



HAL
open science

Data-driven order reduction and velocity field reconstruction using neural networks: The case of a turbulent boundary layer

Antonios Giannopoulos, Jean-Luc Aider

► **To cite this version:**

Antonios Giannopoulos, Jean-Luc Aider. Data-driven order reduction and velocity field reconstruction using neural networks: The case of a turbulent boundary layer. *Physics of Fluids*, 2020, 32 (9), pp.095117. 10.1063/5.0015870 . hal-03006985

HAL Id: hal-03006985

<https://hal.science/hal-03006985v1>

Submitted on 9 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Data-driven order reduction and velocity field reconstruction using Neural Networks: the case of a turbulent boundary layer

Antonios Giannopoulos^{1,2, a)} and J.-L. Aider^{1, b)}

¹⁾Laboratoire de Physique et Mécanique des Milieux hétérogènes (PMMH), UMR7636 CNRS, ESPCI Paris, PSL Research University, Sorbonne Université, Univ. Paris Diderot, 1 rue Jussieu, 75005 Paris, France.

²⁾Photon Lines Recherche, Parc Pereire Bât B, 99 rue Pereire, 78100 – Saint-Germain-en-Laye, France

(Dated: 2 June 2020)

We present a data-driven methodology to achieve identification of coherent structures dynamics and system order reduction of an experimental turbulent boundary layer (TBL) flow. The flow is characterized using time-resolved Optical Flow Particle Image Velocimetry, leading to dense velocity fields that can be used both to monitor the overall dynamics of the flow and to define as many local visual sensors as needed. A Proper Orthogonal Decomposition (POD) is first applied to define a reduced-order system. A non-linear mapping between the local upstream sensors (inputs-sensors) and the full-field dynamics (POD coefficients) as outputs is sought using an optimal Focused Time-Delay (FTD) Artificial Neural Network (ANN). The choices of sensors, ANN architecture and training parameters are shown to play a critical role. It is verified that a shallow ANN, with the proper sensor memory size, can lead to a satisfying full-field dynamics identification, coherent structure reconstruction, and system order reduction of this turbulent flow.

I. INTRODUCTION

Current experimental methods such as Particle Image Velocimetry (PIV) or numerical methods such as Large Eddy Simulations (LES), used to characterize 3D non-stationary flows, produce a large amount of data leading to high-order systems. It is generally necessary to use an order reduction to derive some information that could be handled either to model or control the targeted flow. Data-driven methods are nowadays becoming more and more efficient and reliable even for fluid mechanics research^{9,17}. Among successful applications, one can cite statistical learning¹⁶ or machine learning^{14,35} algorithms. More recently Artificial Neural Networks (ANNs) have been used extensively for fluid mechanics, because of their architecture flexibility, needed for reduced-order modelling²⁹, as well as for field reconstruction of complex non-linear flows²². A short review on different applications of Deep Learning to Fluid Dynamics can be found in²⁷. Nevertheless, applying identification or modelling algorithms to experimental fluid mechanics is still challenging because of various noises leading to non physical errors as well as a limited number of sensors and the complexity of the experimental setup. Large datasets with both good spatial and temporal resolutions are needed for successful applications of these data-hungry methods to experiments.

In an inverted flag flow experiment, Deng et al.⁹ applied an ANN identification method to reconstruct time-resolved velocity fields from a handful of velocity sensors. It was also proven recently that ANNs can be used to predict the dynamics and reconstruct the time-resolved fields of an experimental Backward-Facing Step (BFS) flow¹⁵. We apply in this study a similar methodology to an experimental TBL flow, with the objective to identify correctly the dynamics, reconstruct the

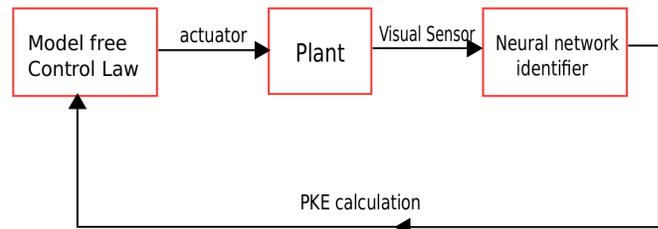


FIG. 1. Control scheme for minimizing perturbation kinetic energy using a neural-network identifier

velocity field and predict the total Perturbation Kinetic Energy (PKE) of the full velocity field. The ANN algorithm will be used first to identify the global dynamics from upstream local sensors and second to reduce drastically the order of the system to a handful of easily measurable quantities. Such an algorithm would be an ideal part of a model-free control loops¹⁴ aiming at reducing the PKE field, as shown in the Fig. 1.

A. Artificial Neural Networks

Artificial Neural Networks (ANNs) can provide a non-linear mapping between one set of inputs-signals (that are easily measurable) and some corresponding output signals (that may be hard to measure in a final application, but still valuable for monitoring or control purposes). Great progresses have been made lately due to the ability to handle very large datasets, the advancements of Graphics Processing Units (GPU) parallel programming, and the increasing availability of multiple optimized toolboxes. A short review of applications of

^{a)}Electronic mail: antonios.giannopoulos@espci.fr

^{b)}Electronic mail: jean-luc.aider@espci.psl.eu

deep ANN to fluid mechanics can be found in Nathan Kutz²⁷. ANNs can be divided into Feed-Back (or recurrent) and Feed-Forward. They can also be discriminated according to their depth, either shallow or deep, depending on the number of hidden layers (one or more). Finally, they can be divided into static or dynamic, if the output of the current step depends on the previous steps as well, giving it a notion of memory.

In the case of a feed-forward ANNs, the output of any layer only modifies the next layer, not the previous ones. On the other hand, in a recurrent ANN the system output is calculated from its previous past time-steps along with the input at the current time-step (eq. 1), hence introducing a notion of output memory in the network.

$$y_t = f_{w,b}(y_{t-1}, x_t) \quad (1)$$

One category of recurrent ANNs are the Long-Short Term Memory (LSTM) networks. They are gradient-based recurrent NNs used for supervised learning both in classification and prediction problems³³. They were first developed by Hochreiter and Schmidhuber¹⁸ to solve the vanishing or exploding gradient problem of the back-propagated error. In an LSTM architecture, the network is left to learn alone the size of the memory of each neuron during the training process. They do so using a sophisticated gate-neuron that determines if the input is important enough or if it should be forgotten and when it should output its value.

B. Focused time-delay Artificial Neural Networks

The neural network used in the present System Identification (SI) study is a fully-connected FTD ANN, which was first introduced for speech recognition³⁴. It is basically a standard feed-forward architecture along with a tapped constant time-delay (of time-step size k) in the input (see Fig. 2). The term "focused" comes from the fact that the notion of memory is introduced only in the input (sensors), and not in the output like in LSTM ANNs. They are used to model long-range temporal dependencies by keeping a number of k past measurements of the input at each time step x_t . This leads to the following expression for the output of the system :

$$y_t = f_{w,b}(x_t, x_{t-1}, \dots, x_{t-k}) \quad (2)$$

where w and b are weight and bias parameters. They have been used for rainfall prediction⁶ as well as for effective acoustic modelling¹⁹, in deeper architectures.

For the above network with a non-linear activation function f_1 in the hidden layer and a linear activation function f_2 in the linear layer, the equation giving the k_{th} neuron output of a single hidden layer network connected to the j_{th} neuron of the previous layer is:

$$y_k = \mathbf{f}_1 \left(\sum_{j=0}^{N_2} w_{kj}^{(2)} \mathbf{f}_2 \left(\sum_{i=0}^{N_1} (w_{ki}^{(1)} x_i + b_i) \right) + b_j \right) \quad (3)$$

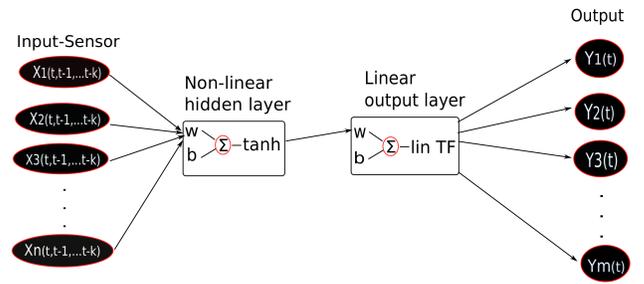


FIG. 2. An example of a FTD ANN to perform a non-linear mapping from N inputs to M outputs.

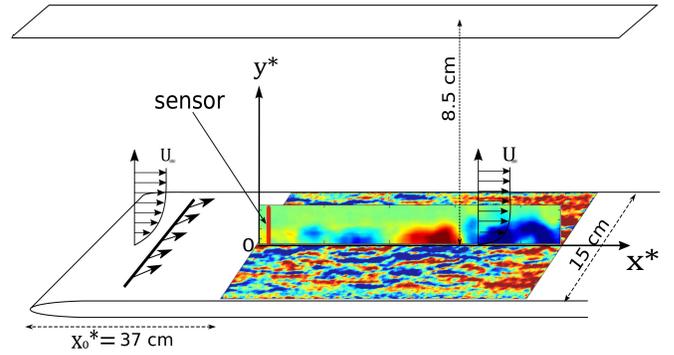


FIG. 3. Flat plate geometry dimensions, perturbation slot and PIV window position.

where N_1 is the number of neurons in the first (hidden) non-linear layer and N_2 the number of neurons in the second (linear) layer. Regarding the choice of the non-linear activation function f_1 , the tan-sigmoid or hyperbolic tangent function is used:

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (4)$$

The correct training process of the ANN (meaning finding the optimum weights and biases connecting the neurons of different layers) consists in dividing the data-set between a training data-set and a validation data-set. For the training data-set we chose one set of weights (randomly initialized in the first iteration) connecting the layers and we use the same weights for each time step to calculate the output of the model. The error of the real vs the model-generated output signal is computed and the weights and biases are updated according to different back-propagation schemes (in our case the Scaled Conjugate Gradient method). The above process is called one "epoch". We continue the process for as many epochs as needed until a satisfactory fit error is achieved. The second data-set is used to test the performance of the network on new data and verify the achieved error, hence avoiding overfitting.

In the following, we first present the experimental setup as well the main characteristics of our Turbulent Boundary Layer

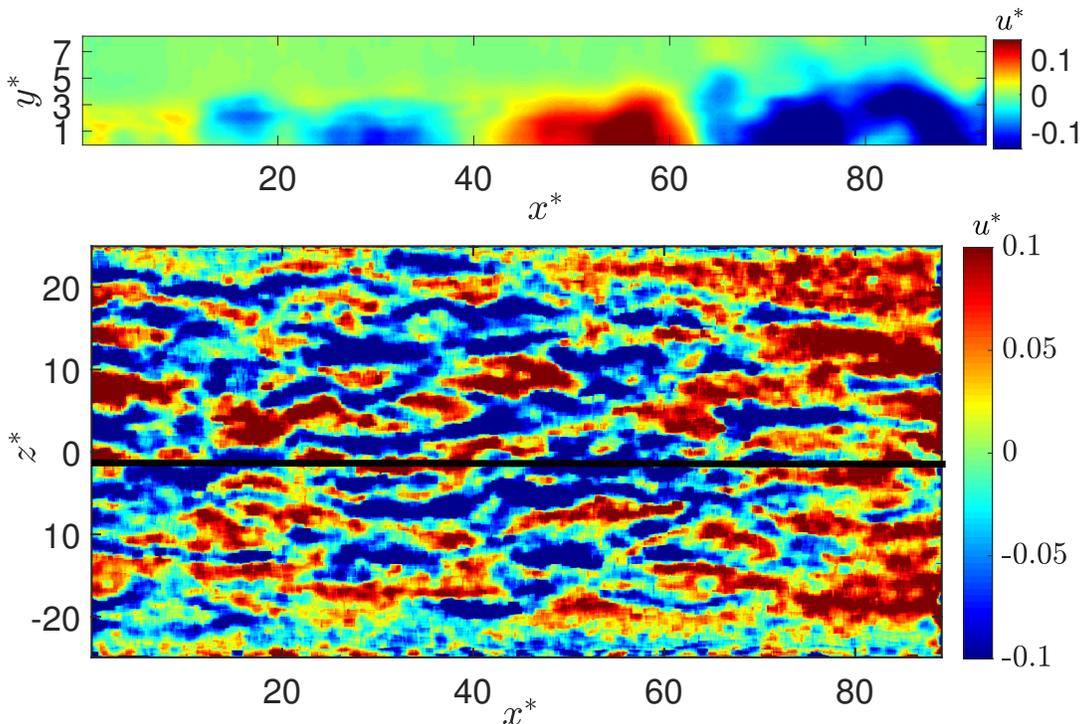


FIG. 4. Instantaneous streamwise velocity fluctuation field of the TBL experiment. Vertical field for $z^* = 0$ (top) and horizontal field for plane $y = 2 d_0^*$ (bottom). Streamwise velocity streaks can be clearly observed.

(TBL) flow. The choice of the sensors of the ANN is then discussed and parametric studies of different training parameters are presented. The efficiency of the ANN for the dynamics identification and velocity field reconstruction is illustrated before turning to the discussion and conclusion.

II. EXPERIMENTAL SETUP

A. Hydrodynamic channel

Experiments were carried out in an hydrodynamic channel in which the flow is driven by gravity. The flow is stabilized by divergent and convergent sections separated by honeycombs leading to a turbulence intensity of 0.6 %. A NACA 0020 profile is used to smoothly start the boundary layer. The test section is 80 cm long with a rectangular cross-section 15 cm wide and 8.5 cm high as shown in Fig. 3.

B. Time-resolved Particle Image Velocimetry measurements

The flow is seeded with 20 μm neutrally buoyant polyamide particles, illuminated by a laser sheet created by a 10 W continuous laser (MX-6185, Coherent, USA) operating at 532 nm. A thin layer of fluorescent paint (FP Rhodamine 6G, Dantec) was applied to the illuminated surface, absorbing the laser wavelength to avoid reflections and to allow cor-

rect near-wall measurements. The camera used was a PCO DIMAX-cs with a sampling frequency of 150 Hz. A narrow-band optical filter centered around 532 nm was mounted on the camera to avoid any optical noise.

The velocity field is calculated from the acquisition of successive snapshots of the vertical laser sheet in the middle of the test section using a home-made optical-flow algorithm. The first version of the code has been developed at ONERA⁵ and later modified, optimized and adapted to the constraints of real-time measurements¹³. The advantage of this algorithm compared to a standard FFT-PIV algorithm is its high computational speed when implemented on GPUs with CUDA functions⁸. The code has been used many times both for time-resolved PIV measurements with a high spatial resolution³¹ as well as for closed-loop flow control experiments^{10–12,32}.

The vertical (x, y) PIV plane was chosen for the present identification because a larger experimental dataset was available. Indeed, only the boundary layer region contains relevant information for the identification algorithm so that only the lower part of the snapshots are used, which allows a significant reduction of computational time. Also, the distribution of streaks in the spanwise direction was found homogeneous for the specific forcing. The extension of the method in the horizontal plane and for different boundary layer forcing scenarios could also be investigated.

The parameters of the PIV experiments used in the following are summarized in table I. One should keep in mind that the optical flow algorithm leads to dense vector fields with one

PIV length(cm)	23.39
Snapshot size	162x1978
Spatial resolution	0.116 mm/pixel
# Snapshots	6400
Acquisition time(s)	41.3

TABLE I. PIV experiment parameters.

vector per pixel.

III. FLOW CHARACTERISATION

In the present study we focus on a specific transition scenario of the zero pressure gradient flat plate boundary layer flow. This flow can experience various mechanisms for its transition to turbulence. In one scenario, when the incoming turbulence intensity is lower than 1% initial environment disturbances are amplified via a receptivity process²⁰ and are carried downstream by the flow with increasing amplitude. A breakdown of the laminar flow occurs due to a selective amplification of a narrow band of frequencies which creates the so-called Tollmien-Schlichting waves. The amplitude of the waves increases up to the point where unstable 3D peaks and valleys occur until they breakdown into hairpin eddies and turbulent spots. These wave-like structures are periodic in space and time. The critical Reynolds number based on the local displacement thickness is defined as:

$$Re_{\delta_0^*} = \frac{\delta_0^*(x)U_\infty}{\nu} \quad (5)$$

where U_∞ is the free-stream velocity, ν is the kinematic viscosity, x the streamwise coordinate (y and z being respectively the vertical and spanwise axis) and

$$\delta_0^* = \int_0^\infty \left(1 - \frac{u}{U_\infty}\right) dy \quad (6)$$

is the local displacement thickness at the inlet of the PIV window, at $x_0 = 37$ cm from the leading edge. The critical Reynolds number is $Re_{\delta_0^*}^{crit} = 520$.

A different transition scenario may occur if the turbulence intensity of the incoming flow is order of 1%²⁶ or higher: the bypass transitions. Groups of streaks are usually observed, which are streamwise structures periodically distributed along the spanwise direction^{1,24}, for Reynolds numbers as low as $Re_{\delta_0^*}(x) = 300$. It is called bypass, because it can bypass the TS instability mechanism altogether.

The dynamics identification and coherent structure reconstruction of a subcritical bypass transition is investigated in the present experiment. The transition is triggered using a finite amplitude pulsating jet (described in the following section). Groups of wavy streaks are observed as shown on Fig. 4. The shape factor H of the forced boundary layer is $H = 1.31$, which is typical of a turbulent flow. The coherent structure advection velocity u_{adv} has been estimated around $0.75 U_\infty$ (based on a space-time diagram of a horizontal profile of the streamwise velocity perturbation inside the boundary layer).

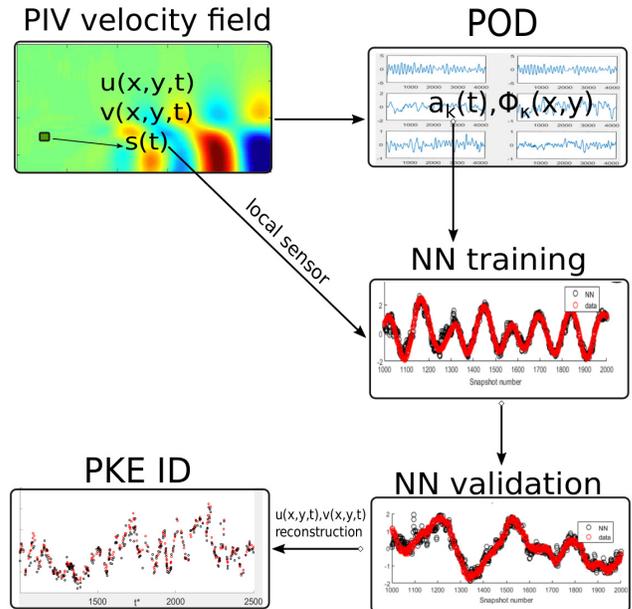


FIG. 5. Block diagram showing the different steps used for the experimental perturbation energy identification through POD and training/validation steps.

In the following all the spatial variables are non-dimensionalized using the displacement thickness at the beginning of the PIV window δ_0^* (at $x_0 = 37$ cm), for the unperturbed flow and noted (x^*, y^*, z^*) . The velocity components are non-dimensionalized with the freestream velocity U_∞ and noted (u^*, v^*, w^*) . Consequently, we refer to the dimensionless time $t^* = tU_\infty/\delta_0^*$.

A. Transition forcing mechanism

The unperturbed flow is a laminar boundary layer. The Reynolds number measured at the inlet of the PIV window is $Re_{\delta_0^*} = 409$ for a freestream velocity $U_\infty = 0.188 \text{ m.s}^{-1}$. No natural instability was observed without forcing. The displacement thickness at $x_0 = 37 \text{ cm}$ is $\delta_0^* = 2.5 \text{ mm}$ leading to a shape factor $H_0 = 2.34$, typical of a laminar boundary layer.

In order to evaluate the potential of ANN on a non-linear flow, the transition to turbulence of the boundary layer is forced using a finite amplitude pulsating jet perturbation. The perturbation is induced by a spanwise slotted inclined (45°) jet as shown on Fig. 3. The injection system is optimized to obtain an homogeneous velocity along the spanwise direction. It is located at $x = 5 \text{ cm}$ downstream the leading edge, as shown in Fig. 3. The jet outlet has a rectangular cross-section which is 0.4 cm long (streamwise direction) and 9 cm wide (spanwise direction). The jet is supplied by a pressurized water tank, monitored by a SMC ITV 1010 electro-pneumatic regulator to control the jet frequency $f_{forcing}$ and amplitude $U_{forcing}$. The frequency of the perturbation was random ($0 < f_{forcing} < 5 \text{ Hz}$), for a constant maximum amplitude. The experiments were repeated multiple times show-

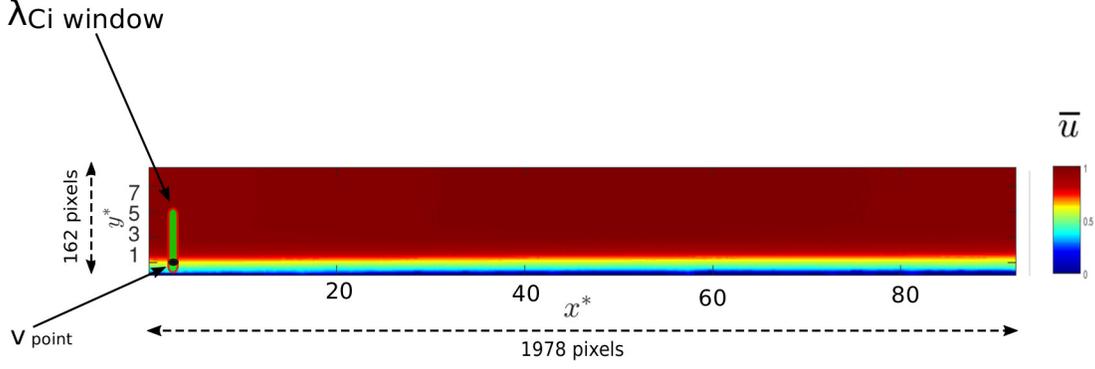


FIG. 6. Location of the double visual sensors shown over the time-averaged unperturbed streamwise velocity field \bar{u} .

ing the same Proper Orthogonal Decomposition (POD) mode structures.

B. Proper Orthogonal Decomposition

Decomposing a dynamical system in modes of decreasing importance can help reducing the order of the variables of the system. N consecutive instantaneous velocity fields $\{\mathbf{U}(n) = (u^*, v^*)\}_{n=1 \dots N}$ were computed from consecutive flow snapshots with an acquisition frequency $f_{ac} = 150 \text{ Hz}$. By calculating the mean field $[\bar{u}, \bar{v}]$ we were able to obtain the velocity fluctuations $u'(t) = u^*(t) - \bar{u}$ and $v'(t) = v^*(t) - \bar{v}$, which contained all the dynamics and were used to create the reduced-order system. The fluctuation matrices organised in columns for each time-step were used to form the so-called "snapshot matrix" to be decomposed. The reduced-order system is obtained using POD, which has been used extensively in fluid mechanics^{2,25}. It allows us to build a ranked and orthonormal basis containing N modes^{23,30}. The first M modes $\{\Phi_m\}_{m=1 \dots M}$ with $M \leq N$ containing a sufficient percentage of the total energy is then chosen to compute the approximated velocity field $\tilde{\mathbf{U}}(n)$:

$$\tilde{\mathbf{U}}(n) = \sum_{m=1}^M \langle \Phi_m, \mathbf{U}(n) \rangle \Phi_m = \sum_{m=1}^M a_m(n) \Phi_m \quad (7)$$

where the scalar product $\langle \cdot, \cdot \rangle$ is the energy-based inner product. The system output to be identified is obtained through the reduced state vector containing the M POD coefficients $a_m(n)$:

$$\mathbf{Y}(n) = [a_1(n) \ a_2(n) \ \dots \ a_M(n)]^T \quad (8)$$

The full-field dynamics are now contained in their POD coefficients $a_m(t)$. The balance between the order and accuracy of the POD reduced-order system is crucial, because for a large number of POD modes the SI methods are much more likely to fail.

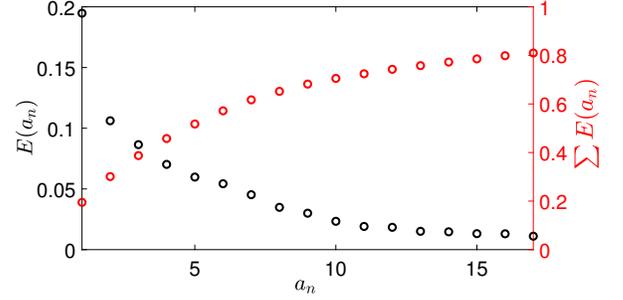


FIG. 7. Energy of each POD modes together with the cumulative total energy of the POD modes up to 80 % of the total energy.

C. Overview of the method

In the present SI method we first apply a POD to the time-resolved PIV snapshot matrix, to perform a first system order compression. Then we proceed to train a FTD ANN for the non-linear mapping between local upstream sensors and the dynamic POD coefficients. All the dynamics information is then included in a 2 easily measurable quantities. More specifically, the available time-resolved PIV dataset is divided into the learning or training part and the validation part. During the training step, the parameters of the model (weight, bias) are updated accordingly using an appropriate learning algorithm which takes into account the error between the expected output and the model-generated output via a back-propagation learning algorithm, in our case the scaled conjugate gradient. After each weight and bias update, an epoch has passed. As many epochs as needed are used to obtain a satisfying fit error. Then, during the validation step, the model generated outputs are compared with the new data, this time expecting equivalent fit results as during the training step. Larger validation fit errors yield over-fitting and make the network useless. The correctly trained network will allow the accurate prediction of the PKE at any moment using just a few local PIV sensors. The different steps of the overall identification process are summarized on the diagram shown in Fig. 5.

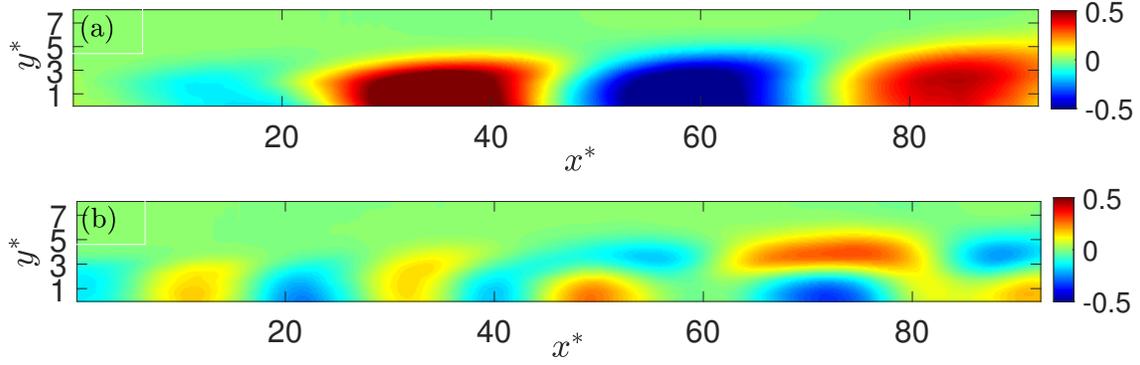


FIG. 8. 3rd (a) and 13th (b) POD mode structures.

IV. VALIDATION CRITERION

To evaluate the efficiency of the identification, one has to define a relevant quantitative criterion to compare the POD coefficient time-series results obtained with the different ANN architectures to the ones obtained experimentally. In the present study, we compute the mean-squared error (MSE) at each time-step n for each POD coefficient $a_m(n)$:

$$MSE_m = \frac{1}{N} \sum_{n=1}^N (a_{exp,m}(n) - a_{NN,m}(n))^2 \quad (9)$$

Then the averaged MSE for all the coefficients ($M = 17$) time-series gives the final evaluation error for the specific ANN architecture:

$$MSE = \frac{1}{M} \sum_{m=1}^M MSE_m \quad (10)$$

V. RESULTS

First, we obtained the full dataset of $N = 6400$ consecutive snapshots, acquired with a sampling frequency of 150 Hz . A POD was applied using the snapshot method. 17 POD modes were needed to obtain satisfying results with at least 80 % the total energy. The energy of the individual POD modes as well as their cumulative energy are shown in Fig. 7. The spatial structure of the third and 13th POD modes can be found in Fig. 8 while their time evolution is shown in Fig. 9. We clearly see the similarity of the 3rd POD mode with the large-scale coherent structures visualized in the instantaneous field of Fig. 4a).

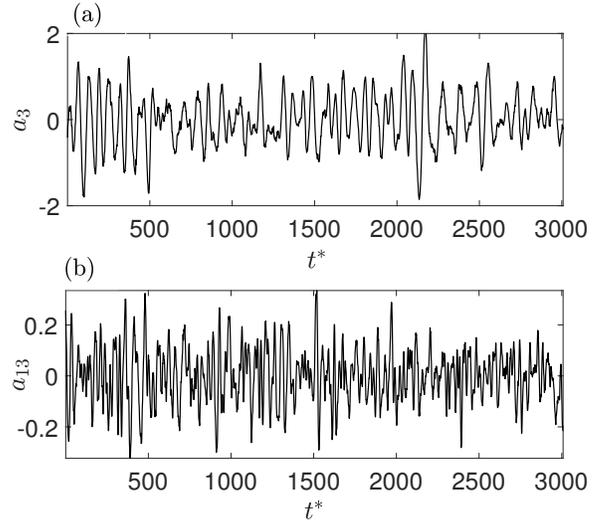


FIG. 9. Evolution of 3rd (a) and 13th (b) dynamic POD coefficient.

VI. SENSORS DEFINITION

A. Choice of the sensor

The visual sensors will be the inputs in the identification process. The choice of the sensor(s) is then a critical step. There are many possible type of sensors as well as many possible size and locations.

The first step is to choose the physical nature of the sensor which should be based on the two components of the instantaneous 2D velocity field measured at each time-step. It can be simply one of, or the two components of the velocity field, or the velocity magnitude measured in a given window defined in the PIV field. Because of their simplicity, they are the first sensors we will test.

As we are interested in the dynamics of coherent structures advected into the boundary layer, one can also choose to compute the vorticity field. As the vortical structures are embedded into the boundary layer, it is better to use more refined de-

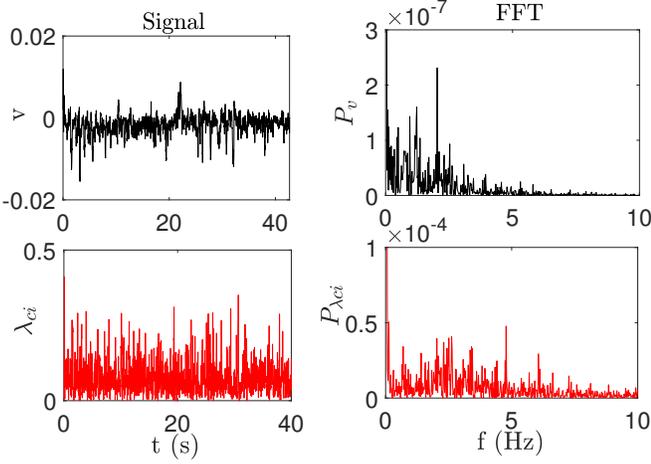


FIG. 10. Time-series and power spectrum of velocity (s_1) and swirling strength (s_2) sensor.

tection criteria, like the Q criterion or the λ_{Ci} criterion, which are well adapted to the identification of vortical structures inside a shear layer.

In the following, we will use the swirling strength λ_{Ci} criterion, which was first introduced by⁷ who analyzed the velocity gradient tensor $\mathcal{D} = \nabla \vec{u}$ and proposed that the vortex core could be defined as a region where $\nabla \mathbf{u}$ has complex conjugate eigenvalues. It was subsequently improved by³⁶ and by⁴. It was also successfully applied by³ to visualize the 3D vortices created by a Jet in Cross-Flow measured by Volumetric Velocimetry or by¹² in a closed-loop flow control experiment using a similar visual sensor. For 2D data, λ_{Ci} can be computed quickly and efficiently using eq. (11) when such a quantity is real (else $\lambda_{Ci} = 0$):

$$\lambda_{Ci} = \frac{1}{2} \sqrt{4 \det(\nabla \mathbf{u}) - \text{tr}(\nabla \mathbf{u})^2} \quad (11)$$

B. Number of sensors

On one hand, reducing the number of sensors leads to a reduction of the number of inputs (and outputs) which generally would make the training of the system simpler. On the other hand we may lose valuable dynamics information, helpful for the identification process. A compromise has to be found. A combination of two inputs has been tested: first, the wall-normal velocity measured at one point (averaged over the five neighbouring pixels) and second, the swirling strength vortex identification criterion λ_{Ci} (as a sum of all the pixels in a 15 pixels-wide window). The combination of the 2 inputs was found to give better results compared to the velocity sensors alone or the swirling strength alone or the local vorticity and velocity combination. The main MSE results can be found in table.

The swirling strength window allows a good detection of every vortices passing into the boundary layer. The spectra

n_1	MSE	Sensor choice
75	0.05	$[\lambda_{Ci}, v]$
110	0.11	v
90	0.32	λ_{Ci}

TABLE II. Comparison of the validation data-set fit error obtained with a single velocity input, a single swirling strength input and their combination. Using the combination of the inputs minimises the MSE and the number of neurons.

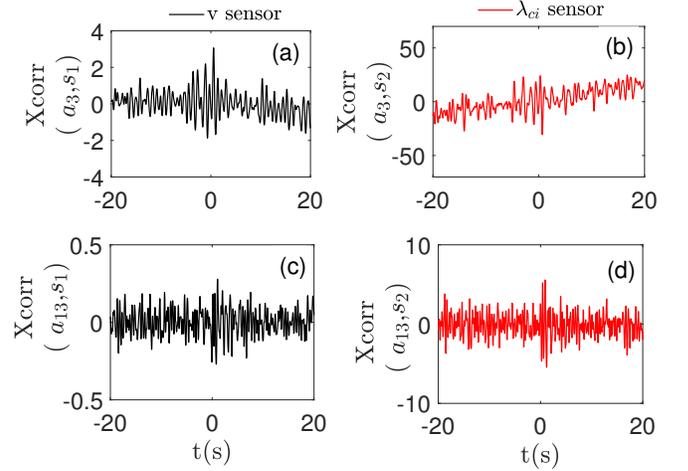


FIG. 11. Cross-correlation of the velocity (s_1) and swirling strength (s_2) sensors with 3rd (a,b) and 13th (c,d) POD coefficient.

of the two sensors (Fig. 10) show that the velocity sensor detects the large structures passing in the neighbourhood with a frequency close to 1 Hz, while the swirling strength criterion gives us more information about the high frequency dynamics up to 4.8 Hz. Some frequencies can be found in both spectra, like a clear 2.32 Hz peak.

The efficiency of the combination of the two sensors can be explained through the cross-correlation of the sensor signals with the time-series signals of the POD coefficients, as illustrated in Fig. 11. One can clearly see that the 3_{rd} POD coefficient is weakly correlated with the swirling strength sensor, while it is strongly correlated to the velocity sensor. On the opposite the 13_{th} POD coefficient, which contains higher frequencies (see Fig. 9), is strongly correlated with the swirling strength sensor. This could be one of the reasons why the training using these two sensors is much more accurate than using a single sensor.

C. Position and size of the visual sensor(s)

The width of the swirling strength window also plays an important role to obtain good SI results. If it is too large it creates an unnecessary smooth event, while if too small it can be too noisy, especially for gradient variables computed from experimental data. A good compromise was found for $0.7 y^*$. The height of the was chosen large ($5 y^*$), so that it contains entirely the advected structures in the wall-normal direction,

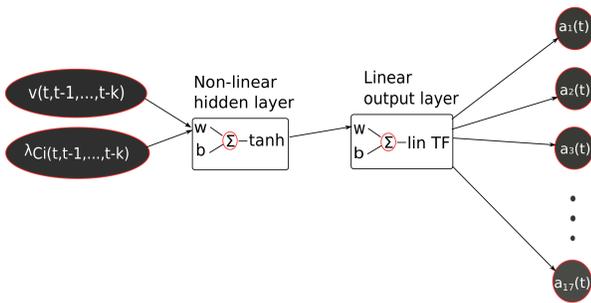


FIG. 12. Structure of the FTD ANN used for this study, with 2 inputs-sensors (v and λ_{Ci}), a k time-delay, 1 hidden layer with 75 neurons and M outputs (in our case $M = 17$ POD modes).

as observed in the POD modes.

The choice of the position of the sensors is also crucial. In general, the sensors should be located as upstream as possible in order to measure and predict early the downstream dynamics. The final position of the chosen sensors is shown in Fig. 6.

VII. PARAMETER SENSITIVITY STUDIES

The architecture and parameters of the ANN can be changed to avoid overfitting and improve the training process, mainly by choosing the appropriate number of neurons in the hidden layer. The depth of the network (meaning number of hidden layers) can also be increased if needed. Generally a deeper network (using the same number of total neurons) will allow the identification of more complex features and information. The drawback is that the successful training can become very tricky, especially since it increases significantly the computational time for the training as well as the complexity of the network. This makes training parameter sensitivity analysis almost impossible.

In the FTD ANNs used in the present study the input time-delay (meaning the size of the sensor memory) can also be chosen accordingly to improve training and validation fit. It was found critical for the success of the training. The training process had a very low computational time: it was of the order of $O(1)$ minute using a Intel Xeon E5-2630 CPU running at 2.2 GHz. This allowed for a full parametric study to find the optimal time delay k for the inputs and the minimum number of neurons for the hidden layer.

For all the ANN calculations (training, validation) the MATLAB Deep Learning Toolbox was used. A basic scheme of the final FTD ANN can be shown in Fig. 12. An anti-causal zero phase low-pass moving average (over four time-steps) filter has been applied to each pixel time-series. The maximum frequency of the forced BL dynamics was 5 Hz, so seven times lower than the low-pass filter cutoff frequency.

We mention that training with LSTM recurrent architectures was found inefficient in our case, with a strong increase of the computational time. A simple shallow Time-Delay ANN scheme with 75 neurons gave the best results. This is in contradiction with the result of Ogunmolu et al.²⁸ who have

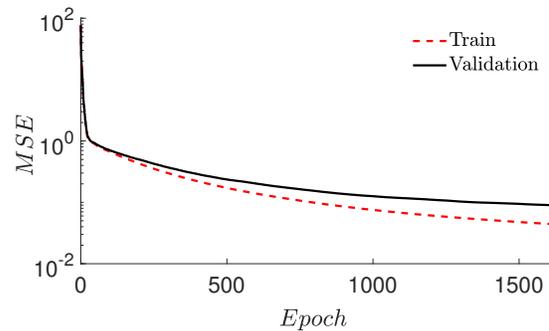


FIG. 13. Training and validation mean-squared error for each epoch of the training process.

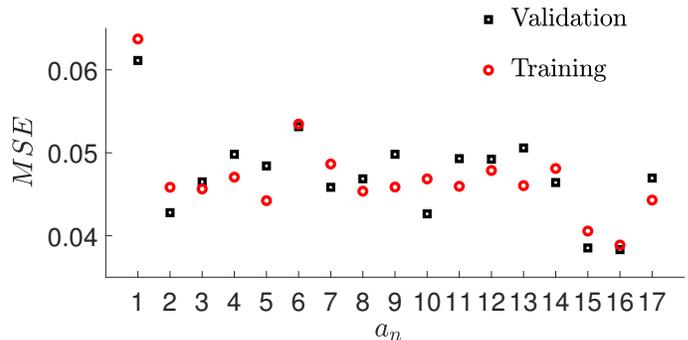


FIG. 14. Training and validation MSE for each POD coefficient time series.

found out that deep architectures are more efficient than single hidden layer models for noisy data sets. In our case a single hidden layer model was sufficient.

The ANN used contained a tan-sigmoid transfer function. As a standard time-delay neural network it also contained an output layer with a linear transfer function to the output. The training time using a scaled conjugate gradient back-propagation algorithm was around 1 minute for a shallow network using a Intel Xeon E5-2630 CPU running at 2.2 GHz. It allowed a full parametric study to find the optimal time-delay for the inputs and the appropriate number of neurons in the hidden layer.

A. Training parameters

The full data-set is divided into training (85% of the snapshots), validation (10% of the snapshots) and over-fitting check data-set (5% of the snapshots) as summarized in Table IV. The third data-set is used as an early stopping criterion to test if a rising error is observed during the training process. As a common practice the data are shuffled randomly before the beginning of the training process. It avoids bad models if the data are initially classified. It also makes fitting faster because neighbouring points are not similar. Block consecutive data-set division failed to give good fit results. We also mention that a data standardization was applied before the training

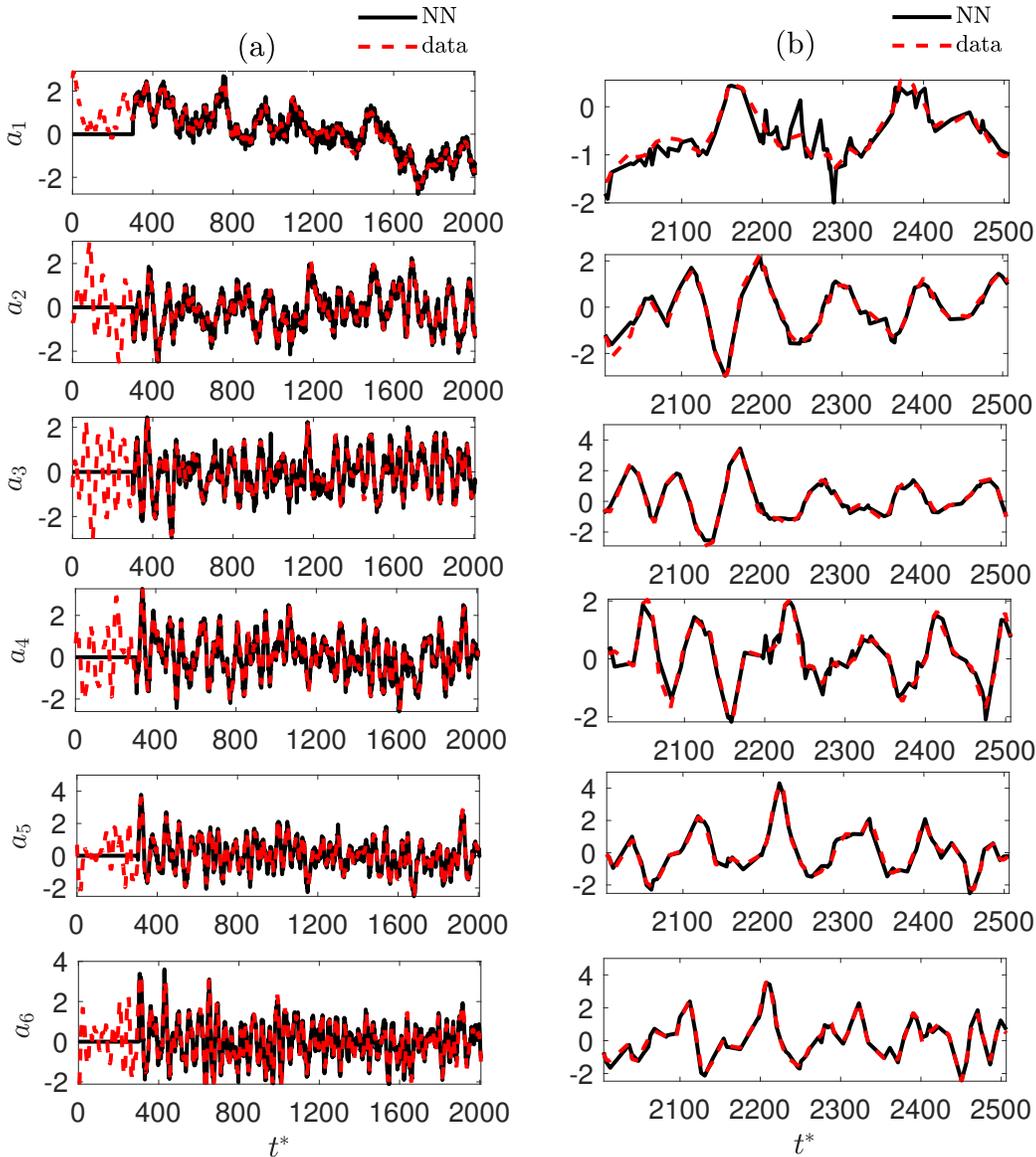


FIG. 15. Comparison of the 1st to 6th POD coefficients time-series obtained from experimental data with the ones obtained with the ANN for training(a) and validation (b) datasets. The network zero signal due to time-delay is evident in the training set.

# Inputs	2
# Outputs	17
# Training Snapshots	5440 (85%)
# Validation Snapshots	640 (10%)
# Overfitting check snapshots	320 (5%)

TABLE III. Neural network training parameters configuration

process which also helps when different inputs or outputs have different scales. Lastly, we mention that the weights and biases of the network are initialized randomly for the first epoch.

The optimum number of neurons for the hidden layer was found to be 75 and the best sensor time-delay corresponded to $k = 300.8 \times t^*$ snapshots (or around 4 s).

The evolution of MSE with the epoch number is shown in Fig. 13 for both training and validation. The MSE decreases sharply for the first epochs and more smoothly for increasing epochs. The optimum architecture and training parameters are summarized in the table IV.

Network layer structure	2-75-17-17
Activation function	Hyperbolic Tangent
Loss function	MSE
Training method	Scaled Conjugate Gradient
Time-delay(s)	4

TABLE IV. Final choice for the neural network parameters.

In Fig. 14 we can see that the MSE for each POD coeffi-

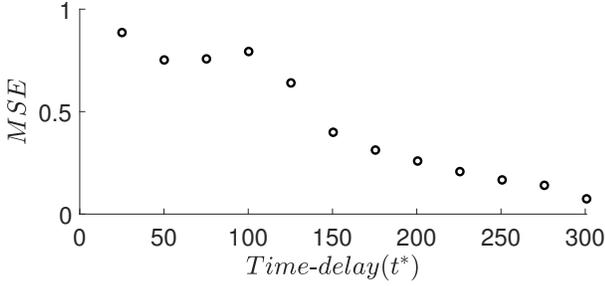


FIG. 16. Network validation performance as a function of the time-delay in the sensors. Increasing the size of the sensor memory decreases the error in a linear way, after approximately 150 time units.

cient is around 5 % for most of the coefficients and for both training and validation, which is very satisfactory. In Fig. 15 we compare the time-series of the six first POD modes with the time-series obtained in the training step (a) and in the validation step (b). Both training and validation data are in very good agreement with the experimental data.

B. Influence of the sensor memory

The influence of the time-delay in the inputs, is illustrated on the plot of the evolution of the MSE as a function of the time-delay shown in Fig. 16. We observe that the sensor memory is critical for the correct dynamics identification: a minimum of $k = 150 \times t^*$ time units is required to obtain for the first time a correct identification with an MSE lower than 50%. A sensor memory of $300.8 \times t^*$ breaks the error limit of 5%. The interpretation of the physical meaning of an optimal neural network is often useful, but not frequently studied. In our case we observe that an abrupt error drop occurs in the region of the sensor memory size of $t^* = 150 \times t^*$; using the structure advection velocity we obtain a characteristic length scale of $112.2 x^*$, which is 1.22 times the size of our PIV domain ($92 x^*$). This shows that the optimal memory size is actually a function of the sensor position, the structure advection velocity and the size of the domain of which we are identifying the dynamics.

C. Influence of the number of neurons in the hidden layer

Keeping an optimal constant time delay of 600 time-steps or $300.8 t^*$, we can study the sensitivity of the validation errors as a function of the number of neurons in the hidden layer, as shown in Fig. 17. A minimum of 70 neurons is needed to achieve an error of the order of 10 %. We also should keep in mind that increasing the number of neurons in the hidden layer increases drastically the number of parameters and the complexity of this fully connected architecture, even if the network is shallow.

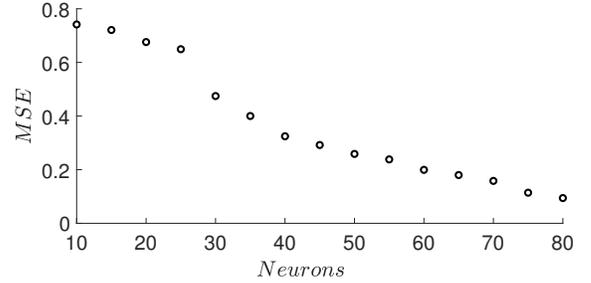


FIG. 17. Network validation performance as a function of the number of neurons in the hidden layer.

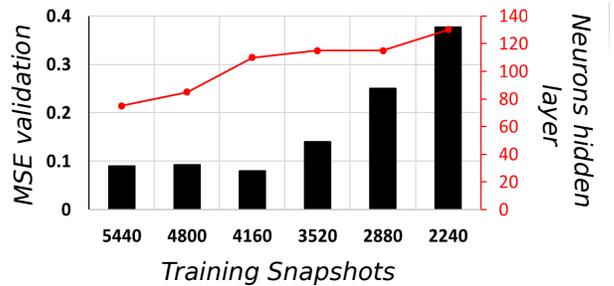


FIG. 18. Network validation performance as a function of the number of snapshots used for the training.

D. Influence of the size of the training data-set

In Fig. 18 we show that decreasing the number of snapshots used for the training of the network leads to an increase of the number of neurons in the hidden layer needed for an equivalent validation error (keeping a constant time-delay of 301 time units t^*). Above a given number of snapshots, increasing the number of neurons does not help: it introduces overfitting hence increasing the validation error.

VIII. OPTIMAL PARAMETER RESULTS

Using the optimal time delay, the right number of neurons in the hidden layer and the proper combination of double upstream sensors, we can check the efficiency of the network in the POD coefficient signal prediction and the time-resolved instantaneous fields reconstruction. The dynamic coefficient signals are accurately predicted. In Fig. 15 it is clear that the fit error is similar for both training (left) and validation (right) datasets, which ensures we avoided overfitting. In Fig. 19 we see that the main large coherent structures are well reconstructed.

Finally, the kinetic energy of these perturbations (PKE) computed for each pixel in the PIV window is defined as:

$$PKE(t) = \frac{1}{2} \left\| \left(u'(t), v'(t) \right) \right\|^2 \quad (12)$$

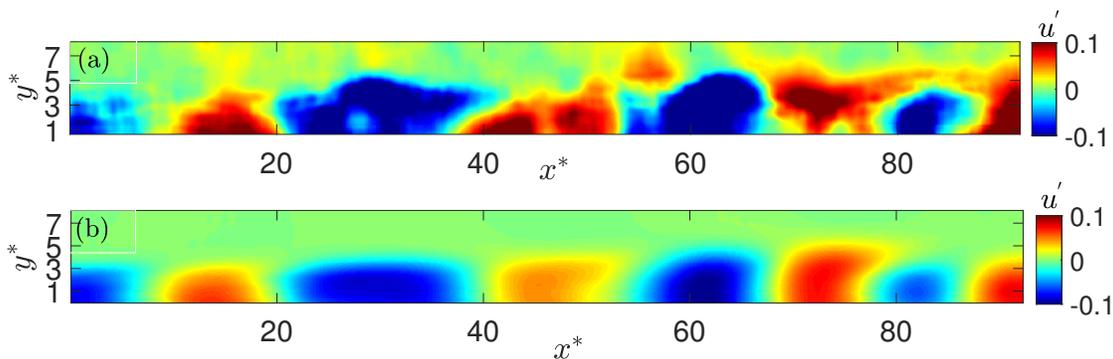


FIG. 19. Instantaneous horizontal fluctuation velocity field comparison between experiment (a) and ANN-reconstructed field (b). Movie online. Time-steps are not consecutive due to the dataset shuffling process during the training-validation division.

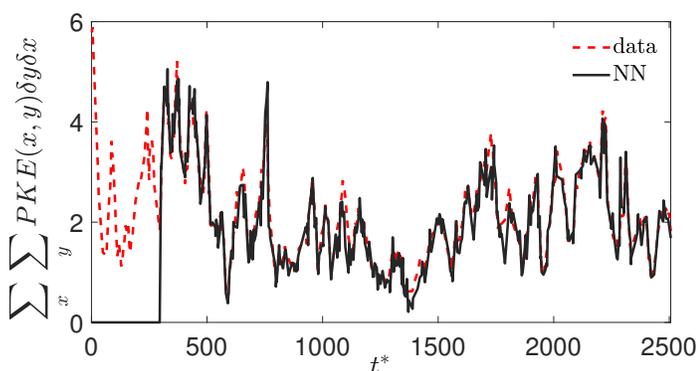


FIG. 20. Comparison of the sum of PKE computed in the PIV window between decomposed experimental field and ANN-identified reconstructed field. Zero network signal is due to sensor time-delay.

The sum of this quantity for each pixel is an interesting control quantity for the future model-free control scheme, since it is an indication of the total advected turbulent fluctuations intensity. So it is calculated and compared, between ANN generated results and experiment. For the validation data-set, the optimal ANN architecture gives us a MSE of 10.46 %. The model and experimental time-series are in very good agreement as shown in Fig. 20.

IX. CONCLUSION

A successful application of an ANN SI framework was presented for the case of an experimental Turbulent Boundary Layer flow. FTD ANNs were proven flexible enough to achieve sufficient prediction precision, even for a high-order multiple-input multiple-output non-linear system. A non-linear mapping between upstream sensors and the global dynamics of the flow (in the form of POD coefficients) was presented. The cross correlation between sensors and coefficient signals was proven a useful tool for the choice of the optimal sensors. Different sensors correlate well with different

coefficient signals. It was revealed why the collaboration of a swirling strength sensor and a local velocity sensor resulted in an accurate network training. The dynamics were identified, the turbulent coherent structures (wavy streaks) were reconstructed, and the system order was reduced to two easily measurable signals. A physical interpretation was given for the optimal sensor memory size in the final ANN architecture.

The difficulty in the estimation of the coherent structures in this TBL flow arises due to streak wavyness and weak three-dimensionality. Still the ANN was proven efficient if the sensor memory size and number of neurons are properly chosen. The achieved fit accuracy is around 5%. The present study is the first step in identifying an efficient model-free control law like in Gautier et al.¹⁴, Li et al.²¹, that comes from deep knowledge and precise prediction of the dynamics of each point in the velocity field. It will target the minimization of the PKE, since this quantity can be tracked in real-time. We also mention that the Reynolds number could also be integrated as an input, while the output could include POD coefficients for the different Reynolds numbers, hence going towards a system of variable velocity in the style of San et al.²⁹.

The presented methodology could also be applied to the same turbulent flow but based on the horizontal (spanwise-streamwise) PIV plane (Fig. 4). Spanwise-distributed sensors and deeper networks are possibly needed for this complicated task but it clearly is now feasible. Different transition scenarios in the boundary layer flow can also be investigated.

X. ACKNOWLEDGEMENTS

This research was supported by CNRS and Photon Lines SAS.

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

¹Beneitez M, Duguet Y, Schlatter P, Henningson DS (2019) Edge tracking in spatially developing boundary layer flows. *Journal of Fluid Mechanics* 881:164–181, doi:10.1017/jfm.2019.763

- ²Bore J (2003) Extended proper orthogonal decomposition: a tool to analyse correlated events in turbulent flows. *Experiments in Fluids* 35(2):188–192, doi:10.1007/s00348-003-0656-3
- ³Cambonie T, Aider JL (2014) Transition scenario of the round jet in crossflow topology at low velocity ratios. *Physics of Fluids* 26(8):084101, doi:10.1063/1.4891850, URL <https://doi.org/10.1063/1.4891850>
- ⁴Chakraborty P, Balachandar S, Adrian RJ (2005) On the relationships between local vortex identification schemes. *Journal of Fluid Mechanics* 535:189–214, doi:10.1017/S0022112005004726
- ⁵Champagnat F, Plyer A, Le Besnerais G, Leclaire B, Davoust S, Le Saint Y (2011) Fast and accurate PIV computation using highly parallel iterative correlation maximization. *Experiments in Fluids* 50:1169–1182, doi:10.1007/s00348-011-1054-x
- ⁶Charaniya N, Dudul S (2012) Focused time delay neural network model for rainfall prediction using indian ocean dipole index. pp 851–855, doi:10.1109/CICN.2012.116
- ⁷Chong M, Perry A, Cantwell B (1990) A general classification of 3-dimensional flow fields. *Physics of Fluids* 2:765–777
- ⁸CPan, XDong, XYang, WJinjun, WRunjie (2015) Evaluating the accuracy performance of lucas-kanade algorithm in the circumstance of piv application. *Sci China-Phys Mech Astron* 58, doi:10.1007/s11433-015-5719-y
- ⁹Deng Z, Chen Y, Liu Y, Kim KC (2019) Time-resolved turbulent velocity field reconstruction using a long short-term memory (lstm)-based artificial intelligence framework. *Physics of Fluids* 31(7):075108, doi:10.1063/1.5111558, URL <https://doi.org/10.1063/1.5111558>
- ¹⁰Gautier N (2014) Flow control using optical sensors. *Mechanics of the fluids*
- ¹¹Gautier N, Aider JL (2013) Control of the separated flow downstream of a backward-facing step using visual feedback. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 469(2160):20130404, doi:10.1098/rspa.2013.0404
- ¹²Gautier N, Aider JL (2015) Frequency-lock reactive control of a separated flow enabled by visual sensors. *Experiments in Fluids* 56(1):16, doi:10.1007/s00348-014-1869-3, URL <https://doi.org/10.1007/s00348-014-1869-3>
- ¹³Gautier N, Aider JL (2015) Real-time planar flow velocity measurements using an optical flow algorithm implemented on gpu. *Journal of Visualization* 18(2):277–286, doi:10.1007/s12650-014-0222-5
- ¹⁴Gautier N, Aider JL, DUriez T, Noack B, MSegond, MAgel (2015) Closed-loop separation control using machine learning. *JFluidMech* 770:442–457
- ¹⁵Giannopoulos A, Aider JL (2020) Prediction of the dynamics of a backward-facing step flow using focused time-delay neural networks and particle image velocimetry data-sets. *International Journal of Heat and Fluid Flow* 82:108533, doi: <https://doi.org/10.1016/j.ijheatfluidflow.2019.108533>, URL <http://www.sciencedirect.com/science/article/pii/S0142727X19308665>
- ¹⁶Guzmán InJ, Sipp D, Schmid PJ (2014) A dynamic observer to capture and control perturbation energy in noise amplifiers. *Journal of Fluid Mechanics* 758:728–753, doi:10.1017/jfm.2014.553
- ¹⁷Guzmán-Inigo J, Sodar M, Papadakis G (2019) Data-based, reduced-order, dynamic estimator for reconstruction of nonlinear flows exhibiting limit-cycle oscillations. *Physical Review Fluids* 4, doi:10.1103/PhysRevFluids.4.114703
- ¹⁸Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Computation* pp 1735–1780
- ¹⁹Huang X, Zhang W, Xu X, Yin R, Chen D (2019) Deeper time delay neural networks for effective acoustic modelling. *Journal of Physics: Conference Series* 1229:012076, doi:10.1088/1742-6596/1229/1/012076, URL <https://doi.org/10.1088/1742-6596/1229/1/012076>
- ²⁰Kachanov YS (1994) Physical mechanisms of laminar-boundary-layer transition. *Annual Review of Fluid Mechanics* 26(1):411–482, doi:10.1146/annurev.fl.26.010194.002211, URL <https://doi.org/10.1146/annurev.fl.26.010194.002211>
- ²¹Li R, Noack BR, Cordier L, Borée J, Harambat F (2017) Drag reduction of a car model by linear genetic programming control. *Experiments in Fluids* 58(8):103, doi:10.1007/s00348-017-2382-2, URL <https://doi.org/10.1007/s00348-017-2382-2>
- ²²Liu B, Tang J, Huang H, Lu XY (2020) Deep learning methods for super-resolution reconstruction of turbulent flows. *Physics of Fluids* 32(2):025105, doi:10.1063/1.5140772, URL <https://doi.org/10.1063/1.5140772>, <https://doi.org/10.1063/1.5140772>
- ²³Lumley JL (1967) The structure of inhomogeneous turbulent flows. In: Yaglom AM, Tatarski VI (eds) *Atmospheric turbulence and radio propagation*, Nauka, Moscow, pp 166–178
- ²⁴Matsubara M, Alfredsson PH (2001) Disturbance growth in boundary layers subjected to free-stream turbulence. *Journal of Fluid Mechanics* 430:149–168, doi:10.1017/S0022112000002810
- ²⁵Mendez M, Raiola M, Masullo A, Discetti S, Ianiro A, Theunissen R, Buchlin JM (2017) Pod-based background removal for particle image velocimetry. *Experimental Thermal and Fluid Science* 80:181–192
- ²⁶Morkovin MV (1985) Bypass transition to turbulence and research desiderata. *Transition in Turbines* pp 161–204
- ²⁷Nathan Kutz J (2017) Deep learning in fluid dynamics. *Journal of Fluid Mechanics* 814:1–4, doi:10.1017/jfm.2016.803
- ²⁸Ogunmolu O, Gu X, Jiang S, Gans N (2016) Nonlinear systems identification using deep dynamic neural networks. CS
- ²⁹San O, Maulik R, Ahmed M (2018) An artificial neural network framework for reduced order modeling of transient flows
- ³⁰Sirovich L (1987) Turbulence and the dynamics of coherent structures. *Quarterly of Applied Mathematics* 45:561–571
- ³¹Varon E, Eulalie Y, Edwige S, Gilotte P, Aider JL (2017) Chaotic dynamics of large-scale structures in a turbulent wake. *Phys Rev Fluids* 2:034604, doi:10.1103/PhysRevFluids.2.034604, URL <https://link.aps.org/doi/10.1103/PhysRevFluids.2.034604>
- ³²Varon E, Aider JL, Eulalie Y, Edwige S, Gilotte P (2019) Adaptive control of the dynamics of a fully turbulent bimodal wake using real-time piv. *Experiments in Fluids* 60(8):124
- ³³Vlachas P, Byeon W, Yi Wan Z, P Sapsis T, Koumoutsakos P (2018) Data-driven forecasting of high-dimensional chaotic systems with long-short term memory networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science* 474, doi:10.1098/rspa.2017.0844
- ³⁴Waibel A, Hanazawa T, Hinton G, Shikano K, Lang KJ (1989) Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37(3):328–339, doi:10.1109/29.21701
- ³⁵Wang Z, Xiao D, Fang F, Govindan R, Pain C, Guo Y (2018) Model identification of reduced order fluid dynamics systems using deep learning. *Int J Numer Meth Fluids* 86:255–268, doi:10.1002/flid.4416
- ³⁶Zhou J, Adrian R, Balachandar S, Kendall T (1999) Mechanisms for generating coherent packets of hairpin vortices. *J Fluid Mech* 387:535–396