



HAL
open science

Wake stabilization with Machine Learning Control (xMLC)

Guy Y. Cornejo Maceda, Bernd R Noack, François Lusseyran, Marek
Morzyński

► **To cite this version:**

Guy Y. Cornejo Maceda, Bernd R Noack, François Lusseyran, Marek Morzyński. Wake stabilization with Machine Learning Control (xMLC). von Karman Institute Lecture Series, Feb 2020, Rhode-Saint-Genèse, Belgium. hal-03874030

HAL Id: hal-03874030

<https://hal.science/hal-03874030v1>

Submitted on 27 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

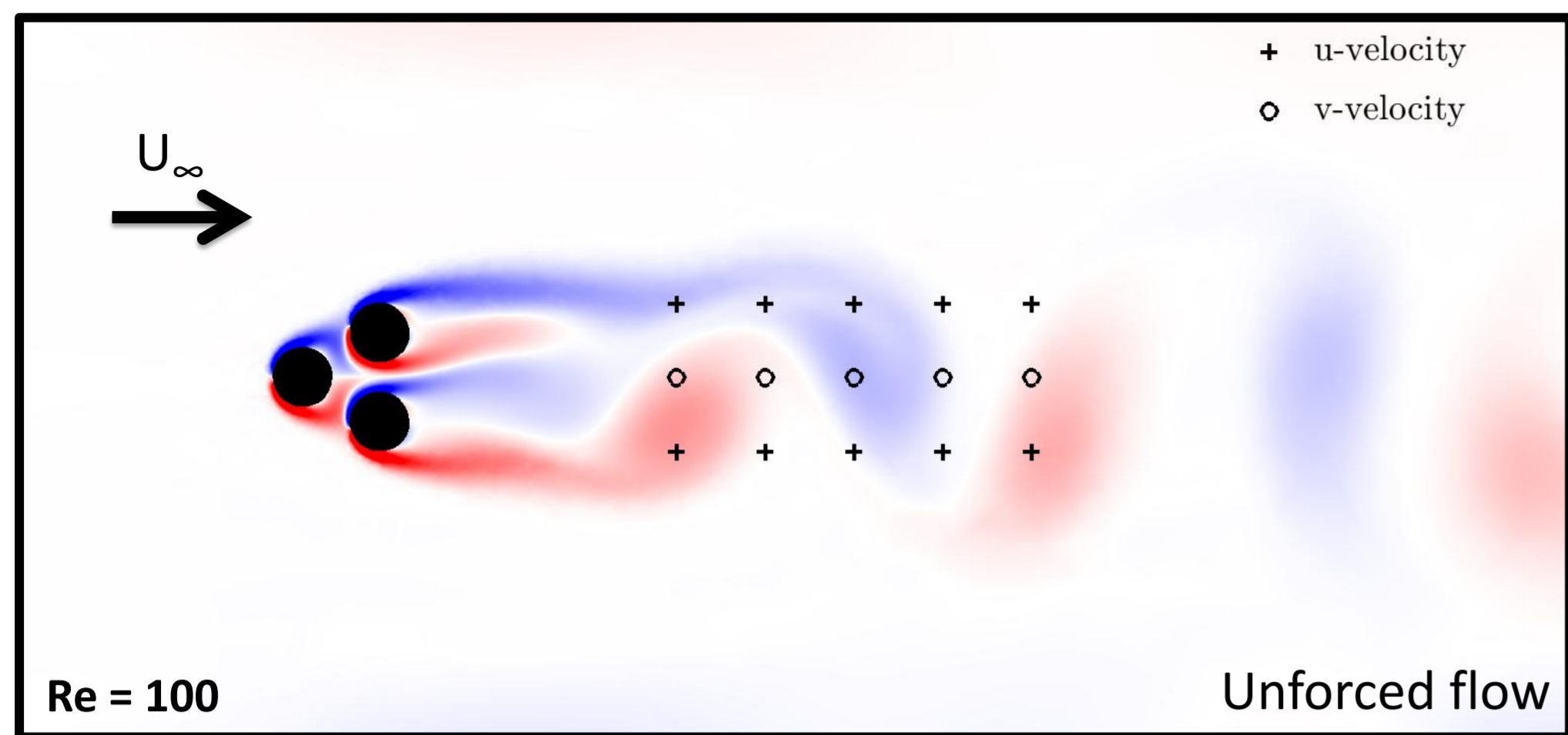
G. Y. Cornejo Maceda^{1*}, B. R. Noack^{1,2,3}, F. Lusseyran¹ and M. Morzyński⁴

¹LIMSI, CNRS, Université Paris-Saclay, F-91405 Orsay
²Harbin Institute of Technology, China

³Technische Universität Berlin, Germany
⁴Poznan University of Technology, Poland

*email: guy.cornejo@limsi.fr
 *website: www.cornejomaceda.com

The Fluidic Pinball : a control benchmark (2D incompressible, Re=100)

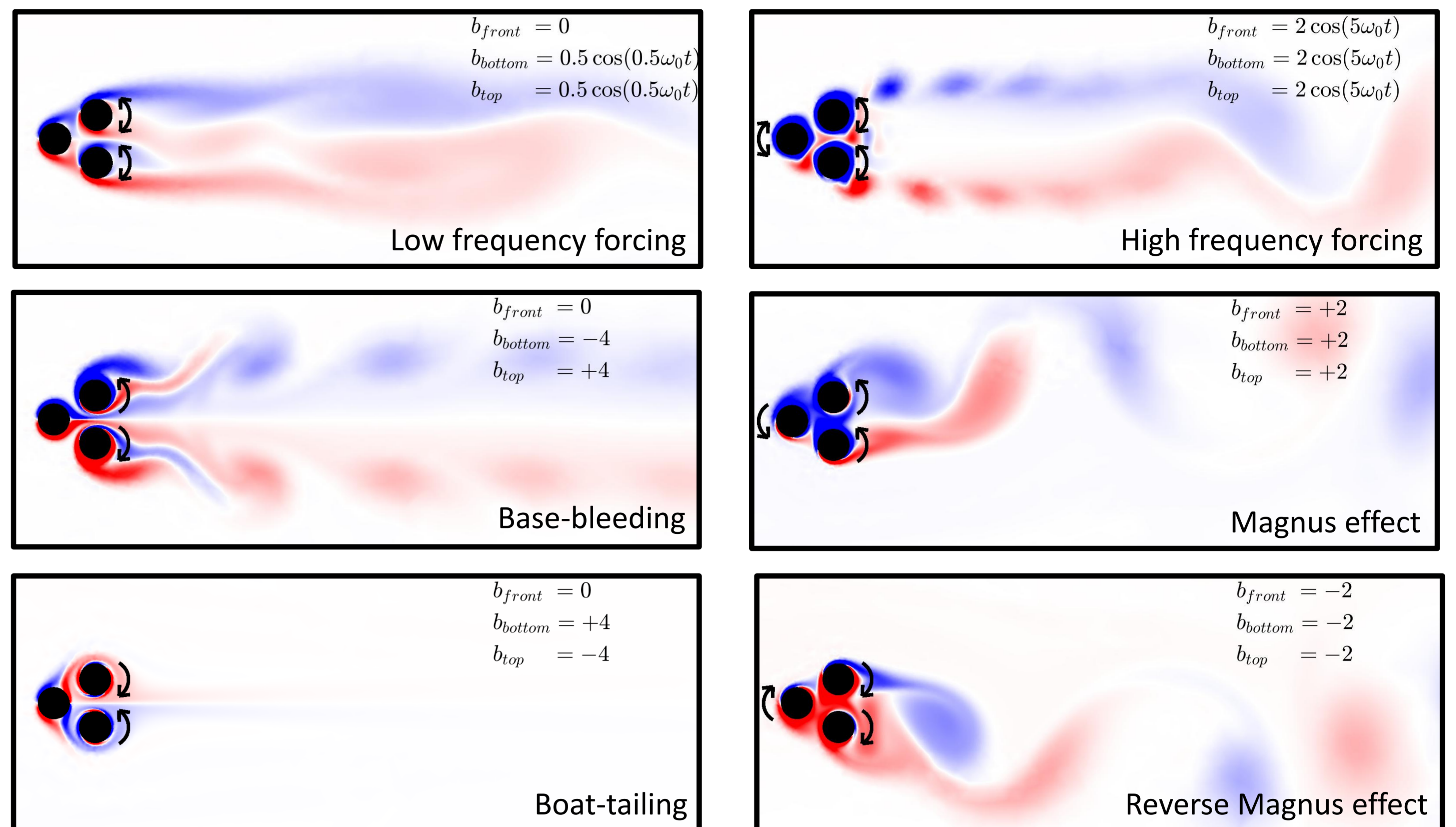


The fluidic pinball is a cluster of three cylinders — whose centers form an equilateral triangle — immersed in an incoming flow. It is a simple configuration that encapsulates key features that are at the heart of engineering problems such as time-delays and frequency crosstalks [1].

The key assets of the fluidic pinball are :

- **Successive bifurcations** revealing rich dynamics, similar to real flows [2].
- **MIMO (Multiple-Input Multiple-Output) control** thanks to 15 velocity probes downstream and the independent rotation of the three cylinders (see figure above).
- Thanks to control the fluidic pinball can reproduce up to six wake suppression mechanisms from literature (see six figures on the right). Low frequency and high frequency forcing reveal key enabler for control : **frequency crosstalks** [4].

The aim of this work is to develop a **self-learning control method** able to take into account the nonlinear interactions between the modes to optimize a given objective: **Machine Learning Control**.

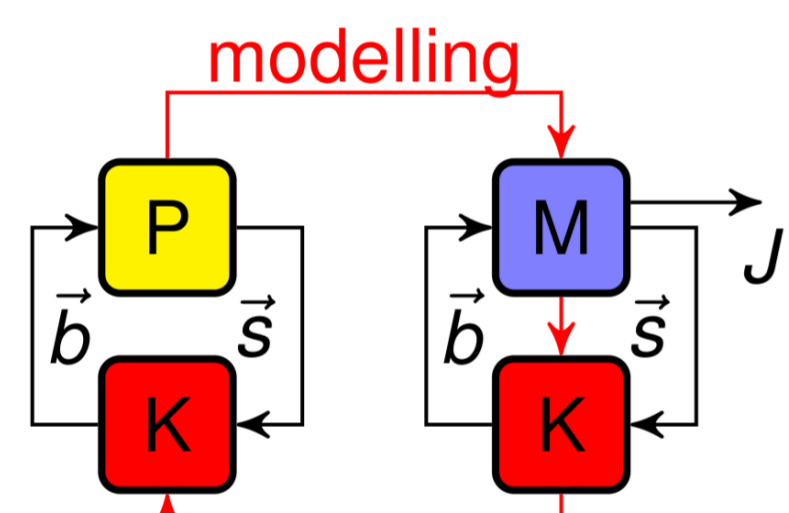


Model-based control vs Machine learning control

Norbert Wiener
1894-1964



Model-Based Control



$$\frac{d\vec{a}}{dt} = F(\vec{a}, \vec{b})$$

$$\vec{s} = G(\vec{a}, \vec{b})$$

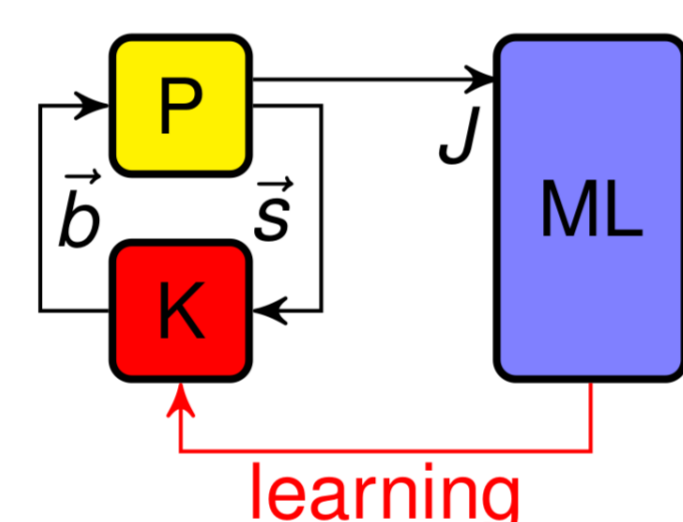
$$\vec{b} = K(\vec{s})$$

Bald eagle

33 millions years ago - Now



Machine Learning Control



$$K_{opt} = \arg \min_{K \in Y^X} J(K) \quad (1)$$

X : Sensor space ($X \subset (\mathbb{R}^{15} \times (\mathbb{R}^+))$)
 Y : Actuation space ($Y \subset \mathbb{R}^3$)

Controlling a dynamical system is to give an input that will change the system towards a desired state, assuring different constraints such as **noise rejection**, **uncertainty mitigation** and **robustness**. To this extent, two approaches oppose each other : **model-based control** (linear control theory, ERA/OKID, etc), pioneered by Norbert Wiener (1894-1964) and machine learning control. Model-based control victories are mainly in opposition and phasor control, and examples including MIMO control and frequency crosstalk are dim.

To override the intrinsic limitations of model-based control, a new approach is considered building on recent innovations in artificial intelligence : **machine learning control**.

Machine Learning Control (MLC, [4]) is a model-free control method based on **genetic programming** and building on the pioneering work of Dracopoulos [5]. It's a biologically-inspired method that mimics the Darwinian evolution to build fitter controllers, in the same way the bald eagle evolved through millions of years to be able to fly under gusty conditions. The cornerstone is the formulation of the control problem as a function optimization problem; such **non-convex problem** can then be solved thanks to MLC.

MLC strength relies on its ability to build nonlinear control laws **reproducing known control methods** including model-based (ERA/OKID), open-loop strategies (multi-frequency forcing) and closed-loop strategies (phasor control, ARMAX) and also **linear and nonlinear combinations** of them.

Acknowledgement :

Our research is supported by the ASTRID-ANR-17- FLOWCON project, *Contrôle d'écoulements turbulents en boucle fermée par apprentissage automatique*.

References :

- [1] Brunton, S.L., Noack, B. R., Closed-Loop Turbulence Control: Progress and Challenges, 2015 Appl. Mech. Rev. 67, 050801.
- [2] Deng, N., Noack, B. R., Morzyński, M., Pastur L., Low-order model for successive bifurcations of the fluidic pinball, 2020 JFM, 884.
- [3] Noack, B. R., Morzyński, M., The Fluidic Pinball: A Toolkit for Multiple-Input Multiple-Output Flow Control. Tech. Rep. 02/2017.
- [4] Duriez, T., Brunton, S.L., Noack, B.R., Machine Learning Control Taming Nonlinear Dynamics and Turbulence, Springer, 2017.
- [5] Dracopoulos, D., Evolutionary Learning Algorithms for Neural Adaptive Control, Springer, 1997.

MLC applied to the Fluidic Pinball Define the search space Y^X

Build a representation of Y^X to solve $K_{opt} = \arg \min_{K \in Y^X} J(K)$

J = net drag power
 = drag power + actuation power

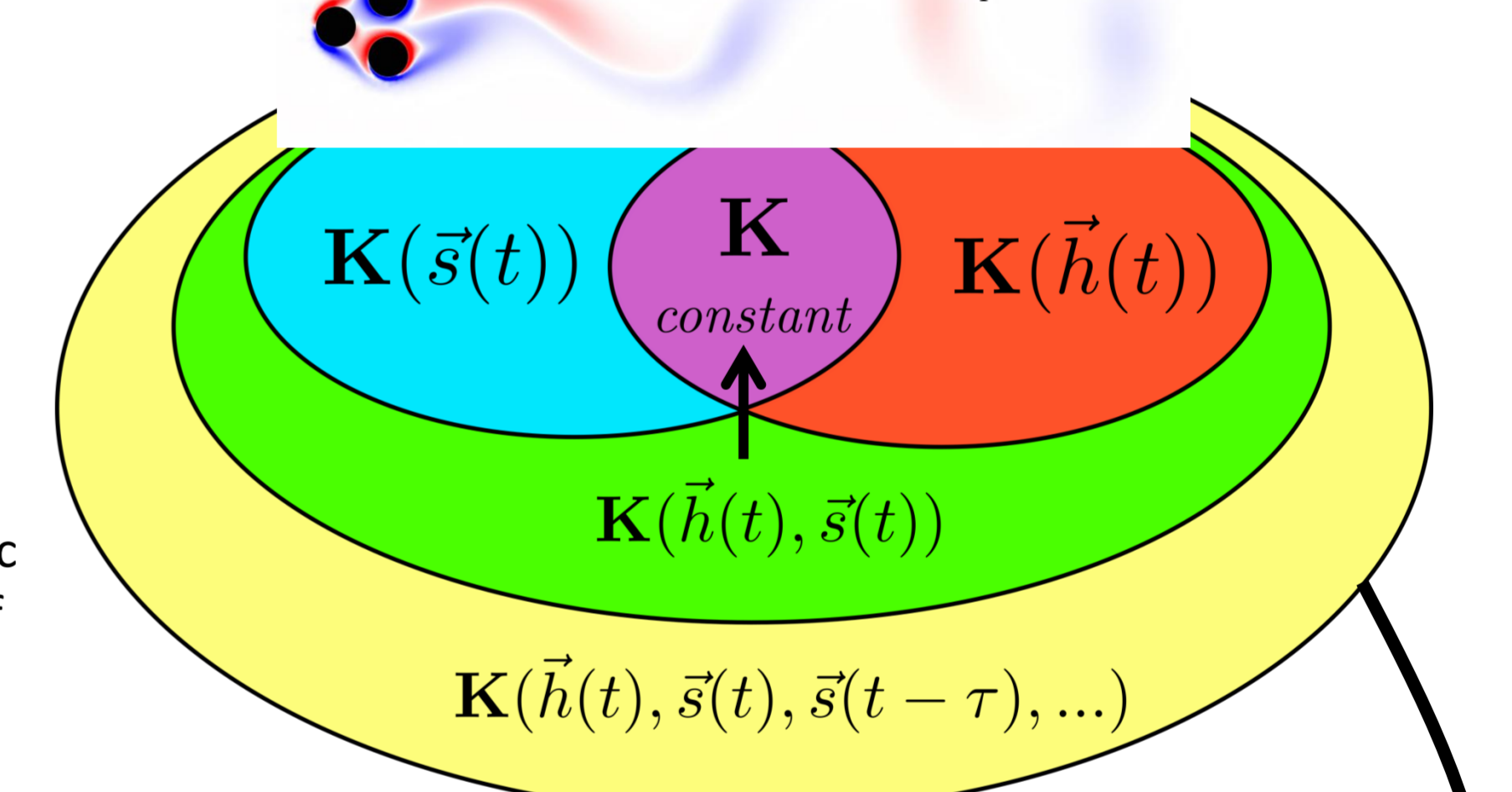
$$\vec{h}(t) = (h_1(t), \dots, h_i(t), \dots)$$

$$h_i(t) = \cos(\omega_i t)$$

$$\vec{s}(t) = (s_1(t), \dots, s_i(t), \dots)$$

$$s_i(t) = u_i(t) \text{ or } v_i(t)$$

44.3% net drag power reduction > 42.7% (best boat-tailing)
 $b_{front} = -0.15$
 $b_{bottom} = +1.76$
 $b_{top} = -1.80$



Our MLC code (xMLC) has been benchmarked on the fluidic pinball. The control problem considered is the reduction of the net drag power, and in order to minimize J in equation (1), we must first define the search space Y^X also known as **the space of control laws**.

The figure on the right is an Euler diagram of several search spaces explored, they comprise **multi-frequency forcing** (red set), **direct feedback control** (blue set) and a mix of the two (green set). xMLC has been tested on these sets and gave the control law on the right, outperforming previous results. However the control law is independent of time and lives in the purple set. To further explore the dynamics, **time-delayed sensors** have been added to the search space (yellow set) to take into account the **flow advection**.

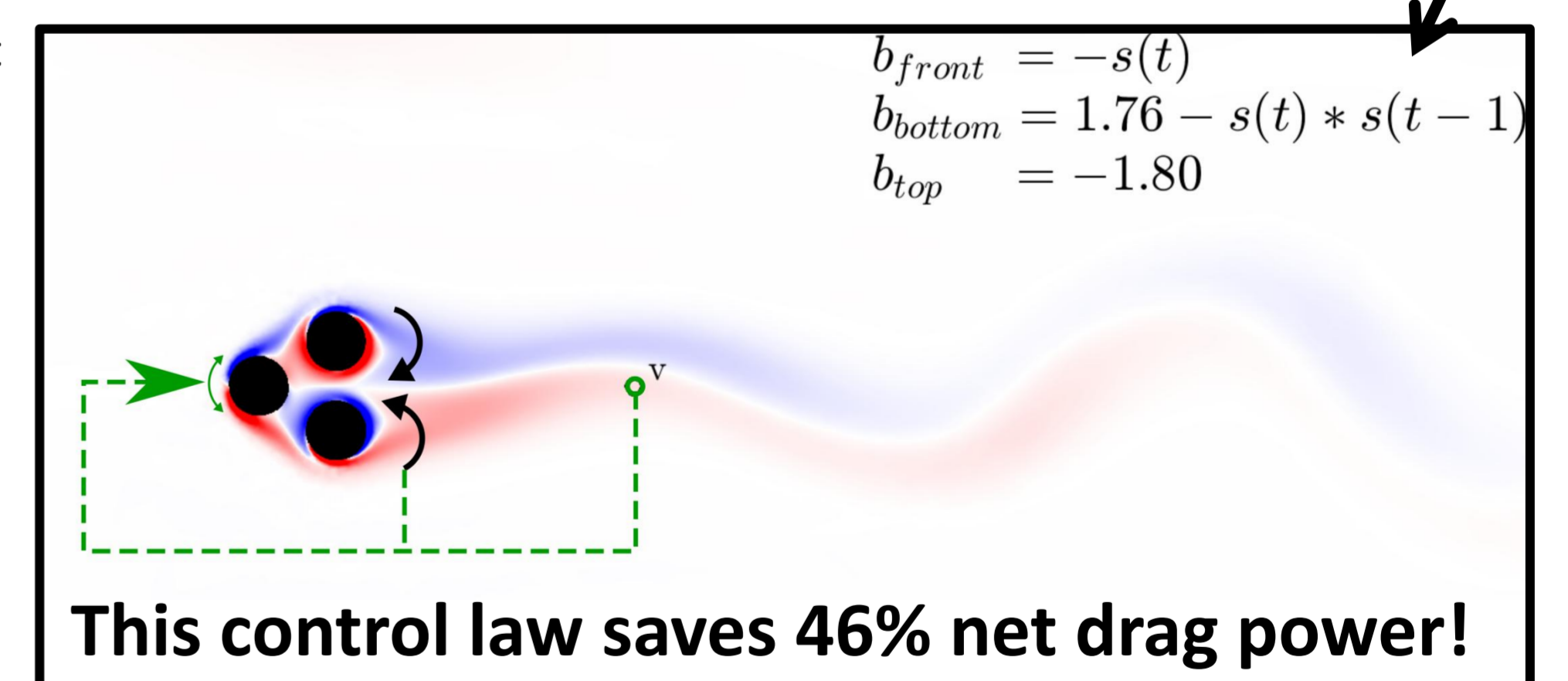
Best control law

After 100 individuals — from the yellow set — that evolved through 10 generations, xMLC gave an even better control law, managing 46.0% net drag power reduction.

The control law found is a non-trivial one built upon flow information ensuring some robustness in the control. This solution found with xMLC **combines two strategy from literature : boat-tailing** (two back cylinders) and **phasor control** (front cylinder).

Moreover, among the fifteen sensors given to xMLC, only one of them has been selected for the control. In this sense, xMLC works as a **sensor optimizer**.

Figures on the right show flow snapshots, phase portrait based on C_L , C_D , and PSD of C_L for the natural flow (— black), best boat-tailing solution (— blue) and xMLC (— red).



Conclusion

xMLC success lies in the fact that it is able to rediscover results from literature and combines them to even better performance.

Thanks to a meta-parameter optimization of the MLC algorithm and an implicit complexity penalization, the learning process has been accelerated by a factor 3.

Future developments seek to improve the learning rate by a factor 10, allowing multi-parameter testing for experiments and simulations with the potential to **reveal hidden control mechanisms** unreachable with model-based approaches. This new approach constitutes a change of paradigm where optimal solutions are first derived in an **unsupervised** manner and where **understanding** and modeling comes afterwards.

