Citation (APA)
Mohammadikalakoo, B., Kotsonis, M., & Doan, N. A. K. (2025). Real-time single-step deep reinforcement learning framework for control of Tollmien-Schlichting waves. *Physical Review Fluids*, *10*(12). https://doi.org/10.1103/pshh-6txb

Real-time single-step deep reinforcement learning framework for control of Tollmien-Schlichting waves

B. Mohammadikalakoo,* M. Kotsonis, and N. A. K. Doan
Department of Flow Physics and Technology, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft 2629HS, The Netherlands

(Received 23 June 2025; accepted 4 November 2025; published 8 December 2025)

A real-time adaptive control framework based on the single-step deep reinforcement learning (SDRL) algorithm is proposed and implemented in numerical simulations for attenuating Tollmien-Schlichting (TS) wave instabilities in a two-dimensional incompressible boundary layer over a flat plate. The perturbation formulation of the Navier-Stokes equations is solved in the time domain using a finite-volume discretization scheme. The controller utilizes volume-distributed body forces as proxies for dielectric barrier discharge plasma actuators, with instantaneous wall-pressure feedback in a closed-loop configuration to dynamically adjust the amplitude of the controlling body force. By learning an opposition control strategy—generating TS waves with the same amplitude but opposite phase to the incoming TS instabilities—the SDRL-based controller achieves effective suppression of both single- and multifrequency TS waves. The performance of the SDRL-based controller is evaluated against a classical model-free adaptive approach, the filtered-x least mean square– (FXLMS) based controller, focusing on effectiveness in terms of TS wave suppression and convergence speed. The SDRL-based controller achieves up to 94.81% suppression of TS waves, demonstrating superior suppression performance and faster convergence than the FXLMS-based controller in most control cases. Additionally, the robustness of the controllers against measurement and actuation noises is examined. These results establish the SDRL framework as a low-cost, model-free control approach for robust real-time control of convective flow instabilities in the presence of noise sources. Relying on compact neural networks and minimal sensor feedback, the methodology enables a practical and experimentally feasible artificial intelligence-based controller for convective instabilities.

DOI: 10.1103/pshh-6txb

I. INTRODUCTION

The transition from laminar to turbulent flow in two-dimensional subsonic boundary layers, particularly under low external disturbance conditions, is mainly driven by the growth and breakdown of Tollmien-Schlichting (TS) waves [1]. To extend the laminar flow and enhance aerodynamic efficiency, numerous researchers have focused on suppressing or mitigating TS wave growth. Such efforts aim to delay transition and reduce skin friction drag through passive and active flow control strategies [2–7].

Active flow control strategies, essential for suppressing TS waves in their early stages of linear growth, revolve around the principle of wave superposition. The conventional method involves actuators installed at or near the surface that introduce a counterwave with the same amplitude but opposite phase to the incoming instabilities. This technique was first validated by experiments using

*Contact author: b.mohammadikalakoo@tudelft.nl

vibrating ribbons in controlled environments to generate TS waves [8,9]. It involves fine-tuning the actuator's amplitude and phase to effectively dampen the incoming disturbances. Further refinement in this area has been achieved through direct numerical simulations of Blasius boundary layer flows, where a sensor-actuator setup applies a spectral control strategy to convert sensor data into actuating signals, adjusting amplification and phase to suppress the TS waves effectively [10]. These studies have shown the potential to eliminate convective instabilities of small and large amplitudes through wave cancellation even without feedback. However, achieving precise wave cancellation, especially of multifrequency instabilities, requires feedback to fine-tune the control amplitude and phase.

Although these techniques are effective in laboratory settings, the natural occurrence of TS waves in flight conditions presents additional challenges. The challenge in predicting instantaneous wave characteristics, such as amplitude, frequency, and phase, necessitates advanced control schemes. Closed-loop systems, which incorporate sensors, allow adaptive control strategies that dynamically adjust to changing conditions, such as variations in Reynolds number and angle of attack. These adaptive systems can either rely on a predefined model of flow dynamics (model-based strategies) [11,12] or operate without one (model-free strategies) [13–17].

Model-based control strategies are highly dependent on an accurate understanding of flow dynamics, often using linear system theory to anticipate and counteract wave behavior. These strategies have been widely applied to control convective instabilities within two-dimensional boundary layer flows [11,12,18,19]. However, their effectiveness can diminish significantly if the flow model does not accurately reflect the real-time flow conditions or if global flow parameters change drastically.

In contrast, model-free control strategies use system identification techniques to adapt to the flow conditions without a predetermined model. Adaptive control as a widely utilized model-free control strategy has proven particularly effective, with various experimental studies on flat plates, axisymmetric bodies [20–22], and unswept wings [23–25] demonstrating successful cancellation of naturally occurring TS instabilities. Advances in adaptive filter-based control, such as those employing the filtered-x least mean square (FXLMS) algorithm, have reported amplitude reductions in both artificially induced and naturally occurring TS waves by over 94% [13]. Studies on filter-based adaptive control for convective instabilities have advanced through various approaches, including numerical simulations [4], in-flight experiments [16], and experimental wind tunnel tests [15].

The control of TS waves under realistic aerodynamic conditions remains a complex challenge due to the inherent spectral unpredictability of the naturally occurring TS waves [12]. Moreover, both model-based and model-free approaches are fundamentally constrained by their reliance on linear wave superposition. As a result, control is only feasible within a limited spatiotemporal window—where boundary layer instabilities are strong enough to be detected by sensors but weak enough to admit linear dynamics. Expanding this control window and addressing the inherent sensitivity and cost limitations of sensors requires exploring flow control strategies capable of effectively suppressing TS waves even in their highly nonlinear development stages. At this point, artificial intelligence presents a potentially promising solution for overcoming these limitations.

Artificial intelligence has revolutionized numerous fields, including fluid dynamics, where it plays a crucial role in solving complex problems such as turbulence modeling, aerodynamic optimization, and flow control. Artificial intellgence's capability to solve high-dimensional and nonlinear problems makes it particularly effective in developing advanced control laws [26]. Active flow control applications often involve complex decision-making, where deep reinforcement learning (DRL) has emerged as a powerful approach [27]. By combining reinforcement learning with deep neural networks, DRL enables an agent to learn optimal actions through trial and error to maximize rewards in a given environment. Many recent studies have demonstrated the potential of DRL for effective flow control in both laminar [6,7,27–33] and turbulent [34–36] flow regimes. Nevertheless, employing full DRL frameworks is not always feasible, particularly for fluid dynamics problems. Training deep networks in DRL settings often requires large datasets, substantial computational resources, and extensive interaction with costly numerical solvers, which can be prohibitive. Moreover, the complexity of designing and tuning deep neural networks further adds to the practical challenges. To mitigate these challenges, some studies have proposed more efficient strategies—for instance, developing reinforcement learning–based wall models that bypass the need for empirical calibration [37] or coupling DRL with low-dimensional neural ODE-based surrogate models to accelerate training [38]. While these approaches significantly reduce computational cost, their applicability is still limited to specific flow configurations and requires careful validation before general deployment.

To address these issues, Ref. [39] proposed single-step deep reinforcement learning (SDRL) as a simplified alternative, where the agent interacts with the environment only once per episode. This technique is particularly advantageous in scenarios where the policy to be learned is state independent, as it allows for the use of smaller neural networks and reduces the complexity of the learning task. Such characteristics make SDRL especially suitable for flow control applications, where interactions with the environment (e.g., numerical solvers) are computationally expensive. The initial applications of SDRL primarily focused on shape and parameter optimization [32,39–41], with extensions to open-loop control problems as demonstrated in Ref. [42].

Recent progress in optimization-based flow control has highlighted the potential of model-free learning approaches for the suppression of TS waves. In particular, simulations in the frequency domain showed the possibility of effective implementation of the SDRL algorithm as the controller in a closed-loop system. The algorithm controls the phase and amplitude of unsteady suction and blowing jets at the wall to suppress TS waves on a flat plate, following a similar approach to Ref. [6], achieving up to 99.96% (40 dB) TS wave amplitude suppression. The strategy identified by the SDRL is opposition control, generating a counteracting wave that mirrors the amplitude of TS waves with an opposed phase. Comparative analysis with the particle swarm optimization (PSO) technique [7] showed that SDRL achieves faster convergence—up to three times quicker in single-frequency linear control scenarios. Overall, the SDRL-based controller outperformed PSO in identifying more effective solutions, highlighting its potential to efficiently suppress TS waves.

Considering the success of this algorithm in the frequency domain, the present study introduces the use of SDRL for controlling TS waves in the time domain, paving the way for its deployment in real-time applications. Extending the SDRL application to real-time controllers constitutes a pivotal advancement toward addressing the fundamental challenge of causality-constrained flow control, enabling the system to respond to the inherently transient nature of convective instabilities in realistic and noisy environments. To demonstrate the feasibility of this technique and move toward more practical implementations, the concept of dielectric barrier discharge (DBD) plasma actuators is used in this study. These actuators, noted for their ability to control unsteady flow instabilities, consist of two electrodes divided by a dielectric layer. When an ac high voltage is applied to the electrodes, a strong electric field ionizes the nearby air. This ionized gas, or plasma, transfers momentum to the flow through collisional processes, thereby facilitating flow control [43]. Cited for their simplicity and robustness, DBD actuators also offer the benefits of low power consumption and cost efficiency. A significant advantage of these actuators is their rapid response time, typically within several milliseconds, attributed to their lack of mechanical parts [15]. Over the past few decades, these actuators have been explored for TS wave cancellation through both experimental tests [2,13,15,17,44,45] and numerical simulations [4,46–49], demonstrating their potential for convective instabilities control.

In the current work, a real-time, model-free SDRL-based controller is introduced in a simulated environment towards suppressing single- and multifrequency TS waves. Its performance in terms of controller effectiveness and speed is compared with FXLMS as a reference classical adaptive controller. A closed-loop control framework is used, composed of a wall-mounted DBD plasma actuator, along with the controller algorithm for selecting the applied voltage to the actuator based on sensor feedback that relies on measuring the wall pressure perturbations upstream (reference sensor) and downstream (error sensor) of the actuator. This study demonstrates the feasibility of using the SDRL algorithm inside a closed-loop control system for effective real-time control of TS waves while highlighting the potential advantages and disadvantages compared to a classical control technique. Importantly, it investigates the robustness against uncorrelated and random noises in the

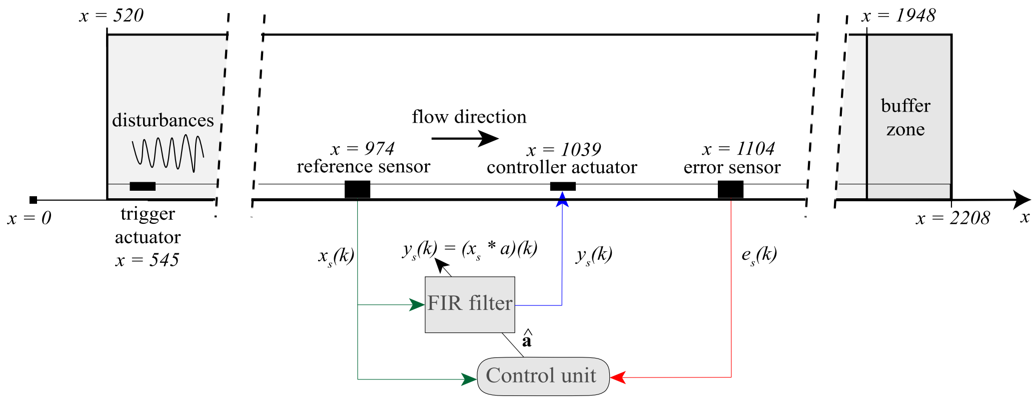

FIG. 1. Schematic of the filter-based controller; $x_s(k)$, $y_s(k)$, and $e_s(k)$ refer to reference, actuator, and error signals, respectively. The computational domain starts at $x = 520$ and ends at $x = 2208$ with a buffer region at the end described in Sec. II B. Details of the control unit and its parameters are presented in Sec. III and Fig. 3. The distances and proportions are not drawn to scale. All the sensors and actuators are located over the wall at $y = 0$.

measurement and actuation signals, a key advantage not achievable with the previous frequency-domain study.

The paper is structured as follows. Section II details the methodology, including the flow problem, numerical setup, test cases, and control algorithm. Section III describes the control framework and the process of updating control parameters using SDRL and FXLMS algorithms. Section IV evaluates and discusses the controller's performance across various single- and multifrequency control cases while comparing the performance of two controllers in terms of effectiveness, convergence speed, and robustness to measurement and actuation noises. Finally, Sec. V highlights the study's main findings and implications.

II. METHODOLOGY

This section presents the flow problem, numerical method, and control technique utilized for simulating and controlling TS waves in an incompressible two-dimensional (2D) laminar boundary layer. The 2D Navier-Stokes in perturbation formulation are utilized, similarly to the reference study [4].

A. Flow problem

For the entirety of this work, the development and control of TS waves in a flat plate (i.e., zero pressure gradient) boundary layer flow at $U_\infty = 30$ m/s freestream velocity is considered. Figure 1 displays a conceptual representation of the flow control system while the computational domain is presented in Fig. 2 alongside the stability diagram based on the linear stability theorem (LST). The spatial dimensions are globally nondimensionalized using the displacement thickness defined at the inflow of the simulation domain ($x = 0.4$ m), denoted as $\delta_0^* = 0.77$ mm.

The freestream velocity is selected based on the methodology in Ref. [50], which provides experimentally derived body force distributions. This choice ensures consistency with the actuator data from that study. Given the actuator's strength and the effectiveness of the FXLMS-based controller at that velocity, using the same streamwise velocity provides a reliable basis for comparison. The Reynolds number for all test cases based on the inflow displacement thickness ($\mathrm{Re}_{\delta_0^*} = \frac{U_\infty \delta_0^*}{\nu}$) is 1538, with kinematic viscosity $\nu = 1.5 \times 10^{-5} \frac{\mathrm{m}^2}{\mathrm{s}}$.

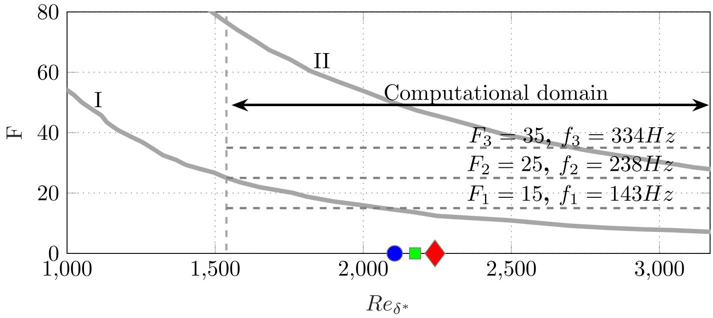

FIG. 2. Stability diagram based on linear stability theory analysis. The solid gray line is the neutral curve. The area between branches I and II marks the unstable window. The location of the actuator is presented with a green square marker, while reference and error sensors are indicated with blue circle and red diamond markers, respectively. F is a nondimensional frequency based on Reynolds at the inflow, which is defined in Eq. (4).

Two pressure sensors are located upstream ($x = 974$) and downstream ($x = 1104$) of the plasma actuator to monitor the pressure fluctuations at the wall. The effect of the plasma actuator is modeled as a body force distribution centered at ($x = 1039$). This distribution is derived from experimental measurements of a reference plasma actuator [50] and controlled by an applied voltage. The implementation of experimentally derived body force distributions is detailed in Sec. II B. The applied voltage is defined by the control unit (SDRL or FXLMS algorithms) as explained in the control framework, Sec. III, with the main goal of reducing the amplitude of the error sensor signal, consequentially signifying TS wave suppression.

Spatial LST, based on the incompressible Orr-Sommerfeld equation [51], is applied to the baseline steady-state solutions to identify frequencies of unstable TS waves presented as stability diagram in Fig. 2. Based on this stability diagram, the actuator is strategically placed at a location where triggered TS wave modes are unstable yet in their early stage of growing, ensuring that their amplitudes remain low enough so that the actuator voltage does not exceed the available operational range. The reference sensor is positioned sufficiently upstream to minimize the actuator's influence on the reference signal while maintaining close enough proximity to avoid significant delays in information transfer between the sensor and the actuator. Similarly, the error sensor is placed downstream of the actuator to ensure any spatial transients in the controlled instabilities have decayed by the time they reach the error monitoring location.

B. Numerical setup and test cases

To fully resolve the spatial and temporal scales governing the propagation and amplification of TS waves, the 2D Navier-Stokes equations are considered in the perturbation formulation, where the steady base flow (laminar boundary layer) remains fixed and the Navier-Stokes equations are solved exclusively for TS unsteady perturbations. The perturbation formulation method allows the base flow to be computed once, after which the perturbation equations can be solved separately with smaller time steps and finer spatial grids. This greatly increases computational efficiency, which is crucial given that the primary aim of this study is to provide a proof of concept for the control of convective instabilities with the SDRL-based controller. The 2D incompressible Navier-Stokes equations, including an applied body force ($\hat{\mathbf{F}}$), are given by Eq. (1):

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} - \nu \nabla^2 \mathbf{U} = -\frac{\nabla P}{\rho} + \frac{\hat{\mathbf{F}}}{\rho}. \tag{1}$$

The flow is decomposed as the sum of a steady base flow ($\mathbf{U}_0, P_0$) and a fluctuating component (u, p). The base flow, which remains constant over time, can be obtained either from analytical solutions of the Falkner-Skan equations or through numerical solutions of the steady Navier-Stokes equations. In this study, the base flow is determined using a numerical solution of the steady Navier-Stokes equations. This approach leads to the final formulation for the perturbations, as expressed in Eq. (2),

$$\frac{\partial u}{\partial t} + (\mathbf{U}_0 \cdot \nabla)u + (u \cdot \nabla)\mathbf{U}_0 + (u \cdot \nabla)u = -\frac{\nabla p}{\rho} + \nu \nabla^2 u + \frac{\hat{\mathbf{F}}}{\rho}, \tag{2}$$

$\hat{\mathbf{F}}$ represents both the mean and fluctuating components of the body force, incorporating the plasma actuator's effect on the flow. In this study, the "^" symbol is used to distinguish the body force from the frequency modes of TS waves.

The governing equations are solved using a second-order finite-volume discretization in OpenFOAM with a second-order backward implicit scheme for temporal advancement. A Dirichlet condition enforcing zero perturbation velocity was applied at the inflow, plate, and top boundaries, while a Neumann condition ensuring a zero perturbation velocity gradient was imposed at the outflow. For perturbation pressure, zero-gradient conditions were enforced on all boundaries, except for the top boundary, where a fixed zero-pressure condition was applied.

All cases employ structured rectangular meshes, consisting of approximately 0.1 million tetrahedral cells. The mesh spans $x = 520$ to 2208 and $y = 0$ to 32, with linear clustering near the wall throughout the domain. A locally refined mesh is applied around the body force distribution to enhance resolution in that region. For the highest frequency case ($F = 35$), the grid density corresponds to approximately 20 cells per TS wavelength. To prevent reflections from outflowing disturbances, a buffer region with gradually increasing cell size is introduced at the downstream boundary ($x = 1948$). This buffer extends to the end of the computational domain ($x = 2208$) with a stretch factor of 10 in the streamwise direction.

To investigate the boundary layer stability problem, the disturbance must be introduced in the numerical domain. In this study, an approach similar to that in Ref. [52] is used in which an unsteady horizontal body force $f_g(x, y, t)$ is used in the form of Eq. (3) for introducing the disturbance,

$$f_g(x, y, t) = \sum_{i=1}^{n} (W_i \cdot \sin(2\pi \omega_i t)) \cdot \sin\left(\frac{\pi(x - x_1)}{x_2 - x_1}\right) \cdot \sin\left(\frac{\pi(y - y_1)}{y_2 - y_1}\right), \tag{3}$$

where $x_1 < x < x_2, y_1 < y < y_2$.

The differences $(x_2 - x_1)$ and $(y_2 - y_1)$ determine the lengths of the force distribution in the x and y directions. Following the available literature [4,52], the streamwise length is approximately half that of the target TS wavelength, while the height is about a quarter of the TS wavelength. The center of body force is consider at the proximity of the computational domain inflow ($x = 545$). The temporal variation of the fluctuating force is governed by the sum of n sinusoidal modes, each characterized by an amplitude (W_i) and frequency (ω_i).

To effectively simulate the effect of the controlling plasma actuator, an experimentally derived body force distribution from plasma actuators [50] is considered with a control parameter of applied voltage. This defines not only the amplitude of the force distribution but also its spatial extent and shape. The reference experimental data cover peak-to-peak voltages from 8 to 16 kV in 2-kV steps, offering five distinct force distributions. These are integrated into the solver using a 2D Bernstein surface interpolation, with coefficients stored in the controller environment for real-time force calculations across the specified voltage range, following the methodology outlined in Ref. [4]. To ensure that the body force interpolation remains within a meaningful range, strict upper and lower bounds are set to block any drift outside the original experimental voltage range. Therefore, body forces are coerced to zero below 8 kV and held constant above 16 kV. The selected voltage range of 8 to 16 kV provides a suitable balance between actuation authority and avoiding undesired secondary effects. This range influences both the learning dynamics and the control effectiveness. A larger voltage range increases the exploration space of the controller algorithm, which can slow convergence, whereas a smaller range limits the achievable actuation authority. In addition, excessively low voltages do not generate sufficient force to counteract incoming instabilities, while excessively high voltages can induce secondary TS waves that deteriorate performance.

The above modeling constructs a correlation between voltage and force, capturing the magnitude and spatial distribution of the body force with less than 10% of mean space-averaged error between experimental and numerical results. Interested readers are referred to Ref. [53] for more details.

As test cases, one single-frequency (case A) and two multifrequency control cases (B1 and B2) are considered in this study. A nondimensional frequency number F is set based on the Reynolds number at the inflow:

$$F = \frac{\omega_0}{\mathrm{Re}_0^*} \times 10^6, \quad \omega_0 = \frac{2\pi\delta^* f}{U_\infty}, \tag{4}$$

where ω_0 in Eq. (4) refers to the local nondimensional frequency of the mode while F is the nondimensional frequency based on the inflow Reynolds number; f is the physical dimensional frequency of the mode.

The single-frequency control case A corresponds to the nondimensional frequency mode $F = 25$, which has a dimensional equivalent of $f = 238$ Hz based on Eq. (4). The multifrequency control case B1 consists of a combination of three modes with nondimensional frequencies $F_1 = 15$, $F_2 = 25$, and $F_3 = 35$ (corresponding to $f_1 = 143$ Hz, $f_2 = 238$ Hz, and $f_3 = 334$ Hz), each with equal initial amplitudes. These excitation frequencies are selected such that the incoming flow is initialized with the most stable modes, which grow downstream and become unstable upon reaching the actuator and sensors. The corresponding instability modes are indicated on the stability diagram (Fig. 2) as predicted by linear stability theory.

Following case B1, the last control case (B2) has three slightly lower frequency modes with nondimensional frequencies $F_1 = 11$, $F_2 = 21$, and $F_3 = 31$ (corresponding to $f_1 = 103$ Hz, $f_2 = 198$ Hz, and $f_3 = 294$ Hz), again with equal initial amplitudes. This configuration creates a weaker multifrequency test case in which the actuator voltage limits become a decisive factor, as discussed in Sec. IV B.

C. Single-step deep reinforcement learning

Deep reinforcement learning merges reinforcement learning with deep neural networks to help an agent learn optimal actions through trial and error to maximize rewards in an environment. The process involves the agent observing its current state, taking an action, and receiving a reward after moving to a new state, continuing until a termination state is reached. Deep neural networks, resembling human neural structures, are used for decision-making, where neurons process and propagate information through layers, adjusting weights and biases via back-propagation in a training process to improve decision accuracy.

Single-step DRL is a recent subset of DRL based on the idea that modified versions of standard DRL algorithms can be used more efficiently. In this approach, it is sufficient for an agent to interact with its environment in just one step per episode. This method is particularly effective when the behavior to be learned is independent of state history. The main novelty of single-step DRL is that, unlike typical DRL where a deep neural network learns the best action-based responses to maximize rewards through observations, it directly learns the optimal action mapping f^* from a single, constant input state S_0. This means the agent is constantly given S_0 to determine the best action $\hat{a}$ that yields the highest reward, simplifying the learning process. As a result, single-step DRL can employ significantly smaller networks than usual because the agent only needs to transform a constant input state into an action, bypassing the need to learn complex state-action relationships.

The present study leverages a model-free, off-policy single-step deep reinforcement learning algorithm introduced in Ref. [40], a variant of policy-based gradient reinforcement learning, which interacts with the actual environment rather than a surrogate model. In terms of behavioral modeling, SDRL is characterized by its reliance on a parameterized probability distribution of actions, $\pi_\theta(a)$, which is continuously optimized using a policy gradient method.

For action selection, SDRL utilizes a d-dimensional multivariate normal distribution. This choice supports a robust exploration and exploitation balance by incorporating a full covariance matrix, unlike the simplified approach of single-step PPO algorithms that assume independence and equal variance across all variables. The use of a full covariance matrix not only enhances the sampling effectiveness but also aligns exploration strategies with the steepest gradient directions of the objective function, thus facilitating more efficient policy optimization [40].

Additionally, the implementation involves three neural networks that output the necessary statistical parameters—mean, standard deviation, and correlation coefficients—using hypersphere decomposition [54,55] to ensure the covariance matrices are symmetric and positive semidefinite. This structural configuration allows for dynamically adjusting the model architectures and parameters effectively to suit specific environmental challenges. Moreover, actions, initially drawn within a normalized range $[-1; 1]^d$, are subsequently mapped to real-world physical ranges through a problem-specific transformation step. Clipping the action values within $[-1, 1]$ ensures stability in learning by preventing extreme values that could destabilize training. Additionally, it prevents numerical instabilities and helps regulate policy updates, avoiding large gradient variations that could hinder convergence. Finally, the policy parameters are optimized with the Adam algorithm [56] through stochastic gradient ascent of the loss function defined in Eq. (5),

$$L(\theta) = \mathbb{E}_{a\sim\pi_\theta}[\max(\hat{r}, 0) \log \pi_\theta(a)], \tag{5}$$

where $\hat{r}$ represents the whitened reward, normalized to have zero mean and unit variance, making it an effective advantage estimator. Normalizing the data to have zero mean and unit standard deviation may introduce a bias but it lowers variance, which decreases the actions required to estimate the expected value [41]. Additionally, using the maximum value helps to exclude actions with negative advantages, preventing instability in learning that can occur when multiple minibatch gradient steps are performed with the same data, as each step may increasingly diverge from the initial policy [41].

III. CONTROL FRAMEWORK

Robust, autonomous control of TS waves requires a capable automatic control system that can operate under realistic implementation conditions, requiring simple mathematical operations to ensure efficient execution in high-speed real-time embedded processors. For this reason, the SDRL and FXLMS algorithms are considered for the control framework, as they offer adaptive control capabilities with computational efficiency, making them suitable for real-time applications. This section provides a mathematical overview of both FXLMS- and SDRL-based controllers. For a comprehensive introduction to active noise control techniques like FXLMS, refer to Hansen's book [57].

The control system uses a reference signal, $\boldsymbol{x_s}(k)$, sensed upstream of the actuator to capture incoming disturbances, which in this study are unsteady pressure values on the flat plate. Similarly, an error signal of wall pressure, $\boldsymbol{e_s}(k)$, is sensed downstream of the actuator. The control objective is to use finite impulse response (FIR) filters [58] to process these signals and drive the actuator to attenuate pressure perturbation downstream of the actuator. FIR filters are chosen for their stability, linear phase response, and ability to effectively remove high-frequency noise without introducing phase distortion, ensuring accurate signal processing for the control system. A schematic of the working principle is shown in Fig. 1. The communication between the controller algorithm and the actuator and sensors forms a closed-loop controller that continuously adapts the system.

As already discussed, the controller operates by updating the coefficients of the FIR filter to minimize the error signal. The output of the FIR filter is computed as follows:

$$\boldsymbol{y_s}(k) = \boldsymbol{a}(k)^T \cdot \boldsymbol{x_s}(k) = \sum_{i=0}^{L-1} a_i(k)\, x_s(k-i), \tag{6}$$

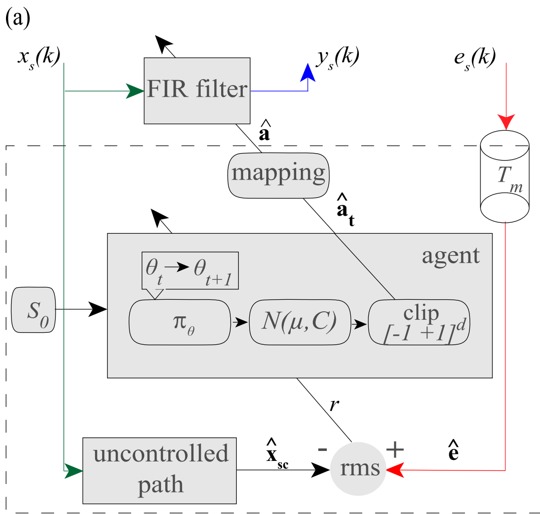

FIG. 3. Schematic of the control unit presented in Fig. 1: (a) SDRL and (b) FXLMS. $\boldsymbol{x}_s(k)$, $\boldsymbol{y}_s(k)$, and $\boldsymbol{e}_s(k)$ refer to reference, actuator, and error signals, respectively. The dashed rectangle includes the operations performed inside a time horizon buffer of T_m.

where

$$\boldsymbol{a}(k) = [a_0(k), a_1(k), \dots, a_{L-1}(k)]^T, \quad \boldsymbol{x}_s(k) = [x_s(k), x_s(k-1), \dots, x_s(k-L+1)]^T, \qquad (7)$$

where $\boldsymbol{x}_s(\boldsymbol{k})$ represents the L most recent samples of the reference signal and $\boldsymbol{a}(k)$ is the filter coefficient vector at time step k. To achieve the desired cancellation effect, the weights of the filter $\boldsymbol{a}(k)$ must be adapted. The update of the FIR coefficients can be implemented using various algorithms and control techniques. Two controllers discussed here employ different methodologies for updating the FIR coefficients, as shown in Fig. 3, though both aim to attenuate the pressure perturbation at the error sensor downstream. Below, the procedure is outlined, along with the key differences between these two controllers.

A. SDRL-based FIR coefficient update

The main approach for updating the FIR filter coefficients in this work is the SDRL algorithm as a policy-based optimization algorithm. In this control strategy, the FIR coefficients are sampled from a parameterized probability distribution of actions $\pi_\theta(\hat{\boldsymbol{a}}_t)$ which is optimized by gradient ascent.

The agent at the first iteration (each new set of evaluations) tries a random policy based on an initial set of parameters ($\boldsymbol{\theta}_0$) and defines the corresponding actions. As illustrated in Fig. 3(a), the agent samples a set of actions ($\hat{\boldsymbol{a}}_t$) from the current policy (π_θ) and iteratively updates the policy parameters ($\boldsymbol{\theta}_{t+1}$) to maximize the resulting rewards (r) in subsequent actions. A key feature of SDRL compared to other DRL methods is the use of a compact policy network, resulting from the fact that the input state is a single constant value (S_0). This property is inherent to the SDRL framework, which learns an optimal action from a fixed state rather than a time-varying trajectory. The required temporal dependence is not encoded in the agent's state but is instead carried by the reference signal and the FIR filter that the agent parameterizes. The resulting actuation inherits the spectral content of the incoming TS waves and achieves cancellation through amplitude-phase shaping, while the policy itself remains state independent. This does not imply that the physical control problem is state independent; rather, SDRL optimizes a stationary parametrization of the controller over a monitoring horizon. For problems requiring explicit state dependence, the framework could be extended to a multistep DRL formulation with a Markovian state representation.

Based on the schematic of the SDRL action loop, Fig. 3(a), the input of the agent is always constant (S_0) and repeated at each iteration while the agent draws actions in $[-1, +1]^d$ by clipping from a probability distribution function, which is a d-dimensional multivariate normal distribution, $\mathcal{N}(\mu, \boldsymbol{C})$. μ is the mean value and $\boldsymbol{C}$ is the full covariance matrix defined in Ref. [40]. The selection of d, the number of FIR filter coefficients significantly impacts both filter performance and computational efficiency. If d is too low, then the filter may inadequately suppress unwanted frequencies, causing signal distortion, insufficient smoothing in control applications, and reduced control over time-delay effects. Conversely, an excessively high d increases computational cost, causes excessive smoothing that slows system response, and prolongs impulse response duration, introducing unwanted delays. To balance these trade-offs, this study selects $d = 25$, representing a practical compromise between smooth action generation and computational efficiency. This value is chosen based on the characteristic timescales of the TS waves and the convective delay between sensors, providing a temporal window sufficient to represent the wave packet. Preliminary tuning confirmed that smaller values reduced control effectiveness, while larger values increased computational cost without improving performance.

Three neural networks are implemented to adjust the mean, standard deviation, and correlation information of that distribution. The first network outputs the mean, m, in $[-1, 1]^d$ using a hyperbolic tangent activation function on the output layer. The second network outputs the standard deviations, σ, in $[0, 1]^d$ using a sigmoid activation function on the output layer. Finally, the third network outputs the correlation coefficients, ρ, in $[0, 1]^d$, also using a sigmoid activation function on the output layer. The sampled actions, $\hat{\boldsymbol{a}}_t$, are scaled by a fixed action bound of 0.04 to produce the actual filter coefficients, $\hat{\boldsymbol{a}}$, resulting in values within the range $[-0.04, 0.04]$. These coefficients are then fed into the control unit to construct the FIR filter used to generate the voltage signal for the actuator. This scaling-based mapping is essential for enforcing action bounds and preventing actuator saturation, which is particularly important in practical control applications [44]. In this work, the action bound is chosen based on the maximum amplitude of pressure perturbations observed across all studied control cases. The $[-0.04, 0.04]$ bound was selected through preliminary tuning to ensure both stable learning dynamics and physically meaningful actuation levels. Smaller bounds limited the controller's ability to generate sufficiently strong opposition, while significantly larger bounds led to unstable training behavior due to excessively large coefficients during exploration. The chosen range represents a practical compromise.

The reward of each case is calculated based on the attenuation in root-mean-square (rms) pressure perturbation at the error sensor location ($e_{\rm rms}$) compared to the uncontrolled case ($e_{\rm rms}$) which is presented in Eq. (8),

$$r(\%) = \frac{e_{\mathrm{rms}_0} - e_{\mathrm{rms}}}{e_{\mathrm{rms}_0}}. \tag{8}$$

The root mean square of the pressure data is computed over a monitoring time horizon (T_m), which is detailed in Sec. III A 1. These reward values are fed back to the agent to update the policy with the main aim of maximizing the cost function defined in Eq. (5). Accurately computing the expected value in the policy loss [Eq. (5)] requires sampling a large number of action-reward tuples before the algorithm can proceed to update the agent parameters. The number of samples in this study is constrained by the high computational cost of solving the unsteady incompressible Navier-Stokes equations. Consequently, the number of generated action-reward tuples is generally limited. To address this, SDRL enhances the reliability of loss evaluation by incorporating data from multiple previous iterations. A decay parameter, $\eta = 0.98$, is applied to prioritize recent data by exponentially diminishing the contribution of rewards from earlier generations, following the loss formulation recommended in Ref. [40]. This strategy helps stabilize policy updates, especially towards the end of the optimization process when approaching local or global minima.

In the implementation, each of the three distinct neural networks undergoes training for $N_{\rm ep}$ epochs, where each epoch represents a complete traversal through the dataset. The data are partitioned into N_b minibatches, each containing two samples (minimum samples for saving

TABLE I. Details of the SDRL hyperparameters. The "architecture" row indicates the number of neurons in each hidden layer; only the hidden layer sizes are listed.

Parameter	Description	μ	σ	ρ
lr_0	Initial learning rate	5×10^{-4}	5×10^{-4}	10^{-4}
N_{ep}	Epochs number	128	16	16
N_h	Learning episode history	1	8	16
N_b	Minibatch number	1	4	8
τ	Improvement threshold	0.1		
λ_{lr}	Learning rate decay	50%		
Architecture	[2, 2, 2]			

computational power), corresponding to the number of episodes evaluated per iteration. The networks may differ in both their architectural details and their hyperparameters, as illustrated in Table I, where the hyperparameters are individually defined for each network. The values presented in Table I are mainly based on suggestions in Ref. [40]. The learning rate for the three neural networks follows an adaptive learning rate procedure to ensure efficient convergence. The method assesses the progression of the learning process by comparing the average reward over a window size of 15 episodes. The improvement condition is 10% increase of the average reward over the window, and if the improvement condition is not met, indicating suboptimal progress, then the learning rates adjust downwards based on the learning rate decay of 50%. The updated learning rates, lr_μ, lr_σ, and lr_ρ, are computed in Eq. (9):

$$\mathrm{lr}_{\mathrm{updated}} = \max(\mathrm{lr} \times 0.5, 10^{-10}). \tag{9}$$

This approach ensures that the learning rates are dynamically changed based on the progress of the algorithm, without diminishing to ineffectively small values, maintaining a minimum threshold of 10^{-10}.

The SDRL's iterative process continues until it meets the convergence criteria, which is having less than 0.5% variation in the reward moving average across a window of 20 episodes. Setting stricter convergence criteria can extend the convergence time without significantly improving the suppression of TS waves. More details regarding SDRL, its hyperparameters, and its formalism can be found in Ref. [40]. The implementation used in the present work is derived from the one available on the GitHub repository [59] related to Ref. [40].

Delay and monitoring time horizon

The monitoring time horizon is a crucial aspect of signal processing and machine learning–based control systems, particularly in reinforcement learning and sequential decision-making tasks. It ensures accurate detection, filtering, and delay compensation, all of which are critical for real-time control applications. The primary objective is to establish a robust observation framework that efficiently captures the critical characteristics of dynamic signals.

The control frequency f_c is defined as:

$$f_c = \frac{1}{T_c}, \quad T_c = T_d + T_m, \tag{10}$$

where T_c is the control period, defined as the time between two consecutive actions. In this study, the control period comprises two main components: the delay time (T_d) and the monitoring time horizon (T_m).

The delay corresponds to the time required for TS waves to convect from the actuator to the error sensor location. TS waves propagate in low Reynolds number flows with a phase speed of approximately 30% of the freestream velocity [60]. Therefore, the modified TS waves over the actuator need approximately 5 ms to travel from the actuator to the error sensor, given a freestream velocity of 30 m/s and a distance of 5 cm between the actuator and the error sensor. This delay must be explicitly accounted for in reinforcement learning–based controllers to ensure accurate policy updates and optimal decision-making.

The monitoring time horizon, T_m, is the time horizon where the statistics of the error signal are extracted in terms of root-mean-square amplitude to provide feedback to the policy optimization algorithm. To ensure that sufficient temporal information is captured, the observation window must span multiple periods of the dominant signal frequency. Considering the highest frequency ($f_{\max} = 334$ Hz), the minimum required time horizon can be set as $T_m \geqslant 5 \times \frac{1}{f_{\max}}$. This ensures that at least five periods of the highest frequency TS mode are captured, allowing the extracted error signal statistics to be representative of the system's response to control actions. Based on this criterion, a conservative choice of $T_m = 15$ ms is adopted for all control scenarios. Consequently, the total control period is computed as $T_c = 20$ ms leading to a control frequency of $f_c = 50$ Hz, as determined from Eq. (10).

B. Least mean square–based FIR coefficient update (FXLMS)

The second approach for updating the FIR coefficient used in this study is the FXLMS algorithm [58] presented in Fig. 3 which adapts the filter coefficients by comparing a filtered version of the reference signal with the error signal at a time instant k to compute an updated set of filter coefficients for the time instant $k + 1$:

$$\boldsymbol{a}(k+1) = \boldsymbol{a}(k) - 2\hat{\mu}\boldsymbol{e}_s(k)\boldsymbol{f}_s(k). \tag{11}$$

In this context, $\hat{\mu}$ represents the convergence coefficient, which is instrumental in adjusting the adaptation process's convergence rate, as described in Ref. [57]. The formula for $\hat{\mu}$ is given by $\hat{\mu} = \frac{\beta}{\boldsymbol{x_s}^T(k)\boldsymbol{x}_s(k)}$, where β is a predetermined constant for convergence. By normalizing with respect to the strength of the reference signal, this approach improves the robustness of the adaptation process [4].

The error signal is denoted by $\boldsymbol{e}_s(k)$, and $\boldsymbol{f}_s(k)$ represents a filtered form of the reference signal. In the control algorithm, $\boldsymbol{f}_s(k)$ is generated by passing the reference signal through a secondary FIR filter (FIR filter II). This filter serves as a digital model of the natural cancellation path, facilitating a direct comparison between the reference and error signals during the update process. To ensure this comparison is effective for updating, the cancellation path's replica, FIR filter II, is derived using a system identification routine that employs a modified version of the LMS adaptive algorithm. Specifically, the routine drives the actuator with a preset signal—here the reference signal $\boldsymbol{x}_s(k)$. Concurrently, $\boldsymbol{x}_s(k)$ is processed through the secondary filter. The output is then subtracted from the error signal, and the difference is utilized in the LMS routine to adjust FIR filter II's coefficients. As a result of using these signal discrepancies for updates, the secondary filter's final weight coefficients ensure that $\boldsymbol{f}_s(k) = \boldsymbol{e}_s(k)$. This implies that upon convergence, FIR filter II accurately models the natural cancellation path as required by the primary control system. After the system identification algorithm stabilizes and FIR filter II is established [shown as circuit A in Fig. 3(b)], the control sequence can proceed (circuit B). The system identification and control algorithms are integrated into the OpenFOAM framework and operate simultaneously with the flow solver. Key parameters—such as primary and secondary filter lengths, sampling rate, weight update rate, and convergence constants β—are set prior to the simulation based on established reference work [4].

IV. RESULTS

The controller performance in terms of TS wave suppression will be discussed in this section for the SDRL-based controller and then compared with the FXLMS-based controller in Sec. IV C. Two different categories of test cases are considered as discussed in Sec. II B including the single- and

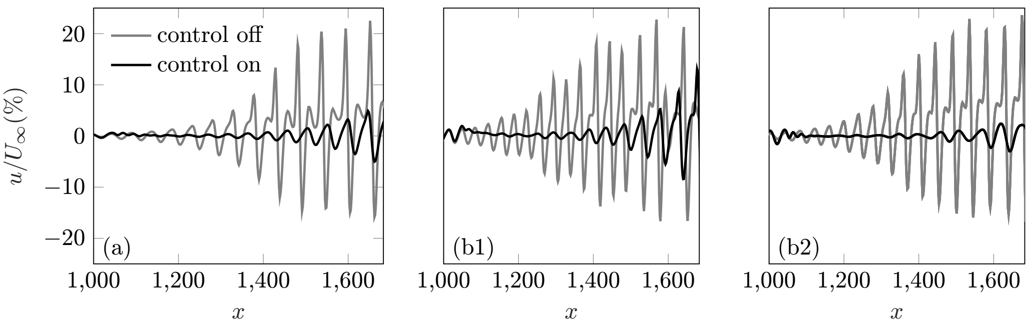

FIG. 4. Normalized maximum streamwise (u) disturbance velocity, with and without control for the single-frequency control case A, and multifrequency control cases (B1 and B2). Snapshot is taken at $t = 5$ s.

multifrequency control cases. The convergence speed of the controllers, along with their robustness to measurement and actuation noises, will also be discussed in this section.

A. Control case A: Nonlinear single-frequency control case

Control case A is explored in the context of nonlinearly developing single-frequency ($F = 25$) TS waves introduced at the domain inflow and controlled with a DBD actuator located at $x = 1039$. The reference sensor is located at $x = 974$ and the error sensor is located at $x = 1104$. The location of the actuators and sensors with respect to the computational domain is presented in Fig. 2. The normalized maximum streamwise velocity is shown in Fig. 4. This demonstrates that the SDRL-based controller effectively reduces the instability amplitude, achieving an order-of-magnitude decrease in the TS wave amplitude for case A. However, downstream of the actuator's position, the boundary layer remains within the unstable regime for the investigated mode, leading to a rapid reamplification of the residual disturbance. The suppression of 6.95 dB (79.82% reduction in pressure perturbation root mean square) is observed in case A over the error sensor location, as seen in Table II.

The sensors' data, including the reference and error signals alongside the applied voltage by the actuator, are presented in Fig. 5. As shown in Fig. 5(a), case A, the pressure perturbations start to attenuate noticeably after 2 s, while the full convergence for the controller takes longer considering the applied voltage signal. The stochastic exploration of the controller is obvious from the applied voltage in Fig. 5(b), case A, where the initial values are oscillating between 0 and 16 kV while after convergence, the range of oscillation is limited between 8 and 12 kV.

TABLE II. Summary of test cases and pressure perturbation reduction ΔP_e, defined as the reduction in pressure perturbation at the error sensor location in the controlled case compared to the uncontrolled case.

Cases	Frequencies (F)	Controller	ΔP_e (dB)(%)
A1	Single (25)	SDRL	–6.95 (79.82%)
A2	Single (25)	FXLMS	–5.29 (70.42%)
B1.1	Multi (15, 25, 35)	SDRL	–10.38 (91.00%)
B1.2	Multi (15, 25, 35)	FXLMS	–9.10 (87.65%)
B2.1	Multi (11, 21, 31)	SDRL	–12.85 (94.81%)
B2.2	Multi (11, 21, 31)	FXLMS	–11.51 (92.94%)

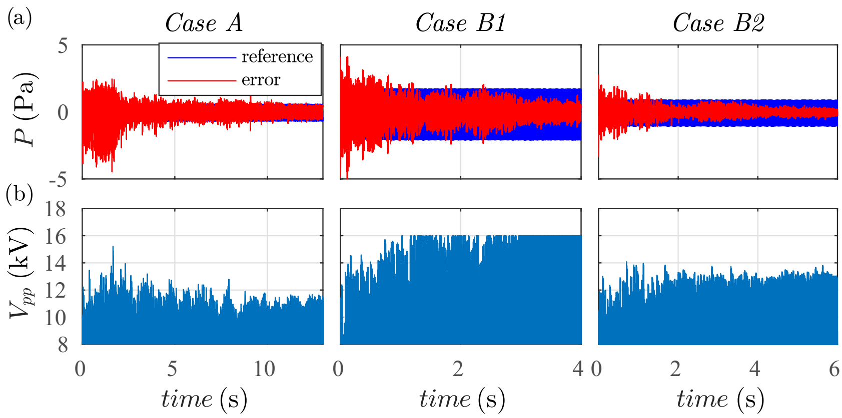

FIG. 5. Controller operation with reference and error signals (a), and actuator applied voltage (b) for single-frequency control case A, and multifrequency control cases B1 and B2.

B. Control cases B: Nonlinear multifrequency control

This section explores the SDRL-based controller performance in nonlinear multifrequency control cases. A wave train consisting of three TS modes ($F = 15$, 25, and 35) is initiated at the inflow for case B1. All modes are assigned equal initial amplitudes. A snapshot of the streamwise and wall-normal velocities and pressure fields is depicted in Fig. 6 for the nonlinear multifrequency case B1. The suppression of the pressure and velocity perturbations are evidence of the successful control of TS waves especially in the vicinity of the actuator located at $x = 1039$. The pressure and velocity perturbation at the downstream part of the computational domain is related to the growth of the controlled TS waves downstream.

Figure 5(a), case B1, clearly shows the high suppression of the pressure perturbation, resulting in a 10 dB (91%) attenuation in the root-mean-square amplitude of the error signal, as summarized in Table II. Due to the high strength of incoming disturbances in case B1, the actuator must operate at voltages exceeding the maximum threshold of 16 kV. In this study, the voltage is capped at 16 kV,

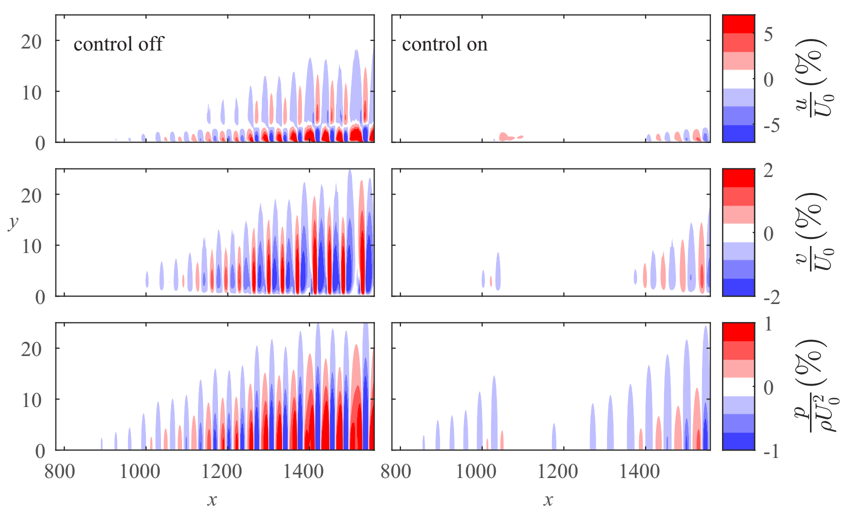

FIG. 6. Contours of normalized streamwise (u), wall-normal (v) disturbance velocity, and pressure (p) for the multifrequency case B1. The actuator is located at $x = 1039$. Snapshot is taken at $t = 5$ s.

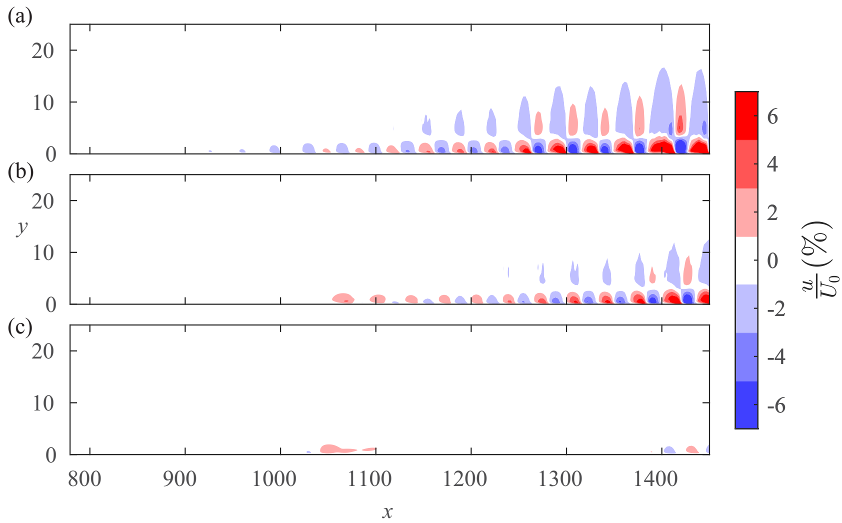

FIG. 7. Contours of normalized streamwise (u) disturbance velocity for the nonlinear multifrequency case B1, (a) baseline control case without actuation, (b) only actuation, and (c) baseline control case with actuation.

which alters the shape of the actuation signal from sinusoidal to trapezoidal. Despite this limitation, the actuator noticeably reduces the TS amplitude. This illustrates amplitude sensitivity via actuator saturation: when the incoming wave train is strong (B1), voltage clipping constrains suppression, whereas the slightly weaker case (B2) achieves larger attenuation without clipping.

The results from case B1 offer deeper insight into the controller's operating mechanism. Figure 7(a) shows the normalized streamwise velocity field resulting solely from the upstream disturbance input, representing the uncontrolled scenario. Figure 7(b) depicts the velocity field induced by the control input alone, while Fig. 7(c) illustrates the fully controlled case. A comparison of these fields reveals that the actuator-generated wave closely matches the uncontrolled wave in amplitude but is out of phase, effectively implementing an opposition control strategy. This opposition control strategy was observed in all control cases (A, B1, and B2), with case B1 presented here as a representative example.

In case B1, the controller's voltage exploration in Fig. 5(b), case B1 shows similar stochastic behavior to case A which converges within 3 seconds, with noticeable attenuation after less than 1 second as illustrated in Fig. 5(a), case B1. This quick convergence in case B1 is related to the voltage clipping stemming from practical restrictions that guide the controller for the voltage oscillation bandwidth. This led to the definition of another multifrequency control case (B2), where lower frequencies generate weaker incoming TS waves by having a wave train consisting of three TS modes ($F = 11, 21,$ and 31) initiated at the inflow. This setup prevents practical voltage clipping while retaining the multifrequency characteristic, evaluating the controller performance within narrower voltage oscillation limits.

Similarly to the case B1, the actuator responds effectively to the presence of multiple frequencies within the TS wave spectrum in case B2, as can be observed in Fig. 4. The pressure perturbation attenuation of 12.85 dB (94%) is recorded as the highest achieved suppression effect by the controller which can be observed in Fig. 8 where the pressure perturbations over the wall is illustrated before and after the DBD plasma activation at $t = 1.1$ s.

Figure 5, case B2 shows that without the clipping effect—as in case A—the controller in case B2 converges slower than case B1, achieving full convergence in approximately 4.5 s. This extra convergence time is due to the identification of a narrow optimal voltage range within the available range of 8 to 16 kV.

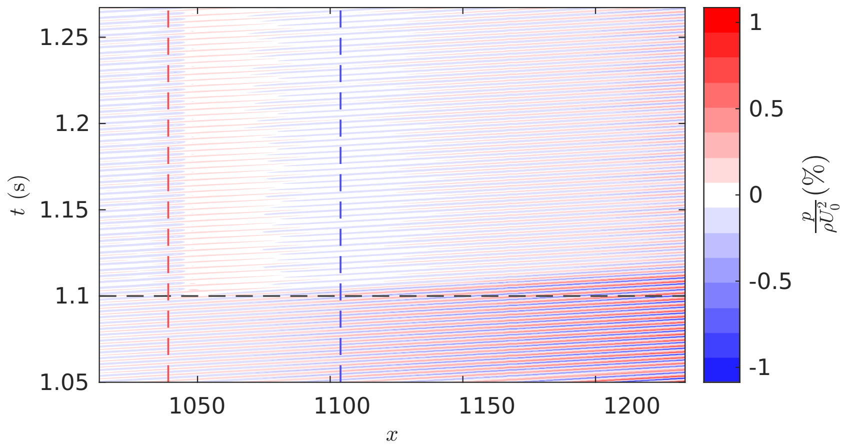

FIG. 8. Contour of normalized pressure (p) perturbation for the multifrequency case B2 under optimal control by the SDRL-based controller. The actuator, located at $x = 1039$, is indicated by a red dashed line, while the error sensor is shown with a blue dashed line. The black dashed line at $t = 1.1$s marks the activation of the DBD actuator.

C. Comparison between the SDRL- and FXLMS-based controllers

This section compares the SDRL-based controller with the previously introduced classical adaptive control method (FXLMS). The evaluation of both controllers in terms of effectiveness in suppressing TS waves and convergence speed is provided, highlighting each approach's strengths and weaknesses. For brevity, the focus is on control case B2, analyzing the output of both controllers to identify similarities and differences. Finally, the robustness of both controllers against measurement and actuation noises is studied through a comparative analysis.

Performance comparison

The reward evolution of the controllers is shown in Fig. 9 for case B2. A 15-episode averaging moving window is used to smooth the oscillations, highlighting overall trends.

Figure 9 illustrates that both controllers achieve similar overall performance in suppressing TS waves as the final converged reward, though the SDRL-based controller exhibits higher performance oscillations due to the stochastic nature of its exploration. Additionally, the SDRL-based controller initially finds a higher reward but takes longer to converge to the optimal control. Figure 10(a) presents the converged applied voltage profiles for both controllers, illustrating their comparable output characteristics. Voltages below 8 kV are interpreted as indicating no actuation by the controllers. The overall voltage profiles are similar, with negligible discrepancies in the amplitude of the

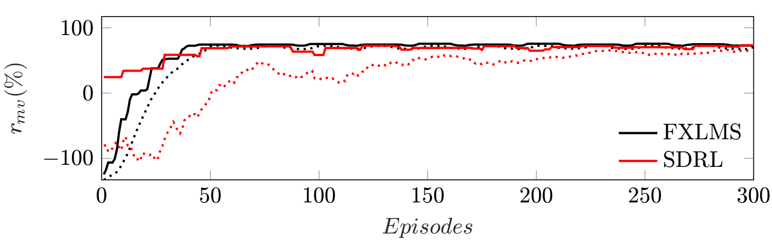

FIG. 9. Controllers reward evolution with episodes for multifrequency control case B2; solid lines: moving maximum; dotted lines: moving average; moving window size: 15 episodes.

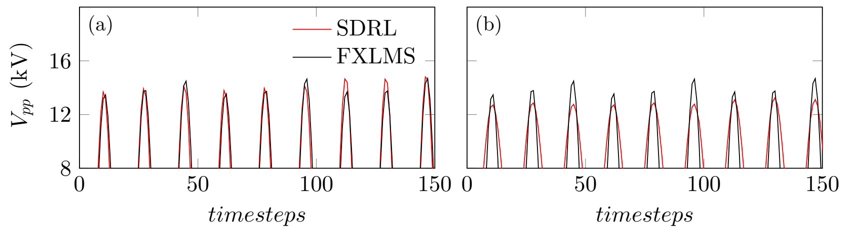

FIG. 10. Converged applied voltage for the multifrequency control case B2 using the FXLMS- and SDRL-based controllers: (a) FXLMS- and SDRL-based controllers and (b) SDRL-based controller with an enlarged duty cycle.

applied voltage. Specifically, there are instances where the voltage amplitude from the SDRL-based controller exceeds that of the FXLMS-based controller and vice versa.

In comparing the applied voltage between the two controllers, it is possible to adjust the duty cycle of the actuation by modifying the voltage signal. In the case of SDRL-based controller, a +8-kV offset was introduced during training, shifting the action space so that the voltage oscillates around 8 kV instead of 0 kV. This modification influenced the reward signal, leading the RL agent to learn a control policy optimized for a higher baseline voltage with a longer duty cycle, while maintaining the same suppression performance at a lower applied voltage, as illustrated in Fig. 10(b).

Figure 10(b) illustrates that the SDRL-based controller can adjust the voltage amplitude according to the duty cycle, a valuable control design parameter when practical restrictions on duty cycle or maximum voltage amplitude are in place. This finding suggests a potential avenue towards minimizing energy expenditure, aimed at identifying the optimal combination of voltage amplitude and duty cycle. The comparison of the reward evolution for the two controllers, as shown in Fig. 11, indicates that the new constraint of a larger duty cycle results in a higher final suppression effect for the SDRL-based controller compared to the FXLMS-based controller. This outcome is quantified in Table II, where the SDRL-based controller achieves a suppression of 12.85 dB, surpassing the 11.51 dB suppression achieved by the FXLMS-based controller. This lower performance in FXLMS-based controller is observable in Fig. 12, comparing the controlled perturbation field in both controllers. The SDRL-based controller achieves a higher reduction of wave amplitudes, as evident in the significant suppression seen in the “control on” plots, leaving minimal postcontrol velocity and pressure perturbation. In contrast, FXLMS-based controller achieves a less pronounced suppression, resulting in residual disturbances downstream. The higher streamwise velocity observed in the FXLMS-based controller in Fig. 12 is directly linked to the higher voltage applied by the controller as presented in Fig. 10(b).

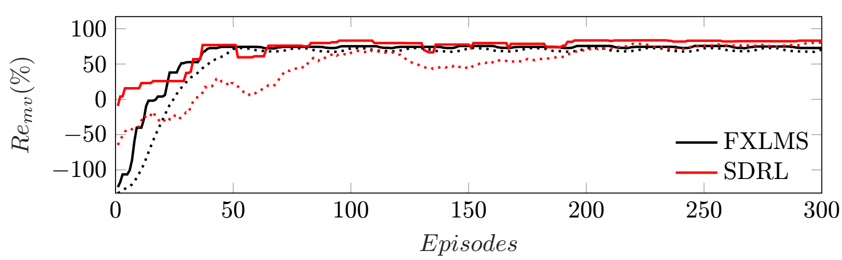

FIG. 11. Controllers reward evolution with episodes for multifrequency control case B2 with an enlarged duty cycle; solid lines: moving maximum; dotted lines: moving average.

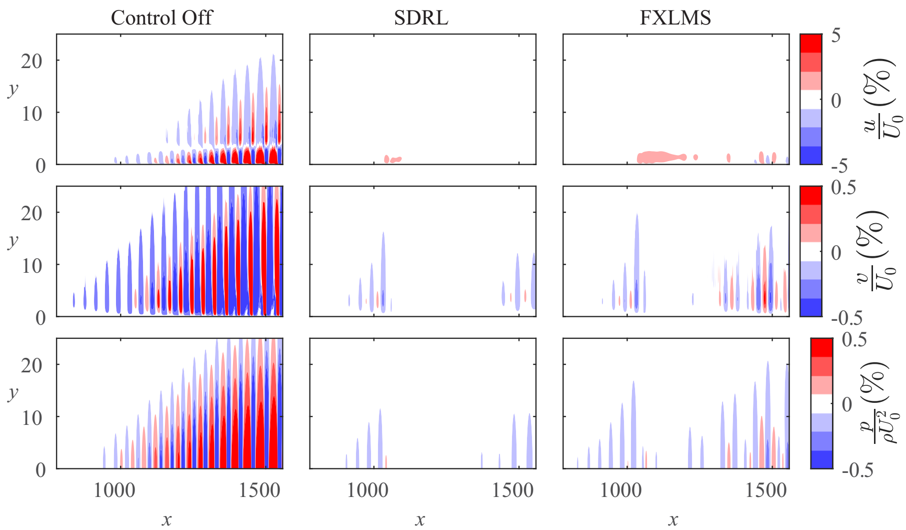

FIG. 12. Contours of normalized streamwise (u), wall-normal (v) disturbance velocity, and pressure (p) for the multifrequency case B2 for both SDRL- and FXLMS-based controllers. The actuator is located at $x = 1039$. Snapshot is taken at $t = 5$ s.

Additionally, a mean flow distortion is observed in Fig. 12, which diminishes substantially in a short distance downstream of the actuator (less than 10 TS wavelengths). The mean velocity component arises from the directional nature of the body force, which can have a local stabilizing effect on the boundary layer, further enhancing the damping of the TS waves [4].

In terms of achieving higher rewards early in the process, the SDRL-based controller demonstrates a faster initial response, as indicated by the sharper rise in its moving maximum of reward presented in Fig. 11. However, its moving average converges later than FXLMS-based controller, suggesting that while it adapts quickly, it takes longer to consistently maintain high performance. This delay indicates fluctuations in SDRL's performance, likely due to more exploration or less consistent updates. In contrast, FXLMS has a slower initial rise but stabilizes its performance, both in terms of maximum and average, earlier than the SDRL-based controller. Therefore, SDRL-based controller offers faster adaptation, whereas FXLMS-based controller excels in achieving steady, stabilized performance sooner. The preferred controller depends on whether the priority is quick initial rewards or long-term stability.

D. Robustness to noises

This section addresses the robustness of SDRL- and FXLMS-based controllers in the presence of measurement and actuation noises. Uniform white noise is introduced into the sensor and actuator data at three different signal-to-noise ratio (SNR): 5, 10, and 17 dB. White noise is chosen due to its broadband nature, ability to model real-world sensor and actuator electronic and acoustic noise, and statistical simplicity. The selected SNR levels represent different noise intensities, allowing for a comprehensive robustness evaluation under varying noise conditions. Four different scenarios are considered in this robustness study, including measurement noise on the feedback line, $\boldsymbol{e}_s(k)$ (error noise); noise on the reference sensor, $\boldsymbol{x}_s(k)$ (reference noise); and noise on actuation signal, $\boldsymbol{y}_s(k)$ (actuation noise). Additionally, a fourth scenario evaluates the effect of introducing both measurement noises and actuation noises simultaneously (combined noise). Figure 13 illustrates the episode evolution of the reward for the SDRL- and FXLMS-based controllers for these four scenarios alongside the noise-free control case. The controller robustness (Γ) is defined in Eq. (12)

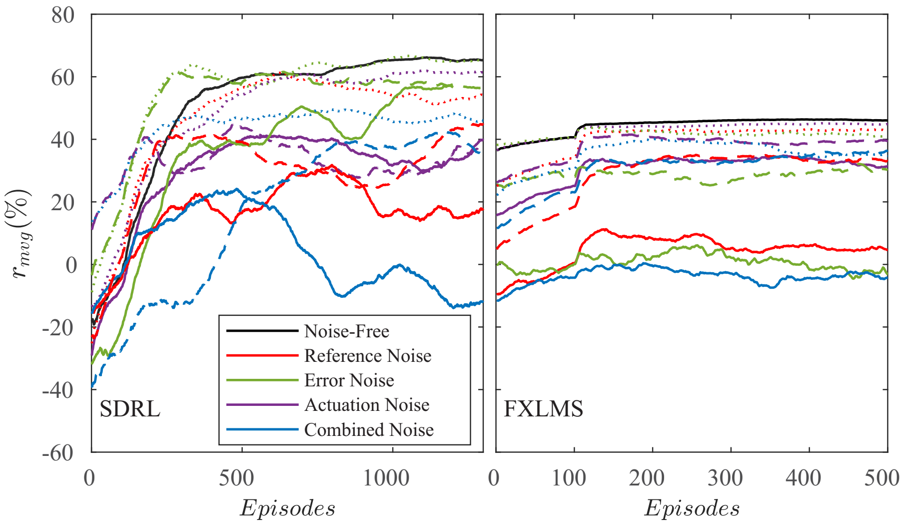

FIG. 13. Controllers robustness to measurement and actuation noises for multifrequency control case B1; black line: noise-free; colored solid lines: SNR = 5 dB; dashed lines: SNR = 10 dB; dotted lines: SNR = 17 dB.

and presented in Table III for both controllers,

$$\Gamma = \left(1 - \frac{r_{\text{noise-free}} - r_{\text{noise}}}{r_{\text{noise-free}}}\right) \times 100. \tag{12}$$

The r in Eq. (12) represents the converged reward of the algorithm, where r_{noise} corresponds to the reward obtained in the presence of noise (including measurement and actuation noises based on the studied scenario) and $r_{\text{noise-free}}$ denotes the reward achieved in the absence of noise similar to the control cases discussed in Sec. IV B.

The rapid increase observed in the reward values of the FXLMS-based controller around episode 100 in Fig. 13 occurs due to the activation threshold of the actuator at 8 kV. Initially, the actuator remains inactive, resulting in no suppression of TS waves and correspondingly low rewards. As the FXLMS algorithm gradually increases the voltage, it eventually surpasses the 8-kV threshold around

TABLE III. Robustness comparison of SDRL- and FXLMS-based controllers at different noise levels.

Noise type	SNR (dB)	SDRL robustness (%)	FXLMS robustness (%)
	5	23.77	9.99
Reference noise	10	67.61	71.76
	17	81.17	93.46
	5	86.74	−6.73
Error noise	10	86.80	65.70
	17	99.60	88.98
	5	53.13	67.41
Actuation noise	10	54.40	85.86
	17	94.07	97.44
	5	−20.55	−9.34
Combined noises	10	58.95	79.01
	17	71.32	75.47

episode 100, activating the actuator. At this point, the suppression of TS waves begins, causing a sudden increase in the reward values.

Analyzing the performance of the SDRL- and FXLMS-based controllers across different types of noise at varying SNRs helps pinpoint their respective strengths and weaknesses under specific conditions. The robustness of both controllers against noise from the reference sensor is comparable. However, SDRL-based controller outperforms FXLMS-based controller in the majority of scenarios with low SNR, while FXLMS-based controller performs better in medium and high SNR cases.

As shown in Fig. 13, the SDRL-based controller shows varied impacts from different noise types. Reference noise proves most detrimental to performance, significantly impacting reward values. The reason is structural: In the SDRL-based controller, as shown in Fig. 1, the reference signal is simultaneously (i) convolved with the learned FIR filter to generate the actuator voltage and (ii) used to compute the instantaneous reward that drives policy updates. Reference noise therefore infiltrates the loop along two distinct pathways—directly contaminating the actuation command and indirectly corrupting the reward signal—so its effect is compounded. This dual influence makes achieving robustness to reference noise intrinsically more challenging than mitigating noise that enters after the convolution. In contrast, reference noise has a moderate impact on the FXLMS-based controller performance. This happens because the reference signal is first filtered through a secondary FIR filter before being fed into the FXLMS algorithm. This filtering helps mitigate some of the noise effects, making adaptation to the noise more feasible. However, a direct effect on the voltage signal still exists, similarly to the SDRL-based controller, due to the convolution of the noisy reference signal with the FIR filter, as seen in Fig. 3(b).

The SDRL-based controller stands out with consistently high robustness for error signals at all SNRs. It maintains strong performance even when the SNR is low, unlike the FXLMS-based controller, which only achieves adequate robustness at higher SNRs. Here SDRL's robust response makes it clearly superior in maintaining higher robustness against error noise across a broad range of operating conditions. The error noise generally has a moderate effect on the SDRL-based controller performance, offering some potential for adaptation since the noise on the feedback line affects the SDRL-based controller through the reward and has an indirect effect on the voltage. Therefore, this type of noise is the least detrimental source of noise for the SDRL-based controller. This noise also has some advantages for the controller speed, as can be seen in the faster convergence of the controller with error noise compared to the noise-free control case in Fig. 13. This faster SDRL-based controller convergence is attributed to noise-induced exploration. In reinforcement learning, a balance between exploration (trying new actions) and exploitation (using the best-known actions) is essential. Noise in $\boldsymbol{e}_s(k)$ acts as a perturbation that indirectly encourages the agent to explore a wider range of actions. This can be particularly beneficial in the early stages of training, as it prevents the agent from converging prematurely to a suboptimal policy. In this system, the error signal $\boldsymbol{e}_s(k)$ influences the reward received by the agent. In the presence of noise, the perceived error fluctuates even for the same underlying system behavior, causing the reward signal to vary. This stochasticity drives the agent to try different actions, effectively boosting exploration. As a result, the agent may converge faster because it quickly identifies a "good enough" policy based on noisy feedback. While this accelerates convergence, it can lead to suboptimal learning outcomes. The agent might prematurely settle on a policy that minimizes the noisy error rather than optimizing the true system performance. In contrast, in a noise-free case, the consistent reward signal allows the agent to explore subtle action adjustments over a longer time, potentially leading to a more optimal policy. This behavior mirrors phenomena observed in optimization, such as stochastic gradient descent (SGD), where added noise can help escape local minima but may also result in convergence to a suboptimal solution.

On the other side, the FXLMS-based controller performance, depicted in Fig. 13, is significantly impacted by error noise, as evidenced by the robustness data in Table III. This adverse effect occurs because the noise on the error signal not only propagates through the feedback line of the FXLMS algorithm but also impacts the cancellation path and indirectly affects the filtered reference signal, as shown in Fig. 3(b).

Actuation noise minimally impacts the SDRL-based controller because it indirectly influences the reward through the error signal. This indirect effect makes control in this noisy environment challenging, as the controller's output is continuously altered by the noise, and the controller only perceives the secondary effects of this noise through the feedback line. Figure 13 shows that the effect of noise on controller output is less severe, even in high-level noise (SNR = 5), compared to the effect of noise on algorithm inputs. This is because, although output noise can degrade the performance or quality of the final output, it usually does not propagate back to influence the core functionalities of the system's control mechanisms. The FXLMS-based controller demonstrates its capability by having higher robustness at the lowest SNR for actuation noise and maintaining high robustness throughout, including at the highest SNR tested. The SDRL-based controller, although improving at higher SNRs, does not match FXLMS's performance across the SNR spectrum. This makes FXLMS-based controller the more effective choice when actuation noise is present, offering reliable control even in less favorable noise conditions.

Both controllers show vulnerabilities at the lowest SNR for combined noises, but FXLMS-based controller recovers more effectively and outperforms SDRL-based controller at medium SNR, maintaining good robustness at high SNR. This adaptability of FXLMS-based controller suggests it is better suited for environments with fluctuating SNR conditions, making it a more versatile and dependable choice in complex noise environments. While FXLMS-based controller generally provides better performance across most noise types and SNRs with higher performance stability, SDRL-based controller excels in controlling high-level noise cases with high robustness throughout. The type of noise critically affects their robustness, with direct and indirect impacts varying based on the noise's source and the controller's specific feedback and filtering mechanisms. Choosing between these controllers should be guided by the specific noise types and SNR conditions anticipated in the operational environment. FXLMS-based controller offers broad reliability, making it ideal for variable conditions, whereas SDRL-based controller could be strategically deployed where its specific strengths align closely with the environmental characteristics.

Although the robustness analysis presented here focuses on white noise due to its broadband nature and statistical simplicity, similar qualitative trends are expected under colored noise, with frequency-dependent effects. If the dominant spectral content of the colored noise is far from the TS instability frequencies, then its impact on control performance is expected to be limited and comparable to or weaker than that of white noise. Conversely, if the noise spectrum contains significant energy near the unstable TS modes, then it can interfere more strongly with sensing and actuation, potentially leading to greater performance degradation. In such cases, the noise becomes indistinguishable from the actual TS disturbances, which reduces the controller's ability to identify and cancel the instability waves. The relative performance trends between SDRL and FXLMS are expected to remain similar under colored noise. A detailed investigation of colored-noise robustness represents valuable potential future work and lies beyond the scope of the present study.

Finally, sensitivity to operating conditions beyond noise, namely variations in Reynolds number and disturbance amplitude, is discussed. Variations in Re primarily affect (i) the TS neutral stability curve and the most-amplified frequencies and (ii) the convective delay between sensors and actuator. For moderate Re changes, the controller remains effective because the upstream reference sensor continuously captures the actual incoming spectrum, and the FIR filter parametrization shapes amplitude and phase at those frequencies over the monitoring horizon. Small shifts in convective speed (and thus delay) are implicitly compensated by the convolution. Noticeable degradation is expected only if Re shifts the dominant energy outside the controller's practical bandwidth (set by the sampling rate and FIR length) or if the delay mismatch becomes a significant fraction of the TS period—conditions that would require retuning of the sampling frequency or FIR horizon.

Regarding sensitivity to TS wave disturbance amplitude variation, the controller maintains its effectiveness until two practical limits are encountered: (i) nonlinearity of the instability, which weakens opposition control, and (ii) actuator saturation. This behavior is evident in the multi-frequency cases: stronger instability (case B1) drives the voltage signal toward the upper bound, causing clipping and limiting the achievable cancellation, whereas slightly weaker instability (case

B2) enables higher attenuation without clipping. Sensitivity to amplitude is therefore governed primarily by physical nonlinearity and actuator limitations, rather than by the learning algorithm itself.

V. CONCLUSION

This study demonstrates the feasibility of implementing the SDRL algorithm in a closed-loop control system for effective real-time suppression of TS waves in a two-dimensional laminar boundary layer using DBD plasma actuators. The SDRL-based controller achieves up to 94.81% suppression of multifrequency TS waves. The opposition control strategy is identified as the dominant mechanism, where the controller generates a counterwave that cancels out the TS wave instabilities.

A comparative analysis with the classical adaptive FXLMS controller is performed, which highlights that the SDRL provides better control over multifrequency TS waves, offering higher suppression and faster adaptation in most cases. While FXLMS stabilizes faster, SDRL adapts faster in multifrequency control cases, achieving higher initial rewards, though it faces challenges in reaching optimal convergence. The control outputs, including the DBD plasma actuator voltage and overall suppression mechanism, remain similar for both controllers.

The robustness of SDRL against measurement and actuation noise is evaluated and compared with the FXLMS-based controller. The SDRL-based controller demonstrates exceptional robustness against noisy error signals across a wide range of SNRs, making it highly effective in the control of TS waves, particularly in environments with noisy feedback, such as experimental wind tunnel tests. Its superior resilience to noise, attributed to noise-induced exploration, enables faster convergence, similarly to the SGD method. While actuation noise can degrade output quality, it has a lesser impact on the system's core control mechanisms and therefore the final performance of the controllers. The robustness study confirms that FXLMS generally performs better across most noise conditions, offering broad reliability, but SDRL excels in high-noise scenarios, making it a strategic choice when robustness is a priority.

By leveraging compact neural networks and operating with only two wall-pressure sensors, this methodology provides a lightweight and scalable controller, especially attractive for experimental deployment, where computational overhead and sensor density are often limiting factors. Unlike FXLMS and most model-free controllers, SDRL does not require separate system identification, making it a more efficient and adaptable approach for real-time control of convective instabilities. While SDRL performs well in the linear growth regime, its effectiveness in more complex, nonlinear conditions may benefit from further refinement. Future work will explore advanced DRL techniques to improve control in nonlinear growth regimes and enhance energy efficiency by optimizing voltage amplitude and duty cycles. Additionally, hybrid control strategies that integrate SDRL with classical techniques could be investigated as a future study to enhance overall robustness.

Finally, while the present implementation relies on reference sensing and FIR filtering to adapt to the incoming spectral content, the framework could be extended to allow the reinforcement learning agent to actively select actuation frequencies. This would involve augmenting the action or state space to include frequency as a decision variable, enabling the agent to explore and identify effective frequencies for control. Such an extension would increase the optimization dimensionality and exploration requirements, and is therefore a potential case study for future work.

ACKNOWLEDGMENTS

The authors thank the supervisory team for their continuous guidance and support throughout this work. Gratitude is also extended to colleagues and collaborators whose feedback and discussions contributed to the development of this study. This research was supported by the AIFluids Lab at Delft University of Technology (TU Delft).

The authors report no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available upon publication because it is not technically feasible and/or the cost of preparing, depositing, and hosting the data would be prohibitive within the terms of this research project. The data are available from the authors upon reasonable request.

APPENDIX: REPEATABILITY OF CONTROLLER PERFORMANCE

The repeatability of the controller was evaluated through five independent training runs conducted under identical simulation and control settings, control case B2. Each run was initialized with a different random seed, influencing stochastic aspects of the learning process such as policy initialization and exploration.

Figure 14 illustrates the episode-wise evolution of the moving average reward for all five repetitions, together with the mean trajectory and the standard deviation envelope. Quantitative metrics are summarized in Table IV.

The final mean reward, averaged over the last 50 episodes, was 76.5%, 72.9%, 73.3%, 78.3%, and 68.6% for the five runs, corresponding to an overall mean of 73.9% with a standard deviation of 3.7%. The cumulative performance, measured by the area under the reward curve, ranged from 1.33×10^4 to 1.74×10^4, with an overall mean of 1.53×10^4 and a standard deviation of 0.18×10^4. These results indicate consistent convergence behavior and comparable cumulative learning performance across repetitions.

It is also observed that the standard deviation among runs is relatively high during the initial training episodes and gradually decreases as the learning progresses. This behavior is attributed to the stochastic nature of policy exploration and random initialization of the neural network parameters, which cause noticeable trajectory divergence during early learning stages. As the controller accumulates experience and the policy updates stabilize, the exploration rate diminishes, leading to convergence of the reward trajectories and a marked reduction in variance. Such a trend is typical for reinforcement learning processes and indicates that, despite early variability, the controller consistently converges toward a similar optimal policy.

Overall, the results demonstrate that the controller achieves repeatable performance across independent runs. The observed variability, on the order of 5–10% of the mean reward, is within

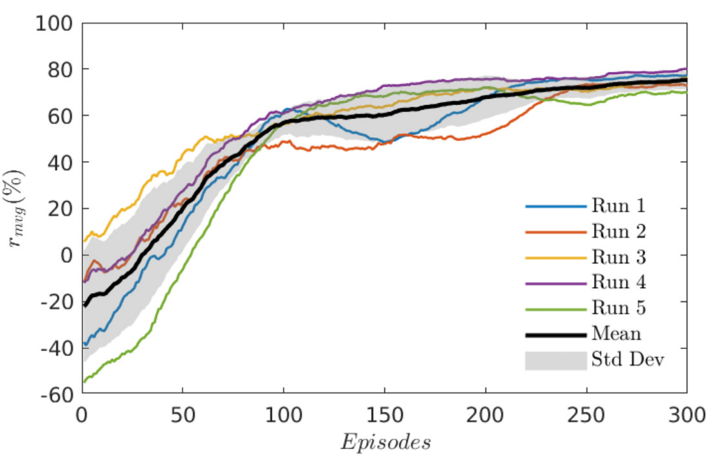

FIG. 14. Reward moving average evolution over training episodes for five independent repetitions (colored lines) of the multifrequency control case B2, with the mean trajectory (black line) and standard deviation envelope (shaded region). The moving average window size is 50 episodes.

TABLE IV. Quantitative comparison of controller performance across five repetitions.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5	Mean ± Std
Final reward (%, last 50 episodes)	76.5	72.9	73.3	78.3	68.6	73.9 ± 3.7
Area under curve ($\times 10^4$)	1.44	1.39	1.74	1.74	1.33	1.53 ± 0.18

the range typically reported for reinforcement learning–based controllers. While larger sample sizes are generally recommended to achieve fully converged statistical metrics, the present analysis demonstrates satisfactory repeatability considering the high computational cost of CFD-based training.

[1] H. Schlichting and K. Gersten, *Boundary-Layer Theory*, 9th ed. (Springer, Berlin, 2017).

[2] A. S. W. Thomas, The control of boundary-layer transition using a wave-superposition principle, J. Fluid Mech. **1**37 233 (1983).

[3] C. Cossu and L. Brandt, Stabilization of Tollmien–Schlichting waves by finite amplitude optimal streaks in the Blasius boundary layer, Phys. Fluids **14**, L57 (2002).

[4] M. Kotsonis, R. Giepman, S. Hulshoff, and L. Veldhuis, Numerical study of the control of Tollmien–Schlichting waves using plasma actuators, AIAA J. **51**, 2353 (2013).

[5] T. Michelis, A. B. Putranto, and M. Kotsonis, Attenuation of Tollmien–Schlichting waves using resonating surface-embedded phononic crystals, Phys. Fluids **35**, 044101 (2023).

[6] B. Mohammadikalakoo, M. Kotsonis, and N. A. K. Doan, Control of Tollmien–Schlichting waves using particle swarm optimization, Phys. Fluids **36**, 124130 (2024).

[7] B. Mohammadikalakoo, M. Kotsonis, and N. A. K. Doan, Optimization of Tollmien–Schlichting waves control: Comparison between deep reinforcement learning and particle swarm optimization approach, in *13th International Symposium on Turbulence and Shear Flow Phenomena (TSFP13)* (Montréal, Canada, 2024).

[8] R. Milling, Tollmien–Schlichting wave cancellation, Phys. Fluids **2**4, 979 (1981).

[9] H. W. Liepmann, G. L. Brown, and D. M. Nosenchuck, Control of laminar-instability waves using a new technique, J. Fluid Mech. **118**, 187 (1982).

[10] R. D. Joslin, Validation of three-dimensional incompressible spatial direct numerical simulation code: A comparison with linear stability and parabolic stability equation theories for boundary-layer transition on a flat plate, National Aeronautics and Space Administration, Scientific and Technical Information Program, Vol. 3205 (1992).

[11] O. Semeraro, S. Bagheri, L. Brandt, and D. S. Henningson, Transition delay in a boundary layer flow using active control, J. Fluid Mech. **731**, 288 (2013).

[12] H. J. Tol, C. C. de Visser, and M. Kotsonis, Experimental model-based estimation and control of natural Tollmien–Schlichting waves, AIAA J. **57**, 2344 (2019).

[13] D. J. Sturzebecher and W. H. Nitsche, Active cancellation of Tollmien–Schlichting instabilities on a wing using multi-channel sensor actuator systems, Int. J. Heat Fluid Flow **24**, 572 (2003).

[14] A. Kurz, N. Goldin, R. King, C. Tropea, and S. Grundmann, Hybrid transition control approach for plasma actuators, Exp. Fluids **54**, 1610 (2013).

[15] M. Kotsonis, R. K. Shukla, and S. Pröbsting, Control of natural Tollmien–Schlichting waves using dielectric barrier discharge plasma actuators, Int. J. Flow Contr. **7**, 37 (2015).

[16] B. Simon, N. Fabbiane, T. Nemitz, L. Küster, and R. Radespiel, In-flight active wave with delayed-x-lms control algorithm in a laminar boundary layer, Exp. Fluids **57**, 160 (2016).

[17] D. B. S. Audiffred, A. V. G. Cavalieri, P. P. C. Brito, and E. Martini, Experimental control of Tollmien–Schlichting waves using the Wiener-Hopf formalism, Phys. Rev. Fluids **8**, 073902 (2023).

[18] B. A. Belson, O. Semeraro, C. W. Rowley, and D. S. Henningson, Feedback control of instabilities in the two-dimensional Blasius boundary layer: The role of sensors and actuators, Phys. Fluids **25**, 054106 (2013).

[19] H. J. Tol, M. Kotsonis, C. C. de Visser, and B. Bamieh, Localised estimation and control of linear instabilities in two-dimensional wall-bounded shear flows, J. Fluid Mech. **824**, 818 (2017).

[20] D. Ladd and E. Hendricks, Active control of 2-D instability waves on an axisymmetric body, Exp. Fluids **6**, 69 (1988).

[21] P. Pupator and W. Saric, Control of random disturbances in a laminar boundary layer, AIAA Paper **89**, (1989).

[22] D. Ladd, Control of natural instability waves on an axisymmetric body, AIAA J. **28**, 367 (1990).

[23] M. Baumann and W. Nitsche, Investigation of active control of Tollmien–Schlichting waves on a wing, in *Transitional Boundary Layers in Aeronautics,* Vol. 46 (KNAW, Amsterdam, 1996), pp. 89–98.

[24] M. Baumann and W. Nitsche, Experiments on active control of Tollmien–Schlichting waves on a wing, in, *New Results in Numerical and Experimental Fluid Mechanics,* Vol. 60 (Springer, Heidelberg, 1997), pp. 56–63.

[25] M. Baumann, D. Sturzebecher, and W. Nitsche, Active control of TS-instabilities on an unswept wing, in *Laminar-Turbulent Transition, IUTAM Symposium* (Springer-Verlag, Berlin, 2000), pp. 155–160.

[26] F. Ren, H. B. Hu, and H. Tang, Active flow control using machine learning: A brief review, J. Hydrodynam. **32**, 247 (2020).

[27] J. Rabault, M. Kuchta, A. Jensen, U. Réglade, and N. Ceradi, Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control, J. Fluid Mech. **865**, 281 (2019).

[28] J. Rabault and A. Kuhnle, Accelerating deep reinforcement learning strategies of flow control through a multi-environment approach, Phys. Fluids **31**, 094105 (2019).

[29] R. Paris, S. Benedtdine, and J. Dandois, Robust flow control and optimal sensor placement using deep reinforcement learning, J. Fluid Mech. **913**, A25. (2021).

[30] J. Li and M. Zhang, Reinforcement-learning-based control of confined cylinder wakes with stability analyses, J. Fluid Mech. **9**32, A44 (2022).

[31] W. Chen, Q. Wang, L. Yan, G. Hu, and B. R. Noack, Deep reinforcement learning-based active flow control of vortex-induced vibration of a square cylinder, Phys. Fluids **35**, 053610 (2023).

[32] C. Wang, P. Yu, and H. Huang, Reinforcement-learning-based parameter optimization of a splitter plate downstream in cylinder wake with stability analyses, Phys. Rev. Fluids **8**, 083904 (2023).

[33] W. Hu, Z. Jiang, M. Xu, and H. Hu, Efficient deep reinforcement learning strategies for active flow control based on physics-informed neural networks, Phys. Fluids **36**, 074112 (2024).

[34] F. Ren, J. Rabault, and H. Tang, Applying deep reinforcement learning to active flow control in weakly turbulent conditions, Phys. Fluids **33**, 037121 (2021).

[35] L. Guastoni, J. Rabault, P. Schlatter, H. Azizpour, and R. Vinuesa, Deep reinforcement learning for turbulent drag reduction in channel flows, Eur. Phys. J. E **46**, 4 (2023).

[36] B. Font, F. Alcántara-Ávila, J. Rabault, R. Vinuesa, and O. Lehmkuhl, Deep reinforcement learning for active flow control in a turbulent separation bubble, Nat. Commun. **16**, 1422 (2025).

[37] H. J. Bae and P. Koumoutsakos, Scientific multi-agent reinforcement learning for wall-models of turbulent flows, Nat. Commun. **13**, 1443 (2022).

[38] A. J. Linot, K. Zeng, and M. D. Graham, Turbulence control in plane Couette flow using low-dimensional neural ode-based models and deep reinforcement learning, Int. J. Heat Fluid Flow **101**, 109139 (2023).

[39] J. Viquerat, J. Rabault, A. Kuhnle, H. Ghraieb, A. Larcher, and E. Hachem, Direct shape optimization through deep reinforcement learning, J. Comput. Phys. **4**28, 110080 (2021).

[40] J. Viquerat, R. Duvigneau, P. Meliga, A. Kuhnle, and E. Hachem, Policy-based optimization: Single-Step policy gradient method seen as an evolution strategy, in *Neural Computing and Applications* (Springer, Berlin, 2022).

[41] H. Ghraieb, J. Viquerat, A. Larcher, P. Meliga, and E. Hachem, Single-step deep reinforcement learning for two- and three-dimensional optimal shape design, AIP Adv. **12**, 085108 (2022).

[42] H. Ghraieb, J. Viquerat, A. Larcher, P. Meliga, and E. Hachem, Single-step deep reinforcement learning for open-loop control of laminar and turbulent flows, Phys. Rev. Fluids **6**, 053902 (2021).
[43] E. Moreau, Airflow control by non-thermal plasma actuators, J. Phys. D **40**, 605 (2007).
[44] S. Grundmann and C. Tropea, Active cancellation of artificially introduced Tollmien–Schlichting waves using plasma actuators, Exp. Fluids **44**, 795 (2008).
[45] P. P. C. Brito, P. Morra, A. V. G. Cavalieri, P. Jordan, D. M. Luchtenburg, D. Sipp, A. Agarwal, and A. P. L. Oliveira, Experimental control of Tollmien–Schlichting waves using pressure sensors and plasma actuators, Exp. Fluids **62**, 32 (2021).
[46] W. W. Bower, J. T. Kegelman, A. Pal, and G. H. Meyer, A numerical study of two-dimensional instability-wave control based on the Orr–Sommerfeld equation, Phys. Fluids **30**, 998 (1987).
[47] L. D. Kral and H. F. Fasel, Numerical investigation of three-dimensional active control of boundary-layer transition, AIAA J. **29**, 1407 (1991).
[48] T. Albrecht, R. Grundmann, G. Mutschke, and G. Gerbeth, On the stability of the boundary layer subject to a wall-parallel Lorentz force, Phys. Fluids **18**, 098103 (2006).
[49] P. C. Dörr and M. J. Kloker, Numerical investigations on Tollmien–Schlichting wave attenuation using plasma-actuator vortex generators, AIAA J. **56**, 1305 (2018).
[50] M. Kotsonis, S. Ghaemi, L. Veldhuis, and F. Scarano, Measurement of the body force field of plasma actuators, J. Phys. D **44**, 045204 (2011).
[51] L. M. Mack, Boundary-layer Linear Stability Theory, AGARD Report No. 709, Jet Propulsion Laboratory, California Institute of Technology, 1984.
[52] F. P. Bertolotti, Th Herbert, and P. R. Spalart, Linear and nonlinear stability of the Blasius boundary layer, J. Fluid Mech. **242**, 441 (1992).
[53] M. Kotsonis, Dielectric Barrier Discharge Actuators for Flow Control: Diagnostics, Modeling, Application, Ph.D. thesis, TU Delft, The Netherlands, 2012.
[54] R. Rebonato and P. Jäckel, The most general methodology to create a valid correlation matrix for risk management and option pricing purposes, SSRN working paper (2011), doi:10.2139/ssrn.1969689.
[55] K. Numpacharoen and A. Atsawarungruangkit, Generating correlation matrices based on the boundaries of their coefficients, PLoS One, **7**, e48902 (2012).
[56] D. P. Kingma and J. L. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.
[57] C. H. Hansen, *Understanding Active Noise Cancellation* (Spon Press, London, 2001).
[58] S. M. Kuo and D. R. Moran, *Active Noise Control Systems* (John Wiley & Sons, New York, 1996).
[59] J. Viquerat, Policy-based optimization: Single-step policy gradient seen as an evolution strategy (2022), https://github.com/jviquerat/pbo.
[60] H. Schlichting, E. Krause, H. Oertel, K. Gersten, and C. Mayes, *Boundary-Layer Theory* (Springer, Berlin, 2003).