weights from the classification network and $f_i(\mathbf{z} \mid \boldsymbol{\theta}_i)$ the prediction from each expert CRi. It is important to stress that the weights may vary for each grid element, therefore tuning the correction performed. The behavior of the trained MoE model is assessed in Figure 23, which presents the variation of the expert weighting coefficients $w_{i\in[0,2]}$ along the channel height, illustrating how the classification network attributes each expert to its corresponding flow region. It can be seen that these distributions are smooth and they only depend on the radius of the compressor stage. Again, future analyses will substitute geometric information with local features of the flow field.

6.4.2 MoE results

The mixture of experts architecture is integrated within a RANS simulation of the CME2 axial compressor and results are now discussed. Similarly to what was proposed in subsection 6.3, the results are discussed for each plane where PIV data are available. Figure 24 shows for **P1** that the axial velocity field $\bar{u}_z^{\text{MoE}}$ obtained with the new model, which is displayed in (a,v), exhibits strong similarities with the field $\langle \bar{u}_z^a \rangle$ obtained by DA in (a,iv), and therefore the experimental measurements $\bar{u}_z^{\text{PIV}}$ as well. The MFEnKF results on this plane exhibited an average discrepancy with PIV of 2.59 %, which is significantly lower than the 6.87 % value observed for RANS. The combination of RANS with the correction model CR1 provided an average discrepancy of 3.08 %, very close to the MFEnKF result. The RANS+MoE model yields a higher average L2 error norm of 4.17 %. This outcome can be considered as satisfactory, as it remains significantly lower than the prior RANS field prediction and it is of the same order of magnitude observed for the DA results and the RANS + CR1 model results. Looking at the distribution of the L2 discrepancy norm $\delta(\bar{u}_z)$ over the plane **P1**, one can see that the topology of the RANS + MoE model is similar to the one of the initial

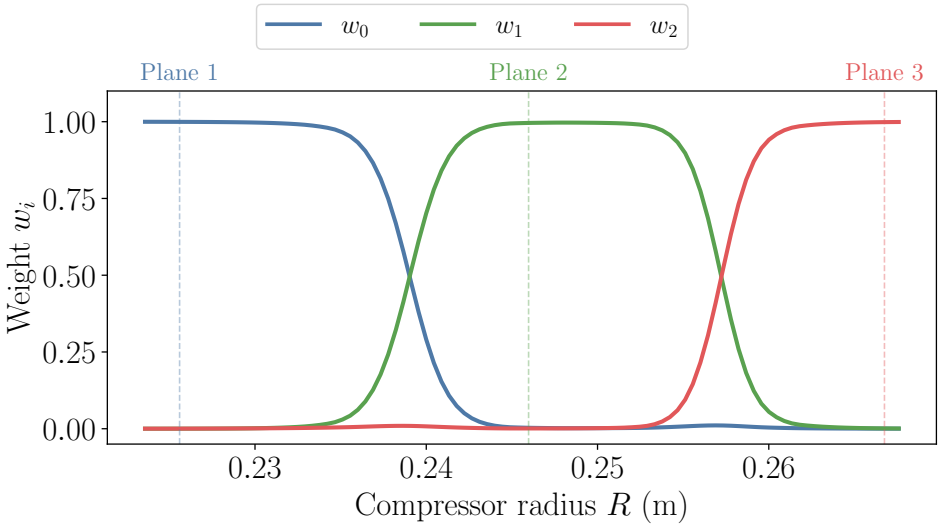


Fig. 23: Spatial distribution of the expert weighting coefficients across the computational domain.

RANS simulation, but the discrepancy is reduced everywhere. This suggests that the MoE correction acts as a global attenuation of the RANS error rather than a local correction of specific flow features.

Figure 25 shows the mean axial velocity field on **P2**. As a reminder, the average discrepancies with respect to the PIV data of the DDES and RANS simulations were 6.21 % and 2.91 %, respectively. The MFEnKF analysis in (a,iv) reduced this error to 1.81 %, and the combination of RANS and CR2 model produces an average L2 error norm of 2.65 %. The application of the MoE architecture shown in Figure 25 (a,v) yields an average L2 discrepancy norm of 2.78 %, which is very similar to what is observed for the RANS+CR2 model combination and slightly lower than the results obtained for the baseline RANS prediction. This observation suggests that the ML training was able to provide a robust behavior of both the experts and the classification network, as the accuracy of the RANS prediction is preserved and even slightly improved. The examination of the distribution of the L2 discrepancy norm $\delta(\bar{u}_z)$ over the plane **P2** shows that the baseline RANS solution and the RANS + MoE model exhibit very similar distributions, highlighting the robustness of the model and the consistency observed on the plane **P1**.

Finally, Figure 26 shows the analysis of data sampled on **P3**. It is reminded that this plane is located at 79 % of the rotor blade height, and in this region three-dimensional effects due to tip vortices are present. On this plane, the average L2 discrepancy norm for the RANS baseline model was 5.55 % while the MFEnKF analysis reduced the L2 error norm to 3.81 %. The combination of RANS with the CR3 model failed to provide accurate predictions, with an L2 error norm rising to 7.66 %, thus degrading the results with respect to the RANS baseline prediction. The RANS + MoE model provides a significant improvement in accuracy in this region, with an average L2

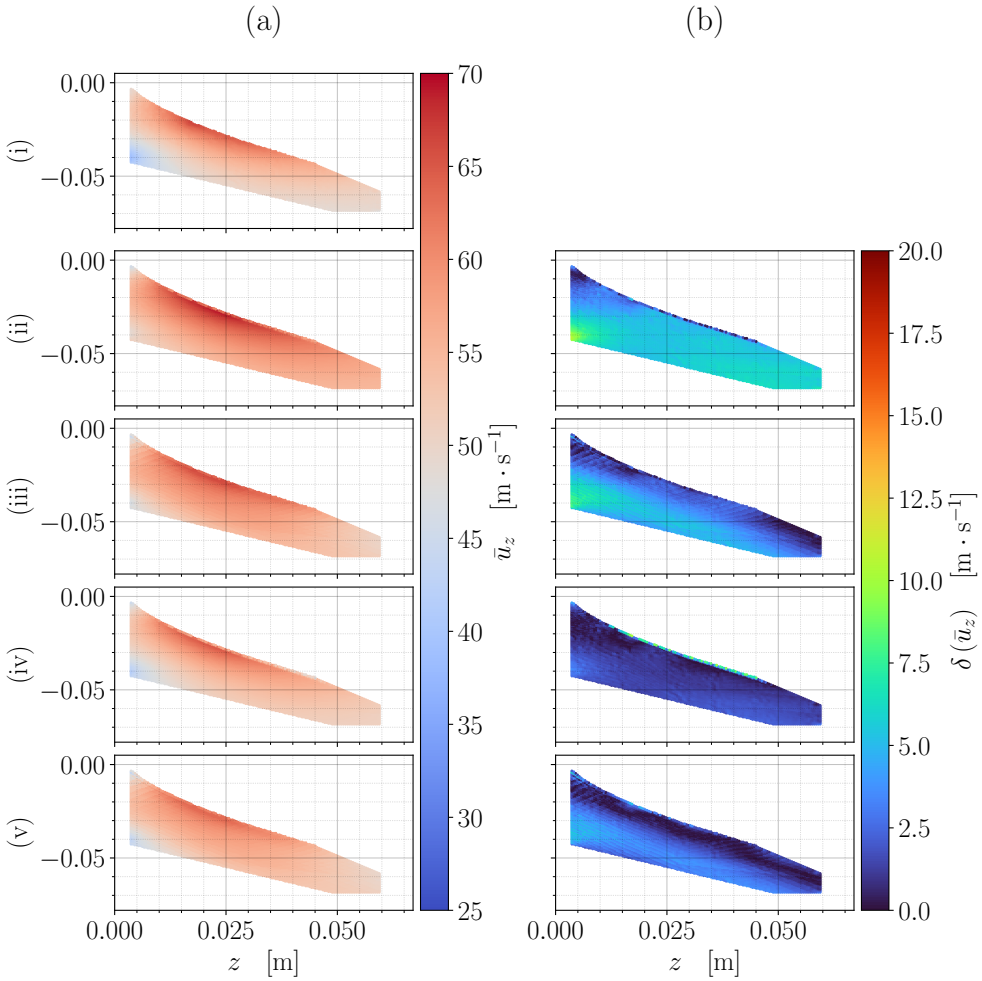


Fig. 24: Mean axial velocity $\bar{u}_z$ on **P1**. Column (a): isocontours of $\bar{u}_z$. Column (b): L2 norm measuring the discrepancy $\delta(\bar{u}_z) = \left|u_z^{\mathrm{PIV}} - \bar{u}_z\right|$ relative to the reference PIV data. Line (i) Reference experimental data (PIV); line (ii) DDES results; line (iii) RANS results; line (iv) DA state estimation; line (v) RANS + MoE ML model.

discrepancy norm of 3.84 % which is in close agreement with the MFEnKF result. This can be particularly appreciated by analyzing the topology of the L2 error distribution in Figure 26 (b,v). This result suggests that the specialization of each expert allows the MoE model to correctly identify the local physical phenomena acting in this plane, rather than attempting to understand the full complexity of the flow with a single-network model.

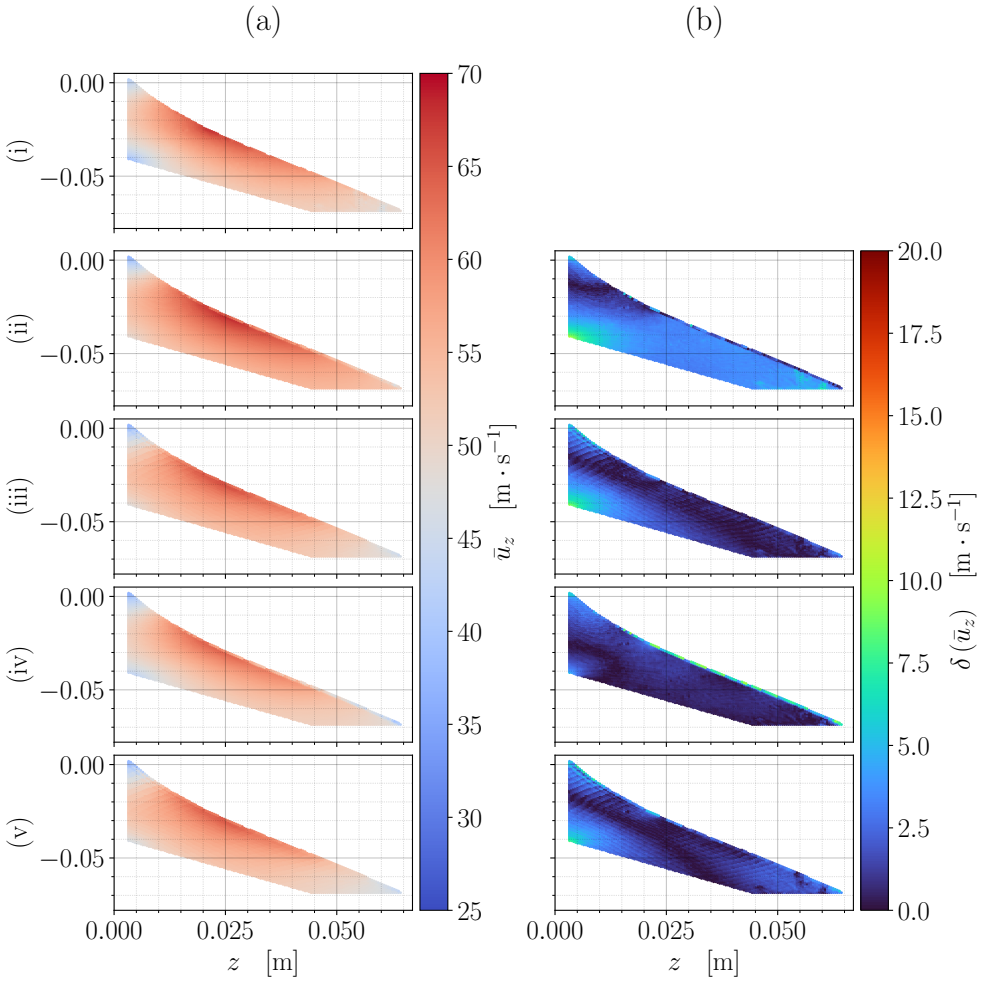


Fig. 25: Mean axial velocity $\bar{u}_z$ on **P2**. Column (a): isocontours of $\bar{u}_z$. Column (b): L2 norm measuring the discrepancy $\delta(\bar{u}_z) = \left|u_z^{\mathrm{PIV}} - \bar{u}_z\right|$ relative to the reference PIV data. Line (i) Reference experimental data (PIV); line (ii) DDES results; line (iii) RANS results; line (iv) DA state estimation; line (v) RANS + MoE ML model.

7 Conclusions

This work presents the development and assessment of a hybrid methodology coupling numerical simulation, data assimilation (DA), and machine learning (ML) for the analysis of the flow within the CME2 compressor. First, the DA algorithm used for the present analysis, the Multi-Fidelity Ensemble Kalman Filter (MFEnKF), was implemented combining PIV data and model realizations from RANS and DDES runs. These two modeling techniques used different representation strategies for the movement of the rotor. Hyperlocalization of the state estimation was performed in three

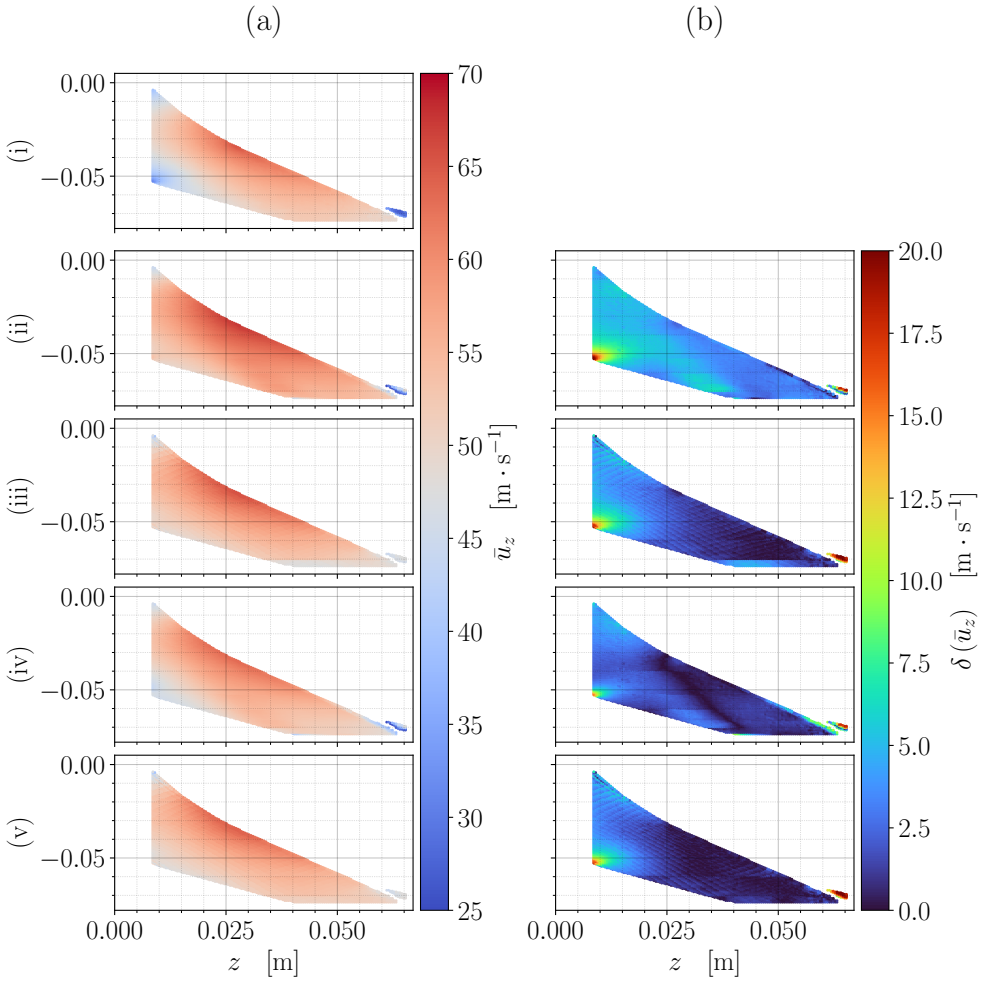


Fig. 26: Mean axial velocity $\bar{u}_z$ on **P3**. Column (a): isocontours of $\bar{u}_z$. Column (b): L2 norm measuring the discrepancy $\delta(\bar{u}_z) = \left|u_z^{\text{PIV}} - \bar{u}_z\right|$ relative to the reference PIV data. Line (i) Reference experimental data (PIV); line (ii) DDES results; line (iii) RANS results; line (iv) DA state estimation; line (v) RANS + MoE ML model.

regions in the rotor region of the compressor surrounding the planes where experimental data was available. The MFEnKF was able to significantly reduce the discrepancy of the base model with the available observations while ensuring an accurate prediction of the governing physical features of the flow. In addition, the MFEnKF was able to expand the size of the initial PIV dataset, therefore producing a larger number of high fidelity samples which can be used for machine learning training purposes. ML correction tools were then developed using Neural Networks in the form of source terms to be integrated within the RANS momentum equation. More precisely, the

developed architecture retrieves various physical data, such as the velocity, the velocity gradients and turbulent viscosity, to predict a volume forcing with the objective to reproduce the DA solution. The assessment of this methodology shows a global satisfactory accuracy across most of the application domain. More precisely, results using the ML models CR1, and CR2, which are trained using data in the proximity of planes **P1** and **P2**, yield average errors close to the ones observed for the DA prediction when used on **P1** and **P2** respectively. Nevertheless, this study also highlighted the robustness limits inherent to this ML approach. In regions with high aerodynamic complexity, the lack of complete physical information prevents the generalization of the ANN model, which then can degrade the prediction of the base RANS model. This has been observed for example for the model CR3, which was trained using data in the proximity of the blade tip.
These findings prompted further development to consolidate the hybrid DA/ML approach. For this reason, a mixture of experts (MoE) approach has been developed to combine the performance of the three models CR1, CR2 and CR3 in a single model. More precisely, the MoE recognizes which blend of the base ML models should be used in each grid element of the domain, trying to improve the global accuracy of the prediction. The results of such method for the flow prediction in planes **P1**, **P2** and **P3** shows that the model globally produces results similar to the DA state estimation, therefore opening perspectives of applications in several fields where sparse data may be used for training of ML models.

A Spalart-Allmaras modeling configuration

Here is the complete set of equations used for the Spalart-Allmaras model as it is coded in the numerical solver of OpenFOAM:

$$
\begin{aligned}
\tilde{S} \equiv S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \qquad & f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \\
S = \sqrt{2 W_{ij} W_{ij}}, \qquad & W_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) \\
f_{t2} = C_{t3} \exp\left(-C_{t4} \chi^2\right), \qquad & f_w = g \left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6} \right]^{1/6} \\
g = r + C_{w2}(r^6 - r), \qquad & r \equiv \frac{\tilde{\nu}}{\tilde{S} \kappa^2 d^2}
\end{aligned}
\tag{30}
$$

B Ensemble Kalman Filter

The stochastic Ensemble Kalman Filter (EnKF) builds upon the fundamental principle of the classic Kalman Filter (KF) [61], but tailored for high-dimensional and non-linear systems. Initially proposed by [22] for state estimation and later generalized for parameter optimization [23], this methodology utilizes a population of n_e distinct

σ	C_{b1}	C_{b2}	κ	C_{w1}	C_{w2}	C_{w3}	C_{v1}	C_{t1}	C_{t2}	C_{t3}	C_{t4}
2/3	0.1355	0.622	0.41	3.2391	0.3	2	7.1	1	2	1.1	2

Table 4: Values chosen for the model constants for the turbulence closure used in the present analysis. These values are the ones provided as default in the implementation of the code.

realizations commonly referred to as *ensemble members*. The primary features of the EnKF algorithm include a *forecast* and an *analysis*, and are now detailed:

1. **Forecast:** Every state vector $i \in [1, n_e]$ is advanced from time $k-1$ to k, using the model $\mathcal{M}$:
$$\boldsymbol{x}_{i,k}^{f} = \mathcal{M}_{k-1:k}\left(\boldsymbol{x}_{i,k-1}^{a}, \boldsymbol{\varpi}_{i,k-1}^{a}\right) \tag{31}$$
in which f and a superscripts refer to *forecast* and *analysis* states respectively. In computational fluid dynamics, the model usually represents a discretized formulation of the governing equations, namely the compressible Navier-Stokes equations in that case. The numerical model $\mathcal{M}\,()$ advances the state vector which encompasses the physical quantities of interest, such as velocity, pressure or any other relevant quantities defined at time $k-1$. Simultaneously, the time propagation can include parameter vectors $\boldsymbol{\varpi}_{i,k-1}$ which characterize underlying physical features and laws, or boundary conditions. The EnKF algorithm is then solved using the so called *extended state* [19]. The ensemble representation gives an estimate of the model's uncertainty. Indeed, the ensemble average is assumed to be an approximation of the state ground truth $\boldsymbol{x}_{k-1}^{\text{truth}}$, which is used to define the error covariance matrix $\boldsymbol{P}_{k-1}$:
$$\boldsymbol{P}_{k-1} = \mathbb{E}\left(\left(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{\text{truth}}\right)\left(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{\text{truth}}\right)^{\top}\right) \tag{32}$$
2. **Observation data construction:** A number of n_o observation data available at time $k-1$ is set in a vector $\boldsymbol{y}_{k-1} \in \mathbb{R}^{n_o}$. This observation vector is then arranged into a matrix $\boldsymbol{y}_{i,k-1} \in \mathbb{R}^{n_o, n_e}$ such that:
$$\boldsymbol{y}_{i,k-1} \equiv \boldsymbol{y}_{k-1} + \boldsymbol{\varepsilon}_i \tag{33}$$
where $\boldsymbol{\varepsilon}_i \sim \mathcal{N}\left(0, \boldsymbol{\varsigma}_k\right)$. $\boldsymbol{\varsigma}_k$ is a diagonal matrix dimensioned as $\mathbb{R}^{n_o, n_o}$ and representing the unbiased and uncorrelated uncertainty of the given observation data. The $\mathcal{H}\left(\cdot\right)$ operator is introduced as a projection operator from the model space to the observation space. Thus, $\mathcal{H}\left(\boldsymbol{x}\right)$ is used to obtain the interpolation of the model's data onto the position of the observation sensors.

3. **Ensemble averaging:** $\boldsymbol{x}_{i,k}^{f}$ and $\mathcal{H}\left(\boldsymbol{x}_{i,k}^{f}\right)$ are averaged over the n_e ensemble members.
$$\boldsymbol{\mu}_{k}^{\boldsymbol{x}^f} = \frac{1}{n_e}\sum_{i=1}^{n_e} \boldsymbol{x}_{i,k}^{f} \quad ; \quad \boldsymbol{\mu}_{k}^{\mathcal{H}(\boldsymbol{x}^f)} = \frac{1}{n_e}\sum_{i=1}^{n_e} \mathcal{H}\left(\boldsymbol{x}_{i,k}^{f}\right) \tag{34}$$
4. **Anomaly matrices:** Using the hypothesis of statistical independence between the n_e ensemble members, the error covariance matrix is propagated from time $k-1$ to

time k via some anomaly matrices $\mathbf{\Lambda}_k^{\boldsymbol{x}^f}$, through the expression $\boldsymbol{P}_k^f \approx \mathbf{\Lambda}_k^{\boldsymbol{x}^f} \left(\mathbf{\Lambda}_k^{\boldsymbol{x}^f}\right)^\top$, for which each column describes the deviation of the i^{th} ensemble member from the ensemble mean. Analogously, the projection of the model onto the observation space can be expressed by means of another anomaly matrix $\mathbf{\Lambda}_k^{\mathcal{H}(\boldsymbol{x}^f)}$. Therefore:

$$\mathbf{\Lambda}_k^{(\cdot)} = \frac{1}{\sqrt{n_e - 1}} \left((\cdot)_{1,k} - \boldsymbol{\mu}_k^{(\cdot)}, \ldots, (\cdot)_{n_e,k} - \boldsymbol{\mu}_k^{(\cdot)} \right) \tag{35}$$

where $(\cdot)$ may be replaced with $\boldsymbol{x}^f$ and $\mathcal{H}\left(\boldsymbol{x}^f\right)$ to respectively obtain $\mathbf{\Lambda}_k^{\boldsymbol{x}^f}$ and $\mathbf{\Lambda}_k^{\mathcal{H}(\boldsymbol{x}^f)}$.

5. **Covariance matrices:** Covariance matrices $\mathbf{\Xi}^{(\cdot),(\cdot)}$ are calculated from the previously computed anomaly matrices.

$$\mathbf{\Xi}_k^{\boldsymbol{x}^f,\mathcal{H}(\boldsymbol{x}^f)} = \mathbf{\Lambda}_k^{\boldsymbol{x}^f} \left(\mathbf{\Lambda}_k^{\mathcal{H}(\boldsymbol{x}^f)}\right)^\top \tag{36}$$

$$\mathbf{\Xi}_k^{\mathcal{H}(\boldsymbol{x}^f),\mathcal{H}(\boldsymbol{x}^f)} = \mathbf{\Lambda}_k^{\mathcal{H}(\boldsymbol{x}^f)} \left(\mathbf{\Lambda}_k^{\mathcal{H}(\boldsymbol{x}^f)}\right)^\top \tag{37}$$

6. **Kalman Gain matrix:** The Kalman Gain matrix $\boldsymbol{K}$ determines the optimal weighting when combining the model prediction and the observations, based on their respective error statistics, to minimize the posterior estimation error.

$$\boldsymbol{K}_k = \mathbf{\Xi}_k^{\boldsymbol{x}^f,\mathcal{H}(\boldsymbol{x}^f)} \left(\mathbf{\Xi}_k^{\mathcal{H}(\boldsymbol{x}^f),\mathcal{H}(\boldsymbol{x}^f)} + \boldsymbol{\varsigma}_k\right)^{-1} \tag{38}$$

7. **Analysis:** This step computes the updated state vector $\boldsymbol{x}_{i,k}^a$ and model parameters $\boldsymbol{\varpi}_{i,k}^a$ by correcting the forecast state vector $\boldsymbol{x}_{i,k}^f$ and the previous model parameters $\boldsymbol{\varpi}_{i,k-1}^f$ using the difference between the observation vector $\boldsymbol{y}_k$ and the sampled data $\mathcal{H}\left(\boldsymbol{x}_{i,k}^f\right)$. This difference, known as the *innovation term*, is weighted by the Kalman Gain matrix $\boldsymbol{K}$.

$$\boldsymbol{x}_{i,k}^a = \boldsymbol{x}_{i,k}^f + \boldsymbol{K}_k \left(\boldsymbol{y}_k - \mathcal{H}\left(\boldsymbol{x}_{i,k}^f\right)\right) \tag{39}$$

$$\boldsymbol{\varpi}_{i,k}^a = \boldsymbol{\varpi}_{i,k}^f + \boldsymbol{K}_k \left(\boldsymbol{y}_k - \mathcal{H}\left(\boldsymbol{x}_{i,k}^f\right)\right) \tag{40}$$

Both corrections given in Equation 39 and Equation 40 are proportional to the discrepancy between the observations $\boldsymbol{y}$ and the model equivalent values $\mathcal{H}(\boldsymbol{x})$ weighted by the Kalman gain $\boldsymbol{K}$.
The performance of the EnKF heavily depends on how well the model and observation uncertainties are characterized. The update estimation is the most effective

when the uncertainty regions of both sources of information show significant intersection. However, if the observations are treated with low confidence, the analysis will remain biased toward the forecast state of the model, resulting in a limited increase in accuracy. Despite its great accuracy, the EnKF relies on a relatively high number of numerical simulations, which can rapidly become computationally expensive when dealing with high dimensional systems. Constraining the ensemble size to mitigate this cost often introduces sampling errors and a systematic underestimation of state variance, which can ultimately lead to filter divergence and a failure to converge.

C Multi-Fidelity Ensemble Kalman Filter

First introduced by [24], the Multi-Fidelity Ensemble Kalman Filter is a dedicated Data Assimilation procedure combining full order models, referred to as FOM, along which an ensemble of reduced order stand-in models (ROM) are utilized to increase the computational efficiency while increasing physical accuracy of Data Assimilation. This multi-level method leverages the approach of the linear control variate, which is a variance reduction technique often used in Monte Carlo methodologies [82].
We consider a multi-fidelity framework comprising a high-fidelity model, denoted by $\boldsymbol{\chi}$, and a computationally efficient low-fidelity surrogate, represented by $\boldsymbol{\gamma}$. $\boldsymbol{\gamma}$ is a control variate of $\boldsymbol{\chi}$ if $\boldsymbol{\gamma}$ shows a strong correlation with the $\boldsymbol{\chi}$ variate and its expectation $\boldsymbol{\mu_\gamma}$ is *a priori* known.
The main objective of the control variate methodology is to build estimations of statistics for the high fidelity state, thanks to information coming from the lower-fidelity states, which becomes *de facto* computationally less expensive. This goal is reached with the usage of the total variate, expressed with $\boldsymbol{\varkappa}$ and defined with a linear combination involving $\boldsymbol{\mu_\chi}$, $\boldsymbol{\mu_\gamma}$, respectively the expectations of $\boldsymbol{\chi}$ and $\boldsymbol{\gamma}$:

$$\boldsymbol{\varkappa} = \boldsymbol{\chi} - \lambda\,(\boldsymbol{\gamma} - \boldsymbol{\mu_\gamma}) \tag{41}$$

The random variable $\boldsymbol{\chi}$ is called the *principal variate*, $\boldsymbol{\gamma}$ the *control variate* and $\boldsymbol{\varkappa}$ the *total variate*. In Equation 41, variable $\boldsymbol{\varkappa}$ has a lower variance compared to the principal variable $\boldsymbol{\chi}$, while their expectations exhibit $\boldsymbol{\mu_\varkappa} = \boldsymbol{\mu_\chi}$. When working in a practical framework, the $\boldsymbol{\mu_\gamma}$ is not known and the MFEnKF framework needs a new variable. Therefore, the variable $\boldsymbol{\eta}$, referred to as the *ancillary variate*, is introduced and selected to obtain $\boldsymbol{\mu_\gamma} = \boldsymbol{\mu_\eta}$. The ancillary variate is also computed on an ensemble framework and runs on the same low-resolution model as the control variate $\boldsymbol{\gamma}$. Thanks to the introduction of the ancillary variate $\boldsymbol{\eta}$, Equation 41 can now be written as:

$$\boldsymbol{\varkappa} = \boldsymbol{\chi} - \lambda\,(\boldsymbol{\gamma} - \boldsymbol{\eta}) \tag{42}$$

Equation 42 is now transposed to a multi-level approach. Two models are now defined. On the one hand, the CFD model $\mathcal{M}^{\mathbf{F}}$ is meant to be the *fine* model, leveraging a fine grid and accurate turbulence modeling. On the other hand, the CFD model $\mathcal{M}^{\mathbf{C}}$ uses a coarser grid and lower-accuracy turbulence modeling. Analogously to the control variates method presented earlier, three sub-ensembles are built to fit the ensemblist

methodology of the MFEnKF. The first sub-ensemble corresponds to the principal variate $\boldsymbol{\chi}$, composed of $N_{\boldsymbol{\chi}}$ members and run using the fine model $\mathcal{M}^{\mathbf{F}}$. In a similar manner, the control variate $\boldsymbol{\gamma}$ is represented by $N_{\boldsymbol{\gamma}} = N_{\boldsymbol{\chi}}$ ensemble members run with the coarse model $\mathcal{M}^{\mathbf{C}}$. Finally, the ancillary variate $\boldsymbol{\eta}$ is represented with $N_{\boldsymbol{\eta}}$ ensemble members, also run on the coarse model $\mathcal{M}^{\mathbf{C}}$. The total number of ensemble members N_e simultaneously run is equal to:

$$N_e = 2N_{\boldsymbol{\chi}} + N_{\boldsymbol{\eta}} \tag{43}$$

One can then notice that $N_{\boldsymbol{\gamma}} + N_{\boldsymbol{\eta}}$ ensemble members are run with the coarse model $\mathcal{M}^{\mathbf{C}}$, while $N_{\boldsymbol{\chi}}$ are run using the fine model $\mathcal{M}^{\mathbf{F}}$. Thus, the control variate theorem implies that a state estimation can be realized with a limited number of principal ensemble members $N_{\boldsymbol{\chi}}$, while using a much larger amount of coarse-model simulations $N_{\boldsymbol{\gamma}} + N_{\boldsymbol{\eta}}$. Steps of the algorithm are now detailed:

1. **Time Forecast:** Members from the three sub-ensembles are advanced in time thanks to their corresponding non-linear models $\mathcal{M}$:

$$\begin{aligned} \boldsymbol{x}_{i,k}^{\boldsymbol{\chi}^f} &= \mathcal{M}^{\boldsymbol{\chi}}\left(\boldsymbol{x}_{i,k-1}^{\boldsymbol{\chi}^a}\right) \\ \boldsymbol{x}_{i,k}^{\boldsymbol{\gamma}^f} &= \mathcal{M}^{\boldsymbol{\gamma}}\left(\boldsymbol{x}_{i,k-1}^{\boldsymbol{\gamma}^a}\right) \\ \boldsymbol{x}_{i,k}^{\boldsymbol{\eta}^f} &= \mathcal{M}^{\boldsymbol{\eta}}\left(\boldsymbol{x}_{i,k-1}^{\boldsymbol{\eta}^a}\right) \end{aligned} \tag{44}$$

2. **Projection from principal space $\boldsymbol{\chi}$ to control space $\boldsymbol{\gamma}$:** As $\mathcal{M}^{\boldsymbol{\chi}} \equiv \mathcal{M}^{\mathbf{F}}$ and $\mathcal{M}^{\boldsymbol{\gamma}} \equiv \mathcal{M}^{\mathbf{C}}$ are defined with different fidelities, physical quantities obtained on the principal and the control members become decorrelated along time advancement. Then, physical fields from the principal ensemble members are projected to the control ensemble members with the projection operator $\boldsymbol{\mathcal{P}}^{-}$, which produces a mapping of the fine model to the coarse model:

$$\boldsymbol{x}_{i,k}^{\boldsymbol{\gamma}^f} = \boldsymbol{\mathcal{P}}^{-}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\chi}^f}\right) \tag{45}$$

3. **Observation operator for the total variate:** The observation operator $\mathcal{H}(\cdot)$ may be used for the principal, as well as for the control and ancillary members using the projection operator $\boldsymbol{\mathcal{P}}^{+}$, contrary to the total variate ensemble members which are not directly observable. Nevertheless, a twin operator defined as $\widetilde{\mathcal{H}(\cdot)}$ is introduced and equal to:

$$\widetilde{\mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\chi}^f}\right)} = \mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\chi}^f}\right) - \lambda\left(\mathcal{H}\left(\boldsymbol{\mathcal{P}}^{+}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\gamma}^f}\right)\right)\right) + \lambda\left(\mathcal{H}\left(\boldsymbol{\mathcal{P}}^{+}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\eta}^f}\right)\right)\right) \tag{46}$$

   where λ is here equal to $1/2$.
4. **Computation of the covariance matrices:** The approximated covariance matrices are built using the anomaly matrices from the three sub-ensembles. Assuming the principal and the ancillary are defined as random and independent, we

have:

$$\begin{aligned}\boldsymbol{\Upsilon}^{\boldsymbol{x}_{i,k}^{\boldsymbol{\varkappa}^f},\widetilde{\mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\varkappa}^f}\right)}} &= \boldsymbol{\Upsilon}^{\boldsymbol{x}_{i,k}^{\boldsymbol{\chi}^f},\mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\chi}^f}\right)} \\ &+ \frac{1}{4}\left(\boldsymbol{\mathcal{P}}^+\left(\boldsymbol{\Upsilon}^{\boldsymbol{x}_{i,k}^{\boldsymbol{\gamma}^f},\mathcal{H}\left(\boldsymbol{\mathcal{P}}^+\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\gamma}^f}\right)\right)}\right) + \boldsymbol{\mathcal{P}}^+\left(\boldsymbol{\Upsilon}^{\boldsymbol{x}_{i,k}^{\boldsymbol{\eta}^f},\mathcal{H}\left(\boldsymbol{\mathcal{P}}^+\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\eta}^f}\right)\right)}\right)\right) \\ &- \frac{1}{2}\left(\boldsymbol{\Upsilon}^{\boldsymbol{x}_{i,k}^{\boldsymbol{\chi}^f},\mathcal{H}\left(\boldsymbol{\mathcal{P}}^+\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\gamma}^f}\right)\right)} + \boldsymbol{\mathcal{P}}^+\left(\boldsymbol{\Upsilon}^{\boldsymbol{x}_{i,k}^{\boldsymbol{\gamma}^f},\mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\chi}^f}\right)}\right)\right)\end{aligned} \tag{47}$$

$$\begin{aligned}\boldsymbol{\Upsilon}^{\widetilde{\mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\varkappa}^f}\right)},\widetilde{\mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\varkappa}^f}\right)}} &= \boldsymbol{\Upsilon}^{\mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\chi}^f}\right),\mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\chi}^f}\right)} \\ &+ \frac{1}{4}\left(\boldsymbol{\Upsilon}^{\mathcal{H}\left(\boldsymbol{\mathcal{P}}^+\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\gamma}^f}\right)\right),\mathcal{H}\left(\boldsymbol{\mathcal{P}}^+\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\gamma}^f}\right)\right)} + \boldsymbol{\Upsilon}^{\mathcal{H}\left(\boldsymbol{\mathcal{P}}^+\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\eta}^f}\right)\right),\mathcal{H}\left(\boldsymbol{\mathcal{P}}^+\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\eta}^f}\right)\right)}\right) \\ &- \frac{1}{2}\left(\boldsymbol{\Upsilon}^{\mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\chi}^f}\right),\mathcal{H}\left(\boldsymbol{\mathcal{P}}^+\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\gamma}^f}\right)\right)} + \boldsymbol{\Upsilon}^{\mathcal{H}\left(\boldsymbol{\mathcal{P}}^+\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\gamma}^f}\right)\right),\mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\chi}^f}\right)}\right)\end{aligned} \tag{48}$$

The observation error covariance matrices are constructed using independent and random variables such that:

$$\boldsymbol{\vartheta}^{\boldsymbol{\chi}} \sim \mathcal{N}\left(0, \boldsymbol{\varsigma}^{\boldsymbol{\chi}}\right) \tag{49}$$

$$\boldsymbol{\vartheta}^{\boldsymbol{\gamma}} = \boldsymbol{\vartheta}^{\boldsymbol{\chi}} \tag{50}$$

$$\boldsymbol{\vartheta}^{\boldsymbol{\eta}} \sim \mathcal{N}\left(0, \boldsymbol{\varsigma}^{\boldsymbol{\eta}}\right) \tag{51}$$

which leads to:

$$\boldsymbol{\Upsilon}^{\boldsymbol{\vartheta}^{\boldsymbol{\chi}},\boldsymbol{\vartheta}^{\boldsymbol{\chi}}} = \boldsymbol{\Upsilon}^{\boldsymbol{\vartheta}^{\boldsymbol{\gamma}},\boldsymbol{\vartheta}^{\boldsymbol{\gamma}}} = \boldsymbol{\Upsilon}^{\boldsymbol{\vartheta}^{\boldsymbol{\eta}},\boldsymbol{\vartheta}^{\boldsymbol{\eta}}} = 2\boldsymbol{\Upsilon}^{\boldsymbol{\vartheta}^{\boldsymbol{\varkappa}},\boldsymbol{\vartheta}^{\boldsymbol{\varkappa}}} \tag{52}$$

and the variables $\boldsymbol{\vartheta}^{\boldsymbol{\chi}}$, $\boldsymbol{\vartheta}^{\boldsymbol{\gamma}}$ and $\boldsymbol{\vartheta}^{\boldsymbol{\eta}}$ are used to obtain the noised observation data $\boldsymbol{y}^{\boldsymbol{\chi}}$, $\boldsymbol{y}^{\boldsymbol{\gamma}}$ and $\boldsymbol{y}^{\boldsymbol{\eta}}$ respectively.

5. **Kalman Gain computation:** The Kalman Gain matrix is computed to estimate the updated state:

$$\widetilde{\boldsymbol{K}} = \boldsymbol{\Upsilon}^{\boldsymbol{x}_{i,k}^{\boldsymbol{\varkappa}^f},\widetilde{\mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\varkappa}^f}\right)}}\left(\boldsymbol{\Upsilon}^{\widetilde{\mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\varkappa}^f}\right)},\widetilde{\mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\varkappa}^f}\right)}} + \boldsymbol{\Upsilon}^{\boldsymbol{\vartheta}^{\boldsymbol{\varkappa}},\boldsymbol{\vartheta}^{\boldsymbol{\varkappa}}}\right)^{-1} \tag{53}$$

6. **Ensemble updates:** Principal, control and ancillary variables are updated:

$$\boldsymbol{x}_{i,k}^{\boldsymbol{\chi}^a} = \boldsymbol{x}_{i,k}^{\boldsymbol{\chi}^f} + \widetilde{\boldsymbol{K}}\left(\boldsymbol{y}_{i,k}^{\boldsymbol{\chi}} - \mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\chi}^f}\right)\right) \tag{54}$$

$$\boldsymbol{x}_{i,k}^{\boldsymbol{\gamma}^a} = \boldsymbol{x}_{i,k}^{\boldsymbol{\gamma}^f} + \boldsymbol{\mathcal{P}}^-\left(\widetilde{\boldsymbol{K}}\left(\boldsymbol{y}_{i,k}^{\boldsymbol{\gamma}} - \mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\gamma}^f}\right)\right)\right) \tag{55}$$

$$\boldsymbol{x}_{i,k}^{\boldsymbol{\eta}^a} = \boldsymbol{x}_{i,k}^{\boldsymbol{\eta}^f} + \boldsymbol{\mathcal{P}}^-\left(\widetilde{\boldsymbol{K}}\left(\boldsymbol{y}_{i,k}^{\boldsymbol{\eta}} - \mathcal{H}\left(\boldsymbol{x}_{i,k}^{\boldsymbol{\eta}^f}\right)\right)\right) \tag{56}$$

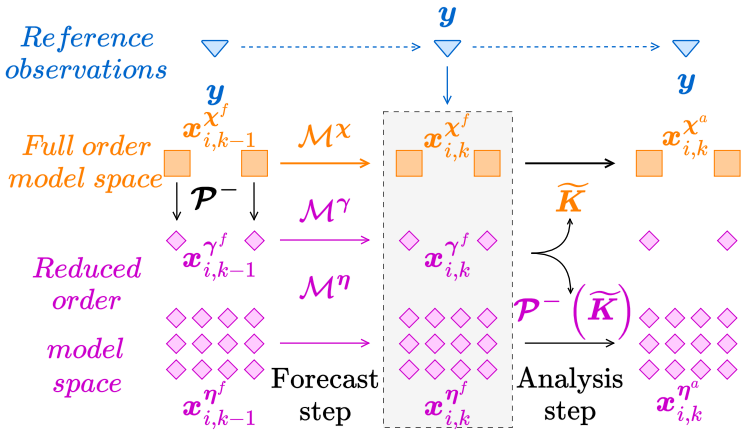


Fig. 27: Schematic representation of the MFEnKF algorithm

Acknowledgments

This research work was carried out using TGCC storage and computing resources on the HPC Joliot Curie's Skylake partition, in the framework of the GENCI grant A020A01741. The present work is funded by the French Ministry of the Armed Forces - Defense Innovation Agency (AID) and by the Hauts-de-France Region.

References

[1] Day, I.J.: Stall, surge, and 75 years of research. Journal of Turbomachinery **138**(1), 011001 (2016) https://doi.org/10.1115/1.4031473

[2] Lord, W.K., Suciu, G.L.: Engine architecture for high efficiency at small core size. Mechanical Engineering **138**(03), 52–53 (2016) https://doi.org/10.1115/1.2016-Mar-4 . Pratt & Whitney

[3] Liu, E., Zhou, Z., Zheng, H.: Control of high-load compressor stator corner flow by blade slotting. International Journal of Aerospace Engineering **2025**(1), 3743172 (2025) https://doi.org/10.1155/ijae/3743172

[4] Kumar, A., Kumar, A., Chhugani, H., More, S., Pradeep, A.M.: Understanding the effect of three-dimensional design in tandem blade. Journal of Turbomachinery **146**(3), 031002 (2024) https://doi.org/10.1115/1.4063924

[5] Li, J., Du, J., Nie, C., Zhang, H.: Review of tip air injection to improve stall margin in axial compressors. Progress in Aerospace Sciences **106**, 15–31 (2019) https://doi.org/10.1016/j.paerosci.2019.01.005

[6] Rao, G.A., Mahulikar, S.P.: Integrated review of stealth technology and its role in airpower. The Aeronautical Journal **106**(1066), 629–642 (2002) https://doi.org/

10.1017/S0001924000011702

[7] Yildirim, A., Gray, J.S., Mader, C.A., Martins, J.R.R.A.: Boundary-layer ingestion benefit for the starc-abl concept. Journal of Aircraft **59**(4), 896–911 (2022) https://doi.org/10.2514/1.C036103

[8] Cousins, W.T.: History, philosophy, physics, and future directions of aircraft propulsion system/inlet integration. Turbo Expo, vol. Volume 2: Turbo Expo 2004, pp. 305–320 (2004). https://doi.org/10.1115/GT2004-54210

[9] Camp, T.R., Day, I.J.: A study of spike and modal stall phenomena in a Low-Speed axial compressor. Journal of Turbomachinery **120**(3), 393–401 (1998) https://doi.org/10.1115/1.2841730

[10] Wilson, K.N., Jaman, G.G., Thapa, A., Vivekananda, A., Lowe, M., Grima, Z., Schoen, M.P.: A review of stall detection in subsonic axial compressors. Machines **13**(1) (2025) https://doi.org/10.3390/machines13010013

[11] Pope, S.B.: Turbulent Flows. Cambridge University Press, Cambridge (2000). https://doi.org/10.1017/CBO9780511840531

[12] Sagaut, P.: Large Eddy Simulation for Incompressible Flows: An Introduction, 3rd edn. Scientific Computation. Springer, Berlin, Heidelberg (2006). https://doi.org/10.1007/b137536

[13] Lumley, J.L.: Stochastic Tools in Turbulence. Applied Mathematics and Mechanics, vol. 12. Academic Press, New York (1970)

[14] Wilcox, D.C.: Turbulence Modeling for CFD, 3rd edn. DCW Industries, La Cañada, CA (2006)

[15] Duraisamy, K., Xiao, H., Iaccarino, G.: Turbulence modeling in the age of data. Annual Review of Fluid Mechanics **51**, 1–23 (2019)

[16] Schmelzer, M., Dwight, R., Cinnella, P.: Discovery of algebraic Reynolds-stress models using sparse symbolic regression. Flow, Turbulence and Combustion **104**, 579–603 (2020)

[17] Volpiani, P.S., Meyer, M., Franceschini, L., Dandois, J., Renac, F., Martin, E., Marquet, O., Sipp, D.: Machine learning-augmented turbulence modeling for RANS simulations of massively separated flows. Physical Review Fluids **6**, 064607 (2021)

[18] Daley, R.: Atmospheric Data Analysis. Cambridge Atmospheric and Space Science Series. Cambridge University Press, Cambridge (1991)

[19] Asch, M., Bocquet, M., Nodet, M.: Data Assimilation: Methods, Algorithms, and Applications. Fundamentals of Algorithms, vol. 11. SIAM, Philadelphia, PA

(2016). https://doi.org/10.1137/1.9781611974546

[20] Evensen, G., Vossepoel, F.C., van Leeuwen, P.J.: Data Assimilation Fundamentals. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-96709-3

[21] Simon, D.: Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches. Wiley, Hoboken, NJ (2006). https://doi.org/10.1002/0470045345

[22] Evensen, G.: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. Journal of Geophysical Research **99**(C5), 10143 (1994) https://doi.org/10.1029/94JC00572

[23] Evensen, G.: The ensemble kalman filter for combined state and parameter estimation - monte carlo techniques for data assimilation in large systems. IEEE Control Systems **29**, 83–104 (2009)

[24] Popov, A.A., Mou, C., Sandu, A., Iliescu, T.: A Multifidelity Ensemble Kalman Filter with Reduced Order Control Variates. SIAM Journal on Scientific Computing **43**(2), 1134–1162 (2021) https://doi.org/10.1137/20M1349965

[25] Villiers, R., Mons, V., Sipp, D., Lamballais, E., Meldi, M.: Enhancing unsteady reynolds-averaged navier–stokes modelling from sparse data through sequential data assimilation and machine learning. Flow, Turbulence and Combustion **115**, 991–1029 (2025)

[26] Valero, M.M., Meldi, M.: Improved state estimation for turbulent flows combining ensemble data assimilation and machine learning. Computers & Fluids **300**, 106757 (2025)

[27] Moussie, T., Errante, P., Meldi, M.: Multi-level ensemble Kalman Filter algorithms enhanced by Convolutional Neural Networks. Computers and Fluids **314**, 107082 (2026)

[28] Moubogha, J.M., Joseph, P., Roussette, O., Dazin, A.: Experimental flow investigation inside an axial compressor with active flow control. In: Turbo Expo: Power for Land, Sea, and Air (2022). https://doi.org/10.1115/GT2022-82126 . American Society of Mechanical Engineers

[29] Baretter, A., Joseph, P., Roussette, O., Dazin, A., Romanò, F.: Scaling laws at stall in an axial compressor with an upstream perturbation. Journal of Propulsion and Power **41**(1), 82–98 (2025) https://doi.org/10.2514/1.B39407

[30] Alekseik, O., Joseph, P., Roussette, O., Dazin, A.: An experimental investigation by particle image velocimetry of the active flow control of the stall inception of an axial compressor. International Journal of Turbomachinery, Propulsion and Power **10**(4) (2025) https://doi.org/10.3390/ijtpp10040040

[31] Ferziger, J.H., Perić, M.: Computational Methods for Fluid Dynamics. Springer, Berlin, Heidelberg (1996). https://doi.org/10.1007/978-3-642-97651-3

[32] Spalart, P., Allmaras, S.: A one-equation turbulence model for aerodynamic flows. In: 30th Aerospace Sciences Meeting and Exhibit, p. 439 (1992)

[33] Wilhelm, D.: Rotating flow simulations with openfoam. Int J Aeronautics Aerospace Res. S **1**, 1–7 (2015)

[34] Spalart, P.R., Deck, S., Shur, M.L., Squires, K.D., Strelets, M.K., Travin, A.: A New Version of Detached-eddy Simulation, Resistant to Ambiguous Grid Densities. Theoretical and Computational Fluid Dynamics **20**(3), 181–195 (2006) https://doi.org/10.1007/s00162-006-0015-0

[35] Carneiro, F., Moura, L., Rocha, P.C., Lima, R.P., Ismail, K.: Application and analysis of the moving mesh algorithm ami in a small scale hawt: Validation with field test's results against the frozen rotor approach. Energy **171**, 819–829 (2019)

[36] Greenshields, C., Weller, H.: Notes on Computational Fluid Dynamics: General Principles. CFD Direct Ltd, Reading, UK (2022)

[37] Caretto, L.S., Gosman, A.D., Patankar, S.V., Spalding, D.B.: Two calculation procedures for steady, three-dimensional flows with recirculation. In: Cabannes, H., Temam, R. (eds.) Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics, pp. 60–68. Springer, Berlin, Heidelberg (1973)

[38] Issa, R.I., Gosman, A.D., Watkins, A.P.: The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme. Journal of Computational Physics **62**(1), 66–82 (1986) https://doi.org/10.1016/0021-9991(86)90100-2

[39] Jeanney, P., Hetherington, A., Ahmed, S.E., Lanceta, D., Saiz, S., Perez, J.M., Le Clainche, S.: Ensemble kalman filter for data assimilation coupled with low-resolution computations techniques applied in fluid dynamics. Applied Mathematics and Computation **531**, 130184 (2026)

[40] Donato, L., Kamal, M.M., Procacci, A., Cafiero, M., Sharma, S., Galletti, C., Coussement, A., Parente, A.: Integrating data assimilation and sparse sensing for updating a digital twin of a semi-industrial furnace. Proceedings of the Combustion Institute **40**(1-4), 105284 (2024)

[41] He, C., Liu, Y.: Time-resolved reconstruction of turbulent flows using linear stochastic estimation and sequential data assimilation. Physics of Fluids **32**(7) (2020)

[42] Artana, G., Camilleri, A., Charlier, J., Memin, E.: Strong and weak constraint variational assimilations for reduced order fluid flow modeling. Journal of Computational Physics **231**, 3264–3288 (2012)

[43] Foures, D.P.G., Dovetta, N., Sipp, D., Schmid, P.J.: A data-assimilation method for Reynolds-averaged Navier-Stokes-driven mean flow reconstruction. Journal of Fluid Mechanics **759**, 404–431 (2014)

[44] Onder, A., Meyers, J.: Optimal control of a transitional jet using a continuous adjoint method. Computers & Fluids **126**, 12–24 (2016)

[45] Mons, V., Chassaing, J.C., Gomez, T., Sagaut, P.: Reconstruction of unsteady viscous flows using data assimilation schemes. Journal of Computational Physics **316**, 255–280 (2016)

[46] Chandramouli, P., Memin, E., Heitz, D.: 4D large scale variational data assimilation of a turbulent flow with a dynamics error model. Journal of Computational Physics **412**, 109446 (2020)

[47] Padmanaban, U.C., Ganapathisubramani, B., Symon, S.: Three-dimensional variational data assimilation of separated flows using time-averaged experimental data. Physical Review Fluids **11**, 014904 (2026)

[48] Rochoux, M.C., Ricci, S., Lucor, D., Cuenot, B., Trouve, A.: Towards predictive data-driven simulations of wildfire spread - Part I: Reduced-cost Ensemble Kalman Filter based on a Polynomial Chaos surrogate model for parameter estimation. Natural Hazards and Earth System Sciences **14**, 2951–2973 (2014)

[49] Leroux, R., Chatellier, L., David, L.: Bayesian inference applied to spatio-temporal reconstruction of flows around a NACA0012 airfoil. Experiments in Fluids **55**, 1699 (2014)

[50] Kato, H., Yoshizawa, A., Ueno, G., Obayashi, S.: A data assimilation methodology for reconstructing turbulent flows around aircraft. Journal of Computational Physics **283**, 559–581 (2015)

[51] Meldi, M., Poux, A.: A reduced order model based on Kalman Filtering for sequential Data Assimilation of turbulent flows. Journal of Computational Physics **347**, 207–234 (2017)

[52] Labahn, J.W., Wu, H., Coriton, B., Frank, J.H., Ihme, M.: Data assimilation using high-speed measurements and LES to examine local extinction events in turbulent flames. Proceedings of the Combustion Institute **37**, 2259–2266 (2019)

[53] Zhang, X., Michelin-Ströfer, C., Xiao, H.: Regularized ensemble Kalman methods for inverse problems. Journal of Computational Physics **416**, 109517 (2020)

[54] Villanueva, L., Valero, M.M., Šarkić-Glumac, A., Meldi, M.: Augmented state estimation of urban settings using on-the-fly sequential data assimilation. Computers & Fluids **269**, 106118 (2024)

[55] Moussie, T., Errante, P., Meldi, M.: Statistical inference of upstream turbulence intensity for the flow around a bluff body with massive separation. Flow, Turbulence and Combustion **113**, 853–889 (2024)

[56] Valero, M.M., Meldi, M.: An immersed boundary method using online sequential data assimilation. Journal of Computational Physics **524**, 113697 (2025)

[57] Villanueva, L., Truffin, K., Borée, J., Meldi, M.: Enhancement of large-eddy simulations for the prediction of an intake flow rig using sequential data assimilation. Journal of Fluid Mechanics **1023**, 47 (2025)

[58] Özalp, E., Nóvoa, A., Magri, L.: Real-time forecasting of chaotic dynamics from sparse data and autoencoders. Computer Methods in Applied Mechanics and Engineering **450**, 118600 (2026)

[59] Villanueva, L., Truffin, K., Meldi, M.: Synchronization and optimization of large eddy simulation using an online ensemble kalman filter. International Journal of Heat and Fluid Flow **110**, 109597 (2024)

[60] Moldovan, G., Mariotti, A., Lehnasch, G., Cordier, L., Salvetti, M.V., Meldi, M.: Multigrid sequential data assimilation for the large-eddy simulation of a massively separated bluff-body flow. Computers and Fluids **281**, 106385 (2024)

[61] Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering **82**(1), 35–45 (1960) https://doi.org/10.1115/1.3662552

[62] Moldovan, G., Lehnasch, G., Cordier, L., Meldi, M.: A multigrid/ensemble Kalman filter strategy for assimilation of unsteady flows. Journal of Computational Physics **443**, 110481 (2021)

[63] Moldovan, G., Lehnasch, G., Cordier, L., Meldi, M.: Optimized parametric inference for the inner loop of the Multigrid Ensemble Kalman Filter. Journal of Computational Physics **471**, 111621 (2022)

[64] Rannou, C., Dazin, A., Marty, J., Tanguy, G., Castillon, L., Moubogha, J.: Effect of the Axial Compressor Tip Clearance Size: Performance and Transition to Rotating Stall. In: Proceedings of the ASME Turbo Expo 2022: Turbomachinery Technical Conference and Exposition, vol. 10A: Turbomachinery - Axial Flow Fan and Compressor Aerodynamics. American Society of Mechanical Engineers, Rotterdam, Netherlands (2022). V10AT29A011

[65] Marty, J., Castillon, L., Joseph, P.: Numerical investigations on the rotating stall

in an axial compressor and its control by flow injection at casing. Journal of Turbomachinery **145**(5), 051009 (2023)

[66] Rannou, C.: Méthodologie pour la modélisation du contrôle actif d'écoulement dans un compresseur axial. PhD thesis, HESAM Université (2023)

[67] Errante, P., Meldi, M., Moussie, T., Martinez-Valero, M., Er, S., Villanueva, L.: CONES. Zenodo (2026). https://doi.org/10.5281/zenodo.20433499 . https://doi.org/10.5281/zenodo.20433499

[68] Cybenko, G.: Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems **2**(4), 303–314 (1989)

[69] Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., Aljaaf, A.J.: A systematic review on supervised and unsupervised machine learning algorithms for data science. Supervised and unsupervised learning for data science, 3–21 (2020)

[70] Alsina, E.F., Chica, M., Trawiński, K., Regattieri, A.: On the use of machine learning methods to predict component reliability from data-driven industrial case studies. The International Journal of Advanced Manufacturing Technology **94**(5), 2419–2433 (2018)

[71] Kapp, S., Choi, J.-K., Hong, T.: Predicting industrial building energy consumption with statistical and machine-learning models informed by physical system parameters. Renewable and Sustainable Energy Reviews **172**, 113045 (2023)

[72] Sharma, S., Sharma, S., Athaiya, A.: Activation functions in neural networks. Towards Data Sci **6**(12), 310–316 (2017)

[73] Vinutha, H., Poornima, B., Sagar, B.: Detection of outliers using interquartile range technique from intrusion dataset. In: Information and Decision Sciences: Proceedings of the 6th International Conference on Ficta, pp. 511–518 (2018). Springer

[74] Jais, I.K.M., Ismail, A.R., Nisa, S.Q.: Adam Optimization Algorithm for Wide and Deep Neural Network. Knowledge Engineering and Data Science **2**(1), 41 (2019) https://doi.org/10.17977/um018v2i12019p41-46

[75] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org (2015). https://www.tensorflow.org/

[76] Chollet, F., et al.: Keras. https://keras.io (2015)

[77] Chicco, D., Warrens, M.J., Jurman, G.: The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. Peerj computer science **7**, 623 (2021)

[78] Gropp, W., Lusk, E., Skjellum, A.: Using MPI: Portable Parallel Programming with the Message-passing Interface. MIT Press, Cambridge, MA, USA (1994)

[79] Barker, B.: Message passing interface (mpi). In: Workshop: High Performance Computing on Stampede, vol. 262 (2015). Cornell University Publisher Houston, TX, USA

[80] Masoudnia, S., Ebrahimpour, R.: Mixture of experts: a literature survey. Artificial Intelligence Review **42**(2), 275–293 (2014)

[81] Oulghelou, M., Cherroud, S., Merle, X., Cinnella, P.: Machine-learning-assisted blending of data-driven turbulence models. Flow, Turbulence and Combustion **115**(3), 1095–1132 (2025)

[82] Rubinstein, R.Y., Marcus, R.: Efficiency of Multivariate Control Variates in Monte Carlo Simulation. Operations Research **33**(3), 661–677 (1985) https://doi.org/10.1287/opre.33.3.661