Reinforcement Learning control for Airborne Wind Energy production

Lorenzo Basile
International Centre for Theoretical Physics
University of Trieste
Trieste, Italy
lbasile@ictp.it

Maria Grazia Berni
University of Trieste
Trieste, Italy
mariagrazia.berni@studenti.units.it

Antonio Celani
International Centre for Theoretical Physics
Trieste, Italy
acelani@ictp.it

Abstract

Airborne Wind Energy (AWE) is an emerging technology that promises to be able to harvest energy from strong high-altitude winds, while addressing some of the key critical issues of current wind turbines. AWE is based on flying devices (usually gliders or kites) that, tethered to a ground station, fly driven by the wind and convert the mechanical energy of wind into electrical energy by means of a generator. Such systems are usually controlled by adjusting the trajectory of the kite using Optimal Control techniques, such as Model-Predictive Control. These methods are based upon a mathematical model of the system to control, and they produce results that are strongly dependent on the specific model at use and difficult to generalize. Our aim is to replace these classical techniques with an approach based on Reinforcement Learning (RL), which can be used even in absence of a known model. Experimental results prove that RL is a viable method to control AWE systems in complex simulated environments, including turbulent flows.

1 Introduction

In the field of renewable power sources, wind energy is particularly appealing since it can potentially power the entire world and it is largely available almost everywhere on the planet [1]. Currently, wind power generation happens through wind turbines, huge three-bladed devices often found in very large on-shore or off-shore wind farms.

Airborne Wind Energy (AWE) is an alternative lightweight technology for wind energy harvesting, based on tethered flying devices, usually power kites. These devices fly under the effect of aerodynamic forces (drag and lift) and convert wind energy to electrical energy by means of a generator, which can be placed either directly on the device or on the ground [2]. Such technology promises to address most of the issues of traditional wind turbines, since kites can be controlled to fly at higher altitudes, where strong and constant winds can be found and since their smaller, lighter structure leads to much lower material costs and environmental impact [3, 4]. However, AWE technology is operationally more complex than traditional turbines and it may face reliability issues in case of critical wind anomalies [5].

In order to safely and profitably operate an AWE system, the trajectory of the kite can be optimized by controlling its attack and bank angle through the lines that connect it to the ground station. Several

Tackling Climate Change with Machine Learning: workshop at NeurIPS 2023.

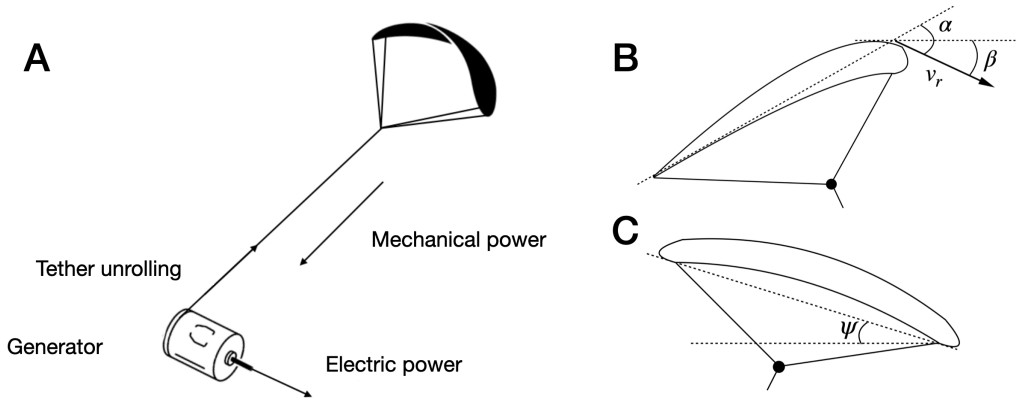

Figure 1: (A) Schematic representation of an AWE system in the yo-yo configuration [6]. (B) Side view of the kite, showcasing the state variables α (attack angle) and β (relative wind-speed angle). Controlling the attack angle allows the kite to soar and glide. (C) Rear view of the kite, depicting the bank angle ψ, which is used to turn the kite left and right.

works [6, 7, 8] have successfully explored the possibility of applying Model-Predictive Control (MPC) to AWE systems. However, there are two shortcomings to this approach. First, traditional Optimal Control techniques are strongly model-dependent, meaning that the control strategies they provide are usually hard to generalize from one configuration to another and that the quality of the results that can be achieved depends on the quality of the model. The issue can be partially tamed by applying Robust Control techniques [9]. Second, MPC for AWE usually focuses on minimizing the excursions from a predefined path [8], which in general is not the one that optimizes power production. This lack of alignment between the objective trajectory of MPC and the final optimization objective has been highlighted by recent research [10] in the context of drone navigation, showcasing the fact that the trajectories obtained with MPC may be suboptimal in real conditions. Such problem is particularly acute in changing environments like the turbulent atmosphere, where the kite should dynamically adapt its trajectory to wind fluctuations in order to efficiently extract energy.

In this study, we take a different standpoint and give a proof of concept that classical control techniques can be replaced with Reinforcement Learning (RL), which can be used even in absence of a known model, to directly aim at the maximization of energy production. RL has already proven successful in addressing flight optimization problems [10, 11, 12] and it has already been employed in the context of AWE in a towing setting, in which kites are used to move a ship or a vehicle on the ground [13]. In this work, we present a way to control AWE systems with RL for electrical energy production.

The code required to reproduce our results is publicly available at `https://github.com/lorenzobasile/KiteRL`.

2 Methods

In a RL setting, a decision maker (the agent) interacts with its surroundings (the environment) and based on this interaction it receives a numerical signal which can be seen as a reward (or a penalty) for the actions it took. By trial and error, on the long run the agent understands which actions are to be preferred in each state in order to maximize the reward it obtains from the environment and optimizes its decision-making strategy accordingly. We adopt two RL algorithms: SARSA [14] and TD3 (Twin Delayed Deep Deterministic policy gradient) [15].

SARSA is a simple temporal-difference control algorithm, in which for each state-action pair the agent keeps an estimate of the future reward it expects to receive. SARSA is a tabular algorithm, meaning that the information the agent keeps about expected rewards is stored in a table indexed by states and actions. This requires the definition of a discrete state space and of a discrete set of actions, which can be a substantial limitation when dealing with complex systems, in which the number of state variables may be too large to handle in a tabular way because of memory constraints or the discretization of continuous state variables may result in a poor approximation.

One feasible approach to overcome the limitations of discrete states and actions, when dealing with continuous problems, is by using policy gradient methods, such as TD3. It is an actor-critic algorithm, directly derived from DDPG [16], which employs neural networks to learn both a parameterized policy and value. The actor is the policy structure that, given the current state of the system, decides which action should be taken, while the critic, computing a scalar quantity called temporal-difference error, informs the actor about the value of the actions it takes. This scalar quantity drives the learning in both actor and critic.

Detailed information about the training setup we employ for our RL algorithms is provided in the Appendix.

3 Setup

3.1 System configuration

Many different configurations can be used for AWE but, in this work, we focus on the "yo-yo configuration" [6] (Fig. 1A), in which a power kite is linked by means of a tether to an electric machine. The movement of the kite unrolls the cable and puts into rotation the shaft of the machine, which acts as a generator (active or traction phase); then, when the line is at its maximum extension, the machine acts as a motor and rewinds the cable (passive or reel-in phase), preparing for a new traction phase. In this work, our aim is the optimization of the sole traction phase. We train our RL algorithms on simulated kite trajectories, computed using a mathematical model based on the one presented in [6].

3.2 Reward structure and state variables

The structure of the reward signal that the environment delivers to the agent is designed to maximize energy production and to keep the kite airborne: at each time step the agent receives a reward equal to the energy produced since last time step or a penalty if the kite falls to the ground.

We test our approach in three increasingly complex simulated wind patterns: constant and uniform wind, constant wind whose speed increases linearly with altitude, and turbulent Couette flow [17]. Due to limitations in the turbulent flow data at our disposal, we constrain the kite to fly at altitudes lower than 100 m. To ensure that this requirement is always met, we assume a tether length of 100 m, and consider the episode terminated when the tether is fully unrolled (or if the kite hits the ground). Throughout all training simulations, our algorithms are tested using a kite of mass 1 kg and characteristic area of 10 m^2.

We provide our algorithms with a small amount of information in the form of three easily measurable state variables, namely the attack angle, the relative wind-speed angle (Fig. 1B) and the bank angle (Fig. 1C). By definition, these variables are continuous, hence a discretization step is needed for SARSA, which can only handle discrete state and action spaces. Two of the state variables (i.e. the attack and bank angles) also serve as the controls of our system: at each decision step the agent can increase, decrease or keep still these angles, resulting in the possibility to control the trajectory of the kite by making it glide, soar and turn.

4 Results

Even though they can access very limited state information, both SARSA and TD3 are able to learn suitable policies to keep the kite airborne in all the three aforementioned wind patterns. From this point, we will only discuss results obtained in the turbulent Couette flow.

In all the simulations that we produce, despite different configurations and different algorithms, the agent learns to drive the kite in an approximately helical motion (Fig. 2). Given the specific wind pattern in which the kite is flying, this means that it is almost always moving crosswind. This result is in agreement with theoretical findings that proved crosswind flight to be optimal for energy production [18]. The results of our algorithms in terms of energy production are reported in Table 1. Our optimization objective is the maximization of energy production, and according to this metric TD3 yields a small improvement (approximately 5%) with respect to SARSA.

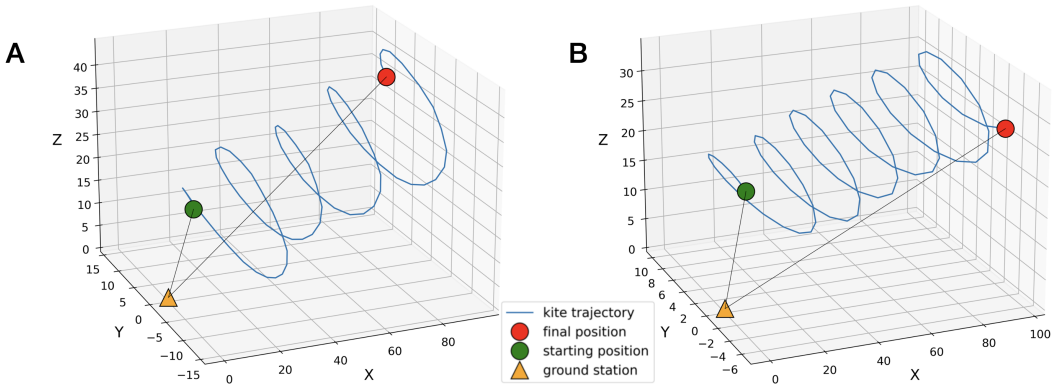

Figure 2: Sample trajectories under the policies learned by (A) SARSA and (B) TD3 in the turbulent Couette flow

Table 1: Average performance in turbulent flow

Algorithm	Episode duration	Energy	Average power
SARSA	8.7 s	0.056 kWh	23.4 kW
TD3	8.2 s	0.059 kWh	25.5 kW

Moreover, TD3 is quicker at fully unrolling the tether, leading to slightly shorter episodes. Combined with the previous consideration, this means that the average power produced using TD3 is higher than using SARSA. This point is particularly meaningful in the perspective of full cycle optimization (also including the reel-in phase) and on-field deployment. Both SARSA and TD3 tend to move the kite upwards, to reach profitable zones where the wind is stronger. Reaching high altitudes is also preferable for the optimization of the passive reel-in phase, as usually in this phase the kite is initially led towards the zenith following a "low-power" trajectory [19]. The effective wind speed measured on the kite and the instantaneous power produced by the system are reported in Fig. 4 and 5 in the Appendix for three sample episodes.

The policies learned by SARSA and TD3 are shown in Fig. 3, always considering three distinct traction episodes. Some large fluctuations are due to the random initialization values of both the attack and bank angle, but the overall pattern is regular and it shows that both algorithms tend to choose relatively low attack angles and make very limited use of the bank angle, which oscillates between two nearby values with SARSA and is discarded altogether by TD3, which learns to keep it constant.

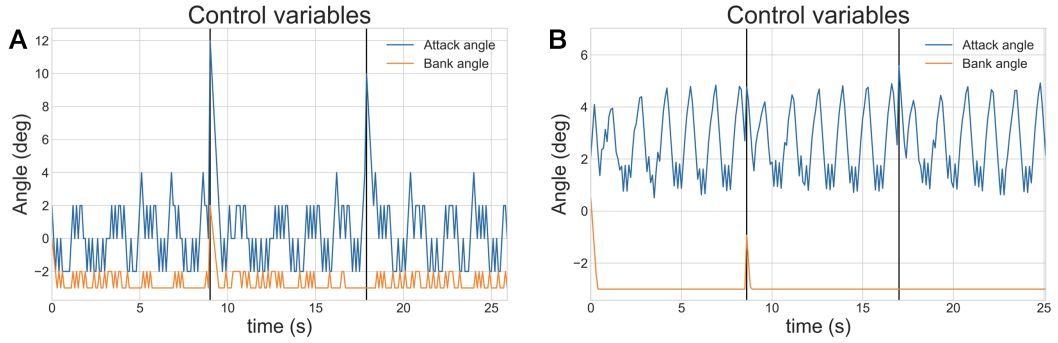

Figure 3: Control variables over 3 episodes using the policy learned by (A) SARSA (B) TD3

5 Conclusions and perspectives

The results outlined in the previous section show how powerful RL can be in addressing the control of AWE systems in a model-free way, allowing simulations in very complex environments like turbulent flow, which would be intractable with a traditional Optimal Control approach.

We are currently working on optimizing the whole working cycle of the system, including the passive reel-in phase, in which the focus is on the minimization of the amount of energy spent to take the kite back to its initial position in the least possible amount of time. Moreover, in the perspective of full automation and optimization of the system, it would be interesting to make the decision to switch between the two phases learnable as well, as it is far from obvious that this switch should only happen when the tether is fully unrolled.

Finally, it is worth pointing out that further work on feature selection could benefit our results, as it is likely that the state variables we employ only provide our agent with a partial view of its surroundings.

Acknowledgements

We would like to thank Claudio Leone, Andrea Mazzolini, Nicole Orzan and Johnson Oyero for many fruitful discussions and suggestions.

References

[1] C. L. Archer and M. Z. Jacobson. "Evaluation of global wind power". *Journal of Geophysical Research*, 110, 2005.

[2] A. Cherubini, A. Papini, R. Vertechy, and M. Fontana. "Airborne Wind Energy Systems: A review of the technologies". *Renewable and Sustainable Energy Reviews*, 51, 2015.

[3] P. Bechtle, M. Schelbergen, R. Schmehl, U. Zillmann, and S. Watson. "Airborne wind energy resource analysis". *Renewable Energy*, 141, 2019.

[4] C. L. Archer and K. Caldeira. "Global assessment of high-altitude wind power". *Energies*, 2(2), 2009.

[5] V. Salma, F. Friedl, and R. Schmehl. "Improving reliability and safety of airborne wind energy systems". *Wind Energy*, 2020.

[6] M. Canale, L. Fagiano, and M. Milanese. "High Altitude Wind Energy Generation Using Controlled Power Kites". *IEEE Transactions on Control System Technologies*, 2009.

[7] B. Houska and M. Diehl. "Optimal Control for Power Generating Kites". *IEEE European Control Conference (ECC)*, 2007.

[8] S. Gros, M. Zanon, and M. Diehl. "Control of Airborne Wind Energy systems based on Nonlinear Model Predictive Control & Moving Horizon Estimation". *IEEE European Control Conference (ECC)*, 2013.

[9] B. Cadalen, P. Lanusse, J. Sabatier, F. Griffon, and Y. Parlier. "Robust control of a tethered kite for ship propulsion". *IEEE European Control Conference (ECC)*, 2018.

[10] Y. Song, A. Romero, M. Müller, V. Koltun, and D. Scaramuzza. "Reaching the limit in autonomous racing: Optimal control versus reinforcement learning". *Science Robotics*, 8(82), 2023.

[11] G. Reddy, J. Wong-Ng, A. Celani, T. Sejnowski, and M. Vergassola. "Glider soaring via reinforcement learning in the field". *Nature*, 562, 2018.

[12] M. Bellemare, S. Candido, P. S. Castro, J. Gong, M. C. Machado, S. Moitra, S. S. Ponda, and Z. Wang. "Autonomous navigation of stratospheric balloons using reinforcement learning". *Nature*, 588, 2020.

[13] N. Orzan, C. Leone, A. Mazzolini, J. Oyero, and A. Celani. "Optimizing Airborne Wind Energy with Reinforcement Learning". *The European Physical Journal E*, 46, 2023.

[14] G. Rummery and M. Niranjan. "On-Line Q-Learning Using Connectionist Systems". *Technical Report CUED/F-INFENG/TR 166*, 1994.

[15] S. Fujimoto, H. van Hoof, and D. Meger. "Addressing Function Approximation Error in Actor-Critic Methods". *International Conference on Machine Learning*, 2018.

[16] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. "Continuous control with deep reinforcement learning". *International Conference on Learning Representations*, 2016.

[17] V. Avsarkisov, S. Hoyas, M. Oberlack, and J. García-Galache. "Turbulent plane Couette flow at moderately high Reynolds number". *Journal of Fluid Mechanics*, 751, 2014.

[18] M. Loyd. "Crosswind Kite Power". *Journal of Energy*, 1980.

[19] L. Fagiano. "Control of Tethered Airfoils for High-Altitude Wind Energy Generation". *PhD thesis, Politecnico di Torino*, 2009.

[20] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". *International Conference on Learning Representations*, 2015.

A Appendix

A.1 Wind speed and power profiles

Here we provide some additional plots relative to the same sample episodes of Fig. 3, in the turbulent flow. Fig. 4 represents the norm of the wind velocity measured on the kite: it is clear that both SARSA and TD3 succeed in taking the kite to high-altitude regions, where the wind is stronger. This behaviour is desirable in terms of energy production, as it results in high-power trajectories (Fig. 5), and a quick unrolling of the tether.

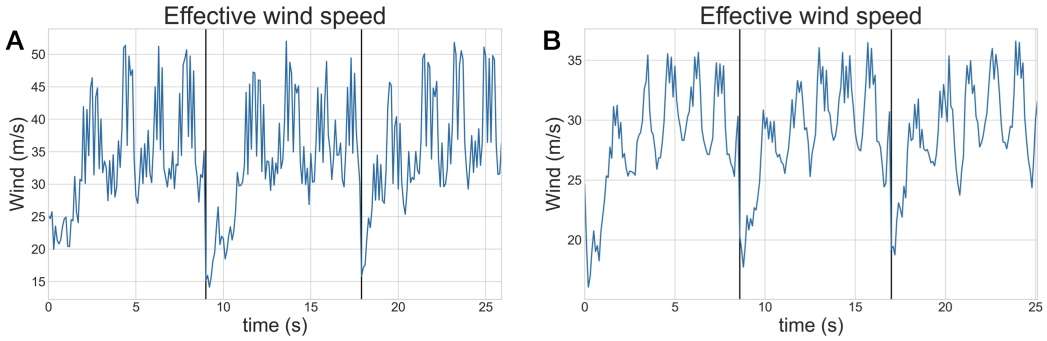

Figure 4: Effective wind speed over 3 episodes using the policy learned by (A) SARSA (B) TD3

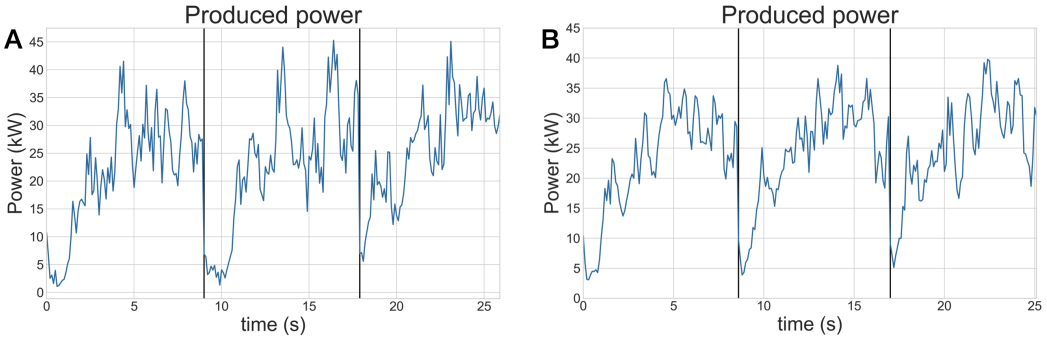

Figure 5: Power produced over 3 episodes using the policy learned by (A) SARSA (B) TD3

A.2 Training details and hyperparameters

A.2.1 SARSA

SARSA can access three discrete state variables: the attack angle α, ranging between -8° and 20° (as in [6]), the bank angle ψ, ranging between -3° and 3°, and the relative wind speed angle β, which can take 10 discrete values between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$. At each decision step (every 0.1 s) the agent can apply one of three actions (increase by one, keep still, decrease by one) on the two control variables α and β, resulting in a total of 9 possible actions.

While learning, actions are chosen using an ϵ-greedy policy to favor exploration. The parameter ϵ is initially set to 0.01 and later decayed using a power law scheduling through training. The same decay schedule is used for the learning rate, which is initially set at 0.1.

A.2.2 TD3

TD3 uses the same state variables and the same ranges as SARSA, but in a continuous way. However, at each decision step the actions of the agent are constrained so that the control variables α and β cannot be varied by more than 1°. The TD3 agent adopted in this work employs the same critic and actor network architecture as in [15], a learning rate for both actor and critic of 0.001 and Adam optimizer [20]. To favor exploration, Gaussian noise with mean 0 and standard deviation 0.2 is added to the policy at training time.