Control of chaotic systems by deep reinforcement learning

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

CONTROL OF CHAOTIC SYSTEMS BY DEEP REINFORCEMENT LEARNING

M.A. Bucci[1,*], O. Semeraro[1], A. Allauzen[1], G. Wisniewski[1], L. Cordier[2], L. Mathelin[1]

1 LIMSI, CNRS, University Paris-Saclay, Orsay (FR)
2 Institut PPrime, CNRS - Poitiers (FR)

1 - Introduction

Fig. 1 - Efficient aerodynamic surfaces can have a deep impact for the design of vehicles, airplanes or wind-power plants.

Can control of external flows improve the efficiency of vehicles, airplanes or wind-power plants?

We address this question by designing controllers that could possibly improve the behaviour of fluid mechanics system [3], in several conditions

- Laminar to turbulent transition
- Separation zone
- Transonic buffet
- Fluid structure interaction
- …

To this end, we test a Deep Reinforcement Learning (DRL) algorithm by applying it for the control of a non-linear, chaotic system governed by the Kuramoto-Sivashinsky (KS) equation [1].

DRL uses reinforcement learning principles for the determination of optimal control solutions and deep Neural Networks for approximating the value function and the control policy [1,2].

2 - Methods I: set-up

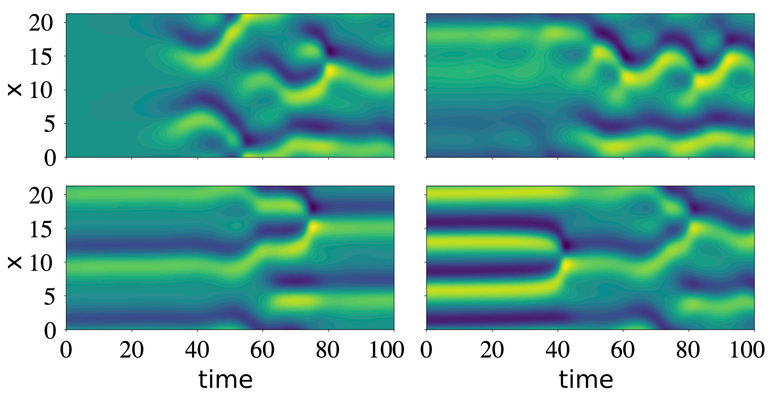

Fig. 2 - Dynamics of the Kuramoto-Sivashinsky (KS) equation, for the domain extent $L = 22$. The null equilibrium and the three non-trivial solutions are shown at $t = 0$ (leftmost in the insets).

The critical parameter for the KS equation is the domain extent; we choose $L = 22$, corresponding to a chaotic regime. The equation reads

$$\frac{\partial x}{\partial t} = -\nabla^4 x - \nabla^2 x - \frac{1}{2}|\nabla x|^2 + g$$

and it is numerically solved in a periodic domain. This regime is characterised by 4 equilibria (Fig. 2).

The controller is designed by solely relying on the measurements taken from 8 localised sensors (Fig. 3); the controller acts introducing a forcing (g) on the system by means of 4 localised actuators.

The entire control design is meant to be as realistic as possible for mimicking realistic conditions (i.e. localised actuations and measurements).

3 - Methods II: Reinforcement Learning

Reinforcement learning (RL) is based on the approximation of the Hamilton-Jacobi-Bellman (HJB) equation

$$\rho\mathcal{J}(x(t),t) = \max_u\{r(x,u) + \nabla\mathcal{J}(x(t),t)f(x,u)\}$$

$\mathcal{J}$ is the *value function*, r the *reward*, f the system, u the control action, and $\rho > 0$ is the discount factor. The discrete version, the Bellman equation leads to different approaches.

Actor-only: one policy π is tested on a long-time trajectory

$$\mathcal{J}_\pi(x_n) = r(x,u) + \gamma\mathcal{J}_\pi(x_{n+1}).$$

The solution satisfies the *Pontryagin maximum principle,* a necessary solution of optimality.
$\gamma = e^{-\rho t}$ is related to the discount factor.

Critic-only: the problem is decomposed in local problems, each associated with a policy/action u

$$Q(x_n,u_n) = r(x,u) + \gamma Q(x_{n+1},u_{n+1}).$$

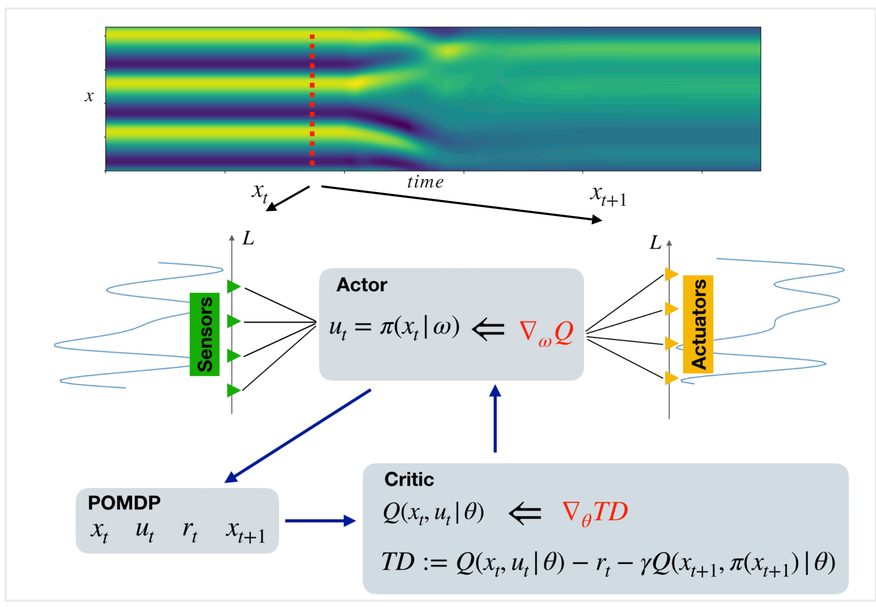

Fig. 3 - In the RL framework, an agent interacts with an environment by making observations and performing actions u. In return, the agent receives a reward that depends directly on the changes of the environment induced by the action. The control policy π and the temporal difference TD are approximated by means of neural networks.

The $Q-$function is the value function when is function of the state x and the action u. In this case the *Bellman principle* is satisfied, which is a necessary and sufficient condition for the optimality.

We use the Deep Determinist Policy Gradient (DDPG) combining the actor and critic strategies (Fig. 3). The control policy π and the $Q-$function are approximated by using Neural Networks (NN).

The optimisation of the NN is done by stochastic gradient descent with adaptive moments (ADAM) and the partial-observed Markov Decision Process (PO-MDP) is iteratively stacked in memory and used to reduce the temporal difference $TD = Q_n - (r + \gamma Q_{n+1})$.

4 - Results

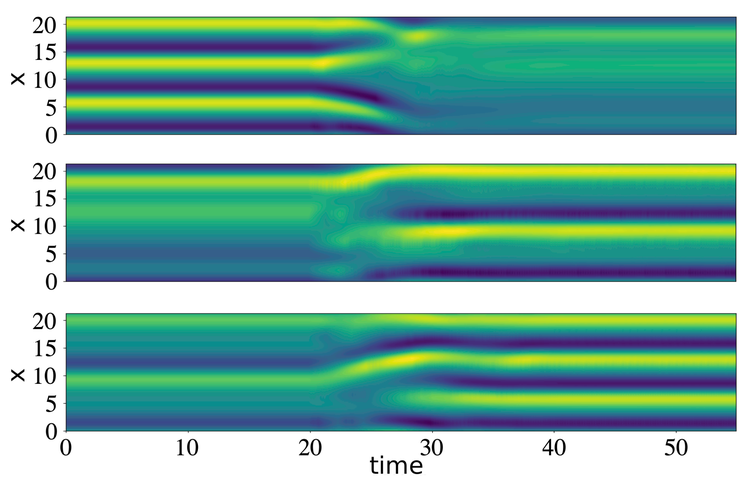

Fig. 4 - Control of the KS system by means of RL.

Three policies are designed, each of which minimising the distance between the current state and the target state, represented by one of the non-trivial equilibria (Fig. 2).
The reward is defined using the 2-norm and reads

$$r(t) = -\|x(t) - E_i\|$$

with the discount factor equals $\gamma = 0.99$.

The training of the control policies takes on average one hour of computation on a standard CPU.

In Fig. 4 we show how in each of the three cases the policies are capable at driving and stabilising the dynamics around the unstable points. The polices are robust with respect of the initial conditions as shown in [1].

5 - Conclusion

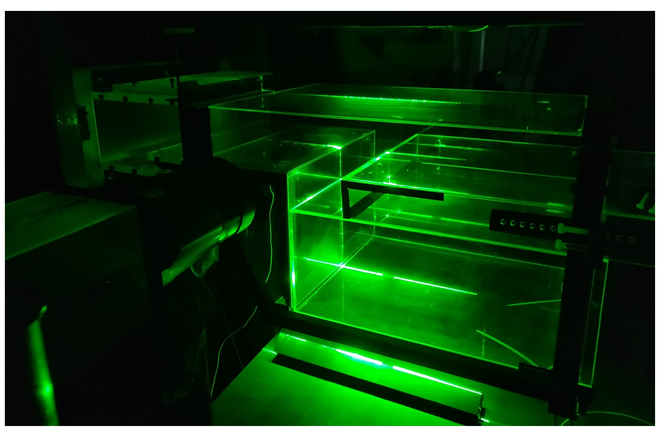

Fig. 5 - Cavity flow, experimental setup at LIMSI.

We tested Deep reinforcement learning for the optimal non-linear control of a chaotic system [1], using an actor-critic algorithm, the Deep Deterministic Policy Gradient.

- Full knowledge of the system is not required.
- In principle, the policy is a global optimum if the cost function is solution of the Bellman equation.
- Current work: 2D numerical cavity flow and 3D experimental counterpart (fig. 5).

6 - References

[1] Bucci, M.A. et al., (2019) arxiv.org/abs/1906.07672
[2] Silver, D. et al., (2014, June), In ICML.
[3] Sipp and Schmid, (2016), Appl. Mech. Rev. 68 - 20801

(*) Michele Alessandro Bucci (Email: **bucci.malessandro@gmail.com**)

FLOWCON PROJECT https://flowcon.cnrs.fr/
DECIPHER @LIMSI https://www.limsi.fr/en/research/aero/menuitem-aero-topics-en/aero-topics-decipher-en

ANR/DGA FLOWCON PROJECT
ANR-17-ASTR-0022

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR