Nonlinear optimal control using deep reinforcement learning

Michele Alessandro Bucci, Onofrio Semeraro, Lionel Mathelin, Laurent Cordier, Alexandre Allauzen, Guillaume Wisniewski

Nonlinear Optimal Control using Deep Reinforcement Learning

Michele Alessandro Bucci, Onofrio Semeraro, Alexandre Allauzen, Laurent Cordier and Lionel Mathelin

Abstract We propose a shift of paradigm for the control of fluid flows based on the application of deep reinforcement learning (DRL). This strategy is quickly spreading in the machine learning community and it is known for its connection with nonlinear control theory. The origin of DRL can be traced back to the generalization of the optimal control to nonlinear problems, leading – in the continuous formulation – to the Hamilton-Jacobi-Bellman (HJB) equation, of which DRL aims at providing a discrete, data-driven approximation. The only a priori requirement in DRL is the definition of an instantaneous reward as measure of the relevance of an action when the system is in a given state. The value function is then defined as the expected cumulative rewards and it is the objective to be maximized. The control action and the value function are approximated by means of neural networks. In this work, we clarify the connection between DRL and rediscuss our recent results for the control of the Kuramoto-Sivashinsky (KS) equation in one-dimension [4] by means of a parametric analysis.

Michele Alessandro Bucci
TAU-Team, INRIA Saclay, LRI, Université Paris-Sud, 650, Rue Noetzlin, 91190 Gif-sur-Yvette (FR), e-mail: michele-alessandro.bucci@inria.fr

Onofrio Semeraro
LIMSI, CNRS, Université de Paris-Saclay, 507, Rue John Von Neumann, 91400 Orsay (FR) e-mail: semeraro@limsi.fr

Alexandre Allauzen
LAMSADE, Université Paris Dauphine, Place du Maréchal de Lattre de Tassigny, 75016 Paris (FR) e-mail: alexandre.allauzen@dauphine.fr

Laurent Cordier
Institut Pprime, CNRS, Université de Poitiers, ENSMA, 11 Boulevard Marie et Pierre Curie, Futuroscope Chasseneuil (FR) e-mail: Laurent.Cordier@univ-poitiers.fr

Lionel Mathelin
LIMSI, CNRS, Université de Paris-Saclay, 507, Rue John Von Neumann, 91400 Orsay (FR) e-mail: mathelin@limsi.fr

1 Introduction

Control theory methods have attracted research in fluid dynamics due to the scientific challenges and the potential impact that such a technology might have in several engineering sectors, ranging from aeronautics to naval and road transport. Further impulse to these developments is undoubtedly due to the current environmental needs. Carbon dioxide emissions are considered among the causes of global warming and any reduction of these emissions can be beneficial in this regard. In this work, we focus on active control based on reinforcement learning (RL) algorithms, one of the main sub-fields of machine learning [7, 3]; RL is mainly used in robotics and has gained popularity in the last years for the super-human performance achieved in solving tasks as complex as solving games such as `go`, [14]. A possible definition can be given by quoting a recent work by [11]: *"RL [...] studies how to use past data to enhance the future manipulation of a dynamical system"*. Not surprisingly, this definition could also apply to control theory algorithms: RL is deeply rooted into optimal control theory [9, 12, 11] as it relies on data-driven based solutions to the Bellman equation [2]. Indeed, while sharing the theoretical ground of the optimal control, RL is fully data-driven and, as such, is characterized by the applicability of data-based approaches, like the ability of using only a limited amount of sensor measurements for determining an optimal control policy.

Following this rationale, we aim at leveraging on RL strategies for the closed-loop nonlinear control. We have demonstrated in our recent work [4] that a nonlinear, chaotic system governed by the Kuramoto-Sivashinsky (KS) equations can be controlled without relying on a priori knowledge of the dynamics of the system, but solely on localized measurements of the system. Effective policies were computed, capable of driving the system to the vicinity of the unstable, non-trivial solutions of the KS in a chaotic regime [5]. Here, we further extend these results; we briefly introduce the basics of nonlinear optimal control theory and RL in §2; a parametric analysis is proposed in §3 aiming at discussing the robustness of the computed controllers.

2 Reinforcement learning: introductory elements

In this section, we briefly introduce the main elements for comparing control theory and the fundamentals of RL. We refer the interested reader to the literature in optimal control [9], including the fluid mechanics applications [8, 13, 6], and RL for deeper insights [7, 10, 14].

Bellman's optimality condition

First of all, we introduce the state-space model

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathcal{F}\left(\mathbf{v}(t), \mathbf{u}(t), t\right), \tag{1a}$$

$$\mathbf{x}(t) = \mathcal{G}\left(\mathbf{v}(t)\right), \tag{1b}$$

describing a dynamical system governed by the nonlinear map $\mathcal{F}$ and propagating the state $\mathbf{v} \in \mathbb{R}^N$. The model is forced by an input vector $\mathbf{u} \in \mathbb{R}^m$, with m being the number of inputs. In the second relation, the map $\mathcal{G}$ associates the observed state $\mathbf{v}$ to the observable $\mathbf{x} \in \mathbb{R}^p$, function of time t, recorded as outputs by p sensors. In the following, we will generally define the observables $\mathbf{x}$ and the input vector $\mathbf{u}$ as *signals*. The control signal $\mathbf{u}$ corresponds to the amplitude in time of localized forcing introduced in the system, typically as actuators.

The optimal control problem applied to the dynamical system in Eq. 1 can be stated as follows:

> To compute the control signal $\mathbf{u} \in \mathbb{R}^m$ using the sensor measurements $\mathbf{x} \in \mathbb{R}^p$, such that an objective function $\mathcal{J}$ is minimized.

A general expression of the objective function is given by

$$\mathcal{J}(\mathbf{v}_t, t, \underset{t \le \tau \le T}{\mathbf{u}(\tau)}) = h\left(\mathbf{v}(T), T\right) + \int_t^T r\left(\mathbf{v}(\tau), \mathbf{u}(\tau), \tau\right)\, \mathrm{d}\tau, \tag{2}$$

where h provides the terminal condition at time T, the optimization horizon, and r is the reward associated with the state $\mathbf{v}$ and the action $\mathbf{u}$. Note that t can be any value less than or equal to T. As previously stated, the objective of the controller is to provide a mapping between the sensor signal $\mathbf{x}$ and the control actions $\mathbf{u}$; this mapping is usually called *policy* and will be indicated as π such that unknown optimal signal is obtained as

$$\mathbf{u}^\star(t) = \pi^\star\left(\mathbf{x}(t), t\right). \tag{3}$$

Hereafter, optimal solutions will be indicated with a (*). When the system in Eq. 1 is known, linear (or linearizable) and time-invariant, a classic approach to optimal control is the linear quadratic regulator (LQR), obtained when the reward r is quadratic; in that case, it is possible to resolve the associated Riccati equation and compute the corresponding policy [9]. Here, we keep the formulation as general as possible and proceed by maximizing the value of the objective function in Eq. 2 on the right-hand side (RHS) as

$$\mathcal{J}^\star(\mathbf{v}(t), t) = \max_{\substack{\pi(\tau) \\ t \le \tau \le T}} \left[\int_t^T r\left(\mathbf{v}(\tau), \mathbf{u}(\tau), \tau\right)\, \mathrm{d}\tau + h\left(\mathbf{v}(T), T\right) \right]. \tag{4}$$

The RHS can be further manipulated by splitting the integral in two contributions

$$\mathcal{J}^\star(\mathbf{v}(t),t) = \max_{\substack{\pi(\tau) \\ t\le\tau\le t+\Delta t}} \left[\int_t^{t+\Delta t} r\,\mathrm{d}\tau + \mathcal{J}^\star(\mathbf{v}(t+\Delta t), t+\Delta t) \right], \tag{5}$$

where the first term defined in the interval $[t, t+\Delta t]$ and corresponds to an *immediate reward* while the remaining terms are now replaced by the optimal *value function*. The term $\mathcal{J}^\star(\mathbf{v}(t+\Delta t), t+\Delta t)$ can be developed in Taylor series about $\mathbf{v}(t)$ and, in the limit for $\Delta t \to 0$, it leads to the well known Hamilton-Jacobi-Bellman (HJB)

$$-\dot{\mathcal{J}}^\star(\mathbf{v}(t),t) = \max_{\pi(t)} \left[r\left(\mathbf{v}(t),\mathbf{u}(t),t\right) + \mathcal{J}^\star_{\mathbf{v}}\left(\mathbf{v}(t),t\right) \mathcal{F}\left(\mathbf{v}(t),\mathbf{u}(t),t\right) \right], \tag{6}$$

with $\mathcal{J}^\star_{\mathbf{v}}$ being the derivative with respect to the state, and the terminal condition $\mathcal{J}^\star(\mathbf{v}(T),T) = h\left(\mathbf{v}(T),T\right)$. This functional equation is continuous in time and defined backward. If the HJB is solved on the whole state-space and its value function is differentiable, the equation provides a necessary and sufficient condition for the optimum. More interestingly for what it follows, it can be shown that the discrete counterpart of the HJB equation is given by the Bellman equation

$$\mathcal{J}^\star(\mathbf{v}_t) = \max_{\mathbf{u}} \left[\Delta t\, r(\mathbf{v}_t,\mathbf{u}_t) + \gamma \mathcal{J}^\star(\mathbf{v}_{t+\Delta t}) \right], \tag{7}$$

where $\gamma = \exp\left(-\Delta t\,\rho\right)$ is the discount factor and Δt the time step. This equation is applied using the Markov decision process (MDP) framework, where the probability of evolving from the present state to the future one under the action $\mathbf{u}$ is expressed by transition matrices. Due to the probabilistic framework, the value function is reformulated in terms of expectation of the cumulative discounted reward defined by

$$\mathcal{J}^\pi(\mathbf{v}_t) = \mathbb{E}\left[\sum_{l=0}^{\infty} \gamma^l r(\mathbf{v}_{t+l\,\Delta t}) \right]. \tag{8}$$

The Bellman equation in (7) is central in dynamic programming, discrete optimal control and RL. We can observe an important property: the discounted infinite-horizon optimal problem is decomposed in a series of local optimal problems; more precisely, by quoting [2]

> "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

This property is the *Bellman's principle of optimality* and allows to solve the optimization problem by breaking it in a sequence of simpler problems.

Reinforcement learning

One of the assumptions in the previous section was the knowledge of the whole action-state space: when considering nonlinear maps $\mathcal{F}$ of large dimensions, the

computational costs would be prohibitive. As an alternative, we can observe that, in the Bellman equation, the model does not appear explicitly: it suffices to observe the state $\mathbf{v}_t$ and measure the reward r for recovering $\mathcal{J}^\pi(\mathbf{v}_t)$ from the interaction of the system with the environment under the policy. If $\mathcal{J}^\pi(\mathbf{v}_t)$ is a solution of eq. 7, we get a data-driven approximation of the optimal solution of the nonlinear control problem. This idea leads to the *reinforcement learning* (RL) framework; in the specific case of the *deep reinforcement learning* (DRL), the policy and the value function are represented by neural networks (NN).

General classification for RL algorithms

A rather general way to classify the RL algorithms can be made by identifying three main classes of techniques: i) *Actor–only*, ii) *Critic–only* and iii) *Actor–Critic*. The word *actor* is synonym of policy, while *critic* indicates the value function.

1. **Actor–only** methods consist of evaluating parametric policies. In this procedure, each policy is evaluated by recording the system for a long time and computing the cumulative discounted reward; the optimization is performed by means of stochastic gradient-descent algorithms for the update of the policy. These algorithms are usually referred to as `REINFORCE` algorithms. From the mathematical viewpoint, the actor–only method satisfies the Pontryagin's maximum principle, a necessary condition for the optimality, where the system is optimized in the vicinity of only one trajectory.
2. **Critic–only** methods are based on the value function approximation; a general expression of the Bellman equation associated with this class of algorithm is given by
$$Q^\pi(\mathbf{v}_t, \mathbf{u}_t) = r(\mathbf{v}_t, \mathbf{u}_t) + \gamma Q^\pi(\mathbf{v}_{t+\Delta t}, \mathbf{u}_{t+\Delta t}). \tag{9}$$
In this way, the state-action value function, or *Q-function*, is written as solution of the Bellman equation and measures the long-term reward of a system evolving along a trajectory emanating from $\mathbf{v}_t$ under an action $\mathbf{u}_t$, and subsequently driven by a policy π. Q-learning algorithms are aimed at approximating the optimal action-value function.
3. **Actor–critic** algorithms combine the two techniques, by providing an approximation for the policy and guaranteeing that the *Q-function* is a solution of the Bellman equation; this condition is satisfied when the analyzed system is Markovian and fully known from the observables.

Deep Deterministic Policy Gradient as an actor-critic algorithm for DRL

In this application, we opted for an actor-critic strategy, the Deep Deterministic Policy Gradient (DDPG) [10], capable of handling continuous action. First of all, we define the so-called *tuple*, composed by the current state $\mathbf{x}_t$, the associated reward r_t, and the state $\mathbf{x}_{t+1}$ obtained under the action $\mathbf{u}_t$. The tuples are iteratively stacked in memory, and define the MDP. Note that in the most general case, we do not consider the full state, but only local measurements such that $\mathbf{x} = \mathcal{G}(\mathbf{v})$: in this

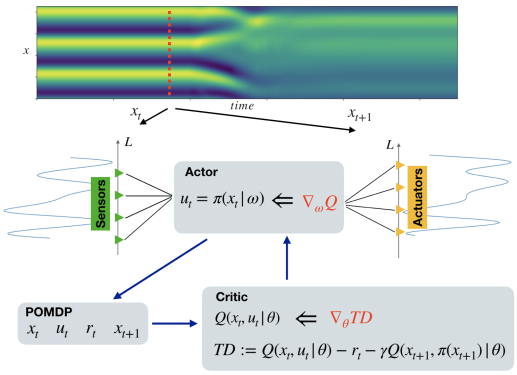

Fig. 1 Sketch of the DDPG algorithm applied for the control of the KS system. The system is detected by means of localized sensors; the current state of the system $\mathbf{x}$ is recorded from these measurements. Based on the action $\mathbf{u}$ and the scalar reward r, the updates of the Q-function and the policy π are performed. More details are provided in the text.

case, the observability of the MDP can be limited, so typically we refer to as partial observability (PO)-MDP. This aspect is crucial as it can lead to non-Markovian representations of the system and, as consequence, non-optimal solutions.

The approximations of policy π and value function Q are obtained by NN. In particular, each element of the i-th layer of the NN approximation can be written as

$$x_j^i = f_j\left(\psi \mathbf{x}^{i-1} + \mathbf{b}\right), \tag{10}$$

where $\{f_j\}$ represents the basis of nonlinear functions (`swish` or `tanh` in the present work) selected for the approximation, with $j = 1, \dots h$ and h the dimension of the hidden layer. The argument of these functions is given as a linear combination of each nodes at each layer x_j^i and the coefficients $\theta = \{\psi, \mathbf{b}\}$; the expansion coefficients θ are the unknowns and are computed using a stochastic, gradient-based optimization. In particular, by following the sketch in Fig. 1, the update of the value function Q, the *critic part*, is obtained by temporal difference TD as

$$TD = Q^\pi(\mathbf{x}_t, \mathbf{u}_t|\theta) - [r(\mathbf{x}_t, \mathbf{u}_t) + \gamma Q^\pi(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})|\theta]\,. \tag{11}$$

The gradient $\nabla_\theta TD$ allows to update the coefficients of the NN approximating the value function. By feeding back into the system the signal $\mathbf{u}$, based on the sensor measurements $\mathbf{x}$, we are able to close the loop and control the system, as sketched in Fig. 1. More in detail, the Q-function allows the update of the *actor part* providing the policy π

$$\mathbf{u}_t = \pi(\mathbf{x}_t|\omega) + \mathcal{N}, \tag{12}$$

and the action $\mathbf{u}_t$. The coefficients ω of the NN approximating π are updated via the gradient $\nabla_\omega Q$. A crucial aspect is the *exploration*: the optimality of the control is guaranteed by the hypothesis that the state-action space is known. To this end, the parameters ω of the NN describing the policy are perturbed and noise $\mathcal{N}$ is introduced on the action; both the noise processes vary over time and are damped as the solution converges. As last note, we stress that one of the main features of DRL is the continuous learning in real time of the optimal policy.

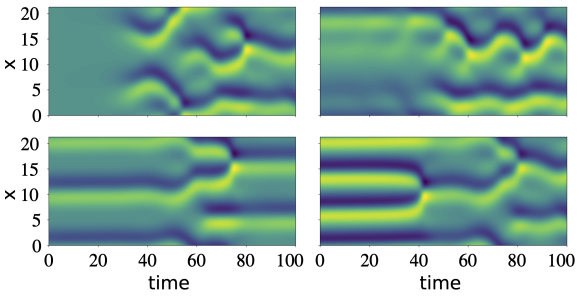

Fig. 2 When the domain is $L = 22$, the KS system exhibits 4 equilibria: the null solution $\mathbf{E}_0$ (top-left), and the non-trivial solutions labelled with $\mathbf{E}_i$ and $i = 1, 2, 3$ (top-right, bottom-left and bottom-right, respectively). All of them are unstable, as shown by the dynamics of the system in the spatio-temporal plots.

3 Control of chaotic regimes: the Kuramoto-Sivashinsky system

In this section, we discuss the control of the one-dimensional Kuramoto–Sivashinsky (KS) equation using DRL. The KS system exhibits a rather rich dynamics, ranging from the steady solution to chaotic regimes. The critical parameter is the domain extent, here indicated with L. In particular, it can be shown that for $L < L_c = 2\pi$, the dynamics is stable and converges towards $\mathbf{E}_0 = \mathbf{0}$, while chaotic dynamics emerges for $L > L_c$. We consider the solutions obtained for $L = 22$, corresponding to a regime characterized by maximum Lyapunov exponent $\lambda_1 \approx 0.043$ and Kaplan-Yorke dimension $D_{KY} \approx 5.2$; for this case, the dynamics is low-dimensional and lies in a space characterized by three non-trivial equilibria and two traveling waves [5]. In Fig. 2, we show the null solution $\mathbf{E}_0$ and the three non-trivial solution labelled $\mathbf{E}_i$, with $i = 1, 2, 3$; each of these solutions is unstable: the dynamics of the system becomes chaotic after a short transient. When increasing the domain extension, the number of positive Lyapunov exponents increases and the dynamics exhibits spatio-temporal chaos. The dynamics in time of the velocity $\mathbf{v} \in \mathbb{R}^N$ is governed by the equation

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\frac{\partial \mathbf{v}}{\partial x} = -\frac{\partial^2 \mathbf{v}}{\partial x^2} - \frac{\partial^4 \mathbf{v}}{\partial x^4} + \mathbf{g}(t), \tag{13}$$

here discretized with a resolution of $N = 64$ grid points on a periodic domain. The periodic domain allows for a Fourier mode expansion for the numerical resolution. Time marching was performed by 3rd-order Runge-Kutta scheme; the nonlinear terms are solved explicitly, while the linear terms are implicit. For all numerical simulations, a time step of 0.05 was adopted.

The control forcing is introduced by the term $\mathbf{g}(t) = \mathbf{B}\mathbf{u}(t)$, where $\mathbf{B} \in \mathbb{R}^{N\times m}$ is the spatial distribution of $m = 4$ localized, Gaussian shaped actuators

$$\mathbf{B}(x_a) = (2\pi\sigma)^{-1/2} \exp\left(-\frac{(x - x_a)^2}{2\sigma^2}\right), \tag{14}$$

placed at $x_a \in \{0, L/4, L/2, 3L/4\}$ and amplitude modulated in time by the forcing in time $\mathbf{u} \in \mathbb{R}^m$, computed by the DDPG and based on $p = 8$ localized sensor measurements, staggered with the respect to the actuators location and equidistant.

It can be shown that the KS equation can be controlled using linear controllers in combination with localized actuation [1]; however, the scope of this investigation is to demonstrate the feasibility of a purely model-free approach to the control of nonlinear flows.

Implementation of DDPG

We choose as objective for our controller to drive the system towards the solution $\mathbf{E}_2$ such that the distance $\|\mathbf{E}_2 - \mathbf{v}\|_2 := -r$ is minimized. As mentioned before, the DDPG policy is based on NN and its structure is as follows:

1. The actor part, representing the mapping between sensors and actuators, has $m = 4$ inputs and $p = 8$ outputs. Two hidden layers are considered, of respective dimensions 128 and 64, with activation functions `swish` and `tanh`.
2. The critic part, representing the value function, consists of an input of dimension $m + p = 12$, and a scalar output. Two hidden layers are introduced of 256 and 128 nodes, both with `swish` activation functions.

`Adam` optimization is applied for the update.

Results

We extend the results of [4] by considering a parametric analysis on the values of the discount factor γ and maximum amplitude of the outputs. In particular, we consider three policies with $\gamma = \{0.95, 0.97, 0.99\}$ and $|\mathbf{u}| < 1.0$, and two other policies with $\gamma = 0.99$ and maximum output amplitude set as $|\mathbf{u}| < \{0.5, 1.5\}$. The policy with $|\mathbf{u}| < 1.0$ and $\gamma = 0.99$ is the same as analysed in [4]. Due to the Markovianity of the system, the controllers are capable of driving the system to the target state $\mathbf{E}_2$ regardless of the initial conditions; here, for sake of conciseness and to make the comparison possible, we choose $\mathbf{E}_1$ as initial condition of all the test-cases.

In Fig. 3*a*-*b*, we show the trajectory in the phase-space (obtained by projecting the dynamics on the first three Fourier modes) and the reward, respectively, for $\gamma = 0.95$ (blue-dashed), $\gamma = 0.97$ (blue-dotted) and $\gamma = 0.99$ (green). The output is bounded as $|\mathbf{u}| < 1.0$. Surprisingly, despite the three controllers are always capable to drive the dynamics of the system towards $\mathbf{E}_2$, the case with $\gamma = 0.99$ is also the one which exhibits smaller excursions in the phase-space before converging, with a higher reward. This behaviour resembles what is observed in model predictive control when longer time-horizon are chosen. In the second set of results, we show how with $\gamma = 0.99$, a different behaviour appears when changing the amplitudes $|\mathbf{u}| < \{0.5, 1, 1.5\}$, respectively depicted with a blue-dashed, red-dotted and green curve in Fig. 3*c*-*d*. In this case, as one would expect, in presence of greater control authority the policies are capable of converging rapidly towards the vicinity of $\mathbf{E}_2$; although the case with $|\mathbf{u}| < 1.5$ is the one showing higher reward, it is also characterized by a behaviour less clear than the case with $|\mathbf{u}| < 1.0$ when considering the phase-space (Fig. 3*c*).

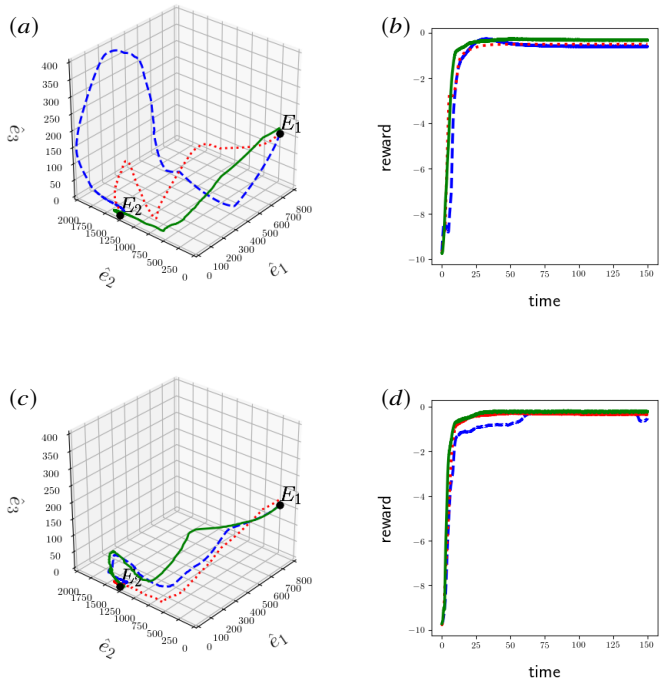

Fig. 3 Five policies for the control of the dynamics of the KS are compared for the same objective function: driving the system to the vicinity of $\mathbf{E}_2$. For simplicity of the discussion, the initial condition is set to be the invariant solution $\mathbf{E}_1$. The insets (a-b) show the behaviour of the system for $|\mathbf{u}| < 1$ and $\gamma = 0.95$ (blue-dashed), $\gamma = 0.97$ (red-dotted), $\gamma = 0.99$ (green); the trajectory is shown in phase space (a), while the corresponding reward is in (b). In the plots (c-d), we fix $\gamma = 0.99$ and consider three amplitudes: $|\mathbf{u}| < 0.5$ (blue-dashed), $|\mathbf{u}| < 1.0$ (red-dotted), $|\mathbf{u}| < 1.5$ (green); the corresponding trajectories (c) and rewards (d) are shown.

4 Conclusions and perspectives

This proceeding is part of a larger research effort aimed at applying reinforcement learning strategies to Navier-Stokes systems. Without any a-priori knowledge of the system, it is possible, by using localized sensors and actuators, to drive the dynamics of the chaotic KS system towards target states, here represented by unstable solutions of the system. The results are encouraging, although there are still numerous questions to be addressed. From the application point of view, the control signals (not-shown here) are highly non-trivial; in this sense, we are currently analysing the extent to which we are capable of reproducing an action comparable to a linearized, optimal control in the vicinity of the unstable state and the associated energy budget. A challenging aspect of this work is represented by the extension to Navier-

Stokes systems of this control strategy. A well-known limitation is represented, for instance, by the presence of time-delays in convective systems [6, 13]; this problem "translates" in RL into the so called credit-assignment problem. Also, it is important to keep a reasonable and realistic set-up, i.e. by limiting the number of sensors and actuators; these choices require a trade-off between the engineering needs and the low-observability, leading to the loss of Markovianity of the system, and low control-authority.

A future path is represented by the re-interpretation of RL from a control-oriented viewpoint: tools in standard, model-based control theory, such as model predictive control and adaptive algorithms [6, 15], rely on the Bellman formalism. The interplay between tools from optimal control theory and RL could help the development of reliable tools for the control of fluid systems.

Acknowledgements The authors gratefully acknowledge Sylvain Caillou for the support in the numerical implementation and Guillaume Wisniewski for interesting discussions. This project was funded by the French *Agence Nationale pour la Recherche* (ANR) and *Direction Générale de l'Armement* (DGA) via the FlowCon project (ANR-17-ASTR-0022).

References

1. Armaou A., Christofides P.D.: Feedback control of the Kuramoto–Sivashinsky equation. Physica D, **137**, 49–61 (2000)
2. Bellman, R.: Dynamic programming and stochastic control processes. Information and control **1**(3) (1958)
3. Brunton, S.L., Noack, B.R., Koumoutsakos, P.: Machine learning for fluid mechanics, Ann. Rev. Fluid Mech., **52** (2019)
4. Bucci, M.A., Semeraro, O., Allauzen, A., Wisniewski, G., Cordier, L., Mathelin, L.: Control of chaotic systems by deep reinforcement learning, Proc. Royal Soc. A, **475**-2231 (2019)
5. Cvitanović, P., and Davidchack, R.L., Siminos, E.: On the state space geometry of the Kuramoto–Sivashinsky flow in a periodic domain, SIAM J. Appl. Dyn. Syst., **9**1, 1-33 (2010)
6. Fabbiane, N., Semeraro, O., Bagheri, S., and Henningson, D.S.: Adaptive and model-based control theory applied to convectively unstable flows. App. Mech. Rev., 66(6), 060801 (2014)
7. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT Press, Boston (2016)
8. Kim, J., Bewley, T.R.: A linear systems approach to flow control. Annu. Rev. Fluid Mech., **39**, 383-417 (2007)
9. Lewis, F.L., Vrabie, D., Syrmos, V.L.: Optimal control. John Wiley & Sons (2012)
10. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning, arXiv preprint 1509.02971 (2015)
11. Matni, N., Proutiere, A., Rantzer, A., Tu, S.: From self-tuning regulators to reinforcement learning and back again, arXiv preprint 1906.11392 (2019)
12. Recht, B.: A tour of reinforcement learning: The view from continuous control. Annu. Rev. of Control, Robotics, and Autonomous Systems, **2** (2019)
13. Schmid, P.J., Sipp, D.: Linear control of oscillator and amplifier flows. Phys. Rev. Fluids, **1**(4)-040501 (2016)
14. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y.: Mastering the game of go without human knowledge, Nature, **550**-7676, 354 (2017)
15. Xiao D., Papadakis G.: Nonlinear optimal control of bypass transition in a boundary layer flow. Phys. Fluids **29**, 054103 (2017)