Multi-row extremum seeking for wind farm power maximization

Journal of Physics: Conference Series **2767** (2024) 032043 doi:10.1088/1742-6596/2767/3/032043

Multi-row extremum seeking for wind farm power maximization

Mario A Rotea[1], Devesh Kumar[2], Emmanuvel J Aju[1] and Yaqing Jin[1]

[1] Center for Wind Energy and Department of Mechanical Engineering, The University of Texas at Dallas, Richardson, TX 75080 USA.

[2] Drive System Design Inc., Farmington Hills, MI 48335, USA.

E-mail: rotea@utdallas.edu

Abstract. This paper presents results from wind tunnel experiments to evaluate power gains from wake steering via yaw control. An experimental scaled wind farm with 12 turbines in an aligned rectangular array is used. Wake steering is performed by yawing turbines using a closed-loop algorithm termed the Log-of-Power Proportional Integral Extremum Seeking Control (LP-PIESC). Two configurations are considered. In the first configuration, the turbines in the first two upstream rows are controlled. In the second case, yaw control is applied to the turbines in the first upstream row and the third row. For both cases, uncontrolled turbines have no yaw misalignment. The results show that by independent parallel maximization of the power sum of a reduced number of turbines, it is possible to obtain a close approximation of the true maximum power. The data shows that the LP-PIESC algorithm can converge relatively fast compared to traditional ESC algorithms.

1. Introduction

Wake steering via yaw control is an active research topic [1, 2]. This wind farm flow control technique involves intentional misalignment of selected turbines to divert their wakes away from turbines downstream in order to boost overall wind plant power output [1]. The yaw angle misalignments determine the lateral displacements of upstream wakes away from the downstream turbines [3, 4, 5, 6]. When the upstream turbines have a non-zero yaw angle (e.g., yaw misalignment) with respect to the incoming flow, they tend to experience power reductions, the redirection of their wakes allow downstream turbines to increase their power production [7, 8, 9]. Thus, intentional yaw misalignment of selected turbines can increase overall wind plant power production, with the ultimate goal of increasing annual energy production (AEP).

Meyers and co-authors [1] provided a detailed review of wake steering by yaw control with numerous references to the literature. From [1], it follows that the need for additional research to develop practical *feedback* control systems that can realize AEP gains by wake steering, despite uncertainty in flow models and atmospheric conditions, remains.

In the current study, we perform multi-row wind farm power optimization through a feedback real-time yaw optimization strategy dubbed the Log-of-power proportional-integral extremum seeking control (LP-PIESC). In [10], the authors conducted wind tunnel experiments with extremum seeking control assuming that only a single row of turbines (the most upstream row) can change its yaw angle to perform wake steering. In the current work, we extend the

approach in [10] to the case when turbines in inner rows within the wind farm can also perform wake steering by yaw control.

A schematic of our approach is shown in Fig. 1. In this method, clusters of turbines that are coupled via wake effects are first defined. Nearest neighbor turbines based on wind direction are used to define the clusters for a specific layout. By cluster we mean a reduced number of turbines, whose proximity to one another indicates the potential for power reducing wake effects. In this way, only the power output of a subset of turbines in the wind farm needs to be known to the wake steering yaw controllers, which reduces communication needs. The hypothesis is that increasing the power of each cluster individually would enhance the power production of the wind farm as a whole. This hypothesis has been tested with positive results in [10] for the case of yaw control on the upstream row only. In the current paper we focus on controlling the yaw angle of both upstream and downstream turbines. Power maximization is performed using the LP-PIESC algorithm within each cluster. The LP-PIESC takes cluster power (i.e., the feedback signal) as input and outputs the optimal yaw angles for selected turbines in each cluster.

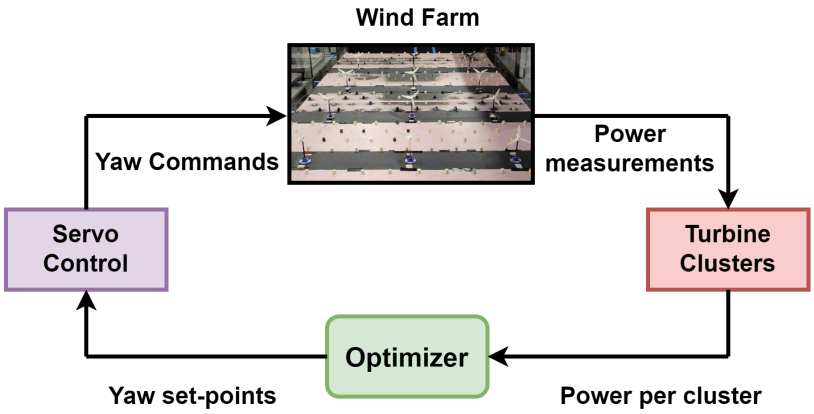

Figure 1: Schematic of real-time power maximization through cluster-based yaw control. Each cluster contains a turbine whose yaw angle is optimized to maximize the sum total power of the turbines in the cluster only. The turbines in a cluster are selected based on layout and wind direction.

Wind tunnel experiments demonstrate the performance of the cluster-based LP-PIESC for an array of 4 (streamwise direction) by 3 (spanwise direction) scaled turbines. That is, an array with 4 rows and 3 columns. Two cases are considered: **Case 1 (consecutive rows)** yaw control of the six turbines in rows #1 and #2 and **Case 2 (alternate rows)** yaw control of the three turbines in row #1 and the three turbines in row #3. Experimental evidence is given to support our hypothesis that increasing the power of each cluster individually enhances the power production of the wind farm as a whole, and that LP-PIESC can determine optimal yaw angles in practical times.

The paper is organized as follows. Section 2 describes the wind tunnel experimental setup and the cluster-based LP-PIESC. Experimental results are shown in Section 3. These results include the static power maps and the time series of power and yaw angle signals. Conclusions and areas of further work are in Section 4. A separate appendix contains the equations used for parameter estimation by the LP-PIESC.

2. Experimental Setup

Experiments were performed in the Boundary Layer and Subsonic Wind Tunnel (BLAST) at UT Dallas. The test section is 2.1 m high, 2.8 m wide and, 30 m long. To develop the turbulent boundary layer for the scaled wind farm cylindrical blocks with 2.5 cm diameter and 3 cm height were placed on the floor of the test section [11], as shown in Fig. 2. The scaled wind turbines

are based on the model proposed by Bastankhah *et al.* [12]. All turbines have the same rotor diameter $D = 0.2$ m and hub-height $z_{\text{hub}} = 0.2$ m. The coefficient of power and coefficient of thrust can reach up to 0.35 and 0.8, respectively, in free stream conditions. The blockage ratio of the turbine was 0.53% based on the rotor sweeping area and wind tunnel cross-section, which led to negligible blockage effects. The wind farm array (4 rows x 3 columns) is placed near the back wall as shown in Fig. 2.

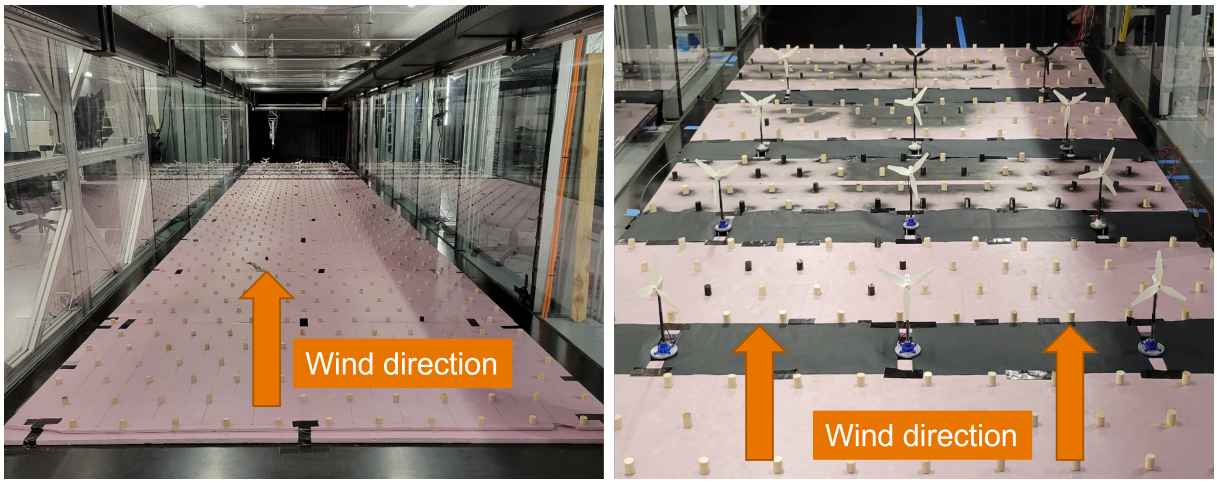

Figure 2: UTD BLAST - test section with roughness elements (**left**); scaled wind turbines in aligned layout (**right**).

The generator is a DCX16L motor from Maxxon with a diameter of 16 mm and a 1-ohm resistance was used as the electrical load for all turbines. This led to an optimal tip-speed ratio of $\lambda \approx 4$ with no yaw misalignment. The turbine loading was kept constant across all tests. Blade pitching is not supported for these turbines; thus, blade angles are fixed at 0°. The incoming flow has hub-height mean wind speed of ~ 6.4 m/s, turbulence intensity of $\sim 9\%$, and shear coefficient of 0.2, resulting from a 110 rpm fan speed and the roughness elements used. Figure 3 shows the overall wind farm layout with the two cases investigated.

2.1. Definition of clusters for power maximization

Power maximization using yaw control is performed by dividing the farm into "clusters," and maximizing the power of each cluster in parallel. Clustering is done using a simple "nearest neighbor" approach based on the layout and the wind direction. The perimeters of these clusters are shown with blue dotted lines and red dotted lines. Table 1 summarizes the cluster power maximized by each LP-PIESC algorithm for Case 1 (i.e., yaw control of first two rows). For brevity, no table is given for Case 2 (i.e., yaw control of alternate rows). Instead, we explain Case 2 as follows: from the diagram on the right-hand side of Fig. 3, T1 is yawed to maximize the power sum of T1 and T4 (blue perimeter), while T7 is yawed to maximize the power sum of T7 and T10 (red perimeter), and this strategy is repeated for the other two columns. This "clustering approach" solely uses communication of power signals within turbines in a cluster, which can simplify implementation. The approach requires knowledge of the wind farm layout and wind direction. While this strategy can increase wind farm power by yawing selected turbines only, it does not guarantee that the power increase is the global maximum for the total wind farm.

2.2. Optimizer implementation with LP-PIESC

The "optimizer block" in Fig. 1 runs six independent LP-PIESC algorithms in parallel to maximize the power of the six clusters defined in Section 2.1. Each LP-PIESC algorithm receives

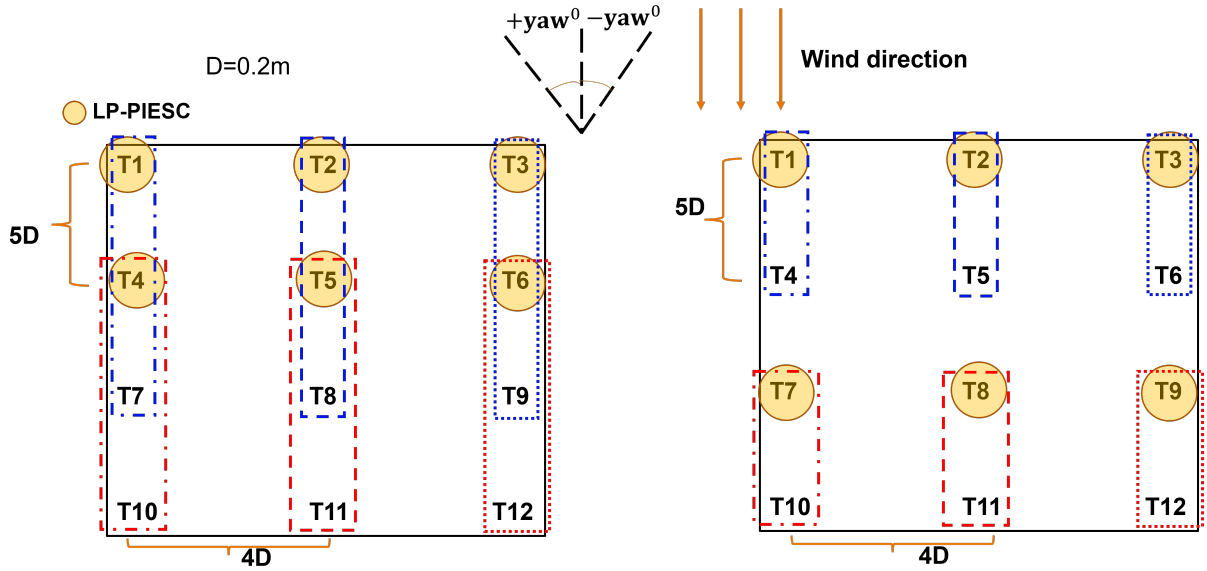

Figure 3: Wind farm layout - LP-PIESC implemented in turbines with yellow circles. Case 1: yaw control on rows #1 and #2 (**left**). Case 2: yaw control on rows #1 and #3 (**right**). Wind turbine clusters enclosed by the dashed rectangular perimeters as explained in Section 2.1.

Table 1: Cluster power maximized for the LP-PIESC algorithms of Case 1.

LP-PIESC implemented on	**Commanded yaw angle**	**Cluster power maximized**
T1	γ_1	T1+T4+T7
T2	γ_2	T2+T5+T8
T3	γ_3	T3+T6+T9
T4	γ_4	T4+T7+T10
T5	γ_5	T5+T8+T11
T6	γ_6	T6+T9+T12

the power signal from the appropriate cluster ($P_{\text{cluster},i}$ where $i = 1, 2, \ldots, 6$) as an input, and calculates the yaw angle commands: γ_k where $k = 1, 2, 3; 4, 5, 6$ for Case 1 and $k = 1, 2, 3; 7, 8, 9$ for Case 2; see Fig. 3. Figure 4 depicts the real-time controller for each LP-PIESC algorithm. First, a 1 s moving average filter is used to condition the cluster power. We normalize the filtered power by 1 W to render the control input nondimensional. The LP-PIESC receives the log of this normalized power and calculates the yaw command to a position servo controller.

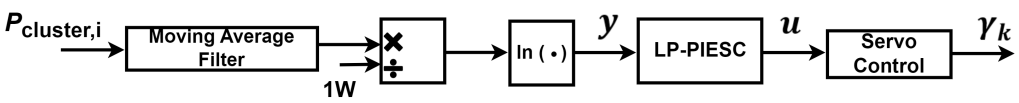

Figure 4: Schematic of the algorithm used to determine the yaw angle γ_k that maximizes the cluster power $P_{\text{cluster,i}}$. (Six algorithms of this form work in parallel in the optimizer block in Fig. 1.)

The hardware implementation of the block diagram in Fig. 4 is described in [10] and not repeated here for brevity. Key elementss include: The data acquisition and the control algorithms were programmed in NI LabVIEW. The LP-PIESC loop has a 50 Hz sampling frequency. The servo control uses an Arduino Mega 2560 to drive the yaw servo motor (Actobotics SG-12). The Arduino generates a 50 Hz PWM signal to command the servo motor.

The Log-of-Power Proportional-Integral Extremum Seeking Control (LP-PIESC) strategy used in this work is from [13]. This algorithm is a gradient ascent method that requires tuning several parameters to maximize the system's performance index without detailed physical models. The log-of-power as the performance index enables consistent convergence for different mean wind speeds. That is, tuning the LP-PIESC parameters at one single wind condition, yields the same performance at any other wind condition in partial load. This property has been documented in [14, 15] using other variants of extremum seeking and has been shown for LP-PIESC in [13]. The PIESC algorithm is from [16]. This algorithm has faster convergence than other widely used variants of extremum seeking [17] for the reasons given below.

Let u denote the calculated yaw angle as shown in Fig. 4. The PIESC controller is given by

$$\begin{cases} u = -k_p\hat{\theta}_1 + \hat{u} + d(t) \\ \dot{\hat{u}} = -\frac{1}{\tau_I}\hat{\theta}_1 \end{cases} \tag{1}$$

where k_p is the proportional gain, τ_I is the time constant and $\hat{\theta}_1$ is a scalar parameter to be estimated. This parameter is representative of the derivative of the log-of-power of a given cluster with respect to the change in the controlled yaw angle for that cluster. The dither signal, denoted by d, is a sinusoid. The proportional term $k_p\hat{\theta}_1$ accelerates convergence relative to the more widespread version of extremum seeking [17], which is essentially the integrator equation in (1) for $\hat{u}$. Intuitively, the effect of adding the proportional term is like the increase in control bandwidth obtained when replacing a pure integral controller with a PI control law. Convergence time improvements are also due to the method used to determine the unknown time-varying parameter $\hat{\theta}_1$. This method is not the traditional perturbation/demodulation method to extract gradient information in earlier versions of ESC [17], but rather it is based on ideas from estimation of time-varying parameters, adaptive control and continuous-time recursive least squares with forgetting [16] [18] [19]. The parameter estimation method is discussed in Appendix A alongside the LP-PIESC tuning parameters for Cases 1 and 2.

3. Experimental Test Results

The cluster-based LP-PIESC was first implemented on Case 1 (turbines T1 to T6) and then on Case 2 (turbines T1 to T3 and T7 to T9). To have a performance baseline, we also show the total wind farm power variation with yaw angles, which we refer to as "static power maps" because we compute them when the flow is converged at each yaw angle.

3.1. Static Power Maps

The static map for Case 1 was obtained by varying the yaw angles of rows #1 and #2 (γ_{row1}, γ_{row2}) from -40° to $+40^\circ$ every 10° for the wind farm setup in Fig 3. All other turbines had zero yaw misalignment. The result is shown on the left-hand side of Fig. 5. The plot shows the farm power output P normalized with the base case P_0 (no yaw misalignment). The total farm power showed maximum improvement of $\sim$9% near (γ_{row1}, γ_{row2})=(30°, 20°) and (-30°, -20°). Similar steps were taken to obtain the static map for Case 2 (i.e., varying rows #1 and #3 yaw angles only) and the result is shown on the right-hand side of Fig. 5. The maximum power improvement was around $\sim$9% near (γ_{row1}, γ_{row3})=(30°, 10°). For negative yaw angles the maximum improvement is $\sim$7.5% near (γ_{row1}, γ_{row3})=(-30°, 0°).

3.2. LP-PIESC Results

Six independent LP-PIESC algorithms were implemented in parallel to search for the optimal yaw angles for Cases 1 and 2. The experimental conditions can be found in Section 2. Each experiment is 7-minutes long. For the first 3 minutes, all turbines have no misalignment. The algorithms are turned on at $t = 3$ min, and run until $t = 7$ min. Average power is calculated for

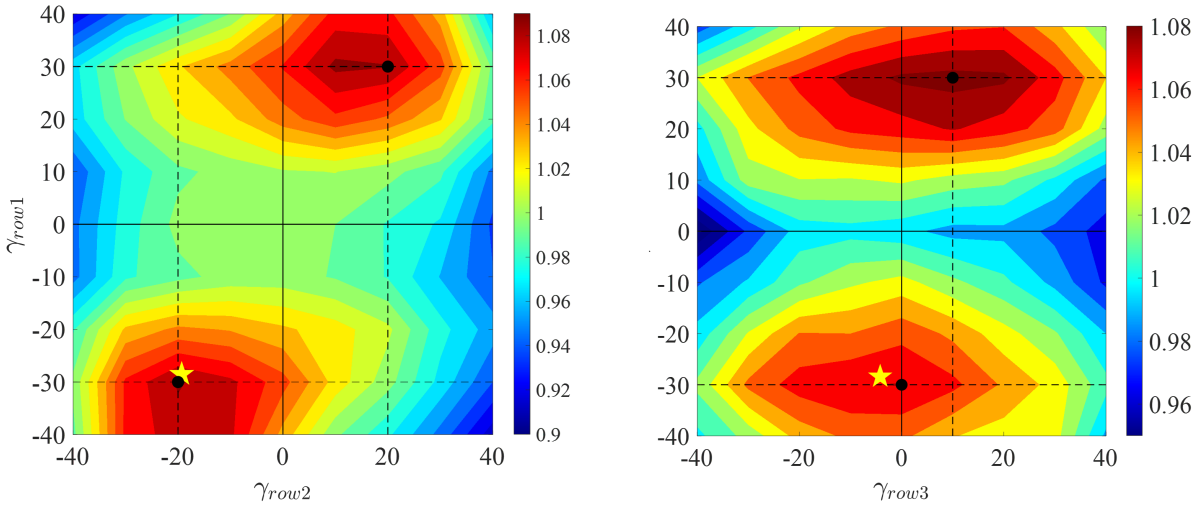

Figure 5: Static maps of the total farm power output normalized by the base case (P_0) for the two cases shown in Fig. 3: Case 1 (**left:**), Case 2 (**right**). Black dots indicate the maximum power points; yellow stars indicate the yaw angles at which the LP-PIESC converged, which are explained in Section 3.2.

the first three minutes (no yaw control) and the last three minutes (control converged to new yaw angles).

3.2.1. Case 1. Turbines T1 to T6 are equipped with LP-PIESC algorithms to maximize the power of their respective clusters, as defined in Section 2.1. The results are shown in Fig. 6. The left-hand side of this figure shows the yaw angles for the six experimental runs. The yaw angles of rows #1 and #2 converge to $\gamma_{\text{row1}} = -29°$ and $\gamma_{\text{row2}} = -19°$, respectively. The percentage changes in total farm average power between no wake steering (0.1 to 3 min) and with wake steering (4 to 7 min) are shown with numerical values in red font. The mean total power change amongst all six runs is 8.9%. The results are in reasonable agreement with the static map in Fig. 5 (left-hand side). Individual turbine power changes are shown on the right-hand side of Fig. 6. This plot is obtained by averaging the power of each turbine and total farm over all six experimental runs. The front three turbines are yawed, so they lose power. The yawing of second-row turbines reduces the gains these turbines could achieve if they were not misaligned, but their misalignment leads to high power gains in rows #3 and #4.

3.2.2. Case 2. Turbines T1 to T3 and T7 to T9 are equipped with independent LP-PIESC algorithms to maximize the power of their respective clusters, as defined in Section 2.1. The results are shown in Fig. 7. The yaw angles of rows #1 and #3 converge to $\gamma_{\text{row1}} = -28.4°$ and $\gamma_{\text{row3}} = -4.2°$, respectively, for all six experimental runs. The percentage changes in total farm average power between no wake steering (0.1 to 3 min) and with wake steering (4 to 7 min) are shown with numerical values in red font. The mean total power change amongst all six runs is 5.6%. Comparison with the static map in Fig. 5 (right-hand side) shows that the algorithms converge to the negative yaw angles close to the local optimum $\gamma_{\text{row1}} = -30°$ and $\gamma_{\text{row3}} = -0°$, but not to the global optimum with positive yaw angles. Individual turbine power changes are shown on the right-hand side of Fig. 7. This plot is obtained by averaging the power of each turbine and total farm over all six experimental runs. The front three turbines are yawed, so they lose power. All other turbines show individual power gains. Note that even row #3 shows a small overall power gain, which could be due to a higher wake velocity downstream of row #2 combined with small yaw misalignment for row #3 turbines.

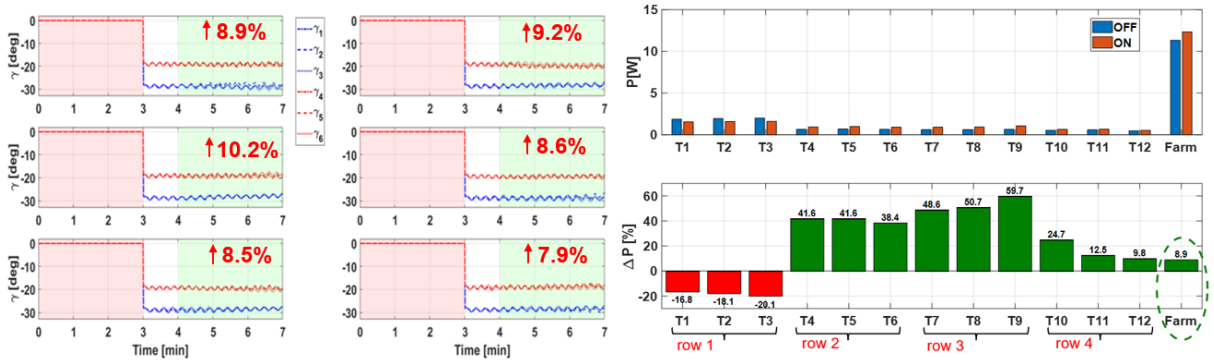

Figure 6: Six independent runs of the LP-PIESC for Case 1. **Left:** time series of individual yaw angles - row #1 (blue), and row #2 (red). Total farm power increase shown in red font. **Right:** Individual power changes due to LP-PIESC implementation averaged over the six runs. The top plot shows the mean power when the LP-PIESC is OFF (0.1 to 3 min) and when the LP-PIESC is converged (4 to 7 min). The bottom plot shows the percentage change in the mean power.

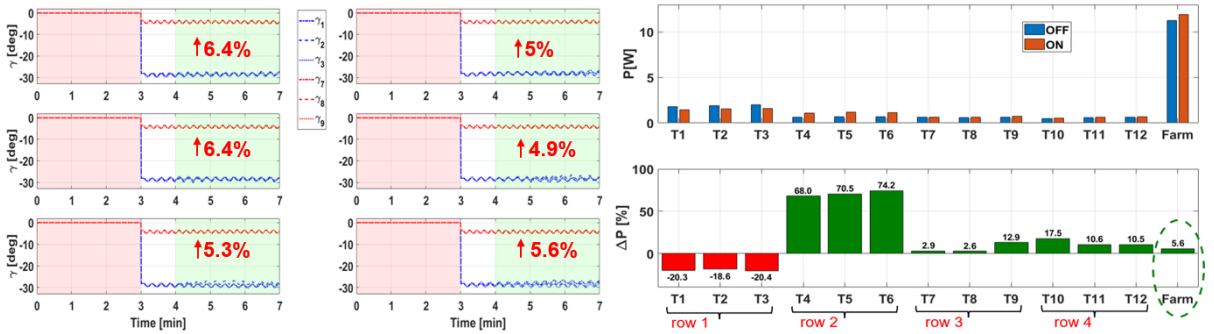

Figure 7: Six independent runs of the LP-PIESC for Case 2. **Left:** time series of individual yaw angles - row #1 (blue) and row #3 (red). Total farm power increase shown in red font. **Right:** Individual power changes due to LP-PIESC implementation averaged over the six runs. The top plot shows the mean power when the LP-PIESC is OFF (0.1 to 3 min) and when the LP-PIESC is converged (4 to 7 min). The bottom plot shows the percentage change in the mean power.

3.2.3. Further discussion. When comparing the results of Case 1 with Case 2, it follows that the latter case achieves less power gain (5.6% compared with 8.9%) but the yaw angles required are smaller; i.e., about the same yaw angle for the first row but $\gamma_{\text{row2}} = -19°$ compared with $\gamma_{\text{row3}} = -4.2°$. Note also that while Case 1 shows convergence close to a global optimum (as per static map in Fig. 5); the same conclusion does not apply to Case 2, which attains convergence close to the local optimum at $\gamma_{\text{row1}} = -30°$, $\gamma_{\text{row3}} = 0°$. While we do not have a rigorous proof, a plausible explanation is that in Case 2, as shown in Fig. 3, the LP-PIESC in each cluster maximizes the power sum of two turbines only, which makes the formulation suboptimal, but simpler to implement. The reduced yaw angle for row #3 can also be advantageous for mitigating increases in side-force fluctuations, which as reported in [11] grow with yaw angle.

In Case 1, turbines T4, T5 and T6 run their own LP-PIESC algorithms to maximize the power of the clusters shown in Table 1 and also participate in the optimization done by turbines T1, T2 and T3, respectively. A reader may wonder if such overlap would create convergence issues due the fact that there is always one turbine participating in two separate clusters. Note from Fig. 6 that no convergence issues arise as a result of second row turbines participating in two separate clusters. The fact that the power increase (8.9%) is close to a global optimum helps rule out any negative interaction between the two optimizations. In addition, Table 2

shows the linear Pearson correlation coefficient computed using the Matlab function `correl` for the power time series between minutes three and four, which is the interval where the transition from no misalignment to optimal yaw angles takes place. Due to space limitations we show the correlation coefficients between the power signals of T1 and T4 (the turbines implementing LP-PIESC) with the power signals of the two clusters these turbines seek to maximize. These "transient" correlation coefficients are consistent with the expected behaviour. The only negative coefficient is the correlation between P_1 and $P_4 + P_7 + P_{10}$, which shows that the power of this cluster increases with a decrease on P_1 during the transition. Note also that an increase in P_4 leads to increase in both clusters as shown in the second row of Table 2, which suggests that there is no adverse interaction between these clusters.

	$P_1 + P_4 + P_7$	$P_4 + P_7 + P_{10}$
P_1	0.31	-0.33
P_4	0.79	0.91

Table 2: Correlation coefficients between selected power signals in the interval $[3, 4]$ min.

Figures 8 and 9 show the zoomed time series for the yaw angles in both configurations (Cases 1 and 2). In both cases, the first row yaw angles determine the convergence time because, the turbines have to turn largest yaw angle amplitude (between 28° and 29°). It takes about 0.01 minute for row #1 to converge. Given that the wind conditions (see Section 2) are similar to below rated conditions for a typical utility-scale turbine, simple scaling with the ratio of rotor diameters yields an approximation, albeit gross, to the actual convergence times. Using such scaling, the convergence time would be approximately 6.5 min for the IEA 3.4 MW reference turbine ($D = 130$ m) and 12 min for the IEA 15 MW reference turbine ($D = 240$ m), assuming maximum yaw rate constraints are met for these turbines. We do not have rigorous proof for the rapid convergence of the LP-PIESC algorithms for yaw control. We have observed this phenomenon in other applications [10, 13].

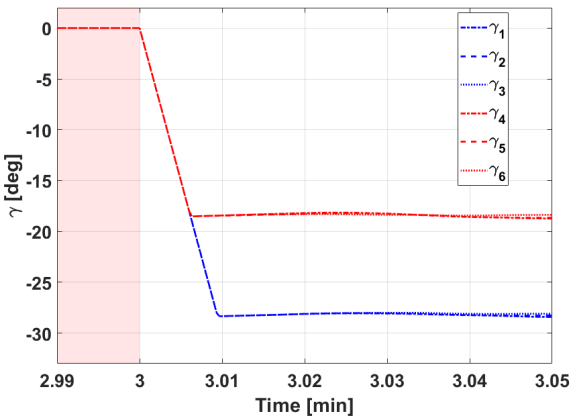

Figure 8: Zoomed time series of the individual yaw angles - row #1 (blue), row #2 (red) for Case 1. Convergence takes about 0.01 min. for row #1 turbines and 0.006 min. for row #2 turbines.

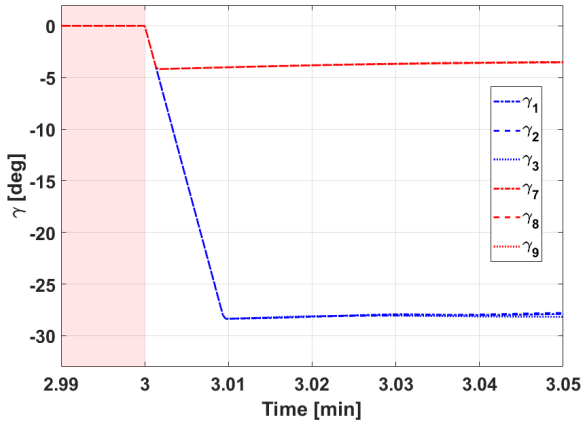

Figure 9: Zoomed time series of the individual yaw angles - row #1 (blue), row #3 (red) for Case 2. Convergence takes about 0.01 min. for row #1 turbines and 0.001 min. for row #3 turbines.

4. Conclusions

Experimental results have shown that multi-row wake steering via cluster-based LP-PIESC yaw control is a viable approach for increasing the power output of a wind farm. Experimental

evidence is given to support our hypothesis that increasing the power of each cluster in parallel enhances the power production of the wind farm as a whole, and that LP-PIESC can determine optimal yaw angles in practical times. Due to the maximization of the power of selected clusters within the wind farm, global maximum farm power may not always be achieved, but power increases are still obtained. The experiments presented assume a fixed wind direction. Varying the wind direction would require changing the clusters that need to be optimized. Autonomous cluster selection and its integration with extremum-seeking control algorithms are beyond the scope of the present study, and require further investigation. In addition, further work is needed to establish guidelines for LP-PIESC parameter design in order to obtain a practical feedback solution for wake steering via yaw-control using cluster-based LP-PIESC.

References

[1] Meyers J, Bottasso C, Dykes K, Fleming P, Gebraad P, Giebel G, Göçmen T and van Wingerden J W 2022 *Wind Energy Science* **7** 2271–2306
[2] Howland M F, Ghate A S, Quesada J B, Pena Martínez J J, Zhong W, Larrañaga F P, Lele S K and Dabiri J O 2022 *Wind Energy Science* **7** 345–365
[3] Adaramola M and Krogstad P 2011 *Renewable Energy* **36** 2078–2086 ISSN 0960-1481
[4] Bastankhah M and Porté-Agel F 2016 *Journal of Fluid Mechanics* **806** 506–541
[5] Howland M F, Bossuyt J, Martínez-Tossas L A, Meyers J and Meneveau C 2016 *Journal of Renewable and Sustainable Energy* **8** 043301
[6] Fleming P, Annoni J, Scholbrock A, Quon E, Dana S, Schreck S, Raach S, Haizmann F and Schlipf D 2017 *Journal of Physics: Conference Series* **854** 012013
[7] Fleming P A, Gebraad P M, Lee S, van Wingerden J W, Johnson K, Churchfield M, Michalakes J, Spalart P and Moriarty P 2014 *Renewable Energy* **70** 211–218 ISSN 0960-1481
[8] Gebraad P M O, Teeuwisse F W, van Wingerden J W, Fleming P A, Ruben S D, Marden J R and Pao L Y 2016 *Wind Energy* **19** 95–114
[9] Howland M F, Lele S K and Dabiri J O 2019 *Proceedings of the National Academy of Sciences* **116** 14495–14500
[10] Kumar D, Rotea M A, Aju E J and Jin Y 2022 *Wind Energy* 1–27
[11] Aju E J, Kumar D, Leffingwell M, Rotea M A and Jin Y 2023 *Renewable Energy*
[12] Bastankhah M and Porté-Agel F 2017 *Energies* **10** 908 ISSN 1996-1073
[13] Kumar D and Rotea M A 2022 *Energies* **15** 1–24 ISSN 1996-1073
[14] Rotea M A 2017 *IFAC-PapersOnLine* **50** 4504 – 4509 ISSN 2405-8963 20th IFAC World Congress
[15] Ciri U, Leonardi S and Rotea M A 2019 *Wind Energy* **22** 992–1002
[16] Guay M and Dochain D 2017 *Automatica* **77** 61–67 ISSN 0005-1098
[17] Krstić M and Wang H H 2000 *Automatica* **36** 595–601 ISSN 0005-1098
[18] Krstic M, Kokotovic P V and Kanellakopoulos I 1995 *Nonlinear and Adaptive Control Design* (John Wiley & Sons, Inc.) ISBN 0471127329
[19] Shaferman V, Schwegel M, Glück T and Kugi A 2021 *European Journal of Control* **62** 105–112
[20] Adetola V and Guay M 2011 *International Journal of Adaptive Control and Signal Processing* **25** 155–167

Appendix A. PIESC algorithm and design parameters

We now describe the method to estimate the parameter $\hat{\theta}_1$ in Eqn. 1. For brevity, we give main ideas only; details can be found in [16] and the references therein. Let $y(t)$ denote the log power signal entering the LP-PIESC algorithm in Fig. 4. Following [16], one can parametrize the time derivative of y as

$$\dot{y} = \theta_0 + \theta_1(u - \hat{u}) = \phi^T\theta \tag{A.1}$$

where $\phi = [1, (u - \hat{u})]^T$ is the regressor and $\theta = [\theta_0, \theta_1]^T$ the two-dimensional unknown time varying parameter. Here θ_0 represents the dynamics between the yaw input to the output y. This parameter is not used in (1), but it is required to estimate θ_1 properly [16]. The parameter θ is estimated by reducing an output prediction error e of the log of power signal y, given by

$$e = y - \hat{y} \tag{A.2}$$

where $\hat{y}$ represents the prediction of y, calculated from the following ODE

$$\dot{\hat{y}} = \phi^T \hat{\theta} + Ke + c^T \dot{\hat{\theta}}. \tag{A.3}$$

In this equation, K is a positive scalar to be determined and $c(t)$ a filtered regressor satisfying

$$\dot{c}^T = -Kc^T + \phi^T \tag{A.4}$$

Note that the output prediction dynamics $\dot{\hat{y}}$ in (A.3) contains a model of the dynamics $\dot{y}$ plus two additional terms, one proportional to the error e and a second one proportional to the time derivative of the time-varying parameter estimate $\dot{\hat{\theta}}$. The parameter estimation update law is

$$\boxed{\dot{\hat{\theta}} = \text{Proj}(\Sigma^{-1}(ce - \sigma\hat{\theta}), \hat{\theta})} \tag{A.5}$$

where Proj$(\cdot)$ is a Lipschitz projection operator designed to ensure that the estimates are bounded within the constraint set and guarantee stability. This projection was implemented per Appendix E of [18] and the constraint set adaptation was adopted as per Adetola et al. [20]. The gain matrix Σ^{-1} is propagated with the following ODE

$$\dot{\Sigma}^{-1} = -\Sigma^{-1} cc^T \Sigma^{-1} + k_T \Sigma^{-1} - \sigma \Sigma^{-2} \tag{A.6}$$

with σ and k_T positive scalar constants. The matrix update law (A.6) is similar to the one in continuous-time least-squares with forgetting [19].

A condition for the convergence of the PI-ESC algorithm [16] is excitation provided by the filtered regressor $c(t)$ in (A.4). This is quantified by the following persistence of excitation (PE) condition: there exists constants $\rho > 0$ and $T > 0$ such that

$$\int_t^{t+T} c(\tau)c(\tau)^T \, d\tau \;\geq \rho I \quad \forall t > 0 \tag{A.7}$$

In our case, the PE condition is satisfied relatively quickly, which provides intuitive justification behind the rapid convergence of the LP-PIESC. For example, for Case 1, integrating $c(t)c(t)^T$ for 0.01 min after LP-PIESC starts (see Fig. 8) yileds $\rho = 0.0108$ for T1 and $\rho = 0.0085$ for T4.

Table A1 shows the designed parameters for Case 1 (yaw control on rows #1 and #2 only). The LP-PIESC parameters for Case 2, are identical, with the exception of three parameters, which are shown in parenthesis. All parameters with the exception of dither frequency and amplitude were obtained by trial and error. These latter parameters are designed as in [10]. The dither frequency was selected within the 3 dB bandwidth of the plant dynamics between changes in yaw angle and the averaged power signal. The response of the filtered turbine power under step change in the yaw angle can be approximated with first-order dynamics. The estimated time constant (τ) from the yaw angle to the filtered turbine power ranges from 1 to 2 s. The largest time constant was adopted; i.e., 2 s giving rise to 0.5 rad/s bandwidth.

Table A1: LP-PIESC Parameters for yaw control for Cases 1 and 2.

Parameter	**Cases 1,2: row #1**	**Case 1: row #2 (Case 2: row #3)**
Dither Frequency (ω)	0.4 rad/s	0.5 rad/s
Dither Amplitude (a)	0.75 deg	0.7 deg
k_T	20 rad/s	20 rad/s
K	20 rad/s	20 rad/s
σ	5 (deg·s/rad)^2	5 (10) (deg·s/rad)^2
k_p	1 deg^2·s/rad	1 (2) deg^2·s/rad
τ_I	9.8×10^{-2} deg^{-2}	1.5 (5) $\times 10^{-2}$ deg^{-2}