Attractor Control Using Machine Learning

Article · November 2013

Source: arXiv

6 authors, including:

Vladimir Parezanovic
Khalifa University

30 PUBLICATIONS **288** CITATIONS

SEE PROFILE

Bernd R. Noack

279 PUBLICATIONS **6,461** CITATIONS

SEE PROFILE

Laurent Cordier
Institut Pprime

125 PUBLICATIONS **1,834** CITATIONS

SEE PROFILE

Attractor Control Using Machine Learning

Thomas Duriez,* Vladimir Parezanovic, Bernd R. Noack, and Laurent Cordier
Institut PPRIME, CNRS - Université de Poitiers - ENSMA,
UPR 3346, Département Fluides, Thermique, Combustion,
CEAT, 43, rue de l'Aérodrome, F-86036 Poitiers Cedex, France

Marc Segond
Ambrosys GmbH, Albert-Einstein-Str. 1-5, D-14469 Potsdam, Germany

Markus Abel
Ambrosys GmbH, Albert-Einstein-Str. 1-5, D-14469 Potsdam, Germany
LEMTA, 2 Avenue de la Fort de Haye F-54518 Vandoeuvre-ls-Nancy Cedex, France and
University of Potsdam, Karl-Liebknecht-Str. 24/25 D-14476 Potsdam, Germany

(Dated: December 16, 2013)

We propose a general strategy for feedback control design of complex dynamical systems exploiting the nonlinear mechanisms in a systematic unsupervised manner. These dynamical systems can have a state space of arbitrary dimension with finite number of actuators (multiple inputs) and sensors (multiple outputs). The control law maps outputs into inputs and is optimized with respect to a cost function, containing physics via the dynamical or statistical properties of the attractor to be controlled. Thus, we are capable of exploiting nonlinear mechanisms, e.g. chaos or frequency cross-talk, serving the control objective. This optimization is based on genetic programming, a branch of machine learning. This machine learning control is successfully applied to the stabilization of nonlinearly coupled oscillators and maximization of Lyapunov exponent of a forced Lorenz system. We foresee potential applications to most nonlinear multiple inputs/multiple outputs control problems, particulary in experiments.

PACS numbers: 05.45.-a,05.45.Gg,47.85.L-,07.05.Mh

Non-equilibrium dynamical systems often show undesirable behaviour. Examples include fluid turbulence [1–3] which may adversely effect the forces on transport vehicles, financial crises with dramatic consequences for the world's economy [4], or biophysical systems [5] with obvious impact on our own life. Control can serve to stabilize nonlinear extended or discrete coupled systems, as lasers [6], quantum systems [7], or delayed feedback systems [6, 8]. Consequently, the control of complex systems is an issue of major importance.

Cybernetics and control theory [9–12] have established a framework for control actions, mainly for stabilizing equilibria or reference trajectories; it is usually based on a linearization of the dynamical systems. In nonlinear dynamics, alternative strategies have been proposed, like time-delayed control with embedding, or synchronization [13–16] or stabilization of unstable periodic orbits [17].

One major complication in extended systems lies in the role of dynamically destabilizing modes and their nonlinear interactions. Often, the cost function used to evaluate the action of the control quantifies a long-term property of the attractor, while the nonlinear control response becomes unpredictable after a much shorter prediction horizon. Thus, model-based control design becomes next to impossible. In particular, stabilization of equilibria is generally not doable in complex systems with limited control authority.

Our approach hints to a pragmatic and fundamental solution out of this dilemma: we propose a model-free control design using the tools from machine learning, in particular the genetic programming (GP) [18, 19] as most suitable method. In general, machine learning comprises such important concepts as support vector machines [20], neural networks [21], or genetic algorithms to determine optimal parameters [22]. GP is a biologically inspired [23] function optimization method. Here, GP is used to identify the optimum feedback law to control the properties of an attractor, focusing on strongly nonlinear dynamical systems. An art in GP lies in the appropriate definition of a cost function to be optimized.

The applications lie dominantly in extended systems, as those mentioned above. Here, we describe the main physical and computational ideas using two examples: the first example is a generalized mean-field model with only 2 oscillating constituents, not controllable by linear methods. The second one is a forced Lorenz system by which we demonstrate the original use of the cost function: we want to maximize destabilization of the system within prescribed bounds. Its applications may lie in mixing systems, as in the case of combustion.

In the following, we restrict ourselves to ordinary differential equations, without loss of generality. The system is represented in phase space by the vector $\mathbf{a} \in \mathbb{R}^{n_a}$, it is measured by sensors $\mathbf{s} \in \mathbb{R}^{n_s}$, and controlled by actuators $\mathbf{b} \in \mathbb{R}^{n_b}$,

$$\frac{d\mathbf{a}}{dt} = \mathbf{F}\left(\mathbf{a},\mathbf{b}\right)\ ,\ \mathbf{s} = \mathbf{H}\left(\mathbf{a}\right)\ ,\ \mathbf{b} = \mathbf{K}\left(\mathbf{s}\right)\ , \tag{1}$$

with $\mathbf{F}$ a general nonlinear function, $\mathbf{H}$ the measurement function, and $\mathbf{K}$ the sensor-based control law. This law shall minimize the state- and actuation-dependent cost function:

$$J = J(\mathbf{a},\mathbf{b}). \tag{2}$$

The cost function value grades how a given control law $\mathbf{K}(\mathbf{s})$ performs relatively to the problem at stake. That function can be formulated in order to put the system in a desirable state as equilibrium (our first example) or in order to optimize a given measure on the system such as Lyapunov exponents (our second example). The lower the value of the cost function, the better the control law solves the problem, thus the cost function is a transcription of the control problem for the designated dynamical system.

We propose a model-free design of the control law: our method integrates concepts of genetic programming into control of dynamical systems. The genetic programming is used to design the best control law $\mathbf{K}(\mathbf{s})$ as a composition of elementary functions. A first set of control law candidates (called individuals) is generated through random composition of the elementary functions. The exploited GP algorithm [24] combines these operations as a tree [19], which allows to generate any linear or nonlinear function as initial generation of individuals. Each individual is attributed a cost through the evaluation of $J(\mathbf{a},\mathbf{b})$. The next set of individuals (called generation) is generated through mutation, cross-over or replication of individuals with specific rate for each process (Fig. 1).

The individuals used to produce the new generation are selected based on how they minimize the cost function. A global extremum of the cost function is typically approximated well in a finite number of generation, if the population contains enough diversity to explore the search space. Though there is no general mathematical proof for convergence, the method has been proved to be successful [25, 26].

In our first study, we consider a generalized mean-field model describing frequency cross-talk for a variety of physical phenomena including fluid flows [27, 28]. That model can be viewed as a generalisation of the Landau model [29] for the phase transition from equilibrium to periodic oscillation. Since we focus on frequency cross-talk, we choose a simple form of this model with two oscillators, coupled through the parametric, nonlinear variation of one growth rate:

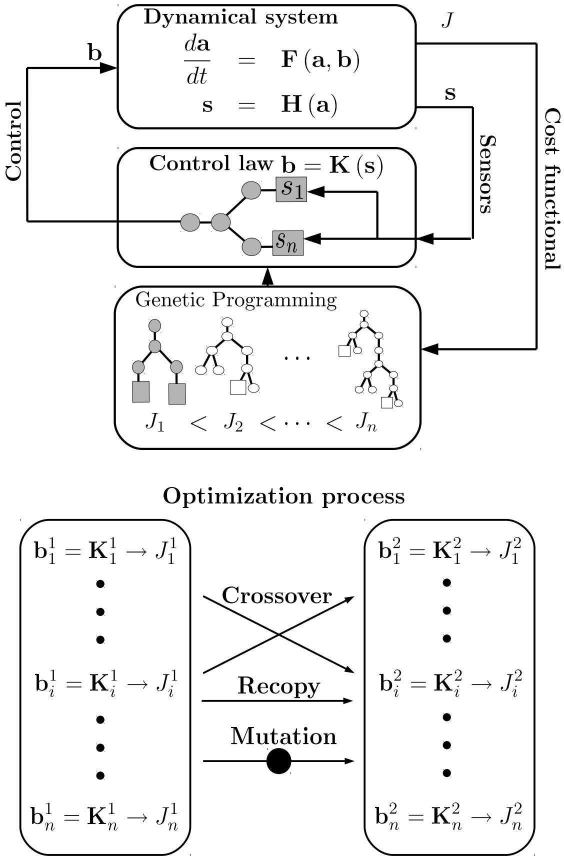
Dynamical system
$\frac{d\mathbf{a}}{dt} = \mathbf{F}(\mathbf{a},\mathbf{b})$
$\mathbf{s} = \mathbf{H}(\mathbf{a})$
J
$\mathbf{b}$
$\mathbf{s}$
Control
Sensors
Cost functional
Control law $\mathbf{b} = \mathbf{K}(\mathbf{s})$
s_1
s_n
Genetic Programming
$J_1 < J_2 < \cdots < J_n$
Optimization process
$\mathbf{b}_1^1 = \mathbf{K}_1^1 \rightarrow J_1^1$
$\mathbf{b}_i^1 = \mathbf{K}_i^1 \rightarrow J_i^1$
$\mathbf{b}_n^1 = \mathbf{K}_n^1 \rightarrow J_n^1$
Crossover
Recopy
Mutation
$\mathbf{b}_1^2 = \mathbf{K}_1^2 \rightarrow J_1^2$
$\mathbf{b}_i^2 = \mathbf{K}_i^2 \rightarrow J_i^2$
$\mathbf{b}_n^2 = \mathbf{K}_n^2 \rightarrow J_n^2$

FIG. 1: Top: Control design using GP. During a learning phase, each control law candidate is evaluated by the dynamical system. This is iterated over many generations of individuals. At the end of the process, the best individual (in grey) is determined and used for control. Bottom: Production of a new generation of individuals: each individual K_i^m is ranked by their cost, J_i^m, i pointing to the i^{th} individual, m to the m^{th} generation. An individual of the subsequent generation can be a copy, a mutation or the result of the cross-over of individuals selected in the preceding generation, according to their cost.

$$\begin{bmatrix} \frac{da_1}{dt} \\ \frac{da_2}{dt} \\ \frac{da_3}{dt} \\ \frac{da_4}{dt} \end{bmatrix} = \begin{bmatrix} \sigma_1 & \omega_1 & 0 & 0 \\ -\omega_1 & \sigma_1 & 0 & 0 \\ 0 & 0 & \sigma_2 & \omega_2 \\ 0 & 0 & -\omega_2 & \sigma_2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ b \end{bmatrix} \tag{3}$$
$$\text{with } \sigma_1 = \sigma_{10} - (a_1^2 + a_2^2 + a_3^2 + a_4^2).$$

Hereafter, we denote the sum of squared amplitudes as energy to avoid linguistic sophistication. We set $\omega_1 = \omega_2/10 = 1$ and $\sigma_{10} = -\sigma_2 = 0.1$, so that the first oscillator, (a_1, a_2), is unstable (would it be decoupled), while the other (a_3, a_4) is stable. When uncontrolled ($b \equiv 0$), the nonlinearity drives the first oscillator to nonlinear saturation through the change of total energy. The actuation effects directly only the stable oscillator. This system is arguably the simplest nonlinear dynamical system to exhibit frequency cross-talk. We choose to stabilize the first oscillator around its fixed point $(0,0)$ and thus a cost function which measures the fluctuation energy of that unstable oscillator. For any useful application, the energy used for control must be small, such that we penalize the actuation energy:

$$J = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2 - t_1} \left[a_1^2(t) + a_2^2(t) + \gamma b^2(t) \right] \mathrm{d}t, \qquad (4)$$

with $\gamma = 0.01$ as penalization coefficient. The quadratic form of state and actuation in the cost function is standard in control theory. Here, $t_1 = 5\tau$ with $\tau = 2\pi/\omega_1$ is chosen large enough to allow transients to decay and $t_2 = 100\tau$ is chosen in order to allow for meaningful statistics.

Knowing the nonlinearity at stake, an open-loop strategy can be designed: exciting the stable oscillator at frequency ω_2 will provoke an energy growth which stabilizes the first oscillator as soon as $a_1^2 + a_2^2 + a_3^2 + a_4^2 > \sigma_{10}$. It should be noted that linear control fails in using the frequency cross-talk mechanism. Indeed, the linearization of (3) yields to two uncoupled oscillators. Thus, the first oscillator is uncontrollable using linear methods.

We apply GP as a generic procedure with full-state observation ($\mathbf{s} \equiv \mathbf{a}$) in order to exploit all potential nonlinear mechanisms to control the unstable oscillator. In order to explore the function space, we use a set of elementary $(+, -, \times, /)$ and transcendental (e.g. $\exp, \sin, \log_2$) functions. The functions are protected to allow them to take arguments in $\mathbb{R}$. Additionally, the actuation command is limited to the range $[-1\,,\,1]$ in order to avoid too many integration failures and emulate an experimental actuator. Up to 50 generations comprising 1000 individuals are processed. The algorithm is stopped only when the last generation is reached or if an evaluation yields $J = 0$.

The control function ultimately returned by the GP process is a large expression, and due to the fact its argument is four-dimensional, it is hard to visualize. The full formula is given in [30]. It can be summarized as follows:

$$b = K_1(a_4) \times K_2(a_1, a_2, a_3, a_4). \qquad (5)$$

The function $K_1(a_4)$ describes a phase control that destabilizes the stable oscillator. The function $K_2(a_1, a_2, a_3, a_4)$ acts as a gain based on the energies of the oscillators. The performance and behaviour of the control law is visible in Fig. 2. The control law is energizing the second oscillator up to $10^0 \gg \sigma_{10}$, as soon as the first oscillator has an energy which is larger than 10^{-10}. This is stabilizing the unstable oscillator very quickly, on a time scale of 10^{-5}. When stabilization has happened, the control stays at very low values that keep the stable oscillator at a correspondingly low energy $\approx 10^{-10}$ while the energy of the unstable oscillator is exponentially growing close to its natural growth rate σ_{10}. The control exploits the frequency cross-talk and vanishes when not needed, i.e. $a_1 \approx a_2 \approx 0$. That control could not be found within a linear framework. In comparison with the best periodic excitation of the stable oscillator (with respect to J), less energy is used.

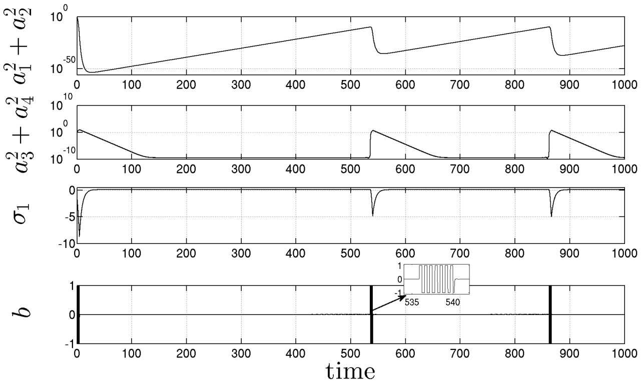

FIG. 2: Controlled generalized mean-field model. When the energy contained in the first oscillator (top) is larger than 10^{-10} the control (bottom) is exciting the second oscillator at frequency ω_2, its energy grows so that σ_1 reaches -5. This results in a fast decay of the energy in the first oscillator after which the control goes in "standby mode". An animation of the controlled system can be found in [30].

As second example, we consider the Lorenz system [31], forced in the third component:

$$\begin{aligned}
\frac{da_1}{dt} &= \sigma \left(a_2 - a_1 \right), \\
\frac{da_2}{dt} &= a_1 \left(\rho - a_3 \right) - a_2, \qquad (6) \\
\frac{da_3}{dt} &= a_1 a_2 - \beta a_3 + b,
\end{aligned}$$

by the control law $b = K(a_1, a_2, a_3)$, which basically influences the growth rate of a_3. The Lorenz system can be stable, periodic or chaotic depending on the parameter set used. We employ $\sigma = 10$, $\beta = 8/3$ and $\rho = 20$, such that the uncontrolled system ($b \equiv 0$) is periodic. Instead of stabilizing an equilibrium, we demonstrate how to render the system chaotic, now. Existing strategies may stabilize or destabilize periodic orbits [17, 32–34]. To reach chaotic behaviour, we aim at maximizing the largest Lyapunov exponent λ_1, again penalizing the actuation as a quadratic term b^2 with a factor γ. If λ_1 is positive, the system is chaotic and well-mixing. When considering a physical model the system needs to be bounded, i.e. the sum of the Lyapunov exponents must be negative. As a suitable cost function to be minimized and render the system chaotic we define:

$$\begin{array}{lll}
J &=& \exp(-\lambda_1) + \dfrac{\gamma}{T} \displaystyle\int_0^T b^2(t)\,\mathrm{d}t \qquad \text{if } \sum_{i=1}^3 \lambda_i < 0, \\
J &\to& \infty \qquad\qquad\qquad\qquad\qquad\quad \text{if } \sum_{i=1}^3 \lambda_i \geq 0,
\end{array} \qquad (7)$$

where $T = 100$ is the integration time and $\lambda_1 \geq \lambda_2 \geq \lambda_3$ are the Lyapunov exponents. These exponents are obtained by a standard algorithm [35, 36]. When the system is not bounded (i.e. it exceeds the bound we set in our program), J is assigned the largest real number possible on the computer. The control law is based on the full state and the basic operations that compose K are $+$, $-$, $\times$, $/$, as well as randomly generated constants. The maximum number of generations is again 50 with 1000 individuals each. To illustrate how the cost function definition influences the problem solved, we consider for γ the values of $\gamma_1 = 1$, $\gamma_2 = 0.01$ and $\gamma_3 = 0$. After 50 generations, the best individuals [30] have maximum Lyapunov exponents of respectively $\lambda_1 = 0.715$, 2.072 and 17.613. The changes in the system and the control function are displayed in Fig. 3. The control laws associated with γ_1 and γ_2 cases are affine expressions of a_3, the diminution of the actuation cost leads to larger amplitude of the feedback. In those cases the most efficient controls lead the system into behaviours close to the canonical Lorenz system ($\rho = 28$, $\lambda_1 = 0.905$). When $\gamma = 0.01$ the nature (from saddle point to spiral saddle point) and position of the central fixed point are changed. When the actuation is not penalized ($\gamma = 0$) the feedback law is a complex and fully non-linear law of all states. The nature and position of all the fixed points are changed as λ_1 reaches higher values. To our knowledge, no other model-based or model-free approach has been proposed in the litterature to optimize the largest Lyapunov exponent and controlling the attractor to reach a chaotic state.

Both examples illustrate how GP is progressing toward the minimum of the cost function. The statistical process that selects the individual for breeding allows individuals which are not optimal to be selected. This keeps diversity in the population and ensures that the GP process is not confined in a local minimum. For the stabilization of the mean-field model, the GP stopped after 35 generations, both oscillator energy and control energy vanished below numerical accuracy of the integration scheme. If the sensors define a subspace of $\mathbb{R}^{n_a}$, reaching this result is not guaranteed, as the controller needs to modulate the destabilizing feedback on the oscillator (a_3,a_4) by monitoring the energy in both oscillators.

We have demonstrated a way to determine the optimum control of a complex dynamical system in a model-free framework. The stabilization of the mean-field model shows that we can obtain closed-loop control exploiting frequency cross-talk. This is of primordial importance for large-scale turbulence control featuring this frequency cross-talk. Currently, an experiment on an actuated turbulent shear layer is run with the proposed machine learning control strategy. At this moment, the achieved mixing enhancement is of 1.5 times larger than any tested open- or closed-loop method (Fig. 4). Similarly, we have implemented a companion 2D DNS simulation with very promising results. This novel approach exhibits a high flexibility both in the class of systems it can address and in the specific problem it can solve exploiting the model-free formulation. Though a model is not needed, the more we know about the system, the better we can design the cost function according to the underlying physics. The major drawback of the model-free approach lies in the evaluation time, as each individual needs a simulation or experiment to be run. This translates in large time requirement should the process be serial. Consequently, in real applications, massive parallelization of computations or experiments will probably be needed. The relation of tree depth, number of generations, number of individuals with convergence is subject of ongoing research and may boost the performance considerably. The model-free control design is particulary interesting for experimental applications for which a model might not even be known - think about climate control or control of financial systems.

We acknowledge funding of the French ANR (Chaire d'Excellence TUCOROM), also, MS and MA acknowledge the support of the LINC project (no. 289447) funded by ECs Marie-Curie ITN program (FP7-PEOPLE-2011-ITN).

[*] thomas.duriez@univ-poitiers.fr

[1] U. Frisch and A. N. Kolmogorov, *Turbulence: The Legacy of A. N. Kolmogorov* (Cambridge University Press, 1995).

[2] S. B. Pope, *Turbulent Flows* (Cambridge University Press, 2000).

[3] B. Eckhardt, T. M. Schneider, B. Hof, and J. Westerweel, Annu. Rev. Fluid Mech. **39**, 447 (2007).

[4] D. Guégan, Annual Reviews in Control **33**, 89 (2009).

[5] E. M. Izhikevich, *Dynamical Systems in Neuroscience* (The MIT Press, 2007).

[6] J. Bechhoefer, Rev. Mod. Phys. **77**, 783 (2005).

[7] N. Behbood, G. Colangelo, F. Martin Ciurana, M. Napolitano, R. J. Sewell, and M. W. Mitchell, Phys. Rev. Lett. **111**, 103601 (2013).

[8] E. Schöll, A. Selivanov, J. Lehnert, T. Dahms, P. Hövel, and A. Fradkov, Int. J. Mod Phys B **26**, 1246007 (2012).

[9] N. Wiener, *Cybernetics or Control and Communication in the Animal and the Machine* (John Wiley, 1948).

[10] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers* (Princeton University Press, 2010).

[11] C. W. Rowley and D. R. Williams, Annu. Rev. Fluid Mech. **38**, 251 (2006).

[12] J. Hœpffner and P. J. Schmid, Applied Mechanics Reviews **62**, 020803 (2009).

[13] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization – A Universal Concept in Nonlinear Science* (Springer, Berlin, 2001).

[14] M. Abel, K. Ahnert, and S. Bergweiler, Phys. Rev. Lett. **103**, 114301 (2009).

[15] E. Gravier, X. Caron, G. Bonhomme, and T. Pierre,

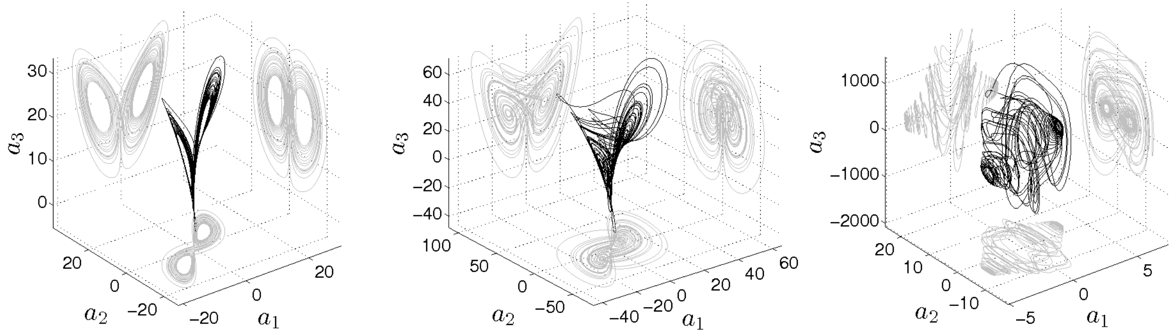

FIG. 3: Controlled Lorenz systems with $\sigma = 10$, $\beta = 8/3$ and $\rho = 20$. Left: $\gamma = 1$. The system exhibits chaotic behaviour ($\lambda_1 = 0.715$) close to the canonical chaotic Lorenz attractor with $\rho = 28$ ($\lambda_1 = 0.905$). Center: $\gamma = 0.01$. The system exhibits more complex trajectories, the nature of the central fixed point has changed and $\lambda_1 = 2.072$. Right: $\gamma = 0$. The nature of all fixed points has changed. The non-penalization of the actuation leads to a change in the scales ($\lambda_1 = 17.613$). An animation of the controlled system can be found in [30].

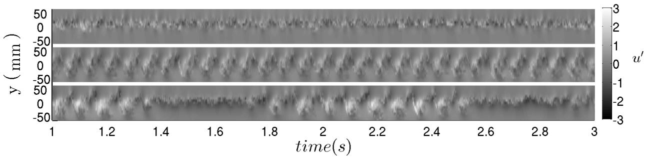

FIG. 4: Unforced (top), periodically forced (middle), and GP based closed-loop controlled (bottom) flows in the TUCOROM experimental mixing layer demonstrator [37]. 24 hot-wires probes record the velocity fluctuations accross the shear flow forced by micro-pulsed jets at the tip of the separating plate. Machine learning control allows to reach fluctuation levels 3 times as high as the natural flow and 1.5 time as high as the best tested open- and closed-loop strategies. It introduces unexpected frequencies like the $\approx 1Hz$ modulation visible on the bottom figure. Details on the running experiment in [30].

Phys. Plasmas **6**, 1670 (1999).

[16] M. G. Rosenblum, A. S. Pikovsky, and J. Kurths, Phys. Rev. Lett. **76**, 1804 (1996).

[17] E. Ott, C. Grebogi, and J. A. Yorke, Phys. Rev. Lett. **64**, 1196 (1990).

[18] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection* (The MIT Press, 1992).

[19] J. R. Koza, F. H. Bennett III, and O. Stiffelman, *Genetic Programming as a Darwinian Invention Machine*, vol. 1598 of *Lecture Notes in Computer Science* (Springer, 1999).

[20] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (The MIT Press, 2002).

[21] J. R. Noriega and H. Wang, IEEE Trans. Neural Networks **9**, 27 (1998).

[22] M. Mitchell, *An Introduction to Genetic Algorithms* (The MIT Press, 1996).

[23] M. Wahde, *Biologically Inspired Optimization Methods: An Introduction* (WIT Press, 2008).

[24] S. Luke, L. Panait, G. Balan, S. Paus, Z. Skolicki, R. Kicinger, E. Popovici, K. Sullivan, J. Harrison, J. Bassett, et al., *A Java-based Evolutionary Computation Research System* (2013), URL `http://cs.gmu.edu/~eclab/projects/ecj/`.

[25] M. A. Lewis, A. H. Fagg, and A. Solidum, in *International Conference on Robotics and Automation* (IEEE, 1992), vol. 3, pp. 2618–2623.

[26] P. Nordin and W. Banzhaf, Adaptive Behavior **5**, 107 (1997).

[27] B. Zielinska, S. Goujon-Durand, J. Dǔsek, and J. Wesfreid, Phys. Rev. Lett. **79**, 3893 (1997).

[28] D. M. Luchtenburg, B. Günter, B. R. Noack, R. King, and G. Tadmor, J. Fluid Mech. **623**, 283 (2009).

[29] L. D. Landau and E. M. Lifshitz, *The Classical Theory of Fields*, vol. 2 (Butterworth-Heinemann, 1975).

[30] *See supplemental material at [url will be inserted by publisher] for description of the best individuals, experiment and animations.*

[31] E. N. Lorenz, J. Atmos. Sci. **20**, 130 (1963).

[32] K. Pyragas, Phys. Lett. A **170**, 421 (1992).

[33] E. Schöll and H. G. Schuster, *Handbook of Chaos Control* (Wiley-VCH, Weinheim, 2007).

[34] N. B. Janson, A. G. Balanov, and E. Schöll, Phys. Rev. Lett. **93**, 010601 (2004).

[35] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, Phys. D **16**, 285 (1985).

[36] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis* (Cambridge University Press, 1997).

[37] V. Parezanović, J.-C. Laurentie, C. Fourment, L. Cordier, B. R. Noack, and T. Shaqarin, in *Proceedings of the 8th International Symposium On Turbulent and Shear Flow Phenomena* (2013).