Application of a neural network for predicting the control effect on sweep events in a fully turbulent channel flow

Original
Application of a neural network for predicting the control effect on sweep events in a fully turbulent channel flow / Saccaggi, E.; Di Cicca, G. M.. - ELETTRONICO. - (2025), pp. 1-11. (AIAA AVIATION FORUM AND ASCEND, 2025 Las Vegas, Nevada, USA 21-25 July 2025) [10.2514/6.2025-3022].

Publisher:
AIAA American Institute of Aeronautics and Astronautics

Published
DOI:10.2514/6.2025-3022

Application of a neural network for predicting the control effect on sweep events in a fully turbulent channel flow

Enrico Saccaggi * and Gaetano Maria Di Cicca†
Politecnico di Torino, Turin, Italy, 10129

The proposed study explores the application of a neural network to predict the control effect on sweep events within the framework of an opposition control strategy. Experiments were conducted in a fully turbulent channel flow at a friction Reynolds number equal to 365. A dataset was created by employing different control parameters sampled with a Latin hypercube sampling method. A neural network was successfully trained to predict the controlled conditionally averaged sweep event given the actuation parameters and the uncontrolled conditionally averaged sweep event. Results show that the model is able to reliably predict the control signature across different actuation parameters. The predicted model was applied to perform an offline optimization of the actuation parameters of the control strategy to find the best actuation parameters that can reduce the sweep event intensity. Moreover, a sensitivity analysis by using the Jacobian matrix was also carried out to understand which parameters influence the predicted response the most.

I. Introduction

Drag reduction significance has become essential to reduce both energy consumption and pollutant emissions in modern transportation systems. Among the various components of drag, skin friction plays a dominant role, accounting for up to 55% of the total drag budget in a commercial aircraft [1]. Active control techniques have shown significant potential for reducing skin friction and have received substantial attention in the literature [2–5]. Choi et al. [6], in their direct numerical simulations demonstrated that by employing blowing and suction at the wall with velocities equal in magnitude but opposite in direction to those at a specified wall-normal location (referred to as the "detection plane"), skin friction reductions up to 25% can be achieved. This approach, known as opposition control, effectively creates a "virtual wall", an imaginary plane with no through-flow, situated midway between the detection point and the wall. The virtual wall mitigates drag by blocking the downwash of high-speed fluid during sweep events, preventing it from reaching the wall and thereby reducing skin friction. Wind tunnel experiments carried out by Rebbeck and Choi [7, 8] demonstrated that real-time opposition control, using a wall-normal jet to block high-speed fluid during sweep events, can effectively reduce the sweep events intensity.

In recent years, data-driven models, such as machine learning algorithms, have been widely applied to fluid dynamics. They can augment domain knowledge and automate tasks related to optimization and flow control problems [9]. For example, they can be used to approximate the system response given some actuation parameters and thus facilitate the design and optimization of the control system. A neural network was applied in [10] to predict the evolution of skin friction in a turbulent channel flow after applying an actuation near the wall comparing the performances of a Convolution Neural Network (CNN) and a Feed Forward Neural Network (FNN). Moreover, Lee et al. [11] approximated with a neural network the correlation between wall-shear stresses and the wall actuations. This strategy allowed a prediction of the optimal wall actuations (blowing or suction) in a turbulent channel flow to achieve skin friction drag reduction.

In the context of opposition control, Park and Choi [12] trained a convolutional neural network (CNN) to predict the wall-normal velocity at $y^+ = 10$ using wall-based quantities such as pressure and wall shear stress. This approach aimed to replicate the control effect originally demonstrated by Choi et al. [6], while improving the practical applicability by reducing the intrusiveness associated with traditional sensor measurements. A similar work has also been performed in [13, 14].

The objective of the present study is to predict through a FNN the effect of the actuation parameters on the Conditionally Averaged Sweep Event (CASE) by experimentally implementing an opposition control strategy in a fully turbulent channel flow. Creating a model that can closely replicate the effect of control on sweep events has

---

[*] PhD candidate, Department of Mechanical and Aerospace Engineering, Corso Duca degli Abruzzi, 24, 10129, Turin, Italy

[†] Associate professor, Department of Mechanical and Aerospace Engineering, Corso Duca degli Abruzzi, 24, 10129, Turin, Italy

several advantages. One is the ability to easily perform a sensitivity analysis to understand each actuation parameter's contribution to the system response. In addition, if the model is sufficiently robust and reliable, it would be possible to perform an offline optimization of the control parameters that would be much faster than online optimization methods, since the function to be evaluated turns out to be much faster. An example of an online optimization applied to an opposition control strategy through a genetic algorithm was conducted by [15].

II. Experimental setup

In this study, experiments were conducted in an 8-meter-long duct with a 7 cm x 30 cm rectangular cross-section (aspect ratio: 4.28). The center-line velocity was set to $U_0 = 3.22\,\mathrm{m\,s^{-1}}$ with a friction velocity of $u_\tau = 0.16\,\mathrm{m\,s^{-1}}$, yielding a friction Reynolds number, $Re_\tau = 365$. Fig. 1 shows the experimental setup schematically. Two hot-wire probes were employed, each of them connected to a Dantec 55M10 CTA Standard Bridge anemometer. Both probes feature a sensitive tungsten wire measuring 5 µm in diameter and 1.25 mm in length. The upstream probe, a Dantec 55P11, was positioned at $x^+ = -65$ and a wall-normal distance of $y^+ = 15$. The superscript + denotes quantities in wall units. The downstream probe, a Dantec 55P15 boundary layer probe, was mounted on a 2-axis stand for precise adjustments in the streamwise and wall-normal directions, with positional accuracies of 25 µm and 40 µm respectively.

Signals were sampled using a National Instrument PCI-MIO-16-XE-10 16-bit acquisition board installed in a workstation. Data post-processing was performed on a laptop connected to the workstation via LAN, with data exchanged through a Python-based client-server architecture.

Following [8], sweep events were detected in real-time using the velocity gradient technique on the longitudinal velocity component measured by the front probe. An analog differentiator computes the time derivative from the hot-wire anemometer's voltage output, while a comparator identifies events when the time derivative exceeds a given threshold. The gradient technique threshold level was calibrated to yield a similar event count to the Variable-Interval Time-Averaging (VITA) method described in [16] having a VITA threshold level $k = 1.0$ and an averaging time $T^+ = 10$. A detected sweep event generates a TTL signal sent to an Arduino Uno Rev3 microcontroller, introducing a delay ΔT to accommodate convection time and actuator response. The microcontroller implements the imposed delay time through a circular buffer of sufficient size to prevent event loss. The Arduino board is connected to the workstation with a USB port to change the delay time automatically. The delayed TTL signal triggers a signal generator (Agilent 33120A), to produce a single-period sine wave function of frequency f and amplitude $\hat{A}$, which is then amplified by a Kenwood KAC-5205 amplifier to drive a loudspeaker. The loudspeaker emits a jet through a 1 mm diameter orifice, corresponding to 11 viscous units.

The rear probe velocity signal is acquired and conditionally averaged, enabling visualization of the sweep event. As an example, in Fig. 2 is shown the uncontrolled and the controlled CASE for the control parameters $f = 35.67\,\mathrm{Hz}$, $\hat{A} = 54\,\mathrm{mV_{pp}}$ and $\Delta T = 1.43\,\mathrm{ms}$.

III. Dataset and model description

In order to train the model, we created a dataset containing a significant number of case studies. The dataset was created by modifying the control parameters since the goal is to predict the velocity characterizing the controlled CASE, denoted as $\hat{u}_c$, using the actuation parameters and the velocity characterizing the uncontrolled CASE, $\hat{u}_{\mathrm{nc}}$, as inputs. Let a control triplet, consisting of actuation frequency, amplitude, and delay time, be defined as ψ. A total of 130 control triplets were sampled to construct the dataset by applying a Latin Hypercube Sampling (LHS) technique [17] in the bounds indicated in Table 1. LHS is a technique used to generate a uniform distribution of points in a multidimensional space. For all the dataset points, $\hat{u}_c(\psi)$ was evaluated experimentally after a conditional sampling of the velocity signal acquired at $x^+ = 65$ and $y^+ = 15$. The dataset generated was split into training and validation, 80% and 20% respectively. The training set is used for understanding relationships and patterns within the data. It allows the model to adjust its parameters to minimize error and improve predictions. On the other hand, the validation set is often used to evaluate the performance of the model during training. Also, it allows tuning of hyperparameters (learning rate, regularization, number of layers, and number of neurons) and helps to avoid overfitting. It is important to highlight that the validation set is not employed to train the model. The dataset was normalized to have all the features in the range between 0 and 1 with the following normalization:

$$\overline{\boldsymbol{x}} = \frac{\mathbf{x} - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})} \tag{1}$$

where $\overline{\boldsymbol{x}}$ is the scaled input vector and ranges between 0 and 1 and $\mathbf{x}$ is the unscaled input vector.

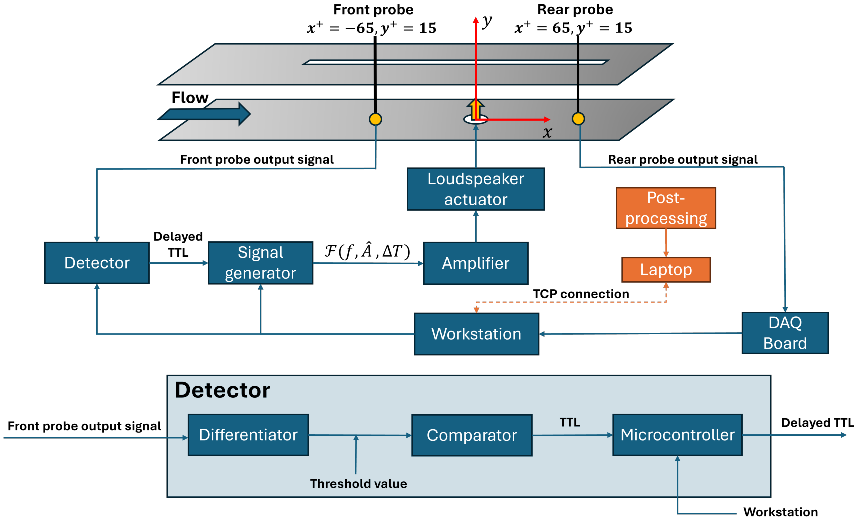

Fig. 1 Control system schematic: the upstream probe signal serves as a trigger for the actuation, while the downstream probe signal undergoes directly to post-processing.

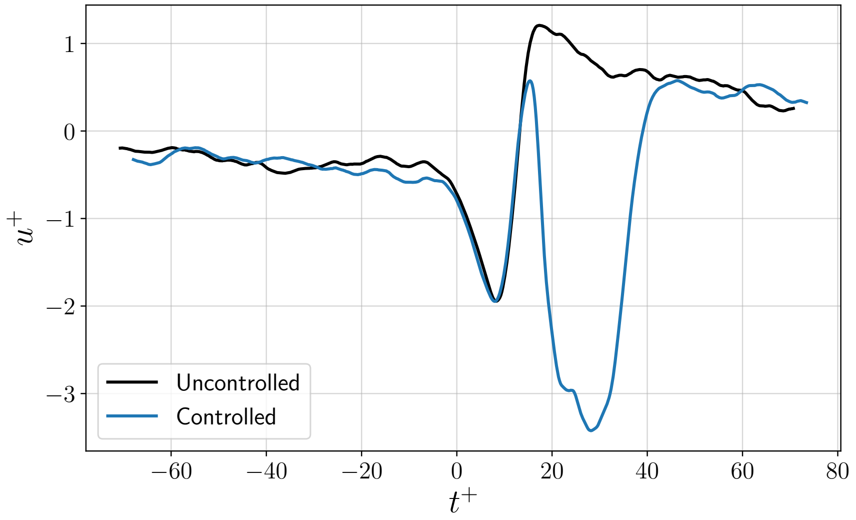

Fig. 2 Uncontrolled and controlled CASEs measured by the rear probe at $x^+ = 65$ and $y^+ = 15$. The control parameters are $f = 35.67\,\mathrm{Hz}$, $\hat{A} = 54.0\,\mathrm{mV_{pp}}$ and $\Delta T = 1.43\,\mathrm{ms}$.

Table 1 LHS sampling bounds.

	Lower bound	Upper bound
Frequency [Hz]	30	60
Amplitude [$\mathrm{mV_{pp}}$]	50	75
Delay time [ms]	0.15	3

This study employs a FNN to predict the control effect on the CASEs. This model was chosen for its ability to represent non-linear relationships that may not be fully captured by traditional regression approaches. As illustrated in Fig. 3, the FNN architecture comprises an input layer with three inputs corresponding to the control triplet ψ, followed by two hidden layers with the number of neurons equal to H_{hid} and $2H_{\text{hid}}$, respectively. H_{hid} is a parameter that needs to be tuned and is a measure of the network size. The network's output layer aligns with the size of the conditional averaged signal being predicted. The application of a ReLU activation function to both hidden layers enables the network to effectively capture and model non-linear relationships within the data. The network output, denoted $\Delta\epsilon$, is then summed with $\hat{u}_{\mathrm{nc}}$ to give the predicted controlled CASE, represented as $\hat{u}_c^*$ for a given triplet ψ. The FNN was trained using the mean squared loss function, also called L_2 loss, which minimizes the average squared difference between $\hat{u}_c$ and $\hat{u}_c^*$. The training was done over 10000 epochs using Adam optimizer [18] with a learning rate of 1.2×10^{-4}. To enhance model generalization and mitigate overfitting, a weight decay of 8.6×10^{-6} was applied, effectively penalizing large weights. The hyperparameters given above were tuned to find a proper trade-off between performance and overfitting avoidance. Fig. 4(a, b) illustrates the evolution of the training and validation losses, respectively, as a function of the hidden dimension H_{dim}, while all the other hyperparameters are kept constant. A general decreasing trend in both training and validation losses is observed as the network size increases. However, the validation loss for $H_{\mathrm{dim}} = 256$ exceeds that of $H_{\mathrm{dim}} = 128$, indicating a mild overfitting effect. As expected, the loss curves exhibit steeper descents for larger network sizes. This happens because bigger models have more parameters to be tuned, which, at a fixed learning rate, enables larger models to converge more rapidly compared to their smaller counterparts. The final choice was $H_{\mathrm{dim}} = 128$ to have higher accuracy and to avoid overfitting problems.

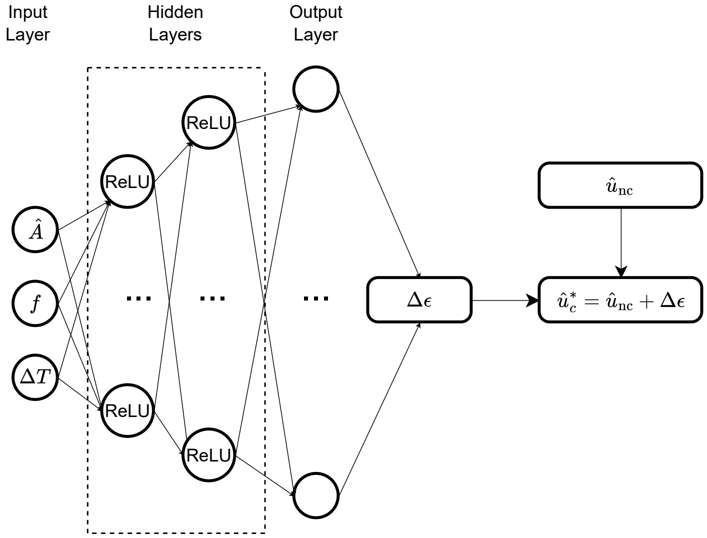

Fig. 3 FNN structure used to predict the controlled CASE $\hat{u}_c^*$ based on the control parameters.

IV. Results

A. Model performances

The performance of the network was evaluated using the validation dataset, and is depicted in Fig. 5. In this figure, the measured value to be approximated by the FNN, which is $\hat{u}_c$, is depicted with a solid blue line while the solid red line indicates the predicted velocity in the controlled CASE, denoted as $\hat{u}_c^*$. A comparison between these two curves reveals that the dynamics of the system is captured with a good agreement by the FNN, providing a faithful approximation of the controlled sweep event.

Network accuracy across all validation samples can be highlighted in Fig. 6 in which is presented a comparison between the velocity characterizing the sampled sweep event, $\overline{\hat{u}_c}$, and the FNN predicted one, $\overline{\hat{u}_c^*}$. The overline $\overline{(\cdot)}$ indicates a normalization with the minimum and maximum values of the dataset (Eq. 1). These results clearly illustrate the model's ability to predict the controlled event accurately, based on the input control triplet and the uncontrolled CASE.

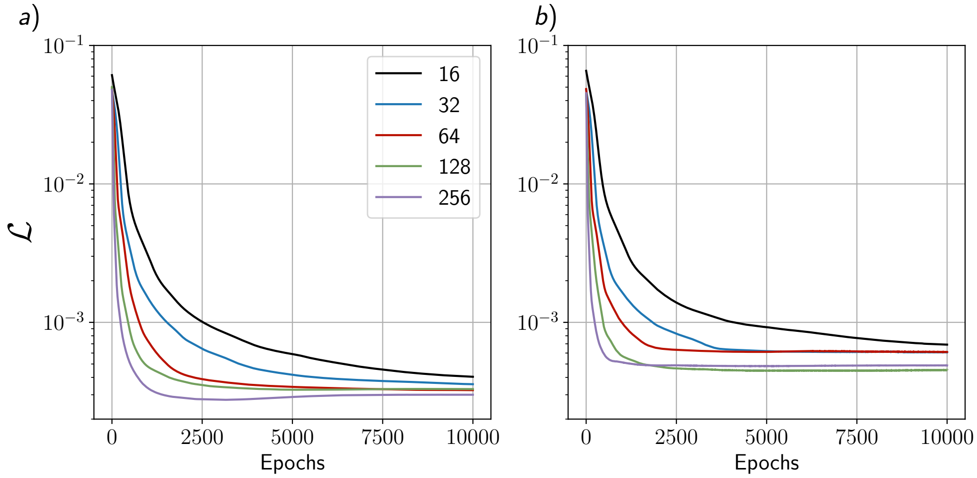

Fig. 4 Training and validation losses of the FNN training process vs epochs for different values of the parameter $H_{\mathbf{hid}}$. a) losses computed by evaluating the network in the training dataset. b) losses computed by evaluating the network in the validation dataset.

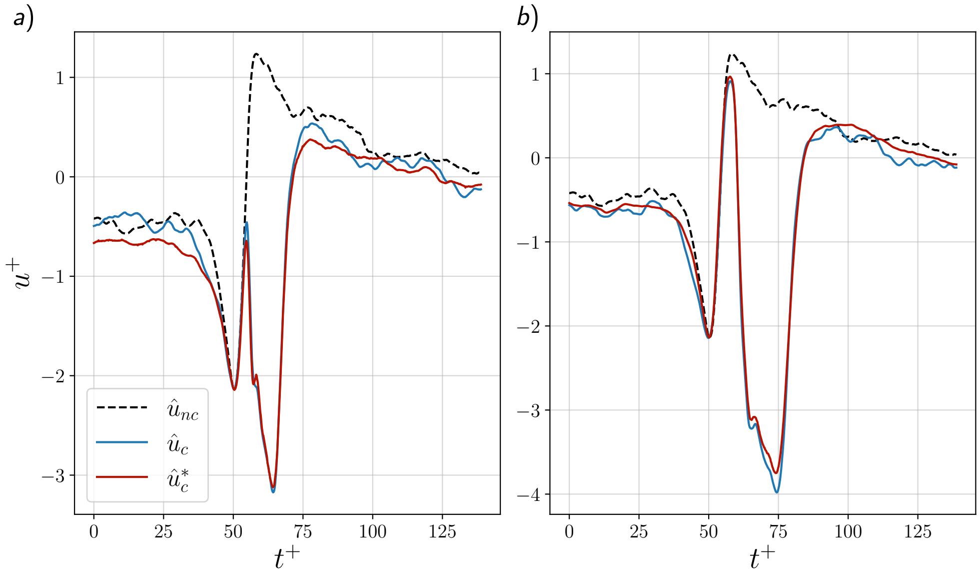

Fig. 5 Evaluation of the FNN with the validation dataset. The black dashed line represents the uncontrolled CASE. The blue and red solid lines correspond to the experimentally sampled controlled CASE and the FNN-predicted event respectively. The control is performed in a) with $\psi = [53.10\,\mathrm{Hz}, 62.4\,\mathrm{mV_{pp}}, 1.01\,\mathrm{ms}]$ and in b) with $\psi = [31.54\,\mathrm{Hz}, 60.0\,\mathrm{mV_{pp}}, 2.71\,\mathrm{ms}]$.

B. Sweep event intensity reduction

The availability of a function that is both faithful and quick to evaluate can be helpful in identifying, within the training bounds, the point at which control is most effective. The effectiveness of the control in reducing the intensity of sweep events can be assessed by comparing the area associated with the uncontrolled case, A_{nc}, to that of the controlled case, A_{c}. For a given control triplet ψ_i the corresponding cost function is expressed as follows:

$$\mathcal{J}(\psi_i) = -\frac{A_{\mathrm{nc}} - A_{\mathrm{c}}(\psi_i)}{A_{\mathrm{nc}}}, \tag{2}$$

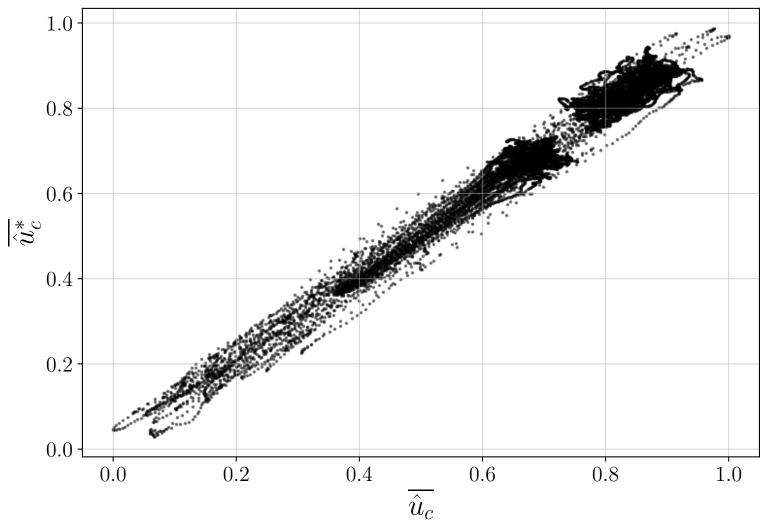

Fig. 6 Comparison between the normalized velocity for the experimentally sampled CASEs, $\overline{\hat{u}_c}$, and the normalized velocity for the FFN predicted CASEs, $\overline{\hat{u}_c^*}$.

where the generic area in either, uncontrolled or controlled cases, is:

$$A_{(\cdot)} = \int_{t_m}^{t_w} u^+ dt, \tag{3}$$

u^+ is the fluctuating component of the streamwise velocity in wall units, t_m is the time in correspondence of the first minimum peak observable in Fig. 2 and t_w is the window ending time that was selected to ensure the robustness of the cost function. The negative sign in Eq. 2 was introduced since the difference in areas between the uncontrolled and controlled CASE should be as large as possible, but the cost function needs to be minimized. The cost function was tailored to the definition of sweep intensity as proposed by [7] and formulated in a similar optimization problem by [15]. A Bayesian optimization algorithm, implemented with scikit-optimize [19], was applied to find the cost function minimum inside the sampling bounds of Tab. 1. A Màtern kernel, which defines the similarity between two different control triplets, was chosen with a smoothing parameter $\nu = 2.5$ [20]. The optimization algorithm converged to the optimized triplet:

$$\psi_{\text{opt}} = \left[30\,\text{Hz}, 0.062\,\text{V}_{\text{pp}}, 0.83\,\text{ms}\right].$$

A Gaussian process, with the same kernel as the one used in the optimization process, was applied to interpolate the cost function value. Its value is represented as a function of the frequency and delay time at a fixed amplitude of $\hat{A} = 0.062\,\text{V}_{\text{pp}}$ in Fig. 7(a) and as a function of amplitude and delay time at a fixed frequency of $f = 30\,\text{Hz}$ in Fig. 7(b). The cost function minimum is marked with a black dot in both Fig. 7(a) and Fig. 7(b). This value can lead to a reward value up to 97%. The analysis shows that reducing the frequency across the input space significantly decreases the cost function, thereby increasing the reward. In particular, the values of $\mathcal{J}$ decrease from around -30% at higher frequencies to around -90% in the range between 30 and 35 Hz, highlighting the important role of frequency tuning in improving the control. For frequencies around 30 Hz, further refining of the delay time and the voltage amplitude leads to an additional decrease in the cost function value. This reflects the fact that, while frequency is the main driving parameter, the optimization of the delay time and the voltage amplitude plays a crucial role in increasing the reward.

C. Sensitivity analysis

We analyze the model's sensitivity by computing the Jacobian matrix for a specific control triplet ψ_{opt}. The Jacobian matrix includes the partial derivatives of the output variables, illustrating how changes in each input parameter affect the model's predictions throughout the output space near ψ_{opt}. In our analysis, we calculate the gradient, applying the chain rule, for each output dimension with respect to each input parameter, resulting in a matrix that reflects the sensitivity of

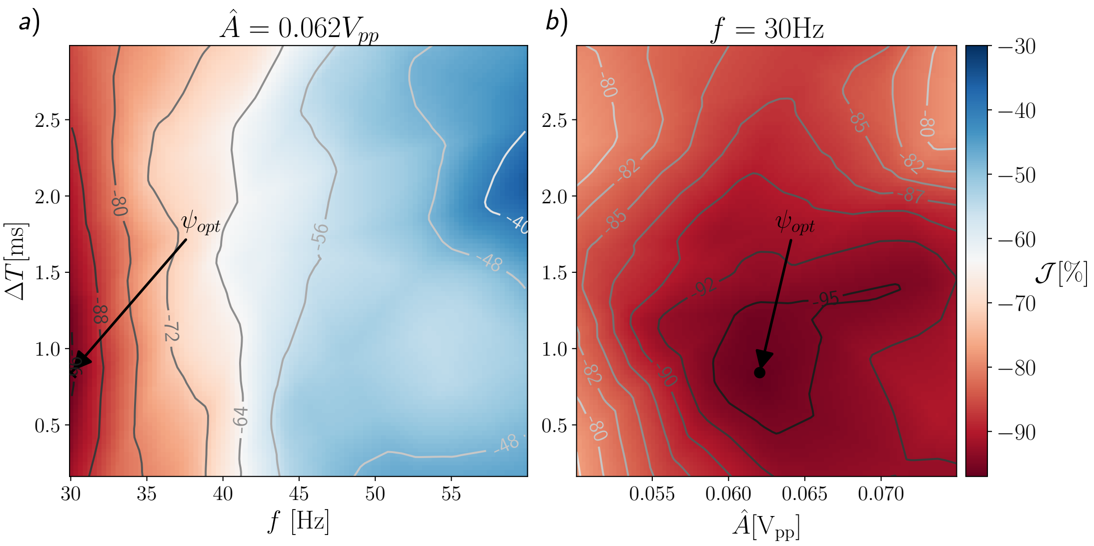

Fig. 7 Cost function value interpolated with a Gaussian Process is presented as a function of: a) frequency and delay time by keeping constant $\hat{A}$ to $0.062\,\mathrm{V_{pp}}$; b) voltage amplitude and delay time by keeping constant f to $30\,\mathrm{Hz}$.

all output points to variations in input parameters. The result of this analysis can be observed in Fig. 8. The uncontrolled CASE and the controlled CASE with the optimal parameters, ψ_{opt} found in subsection IV.B are represented in Fig. 8(a) with dashed and continuous line respectively. Fig. 8(b) depicts the gradient of the output values with respect to the input parameters (f, $\hat{A}$, ΔT). In general, the gradients assume appreciable values in the region affected by the control (from $t^+ = 50$ to $t^+ = 90$), while remaining close to zero elsewhere. Fig. 7 clearly shows that frequency is the primary parameter influencing the cost function value. A similar trend is observed in Fig. 8(b), where the highest gradient, considered in absolute terms, corresponds to the actuation frequency. At $t^+ = 75$, the maximum positive sensitivity with respect to frequency is observed. This can be explained by the fact that, at this time, an increase in frequency would result in a shorter actuation period, leading to an increase in velocity and, consequently, a reduction in the control effectiveness. At $t^+ = 55$, on the other hand, a negative contribution is observed. In this case, reducing the frequency near the onset of actuation leads to a delayed actuation, which allows the velocity in the controlled case to remain higher for a longer period. This behavior is attributed to the fact that, with a lower frequency, the speaker membrane rises more slowly, thus causing the velocity to decrease at a later time. The sensitivity with respect to amplitude, however, shows two slightly negative regions at $t^+ = 55$ and $t^+ = 75$. This can be explained by the fact that an increase in amplitude leads to a reduction in the velocity of the controlled CASE, thereby positively impacting the reduction of sweep event intensity. Finally, Fig. 8(b) illustrates the influence of the delay time on the output. As expected, during the initial phase of control ($t^+ = 55$), the sensitivity is positive, indicating that an increase in delay time leads to an increase in velocity around that time. This can be explained since the actuation system responds with more delay. In contrast, in the final phase, an opposite trend can be observed: an increase in delay time leads to a decrease in the velocity of the controlled CASE. This behavior is due to the fact that the duration of the control effect is regulated mainly by the actuation frequency. For a given frequency, a shorter delay time limits the duration over which the control remains active.

Since Fig. 8(b) indicates that the sensitivity associated with the amplitude parameter is lower than that of the other parameters, an investigation was conducted to assess the error introduced when the amplitude parameter is excluded from the training phase. It is important to emphasize that this analysis is not equivalent to a standard input ablation study [21] or to sensitivity analyses leading to input ablation, such as those based on Shapley values [22, 23]. The key distinction lies in the fact that, in our case, a new network was trained to compensate for the absence of the amplitude-related input. In contrast, ablation studies typically involve setting the input to zero or to a value that physically corresponds to the removal of the respective input feature. In the present case, since it is not physically meaningful to define a value that represents the absence of the amplitude parameter inside the training bounds of Tab. 1, a conventional sensitivity analysis could not be performed. The new model was therefore trained by removing the input related to the voltage amplitude in the training phase and using the same hyperparameters discussed in section III. The network trained with

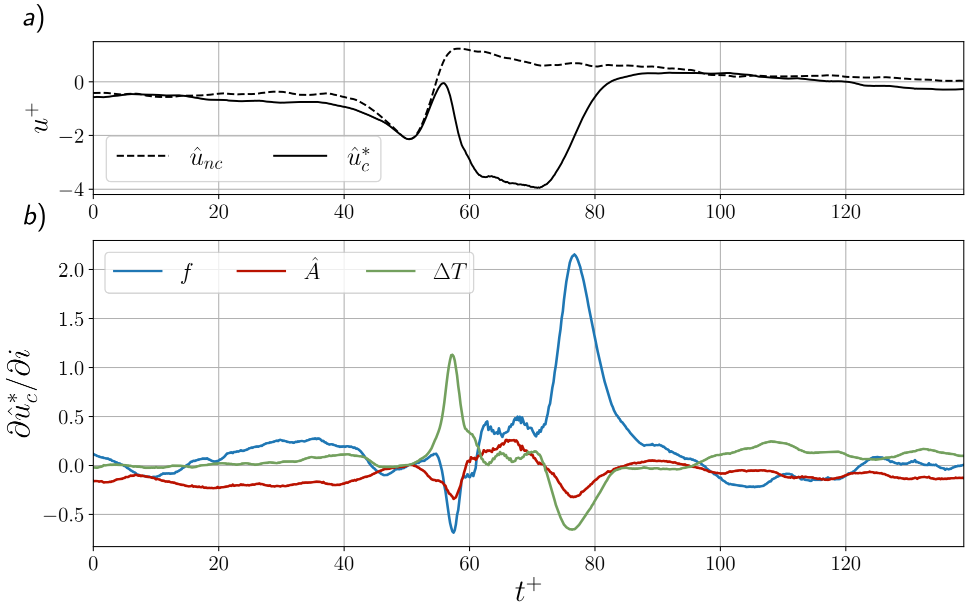

Fig. 8 a) Controlled CASE and uncontrolled CASE with ψ_{opt} as a function of the time t^+. b) Gradients of the output with respect to the i-th input (actuation frequency, voltage amplitude, and delay time) computed at ψ_{opt} as a function of the time.

all the input parameters (f, $\hat{A}$, ΔT) is denoted as $\mathcal{N}$. The new model trained without the voltage amplitude parameter, $\hat{A}$, is denoted as $\mathcal{N}^{\dagger}$. The training and validation datasets were kept identical to ensure that the evaluation was performed under the same conditions.

The Mean Absolute Percentage Error (MAPE) [24] for an output matrix Y composed of N_b CASEs of lenght N_t is defined as:

$$\mathrm{MAPE}(Y) = \frac{100}{N_b N_t} \sum_{i}^{N_b} \sum_{j}^{N_t} \frac{|Y_{ij} - \widetilde{Y}_{ij}|}{\widetilde{Y}_{ij}} \tag{4}$$

where $\tilde{Y}$ is the ground truth (the experimentally sampled controlled matrix of CASEs). The MAPE computed by evaluating $\mathcal{N}$ on the validation dataset was 0.76%, while the MAPE computed by evaluating $\mathcal{N}^{\dagger}$ in the same validation dataset was 0.80%. The difference in performance between the two networks is relatively small, confirming the indication provided by the sensitivity analysis based on the Jacobian matrix: the input feature associated with the voltage amplitude $\hat{A}$ plays a limited role in predicting the output response in the training bounds of Tab. 1. Its absence can therefore be partially compensated by the network during training, resulting in only a marginal increase in error, which remains remarkably low. Furthermore, in Fig. 9, a comparison was performed between $\mathcal{N}$ and $\mathcal{N}^{\dagger}$ with the same inputs taken from the validation set. In this figure, the CASEs are reported as follows: $\hat{u}_{\mathrm{nc}}$ for the uncontrolled CASE, $\hat{u}_{\mathrm{c}}$ for the experimentally sampled controlled CASE, which serves as the ground truth, $\hat{u}_{\mathrm{c}}^*$ for the controlled CASE predicted with $\mathcal{N}$, and $\hat{u}_c^{*\dagger}$ for the controlled CASE predicted with $\mathcal{N}^{\dagger}$. Overall, the main discrepancies between $\hat{u}_c^*$ and $\hat{u}_c^{*\dagger}$ occur near the lower velocity peak at $t^+ \approx 70$, where there is the minimum velocity induced by the control effect, and around $t^+ \approx 55$ where the velocity start to reduce. The error increases near $t^+ \approx 70$ because, at this location, the jet that modifies the sweep event reaches its maximum velocity, which is closely related to the amplitude voltage imposed. Since the network does not have access to that value, the uncertainty in the prediction within this region is higher with respect to that obtained through network $\mathcal{N}$, resulting in a larger error. The results observed in Fig. 9 are consistent with those shown in Fig. 8(b), which indicate that the maximum sensitivity related to the amplitude occurs at $t^+ \approx 55$ and $t^+ \approx 75$.

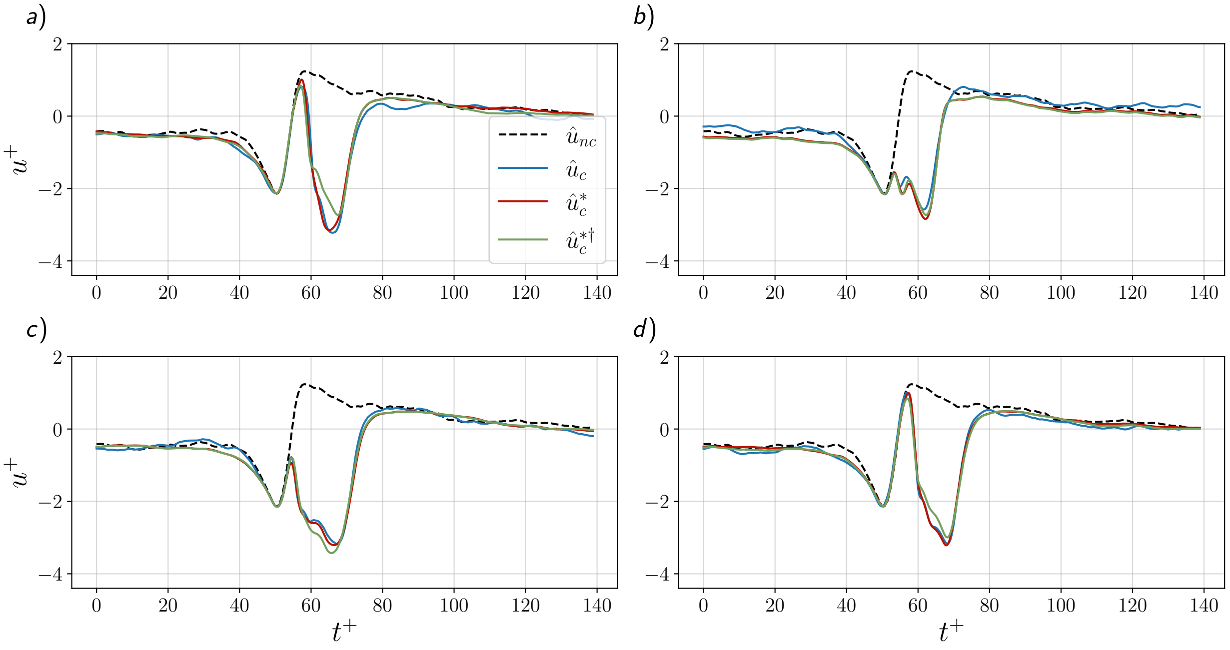

Fig. 9 Comparison between the CASEs predicted by $\mathcal{N}$ and $\mathcal{N}^\dagger$. The dashed black line represents the uncontrolled CASE, the blue line shows the experimentally sampled controlled CASE, the red line corresponds to the predicted controlled CASE from inputs belonging to the validation dataset by using $\mathcal{N}$, and the green line corresponds to the predicted controlled CASE from inputs belonging to the validation dataset by using $\mathcal{N}^\dagger$. a) $\psi = [54.68\,\text{Hz}, 50.4\,\text{mV}_{\mathbf{pp}}, 2.93\,\text{ms}]$**; b)** $\psi = [59.11\,\text{Hz}, 65.6\,\text{mV}_{\mathbf{pp}}, 0.23\,\text{ms}]$**; c)** $\psi = [38.55\,\text{Hz}, 67.8\,\text{mV}_{\mathbf{pp}}, 0.40\,\text{ms}]$**; d)** $\psi = [48.74\,\text{Hz}, 58.2\,\text{mV}_{\mathbf{pp}}, 2.674\,\text{ms}]$**.**

V. Conclusions

In the context of an opposition control strategy, a FNN was trained to be able to predict the conditionally averaged sweep event using the control parameters (frequency, voltage amplitude and delay time) as inputs. A total of 130 experiments in a fully turbulent channel flow at $Re_\tau = 365$ were carried out to create the database used for the model training and its validation. For each experiment, a random control triplet was sampled within the database bounds, and the controlled CASE signature was stored in a database. The FFN model was trained using the experiments included in the training database. After tuning the hyperparameters, the architecture that yielded the lowest error while preventing overfitting consisted of two hidden layers with 128 and 256 neurons, respectively. The MAPE metric was employed to evaluate the performance of the model. The trained model scored a MAPE of 0.76% in the validation database, indicating that the model produced an accurate prediction of the controlled CASE. Since the model is reliable and relatively fast to evaluate, we were able to conduct an offline optimization using a Bayesian optimization algorithm to find the control set of parameters that reduces the most the sweep event intensity inside the LHS bounds of Tab. 1. The optimized triplet $\psi_{\text{opt}} = \left[30\,\text{Hz}, 0.062\,\text{V}_{\text{pp}}, 0.83\,\text{ms}\right]$ yield to a reward up to 97%. We then interpolated the value of the cost function as a function of the control parameters, and we observed that the frequency, responsible for the blowing period, is the parameter that contributes the most to reducing the value of the cost function. We also noted that the contribution of the voltage amplitude and the delay time, albeit to a minor extent, leads to a further improvement. A sensitivity analysis based on the Jacobian matrix confirmed that the frequency is the most influential parameter, while the voltage amplitude has the least effect. We observed that training a model without the voltage amplitude input with the same hyperparameters as the original one, results in a model that is still partially able to compensate for the absence of the voltage amplitude and leads to a MAPE of 0.80%. A comparison between the two models shows that the error is located primarily around the negative peak of the controlled CASE (for $t^+ \approx 70$). This behavior is due to the fact that, at this time, the jet reaches its maximum wall normal velocity, which is strongly related to the voltage amplitude.

Acknowledgments

This publication is part of the project PNRR-NGEU which has received funding from the MUR – DM 118/2023.

References

[1] Cheng, X., Qiao, Z., Zhang, X., Quadrio, M., and Zhou, Y., "Skin-friction reduction using periodic blowing through streamwise slits," *Journal of Fluid Mechanics*, Vol. 920, 2021, p. A50. https://doi.org/10.1017/jfm.2021.439.

[2] Baron, A., and Quadrio, M., "Turbulent drag reduction by spanwise wall oscillations," *Applied Scientific Research*, Vol. 55, 1995, pp. 311–326.

[3] Schoppa, W., and Hussain, F., "Coherent structure generation in near-wall turbulence," *Journal of fluid Mechanics*, Vol. 453, 2002, pp. 57–108.

[4] Abbassi, M., Baars, W., Hutchins, N., and Marusic, I., "Skin-friction drag reduction in a high-Reynolds-number turbulent boundary layer via real-time control of large-scale structures," *International Journal of Heat and Fluid Flow*, Vol. 67, 2017, pp. 30–41. https://doi.org/https://doi.org/10.1016/j.ijheatfluidflow.2017.05.003, URL https://www.sciencedirect.com/science/article/pii/S0142727X17303405.

[5] Wang, Y., Choi, K.-S., Gaster, M., Atkin, C., Borodulin, V., and Kachanov, Y., "Opposition control of turbulent spots," *Journal of Fluid Mechanics*, Vol. 943, 2022, p. A3.

[6] Choi, H., Moin, P., and Kim, J., "Active turbulence control for drag reduction in wall bounded flows," *Journal of Fluid Mechanics*, Vol. 75, 1994, p. 262.

[7] Rebbeck, H., and Choi, K.-S., "Opposition control of near-wall turbulence with a piston-type actuator," *Physics of Fluids*, Vol. 13, No. 8, 2001, pp. 2142–2145. https://doi.org/10.1063/1.1381563, URL https://doi.org/10.1063/1.1381563.

[8] Rebbeck, H., and Choi, K.-S., "A wind-tunnel experiment on real-time opposition control of turbulence," *Physics of Fluids*, Vol. 18, 2006, p. 035103.

[9] Brunton, S. L., Noack, B. R., and Koumoutsakos, P., "Machine Learning for Fluid Mechanics," *Annual Review of Fluid Mechanics*, Vol. 52, No. Volume 52, 2020, 2020, pp. 477–508. https://doi.org/https://doi.org/10.1146/annurev-fluid-010719-060214, URL https://www.annualreviews.org/content/journals/10.1146/annurev-fluid-010719-060214.

[10] Martín-Gil, A., and Flores, O., "Predicting the skin friction's evolution in a forced turbulent channel flow," *Computers Fluids*, Vol. 284, 2024, p. 106417. https://doi.org/https://doi.org/10.1016/j.compfluid.2024.106417, URL https://www.sciencedirect.com/science/article/pii/S0045793024002482.

[11] Lee, C., Kim, J., Babcock, D., and Goodman, R., "Application of neural networks to turbulence control for drag reduction," *Physics of Fluids*, Vol. 9, No. 6, 1997, pp. 1740–1747. https://doi.org/10.1063/1.869290, URL https://doi.org/10.1063/1.869290.

[12] Park, J., and Choi, H., "Machine-learning-based feedback control for drag reduction in a turbulent channel flow," *Journal of Fluid Mechanics*, Vol. 904, 2020, p. A24. https://doi.org/10.1017/jfm.2020.690.

[13] Guastoni, L., Güemes, A., Ianiro, A., Discetti, S., Schlatter, P., Azizpour, H., and Vinuesa, R., "Convolutional-network models to predict wall-bounded turbulence from wall quantities," *Journal of Fluid Mechanics*, Vol. 928, 2021, p. A27. https://doi.org/10.1017/jfm.2021.812.

[14] Cuéllar, A., Güemes, A., Ianiro, A., Flores, , Vinuesa, R., and Discetti, S., "Three-dimensional generative adversarial networks for turbulent flow estimation from wall measurements," *Journal of Fluid Mechanics*, Vol. 991, 2024, p. A1. https://doi.org/10.1017/jfm.2024.432.

[15] Saccaggi, E., and Di Cicca, G. M., "Genetic algorithm optimization of an opposition control strategy in a fully turbulent channel flow," *Proceedings of the 34th Congress of the International Council of the Aeronautical Sciences*, ICAS, Florence, Italy, 2024, pp. 1–17. URL https://www.icas.org/ICAS_ARCHIVE/ICAS2024/data/papers/ICAS2024_1208_paper.pdf.

[16] Blackwelder, R. F., and Kaplan, R. E., "On the wall structure of the turbulent boundary layer," *Journal of Fluid Mechanics*, Vol. 76, No. 1, 1976, p. 89–112. https://doi.org/10.1017/S0022112076003145.

[17] McKay, M. D., Beckman, R. J., and Conover, W. J., "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, Vol. 42, No. 1, 2000, pp. 55–61.

[18] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," , 2017. URL https://arxiv.org/abs/1412.6980.

[19] Head, T., Kumar, M., Nahrstaedt, H., Louppe, G., and Shcherbatyi, I., "scikit-optimize/scikit-optimize," , Oct. 2021. https://doi.org/10.5281/zenodo.5565057, URL https://doi.org/10.5281/zenodo.5565057.

[20] Williams, C., and Rasmussen, C., *Gaussian processes for machine learning*, Vol. 2, MIT press Cambridge, MA, 2006.

[21] Meyes, R., Lu, M., de Puiseau, C. W., and Meisen, T., "Ablation studies in artificial neural networks," *arXiv preprint arXiv:1901.08644*, 2019.

[22] Shapley, L. S., "Notes on the n-person game—ii: The value of an n-person game," 1951.

[23] Lundberg, S. M., and Lee, S.-I., "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, Vol. 30, 2017.

[24] Botchkarev, A., "Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology," *arXiv preprint arXiv:1809.03006*, 2018.