Multi-agent reinforcement learning for dynamic wind farm control

Claire Bizon Monroc

Under the supervision of Ana Bušić, Jiamin Zhu and Donatien Dubuc

Plan

1. Introduction
2. Delay-aware MARL algorithms for the Wind Farm Control Problem
3. WFCRL: a new MARL benchmark for wind farm control
4. Conclusion

01
Introduction

Wake effects decrease a wind farm's power output

Figure: Horns rev offshore wind farm, Vattenfall, 2008

Real world problem

- Upstream turbines create sub-optimal wind conditions for downstream turbines
- This decreases the total amount of power produced.

Solution: changing the yaw can mitigate wake effects

- **Yaw**: angle between the rotor plane and the direction of the incoming wind
- Increasing the yaw of an upstream turbine deflects its wake away from downstream turbines.

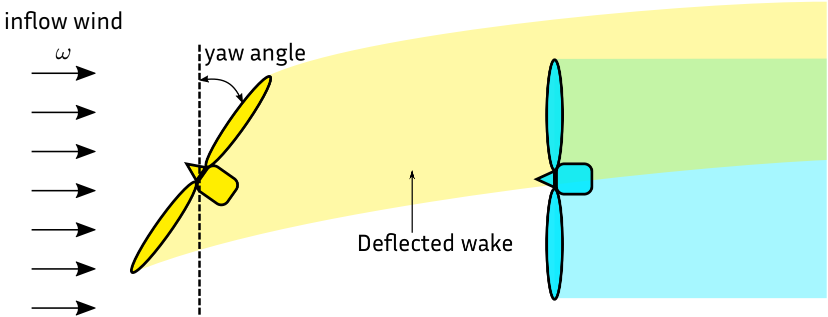


Figure: Wake deflection with yaw control

Wake steering with yaw control

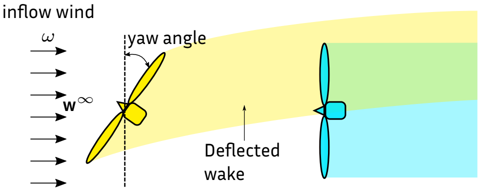
WFCP : Wind Farm Control Problem

Goal

Maximize the total power ouput of a wind farm with M turbines

Controls

$\gamma = (\gamma^1, \ldots, \gamma^M)$: yaws



Measurements

- w^{∞}: free-stream wind conditions (direction ϕ^{∞} and speed u^{∞})
- $\mathcal{P}_{1,t} \ldots, \mathcal{P}_{M,t}$ individual productions at any time t.
- $\mathcal{P}_{farm,t} = \sum_{i=1}^{M} \mathcal{P}_{i,t}$ total power output

Yaw optimization

Static Problem

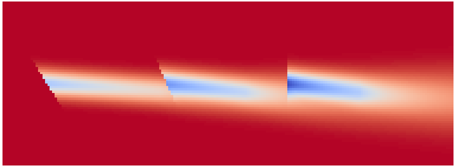
- Wind conditions are constant in time

$$\max_{\gamma} \mathcal{P}_{farm}$$

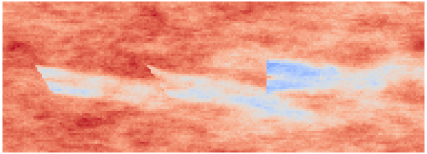
Dynamic Problem

- Wind conditions change at every time-step ($0 < \beta < 1$)

$$\max_{\gamma_0,\dots,\gamma_\infty} \sum_{k=0}^{\infty} \beta^k \mathcal{P}_{farm,k}$$



(a) Static simulation: FLORIS



(b) Dynamic simulation: FAST.Farm

Challenges

Modeling errors

- Steady-state models: FLORIS
- 2D Navier-Stokes: WFSim
- Dynamic Wake Meandering model: FAST.Farm
- Large eddy simulations: SOWFA

computation cost

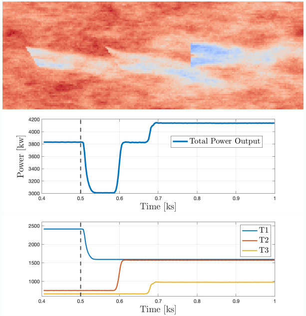
fidelity

Challenges

Modeling errors

- Steady-state models
- 2D Navier-Stokes
- Dynamic Wake Meandering model
- Large eddy simulations

Wake propagation

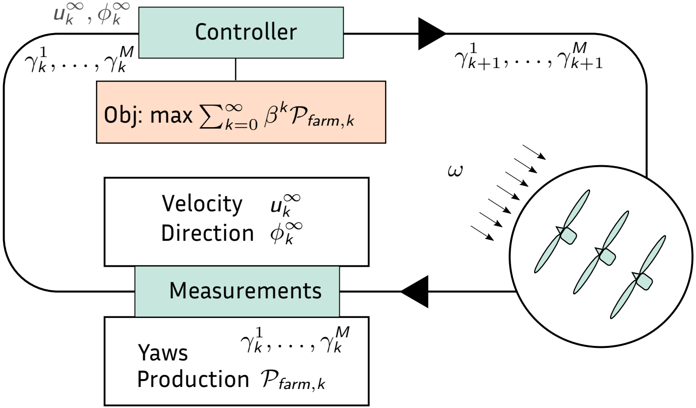


Scaling to large farms

A model-free approach

Adapt policies in the real system !

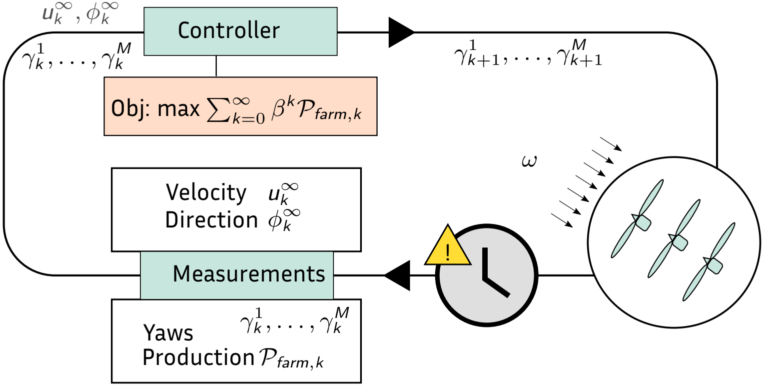
Design data-driven methods that learn from observing control inputs and output measurements collected on the wind farm



... that is robust to wake propagation dynamics

Account for wake propagation times

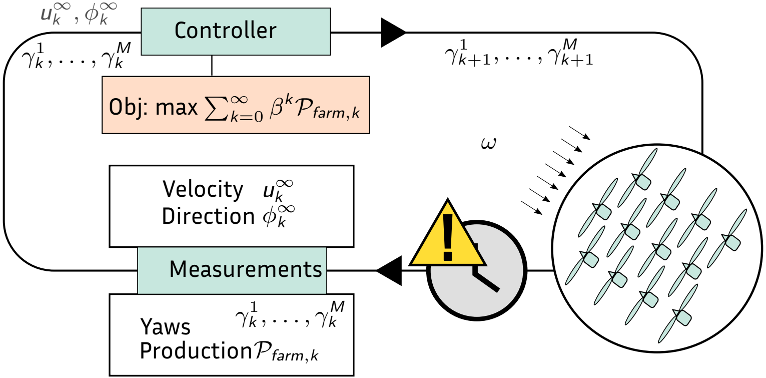
Wake propagation dynamics create a time delay between a change of yaws and a measurable impact on the energy production of the wind farm.



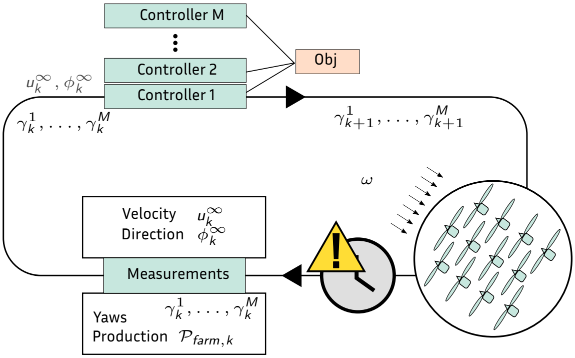
... that is robust to wake propagation dynamics

Account for wake propagation times

Wake propagation dynamics create a time delay between a change of yaws and a measurable impact on the energy production of the wind farm.



A cooperative multi-agent reinforcement learning (MARL) approach



Decentralized learning of local policies

MARL approach

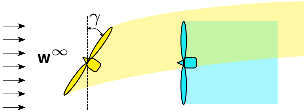
- M agents
- 1 shared reward
- Each agent i receives a partial observation of the system and learns a policy π^i

Used in static and quasi-static simulations (Graf et al. 2019; Xu et al. 2020; Stanfel et al. 2021)

02

Delay-aware MARL algorithms for the Wind Farm Control Problem

WFCP as a delayed Dec-MDP problem



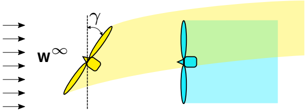
The multi-agent WFCP

Objective:

$$\max_{\pi^1,\dots,\pi^M} \mathbb{E}\left[\sum_{k=0}^{\infty} \beta^k \sum_{i=1}^{M} P_{i,k}\right]$$

- S : full state space (unobserved)
- $\Omega_{i,1\leq i\leq M}$: M observation spaces with γ^i and $\mathbf{w}^\infty$
- $A_{i,1\leq i\leq M}$: M action spaces representing the change in local yaw $\Delta\gamma^i$ between two time-steps
- $r : S \times A \times S \rightarrow [0, R]$: shared reward function

WFCP as a delayed Dec-MDP problem



The multi-agent WFCP : a delayed approach

Objective:

$$\max_{\pi^1,\ldots,\pi^M} \mathbb{E}\left[\sum_{k=0}^{\infty} \beta^k \sum_{i=1}^{M} P_{i,k}\right]$$

- $S_{i,1\leq i\leq M}$: M local state spaces with γ^i and $\mathbf{w}^{\infty}$
- $S = \Pi_i^M S_i$: global space as factorization of local spaces
- $A_{i,1\leq i\leq M}$: M action spaces representing the change in local yaw $\Delta\gamma^i$ between two time-steps
- $r : S \times A \times S \to [0, R]$: shared reward function
- A delay $d \in \mathbb{N}$: the number of time-steps after which the reward can be collected

WFCP as a delayed Dec-MDP problem

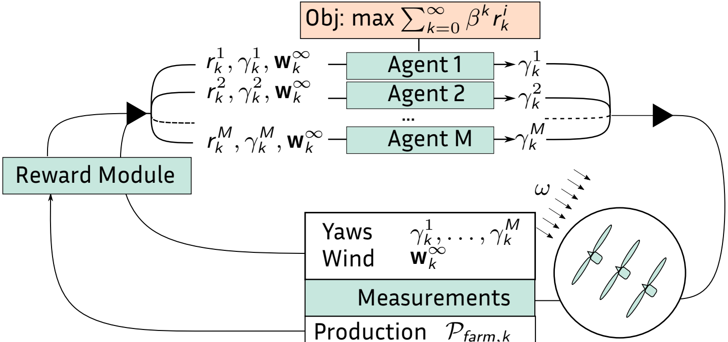
The multi-agent WFCP: a delayed approach with local rewards

Objectives:

$$\max_{\pi^1} \mathbb{E}\left[\sum_{k=0}^{\infty} \beta^k \sum_{i=1}^{M} r_k^1\right] \quad \dots \quad \max_{\pi^M} \mathbb{E}\left[\sum_{k=0}^{\infty} \beta^k \sum_{i=1}^{M} r_k^M\right]$$

- $S_{i,1\leq i\leq M}$: S_i: M local state spaces with γ^i and $\mathbf{w}^\infty$
- $S = \Pi_i^M S_i$: global space as factorization of local spaces
- $A_{i,1\leq i\leq M}$: M action spaces representing the change in local yaw $\Delta\gamma^i$ between two time-steps
- $r^i : S_i \times A_i \times S_i \rightarrow [0, R]$: M local reward functions
- M delays : the number of time-steps after which each local reward can be collected

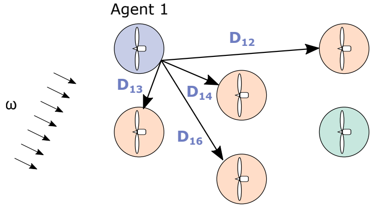
Delay-aware MARL algorithms for the WFCP



$$\mathbf{w}_k^\infty = (u_k^\infty, \phi_k^\infty)$$

Figure: WFCP-MARL: our multi-agent formulation of the dynamic WFCP as a Delayed MDP

Local reward functions



Delay Matrix

	D_{12}	D_{13}	D_{14}		D_{16}	Agent 1
			D_{34}		D_{36}	
	D_{42}			D_{45}	D_{46}	
				D_{65}		Agent 6

Delays approximated with Taylor's frozen wake hypothesis (Taylor 1938):

$$D_{ij} \propto \frac{\text{distance}(i \rightarrow j)}{u_\infty}$$

Reward functions

$$r_{i,k} = \begin{cases} 1 & \text{if} \quad \dfrac{V_2^i - V_1^i}{V_1^i} > \Delta \\ -1 & \text{if} \quad \dfrac{V_2^i - V_1^i}{V_1^i} \leq -\Delta \\ 0 & \text{otherwise} \end{cases}$$

with $V_1^i = \sum_{j=1}^{M} P_{j,k} \quad V_2^i = \sum_{j=1}^{M} P_{j,k+D_{i,j}}$

$$\Delta > 0$$

Empirical results

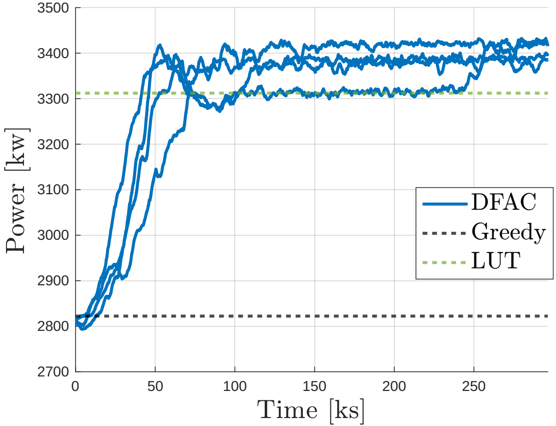


Figure: Average energy produced during 1h

DFAC: Delay-Aware Fourier Actor Critic Agents

- Row of 3 turbines
- Turbulent stationary wind inflow
- Turbulence intensity: 8%
- Average wind speed: 8 m/s
- Simulated on FAST.Farm

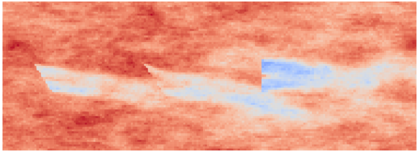


Figure: Row of 3 turbines simulated in FAST.Farm

Empirical results

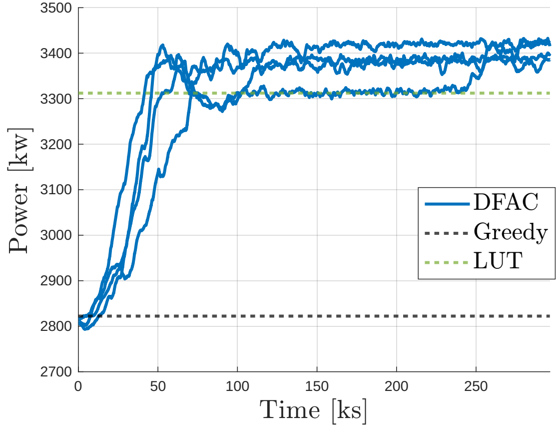


Figure: Average energy produced during 1h

Scaling to larger wind farms:

Layout	DFAC Energy [MWh]	Over baseline (%)	Rise Time (ks)
Layout 1 (3T)	9.34	45.83	22
Layout 3 (7T)	23.02	9.35	38
Layout 4 (16T)	31.44	10.99	302
Layout 5 (32T)	47.90	37.02	-

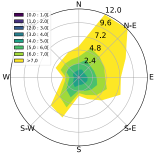
Table: DFAC on WFSim: scaling experiments on layouts with 3, 7, 16 and 32 turbines. Results after 200ks (Layout 1, 3) and 600ks (Layout 4, 5).

Exploiting model knowledge with imitation

Imitate optimal policies from model-based optimization

- Use optimal policies learned with static simulator FLORIS to guide online learning with dynamic simulator FAST.Farm
- Evaluation on WFSim and FAST.Farm under both changing and turbulent wind conditions

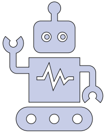
Collect wind data



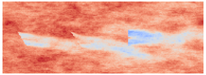
→ Optimize in static model

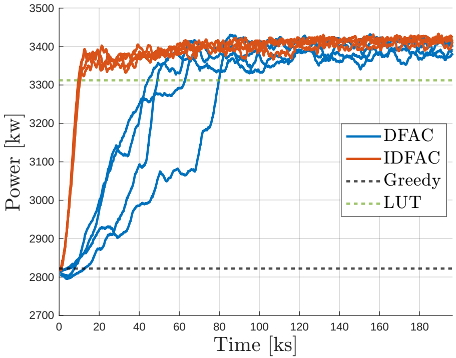
→ Extract policy initialization

→ Adjust in dynamic model

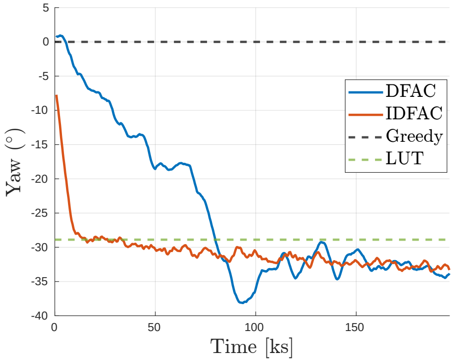


Imitation





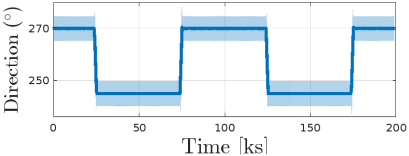
(a) Power output

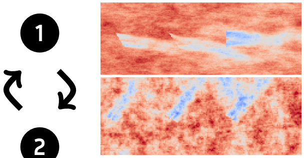


(b) Yaw of the first turbine on 1 run

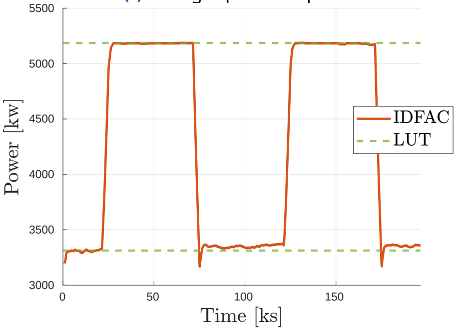
Figure: DFAC and IDFAC algorithms - 1h average of total power output (a) and yaws during the first experiment of each algorithm (b) on 56h of simulation.

Imitation: turbulent wind step

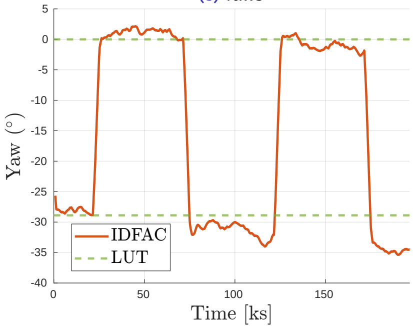




(a) Averaged power output



(b) Yaws



03

WFCRL: a new MARL benchmark for wind farm control

Wind farm control as a Cooperative MARL problem

Decentralized Partially Observable Markov Decision Process (Dec-POMDP)

- M turbines = M agents
- P: Transition kernel
- $0 < \beta < 1$: Discount factor
- r: Shared reward
- T Length of episode (can be ∞)
- Full state space S:
  = **unobserved** wind field
- $o_1, \ldots, o_M$: Agents' observations
  = local wind and actuator measurements
- $a_1, \ldots, a_M$: Agents' actions
  = local changes in actuation

$+$ global observation o_g = all local observations + free-stream wind measurements (for Centralized Training with Decentralized execution)

Objective: $\max_\pi \mathbb{E}_{s_0, a_0, s_1, \ldots} [J] \quad J := \sum_{k=0}^{T} \beta^k r_k$

Wind farm control as a Cooperative MARL problem

Decentralized Partially Observable Markov Decision Process (Dec-POMDP)

- M turbines = M agents
- P: Transition kernel
- $0 < \beta < 1$: Discount factor
- r: Shared reward
- T Length of episode (can be ∞)
- Full state space S:
  = **unobserved** wind field
- $o_1, \ldots, o_M$: Agents' observations
  = local wind and actuator measurements
- $a_1, \ldots, a_M$: Agents' actions
  = local changes in actuation

$+$ global observation o_g = all local observations + free-stream wind measurements (for Centralized Training with Decentralized execution)

Reward examples

Total production, Distance to production target, Loads on turbine structure ...

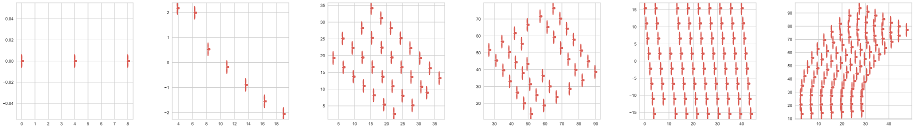
WFCRL environments

Environment creation

1. Wind farm layout
2. Wind condition scenario
3. Wind farm simulator

1. Layouts

21 pre-registered farm layouts, including 5 taken from real wind farms



(a) Turb3_Row1 (3) (b) Ablaincourt (7) (c) Ormonde (31) (d) WMR (36) (e) HornsRev1 (76) (f) HornsRev2 (92)

WFCRL environments

Environment creation

1. Wind farm layout
2. Wind condition scenario
3. Wind farm simulator

2. Wind Scenarios

3 wind condition scenarios:

- Constant wind
- New wind sampled at each episode
- Evolving time series

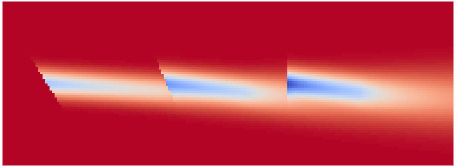
WFCRL environments

Environment creation

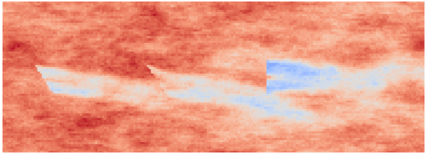
1. Wind farm layout
2. Wind condition scenario
3. Wind farm simulator

3. Simulators: multi-fidelity environments

2 state-of-the-art wind farm simulators with 2 degrees of fidelity
$\rightarrow$ allows evaluation of transfer strategies from static to dynamic models

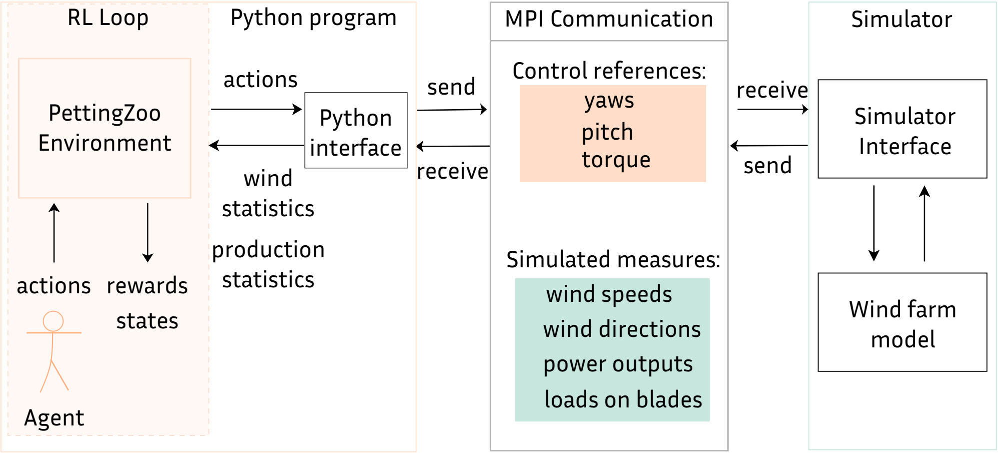


(a) Static simulation: FLORIS



(b) Dynamic simulation: FAST.Farm

WFCRL: Simulator Interfacing



Example: Production maximization benchmark

Goal: Maximize total power production while taking into account turbine loads
Reward design:

$$r_k = r_k^P - \alpha r_k^L$$

- Scaling term: α
- Normalized power output r^P

$$r_k^P = \frac{1}{M} \sum_i^M \frac{\hat{P}_k^i}{(u_{\infty,k})^3}$$

$\hat{P}_k^i$ measured power production; $u_{\infty,k}$ free-stream wind velocity;

- Simulator-dependent load indicator r^L

Evaluation score:

$$\texttt{score}(\pi_1, \ldots, \pi_M) = \sum_{j=1}^{n_w} \rho_j \sum_{k=0}^{T} r_k$$

n_w number of wind conditions; ρ_j weights

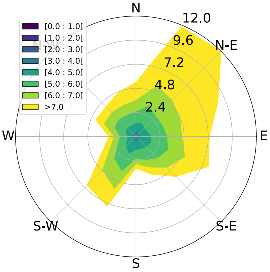
Example: Production maximization benchmark

Reward: $r_k = r_k^P - \alpha r_k^L$

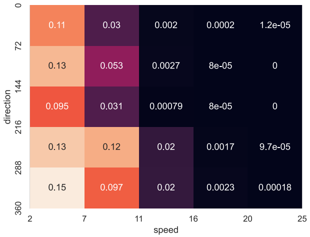
Evaluation score:

$$\texttt{score}(\pi_1, \ldots, \pi_M) = \sum_{j=1}^{n_w} \rho_j \sum_{k=0}^{T} r_k$$

Weights ρ_j estimated from SmartEole measurements campaign at Ablaincourt:



(a) SmartEole Windrose



(b) Empirical distribution $\{\rho_i\}$

Example: Production maximization benchmark

Environment: `Ablaincourt` layout + Floris simulator
Wind Scenario: Sampled at each episode (II)

Benchmark code available at
`www.github.com/ifpen/wfcrl-benchmark`

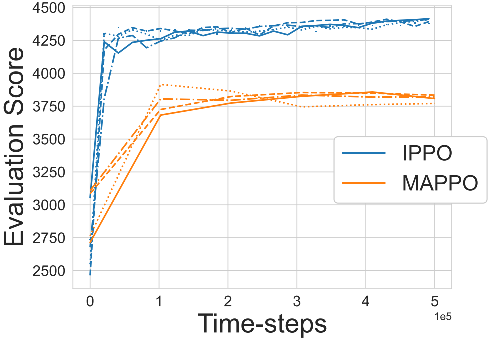


Figure: Evaluation score: training curve

Towards transfer learning strategies

Learning on realistic dynamic simulators is slow.
-> We need efficient strategies that can adapt policies learned with low-fidelity models.

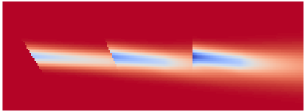


Figure: Static (FLORIS)

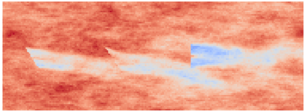


Figure: Dynamic (FAST.Farm)

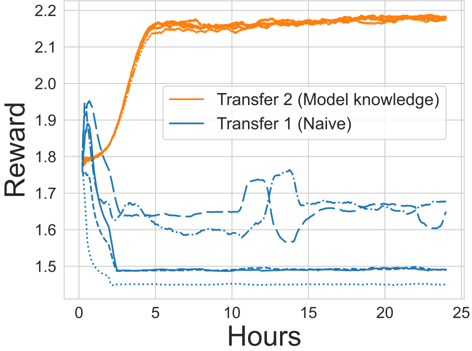


Figure: 2 transfer learning strategies from FLORIS to FAST.Farm

Conclusion

- Wind farm control as a delayed MARL probem
- Experimentally validated algorithms on state-of-the-art dynamic simulators
- Use of static models to guide learning in dynamic conditions
- A new benchmark and interfacing library to bridge RL and wind energy communities

GITHUB

Perspectives

Towards applications on real systems

- Validation on high-fidelity simulations
- Considering loads in a constrained Dec-POMDP
- Tracking a production signal for integration in the grid

Robust learning from static models

- Exploiting bags of models
- Exploiting offline data

WFCRL: future developments as an open-source library

GITHUB

Publications

1. **Bizon Monroc C.**, Bouba E., Bušić A., Dubuc D., and Zhu J. *Delay-aware decentralized q-learning for wind farm control.* In 2022 IEEE 61st Conference on Decision and Control (CDC).
2. **Bizon Monroc C.**, Bušić A., Dubuc D., and Zhu J. *Actor critic agents for wind farm control.* In 2023 American Control Conference (ACC).
3. **Bizon Monroc C.**, Bušić A., Dubuc D., and Zhu J. *Towards fine tuning wake steering policies in the field: an imitation-based approach.*. TORQUE 2024.
4. **Bizon Monroc C.**, Bušić A., Dubuc D., and Zhu J. *Multi-agent reinforcement learning for partially observable cooperative systems with acyclic dependence structure.*. Presented at ARLET Workshop, ICML 2024
5. **Bizon Monroc C.**, Bušić A., Dubuc D., and Zhu J. *WFCRL: A Multi-Agent Reinforcement Learning Benchmark for Wind Farm Control*, NeurIPS 2024 Datasets and Benchmarks Track

Bibliography I

Graf, Peter et al. (2019). "Distributed Reinforcement Learning with ADMM-RL". In: *2019 American Control Conference (ACC)*, pp. 4159–4166.

Stanfel, Paul et al. (Aug. 2021). "Proof-of-concept of a reinforcement learning framework for wind farm energy capture maximization in time-varying wind". In: *Journal of Renewable and Sustainable Energy* 13.4.

Taylor, G. I. (1938). "The Spectrum of Turbulence". In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 164.919, pp. 476–490.

Xu, Zhiwei et al. (2020). "Model-Free Optimization Scheme for Efficiency Improvement of Wind Farm Using Decentralized Reinforcement Learning". In: *IFAC-PapersOnLine* 53.2. 21st IFAC World Congress, pp. 12103–12108.

Thank you.