Neural Network-Based Prediction of Bursting Events in a Turbulent Channel Flow

Enrico SACCAGGI[1,a *] and Gaetano Maria DI CICCA[1,b]

[1]Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino, Italy

[a]enrico.saccaggi@polito.it, [b]gaetano.dicicca@polito.it

Keywords: Wall Turbulence, Flow Control, Machine Learning

Abstract. This study investigates the use of a hybrid architecture of a Feed-forward Neural Network (FNN) combined with the Proper Orthogonal Decomposition (POD) to estimate the effect of control on sweep events within the context of an opposition control strategy. The experiments were carried out in a fully turbulent channel flow with a friction Reynolds number of 365. The POD was conducted to reduce the database dimensionality, and then a FNN was trained to predict the non-normalized spatial modes matrix, subsequently allowing the reconstruction of the Conditionally Averaged Sweep Event (CASE). The results demonstrate that the hybrid neural network is capable of reproducing the control signature for different actuation parameters with an accuracy comparable to a conventional neural network, while requiring approximately 90 times fewer trainable parameters.

Introduction

Choi et al. [1] through direct numerical simulations, demonstrated that the opposition control technique can achieve skin-friction reductions up to 25% by applying wall blowing and suction with velocities equal in magnitude but opposite in direction to those at a prescribed wall-normal location. Wind tunnel experiments reported in [2, 3] further demonstrated that real-time opposition control, implemented through wall-normal jet actuation to block high-speed fluid motions during sweeps, effectively mitigates the intensity of these events. In recent years, data-driven models have shown success in estimating the response of control systems to given actuation parameters. A FNN was implemented in [4] to predict the effect of actuation parameters on the CASE. The objective of the present work is to extend that study by reducing the dimensionality of the FNN by changing the model architecture. Furthermore, the possibility of further reducing the dimensionality is explored through the use of a hybrid POD-FNN, which first predicts the non-normalized spatial modes matrix of the POD and then reconstructs the CASE, while maintaining nearly the same accuracy.

Experimental setup

The experiments were conducted in an 8-meter-long duct with a 7 cm x 30 cm rectangular cross-section (aspect ratio equal to 4.28). The center-line velocity was set to $U_0 = 3.22$ m/s corresponding to a friction velocity of $u_\tau = 0.16$ m/s and a resulting friction Reynolds number of $Re_\tau = 365$. Two hot-wire probes were employed, each connected to a Dantec 55M10 CTA Standard Bridge anemometer. Both probes featured a sensitive tungsten wire with a diameter of 5 μm and a length of 1.25 mm. The upstream probe, a Dantec 55P11, was positioned at $x^+ = -65$ and $y^+ = 15$, while the downstream probe, a Dantec 55P15 boundary-layer type, was mounted on a two-axis stand. The superscript + denotes quantities in wall units. Following [3], sweep events were detected in real time using the velocity-gradient technique applied to the streamwise velocity signal acquired from the upstream probe. An analog differentiator computed the gradient from the hot-wire anemometer voltage output, and a comparator identified events when the time derivative exceeded a threshold. The threshold was calibrated to match the event count from the variable interval time averaging method [5] with k=1.0 and averaging time T^+=10. A detected sweep event generated a TTL signal sent to an Arduino Uno Rev3 board, introducing a delay ΔT to take into account convection time and actuator response time. The delayed TTL signal triggered a signal generator to produce a single-period sine wave of frequency f and voltage amplitude $\hat{A}$, amplified to drive a loudspeaker. The loudspeaker produced a jet through a 1 mm diameter orifice (corresponding to about 11 viscous units). The rear probe velocity signal was acquired and conditionally averaged to visualize the sweep event.

Database description and model creation

The database used in this study is the one presented by [4], constructed by varying the control parameters. The objective is to predict the longitudinal component of the velocity signal for the controlled CASE, $\hat{u}_c \in \mathbb{R}^{1\times n_t}$, using as inputs the actuation parameters and the longitudinal component of the velocity signal for the uncontrolled CASE, $\hat{u}_{nc} \in \mathbb{R}^{1\times n_t}$, where $n_t = 750$ is the number of points in the conditionally averaged window and corresponds to 139 viscous units. Let the control triplet be denoted as θ, characterized by actuation frequency f, voltage amplitude $\hat{A}$, and delay time ΔT. A total of $n_d = 130$ control triplets were sampled to construct the input part of the dataset using a Latin Hypercube Sampling technique within the space 30÷60 Hz, 50÷75mV_{pp} and 0.15÷3ms for frequency, voltage amplitude and delay time respectively. For each dataset sample, $\hat{u}_c(\theta)$ was experimentally evaluated after a conditional sampling of the rear probe velocity signal acquired at $x^+ = 65$ and $y^+ = 15$. The resulting dataset was divided into training and validation subsets, containing 80% and 20% of the data respectively, and normalized to have all the features in the range between 0 and 1. A FNN with a reduced number of trainable parameters compared to the model proposed in [4] was adopted to predict the control effect on the CASEs.

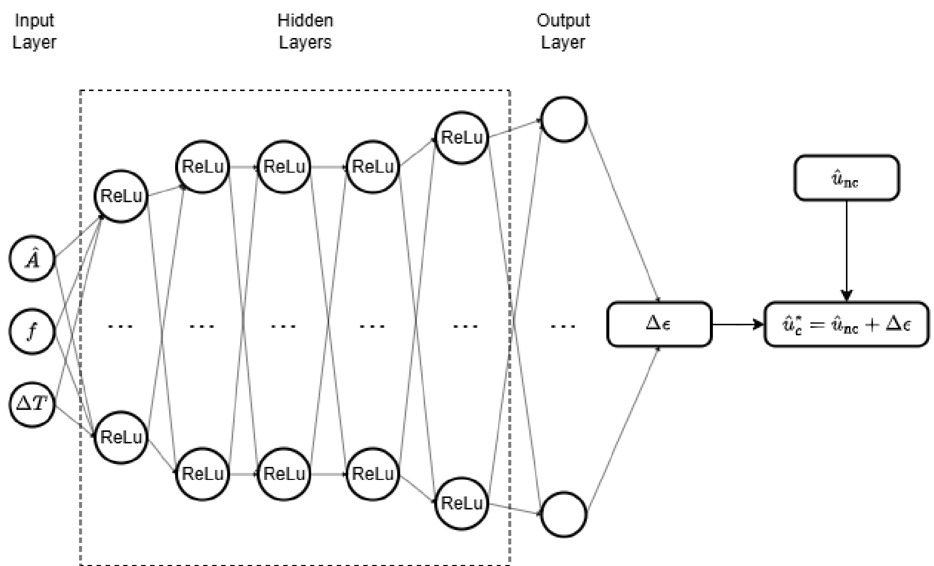

Figure 1. FNN structure used to predict the longitudinal component of the velocity signal for the controlled CASE $\hat{u}_c^*$ based on the control parameters and the longitudinal component of the velocity signal for the uncontrolled CASE, $\hat{u}_{nc}$.

The network architecture here implemented, is illustrated in Fig. 1 and comprises 5 hidden layers of dimension (16, 32, 32, 32, 64) followed by an output layer which produces as output $\Delta\epsilon$. $\Delta\epsilon$ is then added to $\hat{u}_{nc}$ to give the predicted controlled CASE signal, represented as $\hat{u}_c^*$ for a given triplet θ. The hyperparameters of the network, including learning rate, batch size, and activation functions, are the same as [4]. This network will be referred to hereafter as $\mathcal{N}$. Similar results to those obtained with the network presented in [4] were achieved using $\mathcal{N}$, which features a deeper architecture with fewer neurons, leading to a reduction in the number of trainable parameters from approximately 225,000 to 53,000 (over four times fewer).

Hybrid POD-FNN model

To further reduce the size of the model, it is possible to decrease the dimensionality of the output part of the dataset. The first step is therefore to compress this dataset from $\widehat{\boldsymbol{U}}_{\boldsymbol{c}} \in \mathbb{R}^{n_d \times n_t}$ to $\widetilde{\widehat{\boldsymbol{U}}}_{\boldsymbol{c}} \in \mathbb{R}^{n_d \times n_r}$ where $n_r \ll n_t$ denotes the reduced temporal dimension of the output part of the dataset. POD [6] was then applied to decompose the dataset according to:

$$\widehat{\boldsymbol{U}}_{\boldsymbol{c}} = \boldsymbol{\phi} \boldsymbol{\Sigma} \boldsymbol{\psi}^{\boldsymbol{T}}, \quad (1)$$

where $\boldsymbol{\phi} \in \mathbb{R}^{n_d \times n_d}$ denotes the matrix of pseudo-spatial basis vectors, $\boldsymbol{\Sigma} \in \mathbb{R}^{n_d \times n_d}$ is the singular value matrix, and $\boldsymbol{\psi} \in \mathbb{R}^{n_t \times n_d}$ is the matrix of the temporal modes. A reduced-order approximation of the dataset, retaining only the first n_r most energetic modes, is given by:

$$\widetilde{\widehat{\boldsymbol{U}}}_{\boldsymbol{c}} = \widetilde{\boldsymbol{\phi}} \widetilde{\boldsymbol{\Sigma}} \widetilde{\boldsymbol{\psi}}^{\boldsymbol{T}}, \quad (2)$$

where $\widetilde{\boldsymbol{\phi}} \in \mathbb{R}^{n_d \times n_r}$, $\widetilde{\boldsymbol{\Sigma}} \in \mathbb{R}^{n_r \times n_r}$ and $\widetilde{\boldsymbol{\psi}} \in \mathbb{R}^{n_t \times n_r}$ contains only the first n_r columns or singular values. Analysis reveals that retaining only the first 7 modes (n_r = 7) is sufficient to capture more than 99% of the total energy. A reconstruction of $\hat{u}_c$ taking into account only the first 7 modes is presented in Fig. 2.

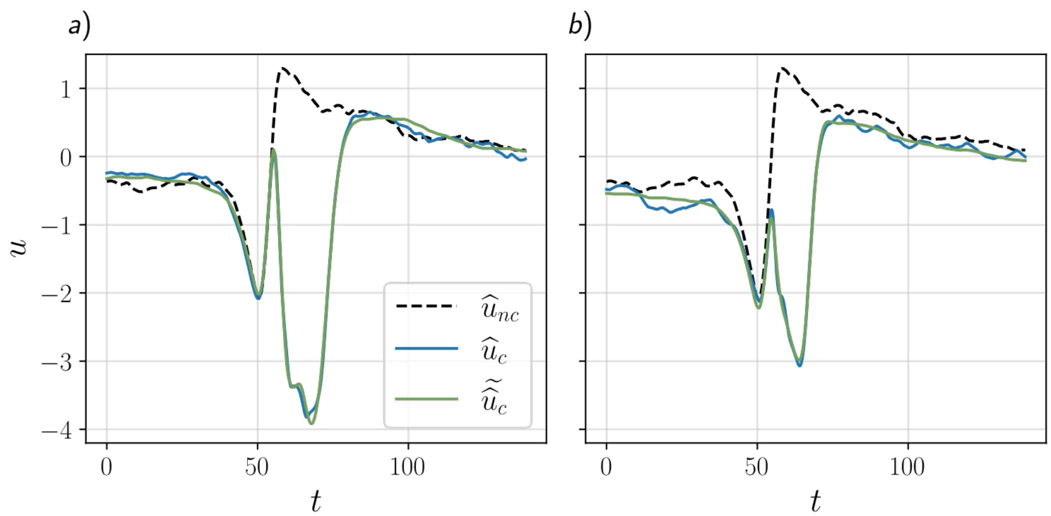

Figure 2. Longitudinal component of the velocity signal (in viscous units) for the CASEs. The superscript + is here omitted for both u and t. The black dashed lines and the blue continuous lines represent the experimentally sampled uncontrolled and controlled CASE, respectively. The green lines indicate the reconstructed CASE with the first 7 POD modes.
a) $\theta = [33.3\,Hz,\ 56.44\,mV_{pp},\ 0.65\,ms]$, *b)* $\theta = [52.49\,Hz,\ 61.59\,mV_{pp},\ 0.95\,ms]$.

Relying on this low-dimensional representation, it becomes possible to predict $\widetilde{\boldsymbol{\alpha}} = \widetilde{\boldsymbol{\phi}} \widetilde{\boldsymbol{\Sigma}} \in \mathbb{R}^{n_d \times n_r}$ from the input-parameter dataset $\boldsymbol{X} \in \mathbb{R}^{n_d \times n_i}$, where n_i=3 denotes the number of control parameters, and subsequently reconstruct $\tilde{\hat{u}}_c$ using regressors such as decision trees [7] or XGBoost regressors [8]. Although this strategy is computationally lightweight, it suffers from significant issues related to the regularization of the cost function when predicting the coefficients of $\widetilde{\boldsymbol{\alpha}}$ since each coefficient must be weighted. An alternative approach that avoids a direct regularization is to predict the coefficients of $\widetilde{\boldsymbol{\alpha}}$ using a FNN, then reconstruct $\tilde{\hat{u}}_c$ and compare the reconstruction with $\hat{u}_c$ within the loss function. In this case, the regularization is implicitly enforced by the training procedure. A schematic of this hybrid POD–FNN architecture is shown in Fig. 3**.** Different architectures of the hybrid POD–FNN model were evaluated by varying the number of neurons per layer and the number of hidden layers. The learning rate and weight decay were kept identical to those used to train $\mathcal{N}$, while a greater number of training epochs (30000 vs 10000) was adopted since the model contains fewer trainable parameters. In addition an early stopping criterion was implemented to avoid overfitting [9].

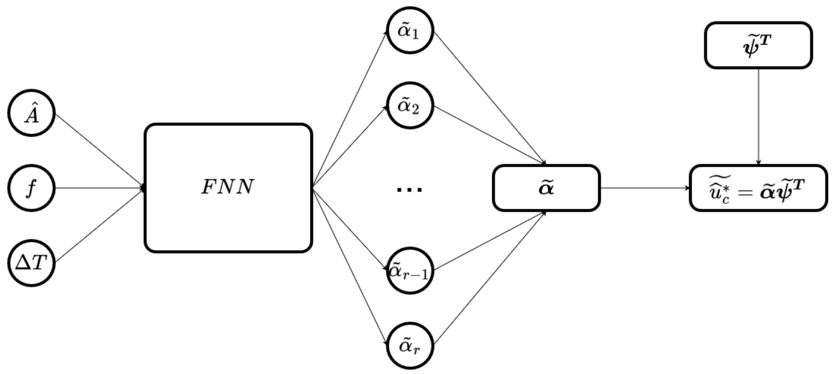

Figure 3. Schematic of the hybrid POD-FNN model. Inputs are the actuation parameters and the temporal modes. Outputs are the predicted controlled CASEs.

Models were assessed on the validation dataset using R^2 and the adjusted R^2 named R^2_{adj} as metrics [10]. R^2_{adj} accounts for the number of trainable parameters and thus penalizes excessively large networks. In Table 1, the metrics R^2 and R^2_{adj} are reported for the tested architectures identified through the sequence (*a, b, c*) where *a* is the number of neurons in the first hidden layer, *b* in the second, and *c* in the third. As expected, R^2 increases with the number of parameters. Conversely, R^2_{adj}, increases up to the architecture (8, 16, 16), named $\widetilde{\mathcal{N}}$, and then slightly decreases, even though the R^2 continues to improve. This behavior reflects the penalization term included in R^2_{adj}, whose denominator increases more rapidly than the improvement in the R^2 value due to the higher amount of trainable parameters. A comparison between the CASEs predicted by $\mathcal{N}$ and $\widetilde{\mathcal{N}}$ for the inputs in the validation dataset is shown in Fig. 4. The measured quantity to be approximated by the FNN, $\hat{u}_c$, is represented by a solid blue line, while the solid red and green lines correspond to the predicted controlled CASE obtained by $\mathcal{N}$ and $\widetilde{\mathcal{N}}$, denoted as $\hat{u}_c^*$ and $\widetilde{\hat{u}}_c^*$, respectively. The comparison between the two curves indicates that both networks successfully capture the dynamics of the system, providing an accurate representation of the controlled sweep event. While $\mathcal{N}$ is slightly more accurate than $\widetilde{\mathcal{N}}$, as reflected by its R^2 value of 0.977 compared to 0.976 for $\widetilde{\mathcal{N}}$, this marginal improvement comes at the cost of a model over 90 times larger in terms of trainable parameters.

Table 1. Comparison between different hybrid POD-FNN models in terms of network architectures, number of parameters, and performance metrics, sorted by the number of parameters. The metrics were computed using the validation dataset.

Architecture	Params	R^2	R^2_{adj}
(4, 4)	71	0.963	0.963
(4, 4, 4)	91	0.965	0.965
(4, 8)	119	0.971	0.971
(8, 8)	167	0.972	0.972
(4, 8, 8)	191	0.974	0.973
(8, 8, 8)	239	0.974	0.974
(8, 16)	295	0.975	0.974
(16, 16)	455	0.975	0.974
(8, 16, 16)	567	0.976	0.975
(16, 16, 16)	727	0.974	0.973
(16, 32)	839	0.975	0.974
(32, 32)	1415	0.974	0.972
(16, 32, 32)	1895	0.976	0.973
(32, 32, 32)	2471	0.976	0.972
(32, 64)	2695	0.975	0.972
(32, 64, 64)	6855	0.976	0.963

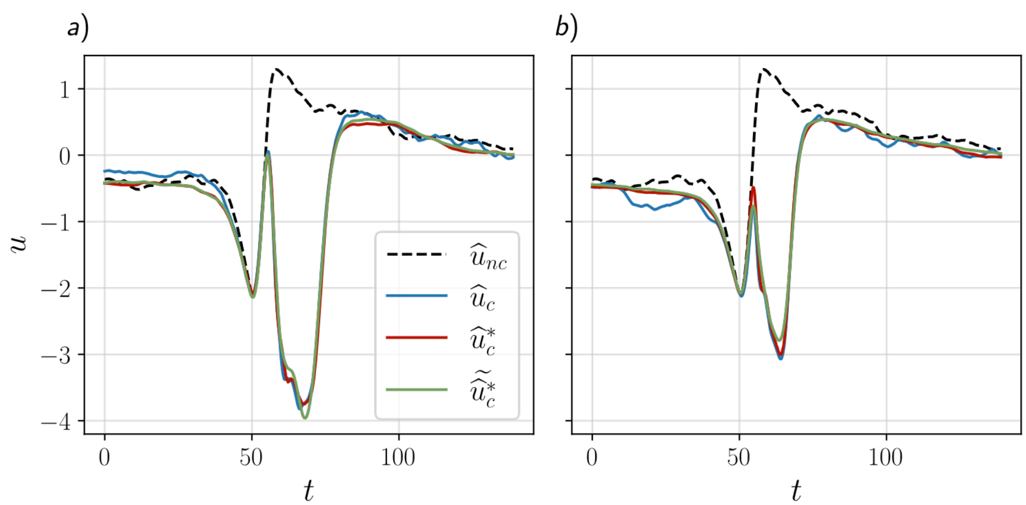

Figure 4. Evaluation of $\mathcal{N}$ and $\widetilde{\mathcal{N}}$ in the validation dataset. The black dashed line represents the uncontrolled CASE. The blue lines corresponds to the experimentally sampled controlled CASEs. The red and green lines indicate the predicted controlled CASEs with $\mathcal{N}$ and $\widetilde{\mathcal{N}}$ respectively. a) $\theta = [33.3\ Hz,\ 56.44\ mV_{pp},\ 0.65\ ms]$, b) $\theta = [52.49\ Hz,\ 61.59\ mV_{pp},\ 0.95\ ms]$.

Conclusion

In the context of opposition control, and with the goal of predicting the controlled CASE more efficiently, the first step consisted in improving the network architecture used in [4], thus reducing the number of trainable parameters by a factor of approximately four. Subsequently, aiming for an even more lightweight model, a POD analysis was performed, revealing that only 7 modes are sufficient to capture 99% of the output dataset energy. Building on this insight, a significantly smaller FNN was trained, featuring roughly 90 times fewer trainable parameters than those adopted in $\mathcal{N}$ while achieving a comparable R^2 value.

Acknowledgments
This publication is part of the project PNRR-NGEU which has received funding from the MUR – DM 118/2023. We acknowledge the work of the student Arthur Pierre Philippe Nuvoloni for his help in defining the architecture of network $\mathcal{N}$.

References

[1] H. Choi, P. Moin, and J. Kim. Active turbulence control for drag reduction in wall bounded flows. J. Fluid Mech., 75:262, 1994. https://doi.org/10.1017/S0022112094000431

[2] H. Rebbeck and K.-S. Choi. Opposition control of near-wall turbulence with a piston-type actuator. Phys. Fluids, 13(8):2142-2145, 2001. https://doi.org/10.1063/1.1381563

[3] H. Rebbeck and K.-S. Choi. A wind-tunnel experiment on real-time opposition control of turbulence. Phys. Fluids, 18:035103, 2006. https://doi.org/10.1063/1.2173295

[4] E. Saccaggi and G. M. Di Cicca. Application of a neural network for predicting the control effect on sweep events in a fully turbulent channel flow. In AIAA AVIATION FORUM AND ASCEND2025, Las Vegas, NV, USA, July 2025. American Institute of Aeronautics and Astronautics (AIAA). AIAA 2025-3022. https://doi.org/10.2514/6.2025-3022

[5] R. F. Blackwelder and R. E. Kaplan. On the wall structure of the turbulent boundary layer. J. Fluid Mech., 76(1):89-112, July 1976. https://doi.org/10.1017/S0022112076003145

[6] G. Berkooz, P. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. Annu. Rev. Fluid Mech., 25:539-575, 1993. https://doi.org/10.1146/annurev.fl.25.010193.002543

[7] J. R. Quinlan. Induction of decision trees. Machine learning, 1(1):81-106, 1986. https://doi.org/10.1023/A:1022643204877

[8] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794, 2016. https://doi.org/10.1145/2939672.2939785

[9] L. Prechelt. Early stopping-but when?. In: Neural Networks: Tricks of the trade. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 55-69. https://doi.org/10.1007/3-540-49430-8_3

[10] M. Ezekiel and K. A. Fox. Methods of correlation and regression analysis. Linear and curvilinear, 1959.