EFMC14
14th European Fluid Mechanics Conference

13-16 SEPT. 2022

ATHENS GREECE
MEGARON ATHENS INTERNATIONAL CONFERENCE CENTRE

Optimal Control tools to minimize dispersion in chaotic flows

Chiara Calascibetta

Dept. Physics & INFN, University of Rome Tor Vergata

calascibetta@roma2.infn.it

L. Biferale (Uni. and INFN Tor Vergata, IT),
F. Borra (LPENS, CNRS Paris, FR), **A. Celani** (ICTP, IT),
M. Cencini (CNR and INFN Tor Vergata, IT).

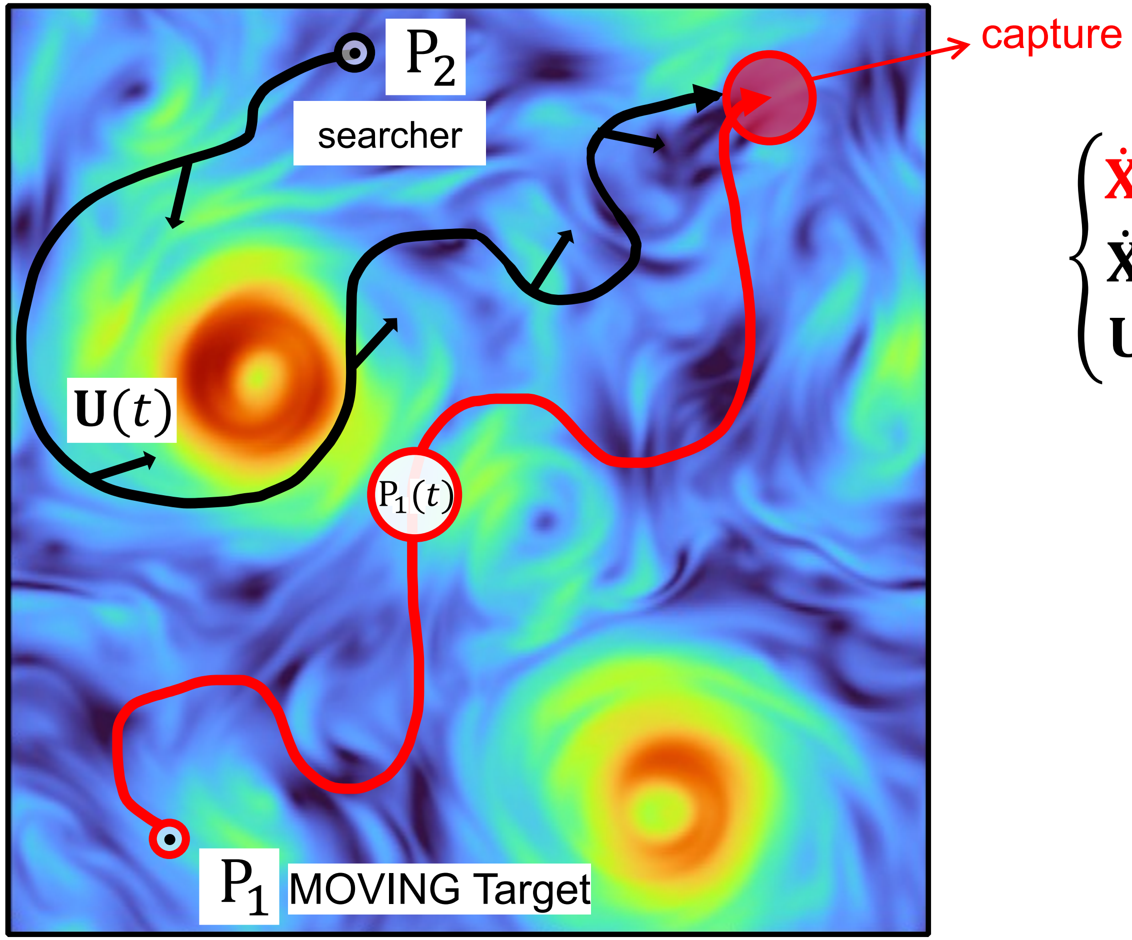
TOR VERGATA
UNIVERSITY OF ROME

INFN

erc

C. Calascibetta, L. Biferale, F. Borra, A. Celani, M. Cencini, **Optimal Control tools to minimize dispersion in chaotic flows**, (*in preparation* 2022).

2 AGENTS



Goal: **minimize** the **separation** in a finite time horizon

Problem setup

$$\begin{cases} \dot{\mathbf{X}}_t^1 = \boldsymbol{v}(\mathbf{X}_t^1) \\ \dot{\mathbf{X}}_t^2 = \boldsymbol{v}(\mathbf{X}_t^2) + \mathbf{U}(t) \\ \mathbf{U}(t) = V_s\,\hat{\boldsymbol{n}}(t) \end{cases} \longrightarrow \mathbf{U}(t) = ?$$

$$\hat{\boldsymbol{n}}\,(t) = (\cos[\theta_t], \sin[\theta_t])$$

Tools:
(1) Heuristic policies
(2) Optimal Control (OC) theory
(3) Reinforcement Learning (RL)

$$\begin{cases} \dot{\mathbf{X}}_t^1 = \boldsymbol{v}(\mathbf{X}_t^1, t) \\ \dot{\mathbf{X}}_t^2 = \boldsymbol{v}(\mathbf{X}_t^2, t) + \mathbf{U}(t) \\ \mathbf{U}(t) = V_s\, \hat{\boldsymbol{n}}(t) \end{cases}$$

$$\mathbf{U}(t) = ?$$

$$\begin{cases} \boldsymbol{R}_t = \mathbf{X}_t^2 - \mathbf{X}_t^1 \\ L = \text{characteristic scale of the flow} \end{cases}$$

(1) (semi) Heuristic policies

Trivial Policy: constantly chooses the direction that points towards the moving target

$$\hat{\boldsymbol{n}}(t) = -\hat{\boldsymbol{R}}_t$$

Surfing policy*: valid at large scales, i.e., $||\boldsymbol{R}_t|| \gg L$. Based on a free parameter τ_s .

Perturbative policy: valid at small scales, i.e., $||\boldsymbol{R}_t|| \ll L$. Based on a free parameter τ_p .

*Monthiller, Rémi, et al. **Surfing on Turbulence: A Strategy for Planktonic Navigation**. *Phys. Rev. Lett.* ***129***, *064502* (2022)

$$\begin{cases} \dot{\mathbf{X}}_t^1 = \boldsymbol{v}(\mathbf{X}_t^1, t) \\ \dot{\mathbf{X}}_t^2 = \boldsymbol{v}(\mathbf{X}_t^2, t) + \mathbf{U}(t) \\ \mathbf{U}(t) = V_s\,\hat{\boldsymbol{n}}(t) \end{cases}$$

$$\mathbf{U}(t) = ?$$

$$\begin{cases} \boldsymbol{R}_t = \mathbf{X}_t^2 - \mathbf{X}_t^1 \\ L = \text{characteristic scale of the flow} \end{cases}$$

(1) (semi) Heuristic policies

Surfing policy* - derivation

- Approximate linearly the underlying flow, $\boldsymbol{v}(\mathbf{X}_t^2, t)$;

$$\dot{\mathbf{X}}_t^2 = \boldsymbol{v}_{t_0} + (\nabla \boldsymbol{v})_{t_0} \cdot \left(\mathbf{X}_{\tau_s}^2 - \mathbf{X}_{t_0}^2\right) + \left(\frac{\partial \boldsymbol{v}}{\partial t}\right)_{t_0} (\tau_s - t_0) + \mathbf{U}(t)\,,$$

(Assuming constant gradients for a time τ_s)

- Find $\mathbf{U}(t)$ such that $-\mathbf{X}_{\tau_s}^2 \cdot \boldsymbol{R}_{t_0}$ is maximum;

$$\hat{\boldsymbol{n}}(t) = -\frac{\left[e^{(\tau_s - t)\,\nabla \boldsymbol{v}_{t_0}}\right]^{\mathrm{T}} \cdot \boldsymbol{R}_{t_0}}{\left\|\left[e^{(\tau_s - t)\,\nabla \boldsymbol{v}_{t_0}}\right]^{\mathrm{T}} \cdot \boldsymbol{R}_{t_0}\right\|}$$

(Assuming constant the direction $\boldsymbol{R}_{t_0}$ for a time τ_s)

↓

$\|\boldsymbol{R}_t\| \gg L$

- Numerically optimize the free parameter τ_s.

*Monthiller, Rémi, et al. **Surfing on Turbulence: A Strategy for Planktonic Navigation**. *Phys. Rev. Lett.* ***129***, *064502* (2022)

$$\begin{cases} \dot{\mathbf{X}}_t^1 = \boldsymbol{v}(\mathbf{X}_t^1, t) \\ \dot{\mathbf{X}}_t^2 = \boldsymbol{v}(\mathbf{X}_t^2, t) + \mathbf{U}(t) \\ \mathbf{U}(t) = V_s\, \hat{\boldsymbol{n}}(t) \end{cases}$$

$$\mathbf{U}(t) = ?$$

$$\begin{cases} \boldsymbol{R}_t = \mathbf{X}_t^2 - \mathbf{X}_t^1 \\ L = \text{characteristic scale of the flow} \end{cases}$$

(1) (semi) Heuristic policies

Perturbative policy - derivation

- Consider linearity between the two agents, i.e., $\boldsymbol{v}(\mathbf{X}_t^2, t) \simeq \boldsymbol{v}(\mathbf{X}_t^1, t) + \nabla \boldsymbol{v}_t\, \boldsymbol{R}_t$, $\longrightarrow \|\boldsymbol{R}_t\| \ll L$

$$\dot{\boldsymbol{R}}_t = \nabla \boldsymbol{v}_t \boldsymbol{R}_t + \mathbf{U}(t) \quad \rightarrow \quad \boldsymbol{R}_{\tau_p} = \underbrace{e^{[(\nabla \boldsymbol{v})_{t_0} \tau_p]} \boldsymbol{R}_{t_0}} + V_s \int_{t_0}^{t_0+\tau_p} dt\, e^{[(\nabla \boldsymbol{v})_{t_0} (\tau_p - t)]}\, \hat{\boldsymbol{n}}(t) \;;$$

(Assuming constant gradients for a time τ_p)

- Find $\mathbf{U}(t)$ such that $\boldsymbol{R}_{\tau_p} \cdot \boldsymbol{R}_{\tau_p}^{free}$ is minimum;

$$\hat{\boldsymbol{n}}(t) = -\frac{\left[e^{(\tau_p - t)\,\nabla \boldsymbol{v}_{t_0}}\right]^{\mathrm{T}} \cdot e^{(\nabla \boldsymbol{v})_{t_0} \tau_p} \cdot \boldsymbol{R}_{t_0}}{\left\|\left[e^{(\tau_p - t)\,\nabla \boldsymbol{v}_{t_0}}\right]^{\mathrm{T}} \cdot e^{(\nabla \boldsymbol{v})_{t_0} \tau_p} \cdot \boldsymbol{R}_{t_0}\right\|}$$

- Numerically optimize the free parameter τ_p.

(2) **Optimal Control theory** – Pontryagin minimum principle

state variables *control variables*

Minimize $J = C_F\left(\boldsymbol{x}(t_f)\right) + \int_{t_0}^{t_f} dt\, [L(\boldsymbol{x}(t), \boldsymbol{u}(\boldsymbol{x},t), t)]$

performance index

Imposing $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},t)$

and other possible constraints,

e.g.: $\begin{cases} \boldsymbol{x}(t_0) = \boldsymbol{x}_\star\,, \quad \boldsymbol{x}(t_0) \le \boldsymbol{x}_\star\,, \\ \|\boldsymbol{u}(t)\|^2 = 1\,, \|\boldsymbol{u}(t)\|^2 \le 1\,, \text{exc.} \end{cases}$

- **Model based** and analytical tool
- Perfect knowledge required

$\|\boldsymbol{R}_{t_0}\| \sim \frac{V_s}{\lambda_{\text{lyapunov}}} \leftarrow$ border of controllability

In our case:

$\|\boldsymbol{R}^\star\| = \|\boldsymbol{R}_{t_0}\|/100$

capture's distance

$(*)\begin{cases} \dot{\mathbf{X}}_t^1 = \boldsymbol{v}(\mathbf{X}_t^1) \\ \dot{\mathbf{X}}_t^2 = \boldsymbol{v}(\mathbf{X}_t^2) + \mathbf{U}(t) \\ \mathbf{U}(t) = V_s\, \hat{\boldsymbol{n}}(t) \end{cases}$

Minimize $J = \left\|\boldsymbol{R}_{t_f}\right\|^2 + c\int_{t_0}^{t_f} dt\, \theta(\|\boldsymbol{R}_t\|^2 - \|\boldsymbol{R}^\star\|^2)$

Imposing $(*)$ and the control constraint $\|\hat{\boldsymbol{n}}(t)\|^2 = 1$

Minimize trajectories' separation

Minimize time of arrival at the desired distance

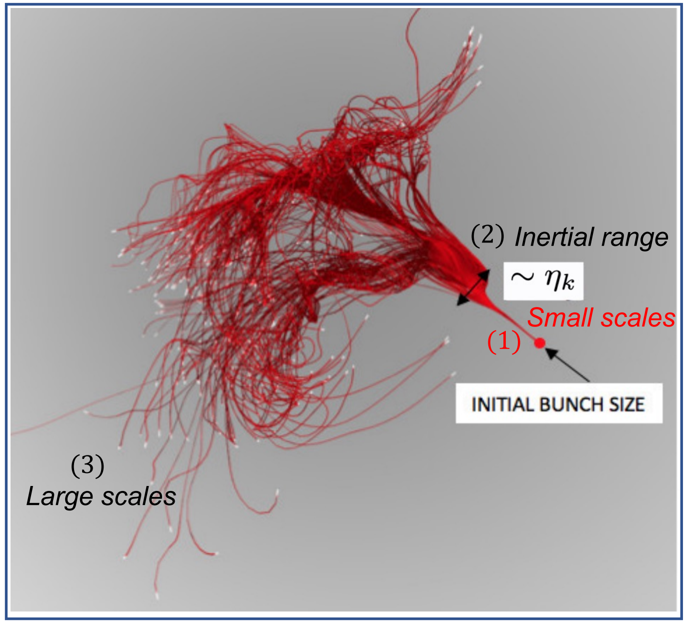
Optimal Control vs heuristic policies at **small scales**

Velocity field*

3D Direct Numerical Simulations $N = 1024^3$

NSEs: $\begin{cases} \partial_t \boldsymbol{v} = -\nabla p - (\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u} + \nu\,\nabla^2\boldsymbol{u} + \mathbf{F} \\ \nabla\cdot\boldsymbol{v} = 0 \end{cases}$

homogeneous and isotropic forcing



DNS parameters

$\eta_k = 0.0043$

$\tau_\eta = 0.023$

$Re \simeq 17000$

(1) **Linear regime** $\|\boldsymbol{R}_{t_0}\| < \eta_k$

$$\begin{cases} \dot{\mathbf{X}}_t^1 = \boldsymbol{v}(\mathbf{X}_t^1) \\ \dot{\mathbf{X}}_t^2 = \boldsymbol{v}(\mathbf{X}_t^2) + \mathbf{U}(t) \\ \mathbf{U}(t) = V_s\,\hat{\boldsymbol{n}}(t) \end{cases}$$

$$\boldsymbol{v}(\mathbf{X}_t^2) \simeq \boldsymbol{v}(\mathbf{X}_t^1) + \nabla\boldsymbol{v}_t\boldsymbol{R}_t$$

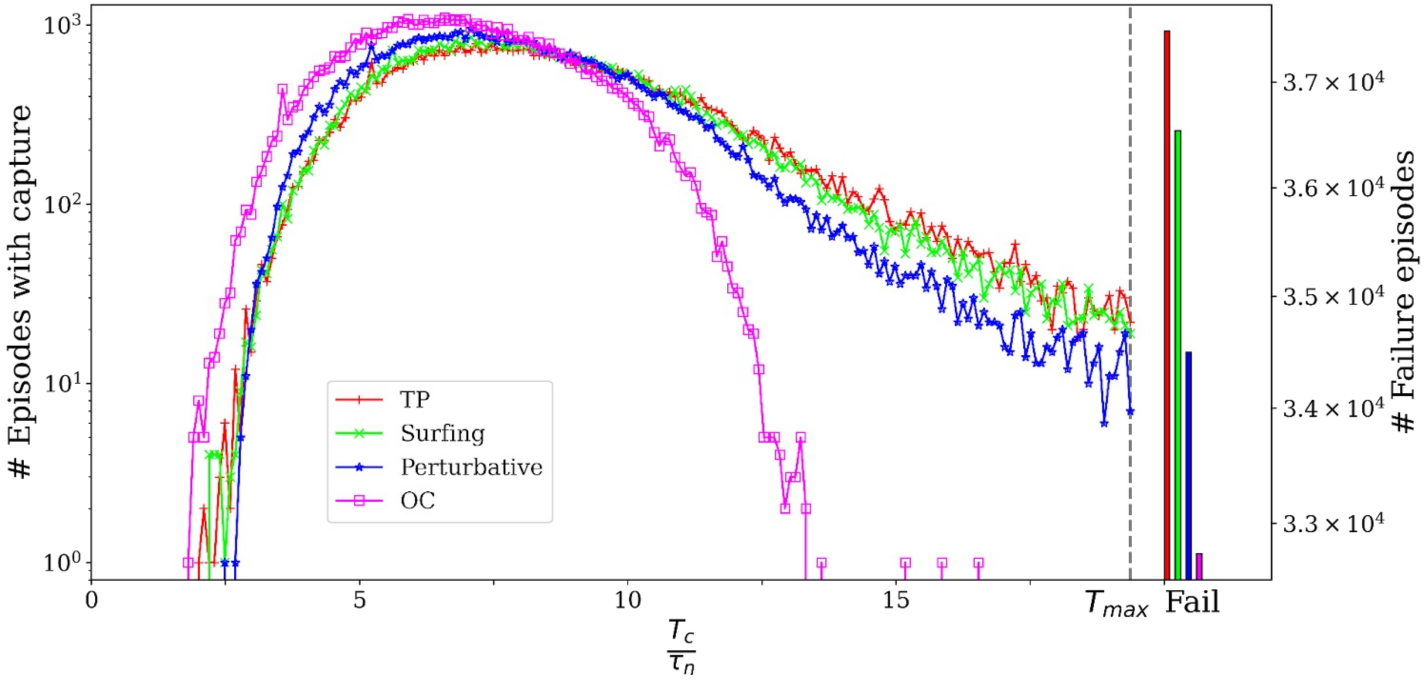
$$\dot{\boldsymbol{R}}_t = \nabla\boldsymbol{v}_t\boldsymbol{R}_t + \mathbf{U}(t)$$

*Buzzicotti et al. **Lagrangian statistics for Navier–Stokes turbulence under Fourier-mode reduction: fractal and homogeneous decimations.** *New J. Phys., 18 (11)* (2016), *p. 113047*

Optimal Control vs heuristic policies in linear regime

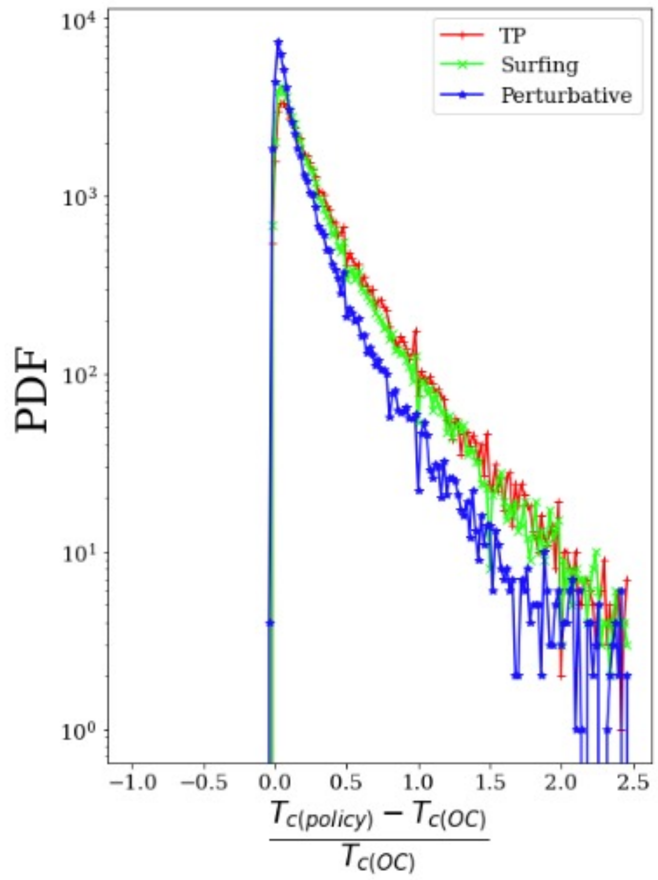
$\dot{\boldsymbol{R}}_t = \boldsymbol{\nabla v}_t \boldsymbol{R}_t + \mathbf{U}(t)$

$T_c =$ **Capture** time: (time of arrival at the desired distance)



PRELIMINARY UNPUBLISHED

PDF of normalized capture time



pros & cons

Optimal Control

+ It is optimized
- It is model based and needs perfect information from the environment
- It is sensitive to variation of the initial condition
- It is difficult to consider a decision time in the control variable

Heuristic policies

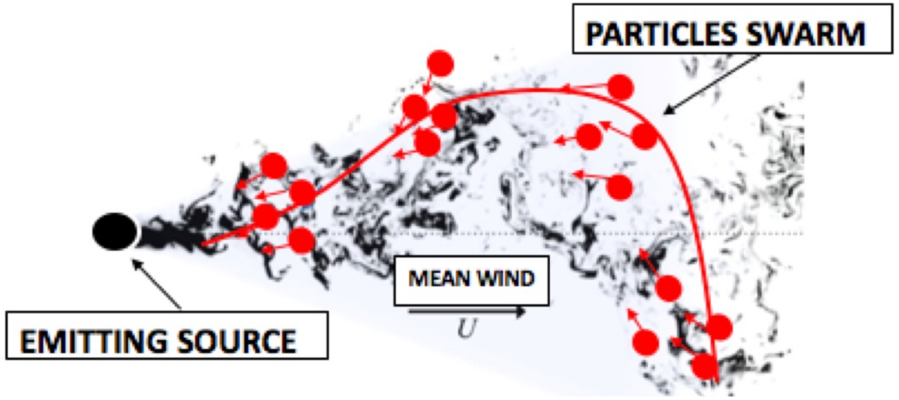
- They are not optimized
+ They need only partial information
+ They are stable wrt variation of the initial condition
+ They work also with a discrete decision time

Next step: Reinforcement Learning

+ It is optimized
+ It is model free
+ It needs partial information
- It is data-hungry

European Research Council
Established by the European Commission
Supporting top researchers from anywhere in the world

Conclusions



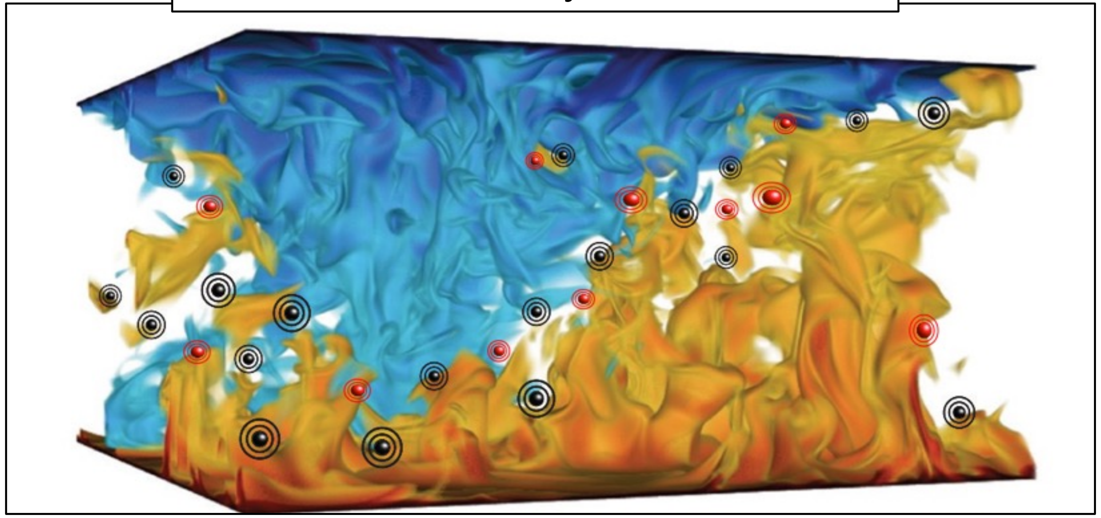
Open questions:

1. How to control a multi-agent system to minimize turbulent dispersion in realistic geophysical flows (beyond the linear regime) ?
2. Can we identify the key degrees-of-freedom to control the agents’ trajectories (key flow structures)?
3. Are the agents able to collaborate with each-other during the navigation?

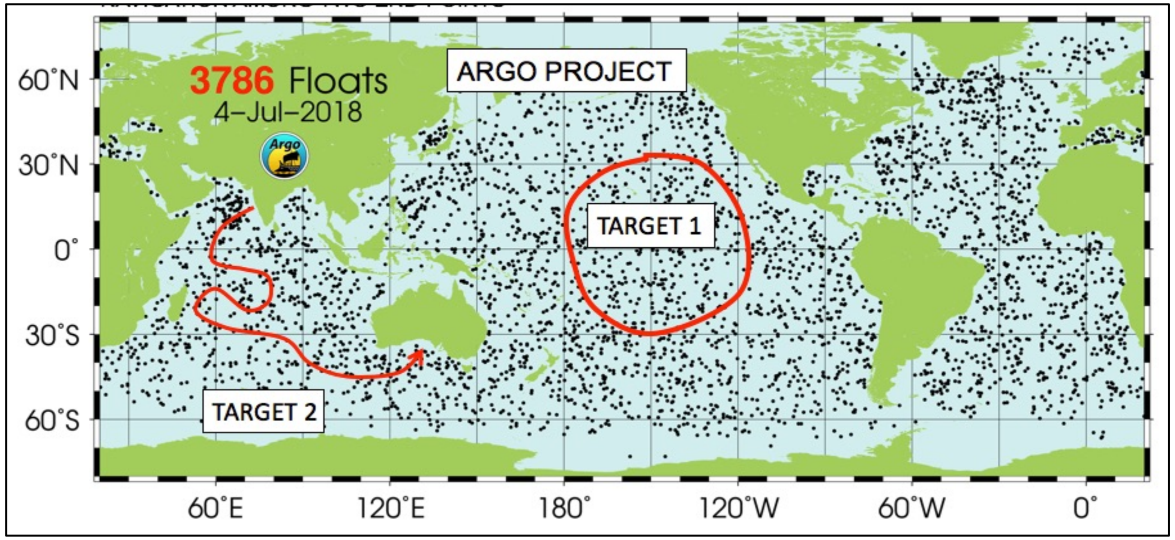
Tools:

- We can use RL to control autonomous swimmers in a realistic way (i.e., with a limited knowledge of the underlying flow - only local or instantaneous features);
- We can use OC as a benchmark to test the RL solutions.

Smart two-way feedback



http://stilton.tnw.utwente.nl/people/stevensr/afid.html



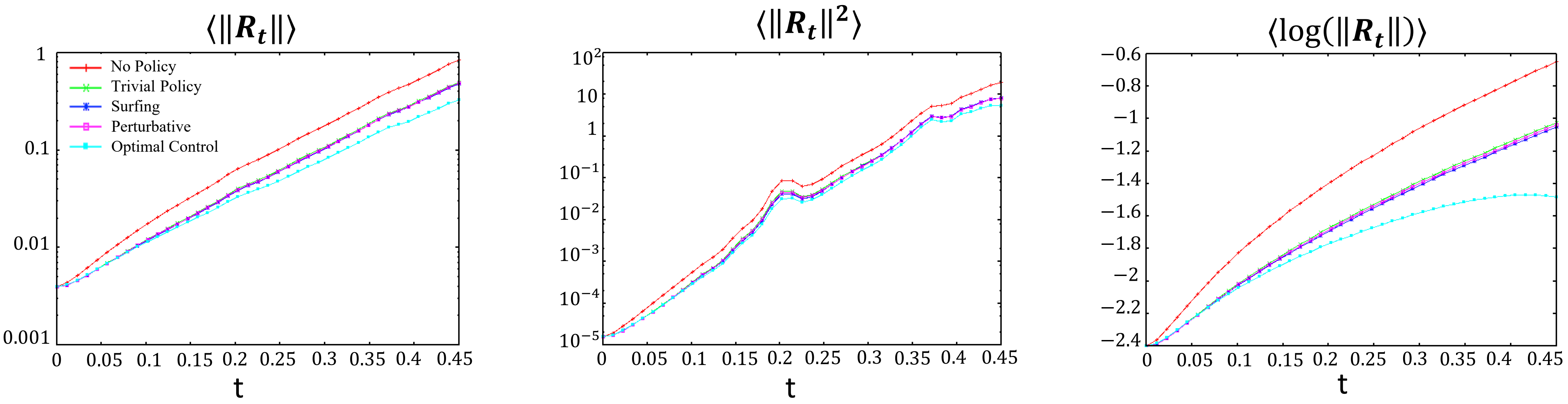
https://argo.ucsd.edu/

Backup slides

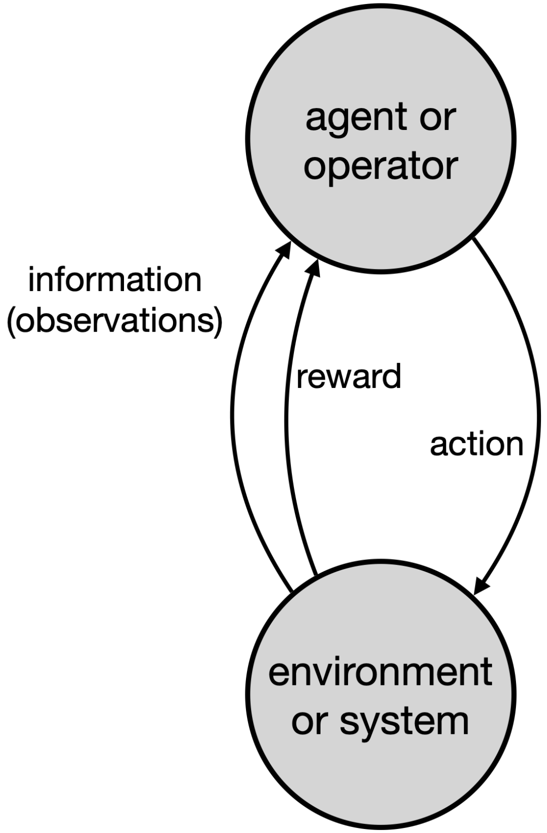
Optimal Control vs heuristic policies at **small scales**

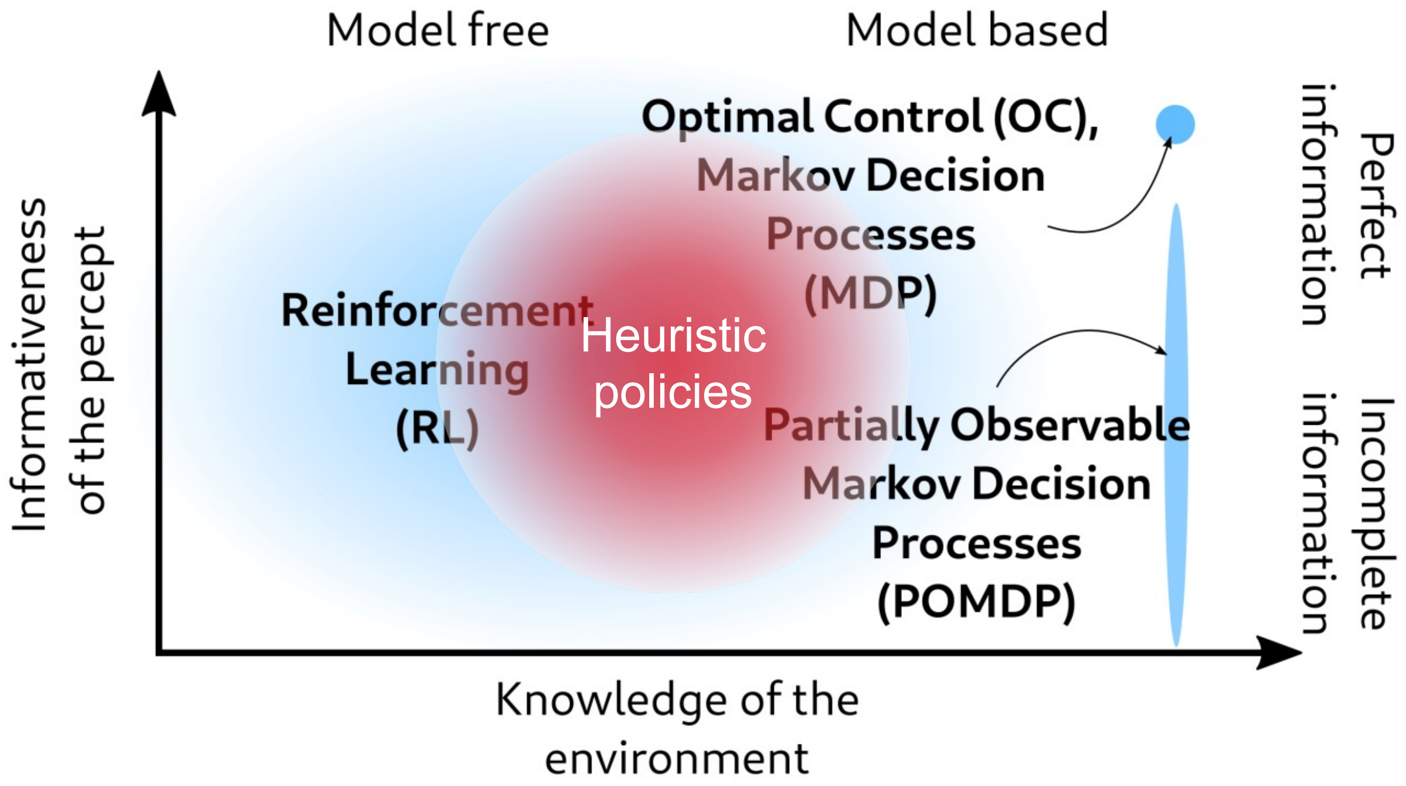
$$\dot{\boldsymbol{R}}_t = \boldsymbol{\nabla v}_t \boldsymbol{R}_t + \mathbf{U}(t)$$

Failures (no capture):



Control Theory

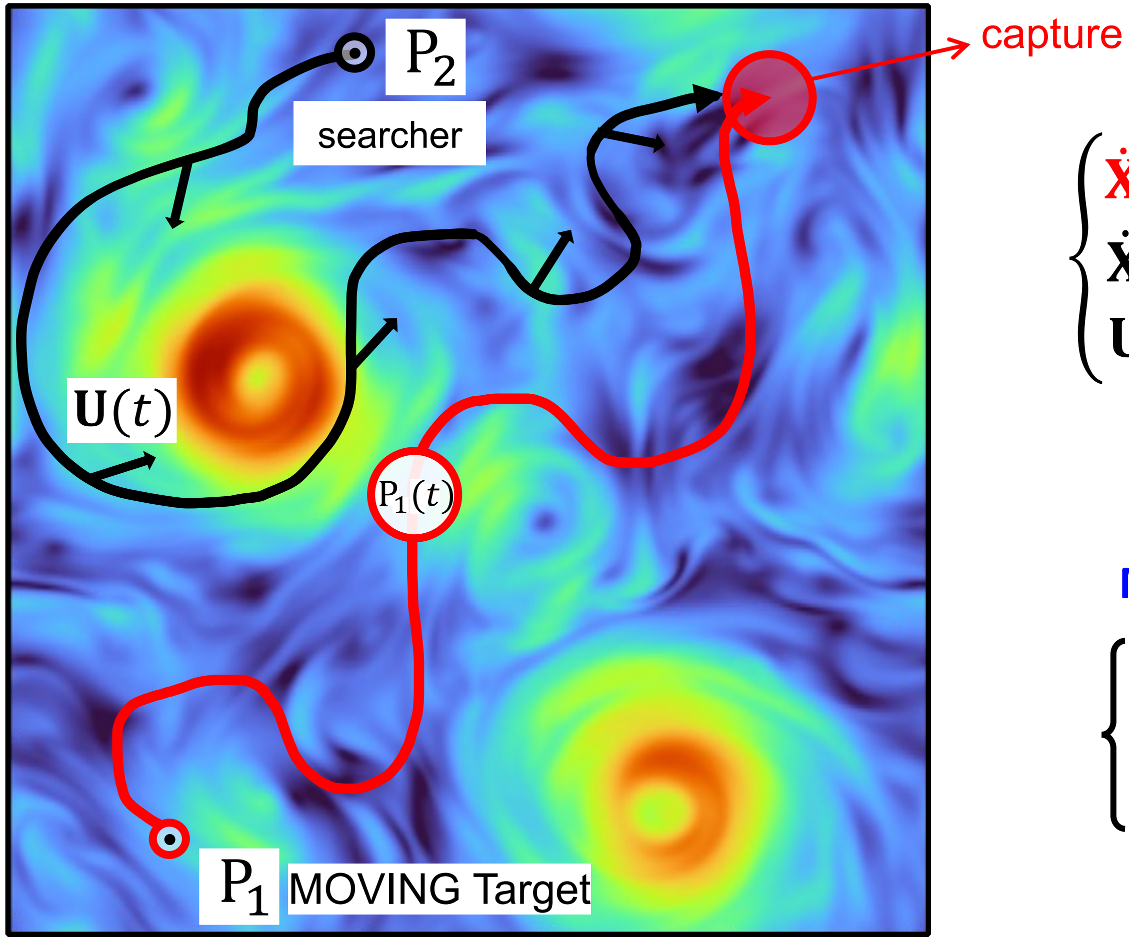




Tools:
Optimal Control (**OC**) theory
Reinforcement Learning (**RL**)
Heuristic policies

C. Calascibetta, L. Biferale, F. Borra, A. Celani, M. Cencini, **Optimal Control tools to minimize dispersion in chaotic flows**, (*in preparation*).

2 AGENTS



Goal: **minimize** the **separation** in a finite time horizon

Problem setup

$$\begin{cases} \dot{\mathbf{X}}_t^1 = v(\mathbf{X}_t^1) \\ \dot{\mathbf{X}}_t^2 = v(\mathbf{X}_t^2) + \mathbf{U}(t) \\ \mathbf{U}(t) = V_s\, \hat{\boldsymbol{n}}(t) \end{cases} \longrightarrow \mathbf{U}(t) = ?$$

$$\hat{\boldsymbol{n}}(t) = (\cos[\theta_t], \sin[\theta_t])$$

MAIN POINT: Different approaches for different range of scales:

$$\begin{cases} \boldsymbol{R}_t = \mathbf{X}_t^2 - \mathbf{X}_t^1 \\ L = \text{Characteristic scale of the flow} \end{cases}$$

$||\boldsymbol{R}_t|| \ll L$ Small scales

$||\boldsymbol{R}_t|| \gg L$ Large scales

Tools:
(1) **Heuristic policies**
(2) Optimal Control (**OC**) theory
(3) Reinforcement Learning (**RL**)

$$\begin{cases} \dot{\mathbf{X}}_t^1 = \boldsymbol{v}(\mathbf{X}_t^1, t) \\ \dot{\mathbf{X}}_t^2 = \boldsymbol{v}(\mathbf{X}_t^2, t) + \mathbf{U}(t) \\ \mathbf{U}(t) = V_s\, \hat{\boldsymbol{n}}(t) \end{cases}$$

$$\mathbf{U}(t) = ?$$

$$\boldsymbol{R}_t = \mathbf{X}_t^2 - \mathbf{X}_t^1$$

(1) (semi) Heuristic policies

Trivial Policy: constantly chooses the direction that points towards the moving target, $\hat{\boldsymbol{n}}(t) = -\hat{\boldsymbol{R}}_t$.

Surfing Policy*: - constant gradients for a time τ_s (free parameter);
- maximization of the searcher displacement along the $\boldsymbol{R}_t$ direction;
- good for slowly varying $\boldsymbol{R}_t$ (i.e. at large scales)

$$\hat{\boldsymbol{n}}(t) = -\frac{\left[e^{(\tau_s - t)\,\nabla \boldsymbol{v}_{t_0}}\right]^{\mathrm{T}} \cdot \boldsymbol{R}_{t_0}}{\left\| \left[e^{(\tau_s - t)\,\nabla \boldsymbol{v}_{t_0}}\right]^{\mathrm{T}} \cdot \boldsymbol{R}_{t_0} \right\|} .$$

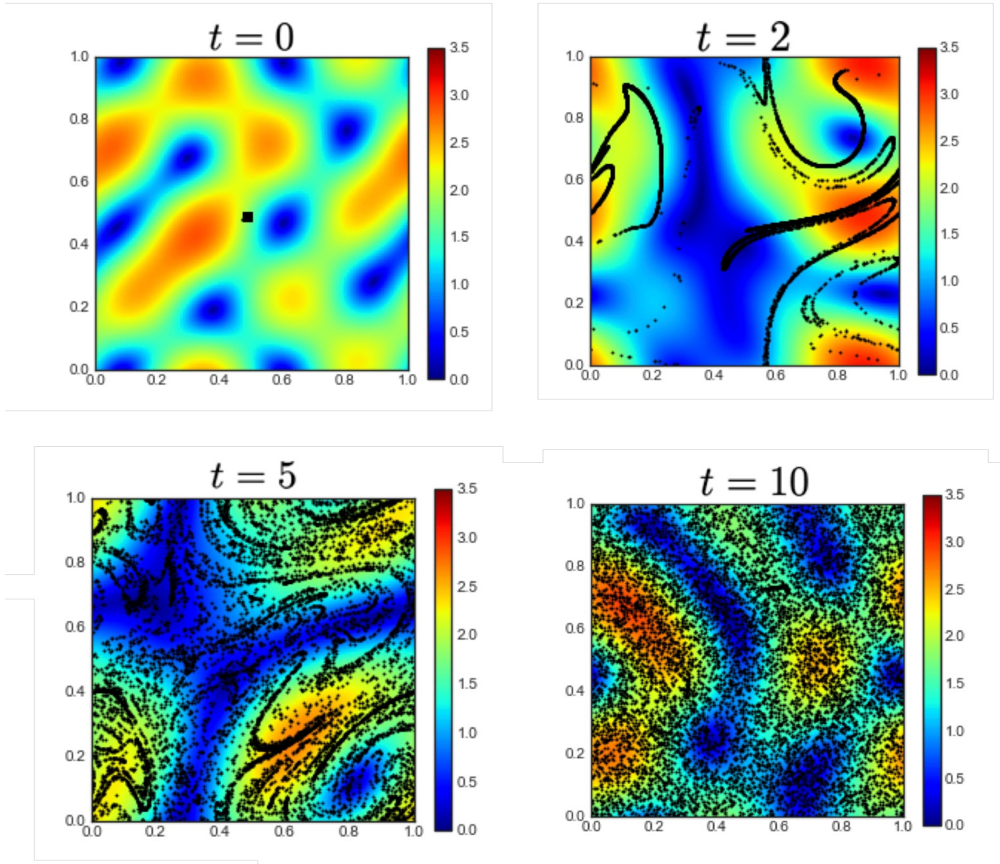
Perturbative Policy: - 0th order OC with constant gradients for a time τ_p (free parameter) ;
- valid at small scales

$$\hat{\boldsymbol{n}}(t) = -\frac{\left[e^{(\tau_p - t)\,\nabla \boldsymbol{v}_{t_0}}\right]^{\mathrm{T}} \cdot e^{(\nabla \boldsymbol{v})_{t_0} \tau_p} \cdot \boldsymbol{R}_{t_0}}{\left\| \left[e^{(\tau_p - t)\,\nabla \boldsymbol{v}_{t_0}}\right]^{\mathrm{T}} \cdot e^{(\nabla \boldsymbol{v})_{t_0} \tau_p} \cdot \boldsymbol{R}_{t_0} \right\|} .$$

*Monthiller, Rémi, et al. **Surfing on Turbulence: A Strategy for Planktonic Navigation**. *Phys. Rev. Lett.* ***129***, *064502* (2022)

Heuristic policies in a 2d stochastic flow

Dispersion of a bunch of particles



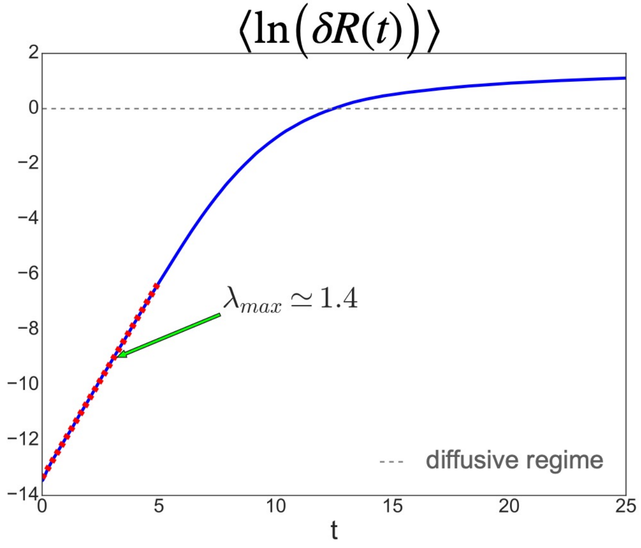
$$\boldsymbol{v}(x, y, t) = \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x}\right)$$

$$\psi(\boldsymbol{x}, t) = \sum_{\boldsymbol{k} \in \mathcal{K}} \left(A(\boldsymbol{k}, t) e^{i(\boldsymbol{k}\cdot\boldsymbol{x})} + \text{c.c.}\right)$$

Dove $\mathcal{K} = \{(k_s, 0), (\pm k_s, k_s), (0, k_s)\}$

Velocity field

- $A(\boldsymbol{k}, t)$ generated by an **Ornstein-Uhlenbeck** process
- $u_{rms} = 1$ (typical velocity)
- $L = \dfrac{2\pi}{k_s} = 1$ (characteristic scale)

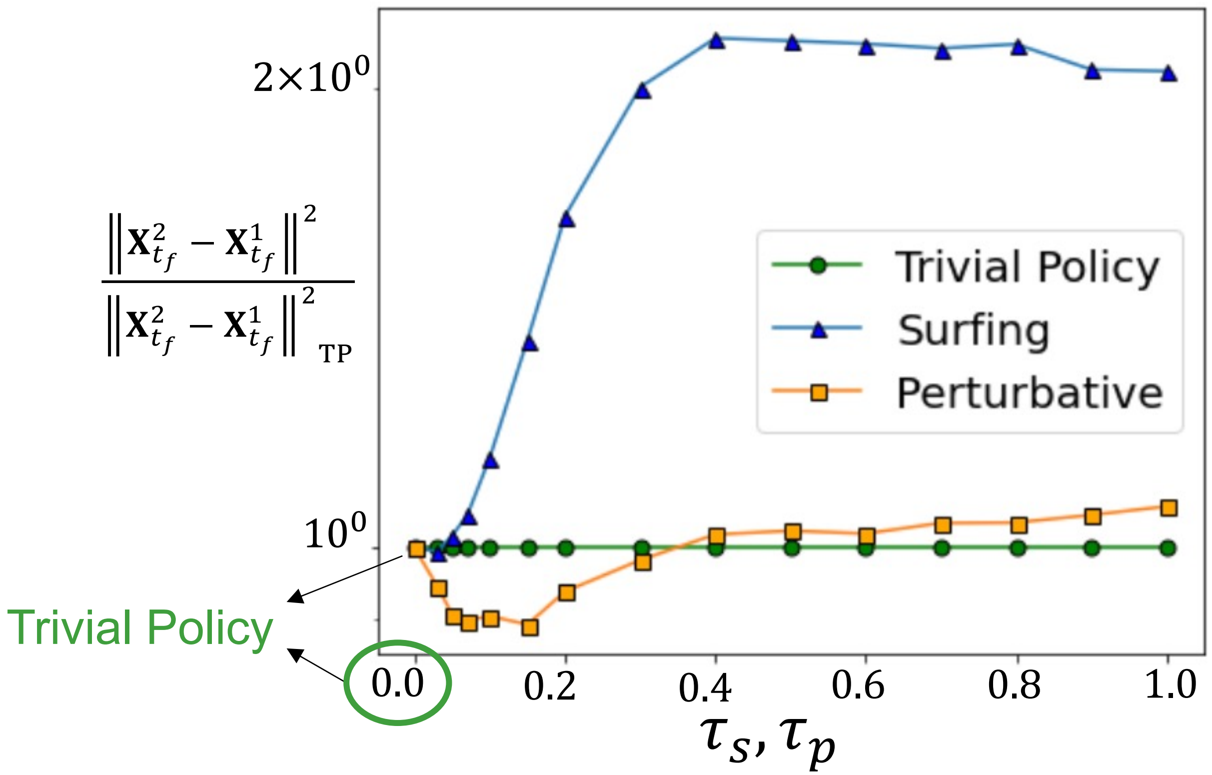


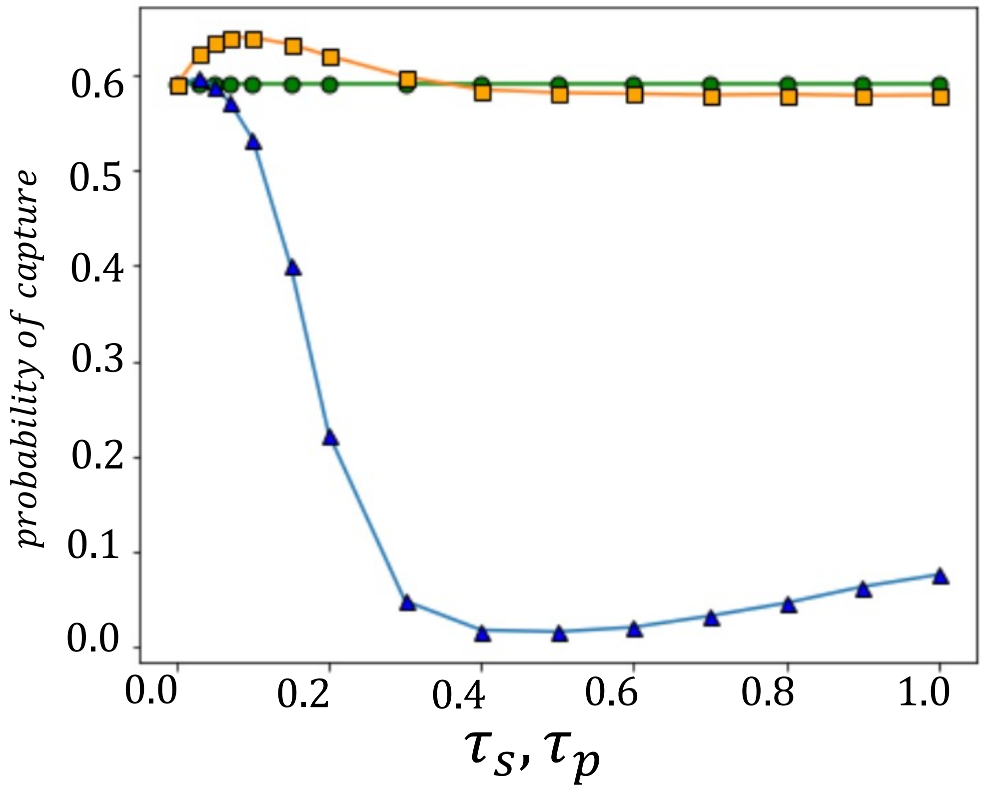
$$\lambda > 0$$

The Lagrangian dynamics is chaotic

Heuristic policies in a 2d stochastic flow

Performance at small scales $\|R_{t_0}\| \ll L$



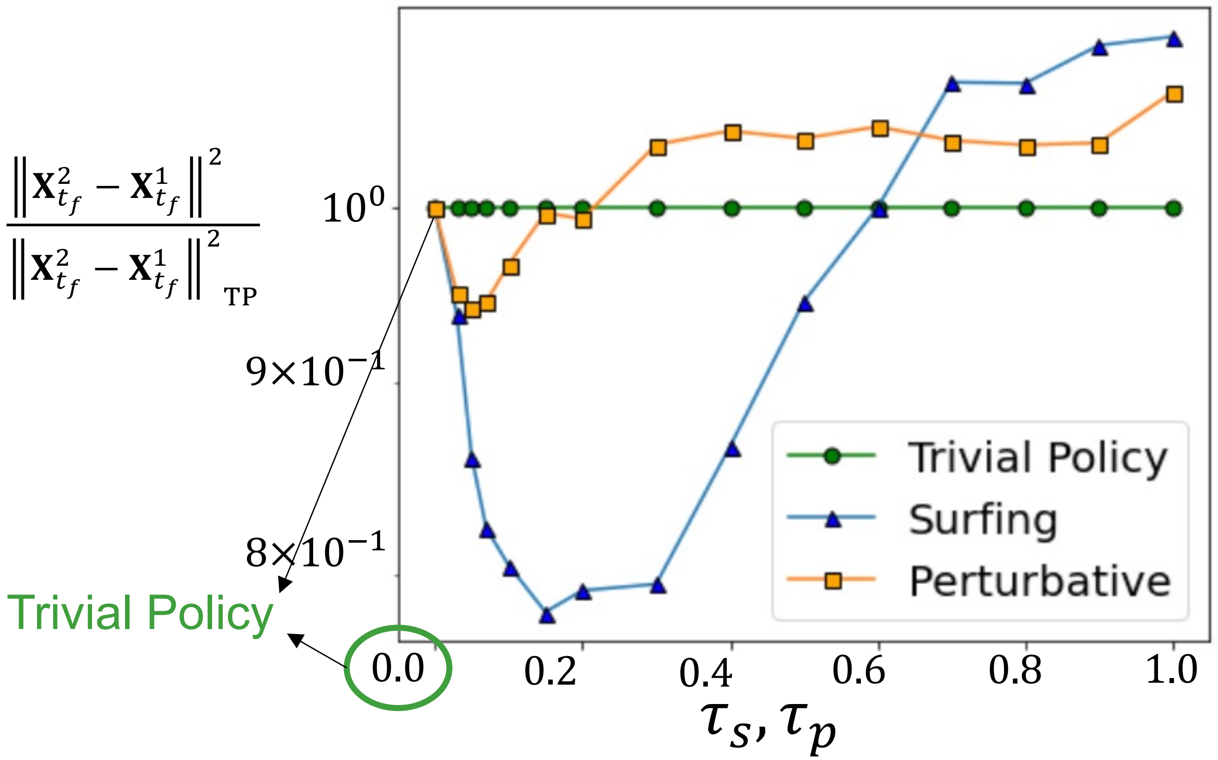


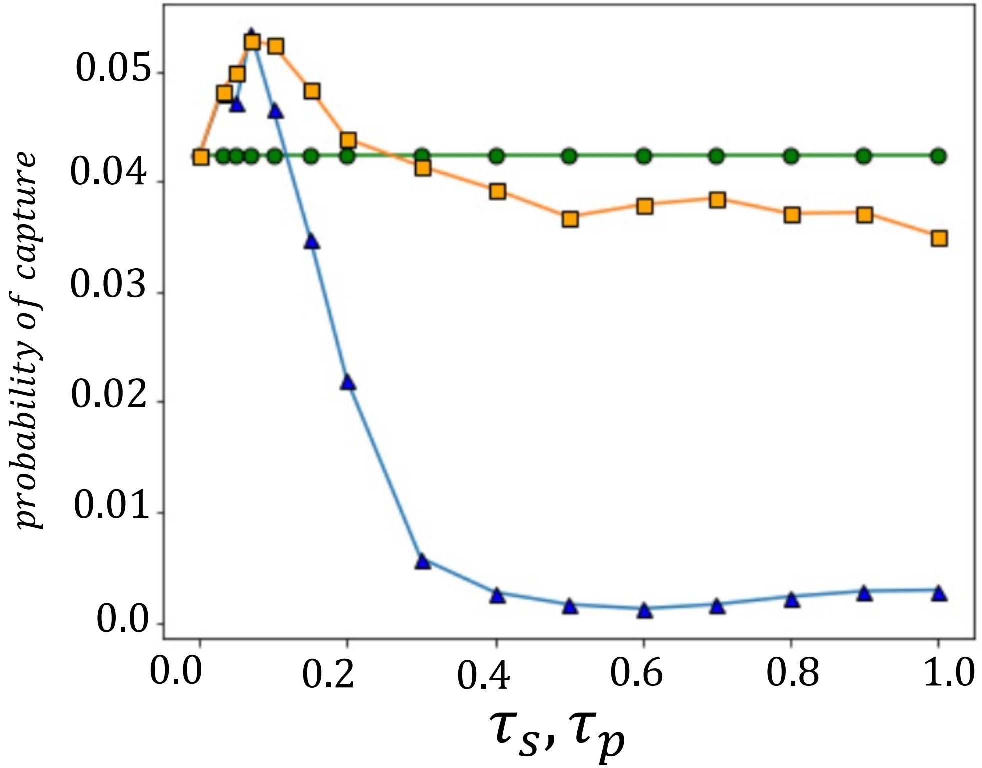
- The surfing policy performs bad at small scales.
- The perturbative policy performs well at small scales, $\exists\ best\ \tau_p \neq 0$.

There is a way to perform better than the Trivial Policy

Heuristic policies in a 2d stochastic flow

Performance at large scales $\left\|R_{t_0}\right\| \gg L$





- The surfing policy performs well at large scales, $\exists\ best\ \tau_s \neq 0$.
- The perturbative policy performs well at large scales, $\exists\ best\ \tau_p \neq 0$.

There is a way to perform better than the Trivial Policy

Optimal Control vs heuristic policies at **small scales**

Velocity field (double gyre flow*)

$$f(x,t) = a(t)x^2 + b(t)x$$

$$\phi(x,y,t) = A\sin(\pi f(x,t))\sin(\pi y), \qquad a(t) = \epsilon\sin(\omega t)$$

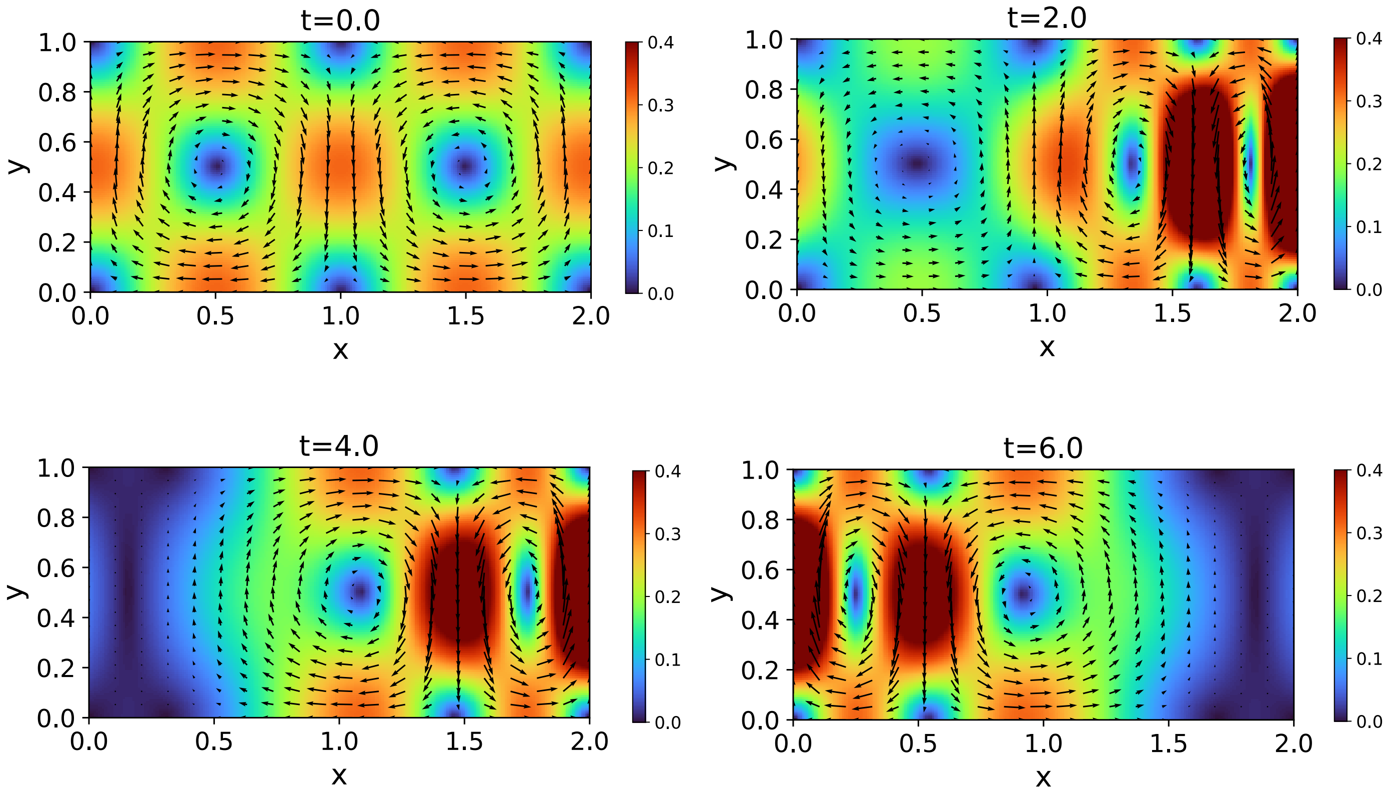
$$b(t) = 1 - 2\epsilon\sin(\omega t).$$

Linear regime

$$\begin{cases} \dot{\mathbf{X}}_t^1 = \boldsymbol{v}(\mathbf{X}_t^1) \\ \dot{\mathbf{X}}_t^2 = \boldsymbol{v}(\mathbf{X}_t^2) + \mathbf{U}(t) \\ \mathbf{U}(t) = V_s\,\hat{\boldsymbol{n}}(t) \end{cases}$$

$$\boldsymbol{v}(\mathbf{X}_t^2) \simeq \boldsymbol{v}(\mathbf{X}_t^1) + \nabla \boldsymbol{v}_t \boldsymbol{R}_t$$

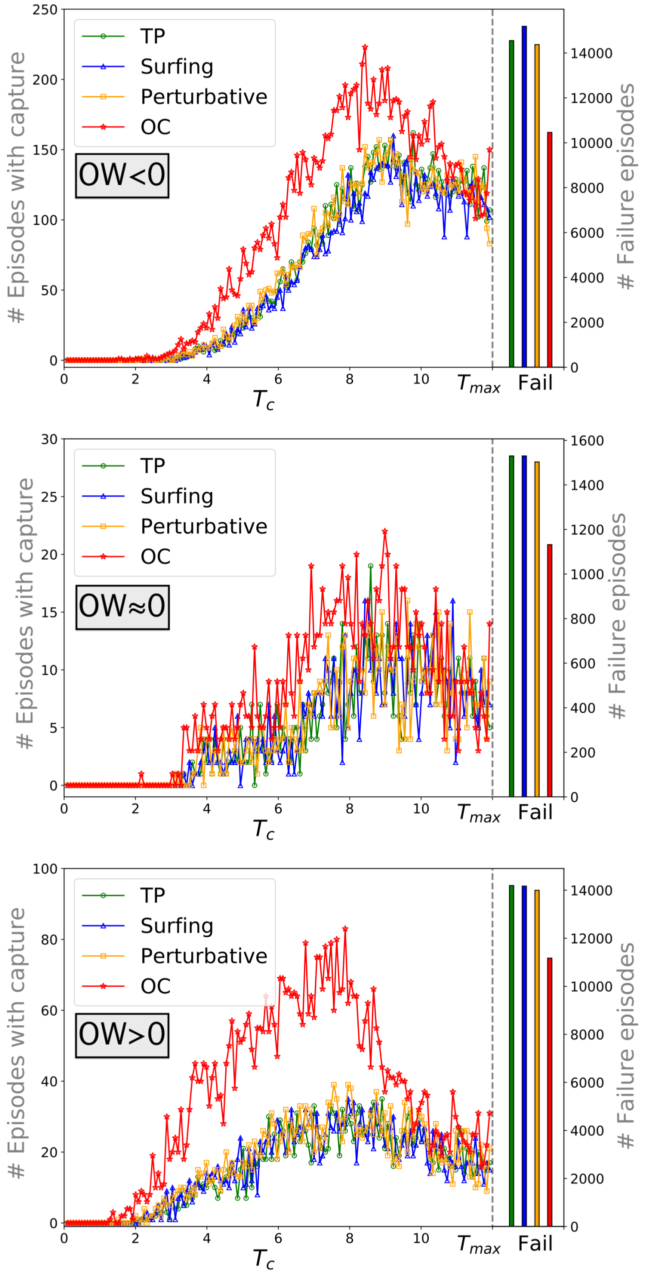
$$\dot{\boldsymbol{R}}_t = \nabla \boldsymbol{v}_t \boldsymbol{R}_t + \mathbf{U}(t)$$



*Krishna K, Song Z, Brunton SL. 2022 **Finite-horizon, energy-efficient trajectories in unsteady flows**. *Proc. R. Soc. A* **478**: 20210255.

Optimal Control vs heuristic policies in linear regime

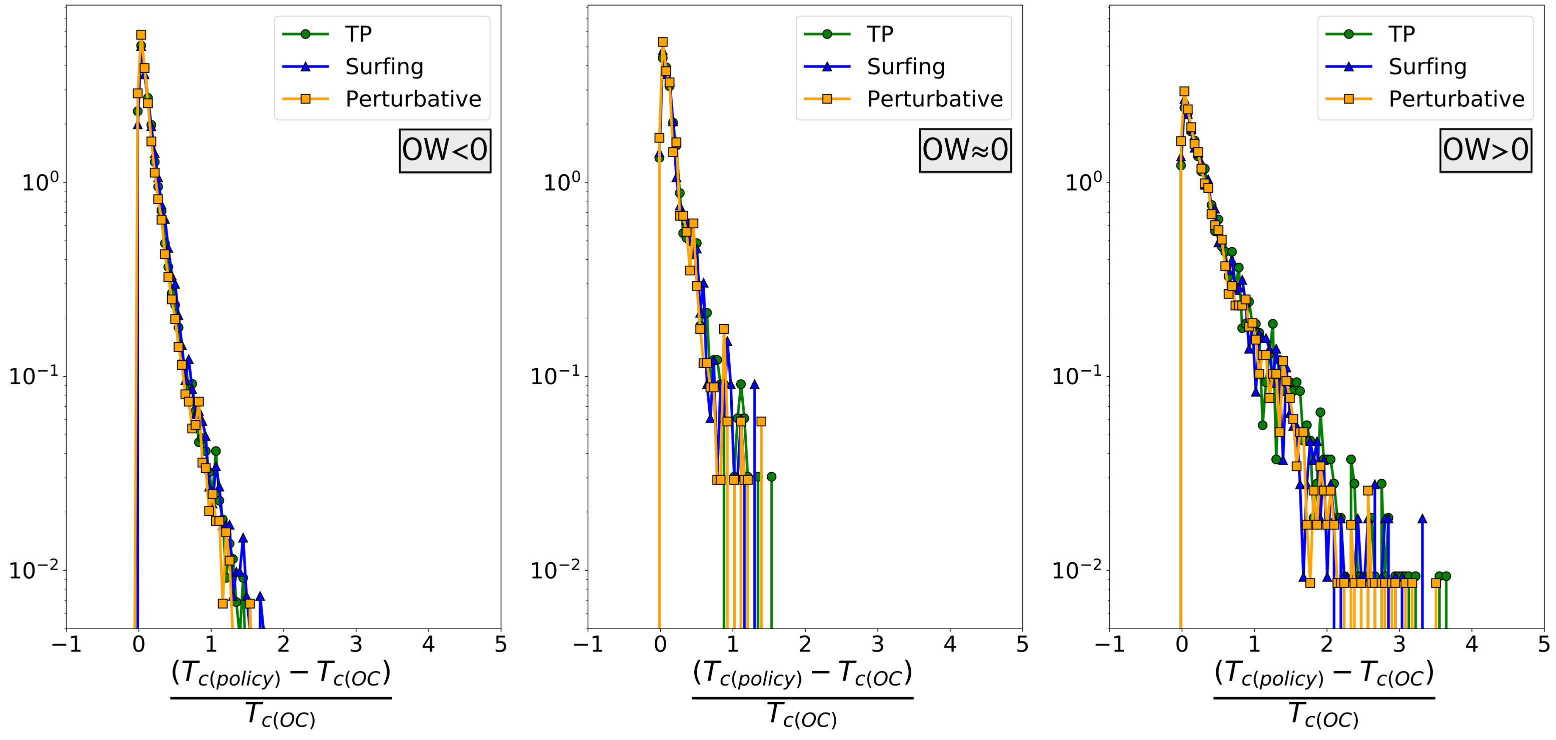
$$\dot{\boldsymbol{R}}_t = \boldsymbol{\nabla v}_t \boldsymbol{R}_t + \mathbf{U}(t)$$



T_c = **Capture** time: (time of arrival at the desired distance)

OW = Average of the Okubo Weiss parameter { < 0 vorticity dominated; > 0 strain dominated }

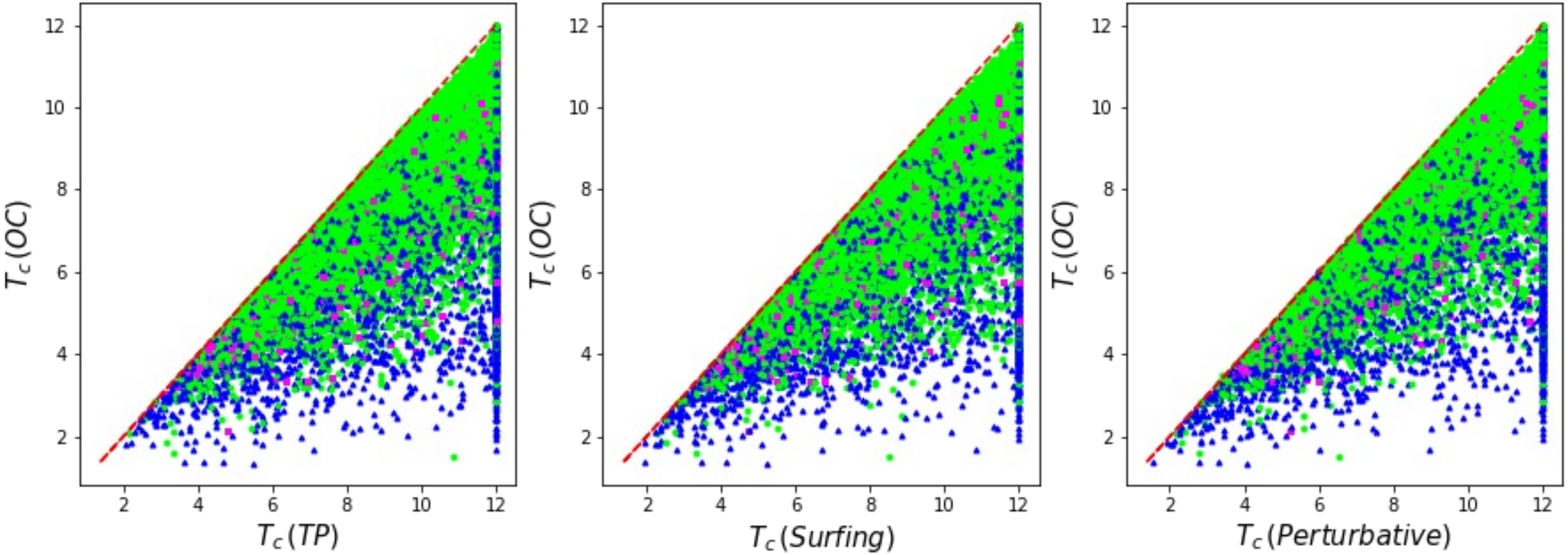
PDF of normalized capture time

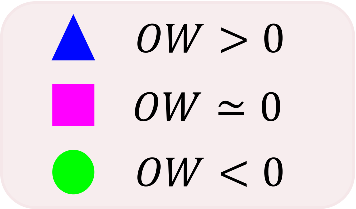


Optimal Control vs heuristic policies at **small scales**

$$\dot{\boldsymbol{R}}_t = \boldsymbol{\nabla v}_t \boldsymbol{R}_t + \mathbf{U}(t)$$

$T_c =$ **Capture** time: (time of arriving at the desired distance)

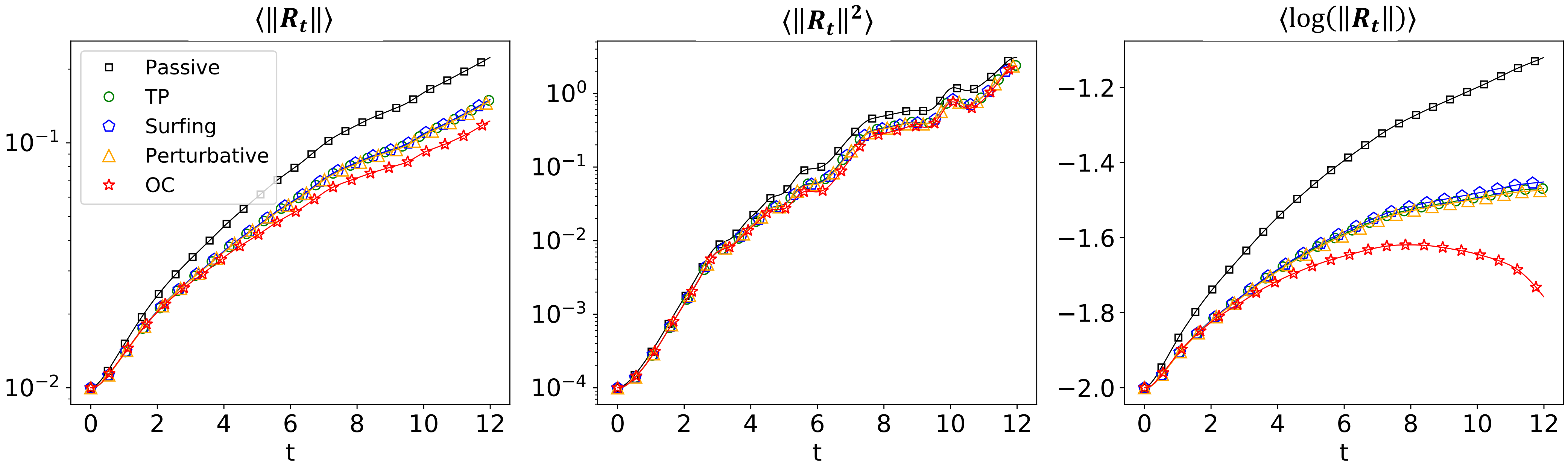




Optimal Control vs heuristic policies at **small scales**

$$\dot{\boldsymbol{R}}_t = \boldsymbol{\nabla v}_t \boldsymbol{R}_t + \mathbf{U}(t)$$

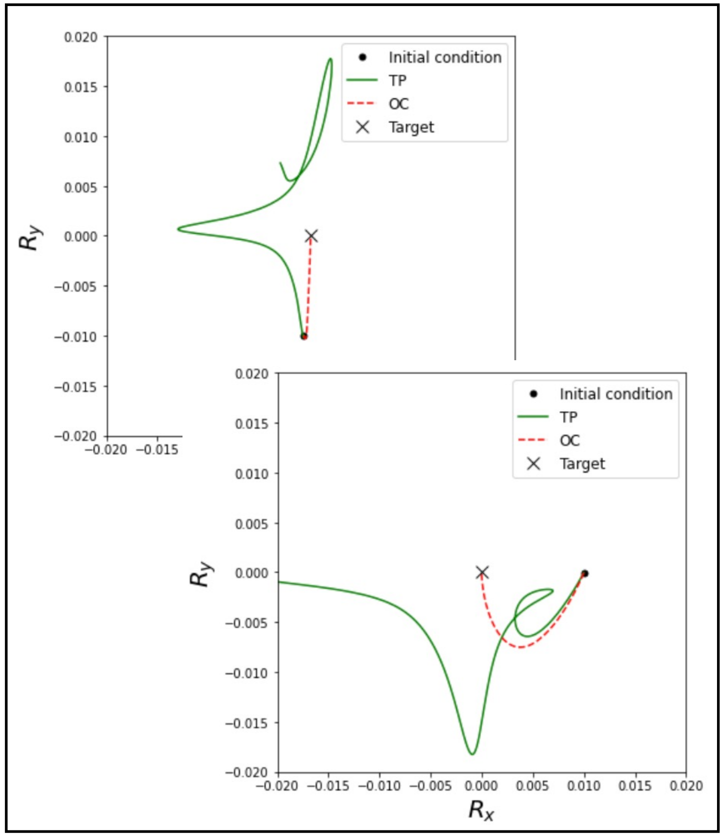
Failures (no capture):



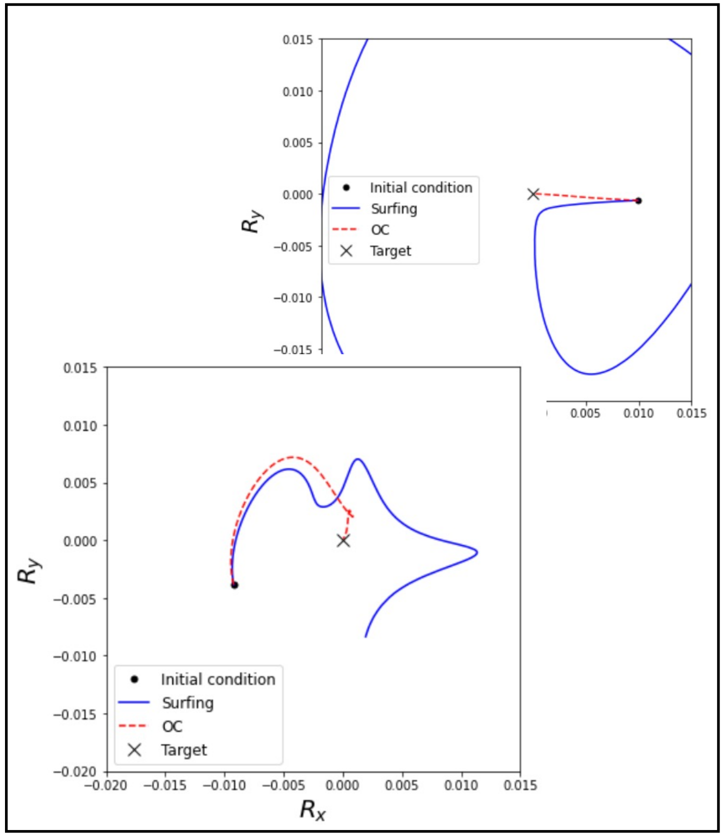
Trajectories examples

$$\dot{\boldsymbol{R}}_t = \nabla \boldsymbol{v}_t \boldsymbol{R}_t + \mathbf{U}(t)$$

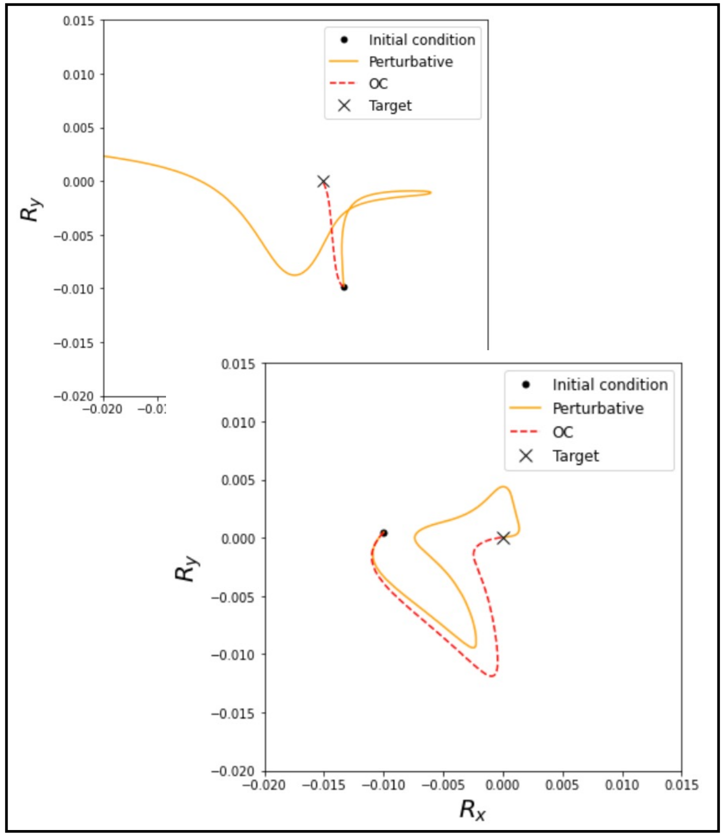
OC vs Trivial Policy



OC vs Surfing



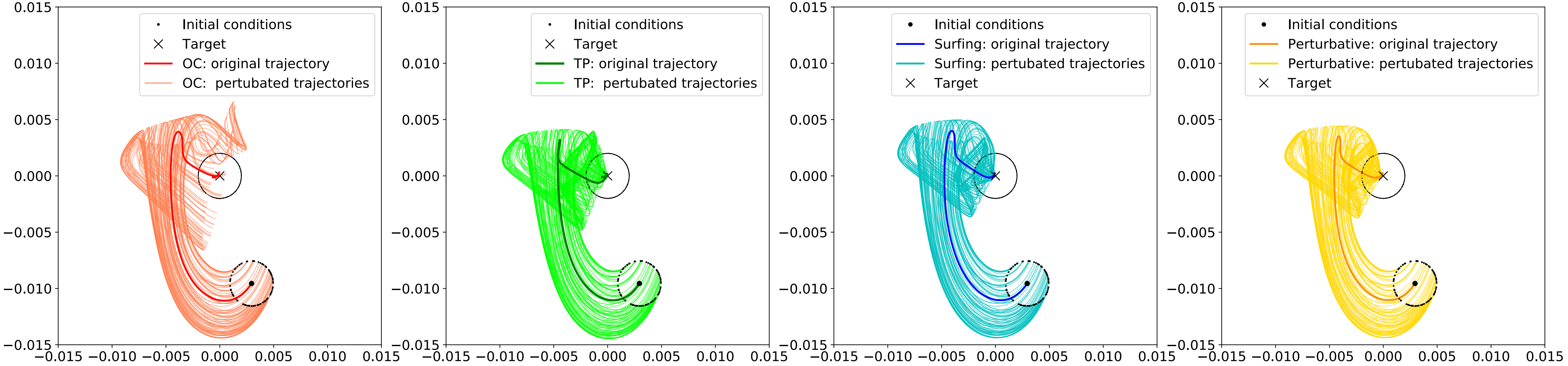
OC vs Perturbative

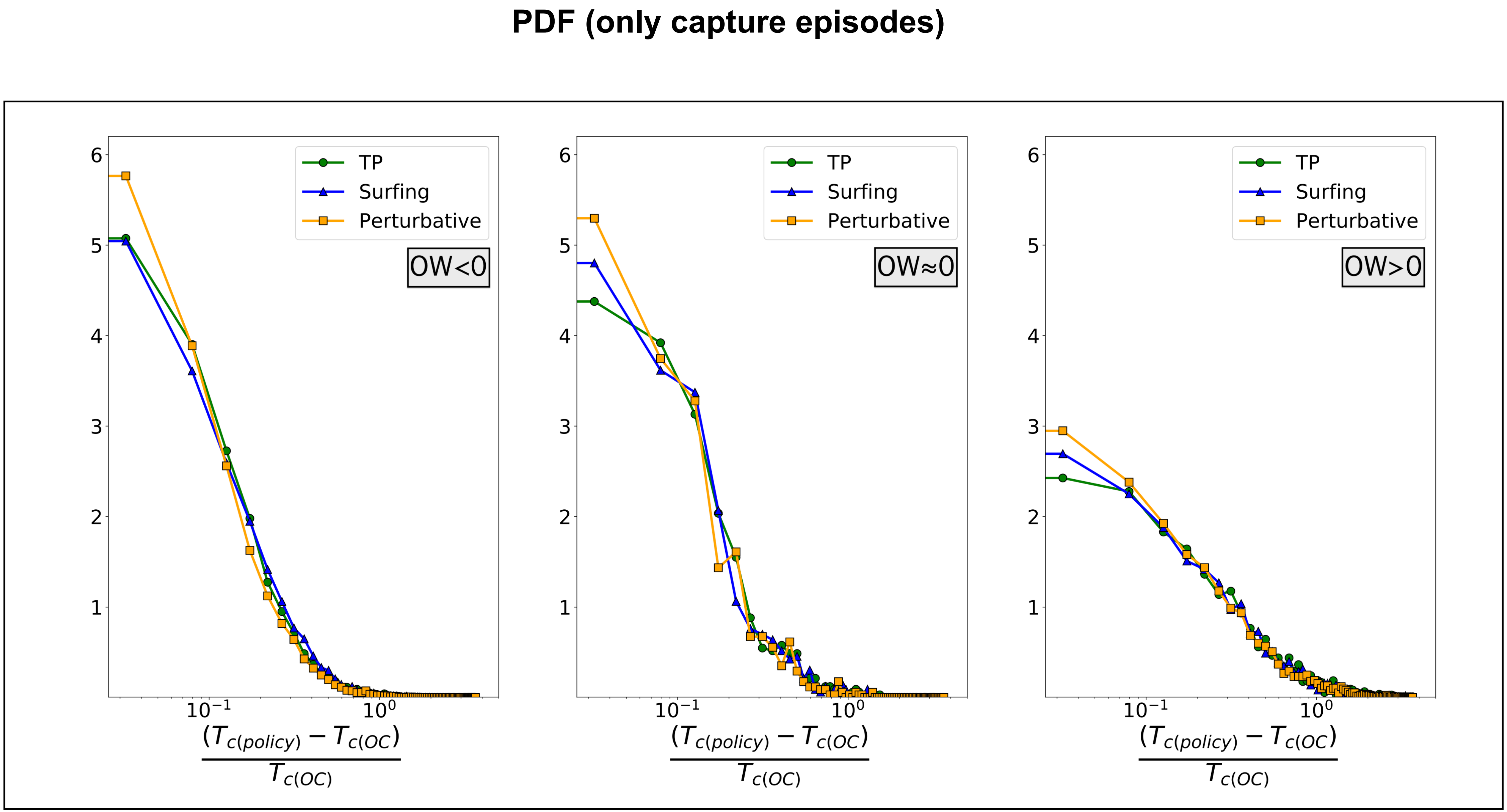


Policies' stability

$$\dot{\boldsymbol{R}}_t = \nabla \boldsymbol{v}_t \boldsymbol{R}_t + \mathbf{U}(t)$$

Perturbation of the initial condition




PDF (only capture episodes)
TP
Surfing
Perturbative
OW<0
OW≈0
OW>0
$\frac{(T_{c(policy)} - T_{c(OC)})}{T_{c(OC)}}$

2. **OC theory** – the basic idea behind the **Pontryagin minimum principle**

state variables *control variables*

Minimize $J = C_F\left(\boldsymbol{x}(t_f)\right) + \int_{t_0}^{t_f} dt\,[L(\boldsymbol{x}(t), \boldsymbol{u}(\boldsymbol{x}, t), t)]$

performance index *Lagrangian function*

Assuming $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t)$

and other possible constraints,

e.g.: $\begin{cases} \boldsymbol{x}(t_0) = \boldsymbol{x}_\star\,, \quad \boldsymbol{x}(t_0) \le \boldsymbol{x}_\star\,, \\ \|\boldsymbol{u}(t)\|^2 = 1\,, \|\boldsymbol{u}(t)\|^2 \le 1 \,, \text{exc.} \end{cases}$

Observe that this is a **constrained** minimization

Lagrangian multipliers

$$\tilde{J} = C_{t_f}\left(\boldsymbol{x}(t_f)\right) + \int_{t_0}^{t_f} dt\,[L(\boldsymbol{x}, \boldsymbol{u}, t) + \boldsymbol{\lambda}^{\mathrm{T}}(t) \cdot (\boldsymbol{f} - \dot{\boldsymbol{x}}) + \ \dots]$$

$$+\,\mu(t)(1 - \|\boldsymbol{u}(t)\|^2)$$

We impose minimum in $\boldsymbol{x}(\cdot)$, $\boldsymbol{u}(\cdot)$, $\boldsymbol{\lambda}(\cdot)$, i.e., $d\tilde{J} \le 0$:

$$\frac{\delta \tilde{J}}{\delta \boldsymbol{x}(t)} = 0 \ \Rightarrow \dot{\boldsymbol{\lambda}} = -\boldsymbol{\partial}_{\boldsymbol{x}} L - (\boldsymbol{\partial}_{\boldsymbol{x}} f)^{\mathrm{T}} \boldsymbol{\lambda}(t)\ ,$$

$$\frac{\delta \tilde{J}}{\delta \boldsymbol{x}(t_f)} = 0 \ \Rightarrow \boldsymbol{\lambda}(t_f) = \boldsymbol{\partial}_{\boldsymbol{x}} C_F\left(\boldsymbol{x}(t_f)\right),$$

$$\frac{\delta \tilde{J}}{\delta \boldsymbol{u}(\boldsymbol{x}, t)} = 0 \ \Rightarrow \boldsymbol{u}^\star(\boldsymbol{x}, t) = \frac{\boldsymbol{\partial}_{\boldsymbol{u}} L + (\boldsymbol{\partial}_{\boldsymbol{u}} \boldsymbol{f})^{\mathrm{T}} \boldsymbol{\lambda}(t)}{2\mu(t)}$$

Note: computationally heavy

It requires iterative searching with backward and forward integration such as to identify the optimal control

2. **OC theory** – the basic idea behind the **Pontryagin minimum principle**

Minimize $J = C_F\left(\boldsymbol{x}(t_f)\right) + \int_{t_0}^{t_f} dt\,[L(\boldsymbol{x}(t), \boldsymbol{u}(\boldsymbol{x},t), t)]$

performance index ... *Lagrangian function*

Assuming $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t)$

and other possible constraints,

e.g.: $\begin{cases} \boldsymbol{x}(t_0) = \boldsymbol{x}_\star\,, \quad \boldsymbol{x}(t_0) \leq \boldsymbol{x}_\star\,, \\ \|\boldsymbol{u}(t)\|^2 = 1\,, \|\boldsymbol{u}(t)\|^2 \leq 1\,, \text{exc.} \end{cases}$

We impose minimum in $\boldsymbol{x}(\cdot)$, $\boldsymbol{u}(\cdot)$, $\lambda(\cdot$

$\frac{\delta \tilde{J}}{\delta \boldsymbol{x}(t)} = 0 \Rightarrow \dot{\lambda} = -\partial_x L - (\partial_x f)^{\mathrm{T}} \lambda(t$

$\|\boldsymbol{R}_{t_0}\| \sim \frac{V_s}{\lambda_{lyapunov}} \leftarrow$ border of controllability

$\frac{\delta \tilde{J}}{\delta \boldsymbol{u}(x,t)} = 0 \Rightarrow \boldsymbol{u}^\star(\boldsymbol{x},t) = \frac{\partial_u L + (\partial_u f)^{\mathrm{T}} \lambda(t)}{2\mu(t)}$

In our case:

$$(*)\begin{cases} \dot{\mathbf{X}}_t^1 = \boldsymbol{v}(\mathbf{X}_t^1) \\ \dot{\mathbf{X}}_t^2 = \boldsymbol{v}(\mathbf{X}_t^2) + \mathbf{U}(t) \\ \mathbf{U}(t) = V_s\,\hat{\boldsymbol{n}}(t) \end{cases}$$

$\|\boldsymbol{R}^\star\| = \|\boldsymbol{R}_{t_0}\|/100$

capture's distance

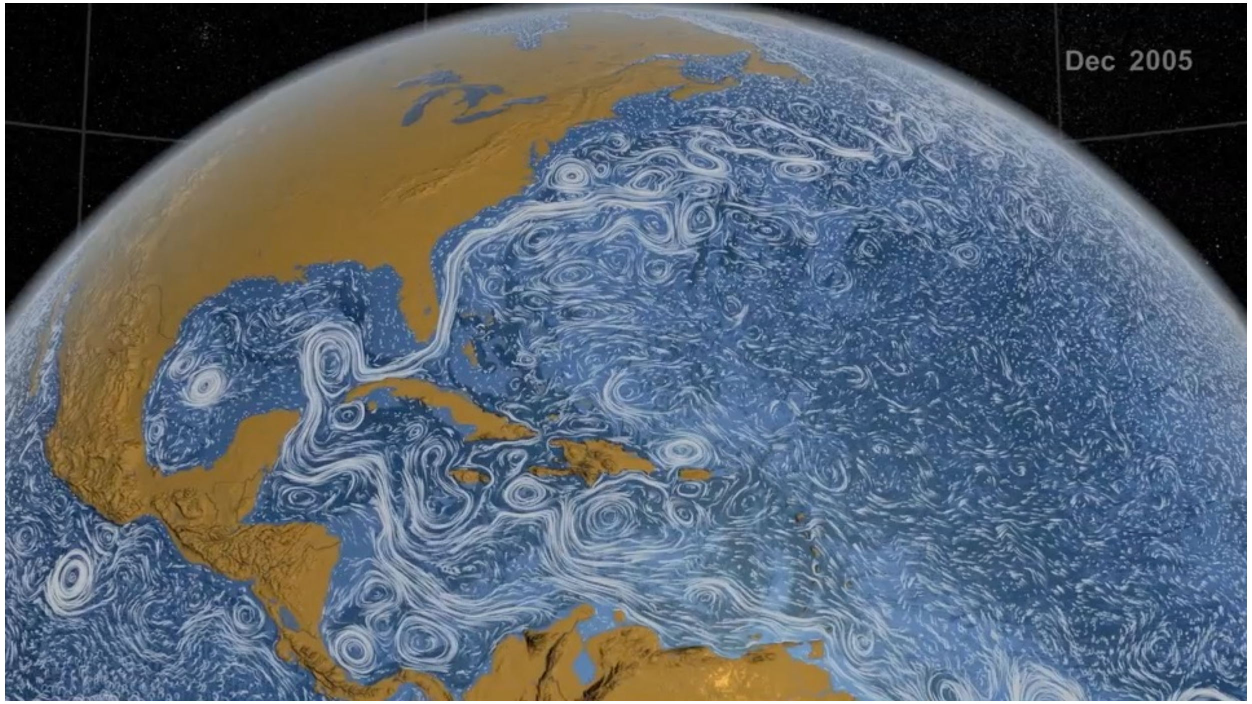
Minimize $J = \left\|\boldsymbol{R}_{t_f}\right\|^2 + c\int_{t_0}^{t_f} dt\,\theta(\|\boldsymbol{R}_t\|^2 - \|\boldsymbol{R}^\star\|^2)$,

assuming $(*)$ and the constraint $\|\hat{\boldsymbol{n}}(t)\|^2 = 1$,

Minimize the trajectories' separation

Minimize the time of arriving at the desired distance

Turbulent flows



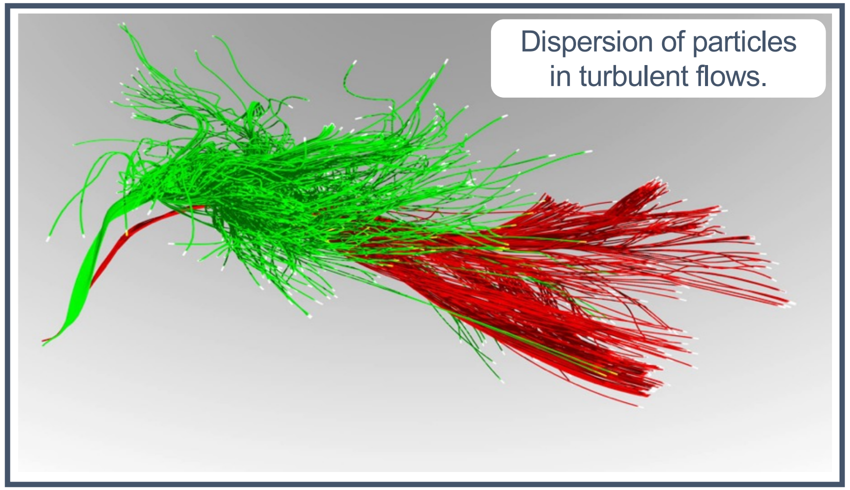
https://svs.gsfc.nasa.gov/3827

*How to exploit **coherent structures**?*

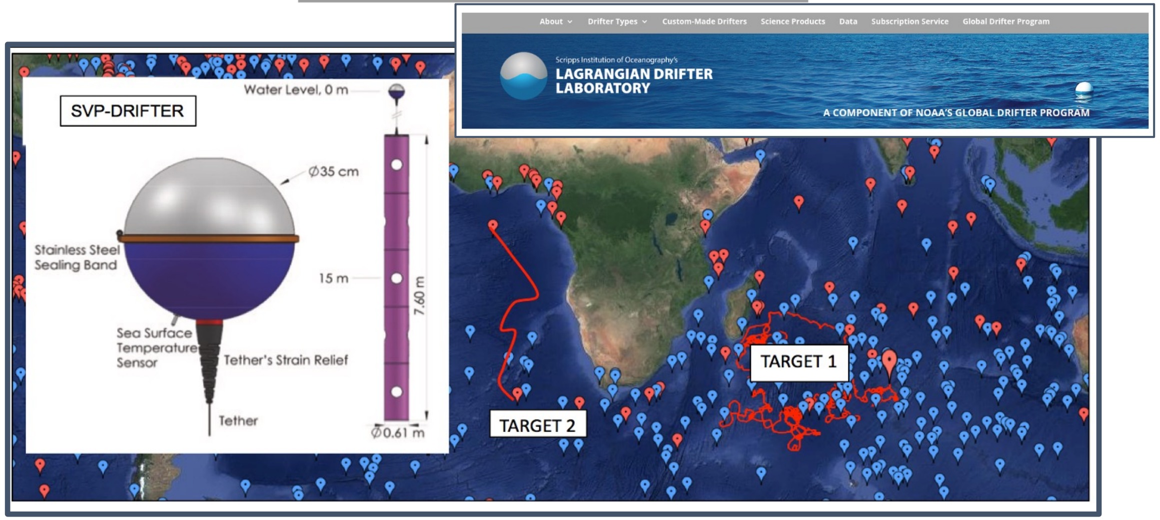
*How to avoid (or exploit) **intense fluctuations** when navigating inside the flow?*

*Which is the best **limited-control** to navigate in such complex flows?*

Theoretical interests:



Engineering applications:



Particles dispersion in complex flows

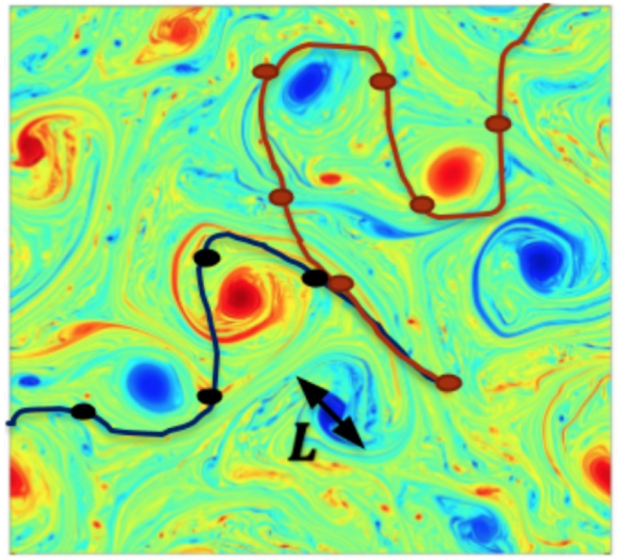
Lagrangian approach

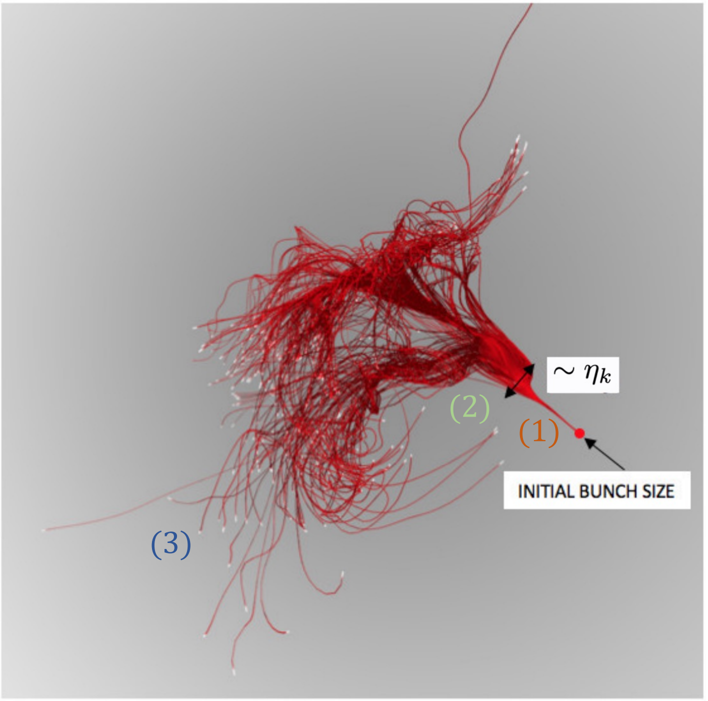
$$\dot{\mathbf{X}} = \boldsymbol{v}(\mathbf{X}_t, t)$$

Eq. of motion of a tracer

Trajectories separation:

$$\delta R_t = \|\mathbf{X}_t^2 - \mathbf{X}_t^1\|$$





(1) Dispersion at small scales

$$\delta R_t \sim \delta R_0 \, e^{\lambda t}$$

Lagrangian Chaos

$$\lambda = \lim_{t\to\infty} \lim_{\delta R_0 \to 0} \frac{1}{t} ln \frac{\delta R_t}{\delta R_0}$$

Lyapunov exponent

(2) Dispersion at intermediate scales

(Inertial range)

$$\langle(\delta R_t)^2\rangle \sim t^3$$

non-differentiable velocity field

If $Re \to \infty$ Fully Developed Turbulence
Richardson's Dispersion

(3) Dispersion at large scales

Advection
+
molecular diffusion

$$\langle(\delta R_t)^2\rangle \sim D^E t$$

effective diffusion