EPTT 2024
Escola de Primavera de
Transição e Turbulência

ABCM
Associação Brasileira de Engenharia e Ciências Mecânicas

EPTT-2024-0052
Neural Networks as Flow Controllers: a Study on Robustness

Tarcísio Déda
William Wolf
Universidade Estadual de Campinas, UNICAMP, Campinas, SP, 13083-860, Brazil
tarcisio.deda@gmail.com, wolf@fem.unicamp.br

Scott T. M. Dawson
Illinois Institute of Technology, Chicago, IL, 60616, United States
scott.dawson@iit.edu

In this extended abstract, we employ neural network surrogate models (NNSMs) for the purpose of modelling nonlinear dynamical systems. The proposed framework allows for training neural network controllers (NNCs) through backpropagation of the differentiable models. The training stage proposed consists of an iterative proccess that gadrually improves the estimation of the system dynamics in regions increasingly closer to an unstable equilibrium point. The equilibrium point is automatically computed through Newton method enabled by the linearization of the NNSMs around an initial guess. By using an input layer with L1 regularization, we aim to obtain a sparse configuration of sensors from an initial set that cheapens operations like matrix inversions of high order systems, which is required when dealing with high order systems. The main goal of the present work is to study the robustness characteristics of the proposed control approach. First, we show the main results involving a series of tests conducted with the Lorenz system, in which we verify the behavior of the closed-loop system subject to phenomena such as plant variations, measurement noise and disturbances. A controller is then trained to stabilize a confined cylinder flow at Reynolds number $\mathrm{Re} = 150$. *We subsequently demonstrate that the controller is able to stabilize unstable flows at lower and higher Reynolds numbers than that for which the NNC was trained. As the Reynolds number increases, we find that reducing the timestep at which control is applied can be required to achieve complete stabilization.*

Keywords: *Flow control, machine learning, nonlinear control*

1. INTRODUCTION

Unsteady flows can contain a wide range of temporal and spatial scales which are governed by nonlinear dynamics. Such nonlinearities, together with the high dimensionality of CFD models, pose difficulties for the development and application of flow control strategies. Machine learning can be a powerful tool for the development of surrogate models and flow control. For example, there is continued and growing interest in the development and application of reinforcement learning-based methods to control fluid flows Garnier *et al.* (2021); Li and Zhang (2022); Bae and Koumoutsakos (2022); Chatzimanolakis *et al.* (2024); Suárez *et al.* (2024); Vinuesa (2024). Focusing on one common test case employing a grid of velocity sensors along the wake of cylinder flows, Refs. Ren *et al.* (2021); Tokarev *et al.* (2020); Xu *et al.* (2020); Rabault *et al.* (2019); Déda *et al.* (2024) demonstrate the application of closed loop drag control by employing deep reinforcement learning with different actuation setups.

Beyond reinforcement learning, there are other methods that have been recently employed to control fluid flows. Aiming to limit the complexity of the control setup, Bieker *et al.* (2020) propose the usage of limited sensor data by employing delay coordinates with the goal of reducing the aerodynamic asymmetry of a fluidic pinball. By applying model predictive control (MPC), it is shown that results can be improved by iteratively retraining a data-based model of the lift dynamics as the system is controlled in closed loop. MPC was applied by Morton *et al.* (2018) by developing models based on Koopman operators that can learn the dynamics of sampled images from a cylinder flow. Delay coordinates for control with limited probing is also applied with machine learning techniques by Déda *et al.* (2023). For a general overview of the application of machine learning methods for the control of fluid flows, we refer to Refs. Brunton and Noack (2015); Brunton *et al.* (2020).

In the present work we take a neural-network-based approach that is largely distinct from those mentioned above.

In particular, we study the application of neural network surrogate models (NNSMs) to train neural network controllers (NNCs) capable of stabilizing flows, as introduced and described in Déda *et al.* (2023, 2024). During the training process of the NNSMs, sensor selection is conducted in order to reduce the number of probes required for control and estimation of the dynamics. An iterative approach for training is presented, where data is sampled by perturbing the controlled system and, therefore, overcoming the lack of data near equilibrium, which is an important region as we aim for stabilization. We also introduce a reduced order equilibrium estimation to provide a set point for NNC training without the need to employ more complex techniques such as selective frequency damping (Åkervik *et al.*, 2006; Li and Zhang, 2022). An assessment of robustness is conducted for the proposed closed-loop approach.

There are several points of motivation for such a robustness-focused study. First, unlike traditional control-theoretic approaches, there is no rigorous theory that can be used to analyze or optimize robustness measures Zhou and Doyle (1998). Second, robustness is important for the potential application of such methods in experimental and/or real-world settings, where time delays, sensor and process noise, and parameter uncertainty could affect controller performance. In particular, it is often desirable for controllers to be effective over a range of parameters, for which robustness to parameter variation is desired. Last, with the recent development of several novel machine-learning-based control strategies, robustness is a metric that should be used for comparisons between them, a factor which is often not studied as extensively as direct performance.

Here, we first present a series of tests involving the chaotic Lorenz system, where we empirically verify the robustness of the controlled system to disturbances, data quantization, time step variations, measurement noise and plant variations. We also test the proposed approaches with a confined cylinder flow often studied in literature. The NNC trained for a configuration at Reynolds number $\mathrm{Re} = 150$ is employed for the task of controlling flow at different values of Re in order to assess the controller robustness to plant variations.

2. METHODOLOGY

In this section, we describe the methodology applied to design closed-loop controllers using neural networks.

2.1 Neural network surrogate models (NNSMs)

We utilize neural networks to learn the dynamics of discrete systems of the form

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k, \mathbf{u}_k)\,, \tag{1}$$

where $\mathbf{x}$ is the vector containing the system states, $\mathbf{u}$ is the vector of control inputs and the subscripts represent the discrete time steps such that time $t = k\Delta t$. The states vector $\mathbf{x}$ consists of a set of sensor measurements from which the dynamics can be inferred using data-driven methods. This approach therefore assumes that this set of sensor measurements is sufficient for representing the state of a surrogate model for the system dynamics. From a flow simulation, one can sample data at a fixed time step to perform the training of a neural network that can be represented by a nonlinear function $\tilde{F}$, the parameters of which are trained such that $\tilde{F} \approx F$. An open-loop sweep is conducted to sample the data needed for training the NNSM. To reduce the amount of states needed for the evaluation, we employ a sparsity layer that weights each input. To encourage a reduction in the number of probes, we propose a cost function of the form

$$\mathcal{L} = \frac{1}{n}\mathbf{g}\cdot\mathbf{g} + r_2\mathbf{w}_\mathrm{h}\cdot\mathbf{w}_\mathrm{h} + r_1\sum|\mathbf{w}_\mathrm{s}|\,, \tag{2}$$

where $\mathbf{g}$ is the array of differences between the labelled data and the NNSM output normalized by the standard deviation of the states sampled for the training data, $\mathbf{w}_\mathrm{h}$ is the stacked weights array for the hidden layers, $\mathbf{w}_\mathrm{s}$ is the array of weights for the sparsity layer, n is the total number of candidate sensors, and r_1 and r_2 are the L1 and L2 regularization factors.

2.2 Estimation of equilibria

Aiming to obtain an equilibrium estimation, we make use of the trained NNSM, which is reduced in complexity (order) when compared to the full order flow solver (since the NNSM is trained for a reduced number of states or probes). The equilibrium estimation is required in the present work since it allows us to obtain a control setpoint. While equilibria can be analytically computed for simple systems such as the Lorenz system that is considered in part of this work, equilibria (often unstable) typically must be computed numerically for fluid flows. In order to do so, here we employ a Newton method, which utilizes the linearization of $\tilde{F}$ through the computation of the Jacobian of the neural network via backpropagation. Further details of the approach can be found in Déda *et al.* (2024).

2.3 Neural Network Controller (NNC)

To train an appropriate NNC to bring the states to equilibrium in closed loop, we build a recurrent model as shown in figure 1. From a data set containing a range of initial conditions, we train the NNC weights to bring states closer to

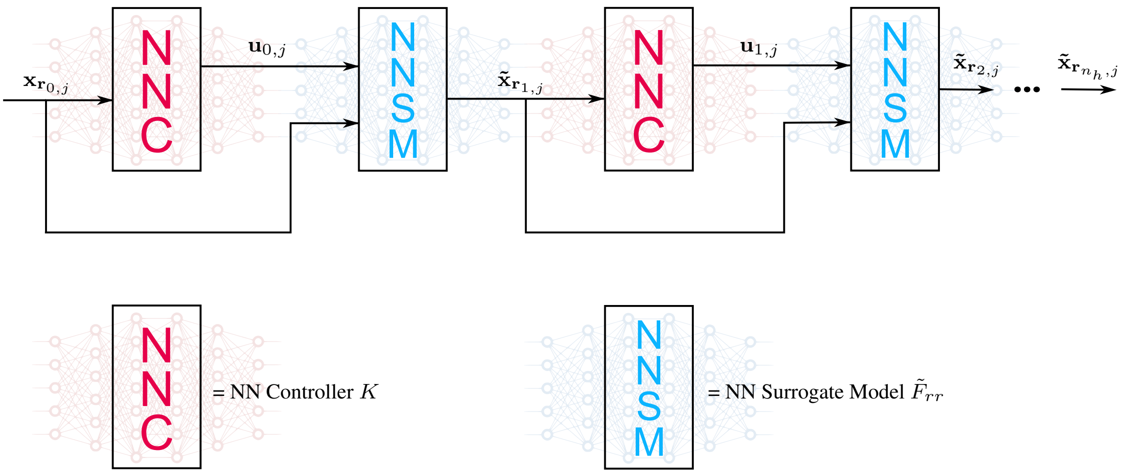

$\mathbf{x_r}_{0,j}$
NNC
$\mathbf{u}_{0,j}$
NNSM
$\tilde{\mathbf{x}}_{\mathbf{r}1,j}$
NNC
$\mathbf{u}_{1,j}$
NNSM
$\tilde{\mathbf{x}}_{\mathbf{r}2,j}$
...
$\tilde{\mathbf{x}}_{\mathbf{r}n_h,j}$
NNC
= NN Controller K
NNSM
= NN Surrogate Model $\tilde{F}_{rr}$

Figure 1. Schematic of the NNC training where the initial dataset is propagated through recurrent evaluations of the NNSM and NNC. Only the NNC weights and biases are updated during training in order to bring the states $\mathbf{x_r}_{1\leq i\leq n_h,j}$ closer to $\mathbf{x_r}^*$.

equilibrium along a finite discrete horizon n_h. The following loss function is targeted for minimization:

$$\mathcal{L} = \frac{1}{p\,n_h}\left(\frac{\mathbf{w_e}}{n}\cdot\sum_{j=1}^{p}\sum_{i=1}^{n_h}\mathbf{e}_{i,j}\circ\mathbf{e}_{i,j} + \frac{\mathbf{w_u}}{m}\cdot\sum_{j=1}^{p}\sum_{i=0}^{n_h-1}\mathbf{u}_{i,j}\circ\mathbf{u}_{i,j}\right)\,, \tag{3}$$

$$\mathbf{e}_{i,j} = \mathbf{w_f}\circ(\tilde{\mathbf{x}}_{i,j} - \mathbf{x}^*)\,, \tag{4}$$

where p is the size of the training dataset. Here, $\mathbf{w_f}$ is an array consisting of the inverse of the standard deviations of the data sampled for training. As in the NNSM case, this is done in order to avoid states with larger numerical amplitudes from being more important in the cost function. Furthermore, the weight vector $\mathbf{w_e}$ has all elements equal to $1/n_r$, which averages the result of the sum using an equal weighting of the states. Similarly, $\mathbf{w_u}$ has all elements equal to w_u, a scalar hyperparameter used to penalise the control inputs.

2.4 Iterative training

More than one step of training for the NNSM and NNC is required since the data sampling might provide a set of states that are considerably far from the equilibrium point – and therefore might lead to estimates of equilibrium that are not precise enough. To address this issue, we take advantage of the trained NNC. After the first application of the training process (for both NNSM and NNC), the NNC may not work properly for stabilization, since the equilibrium estimation may not be precise. However, if the NNC manages to at least bring states closer to the actual equilibrium, it can be utilized to enable the collection of additional data closer to the desired equilibrium point. We achieve this by perturbing the controlled system to obtain more data, which is subsequently used to retrain the NNSM and the NNC. This process can be repeated several times to obtain models that are more accurate in the region near the desired equilibrium point, and thus controllers that are more effective at driving the system to this point.

3. RESULTS

3.1 Lorenz system with control input

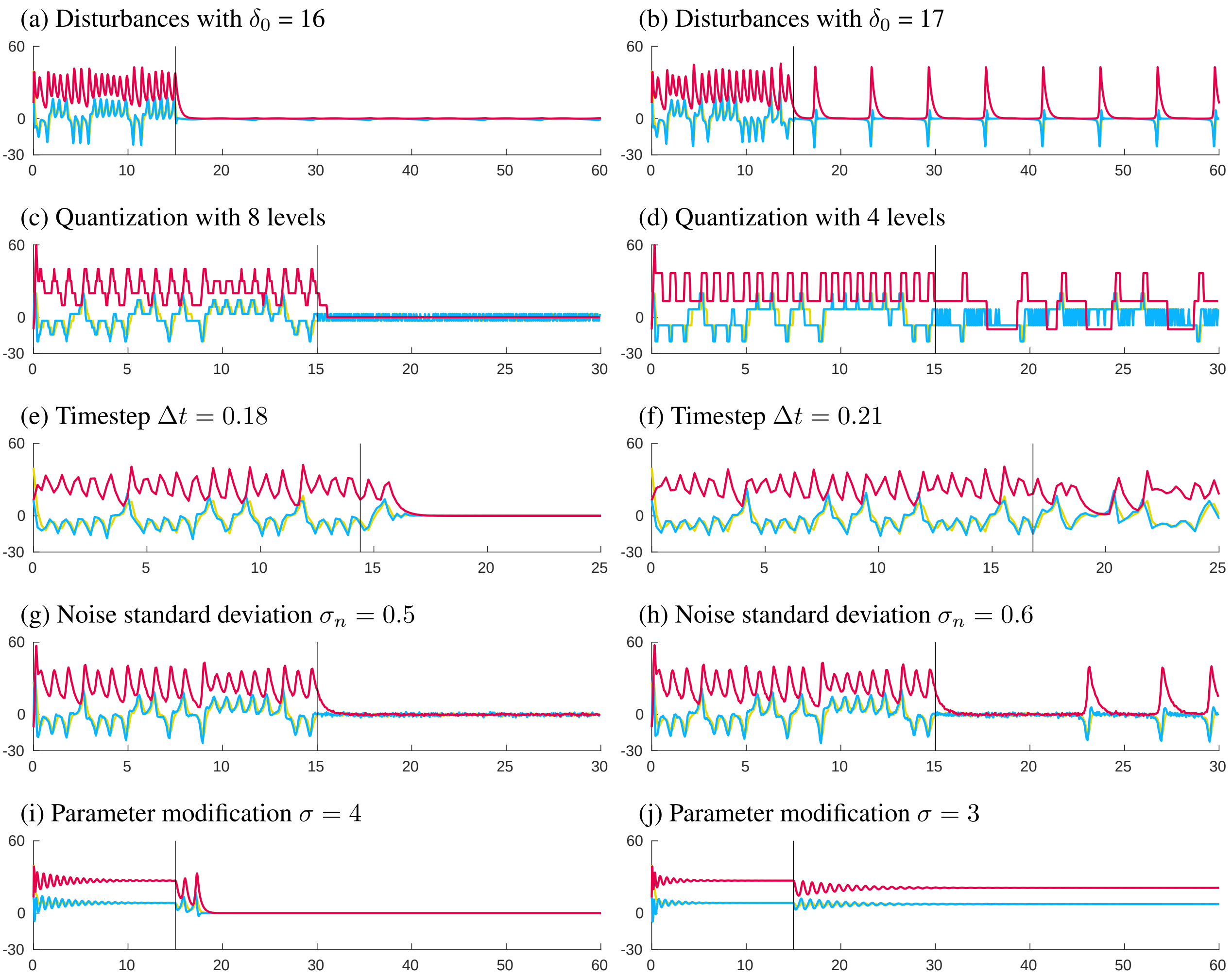
The Lorenz system is chosen for testing the NNC robustness as it is a simple nonlinear system with only three states, allowing for inexpensive simulations that can be used to assess the robustness of the proposed control approach in several ways. Its dynamics are described by the following set of nonlinear ordinary differential equations

$$\dot{x} = \sigma(y-x)\,, \tag{5}$$

$$\dot{y} = \rho x - y - xz + u\,, \tag{6}$$

$$\dot{z} = -\beta z + xy\,, \tag{7}$$

where $\sigma = 10$, $\beta = 8/3$ and ρ = 28 are chosen, making the uncontrolled system ($u(t) = 0$) behave as a chaotic attractor. A single control input u is added to the classic Lorenz system in the equation for y. The training process is conducted


(a) Disturbances with δ_0 = 16

(b) Disturbances with δ_0 = 17

(c) Quantization with 8 levels

(d) Quantization with 4 levels

(e) Timestep $\Delta t = 0.18$

(f) Timestep $\Delta t = 0.21$

(g) Noise standard deviation $\sigma_n = 0.5$

(h) Noise standard deviation $\sigma_n = 0.6$

(i) Parameter modification $\sigma = 4$

(j) Parameter modification $\sigma = 3$

Figure 2. Controlled Lorenz system subject to different undesirable phenomena. The vertical black line depicts the moment when the control is turned on. The colored curves represent the measured states seen by the controller.

with the proposed parameters by sampling data without noise, disturbances, or quantization. These phenomena shall be introduced a posteriori in order to check the robustness of the trained control system.

We present an assessment of the robustness of the closed NNC loop. We choose the Lorenz system due to the possibility of running many trial and error tests. We mainly present the threshold conditions beyond which the controller saturates or fails to keep the Lorenz system in the regions near equilibrium. First, in figures 2 (a) and (b), we show the results of modifying the control input from $u = K(x, y, z)$ to $u = K(x, y, z) + \delta_0 \sin(\omega_d k)$. With a low frequency ω_d and $\delta_0 = 17$ – which corresponds to 38% of the maximum allowed control signal value (45) – the states move away from the region near $[0, 0, 0]$. This happens when u saturates and, therefore, increasing the saturation levels should enhance the ability to keep the states closer to equilibrium. Furthermore, allowing the controller to output higher efforts also tends to increase the robustness to disturbances.

The effects of quantization in the digitization of the signals are also tested. In figures 2 (c) and (d), it is shown that the states move away from the equilibrium when a resolution of 4 levels (equivalent to 2-bit quantization) is used. With 8 levels (3-bit quantization), the states are kept in the region close to equilibrium. A low resolution in quantization worsens the performance of control and this needs to be taken into account in real world applications.

In this work, a standard fourth-order Runge-Kutta scheme is used to simulate the Lorenz system. We verified that the maximum timestep for the simulation to stay stable is $\Delta t = 0.1$. In most of the simulations shown here, we use $\Delta t = 0.05$. Exclusively for the cases shown in figures 2 (e) and (f) we use a simulation time step of 0.03 and sample the states every 6 steps ($\Delta t = 0.18$) or every 7 steps ($\Delta t = 0.21$). When Δt is too large, the effects of aliasing make it impossible for the control loop to keep working properly. This is a limitation of digital control, and a minimum sample rate is required independently of the control technique itself.

Another test that is presented encompasses the addition of noise to the measured signals. Independent Gaussian noise is applied to each of the three states, with a mean value of 0 and standard deviations of $\sigma_n = 0.5$ and $\sigma_n = 0.6$, as shown in figures 2 (g) and (h). In each case, the same σ_n is used for x, y and z. By using data from the controlled system with $\sigma_n = 0.5$ – similar to that shown in figure 2 (g) – we estimated a signal-to-noise ratio of around -2.5dB for x, 5dB

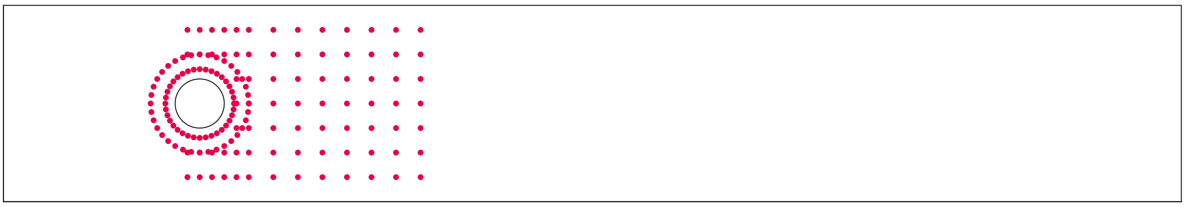


Figure 3. Initial candidate sensor positions for the confined cylinder flow. The box represents the spatial domain used in the simulation.

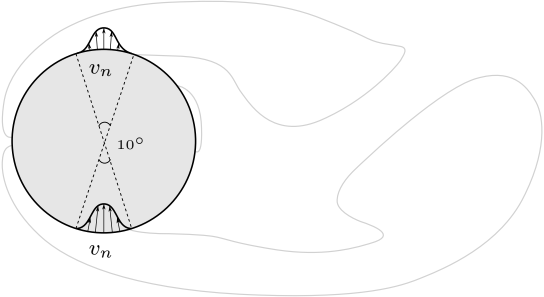


Figure 4. Actuation scheme applied in the cylinder flow. Blowing/suction jets in opposition are modulated by a single control input.

for y and -23dB for z. Wi The states move away from near the equilibrium point due to the high amplification of noise produced by the controller. This makes the control input saturate, rendering the loop unable to hold the states.

Finally, we check the capability of the NNC to control the plant with modifications to the parameter σ. We keep the β and ρ values from the equations the same as the originals and test $\sigma = 4$ and $\sigma = 3$, as shown in figures 2 (i) and (j). With $\sigma = 3$, the controller loses its ability to maintain the system states close to $[0, 0, 0]$. In this scenario, the controller is permanently locked at saturation levels. With the new values of σ, a stable equilibrium is observed. Before the control is turned on, the states converge to that point and, for the $\sigma = 3$ case, the controller is able to drive the states towards the desired point at the origin of the state space.

3.2 Confined cylinder flow setup

We study a 2D cylinder flow at Reynolds $\mathrm{Re} = 150$ based on the cylinder diameter D and maximum velocity U of the inflow profile. For this task, a Nek5000 setup (Li and Zhang, 2022) is employed, where the initial configuration of sensors is equivalent to the one presented by Rabault *et al.* (2019), depicted in figure 3. The simulation time step is set to 2.5e-3 based on the cylinder diameter and the maximum velocity in the channel. The control time step Δt is 40 times that value. Here, 153 locations are chosen totaling 306 measurements for u and v velocity components. No-slip wall boundary conditions are applied at the top and bottom limits of the domain. The actuation scheme is presented in figure 4 and a single control input modulates minijets in opposition, so that there is zero net mass flux into the system at all times. Furthermore, the trained controller is employed to similar flows at different Reynolds numbers for robustness assessment.

From the initial set of probes proposed, we end up with the selected sensors (by the sparsity layer) shown in figure 5. The lift coefficient of the cylinder flow is presented in figure 6 for visualization of the stabilization process. The blue curves represent the perturbed phases, where data is sampled for training. For the first iteration (first set of blue, yellow and red curves), only an open-loop control signal is imposed, while for iterations 2 and 3, the system is perturbed by the open-loop signal while being controlled in closed loop. The yellow part represents and uncontrolled step, where the flow gets back to its natural limit cycle. In red, the system controlled in closed loop is shown. The entire plot shows data gathered from the full order flow solver with the NNC.

At iteration 1, the NNSM is trained with open-loop data, an equilibrium point is estimated and the NNC is trained. Although the NNC is unable to stabilize the actual flow at this very first trial, a new limit cycle with smaller amplitude of oscillations takes place. As we perturb the controlled system in iteration 2, the new perturbed data obtained is often closer to equilibrium, (which corresponds to a lift coefficient of 0). The closed-loop step for iteration 2 presents a drastic reduction in oscillations, although a slow growth rate with high frequency is seen. By iteration 3, the closed-loop step shows complete stabilization. For this case, a comparison between the uncontrolled and controlled flows is presented in figures 7 (a) and (b).

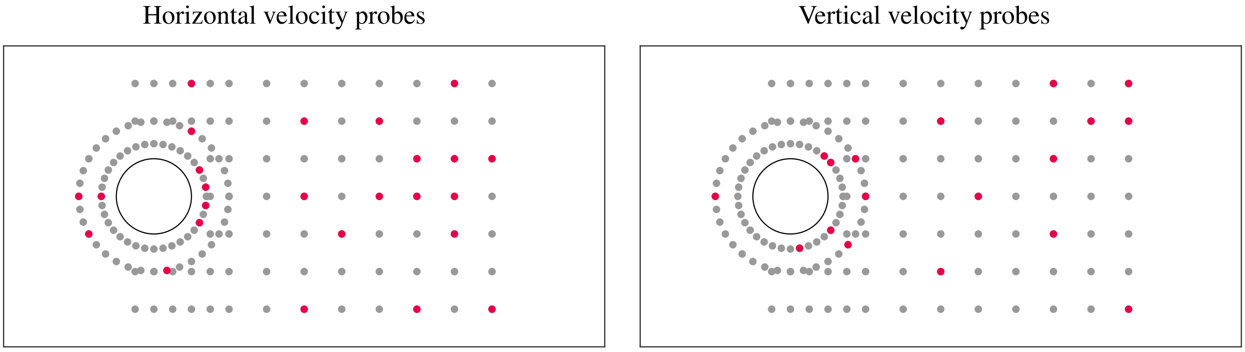


Figure 5. Demonstration of the application of the sparsity layer for the confined cylinder flow. The total number of measurements is reduced from 306 to 47 (25 and 18 probes for the horizontal and vertical velocity components, respectively). The gray dots are deactivated by the present L1 regularization.

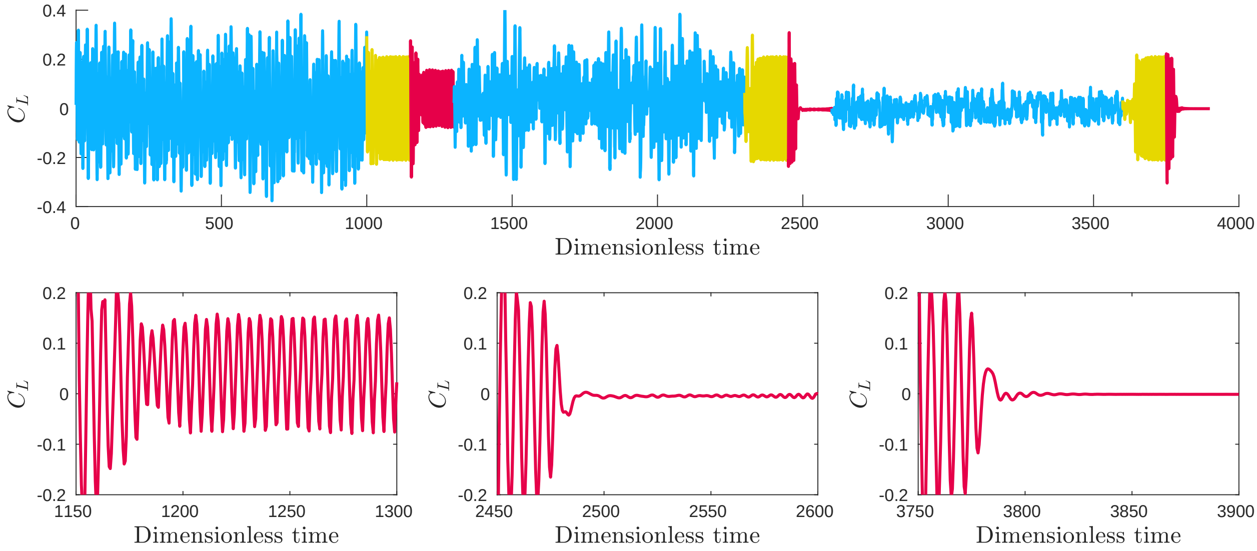


Figure 6. Evolution of lift coefficient for 3 iterations of the perturbed system (blue), uncontrolled (yellow), and controlled (red) stages for the confined cylinder flow. The lower subplots zoom in on the three regions where the controller is activated.

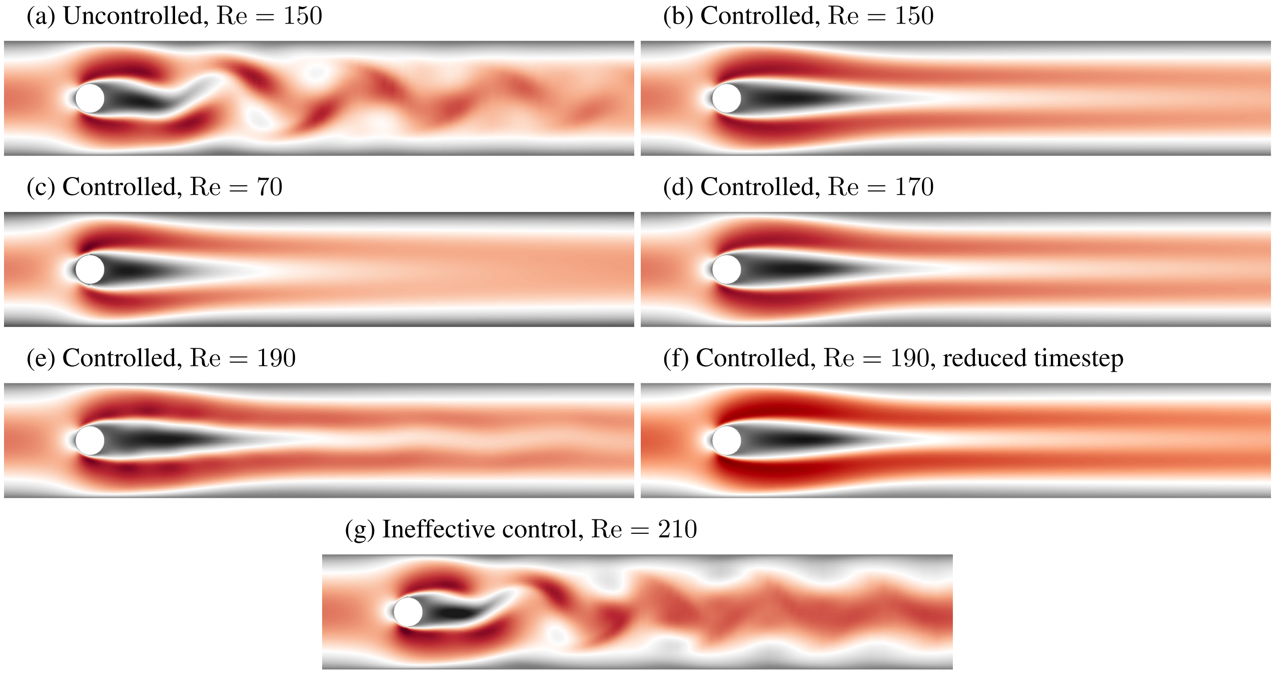

(a) Uncontrolled, $\mathrm{Re} = 150$
(b) Controlled, $\mathrm{Re} = 150$
(c) Controlled, $\mathrm{Re} = 70$
(d) Controlled, $\mathrm{Re} = 170$
(e) Controlled, $\mathrm{Re} = 190$
(f) Controlled, $\mathrm{Re} = 190$, reduced timestep
(g) Ineffective control, $\mathrm{Re} = 210$

Figure 7. Snapshot depicting contours of u-velocity for flow at different Reynolds numbers.

The trained stabilizing controller is tested for different Reynolds numbers without being retrained. For cases where $\mathrm{Re} < 150$, we tested the controller with $Re = 130$, 110, 90 and 70. For all of them, the controller was able to successfully stabilize the flow. Figure 7 (c) shows such a result for $Re = 70$. From the contour plots, it is possible to notice that some level of asymmetry appears, which is expected since the equilibrium flow state is different from the one found for $\mathrm{Re} = 150$. This may lead to a forced equilibrium point with $u \neq 0$, resulting in an asymmetric configuration. For some Reynolds numbers higher than 150, the controller is still able to stabilize the flow, as shown in figure 7 (d). At $\mathrm{Re} = 190$ (figure 7 (e)), the main vortex shedding mode is successfully controlled. However, a new unstable mode appears. We verified that the issue is possibly related to aliasing and, in fact, by reducing the control time step by half, the trained controller was able to completely stabilize the flow (figure 7 (f)). At $\mathrm{Re} = 210$ (figure 7 (g)), the controller was not able to stabilize the flow.

4. CONCLUSIONS AND FUTURE WORK

An empirical study to evaluate robustness of NNCs is conducted for the low-order Lorenz system and a confined cylinder flow. The proposed approaches for NNSM training, equilibrium computation and NNC training across an iterative process are successful in stabilizing a confined cylinder flow with a reduction in the number of sensors needed when compared to the initial candidate set often found in literature. Along three iterations of training, the oscillation amplitudes probed at the vortex street are shown to reduce gradually until complete stabilization is achieved. Along with the stabilization of the proposed flow, the methodology also outputs data close to equilibrium, i.e., in regions where linear approximations could be better fit. The study of the robustness provided by the trained NNCs indicated that there is a significant margin for working with phenomena such as measurement noise, plant variation, digitization error and disturbances. We find that the trained controller is able to stabilize other cylinder flows at lower Reynolds numbers and, in some cases, at higher Reynolds numbers.

Some modifications to the proposed methodology can be studied in future work to account for better functioning of the control loops subject to such real world circumstances. For example, the ability to linearize the NNSMs could be leveraged for implemented extended Kalman filters with the goal of estimating states. This could potentially increase the control effectiveness when the loop is subject to measurement noise. For better adaptability, an alternate structure could be tried for the NNSM, including an input to the models consisting of the parameter system parameters (e.g., Reynolds and Mach numbers). By doing so, training of the NNC could be conducted by taking into account different locations in the parameter space.

5. ACKNOWLEDGEMENTS

The authors acknowledge the financial support received from Fundação de Amparo à Pesquisa do Estado de São Paulo, FAPESP, under Grants No. 2013/08293-7, 2019/19179-7, 2021/06448-0, and 2022/00469-8, and from Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, under Grant No. 308017/2021-8. Financial support from

Universidade Estadual de Campinas, through the FAEPEX Grant No. 2375/24, is also acknowledged. The computational resources used in this work were provided by CENAPAD-SP (Project 551).

6. REFERENCES

Åkervik, E., Brandt, L., Henningson, D.S., Hœpffner, J., Marxen, O. and Schlatter, P., 2006. "Steady solutions of the navier-stokes equations by selective frequency damping". *Physics of fluids*, Vol. 18, No. 6.

Bae, H.J. and Koumoutsakos, P., 2022. "Scientific multi-agent reinforcement learning for wall-models of turbulent flows". *Nature Communications*, Vol. 13, No. 1, p. 1443.

Bieker, K., Peitz, S., Brunton, S.L., Kutz, J.N. and Dellnitz, M., 2020. "Deep model predictive flow control with limited sensor data and online learning". *Theoretical and Computational Fluid Dynamics*, pp. 1–15.

Brunton, S.L. and Noack, B.R., 2015. "Closed-loop turbulence control: progress and challenges". *Applied Mechanics Reviews*, Vol. 67, No. 5, p. 050801.

Brunton, S.L., Noack, B.R. and Koumoutsakos, P., 2020. "Machine learning for fluid mechanics". *Annual Review of Fluid Mechanics*, Vol. 52, pp. 477–508.

Chatzimanolakis, M., Weber, P. and Koumoutsakos, P., 2024. "Learning in two dimensions and controlling in three: Generalizable drag reduction strategies for flows past circular cylinders through deep reinforcement learning". *Physical Review Fluids*, Vol. 9, No. 4, p. 043902.

Déda, T., Wolf, W. and Dawson, S., 2024. "Neural networks in feedback for flow analysis, sensor placement and control". *Physical Review Fluids*, Vol. 9, No. 6, p. 063904.

Déda, T., Wolf, W.R. and Dawson, S.T.M., 2023. "Backpropagation of neural network dynamical models applied to flow control". *Theoretical and Computational Fluid Dynamics*, pp. 1–25.

Garnier, P., Viquerat, J., Rabault, J., Larcher, A., Kuhnle, A. and Hachem, E., 2021. "A review on deep reinforcement learning for fluid mechanics". *Computers & Fluids*, Vol. 225, p. 104973.

Li, J. and Zhang, M., 2022. "Reinforcement-learning-based control of confined cylinder wakes with stability analyses". *Journal of Fluid Mechanics*, Vol. 932, p. A44.

Morton, J., Jameson, A., Kochenderfer, M.J. and Witherden, F., 2018. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., Vol. 31.

Rabault, J., Kuchta, M., Jensen, A., Réglade, U. and Cerardi, N., 2019. "Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control". *Journal of fluid mechanics*, Vol. 865, pp. 281–302.

Ren, F., Rabault, J. and Tang, H., 2021. "Applying deep reinforcement learning to active flow control in weakly turbulent conditions". *Physics of Fluids*, Vol. 33, No. 3, p. 037121.

Suárez, P., Alcantara-Avila, F., Miró, A., Rabault, J., Font, B., Lehmkuhl, O. and Vinuesa, R., 2024. "Active flow control for drag reduction through multi-agent reinforcement learning on a turbulent cylinder at $re\_d = 3900$". *arXiv preprint arXiv:2405.17655*.

Tokarev, M., Palkin, E. and Mullyadzhanov, R., 2020. "Deep reinforcement learning control of cylinder flow using rotary oscillations at low Reynolds number". *Energies*, Vol. 13, No. 22, p. 5920.

Vinuesa, R., 2024. "Perspectives on predicting and controlling turbulent flows through deep learning". *Physics of Fluids*, Vol. 36, No. 3.

Xu, H., Zhang, W., Deng, J. and Rabault, J., 2020. "Active flow control with rotating cylinders by an artificial neural network trained by deep reinforcement learning". *Journal of Hydrodynamics*, Vol. 32, No. 2, pp. 254–258.

Zhou, K. and Doyle, J.C., 1998. *Essentials of robust control*, Vol. 104. Prentice hall Upper Saddle River, NJ.

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.