Optimal Control of Point-to-Point Navigation in Turbulent Time Dependent Flows Using Reinforcement Learning

Michele Buzzicotti[1], Luca Biferale[1(✉)], Fabio Bonaccorso[1,2],
Patricio Clark di Leoni[3], and Kristian Gustavsson[4]

[1] Department of Physics and INFN, University of Rome Tor Vergata,
Via della Ricerca Scientifica 1, 00133 Rome, Italy
{michele.buzzicotti,biferale,fabio.bonaccorso}@roma2.infn.it
[2] Center for Life Nano Science@La Sapienza,
Istituto Italiano di Tecnologia, 00161 Rome, Italy
[3] Department of Mechanical Engineering,
Johns Hopkins University, Baltimore, MD 21218, USA
pato@jhu.edu
[4] Department of Physics, University of Gothenburg,
41296 Gothenburg, Sweden
kristian.gustafsson@physics.gu.se

Abstract. We present theoretical and numerical results concerning the problem to find the path that minimizes the time to navigate between two given points in a complex fluid under realistic navigation constraints. We contrast deterministic Optimal Navigation (ON) control with stochastic policies obtained by Reinforcement Learning (RL) algorithms. We show that Actor-Critic RL algorithms are able to find quasi-optimal solutions in the presence of either time-independent or chaotically evolving flow configurations. For our application, ON solutions develop unstable behavior within the typical duration of the navigation process, and are therefore not useful in practice. We first explore navigation of turbulent flow using a constant propulsion speed. Based on a discretized phase-space, the propulsion direction is adjusted with the aim to minimize the time spent to reach the target. Further, we explore a case where additional control is obtained by allowing the engine to power off. Exploiting advection of the underlying flow, allows the target to be reached with less energy consumption. In this case, we optimize a linear combination between the total navigation time and the total time the engine is switched off. Our approach can be generalized to other setups, for example, navigation under imperfect environmental forecast or with different models for the moving vessel.

1 Introduction

Controlling and planning paths of small autonomous marine vehicles [16] such as wave and current gliders [10], active drifters [13], buoyant underwater explorers, and small swimming drones is important for many geo-physical [11] and

M. Baldoni and S. Bandini (Eds.): AIxIA 2020, LNAI 12414, pp. 223–234, 2021.
https://doi.org/10.1007/978-3-030-77091-4_14

engineering [3] applications. In realistic open environments, these vessels are affected by disturbances like wind, waves and ocean currents, characterized by unpredictable (chaotic) trajectories. Furthermore, active control is also limited by engineering and budget aspects as for the important case of unmanned drifters for oceanic exploration [6,18]. The problem of (time) optimal point-to-point navigation in a flow, known as Zermelo's problem [24], is interesting *per se* in the framework of Optimal Control Theory [5]. In this paper, we extend some of the results from a recent theoretical and numerical study [4], tackling Zermelo's problem for navigation in a two-dimensional fully turbulent flow in the presence of an inverse energy cascade, i.e. with chaotic, multi-scale and rough velocity distributions [1], see Fig. 1 for a summary of the problem. In such a flow, even for time-independent configurations, trivial or naive navigation policies can be extremely inefficient and ineffective if the set of actions by the vessel are limited. We show that an approach based on semi-supervised AI algorithms using actor-critic Reinforcement Learning (RL) [21] is able to find robust quasi-optimal stochastic policies that accomplish the task. Furthermore, we compare RL with solutions from Optimal Navigation (ON) theory [17] and show that the latter is of almost no practical use for the case of navigation in turbulent flows due to strong sensitivity to the initial (and final) conditions, in contrast to what happens for simpler advecting flows [20]. RL has shown to have promising potential to similar problems, such as the training of smart inertial particles or swimming objects navigating intense vortex regions [7–9].

We present here results from navigating one static snapshot of 2D turbulence (for time-dependent flows see [4]). In Fig. 1 we show a sketch of the setup. Our goal is to find trajectories (if they exist) that join the region close to $\boldsymbol{x}_A$ with a target close to $\boldsymbol{x}_B$ in the shortest time, supposing that the vessels obey the following equations of motion:

$$\begin{cases} \dot{\boldsymbol{X}}_t = \boldsymbol{u}(\boldsymbol{X}_t, t) + \boldsymbol{U}^{ctrl}(\boldsymbol{X}_t) \\ \boldsymbol{U}^{ctrl}(\boldsymbol{X}_t) = V_{\mathrm{s}}\boldsymbol{n}(\boldsymbol{X}_t) \end{cases} \tag{1}$$

where $\boldsymbol{u}(\boldsymbol{X}_t, t)$ is the velocity of the underlying 2D advecting flow, and $\boldsymbol{U}^{ctrl}(\boldsymbol{X}_t) = V_{\mathrm{s}}\boldsymbol{n}(\boldsymbol{X}_t)$ is the control slip velocity of the vessel with fixed intensity V_{s} and varying steering direction: $\boldsymbol{n}(\boldsymbol{X}_t) = (\cos[\theta_t], \sin[\theta_t])$, where the angle is evaluated along the trajectory, $\theta_t = \theta(\boldsymbol{X}_t)$. We introduce a dimensionless slip velocity by normalizing with the maximum velocity $u_{\max}$ of the underlying flow: $\tilde{V}_{\mathrm{s}} = V_{\mathrm{s}}/u_{\max}$. Zermelo's problem reduces to optimize the steering direction θ in order to reach the target [24]. For time independent flows, optimal navigation (ON) control theory gives a general solution [14,22]. Assuming that the angle θ is controlled continuously in time, the optimal steering angle must satisfy the following time-evolution:

$$\dot{\theta}_t = A_{21}\sin^2\theta_t - A_{12}\cos^2\theta_t + (A_{11} - A_{22})\cos\theta_t \sin\theta_t, \tag{2}$$

where $A_{ij} = \partial_j u_i(\boldsymbol{X}_t)$ is evaluated along the agent trajectory $\boldsymbol{X}_t$ obtained from Eq. (1). The set of equations (1–2) may lead to chaotic dynamics even for

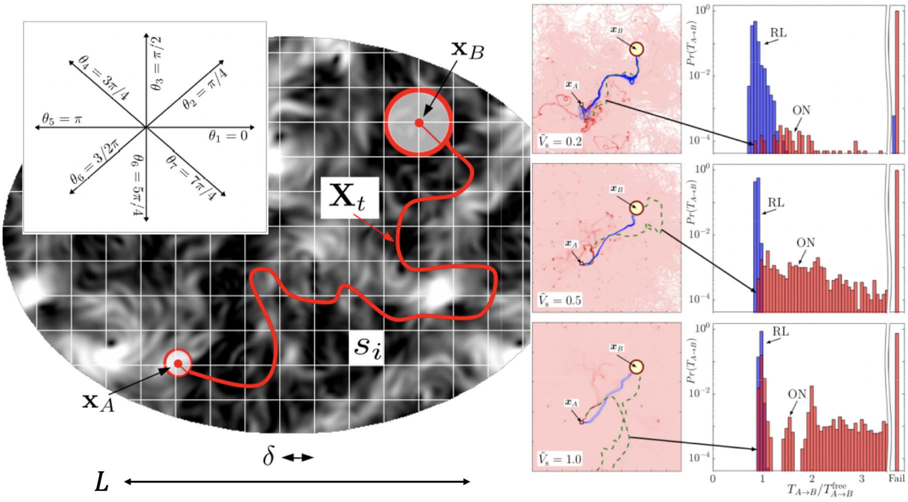

Fig. 1. Left: Image of one turbulent snapshot used as the advecting flow, with the starting, $\boldsymbol{x}_A$, and ending point, $\boldsymbol{x}_B$, of our problem. We also show an illustrative navigation trajectory $\boldsymbol{X}_t$. The flow is obtained from a spatially periodic snapshot of a 2D turbulent configuration in the inverse energy cascade regime with a multi-scale power-law Fourier spectrum, $E(k) = \sum_{k<\boldsymbol{k}<k+1} |\boldsymbol{u}(\boldsymbol{k})|^2 \sim k^{-5/3}$. For RL optimization, the initial conditions are taken randomly inside a circle of radius d_A centered around $\boldsymbol{x}_A$. Similarly, the final target is the circle of radius d_B centered around $\boldsymbol{x}_B$. The flow area is covered by a grid-world with tiles s_i with $i = 1, \dots, N_s$ and $N_s = 900$ of size $\delta \times \delta$ which identify the state-space for the RL protocol. The large-scale periodicity of the underlying flow is L, and we fixed $\delta = L/10$. Every time interval Δt, the unmanned vessel selects one of the 8 possible actions a_j with $j = 1 \dots 8$ (the steering directions θ_j depicted in left top inset) according to a policy $\pi(a|s)$, where π is the probability distribution of the action a given the current state s of the agent at that time. The policy is optimized during the learning to maximize the total reward, r_{tot}, proportional to minus the navigation time, $r_{tot} \sim -T_{\boldsymbol{x}_A \to \boldsymbol{x}_B}$, so that the maximal reward corresponds to the fastest trajectory. For the policy to converge, the actor-critic method requires to accumulate experience over a number of the order of 1000 different trajectories, with small variations depending on the values of the slip velocity $\tilde{V}_s$ and the specific flow properties. In a second series of experiments we added an additional action, the possibility to switch off the power, i.e. to let $V_s = 0$. This allows the vessel to fully take advantage of the flow and save energy (see Sect. 3.2). Right: spatial concentrations of trajectories for three values of $\tilde{V}_s$. The flow region is color coded proportionally to the time the trajectories spend in each pixel area for both ON (red) and RL (blue). Light colors refer to low occupation and bright to high occupation. The green-dashed line shows the best ON out the 20000 trajectories. Right histograms: arrival time distribution for ON (red) and RL (blue). Probability of not reaching the target within the upper time limit is plotted in the *Fail* bar. (Color figure online)

time-independent flows in two spatial dimensions. Due to the sensitivity to small perturbations in chaotic systems the ON approach becomes useless for many practical applications.

2 Methods

RL applications [21] are based on the idea that an optimal solution can be obtained by learning from continuous interactions of an agent with its environment. The agent interacts with the environment by sampling its states s, performing actions a and collecting rewards r. In our case the vessel acts as the agent and the two-dimensional flow as the environment. In the approach used here, actions are chosen randomly with a probability that is given by the policy $\pi(a|s)$, given the current flow-state s. The goal is to find the optimal policy $\pi^*(a|s)$ that maximizes the total reward, $r_{\text{tot}} = \sum_t r_t$, accumulated along one episode. For the purpose to find the fastest trajectory we used r_t composed of three different terms;

$$r_t = -\Delta t + \frac{|\boldsymbol{x}_B - \boldsymbol{X}_{t-\Delta t}|}{V_s} - \frac{|\boldsymbol{x}_B - \boldsymbol{X}_t|}{V_s}. \tag{3}$$

The first term accumulates a large penalty if it takes long for the agent to reach the end point, while the second and third terms describe the change in free-flight time to the target, i.e. the difference in time it would take, if the flow is neglected, to reach the target from the locations at this and the previous state change [2]. It follows the the total reward is proportional to minus the actual time taken by the trajectory to reach the target,

$$r_{tot} \sim -T_{\boldsymbol{x}_A \to \boldsymbol{x}_B},$$

neglecting a constant term that does not depend on the training, see [4] and Fig. 1 for precise definition of flow-states and agent-actions. An episode is finalized when the trajectory reaches the circle of radius d_B around the target. In order to converge to robust policies each episode is started with a uniformly random position within a given radius, d_A, from the starting point. To estimate the expected total future reward we follow the one-step actor-critic method [21] based on a gradient ascent in the policy parametrization. In the second part of our work, we modify the navigation setup by allowing the unmanned vessel to turn off its ‘engine’, to allow it to navigate just following the flow without its own propulsion speed. In this framework, navigation can be optimal with respect to minimal energy consumption rather than time, or to a tradeoff between energy consumption and time. To repeat the training of the optimal policy taking into account of both aspects, energy and time, we modified our RL scheme as follows. First, we added the new action to turn off the vessel propulsion speed, i.e. letting $V_s = 0$, in addition to the eight possible navigation angles considered before. Second, we modified the reward function in order to weigh the relative importance of navigation time and energy consumption. This was obtained by adding a new term describing the time the vessel consumes energy, $-\lambda \Delta t_{pow}$, to the instantaneous reward in Eq. (3) as follows

$$r_t = -(\Delta t + \lambda \Delta t_{pow}) + \frac{|\boldsymbol{x}_B - \boldsymbol{X}_{t-\Delta t}|}{V_s} - \frac{|\boldsymbol{x}_B - \boldsymbol{X}_t|}{V_s}. \tag{4}$$

The total reward becomes proportional to minus the sum of the two time contributions,

$$r_{tot} \sim -\left(T_{\boldsymbol{x}_A \to \boldsymbol{x}_B} + \lambda T_{pow}\right). \tag{5}$$

The time Δt_{pow} counts the time the vessel navigates with self propulsion, giving a total time T_{pow} where energy is spent. The factor λ weighs the importance of energy consumption time and total navigation time in the optimisation. We have repeated the training of the RL optimal policy with the new time-energy combined goals in the time-independent flow shown in Fig. 1, as well as in a more realistic time-dependent 2D turbulent flow. The latter was obtained by solving the incompressible Navier-Stokes equations on a periodic square domain with side length $L = 2\pi$ and $N = 512^2$ number of collocation points, see [4] for more details about the flow.

3 Results (Time-Independent Flows)

3.1 Shortest Time, No Energy Constraints

In the right part of Fig. 1 we show the main results comparing RL and ON approaches [4]. The minimum time taken by the best trajectory to reach the target is of the same order for the two methods. The most important difference between RL and ON lies in their robustness as seen by plotting the spatial density of trajectories in the right part of Fig. 1 for the optimal policies of ON and RL with three values of $\tilde{V}_s$. We observe that the RL trajectories (blue coloured area) form a much more coherent cloud in space, while the ON trajectories (red coloured area) fill space almost uniformly. Moreover, for small navigation velocities, many trajectories in the ON system approach regular attractors, as visible by the high-concentration regions. The rightmost histograms in Fig. 1 show a comparison between the probability of arrival times for the trajectories illustrated in the two-dimensional domain, providing a quantitative estimation of the better robustness of RL compared to ON. Other RL algorithms, such as Q-learning[21], could also be implemented and compared with other path search algorithms such as A^* which is often used in computer science [12,19].

3.2 Minimal Energy Consumption

In this section we present results on the simultaneous optimisation of minimal travel time and energy consumption. To begin with, we consider the same time-independent flow as in the previous section. In Fig. 2 we show three sets of trajectories following three policies obtained by optimising the reward (5) for $\lambda = 0$, 2 and 6. The trajectories are superposed on the flow velocity amplitude $|\boldsymbol{u}(\boldsymbol{X}_t, t)|$ (left panel) and the Okubo-Weiss parameter Δ_{OW} [15,23] (right panel), defined as;

$$\Delta_{OW} = (A_{11} - A_{22})^2 + (A_{21} + A_{12})^2 - (A_{21} - A_{12})^2. \tag{6}$$

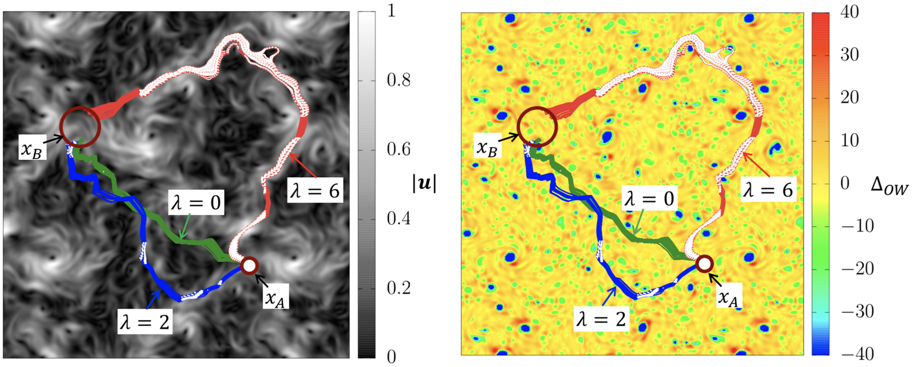

Fig. 2. Three sets of fifty trajectories going from point $\boldsymbol{x}_A$ to point $\boldsymbol{x}_B$, following the optimal policies for three values of λ with propulsion speed either turned on, $\tilde{V_s} = 0.8$ (color), or turned off, $V_s = 0$ (white). (Left panel) The trajectories are plotted on top of the amplitude of the time-independent flow velocity, $|\boldsymbol{u}|$. (Right panel) Same trajectories plotted over the Okubo-Weiss parameter, $\varDelta_{OW}$, see Eq. (6). (Color figure online)

Here A_{ij} is the fluid-gradient matrix as defined after Eq. (2). The decomposition in Eq. (6) is particularly useful to distinguish strain dominated ($\varDelta_{OW} > 0$, orange-red colors) from vortex dominated ($\varDelta_{OW} < 0$, green-blue colors) regions of the flow. Colored regions of the trajectories show where the action is to have the propulsion on and white regions show where the propulsion is off. When $\lambda = 0$, the energy consumption does not matter for the reward, and the only difference compared to the case in the previous section is that the policy can now choose one additional action: the zero self propulsion speed. However, as seen from Fig. 2, this action is rarely chosen when $\lambda = 0$, and the vessel navigates with a constant self-propelling velocity. On the other hand, when the energy-dependent reward is activated, as in the case of $\lambda = 2$, we observe a difference in the optimal path followed by the vessel. This is because it has to balance the penalties from the total navigation time and the time with self-propulsion. When λ becomes larger, this difference in the optimal path becomes more significant. For $\lambda = 6$ we observe trajectories that are much longer and dominated by passive navigation, just following the flow. To have a more accurate comparison of the arrival time to the target, $T_{\boldsymbol{x}_A \to \boldsymbol{x}_B}$, and the total active navigation time, T_{pow}, for the different values of λ, we show in Fig. 3 the evolution of these two terms as functions of the episode number during the training of the three different policies. The total reward (5) is a linear combination of these two terms, where T_{pow} is multiplied by the factor λ. We first observe that the training converges after around 10k episodes. Second, we see that for $\lambda = 0$, both $T_{\boldsymbol{x}_A \to \boldsymbol{x}_B}$ and T_{pow} lies close to each other for all episodes, suggesting that the optimal policy never found a state where it is better to navigate with zero propulsion to reach the target faster. For values of λ larger than zero, the found policies end up with T_{pow} below the value of the $\lambda = 0$ case, with the consequence of saving energy

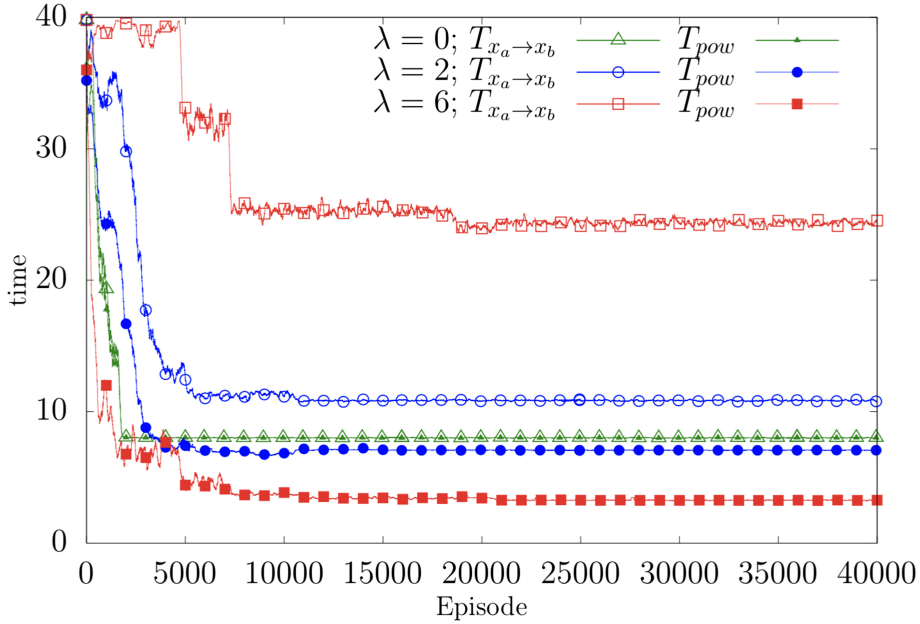

Fig. 3. Total navigation time, $T_{\boldsymbol{x}_A \to \boldsymbol{x}_B}$ (open symbols) and total power-on time, T_{pow} (full symbols), measured for different trajectories obtained during the training as a function of the episode number, and for three different values of λ.

even though the time to reach the target is longer. A final result for this case of time-independent flow is shown in Fig. 4, where we present the Probability Density Functions (PDFs) of the total navigation time, $T_{\boldsymbol{x}_A \to \boldsymbol{x}_B}$ (main panel) and of the power-on navigation time T_{pow} (inset). The distributions are sampled over 40k trajectories with initial conditions close to $\boldsymbol{x}_A$ that follows the optimal policies obtained for five values of λ. These PDFs show that for $\lambda = 0$, both times are of the order of $1.2\,T^{free}_{A \to B}$, where $T^{free}_{A \to B} = |\boldsymbol{x}_B - \boldsymbol{x}_A|/V_s \sim 6.4$ is the free-flight time to go from point A to point B with a fixed self propulsion speed V_s and without flow. For larger λ, the total navigation time increases while the power-on time decreases monotonically up to $\lambda = 6$. Increasing λ up to 10 we do not observe further reduction of T_{pow}, the PDF only becomes more peaked around the value $\approx 0.4\,T^{free}_{A \to B}$ as found for $\lambda = 6$. This result suggests that we have found the minimal amount of propulsion required for the vessel to be able to navigate to the target.

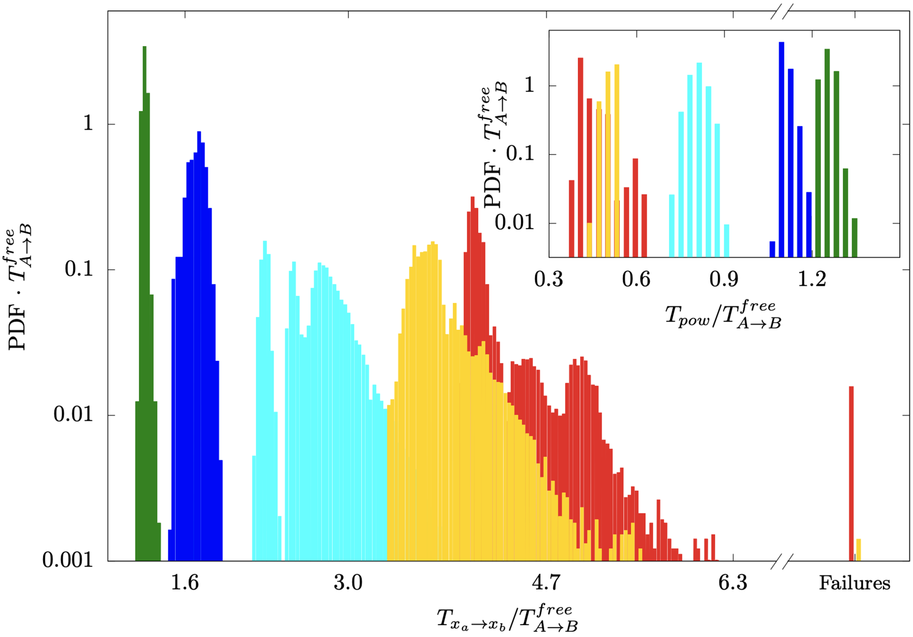

Fig. 4. (Main panel) PDFs of the arrival time, $T_{\boldsymbol{x}_A\to\boldsymbol{x}_B}$ normalized by the free-flight time $T^{free}_{A\to B}$, and measured over 40k different trajectories evolving on a time-independent flow. The different colors indicate different values of λ, from $\lambda = 0$ (no extra cost for using power-on, green color) up to $\lambda = 10$ (yellow color). The failures bars indicate the probability that a trajectory following a given policy does not reach the final target. (Inset) PDFs of the power-on time, T_{pow}, normalized by $T^{free}_{A\to B}$, measured along the same 40k trajectories shown in the main panel. (Color figure online)

4 Results (Time-Dependent Flow)

In this last section we consider the same optimal navigation problem as in the previous section, but with a more realistic time-dependent flow. For this case we adopted a small self-propulsion velocity, $\tilde{V}_s = 0.2$, i.e. only 20% of the maximal flow velocity amplitude. In Fig. 5 we present, as in the previous section, the PDFs of both $T_{\boldsymbol{x}_A\to\boldsymbol{x}_B}$ (solid lines full symbols) and T_{pow} (dashed lines empty symbols) obtained over 60k different trajectories following the converged optimal policies for $\lambda = 0$, and $\lambda = 2$. These results show that, as for the time-independent case, when $\lambda > 0$ RL finds a solution that spends less energy at the cost of a longer total navigation time compared to the solution when $\lambda = 0$. Let us stress that with a probability of the order of 1 in 1000 we observed trajectories that were not able to reach the final target, as indicated by the failure bars reported in Fig. 5. Finally, Fig. 6 shows six different snapshots at different times during the evolution of two different sets of trajectories that follows the optimal policies obtained for $\lambda = 0$ and $\lambda = 2$. The trajectories are superposed on the time-dependent flow velocity. Similar to Fig. 2, white regions on the trajectories show

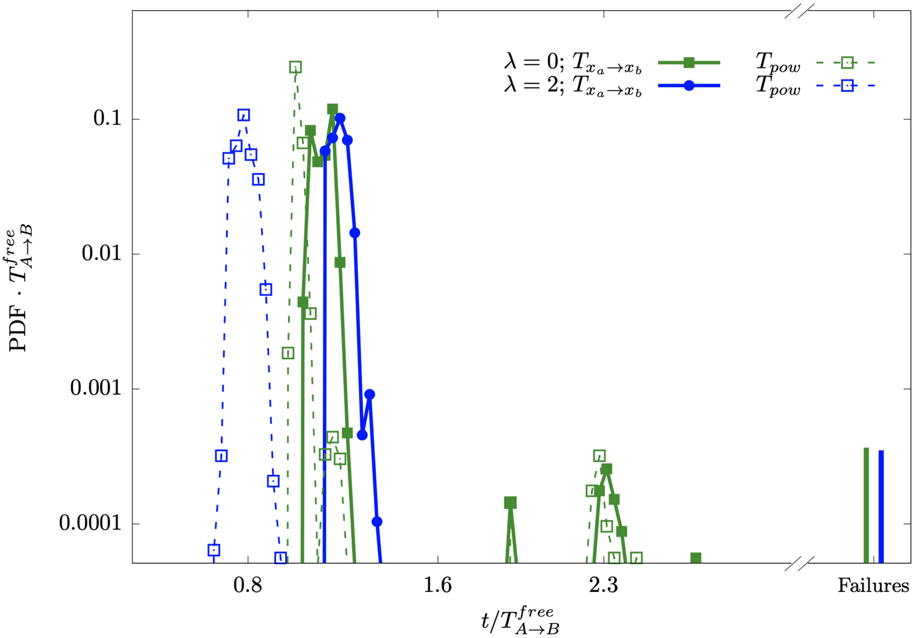

Fig. 5. PDFs of the arrival time, $T_{x_A \to x_B}$ (solid lines, full symbols) and of the power-on time, T_{pow} (dashed lines, open symbols) measured over 60k different trajectories following the optimal policy in a time-dependent flow, both normalized by the free-flight time $T^{free}_{A \to B}$. Colours distinguish the two values of λ used during the training. The navigation speed used along these trajectories is $\tilde{V}_s = 0.2$, hence, 20% of the maximum flow velocity amplitude. The failures bars indicate the probability of a trajectory to not reach the target after a long navigation time. (Color figure online)

where the vessel is navigating with zero self propulsion speed. We remark that even when $\lambda = 0$, the found optimal policy chooses the $V_s = 0$ action in the region close to the target. As a result, the PDFs of the total navigation time and the power-on time are not identical even for the case of $\lambda = 0$. This is a very nice example of the fact that the resulting policy in RL benefits from the added control when the set of allowed actions is enlarged and that, in our particular application, passively moving with the flow can be better than navigating when the flow blows you in the right direction, independently of the requirement to minimize energy.

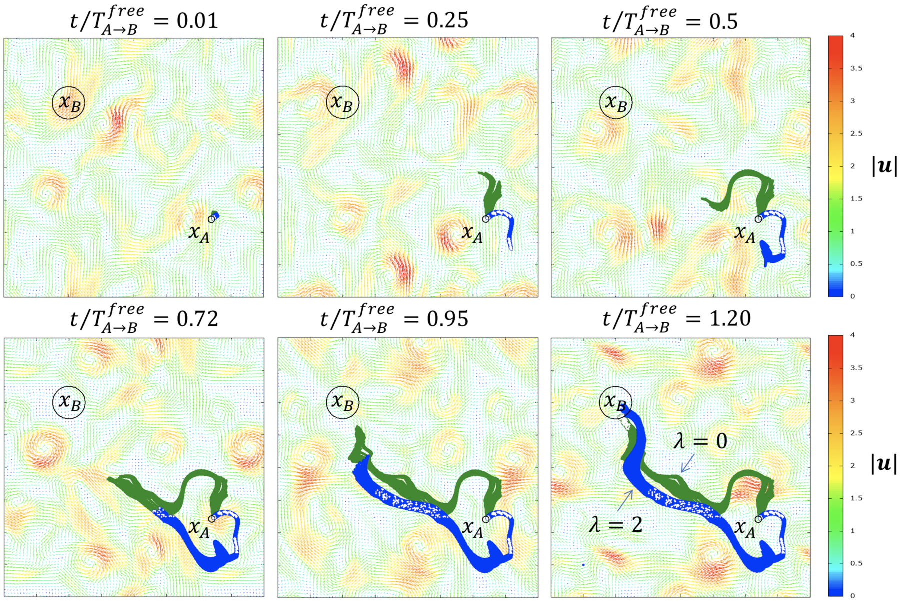

Fig. 6. Six snapshots at different times taken during the evolution of two sets of trajectories in a time-dependent flow, following the optimal policies for $\lambda = 0$ (green color) and $\lambda = 2$ (blue color). The six times are normalized to the free flight time. The flow streamlines are coloured proportionally to their amplitude, while the white points along the navigation trajectories indicate locations where the selected action was passive navigation, i.e. $V_S = 0$. (Color figure online)

5 Conclusions

We have first discussed a systematic investigation of Zermelo's time-optimal navigation problem in a realistic 2D turbulent flow, comparing both RL and ON approaches [4]. We showed that RL stochastic algorithms are key to bypass unavoidable instability given by the chaoticity of the environment and/or by the strong sensitivity of ON on the initial conditions in the presence of non-linear flow configurations. RL methods offer also a wider flexibility, being applicable to energy-minimization problems and in situations where the flow evolution is known only in a statistical sense as in partially observable Markov processes. Let us stress that, instead of starting from a completely random policy as we did here, it is also possible to implement RL to improve a-priori policies designed for a particular problem. For example, one can use an RL approach to optimize an initial *trivial policy*, where the navigation angle is selected as the action that points most directly toward the target. In the second part of this work, we further analyzed the more complex problem where the optimization of the total navigation time is balanced by the energy consumption required to reach the target. Also in this case, we found that RL is able to converge to non-trivial solutions where the vessel navigates most of the time as a passive object transported by the flow, with only a minimum number of corrections to its trajectory required to reach the final target.

Acknowledgments. This project has received partial funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 882340). K.G. acknowledges funding from the Knut and Alice Wallenberg Foundation, Grant No. KAW 2014.0048, and Vetenskapsrådet, Grant No. 2018-03974. F.B acknowledges funding from the European Research Council under the European Union's Horizon 2020 Framework Programme (No. FP/2014–2020) ERC Grant Agreement No. 739964 (COPMAT).

References

1. Alexakis, A., Biferale, L.: Cascades and transitions in turbulent flows. Phys. Rep. **767–769**, 1–101 (2018)
2. Andrew, Y.N., Harada, D., Russelt, S.: Policy invariance under reward transformations: theory and application to reward shaping. ICML **99**, 278 (1999)
3. Bechinger, C., Di Leonardo, R., Löwen, H., Reichhardt, C., Volpe, G., Volpe, G.: Active particles in complex and crowded environments. Rev. Mod. Phys. **88**(4), 045006 (2016)
4. Biferale, L., Bonaccorso, F., Buzzicotti, M., Clark Di Leoni, P., Gustavsson, K.: Zermelo's problem: optimal point-to-point navigation in 2D turbulent flows using reinforcement learning. Chaos: Interdisc. J. Nonlinear Sci. **29**(10), 103138 (2019)
5. Bryson, A.E., Ho, Y.: Applied Optimal Control: Optimization, Estimation and Control. Routledge, New York (1975)
6. Centurioni, L.R.: Drifter technology and impacts for sea surface temperature, sea-level pressure, and ocean circulation studies. In: Venkatesan, R., Tandon, A., D'Asaro, E., Atmanand, M.A. (eds.) Observing the Oceans in Real Time. SO, pp. 37–57. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-66493-4_3
7. Colabrese, S., Gustavsson, K., Celani, A., Biferale, L.: Flow navigation by smart microswimmers via reinforcement learning. Phys. Rev. Lett. **118**(15), 158004 (2017)
8. Colabrese, S., Gustavsson, K., Celani, A., Biferale, L.: Smart inertial particles. Phys. Rev. Fluids **3**(8), 084301 (2018)
9. Gustavsson, K., Biferale, L., Celani, A., Colabrese, S.: Finding efficient swimming strategies in a three-dimensional chaotic flow by reinforcement learning. Eur. Phys. J. E **40**(12), 1–6 (2017). https://doi.org/10.1140/epje/i2017-11602-9
10. Kraus, N.D.: Wave glider dynamic modeling, parameter identification and simulation. Ph.D. thesis, University of Hawaii at Manoa, Honolulu, May 2012 (2012)
11. Lermusiaux, P.F., et al.: A future for intelligent autonomous ocean observing systems. J. Mar. Res. **75**(6), 765–813 (2017)
12. Lerner, J., Wagner, D., Zweig, K.: Algorithmics of Large and Complex Networks: Design, Analysis, and Simulation, vol. 5515. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02094-0
13. Lumpkin, R., Pazos, M.: Measuring surface currents with surface velocity program drifters: the instrument, its data, and some recent results. In: Lagrangian Analysis and Prediction of Coastal and Ocean Dynamics, pp. 39–67 (2007)
14. Mannarini, G., Pinardi, N., Coppini, G., Oddo, P., Iafrati, A.: VISIR-I: small vessels-least-time nautical routes using wave forecasts. Geosci. Model Dev. **9**(4), 1597–1625 (2016)

15. Okubo, A.: Horizontal dispersion of floatable particles in the vicinity of velocity singularities such as convergences. In: Deep Sea Research and Oceanographic Abstracts, vol. 17, pp. 445–454. Elsevier (1970)
16. Petres, C., Pailhas, Y., Patron, P., Petillot, Y., Evans, J., Lane, D.: Path planning for autonomous underwater vehicles. IEEE Trans. Robot. **23**(2), 331–341 (2007)
17. Pontryagin, L.S.: Mathematical Theory of Optimal Processes. Routledge, London (2018)
18. Roemmich, D., et al.: The Argo program: observing the global ocean with profiling floats. Oceanography **22**(2), 34–43 (2009)
19. Russell, S., Norvig, P.: Artificial intelligence: a modern approach (2002)
20. Schneider, E., Stark, H.: Optimal steering of a smart active particle. arXiv preprint arXiv:1909.03243 (2019)
21. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (2018)
22. Techy, L.: Optimal navigation in planar time-varying flow: Zermelo's problem revisited. Intell. Serv. Robot. **4**(4), 271–283 (2011). https://doi.org/10.1007/s11370-011-0092-9
23. Weiss, J.: The dynamics of enstrophy transfer in two-dimensional hydrodynamics. Phys. D: Nonlinear Phenomena **48**(2–3), 273–294 (1991)
24. Zermelo, E.: Über das navigationsproblem bei ruhender oder veränderlicher windverteilung. ZAMM-J. Appl. Math. Mech./Zeitschrift für Angewandte Mathematik und Mechanik **11**(2), 114–124 (1931)