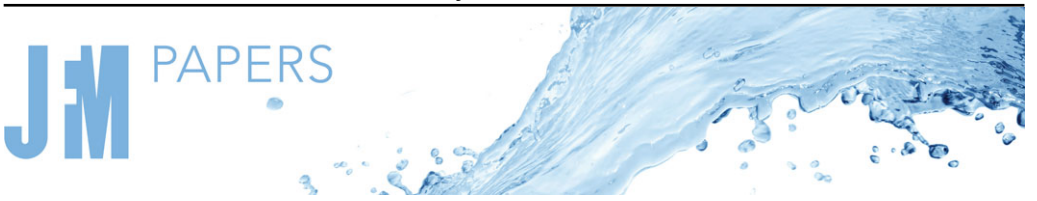
J. Fluid Mech. (2025), *vol.* 1009, A3, doi:10.1017/jfm.2025.160



Closed-loop supersonic flow control with a high-speed experimental deep reinforcement learning framework

Haohua Zong[1], **Yun Wu**[1], **Jinping Li**[2], **Zhi Su**[2] **and Hua Liang**[2]

[1]School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, PR China

[2]National Key Lab of Aerospace Power System and Plasma Technology, Air Force Engineering University, Xi'an 710038, PR China

Corresponding author: Haohua Zong, haohua_zong@163.com

(Received 6 September 2024; revised 11 December 2024; accepted 30 January 2025)

Although active flow control based on deep reinforcement learning (DRL) has been demonstrated extensively in numerical environments, practical implementation of real-time DRL control in experiments remains challenging, largely because of the critical time requirement imposed on data acquisition and neural-network computation. In this study, a high-speed field-programmable gate array (FPGA) -based experimental DRL (FeDRL) control framework is developed, capable of achieving a control frequency of 1–10 kHz, two orders higher than that of the existing CPU-based framework (10 Hz). The feasibility of the FeDRL framework is tested in a rather challenging case of supersonic backward-facing step flow at Mach 2, with an array of plasma synthetic jets and a hot-wire acting as the actuator and sensor, respectively. The closed-loop control law is represented by a radial basis function network and optimised by a classical value-based algorithm (i.e. deep Q-network). Results show that, with only ten seconds of training, the agent is able to find a satisfying control law that increases the mixing in the shear layer by 21.2 %. Such a high training efficiency has never been reported in previous experiments (typical time cost: hours).

Key words: supersonic flow, mixing enhancement, jets

1. Introduction

Active flow control (AFC) is a technique that leverages actuators to manipulate the external flow field in a way that benefits the overall aerodynamic performance of a target (Gad-el Hak 2000; Cattafesta III & Sheplak 2011). From the perspective of control theory, the essential goal of AFC is to find an optimal control law that maximises the value of a

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/jfm.2025.160 Published online by Cambridge University Press

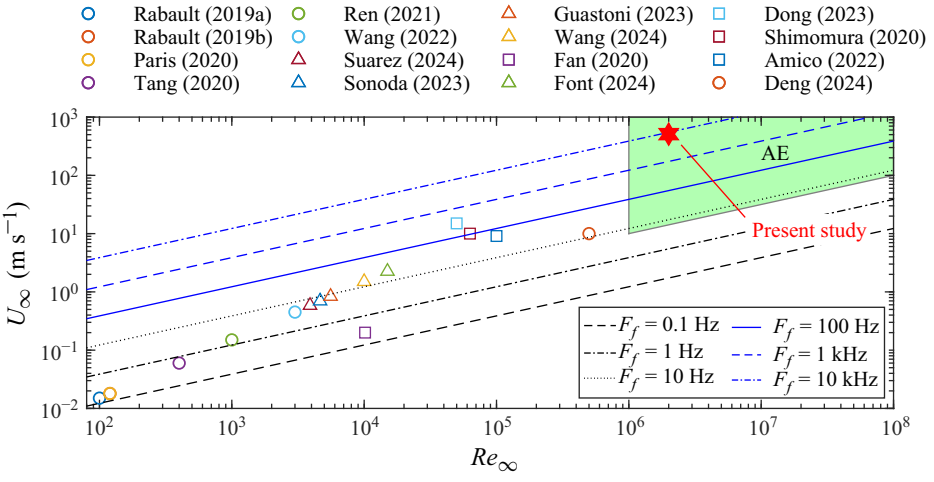


Figure 1. Summary of recent DRL control studies. The green zone indicates the parameter range relevant to aerospace engineering (AE).Here, F_f denotes the characteristics frequency of the flowand ○, △ and □ represent 2-D numerical studies, 3-D numerical studies and experimental studies, respectively.

target function (Brunton, Noack & Koumoutsakos 2020). Traditional model-based control strategies, although promising, rely on reduced-order modelling of the flow and can only be applied to simple geometries such as the bluff body flow and cylinder flow (Pastoor *et al.* 2008). For closed-loop control of complex flows, latest data-driven machine learning approaches represented by deep reinforcement learning (DRL), attract significant attention from the community (Vignon, Rabault & Vinuesa 2023). In figure 1, relevant studies are plotted in a map spanned by the free-stream velocity (U_∞) and Reynolds number (Re_∞). Note that, for numerical studies where only the Reynolds number is reported, the free-stream velocity is derived by assuming a reference length scale of $L_{ref} = 0.1$ m. Namely, $U_\infty = Re_\infty \nu / L_{ref}$, where ν denotes the kinematic viscosity. This assumption of the length scale is reasonable for most of the laboratory models.

Rabault *et al.* (2019), using a two-dimensional (2-D) low-Reynolds-number numerical environment, demonstrated for the first time that a DRL agent could learn an effective vortex-valley stabilisation strategy in cylinder flow, leading to a drag reduction of $DR = 8\,\%$ at a free-stream Reynolds number of $Re_\infty = 100$. Subsequently, this numerical framework was referenced by other groups, and extensive efforts were made to examine the impacts of hyper-parameters (Rabault & Kuhnle 2019; Sonoda *et al.* 2023), accelerate the training speed of the agent (Wang *et al.* 2022; Deng, Hu & Chen 2024; Suárez *et al.* 2024), improve the robustness of learnt control strategies (Ren, Rabault & Tang 2021; Tang *et al.* 2020; Jia & Xu 2024) and optimise the sensor placement (Paris, Beneddine & Dandois 2021; Wang *et al.* 2024). Although valuable instructions are obtained by these low-Reynolds-number studies (mostly two-dimensional at $Re_\infty < 1000$), practical implementations of DRL in experiments remain challenging, particularly for high-speed high-Reynolds-number aeronautical flows. This difficulty mainly arises from the critical time requirement imposed to the control law execution and neural-network training. Specifically, to couple with the flow dynamics, the control should be performed on the characteristic scale of the dominant flow structures. Under this assumption, the control loop frequency (denoted as F_c) should be at least several times higher than the characteristic frequency of the flow (denoted as F_f) (Rabault & Kuhnle 2019).

https://doi.org/10.1017/jfm.2025.160 Published online by Cambridge University Press

Namely

$$F_c \geqslant F_f = \frac{U_\infty}{L_{ref}} = \frac{U_\infty^2}{Re_\infty \nu}. \tag{1.1}$$

For aerospace engineering (AE) applications where U_∞ and Re_∞ range typically in 10–1000 m s^{-1} and 10^6-10^8, the characteristic frequency of the flow is estimated to be 100 Hz–10 kHz. This, in combination with (1.1), indicates that the control frequency in an experimental DRL system should at least exceed 1 kHz to be truly meaningful. However, the reality is that, almost all of the existing experimental DRL studies (Fan *et al.* 2020; Shimomura *et al.* 2020; Amico, Cafiero & Iuso 2022; Dong *et al.* 2023) used CPU as the main controller, taking charge of both the execution and training of the neural networks. In such a framework, the sensor signals and the control commands are stored in the buffer of a data acquisition card, before transmitting to RAM and being processed by CPU. The corresponding time delay caused by this buffer mechanism can easily reach O(10 ms), which practically limits the maximum control frequency to below 100 Hz, far away from AE applications. Actually, in Fan *et al.* (2020) and Shimomura *et al.* (2020), F_c is only 10 Hz. Apart from the above-mentioned real-time execution challenge, in supersonic/hypersonic experiments, timely training and fast convergence of the neural network are also a crucial concern, since most of these wind tunnels operate in transient mode with a typical test time of O(10 s).

Driven by the above need, a novel high-speed experimental DRL control framework is proposed, using field-programmable gate array (FPGA) as the main controller. The maximum control frequency of this framework reaches O(1–10 kHz), two orders higher than conventional CPU-based implementation. The feasibility of this framework is tested in a supersonic backward-facing step (BFS) flow, which is rather challenging since the entire DRL process has to be completed in one run of the wind tunnel (10 s). The paper is organised as follows. Section 2 details the control methodology and framework. In § 3, the experimental set-up, including the flow environment and measurement systems, is introduced. In § 4, the DRL learning curves and high-speed schlieren results are presented. Major findings are summarised in § 5.

2. High-speed experimental DRL control framework

Based on the review inthe previous section, it becomes clear that, to realise real-time closed-loop control in high-speed experiments, the key is to cut down the control period. In DRL, the minimum control period a hardware system can implement reads as

$$T_c = T_{acq} + T_{nn} + T_{out} + T_{com}, \tag{2.1}$$

where T_{acq}, T_{nn}, T_{out} and T_{com} represent, respectively, the time delays caused by data acquisition, neural-network computation, control command output and necessary hardware communication.

To minimise each of the four terms in § 2.1, an FPGA-based experimental DRL (FeDRL) framework is proposed, as shown in § 2. Compared with the traditional CPU-based DRL framework (Fan *et al.* 2020; Shimomura *et al.* 2020), three modifications are made. First, the execution and the training processes of the neural network are effectively split, finished respectively by a time-critical FPGA chip and a high-performance CPU chip. Second, the original fully connected neural network (FCNN) used to represent the policy and the Q-function, is substituted by a radial basis function (RBF) network. For universal approximation, the RBF network requires only one hidden layer, thus running and training much faster than a conventional FCNN. Third, the data acquisition

https://doi.org/10.1017/jfm.2025.160 Published online by Cambridge University Press

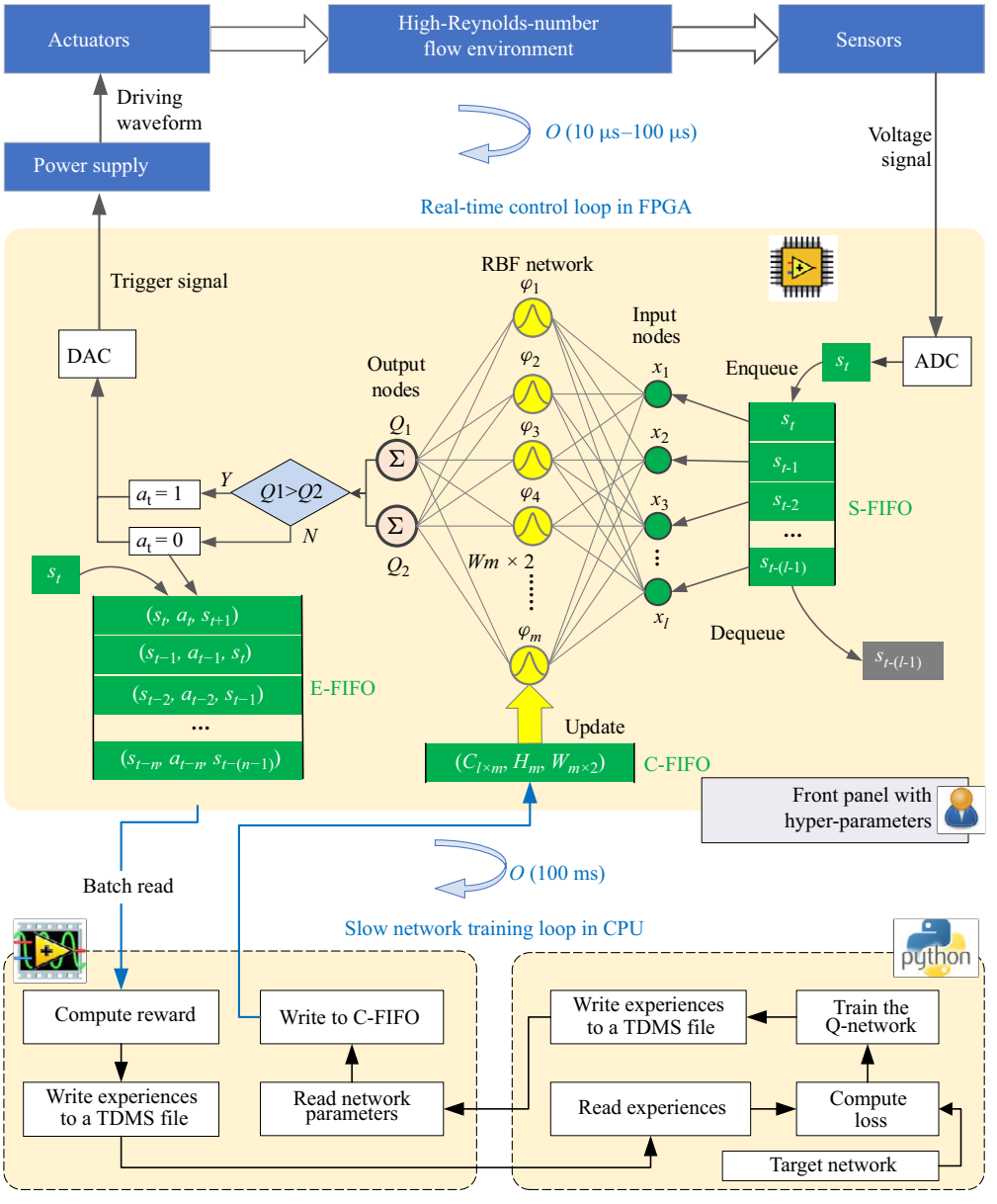


Figure 2. High-speed experimental DRL framework powered by FPGA and RBF network.

(DAQ) devices including an analogue-to-digital converter (ADC) and a digital-to-analogue converter (DAC) necessary for reading the sensor signal and outputting the command signal are hooked directly on the FPGA chip. Thus, no buffer mechanism is required, and the communication time delay (T_{com}) is eliminated. As a result, the highest control frequency (minimum control period) that can be reached by this novel framework equals to $F_{c,max} = 1/(T_{acq} + T_{nn} + T_{out})$. Using high-end FPGA chips and DAQ modules, we can easily push $F_{c,max}$ to O(1–10 kHz).

To illustrate the basic workflow of this framework, a classical value-based DRL algorithm, namely the deep Q-network (DQN), is implemented. As show in figure 2, two

https://doi.org/10.1017/jfm.2025.160 Published online by Cambridge University Press

loops are configured in the FeDRL framework: a real-time control loop with a period of O(10–100 μs), and a slow network training loop running once every hundreds of ms. In each control period, the sensor signal, namely the instantaneous flow state (s_t), is first sampled by an ADC and then inserted into a first-in first-out (FIFO) sequence (denoted as S-FIFO, "S" for state). This sequence (length: l), including the current state and its time histories, will be used as the input of the RBF network, namely $\mathbf{x} = [s_t, s_{t-1}, s_{t-2}, \ldots, s_{t-l+1}]$. Previous DRL control studies have shown that, for situations where the observation of a flow system is sparse in space, such an augmentation of the flow states can help the agent find better control strategies, close to that in full observation (Pino *et al.* 2023; Wang *et al.* 2024). The outputs of the network are essentially the Q-values of different actions. In figure 2, a simplest case is shown, with only two actions indicating the ON/OFF status of the actuator. Mathematically, the Q-functions represented by the network are as follows:

$$\begin{cases} \mathbf{Q_{2\times1}} = \mathbf{W_{2\times m}} \cdot [\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_m(\mathbf{x})]^{\mathbf{T}} + \mathbf{B_{2\times1}} \\ \phi_i = \exp(-h_i \, ||\mathbf{x} - \mathbf{c_i}||^2), \qquad \text{for } i = [1, 2, \ldots, m] \end{cases} \tag{2.2}$$

where ϕ_i is the basis function, chosen as a Gaussian kernel. The centre and variance of ϕ_i are denoted as $\mathbf{c_i}$ and h_i, respectively. In matrix form, $\mathbf{C_{l\times m}} = [\mathbf{c_1^T}, \mathbf{c_2^T}, \ldots, \mathbf{c_m^T}]^{\mathrm{T}}$ and $\mathbf{H_{m\times1}} = [h_1, h_2, \ldots, h_m]$. The weights connecting the hidden layer to the output layer and the biases of the output layer are presented by $\mathbf{W_{2\times m}}$ and $\mathbf{B_{2\times1}}$, respectively.

Using an epsilon-greedy strategy, an action a_t can be selected from the Q-function, which is then converted into the driving waveform of the actuator via an DAC and a power supply. For training purposes, the experiences collected in each control period are organised as a tuple (s_t, a_t, s_{t+1}) and enqueued into an experience-FIFO (abbreviated as E-FIFO, length:n). In the training loop, a Labview program is set up, which takes advantage of its NI-FPGA module, acting as a bridge between the controller (FPGA code) and the trainer (Python code). To avoid frequent data requests, the experiences stored in the E-FIFO are read by the Labview program in batch mode. After that, the rewards (r_t) are assigned and appended into the experience tuple as an extra column. These experiences are transferred to the Python program via a technical data management streaming (TDMS) file, based on which the prediction losses are computed, and the RBF network together with its target is updated. To keep the control law up to date, the weights of the updated network are first sent back to the Labview program via another TDMS file, and then written to the coefficient-FIFO (C-FIFO) of the FPGA.

3. Practical implementation in supersonic BFS flow

3.1. *Experimental set-up*

The supersonic BFS flow, which can be found in the flame holder of scramjets and the tail of rockets, is chosen to test the FeDRL framework, with an aim of mixing enhancement in the shear layer. Experiments are conducted in the supersonic wind tunnel at the Air Force Engineering University. It is a transient suction-type wind tunnel, driven by a 200 m^3 vacuum chamber. The free-stream wind speed is Mach 2 (U_∞ = 510 m s^{-1}), and the maximum test time in one run is approximately 10 s. This wind tunnel has a square test section (200 mm × 200 mm), providing optical access from the top and side windows. As shown in figure 3, the BFS model is made of a flat plate (material: Polyetheretherketone) with a sharp leading edge. The plate width, thickness and step height (denoted as H) are 80 mm, 15 mm and 10 mm, respectively. The distance between the step corner and the leading edge is 190 mm. The Reynolds number based on the free-stream velocity and

https://doi.org/10.1017/jfm.2025.160 Published online by Cambridge University Press

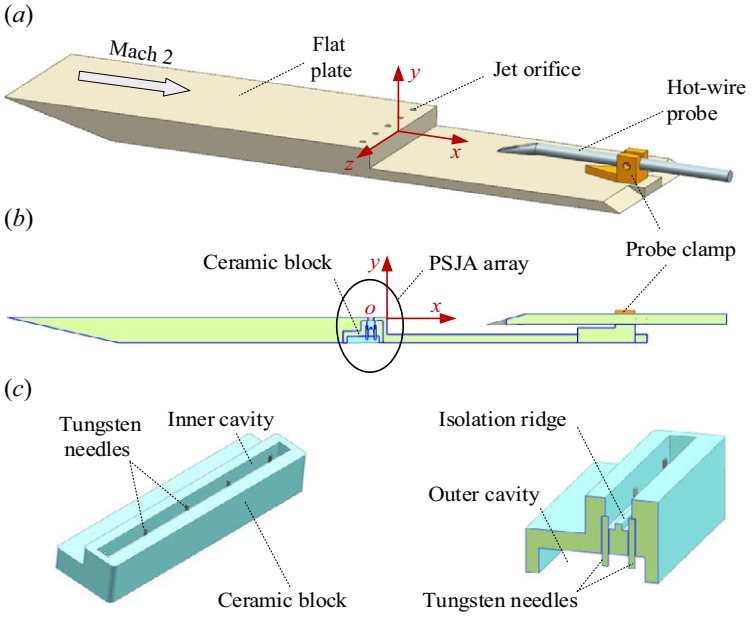


Figure 3. (*a*) Three-dimensional view of the BFS model. (*b*) Cross-sectional view in the xy-plane. (*c*) Structure of the ceramic block.

plate length reaches $Re_\infty = 2 \times 10^6$. A coordinate system is established at the middle of the step corner, with x, y and z along the streamwise, wall-normal and spanwise directions, respectively. A hot-wire probe is clamped in the far downstream of the step, acting as the sensor and measuring the velocity fluctuations in the shear layer. This probe is specially designed for supersonic flow, with a wedge front and a stiff steel body. The diameter of the wire is 6.5 μm, corresponding to a frequency response of more than 10 kHz. The hot-wire probe is connected to a constant-temperature anemometry sensor (CTA-4, Hanghua Technology). The operating temperature of the wire (T_{wire}) is set as 410 K, which translates to an overheat ratio of 2.5 in the cold supersonic flow.

To disturb the supersonic BFS, a spanwise array of five plasma synthetic jet actuators (PSJAs) are installed in front of the step corner, as show in figure 3(*b*). This actuator is selected due to the unique combination of high jet velocity ($O(100\,\mathrm{m\ s^{-1}})$) and high frequency (>5 kHz). Unlike previous designs, where independent shells and cavities are created for each actuator (Zong & Kotsonis 2017; Zong *et al.* 2024), an integrated ceramic block is designed in this study which hosts five pairs of tungsten needles and a slot internal cavity shared by all the actuators (see figure 3*c*). This ceramic block has a L-shape cross-section, embedded into the flat plate from the bottom. The corresponding jet orifices are created on the top surface of the flat plate. The jet orifice diameter (D), spanwise spacing and distance to the step corner are 3 mm, 15 mm and 8 mm, respectively. The width, height and length of the internal cavity are 4 mm, 5 mm and 66 mm, respectively, resulting in total cavity volume of $V_{ca} = 1320\ \mathrm{mm}^3$.

As shown in figure 4(*a*), the sequential discharge circuit developed by Zong & Kotsonis (2017) is used to feed the PSJA array. All the plasma actuators are connected in serial, such that the same discharge current is shared. In each period, a high-voltage trigger pulse is first produced by a nano second (NS) power supply to create air breakdown in between the electrodes. Then, the electrical energy stored in the capacitor C1 is rapidly

https://doi.org/10.1017/jfm.2025.160 Published online by Cambridge University Press

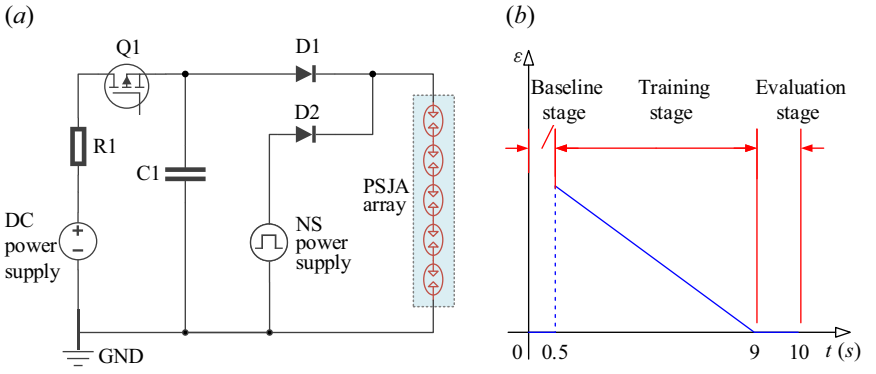


Figure 4. (*a*) The sequential discharge circuit used to feed the PSJA array. (*b*) Sketch of the three test stages in one run of the wind tunnel.

deposited into the actuator cavity, by virtue of arc heating. R1 (500 Ω) is used to limit the charging current of the DC power supply, and the capacitance and DC voltage are set as C_0 =0.1 μF and $U_0 = 1$ kV, respectively. Two high-voltage diodes (D1 and D2) are added for isolation purposes. Different from the original design in Zong & Kotsonis (2017), a high-voltage electronic switch (Q1) is inserted between R1 and D1. During the capacitive discharge stage, Q1 is switched off to prevent current directly flowing from the DC power supply to the PSJA array. Based on the relevant electrical and geometrical parameters, the dimensionless energy deposition (λ), defined as the ratio of capacitor energy (E_c) to the total enthalpy of the cavity gas (E_g), can be calculated

$$\lambda = \frac{E_c}{E_g} = \frac{0.5 C_0 U_0^2}{c_p \rho_0 V_{ca} T_0}, \tag{3.1}$$

where c_p is the constant-pressure specific heat capacity; ρ_0 and T_0 denote the static density (0.23 kg m^{-3}) and temperature (163 K) in the test section, respectively. Consequently, λ is determined as 10, twice higher than that in Zong & Kotsonis (2018). According to Zong *et al.* (2018), the maximum jet velocity (U_p) scales with the cubic root of λ. Therefore, the value of U_p in the present study is estimated as $2^{1/3}$ times of that in Zong & Kotsonis (2018), namely 146 m s^{-1}.

3.2. *The DRL settings*

Following the framework sketched in figure 2, a closed-loop DRL control system is set up. This system consists of an NI real-time controller which has a built-in FPGA chip (National Instruments, cRIO-9049), a hot-wire sensor and a PSJA array. Two modules are hooked in the chassis of the controller: an analogue input module (NI-9223) for acquiring the hot-wire voltage signal, and an analogue output module (NI-9262) for sending trigger pulses to the actuator power supply. The sampling rate of the ADC device is set as 50 kHz, resulting in a DAQ period of $T_{acq} = 20$ μs. The depth of the S-FIFO is set as $l = 10$, which determines the maximum number of input nodes in the RBF network. The PSJA array is controlled in an ON/OFF manner ($a_t = 0$ or 1). If $a_t = 1$, a trigger pulse with a width of 5 μs is sent to the high-voltage power supply, which then fires one shot of plasma jet at the flow. Limited by the maximum working frequency of the PSJA (5.2 kHz; see Zong & Kotsonis 2018), the control loop frequency is selected as 5 kHz. Additionally, considering that the input nodes of RBF network (l) are no more than 10, the number of hidden neurons is set as $\mathbf{m} = 20$.

https://doi.org/10.1017/jfm.2025.160 Published online by Cambridge University Press

Ideally, we would use the velocity fluctuation in a control period as the instant reward. Practically, due to the compressibility effect, the variations of different flow quantities in the supersonic regime are coupled, and it is impractical to obtain an accurate velocity–voltage calibration curve as that in incompressible subsonic flow. Therefore, we directly use the standard deviation of the normalised voltage signal ($\bar{s}_t$) within the subsequent l steps as a metric to surrogate the instant reward, namely

$$\begin{cases} r_t = \sqrt{\dfrac{1}{l}\displaystyle\sum_{i=1}^{i=l} \bar{s}_{t+i}^2} - 1 \\ \bar{s}_t = (s_t - \mu_0)/\sigma_0, \end{cases} \tag{3.2}$$

where the mean and standard deviation of the undisturbed voltage signal (μ_0 and σ_0) measured in the baseline condition are used for normalisation. By definition, in the first expression, the first term on the right-hand side has an expectation of 1 when no plasma actuation is applied. Therefore, by subtracting 1 from the expression, a positive (negative) reward would correspond to a increase (decrease) in the measured flow fluctuation.

The Keras API is employed to train the network, and the training loop frequency is 2 Hz. Namely, for every 500 ms, the 2500 fresh experiences collected by the FPGA are piled into a replay buffer (size: 10 000), based on which the RBF network is trained approximately for two episodes. Since we use the classical DQN algorithm, the Bellman equation is used iteratively to update the Q-function. The loss function (L) is defined as follows (Mnih *et al.* 2015):

$$L(\boldsymbol{\theta}) = \mathbb{E}\left[\left(r_t + \gamma \cdot \max_{a'_{t+1}} \mathbf{Q}^*(\mathbf{x}_{t+1}, a'_{t+1}) - \mathbf{Q}(\mathbf{x}_t, a_t)\right)^2\right], \tag{3.3}$$

where γ is the discount factor (0.95), influencing the horizon of the agent; θ represents the trainable parameters in the RBF network, including the centre matrix $\mathbf{C}$, variance matrix $\mathbf{H}$, weight matrix $\mathbf{W}$ and bias matrix $\mathbf{B}$. On the right-hand side of (3.3), the sum of the first two terms defines the expected Q-value, while the last term gives the predicated Q-value. Obviously, the expectation of their squared difference is essentially the loss. Note that a separate target network ($\mathbf{Q}^*$) is used to calculate the expected Q-value, and the parameters of the original Q-network are copied to the target Q-network once every five training episodes.

In each training episode, a mini-batch of 1000 experiences is sampled randomly from the replay buffer to compute the loss, and the network parameters are updated according to the following equation:

$$\theta^{i+1} = \theta^i - \alpha \cdot \nabla_{\theta^i} L(\theta^i), \tag{3.4}$$

where the superscripts 'i' and '$i+1$' represent the iteration number. Here, α denotes the learning rate, which is initialised as 0.001 and adjusted dynamically via the Adam optimiser during the course of training (Kingma 2014).

To achieve balanced exploration and exploitation in reinforcement learning (RL), the probability of choosing to explore in the epsilon-greedy strategy, namely the value of ε, is assumed as a piecewise function of time (see (3.5)). As sketched in figure 4(*b*), in each run of the wind tunnel, the complete operation time (10 s) is divided into three stages. The first stage ($t \leqslant 0.5$ s) corresponds to the baseline test without plasma actuation ($\varepsilon = 0,\ a_t = 0$). The second stage ($0.5\ \text{s} < t < 9$ s) denotes the training stage with the exploration probability decreasing linearly from 0.5 to 0. The last stage is the evaluation

https://doi.org/10.1017/jfm.2025.160 Published online by Cambridge University Press

No.	State vector $\mathbf{x}$	$\mathbf{C_{l\times m}}$	$\mathbf{H_{m\times 1}}$	$\mathbf{W_{2\times m}}$	$\mathbf{B_{2\times 1}}$
Case 1	$[s_t, s_{t-1}, \ldots, s_{t-9}]$	Random uniform	1	Xavier uniform	0
Case 2	$[s_t, s_{t-1}]$	Random uniform	1	Xavier uniform	0
Case 3	$[s_t, s_{t-1}, \ldots, s_{t-9}]$	K-means clustering	1	Xavier uniform	0

Table 1. List of test cases and initialisation methods for different parameters.

stage ($t \geqslant 9$ s) where the final control strategy is executed deterministically without exploration, namely $a_t = \mathrm{argmax}\{\mathbf{Q}(\mathbf{x_t}, a_t)\}$. The authors have compared the statistics of learnt strategies computed from this post-training evaluation scheme and that from an independent evaluation run, and marginal differences can be observed. This post-training evaluation scheme allows us to obtain the baseline and actuated statistics simultaneously in one run, eliminating the possible measurement error caused by the drift of wind tunnel operating condition (e.g. ambient temperature).

$$\varepsilon = \begin{cases} 0, & t \leqslant 0.5 \text{ s} \\ 0.5 \times (1 - (t - 0.5)/8.5), & 0.5 \text{ s} < t < 9 \text{ s} \\ 0, & t \geqslant 9 \text{ s}. \end{cases} \tag{3.5}$$

4. Results and discussion

4.1. *Velocity statistics*

In total, three cases with different state dimensionalities and initialisation strategies are selected to perform DRL control, as listed in table 1. In case 1, the input vector of the Q-network is set as $\mathbf{x} = [s_t, s_{t-1}, \ldots, s_{t-9}]$. The Gaussian kernel centre matrix $\mathbf{C_{l\times m}}$ and the weight matrix $\mathbf{W_{2\times m}}$ are initialised by a random uniform initialiser (range: [0, 1]) and an Xavier uniform initialiser in the Keras API, respectively. The variance matrix $\mathbf{H_{m\times 1}}$ and the bias matrix $\mathbf{B_{2\times 1}}$ are directly assigned as 1 and 0, respectively. Case 2 shares the same initialisation strategy as case 1. Nevertheless, the network input is truncated to $\mathbf{x} = [s_t, s_{t-1}]$, for the purpose of comparing the impact of state dimensionality. For case 3, the same input vector as case 1 is used. Inspired by Maceda *et al.* (2023), where the feedback control law in cavity flow can be well interpreted by a cluster-based visualisation, we decided to initialise the Gaussian kernel centre matrix $\mathbf{C_{l\times m}}$ with the K-means clustering method. Specifically, the input state data acquired under the baseline condition (namely the 2500 samples of $\mathbf{x}$ acquired at $t \leqslant 0.5$ s) are divided into m clusters ($m = 20$), resulting in a total of m cluster centroids. Since each centroid is a vector consisting of l elements ($l = 10$); we have $l \times m$ elements, ready to be assigned to the Gaussian kernel centre matrix $\mathbf{C_{l\times m}}$ in one-to-one correspondence. By doing so, we hope that the Q-network can leverage some of the existing knowledge in baseline flow and converge faster in training. Other parameters in case 3, including the variance matrix $\mathbf{H_{m\times 1}}$, the weight matrix $\mathbf{W_{2\times m}}$ and the bias matrix $\mathbf{B_{2\times 1}}$, are initialised with the same strategy as that in case 1.

Figure 5(*a*) shows the time evolution of the low-pass filtered reward (dented as $\overline{r_t}$, filter cutoff frequency: 50 Hz), together with the history of network loss in different cases. Regardless of the initialisation strategy and state dimensionality, cases 1–3 show a similar increasing trend of the reward. The most noteworthy feature is that the DQN agent is so efficient that, even with only ten seconds of learning, the filtered reward, signifying the relative increase of velocity fluctuation, is increased from 0 to a peak value

https://doi.org/10.1017/jfm.2025.160 Published online by Cambridge University Press

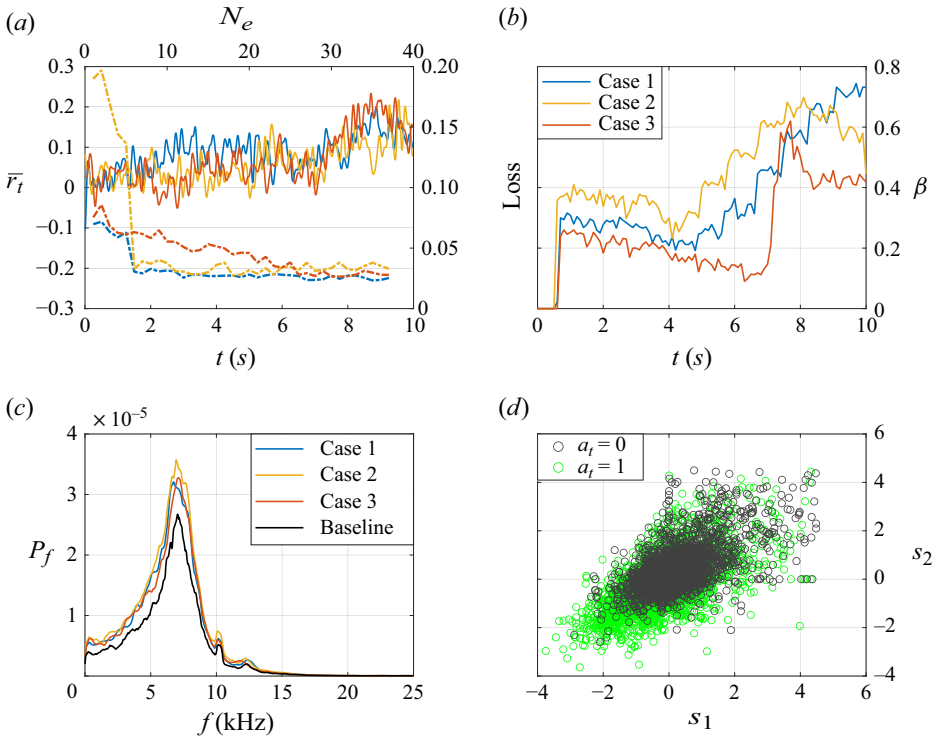


Figure 5. (*a*) Time evolution of the filtered rewards ($\overline{r_t}$, solid lines) and the episodic variation of the network loss (dashed lines). (*b*) Variation of the activation ratio. (*c*) Power spectral densities of the hot-wire anemometry (HWA) voltage. (*d*) Classification of the states in case 2 based on the action.

of 0.2. The number of training episodes spent in this process is one order less than the policy-based DRL algorithm (Rabault *et al.* 2019). In the range 0.5 s $\leqslant t \leqslant$ 1 s, the DQN agent has a high exploration probability of close to 0.5, and random actuation is performed half of the time. Consequently, little improvement of the instant reward can be picked when compared with the baseline stage of $t \leqslant$0.5 s, indicating that random actuation is inefficient in promoting the mixing. During the course of learning, the network loss varies approximately inversely with the reward. Specifically, in the initial stage of $t \leqslant 1.5$ s, a sharp drop of the training loss is observed in cases 1–2, followed by a slow decay afterwards. Comparing case 3 with case 1, the cluster-based initialisation method, for which the authors have high expectations, exhibits a slower decay rate of the training loss than the random initialisation method, possibly due to the different distributions of the states in baseline and actuated conditions.

The activation ratio of the plasma actuator (β), defined as the occurrence probability of $a_t = 1$ in a short period of time, is plotted in figure 5(*b*). In the baseline test stage of $t < 0.5$ s, the activation ratio is kept unchanged at zero. When stepping into the training stage ($t = 0.5$ s), different activation ratios ranging from 0.2 to 0.4 are exhibited in the three cases. This is reasonable, since this initial value of β depends on both the exploration probability (ε) and the initial parameters of the Q-network. As sketched in figure 4(*b*), ε is initialised as 0.5. Therefore, random control is applied only for half of the time. During the other half, deterministic action is chosen according to the blind prediction of the Q-network. Consequently, the initial activation ratio can vary between 0.25 and 0.75. For

https://doi.org/10.1017/jfm.2025.160 Published online by Cambridge University Press

the three cases tested in this study, even though the same seed is used by the pseudo-random generator to initialise the Q-network, the initial activation ratio still differs due to the different network input dimensionality and initialisation strategy (see table 1). In the first half-course of learning ($t < 5$ s), the agent collects experience by shooting plasma randomly, and the activation ratio goes down due to the decreasing exploration probability. By contrast, in the last half, the agent has gained sufficient experience, and starts to taste the fruit of deterministic actuation, leading to a fast growth in the activation ratio. The average values of β in the evaluation stage are 0.72, 0.57 and 0.42, respectively, for cases 1–3, corresponding to an equivalent discharge frequency of 3.6 kHz, 2.9 kHz and 2.1 kHz, respectively.

Based on the velocity signals collected in the evaluation stage ($9\text{ s} \leqslant t \leqslant 10$ s), the power spectra are computed, as shown in figure 5(*c*). For all of the cases, a sharp peak can be found at $f = 7$ kHz. Compared with the baseline, DRL control is able to lift the power spectrum as a whole, and the relative increases of the velocity fluctuation in cases 1–3 are 15.6 %, 21.2 % and 14.4 %, respectively. In the best case (case 2), since we only have two state variables, it is possible to visualise the control strategy with a scatter plot (see figure 5*d*). Consequently, the joint distribution of s_1 and s_2 exhibits an elliptical shape. This is reasonable, since the dimensionless velocities measured at two neighbouring time steps are correlated, and should not differ too much from each other. Active states with $a_t = 1$ are mostly found in the third quadrant, while inactive states with $a_t = 0$ tend to be distributed in the first quadrant. Such a control strategy can be further simplified as a 2-D threshold actuation strategy. Physically, since the instantaneous velocity fluctuation measured by the hot-wire is correlated with the flapping motion of the shear layer, the above threshold actuation strategy indicates that plasma synthetic jets should be shot when the instantaneous wall-normal location of the shear layer is higher than the baseline mean location (i.e. large flow separation area behind the step).

4.2. *Schlieren imaging*

To understand, physically, how plasma synthetic jets modify the supersonic BFS flow, high-speed schlieren imaging is conducted, alongside the training process. The sampling frequency and spatial resolution are set as 50 kHz and 512×384 pixels^2, respectively. More information about the set-up can be found in Kong *et al.* (2022). Figure 6(*a*)–(*d*) shows the response of the BFS flow to a single shot of plasma synthetic jet (PSJ), where t denotes the time delay with respect to discharge ignition. Dominant features, including the expansion fan, shear layer and reattachment shock wave, can be clearly identified. Between $t = 20$ µs and $t = 80$ µs, two shock waves are issued from the exit orifice (see the black arrows), due to the rapid arc heating within the cavity. While interacting with the supersonic cross-flow, they evolve into a bow shape and propagate to the top-right corner along the trajectory specified by the expansion fan. Theoretically, between the two bow shock waves, a jet flow should be present. However, such a jet flow is not observable in the schlieren image, probably being buried by the turbulent boundary layer. The root of the reattachment shock seems to be swept downstream by plasma jets, and the reattachment length is estimated as $4H$ based on a linear extrapolation of the reattachment shock.

The standard deviation of the schlieren grey scale images (denoted as I_{std}), to some extent, can reflect the density fluctuations in the flow field. Figure 6(*e*) shows the grey scale fluctuation amplitude in baseline condition. Strong density fluctuations can be picked in the incoming boundary layer, shear layer and reattachment shock zone, which is largely expected. Figure 6(f–h) further compares the variation of grey scale fluctuation amplitude caused by plasma actuation (denoted as ΔI_{std}) in cases 1–3. Note that, to compute the

https://doi.org/10.1017/jfm.2025.160 Published online by Cambridge University Press

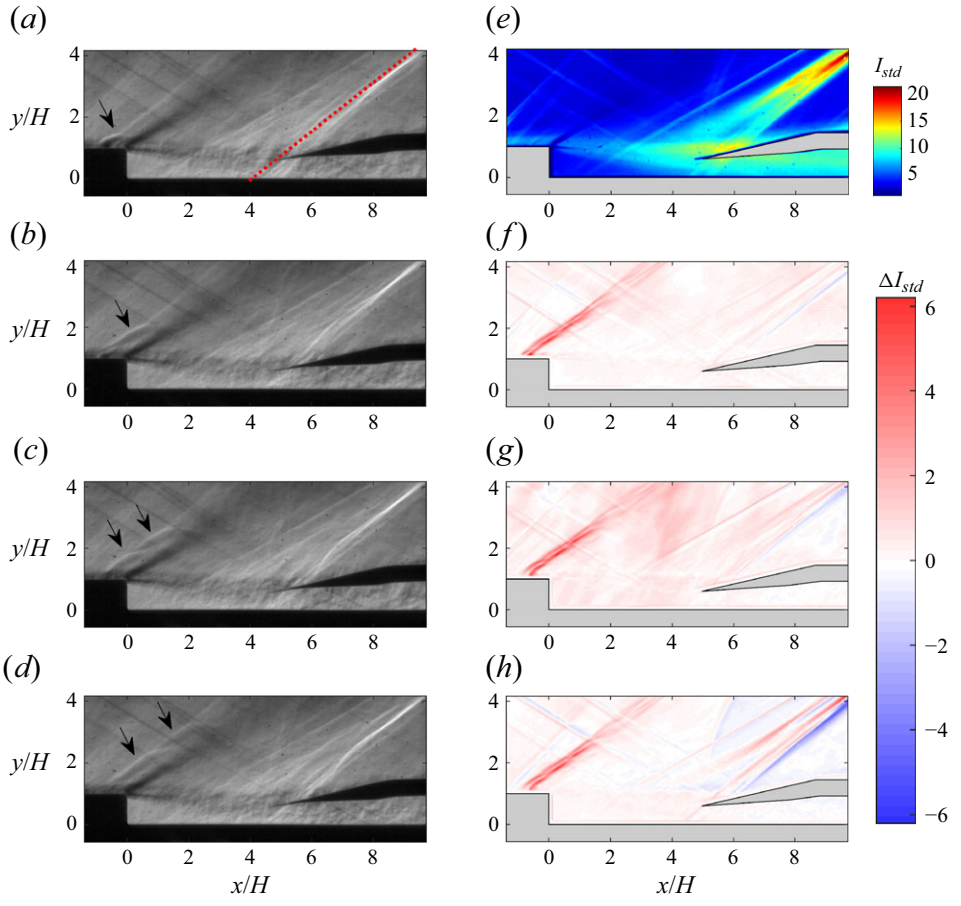


Figure 6. (*a*–*d*) Time response of the BFS flow to a single shot of plasma synthetic jet. From top to bottom, $t = 20\ \mu$s, 40 μs, 60 μs and 80 μs. The red dotted line indicates the reattachment shock. (*e*) Contour of the grey scale fluctuation amplitude in the baseline condition. (*f*–*h*) Variation of the grey scale fluctuation amplitude caused by plasma actuation in cases 1–3.

standard deviation in actuated cases, only the schlieren images acquired in the last 0.2 s (10 000 samples) are used, corresponding to the final control law sought by DRL. As a result, the grey scale fluctuation amplitude is elevated noticeably in the vicinity of the exit orifice, due to the formation of a bow shock. With regard to the shear layer zone, although the velocity fluctuation is elevated noticeably by plasma actuation, the grey scale fluctuation amplitude remains almost the same as that in the baseline.

4.3. *Strategy comparison*

In previous studies of subsonic BFS flow control, plasma actuators are operated ubiquitously in constant-frequency periodic mode (Benard *et al.* 2016*a*,*b*). By tuning the duty cycle and discharge frequency (denoted as F_d) of the plasma actuator, satisfying reduction of the reattachment length was obtained at $U_\infty = 15$ m s^{-1} by exciting the unsteady shear layer. As such, it is natural for us to compare this periodical control strategy with DRL control in the supersonic BFS flow. While implementing the constant-frequency periodical control strategy, the same experimental set-up as that described in § 3 is used, except that the RBF network in FPGA is replaced by a pulse generation module and

https://doi.org/10.1017/jfm.2025.160 Published online by Cambridge University Press

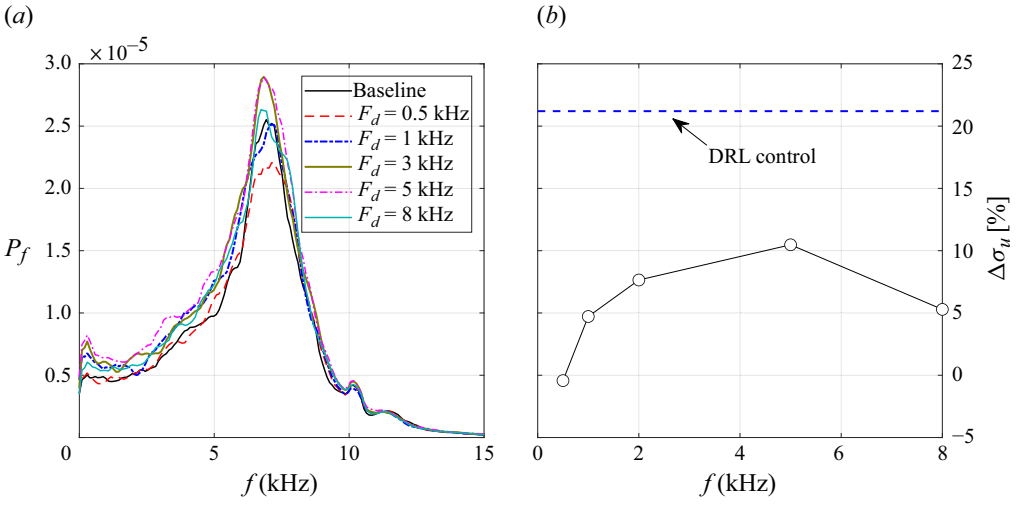


Figure 7. (*a*) Power spectral densities of the HWA voltage and (*b*) relative increase of the HWA voltage fluctuation at different discharge frequencies. The dashed blue line in (*b*) indicates the relative increase of voltage fluctuation under the optimal DRL control (case 2).

the network training is disabled. In total, five discharge frequencies ($F_d = 0.5$, 1, 3, 5, 8 kHz) are tested in one wind tunnel run (10 s), and the sample duration for each discharge frequency is one second. Between two consecutive discharge frequencies, the PSJAs are switched off for one second in order to recover the flow to baseline state. Figure 7(*a*) shows the power spectral densities at increasing discharge frequency. As evidenced, a majority of the discharge frequencies are able to elevate the peak of the power spectrum and the amplitude of low-frequency fluctuations. The only exception is $F_d = 0.5$ kHz, where the power spectrum peak is reduced noticeably by 13.6 %.

To compare the control effectiveness in different cases, the relative increase of raw voltage fluctuations caused by plasma actuation (denoted as $\Delta\sigma_u$) is computed as follows:

$$\Delta\sigma_u = (\sigma_{u,AT} - \sigma_{u,BL})/\sigma_{u,BL}, \tag{4.1}$$

where $\sigma_{u,BL}$ and $\sigma_{u,AT}$ denote the standard deviation of the raw voltage signals in baseline and actuated conditions, respectively. As shown in figure 7(*b*), with increasing discharge frequency, $\Delta\sigma_u$ first increases and then drops. The greatest increase of the raw voltage fluctuation is achieved at $F_d = 5$ kHz, which coincides with the control update frequency used in DRL. However, quantitatively, the peak of $\Delta\sigma_u$ found by periodic control (10.5 %) is only half of that earned by the optimal DRL control (21.2 %, case 2). This difference in control effectiveness can also be seen by comparing their power spectrum peaks in figures 5(*c*) and 7(*a*) (3.5×10^{-5} vs. 2.9×10^{-5}). Intuitively, the superior control authority of DRL can be explained from its ability to identify the events when plasma actuation is beneficial.

5. Summary

In this study, an FPGA-based experimental DRL control framework is proposed, capable of achieving a control frequency of O(1–10 kHz). This high control frequency, together with the fast-response plasma actuators and high-bandwidth hot-wire, enables us to challenge a real-time closed-loop control of the supersonic BFS flow at Mach 2.

https://doi.org/10.1017/jfm.2025.160 Published online by Cambridge University Press

Three cases with different state dimensionalities and initialisation strategies are tested, with an aim of maximising the mixing in the shear layer. The training efficiency of the DQN is found to be much higher than the widely adopted policy-based DRL algorithms. With only ten seconds of learning, the filtered reward, signifying the relative increase of the velocity fluctuation, is increased from 0 to a peak value of 0.2. Compared with the baseline case, the power spectra in actuated conditions are elevated as a whole. The final control law in case 2 suggests that plasma actuators should be shot when the state lies in the third quadrant. Physically, we believe that the mixing enhancement in the shear layer is correlated with the two bow shock waves issued from the jet orifice. In future, side-by-side comparisons between the different control strategies, including classical proportional integral derivative control and DRL control, are recommended.

Funding. This work was sponsored by the National Natural Science Foundation of China (Nos. U2341277, 52441602, 52025064).

Declaration of interests. The authors report no conflict of interest.

Data availability statement. This study is part of a project that aims to implement different DRL strategies in FPGA for purpose of real-time active flow control. The codes will be made open-source in GitHub after the project finishes.

References

Amico, E., Cafiero, G. & Iuso, G. 2022 Deep reinforcement learning for active control of a three-dimensional bluff body wake. *Phys. Fluids* **34**, 105126.

Benard, N., Pons-Prats, J., Periaux, J., Bugeda, G., Braud, P., Bonnet, J. & Moreau, E. 2016*a* Turbulent separated shear flow control by surface plasma actuator: experimental optimization by genetic algorithm approach. *Exp. Fluids* **57** (2), 1–17.

Benard, N., Sujar-Garrido, P., Bonnet, J.-P. & Moreau, E. 2016*b* Control of the coherent structure dynamics downstream of a backward facing step by DBD plasma actuator. *Intl J. Heat Fluid Flow* **61**, 158–173.

Brunton, S., Noack, B. & Koumoutsakos, P. 2020 Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52** (1), 477–508.

Cattafesta L.III & Sheplak, M. 2011 Actuators for active flow control. *Annu. Rev. Fluid Mech.* **43** (1), 247–272.

Deng, X., Hu, G.& Chen, W. 2024 Intelligent active flow control of long-span bridge deck using deep reinforcement learning integrated transfer learning. *J. Wind Engng Indust. Aerodyn.* **244**, 105632.

Dong, X., Hong, H., Deng, X., Zhong, W. & Hu, G. 2023 Surrogate model-based deep reinforcement learning for experimental study of active flow control of circular cylinder. *Phys. Fluids* **35**, 105147.

Fan, D., Yang, L., Wang, Z., Triantafyllou, M. & Karniadakis, G. 2020 Reinforcement learning for bluff body active flow control in experiments and simulations. *Proc. Natl Acad. Sci. USA* **117** (42), 26091–26098.

Gad-el Hak, M. 2000 *Flow Control: Passive, Active, and Reactive Flow Management*. Cambridge University Press.

Jia, W. & Xu, H. 2024 Deep reinforcement learning-based active flow control of an elliptical cylinder: transitioning from an elliptical cylinder to a circular cylinder and a flat plate. *Phys. Fluids* **36**, 074117.

Kingma, D. 2014 Adam: a method for stochastic optimization. arXiv preprint arXiv: 1412.6980.

Kong, Y., Wu, Y., Zong, H. & Guo, S. 2022 Supersonic cavity shear layer control using spanwise pulsed spark discharge array. *Phys. Fluids* **34**, 054113.

Maceda, G. Y., Varon, E., Lusseyran, F. & Noack, B. R. 2023 Stabilization of a multi-frequency open cavity flow with gradient-enriched machine learning control. *J. Fluid Mech.* **955**, A20.

Mnih, V. *et al.*et al. 2015 Human-level control through deep reinforcement learning. *Nature* **518** (7540), 529–533.

Paris, R., Beneddine, S. & Dandois, J. 2021 Robust flow control and optimal sensor placement using deep reinforcement learning. *J. Fluid Mech.* **913**, A25.

Pastoor, M., Henning, L., Noack, B.R., King, R. & Tadmor, G. 2008 Feedback shear layer control for bluff body drag reduction. *J. Fluid Mech.* **608**, 161–196.

https://doi.org/10.1017/jfm.2025.160 Published online by Cambridge University Press

PINO, F., SCHENA, L., RABAULT, J. & MENDEZ, M.A. 2023 Comparative analysis of machine learning methods for active flow control. *J. Fluid Mech.* **958**, A39.

RABAULT, J., KUCHTA, M., JENSEN, A., RÉGLADE, U. & CERARDI, N. 2019 Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *J. Fluid Mech.* **865**, 281–302.

RABAULT, J. & KUHNLE, A. 2019 Accelerating deep reinforcement learning strategies of flow control through a multi-environment approach. *Phys. Fluids* **31**, 094105.

REN, F., RABAULT, J. & TANG, H. 2021 Applying deep reinforcement learning to active flow control in weakly turbulent conditions. *Phys. Fluids* **33**, 037121.

SHIMOMURA, S., SEKIMOTO, S., OYAMA, A., FUJII, K. & NISHIDA, H. 2020 Experimental study on application of distributed deep reinforcement learning to closed-loop flow separation control over an airfoil. *AIAA Scitech 2020 Forum*. https://doi.org/10.2514/6.2020-0579

SONODA, T., LIU, Z., ITOH, T. & HASEGAWA, Y. 2023 Reinforcement learning of control strategies for reducing skin friction drag in a fully developed turbulent channel flow. *J. Fluid Mech.* **960**, A30.

SUÁREZ, P., ALCANTARA-AVILA, F., MIRÓ, A., RABAULT, J., FONT, B., LEHMKUHL, O. & VINUESA, R. 2024 Active flow control for drag reduction through multi-agent reinforcement learning on a turbulent cylinder at $Re_d = 3900$. arXiv preprint arXiv: 2405.17655.

TANG, H., RABAULT, J., KUHNLE, A., WANG, Y. & WANG, T. 2020 Robust active flow control over a range of Reynolds numbers using an artificial neural network trained through deep reinforcement learning. *Phys. Fluids* **32**, 053605.

VIGNON, C., RABAULT, J. & VINUESA, R. 2023 Recent advances in applying deep reinforcement learning for flow control: perspectives and future directions. *Phys. Fluids* **35**, 031301.

WANG, Q., YAN, L., HU, G., CHEN, W., RABAULT, J. & NOACK, B.R. 2024 Dynamic feature-based deep reinforcement learning for flow control of circular cylinder with sparse surface pressure sensing. *J. Fluid Mech.* **988**, A4.

WANG, Y.-Z., HUA, Y., AUBRY, N., CHEN, Z.-H., WU, W.-T. & CUI, J. 2022 Accelerating and improving deep reinforcement learning-based active flow control: transfer training of policy network. *Phys. Fluids* **34**, 073609.

ZONG, H., CHIATTO, M., KOTSONIS, M. & DE LUCA, L. 2018 Plasma synthetic jet actuators for active flow control. In *Actuators*, vol. **7**, pp. 77. MDPI.

ZONG, H. & KOTSONIS, M. 2017 Realisation of plasma synthetic jet array with a novel sequential discharge. *Sensors Actuators A: Phys.* **266**, 314–317.

ZONG, H. & KOTSONIS, M. 2018 Formation, evolution and scaling of plasma synthetic jets. *J. Fluid Mech.* **837**, 147–181.

ZONG, H., WU, Y., LIANG, H., SU, Z. & LI, J. 2024 Experimental study on Q-learning control of airfoil trailing-edge flow separation using plasma synthetic jets. *Phys. Fluids* **36**, 015101.

https://doi.org/10.1017/jfm.2025.160 Published online by Cambridge University Press