Potential of dynamic wind farm control by axial induction in the case of wind gusts

Florian Bürgel[1 [0000−0001−8673−5849]], Robert Scholz[2], Christian Kirches[1 [0000−0002−3441−8822]], and Sebastian Stiller[1 [0000−0002−8792−4390]]

[1] TU Braunschweig, Institute for Mathematical Optimization, Universitätsplatz 2, 38106 Braunschweig, Germany
{f.buergel,c.kirches,sebastian.stiller}@tu-braunschweig.de

[2] University of Heidelberg, Interdisciplinary Center for Scientific Computing, Im Neuenheimer Feld 205, 69120 Heidelberg, Germany
robert.scholz@iwr.uni-heidelberg.de

Abstract. Each wind turbine causes a wake with a reduced wind velocity and an increased turbulence. This wake influences the power and mechanical loads of downstream turbines. This strong connection to turbines in a wind farm is an opportunity for farm-wide optimal control, in particular when dynamic control for a wind gust is required. We use the already known axial-induction-based control and investigate its potential using mathematical optimization in the case of a wind gust in a wind field with turbulence and a simulated farm consisting of two turbines in wind direction. We show that optimal control of the upstream turbine significantly reduces the total tower load and increases the total power.

Keywords: Optimal control · wind farm · wind gust.

1 Introduction

Wind farms are important in future electrical energy grids. Usually, each local controller of a turbine maximizes its own power output (i.e., greedy control) while taking care of its own loads, see, e.g., [6]. However, in farms they interact with each other due to their wake, whose length and spatial distribution is also influenced by control changes. This suggests a farm-wide optimal control.

We use the axial-induction-based control concept (i.e., we control the generator torque and/or the collective blade pitch angle), see, e.g., [3, 1]. The idea is to decrease the power output of the upstream turbine to increase the combined power of both turbines, but the success depends on the turbine allocation, their characteristics and atmospheric conditions, see, e.g., [3]. In contrast, it always makes sense to mitigate loads. For power tracking and load mitigation on farm level we refer to [5]; they consider an average wind velocity of 10 m/s with a turbulence intensity (TI) of around 5 to 6%, see, e.g., [6] for TI definition.

We also consider a wind field with turbulence but with a wind gust—while turbulence is a frequent fluctuation, a wind gust is a peak of a few to some ten seconds, see, e.g., [6]. We show that offline optimal control of the upstream one of two turbines reduces total tower load and increases total power.

2 Simulation

Experimental setup In our experimental setup, we consider two NREL 5-MW turbines with a rating value of 5 MW and a rotor diameter of $D = 126\,\mathrm{m}$, see [8]. They are arranged in a straight line in the wind direction with a distance of $5D$. As we are interested in the potential of axial-induction-based control, their yaw angles are fixed to a yaw offset of $0°$. We denote the average wind velocity value of the (horizontal) ambient wind field by U_{ave} and consider $U_{\mathrm{ave}} = 6$ and $11\,\mathrm{m/s}$; indeed, these are representative examples as NREL 5-MW turbines have a cut-in, rated, and cut-out wind velocity (in m/s) of 3, 11.4, and 25, see [8]. For the wind gust model we follow [18], i.e., $U(t) := U_{\mathrm{ave}} - 0.37\,U_{\mathrm{gust}} \sin(3\pi t/T)\,(1 - \cos(2\pi t/T))$ if $t \in [0, T]$ and $U(t) = U_{\mathrm{ave}}$ otherwise, where the gust begins at time $t = 0$ and has a duration of $T = 10.5\,\mathrm{s}$. In fact, we use a time-shifted version of the formula with $U_{\mathrm{gust}} = 8\,\mathrm{m/s}$. Further, instead of U_{ave} we use a wind field including a TI of 6% as in [8, 5].

Simulation software The range of wake models include control-oriented models in 2D (e.g., Jensen, see [7, 11], FLORIS, see [4], and Gaussian, see, e.g., [2]) as well as high-fidelity simulations like SOWFA, see [12]. We use the mid-fidelity simulation FAST.Farm, which is validated against *Large-Eddy Simulations* (LES), see [9].

We generate the wind field with TurbSim (from OpenFAST, see [14]). For the preparation, the Python toolbox pyFAST, see [15], is helpful. We read the generated wind file by a modified version of readfile_bts.m (from MATLAB Toolbox for OpenFAST, see [13]), add the wind gust, and use a modified version of the write-BLgrid function, see [10], to save it. For the wind turbine (and wake) simulation we employ FAST.Farm (from OpenFAST). We tailored the r-test input files, see [17], to our setup. We use the default polar formulation for dynamic wake and the time step of 0.1 s for high-resolution computed area around the turbine.

Wind farm control We distinguish between three time intervals (in unit s). Usually, both turbines are controlled by the local controller from OpenFAST within the *simulation time interval* $[t_{\mathrm{s}_1}, t_{\mathrm{s}_2}]$, where $t_{\mathrm{s}_1} = 0$. For optimization, we overwrite the generator torques and blade pitches of the upstream turbine via the ROSCO controller interface (for pitch in rad), see [16], within the smaller *dynamic control time interval*, $[t_{\mathrm{c}_1}, t_{\mathrm{c}_2}]$, e.g., $[t_{\mathrm{p}} - 10, t_{\mathrm{p}} + 10]$ with gust peak at t_{p}. For data analysis we choose an *observation time interval*, $[t_{\mathrm{o}_1}, t_{\mathrm{o}_2}]$, where $t_{\mathrm{o}_1} \leq t_{\mathrm{c}_1}$ and t_{o_2} is chosen to include the control effect on the downstream turbine; usually, $t_{\mathrm{o}_2} = t_{\mathrm{s}_2}$. We define the control function $u = (u_\tau, u_\beta)$ over $[t_{\mathrm{c}_1}, t_{\mathrm{c}_2}]$, where u_τ and u_β are the generator torque and blade pitch of the upstream turbine.

Performance indicators We use the following simulation outputs for each turbine as a function: the *power* (in unit W) as $p_u : [t_{\mathrm{s}_1}, t_{\mathrm{s}_2}] \to \mathbb{R}^2_{\geq 0}$, the *velocity of the nacelle* (in m/s) in wind direction, $v_u : [t_{\mathrm{s}_1}, t_{\mathrm{s}_2}] \to \mathbb{R}^2$, the *blade pitch angle* (in °), $\beta_u : [t_{\mathrm{s}_1}, t_{\mathrm{s}_2}] \to \mathbb{R}^2$, and the *generator torque* (in kNm), $\tau_u : [t_{\mathrm{s}_1}, t_{\mathrm{s}_2}] \to \mathbb{R}^2_{\geq 0}$. We define four performance indicators for each turbine i as averages over $[t_{\mathrm{o}_1}, t_{\mathrm{o}_2}]$, namely the *power* output P_i (in MW), the *tower activity* $a_i^{(\mathrm{T})}$, the *pitch* *activity* $a_i^{(\mathrm{P})}$, and the *generator torque activity* $a_i^{(\mathrm{G})}$. The tower load is high when the nacelle oscillates; so, we use the absolute value of the nacelle velocity v_u to estimate the tower load by the so-called tower activity. Analogously, we proceed with the velocity of β_u, i.e., we consider $|\frac{\mathrm{d}}{\mathrm{d}t}\beta_u|$ to estimate the pitch actuators load by pitch activity, and with $|\frac{\mathrm{d}}{\mathrm{d}t}\tau_u|$ for the generator torque activity. Finally,

$$P_i(u) := \tfrac{1}{t_{\mathrm{o}_2}-t_{\mathrm{o}_1}} \int_{t_{\mathrm{o}_1}}^{t_{\mathrm{o}_2}} 10^{-6}\,(p_u(t))_i\,\mathrm{d}t,\ \ a_i^{(\mathrm{T})}(u) := \tfrac{1}{t_{\mathrm{o}_2}-t_{\mathrm{o}_1}} \int_{t_{\mathrm{o}_1}}^{t_{\mathrm{o}_2}} |(v_u(t))_i|\,\mathrm{d}t,$$

$$a_i^{(\mathrm{P})}(u) := \tfrac{1}{t_{\mathrm{o}_2}-t_{\mathrm{o}_1}} \int_{t_{\mathrm{o}_1}}^{t_{\mathrm{o}_2}} \left|\frac{\mathrm{d}}{\mathrm{d}t}(\beta_u(t))_i\right|\mathrm{d}t,\ \ a_i^{(\mathrm{G})}(u) := \tfrac{1}{t_{\mathrm{o}_2}-t_{\mathrm{o}_1}} \int_{t_{\mathrm{o}_1}}^{t_{\mathrm{o}_2}} \left|\frac{\mathrm{d}}{\mathrm{d}t}(\tau_u(t))_i\right|\mathrm{d}t.$$

Objective function The objective function combines the performance indicators in a weighted sum with parameters $\omega = (\omega^{(\mathrm{T})}, \omega^{(\mathrm{P})}, \omega^{(\mathrm{G})}) = (100, 10, 1) \in \mathbb{R}^3_{\geq 0}$:

$$f_\omega(u) := \sum_{i=1}^{2} \left(-P_i(u) + \omega^{(\mathrm{T})}\, a_i^{\mathrm{T}}(u) + \omega^{(\mathrm{P})}\, a_i^{(\mathrm{P})}(u) + \omega^{(\mathrm{G})}\, a_i^{(\mathrm{G})}(u) \right). \quad (1)$$

We add 5 to $f_\omega(u)$ if rates $\max\{|\frac{\mathrm{d}}{\mathrm{d}t}\tau_u|\}$ or $\max\{|\frac{\mathrm{d}}{\mathrm{d}t}\beta_u|\}$ (15 kNm/s and 8°/s, see [8]) are exceeded, the simulation aborts, or outputs negative powers.

3 Optimization

Regarding to the objective function (1), we want optimal control functions u_τ and u_β (of the upstream turbine) in $[t_{\mathrm{c}_1}, t_{\mathrm{c}_2}]$, i.e., we have to solve $\min\limits_{u_\beta, u_\tau} f_\omega(u)$.

To simplify the problem, we reduce the number of control functions: Instead of u_β and u_τ, we use the *virtual wind velocity* u_σ. To substitute u_τ and u_β by u_σ, we "read" the turbine controller in FAST.Farm simulation by generating a wind velocity ramp from 2 to 21 m/s and saving the used τ_u and β_u in a lookup table as matrix $L = (L_\sigma | L_\tau | L_\beta)$ with columns $L_\sigma = (\sigma_1, ..., \sigma_n)^{\mathrm{T}}$ and analog L_τ, L_β. To simplify further, we describe u_σ by a spline, where we use modified Akima piecewise cubic Hermite interpolation, and finally optimize via the spline values.

We analyze the gust to set spline breaks: The first spline has $\{t_{\mathrm{c}_1}, t_{\mathrm{p}}, t_{\mathrm{c}_2}\}$ as breaks (with wind gust peak at t_{p}) and wind velocities $\{U(t_{\mathrm{c}_1}), U(t_{\mathrm{p}}), U(t_{\mathrm{c}_2})\}$ as values. As break we add the time of the maximum absolute difference between the spline interpolation and the wind velocities. We repeat this until we have 7 breaks. They are summarized as vector t_{v} and as $t_{\mathrm{v}}^{\mathrm{c}}$ without t_{c_1} and t_{c_2}.

For the control function u_σ the spline has breaks t_{v} or $t_{\mathrm{v}}^{\mathrm{c}}$ and virtual wind velocities σ_{v} or $\sigma_{\mathrm{v}}^{\mathrm{c}}$ as values; the optimization problem is reduced to $\min\limits_{\sigma_{\mathrm{v}}^{\mathrm{c}}} f_\omega(u)$.

For the initial guess of $\sigma_{\mathrm{v}}^{\mathrm{c}}$, we follow the FAST.Farm controller (instead of wind velocities), i.e., we simulate the gust, read τ_u and β_u at t_{v}, and match each entry t of t_{v} to the corresponding virtual velocity $L_{\sigma,i}$ by computing $\arg\min_i\{|L_{\tau,i} - \tau(t)|/\max\{L_\tau\} + |L_{\beta,i} - \beta(t)|/\max\{L_\beta\}\}$. This makes up the vectors σ_{v} and $\sigma_{\mathrm{v}}^{\mathrm{c}}$ (without bound values). As the controller reacts with a time delay, i.e., usually

$k := \arg\max_i\{\sigma_{\mathrm{v},i}\} > \arg\max_j\{U(t_{\mathrm{v},j})\} =: \ell$, we shift the entries in σ_{v} by $k-\ell$ positions to left (repeating bound values) and denote this initial guess as $\sigma_{\mathrm{v}}^{\mathrm{c},0}$.

We restrict the virtual velocity control inputs at breaks to $\pm 2\,\mathrm{m/s}$ of the initial guess values, i.e., with respect to the lookup table bounds, we set lower and upper bound for an entry i of $\sigma_{\mathrm{v}}^{\mathrm{c}}$ to $\max\{2, \sigma_{\mathrm{v},i}^{\mathrm{c},0} - 2\}$ and $\min\{21, \sigma_{\mathrm{v},i}^{\mathrm{c},0} + 2\}$.

As FAST.Farm simulation does not contain sensitivity information, we use the *nonlinear programming* (NLP) solver fmincon in MATLAB, which computes gradients by finite differences. In particular, we use (the local optimization method of) sequential quadratic programming (SQP) with a step tolerance of 10^{-4}.

4 Results of computational experiments

A workstation with AMD EPYC 7742 64-Core Processor and 96 GByte RAM was limited by Slurm Workload Manager to 16 CPUs and GByte. We employ MATLAB in version R2021b and FAST.Farm (and its suppliers) as described in Sect. 2.

We demonstrate the effect of optimal control, see Sect. 3, compared to baseline (i.e., the local controller) using $U_{\mathrm{ave}} = 6$ and $11\,\mathrm{m/s}$ as discussed in Sect. 2.

For $U_{\mathrm{ave}} = 6\,\mathrm{m/s}$, we use (in s) $[t_{\mathrm{s}_1}, t_{\mathrm{s}_2}] = [0, 330]$, $[t_{\mathrm{c}_1}, t_{\mathrm{c}_2}] = [180, 200]$, and $[t_{\mathrm{o}_1}, t_{\mathrm{o}_2}] = [180, 330]$. The optimization (baseline) takes (in min) 158.9 (4.2) and results in $f_\omega(u) = 3.1414$ (3.2777). In detail, total power (in MW), weighted values of tower activity (in m/s), pitch activity (in °/s), and generator torque (in kNm/s) are 1.0204 (1.0196), 3.9079 (3.9617), 0.0000 (0.0000), and 0.2540 (0.3357); moreover, rates $\max\{|\frac{\mathrm{d}}{\mathrm{d}t}\beta_u|\}$ (in °/s) and $\max\{|\frac{\mathrm{d}}{\mathrm{d}t}\tau_u|\}$ (in kNm/s) are 0.00 (0.00) and 1.50 (5.20).

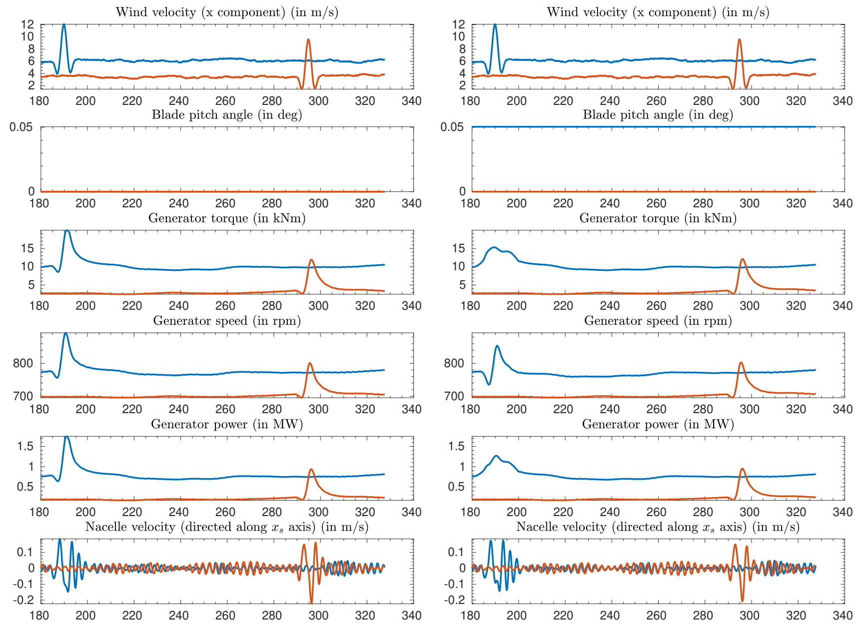
With the same units, we study $U_{\mathrm{ave}} = 11$ with $[t_{\mathrm{s}_1}, t_{\mathrm{s}_2}] = [0, 210]$, $[t_{\mathrm{c}_1}, t_{\mathrm{c}_2}] = [120, 140]$, and $[t_{\mathrm{o}_1}, t_{\mathrm{o}_2}] = [120, 210]$. The optimization (baseline) takes 58.8 (1.8) and results in $f_\omega(u) = 3.5766$ (11.2149—penalized as $\max\{|\frac{\mathrm{d}}{\mathrm{d}t}\tau_u|\} = 15.05$). In detail, total power, weighted values of tower activity, pitch activity, and generator torque are 6.2456 (6.1946), 5.9119 (8.5875), 2.7880 (2.3696), 1.1223 (1.4524); moreover, rates $\max\{|\frac{\mathrm{d}}{\mathrm{d}t}\beta_u|\}$ and $\max\{|\frac{\mathrm{d}}{\mathrm{d}t}\tau_u|\}$ are 7.24 (7.17) and 9.45 (15.05).

So, in both wind cases, see also Figs. 1 and 2, the power is increased and the tower activity is decreased (significantly for $11\,\mathrm{m/s}$), whereas $\max\{|\frac{\mathrm{d}}{\mathrm{d}t}\tau_u|\}$ is significantly decreased. For $11\,\mathrm{m/s}$, we observe a significantly smaller $|v_u|$ (0.3 instead of 0.7) and in the baseline a delayed blade pitch peak, cf. Sect. 3.

Conclusions We investigated the potential of axial-induction-based optimal control in a *simulated* wind farm with two turbines in a wind field with turbulence and a wind gust. In practice, a LIDAR could provide wind information in advance. The offline optimization of the upstream turbine increased the power and decreased the tower activity. So, it is worth thinking about real-time control.

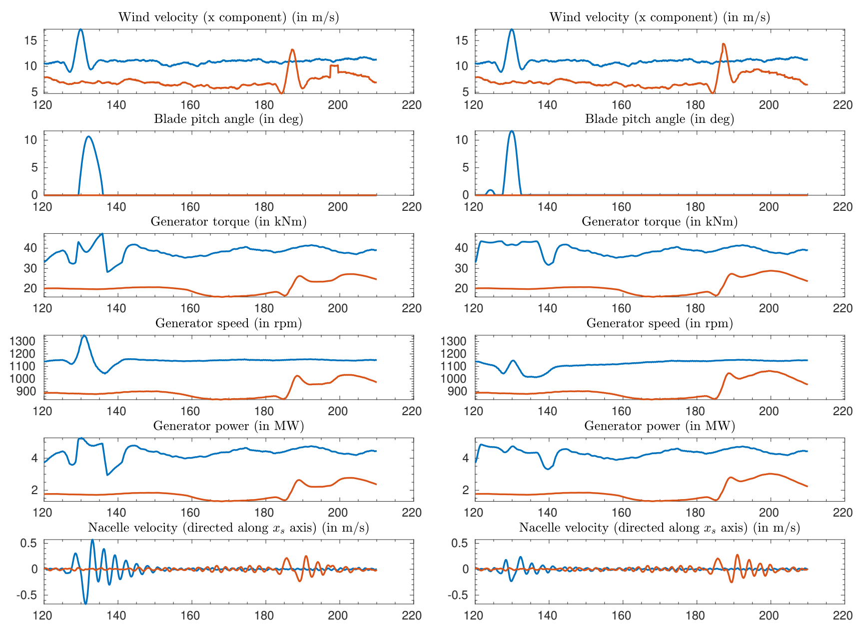
Acknowledgments. This study was funded by the German Federal Ministry of Education and Research (grant numbers 05M18MBA-MOReNet, 05M18VHA-MOReNet).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.



(a) Baseline for 6 m/s. (b) Optimization for 6 m/s.

Fig. 1: Baseline/optimization results of wind velocity 6 m/s over the time in s.



(a) Baseline for 11 m/s. (b) Optimization for 11 m/s.

Fig. 2: Baseline/optimization results of wind velocity 11 m/s over the time in s.

References

1. Annoni, J., Gebraad, P.M.O., Scholbrock, A.K., Fleming, P.A., van Wingerden, J.W.: Analysis of axial-induction-based wind plant control using an engineering and a high-order wind plant model. Wind Energy **19**(6), 1135–1150 (2016). https://doi.org/10.1002/we.1891
2. Bastankhah, M., Porté-Agel, F.: Experimental and theoretical study of wind turbine wakes in yawed conditions. Journal of Fluid Mechanics **806**, 506–541 (2016). https://doi.org/10.1017/jfm.2016.595
3. Gebraad, P.M.O., Fleming, P.A., van Wingerden, J.W.: Comparison of actuation methods for wake control in wind plants. In: 2015 American Control Conference (ACC). pp. 1695–1701 (2015). https://doi.org/10.1109/ACC.2015.7170977
4. Gebraad, P.M.O., Teeuwisse, F.W., van Wingerden, J.W., Fleming, P.A., Ruben, S.D., Marden, J.R., Pao, L.Y.: A data-driven model for wind plant power optimization by yaw control. In: 2014 American Control Conference. pp. 3128–3134 (2014). https://doi.org/10.1109/ACC.2014.6859118
5. Gonzalez Silva, J., Ferrari, R., van Wingerden, J.W.: Wind farm control for wake-loss compensation, thrust balancing and load-limiting of turbines. Renewable Energy **203**, 421–433 (2023). https://doi.org/10.1016/j.renene.2022.11.113
6. Hau, E.: Wind Turbines : Fundamentals, Technologies, Application, Economics. Springer, Berlin, 3rd edn. (2013). https://doi.org/10.1007/978-3-642-27151-9
7. Jensen, N.O.: A Note on Wind Generator Interaction. Tech. Rep. 2411, Risø National Laboratory (1983)
8. Jonkman, J., Butterfield, S., Musial, W., Scott, G.: Definition of a 5-MW Reference Wind Turbine for Offshore System Development. Tech. rep., National Renewable Energy Laboratory (2009). https://doi.org/10.2172/947422
9. Jonkman, J., Doubrawa, P., Hamilton, N., Annoni, J., Fleming, P.A.: Validation of FAST.Farm Against Large-Eddy Simulations. Journal of Physics: Conference Series **1037**(6), 062005 (2018). https://doi.org/10.1088/1742-6596/1037/6/062005
10. Kapp, S.: Binary .wnd specification (2009), https://forums.nrel.gov/t/binary-wnd-specification/180/4, accessed: September 2023
11. Katic, I., Højstrup, J., Jensen, N.O.: A Simple Model for Cluster Efficiency. In: Palz, W., Sesto, E. (eds.) EWEC'86. Proceedings. Vol. 1. pp. 407–410. Raguzzi, A. (1987)
12. National Renewable Energy Laboratory (NREL): SOWFA: Simulator fOr Wind Farm Applications. https://www.nrel.gov/wind/nwtc/sowfa.html, https://github.com/NREL/SOWFA (2020), accessed: July 2021
13. NREL: MATLAB Toolbox for OpenFAST. (2022), https://github.com/OpenFAST/matlab-toolbox
14. NREL: OpenFAST. Version 3.5.0 (2023), https://github.com/OpenFAST/openfast
15. NREL: pyFAST. Version 3.5.0 (2023), https://github.com/OpenFAST/python-toolbox
16. NREL: ROSCO. Version 2.8.0 (2023), https://github.com/NREL/ROSCO
17. NREL: r-test. Version 3.5.0 (2023), https://github.com/OpenFAST/r-test
18. Shirzadeh, K., Hangan, H., Crawford, C.: Experimental and numerical simulation of extreme operational conditions for horizontal axis wind turbines based on the iec standard. Wind Energy Science **5**(4), 1755–1770 (2020). https://doi.org/10.5194/wes-5-1755-2020