

A data-driven approach for wind farm power maximisation using Bayesian Optimisation

Applied on a scaled wind farm and a high-fidelity simulation

By

E.M. de Boer, K.O. Koerten, J. Langeveld, T.L. van der Zijden

Supervised by

prof.dr.ir. J.W. van Wingerden, dr. ir. M. Kok, ir. B.M. Doeckemeijer, ir. D.C. van der Hoek

December 21, 2018



Figure 1: Wakes visualised by fog at the Horns Rev wind farm. Source: Christian Steiness

Abstract

The objective of this research is to show that the power output of a scaled wind farm can be optimised, using a model free optimisation algorithm. Traditionally, wind turbines are controlled to optimise their individual power production. The result of this is that the first turbine has a high power output while the downstream turbines experience turbulent wind, resulting in lower power as well as higher damage. A way to optimise the power production of the entire wind farm is by misaligning some of the turbines in the yaw direction, thereby steering the turbulent wake away from downstream turbines. Finding the right yaw angle, however, is time and energy consuming. In this study, Gaussian processes combined with Bayesian Optimisation have been used to approach the maximum power production of a scaled wind farm in as few measurements as possible. The scaled wind farm is used because measurements that would be time or energy consuming on respectively a high fidelity simulation or an actual wind farm can be taken here.

keywords. wind turbine, wind farm, wind tunnel, wake steering, wake redirection, Gaussian Process, Bayesian Optimisation, data-efficient, model-free, high-fidelity simulation.

1 Introduction

Wind turbines are being built in large amounts. Only in 2017, 10.8% of all wind turbines worldwide have been constructed (*Added wind power in 2017* (2018)). Wind turbines are located close to each other in offshore wind farms to reduce cabling and maintenance costs. Currently, turbines are controlled to maximise their individual power production: generator torque and blade pitch are controlled to optimise power capture and the rotor yaw is aligned with the wind direction. This is called greedy control. Wind turbines extract energy from the wind flow creating an area characterised by higher turbulence and lower wind speeds, called a wake. This wake negatively impacts the power production of downstream turbines. It is possible to steer the wake to a direction more favourable for downstream turbines by misaligning the upstream rotor yaw with respect to the wind. Alternatively, generator torque and individual blade pitch can be changed to influence wake intensity. High-fidelity simulations have shown that, for certain weather conditions, both these control strategies lead to lower local power production, but potentially increased total power production. P. M. O. Gebraad (2014) suggests that, according to current research, there is more potential for yaw-based wake redirection control. Also, Barthelmie et al. (2009) shows that wake effects cause energy losses up to 12% for a wind farm compared to freestream turbines.

Finding the optimal yaw misalignment in a sufficiently short time-span for varying conditions is an unsolved problem. There are several approaches to total wind farm control being researched, as summarised in Doekemeijer et al. (2019). They can be divided into three types of controllers.

1. Open-loop model-based controllers are based on running an optimisation algorithm on a surrogate model, then feeding the control settings derived from this model to the real wind farm. They are highly dependent on the accuracy of the model. Currently, no model is accurate enough for robust results, so a closed-loop control framework is necessary.
2. Closed-loop model-based control adds the advantage of feedback. However, for real-time control of wind farms, this adds the necessity of computing an update of the model with the measurements within a time-step of the control algorithm. Currently, only low-fidelity models have sufficiently low computational cost to be used as a model on the minutes-scale. The control inputs derived from these models do not take all physical effects into account, introducing errors.
3. closed-loop model-free control is based on only using measurements to decide the control settings. Research has been done by P. Gebraad & Van Wingerden (2015) into a data-driven control framework with

inputs generator torque and blade pitch. Model-free control methods typically require a large amount of data points to model the target function properly, and tend to converge to a local optimum.

In their paper, Park & Law (2015) propose a Bayesian Optimisation approach for wind farm control and test it on a scaled wind farm. Using Gaussian Processes, Bayesian Optimization is able to find the global optimum of complex target systems requiring a relatively low amount of data points, solving some problems of model-free control methods.

This paper aims to answer the question: “Is it possible to optimise the total power production of a wind farm, by applying yaw-based wake steering, using Gaussian Processes and Bayesian Optimisation?” Firstly, a control algorithm based on Gaussian Processes and Bayesian Optimization is developed. Using the turbine yaws as inputs and total wind farm power as output this algorithm tries to find the optimal yaw angles for maximum power production in as few measurements as possible. Then, it is implemented on a scaled wind tunnel, parametric simulation (FLORIS) and a high-fidelity simulation (SOWFA) and compared to benchmark experiments, with the aim of showing that BO can be scaled up to real-life wind farms. Finally, different settings of the Gaussian Process and Bayesian Optimization will be tested for this application.

This paper is organised as follows. In section 2, the Bayesian Optimisation approach and the various benchmark experiments are discussed, followed by the test-benches: the scaled wind tunnel, FLORIS and SOWFA. Next, section 3 gives a description of the measurements and the results are presented. In section 4 the results are discussed and recommendations are given. Finally in section 5 the paper is concluded.

2 Method

All test-benches used in this research model a wind farm consisting of three turbines. The different experiments are characterised by the following parameters: rotor diameter D , distance between turbines d (where $d_{1-2} = d_{2-3}$), wind angle ψ , average wind speed U_w and wind turbulence intensity TI . The yaw angles of the two upstream turbines γ_1 and γ_2 are controlled, while the downstream turbine γ_3 always faces the wind for maximum power production.

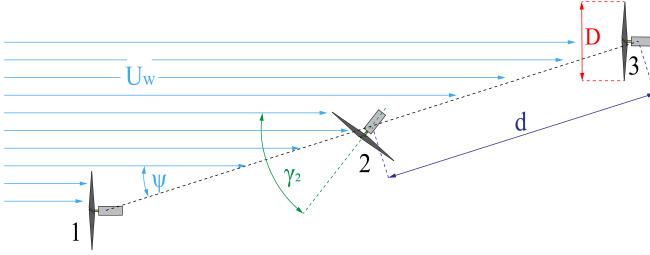


Figure 2: Schematic view of the wind farm. The denoted variables represent the average wind speed U_w , the wind angle relative to the line of turbines ψ , the rotor Diameter D , the yaw angle of turbine 2 relative to the wind direction γ_2 and the distance between neighbouring turbines d .

In this set-up, the steady-state wind farm power production can be interpreted as an unknown objective function $P(\bar{\gamma})$, where P is the total power output of the wind farm and $\bar{\gamma}$ is the input vector with γ_1 and γ_2 which are assumed to be deterministic.

2.1 Bayesian Optimisation approach and benchmark settings

2.1.1 Gaussian Processes and Bayesian Optimisation

Finding the optimal yaw angles producing the most power for the total wind farm of three turbines can be formulated as

$$\bar{\gamma}_{\text{optimal}} = \operatorname{argmax}_{\gamma} P(\bar{\gamma}) \triangleq \sum_{i=1}^3 P_i(\bar{\gamma}) \quad (1)$$

To learn more about the objective function, the power output can be measured for specific yaw angles. It is possible to make predictions based on the previous measurements, which is generally known as regression. The unknown function $P(\bar{\gamma})$ is modelled using Gaussian Process (GP) regression. Making measurements is expensive so the algorithm has to converge to $\bar{\gamma}_{\text{optimal}}$ in as less iterations as possible. Bayesian Optimization (BO) is used to achieve this goal.

Gaussian Processes

The power output for a point that has not been measured yet is treated as a random variable. Only noisy observations of the power output can be measured, due to measurement noise, so Gaussian white noise is added to the random variable. Taking a finite set of input points they can be modelled as a GP. A GP is completely described by its *mean function* $m(\gamma)$ and *covariance function*: $k(\gamma, \gamma')$. The measured power outputs $P^{1:n} = (P^1, \dots, P^n)$ and unmeasured outputs $P^* = P(\bar{\gamma}^*)$ follow a multivariate Gaussian distribution (Rasmussen & Williams (2006)) as

$$\begin{bmatrix} P^{1:n} \\ P^* \end{bmatrix} \sim N \left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & \mathbf{k} \\ \mathbf{k}^T & k(\bar{\gamma}, \bar{\gamma}) \end{bmatrix} \right) \quad (2)$$

Where \mathbf{K} is the covariance matrix, whose (p, q) th entry is defined as $K_{pq} = k(\bar{\gamma}^p, \bar{\gamma}^q)$, which is the covariance between measurement point p and q and $\mathbf{k} = [k(\bar{\gamma}^{n+1}, \bar{\gamma}^1) \quad \dots \quad k(\bar{\gamma}^{n+1}, \bar{\gamma}^n)]$.

The covariance function is used to calculate $k(\bar{\gamma}^p, \bar{\gamma}^q)$. The value of $k(\bar{\gamma}^p, \bar{\gamma}^q)$ describes the of similarity between pairs of yaw angle vectors. A very popular choice for the covariance function is the squared exponential function. However it was argued by Stein (1999) that the squared exponential functions results in a degree of of smoothness that in practice does not occur. A Matern covariance function is proposed instead. In this research a Matern covariance function with smoothness parameter $\frac{3}{2}$ is used.

The historical data set at the n^{th} iteration, $\mathbf{D}^{1:n} = \{(\bar{\gamma}^i, P^i) | i = 1, \dots, n\}$ is used to predict the unmeasured outputs $P^* = P(\bar{\gamma}^*)$. Using the equations of Rasmussen & Williams (2006) an expression for the *posterior distribution* on P^* is derived where $P^* \sim N(\mu(\bar{\gamma}^* | \mathbf{D}^n), \sigma^2(\bar{\gamma}^* | \mathbf{D}^n))$ with the mean and variance function expressed as

$$\mu(\bar{\gamma} | \mathbf{D}^n) = \mathbf{k}^T (\mathbf{K})^{-1} \mathbf{P}^{1:n} \quad (3)$$

$$\sigma^2(\bar{\gamma} | \mathbf{D}^n) = k(\bar{\gamma}, \bar{\gamma}) - \mathbf{k}^T (\mathbf{K})^{-1} \mathbf{k} \quad (4)$$

Hyperparameters

Four free parameters, called hyperparameters, corresponding to the covariance function need to be initiated. They are chosen based on prior knowledge about the system. Length-scales λ_{γ_1} and λ_{γ_2} describes how smooth the objective function is expected to be. Scaling factor λ_f determines the variation of the measurement outputs from their mean. The parameters σ_f represents how much measurement noise is expected to be in the measurement output.

The optimal hyperparameters are the ones that best explain the measurements already done. So When measurements are added the hyperparameters need to be tuned. This can be done by maximising the *log marginal likelihood*. Rasmussen & Williams (2006)

An example of a one dimensional GP is shown in figure 3 to illustrate the method. The objective function is represented by the black curve where the measurement points are sampled from. The posterior mean function of the GP is the blue line. The posterior variance is depicted by the light ans dark blue areas, which corresponds to the 68% and 95% confidence bounds. Notice that, where more samples are taken, the confidence bounds are smaller.

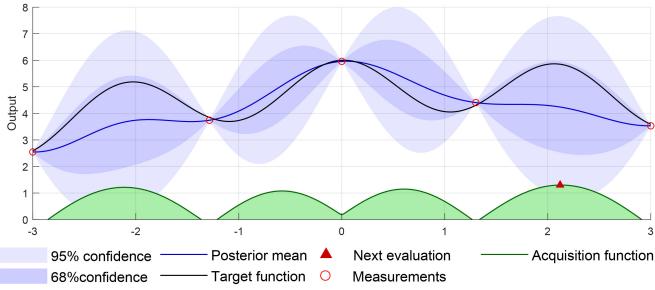


Figure 3: A one-dimensional Gaussian process and it's acquisition function. The data used is random, for visualisation.

Bayesian Optimisation

In BO the posterior distribution is used to define an *acquisition function* (AF). The yaw angles maximising the AF are chosen as the next measurement point. The Upper Confidence Bound (UCB), the Probability of Improvement (PI) and Expected Improvement (EI) acquisition function are studied in this research. They have been proposed by Cox & John (1992), Jones (2001) and Jonas Mockus & Zilinskas (1978) respectively. Every acquisition function has its own parameter which can be varied. High values of the parameter will give more exploring measurement points whereas lower values will result in more exploitation. A trade-off between exploration and exploitation should be made. Initially it is important to do some exploring. However, after enough exploring has been done a maximum should be found and it should be exploited as much as possible. The behaviour over time of different AF and their corresponding parameters differ. When the true optimum of the objective function P^* is known, the behaviour of an AF can be analysed by calculating the *cumulative regret*. The cumulative regret is defined as the sum of the errors between the true optimum and a recommendation on the maximum \hat{P}^* which corresponds to the maximum value of the Gaussian mean.

$$\text{regret} = \sum_{i=1}^n (P^* - P(\gamma_i)) \quad (5)$$

Where After maximising the AF a measurement for the power output at the chosen measurement point is taken. As final step the new data point γ^{n+1} is added to the historical data set. The algorithm runs for t iterations. After t iterations the algorithm gives a recommendation on the maximum, which is the maximum of the mean of the GP.

Algorithm 1 Bayesian Optimization

```

1: procedure PRIOR(Define prior  $\mu$  and hyperparameters )
2:    $r \leftarrow a \bmod b$ 
3:   for  $i = 1, 2, \dots$  do
4:      $a \leftarrow b$ 
5:      $b \leftarrow r$ 
6:      $r \leftarrow a \bmod b$ 
7:   return
```

2.1.2 Greedy control and grid search

There are three benchmark experiments in this research: greedy control, grid search and model-based feed-forward control. The most widely used control approach in real-life wind farms is greedy control: the rotor yaw is aligned with the wind direction. This only shows a single point of the target function.

In order to get a total picture of the target function, a grid search (GS) is executed. The GS consists of a sweep of the input space with a predetermined step size for both turbine 1 yaw γ_1 and turbine 2 yaw γ_2 . The GS will be able to find the global optimum of the target function with a resolution determined by the step size. The GS provides an overview of the target function to give insight into the evaluations that the BO algorithm is making and a metric to evaluate if BO finds the global optimum.

2.1.3 Model-based feed-forward control

The final benchmark setting is an open-loop model-based control approach, based on the parametric low-fidelity FLOW Redirection and Induction in Steady-state (FLORIS) model. The parameters of the test-bench are fed into FLORIS, in which an optimisation algorithm tests a large number of possible control settings on the simplified model, in order to iteratively find the optimal settings for the yaw angles. These angles are then implemented into either the wind tunnel or the high-fidelity model SOWFA. P. Gebraad et al. (2016) have shown that optimisation using FLORIS is accurate enough to yield a significant power increase. It is expected that the optimum found by FLORIS is slightly off the real optimum because of simplifications in the model.

2.2 Test-benches

2.2.1 The scaled wind farm setup



Figure 4: The wind tunnel and scaled wind farm setup.

For this research, access is available to a SMART BLADE Flow Visualisation (FlowVis) Educational Wind tunnel (EdWin), containing 3 scaled wind turbines (see figure 4). The wind speed averaged over the flow field can

be set from 0 to $(7.1 \pm 0.4) \text{ m/s}$. The turbines are positioned at a horizontal distance d of 50cm from each other and the rotor diameters D are 15cm. The output signals of each individual turbine are the rotor angular velocity (Ω) and current (I). To control the set-up, commands can be given and data is sent from and to Matlab. Rotor yaws $\bar{\gamma}$ are used as the input for the wind farm model.

To control the generator torque, a conventional control technique as described in Pao & Johnson (2011) is chosen. It is based on maximising individual power production by adjusting generator torque T , related with a constant to generator current I , using a feedback loop with rotational velocity Ω as input. More information about torque control can be found in Appendix B.2.3.

Using the torque constant $K_T = 3.52 * 10^{-3} \text{ NmA}^{-1}$ provided by the manufacturer, the individual mechanical power can be found using the following relation:

$$P_{\text{mech}} = K_T \cdot I \cdot \Omega \quad (6)$$

The total power output is found by summing up the individual power outputs. A more detailed description of the scaled wind farm can be found in Appendix B. Figure 5 shows a schematic overview of the control setup.

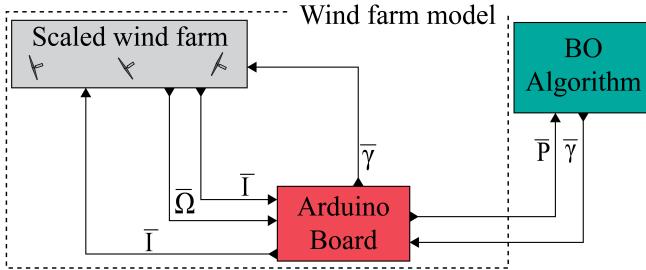


Figure 5: Schematic view of the control structure. The used variables are the current I , angular velocity Ω , yaw angle γ and power P , noted as vectors because they contain the variables of each individual turbine. The wind farm model inside the dotted line can be considered equivalent to the FLORIS and SOWFA models in terms of inputs and outputs.

While this set-up allows for showing the potential of the Bayesian Optimisation approach on a physical system, it also poses limitations. The modified version of the wind tunnel has used a large part of the space of the settling chamber as space for more realistic dimensions of the model wind farm. This causes the flow to be less uniform and more turbulent than in the original design of the Ed-Win. More information about the wind field and turbine characteristics can be found in Appendix B.1.1 and B.2.4. To gain insight into the scalability of the Bayesian Optimisation algorithm, other test-benches have to be used.

2.2.2 FLORIS

Given a yaw angle input and a small number of parameters, FLORIS can compute a wind farm power output. Iterations take approximately 1 second. FLORIS proves to be a useful test-bench for checking alterations to the Bayesian Optimisation control algorithm. However, because of its low fidelity, results from FLORIS will not give conclusive insight into the scalability of the Bayesian Optimisation algorithm, so they will not be discussed in the 'Measurements and results' chapter.

2.2.3 SOWFA

To determine the scalability of the Bayesian Optimisation algorithm to real-life wind farms, it is applied to the Simulator fOr Wind Farm Applications (SOWFA), developed by the National Renewable Energy Laboratory (NREL). SOWFA data has been validated against field tests, and Doekemeijer et al. (2018) shows that SOWFA provides accurate flow data at a fraction of the cost of field tests. In this study SOWFA also simulates a three-turbine wind farm, but with real-life dimensions. It consists of three NREL 5MW turbines with the following parameters: rotor diameter $D = 126.4\text{m}$ and distance between turbines $d = 362\text{m}$. SOWFA takes more physical effects into account than FLORIS, and is therefore computationally more expensive. With a domain of $2\text{km} \times 1\text{km} \times 650\text{m}$, a mesh size of $10\text{m} \times 10\text{m} \times 10\text{m}$ and $\delta t = 0,50\text{s}$, solved using the Actuator Disk Model, it takes around 36 hours on 80 cores to compute 20.000 simulation seconds of the three-turbine case for this research, allowing for around 50 iterations of the Bayesian Optimisation algorithm,

3 Measurements and results

3.1 Wind tunnel

3.1.1 Greedy control and grid search

All experiments in the wind tunnel are run for wind degrees $\psi = [-12, 12]$ with a step size of $\psi_{\text{step}} = 4^\circ$ and wind speed setting 1 and 2 and their results can be found in APPENDIX E. A graph comparing the results of the Bayesian Optimisation algorithm with the benchmark experiments is shown in 3.1.7. As a benchmark, an experiment is performed to evaluate the total power production of the wind farm with greedy control settings, $\gamma_{1,2,3} = 0$. The results can be found in APPENDIX E.X.

In order to get a total picture of the power production as a function of varying γ_1 and γ_2 , a grid search is run. An input domain of $\gamma_{1,2,\text{range}} = [-30, 30]$ with a resolution of $\gamma_{\text{step}} = 3^\circ$ is chosen, resulting in a total experiment time of [59 MINUTES?] per GS. The determination of the resolution size can be found in Appendix B.2.5. The results can be found in APPENDIX E.X..

3.1.2 Model-based feed-forward control

Using the scaled wind farm dimensions and the averaged wind speeds found in 3.1.1, the recommended yaw settings for $\gamma_{1,2}$ are computed in FLORIS for wind degrees $\psi = [-12, 12]$ with a step size of $\psi_{step} = 4^\circ$ and wind speed setting 1 and 2. They are then evaluated in the wind tunnel. The computed yaw angles and the found powers can be found in APPENDIX E.X.

3.1.3 Bayesian optimisation in the scaled wind farm

The Bayesian Optimisation algorithm as described in 2.1.1 is used to control the wind tunnel set-up for wind angle $\psi = 0$ and wind speed setting 2. GRAPH X shows the mean and covariance after 30 iterations and GRAPH Y shows the acquisition function after 30 iterations.

GRAPH X

GRAPH Y

STUK TEXT OVER GRAFIEK

The Bayesian Optimisation algorithm is used to control the wind farm for wind degrees $\psi = [-12, 12]$ with a step size of $\psi_{step} = 4^\circ$ and wind speed setting 1 and 2. The algorithm is run for 100 iterations. At the 50th, 75th and 100th iteration, the yaw angle recommendation $\gamma_{1,2,3}$ and corresponding power output is recorded. The results can be found in APPENDIX E.

The results of all experiments with different control strategies are plotted in GRAPH Z. The graph shows the gain of all strategies with respect to greedy control settings ($\gamma_{1,2,3} = 0$).

BAR GRAPH. X-AXIS WIND ANGLE. Y-AXIS GAIN.
PLOT OF OPTIMUM BY GRID SEARCH, OPTIMUM BY FLORIS, 50TH 75TH AND 100TH ITERATION BY BO

3.2 SOWFA measurements

As a benchmark, an experiment is performed to evaluate the total power production of the wind farm with greedy control settings, $\gamma_{1,2,3} = 0$, for wind degrees $\psi = [0, 8]$ with a step size of $\psi_{step} = 4^\circ$, a wind speed of $V = 8m/s$ and turbulence intensity of $TI = 0.05$. The results can be found in APPENDIX E.

Using the dimensions as described in 2.2.3, a wind speed of $V = 8m/s$ and turbulence intensity of $TI = 0.05$, the recommended yaw settings for $\gamma_{1,2}$ are computed in FLORIS for wind degrees $\psi = [-0, 8]$ with a step size of $\psi_{step} = 4^\circ$. They are then evaluated in SOWFA. The computed yaw angles and the found powers can be found in APPENDIX E. The Bayesian Optimisation algorithm is used to control the wind farm for wind degrees $\psi = [0, 8]$ with a step size of $\psi_{step} = 4^\circ$, a wind speed of $V = 8m/s$ and turbulence intensity of $TI = 0.05$. The algorithm is run for 100 iterations. At the 50th, 75th and 100th iteration, the yaw angle

recommendation $\gamma_{1,2,3}$ and corresponding power output is recorded. The results can be found in APPENDIX E. The results of all experiments with different control strategies are plotted in GRAPH Z. The graph shows the gain of all strategies with respect to greedy control settings ($\gamma_{1,2,3} = 0$).

BAR GRAPH. X-AXIS WIND ANGLE. Y-AXIS GAIN.
PLOT OF OPTIMUM BY FLORIS, 50TH 75TH AND 100TH ITERATION BY BO

4 Discussion

Results wind tunnel

It is found that the potential improvement by yaw steering is between ... per cent. Yaw steering effectiveness varies with wind angle and wind speed. The optimum found by FLORIS is ... per cent off. It is expected that this is because of unknown turbulence intensity and the gradient in wind speed in the wind tunnel. The Bayesian Optimisation algorithm converges to an optimum that is ... better than the power output of greedy control and ... close to the optimum found by the Grid Search. It takes ... iterations to come close. This shows one of the advantages of the closed-loop model-free approach over the open-loop model-based approach. This shows that the BO algorithm can be applied to a physical system to find an optimum, but does not show the scalability.

Results SOWFA

It is found that the power improvement by yaw steering applying the open-loop model-based approach is ... per cent. The Bayesian Optimisation algorithm converges to an optimum that is ... better than the power output of greedy control and ... close/better to the optimum found by the open-loop model-based approach. It takes ... iterations to come close.

4.1 Recommendations

Although the Bayesian Optimisation algorithm is able to increase the power output of a wind farm, there are several areas where it can be improved. Research has shown that it can be beneficial to change the optimisation strategy when close to the optimum. McLeod et al. (2018) suggest to select between multiple acquisition functions and traditional local optimisation at each step, while Park & Law (2015) suggest to combine BO with trust region optimisation algorithms. Currently, the Bayesian Optimisation algorithm keeps on running until it is manually stopped or reaches the maximum number of iterations, even when it has found an optimum. This could result in unnecessary iterations. Chapter 6 of Bijl (2018) proposes a way to approximate the probability density function of the expected maximum of a Gaussian Process. This probability density function can be used as a metric for how certain

the algorithm is of being close to the expected maximum. This might be used as a stopping criterion. Alternatively, McLeod et al. (2018) suggest a different stopping criterion. Apart from improving the Bayesian Optimisation algorithm, several additions have to be made before it can be used in field tests. Firstly, it needs to take into account time-varying wind conditions. In Bogunovic et al. (2016), it is suggested to forget old measurements in a smooth fashion. When assuming that the optimal yaw angles are the same for wind speed, wind angle and turbulence intensity, this could result in sub-optimal operation: the algorithm is forgetting perfectly good settings. Using wind speed, wind angle and turbulence intensity as extra inputs, a look-up table can be created. A hybrid of Bogunovic et al. (2016) and the look-up table might be used to control a wind farm in real time. Secondly, it needs to take into account loads. Currently, wind farm operators are hesitant about using yaw steering control approaches, since it can influence the loads. The research of Damiani et al. (2018) on yaw misalignment effects on loads could be a starting point for integrating loads in the optimisation strategy. Finally, the algorithm needs to be adapted so that it can be applied on wind farms with more than three turbines. Chapter 5 of Bijl (2018) gives suggestions on including multiple inputs in a Bayesian Optimisation algorithm.

It is suggested that these alterations to the algorithm are tested on a parametric simulation like FLORIS and then validated on a high-fidelity simulation like SOWFA. While the scaled wind tunnel is useful for showing the application of BO on a physical system and for visualisation purposes, it seems to have a smaller resemblance to real-life wind farms than FLORIS, with a longer iteration time. SOWFA proved useful as a cheaper surrogate for field tests.

5 Conclusion

In this study, an answer was searched for the question: “Is it possible to optimise the total power production of a wind farm, by applying yaw-based wake steering, using Gaussian Processes and Bayesian Optimisation?” In experiments on a scaled wind farm in a wind tunnel it was shown that the algorithm increased the power by ... to ... per cent with respect to greedy control and can come ... close to the global optimum within ... iterations. In the high-fidelity simulation software SOWFA it was shown that the algorithm increased the power by ... to ... per cent with respect to greedy control and by ... per cent with respect to open-loop model-based control.

This is certainly an improvement to the currently widely used greedy control approach. BO also has advantages over other state-of-the-art control approaches. It is shown that BO can approach the optimum better than an open-loop model-based controller, because of the error intro-

duced by inaccuracies in the model.

Comparison to closed-loop model-free control approaches.

No comparisons have been made to closed-loop model-based control approaches.

These experiments show that a Bayesian Optimisation algorithm can be used to optimise the total power production of a wind farm. This is the first time Bayesian Optimisation is applied to a high-fidelity simulation like SOWFA, an important step before BO can be tested on real-life wind farms.

Suggestions are made to make the algorithm more efficient, adapt to time-varying conditions, to take loads into account and to scale it up to additional wind turbines, which need to be researched and applied before the algorithm is ready to be used in field tests.

6 Acknowledgements

The authors would like to thank prof.dr.ir. J.W. van Wingerden, dr. ir. M. Kok, ir. B.M. Doekemeijer and ir. D.C. van der Hoek for their supervision and for providing us the time and means to exploit the wind tunnel, FLORIS and SOWFA. The authors are also grateful to ing. W.J.M. van Geest, for providing us with the necessary technical assistance, and dr. R. Delfos for using his anemometer.

This research was supported by Delft Center for Systems and Control.

(Klopt dat? Moet die laatste zin er in? of TU Delft?)

References

- Added wind power in 2017. (2018). <https://wwindea.org/blog/2018/02/12/2017-statistics/>.
- Barthelmie, R., Frandsen, S., Hansen, K., Schepers, J., Rados, K., Schlez, W., ... Neckelmann, S. (2009). Modelling the impact of wakes on power output at nysted and horns rev. In *European wind energy conference*.
- Bastankhah, M., & Porté-Agel, F. (2017). A new miniature wind turbine for wind tunnel experiments. part i: Design and performance. *Energies*, 10(7).
- Bijl, H. (2018). *Lgg and gaussian process techniques: For fixed-structure wind turbine control* (Unpublished doctoral dissertation). Delft University of Technology.
- Bogunovic, I., Scarlett, J., & Cevher, V. (2016). Time-varying gaussian process bandit optimization. In *Artificial intelligence and statistics*.

- Cox, D. D., & John, S. (1992). A statistical method for global optimization. In *[proceedings] 1992 ieee international conference on systems, man, and cybernetics* (pp. 1241–1246).
- Damiani, R., Dana, S., Annoni, J., Fleming, P., Roadman, J., Dam, J. v., & Dykes, K. (2018). Assessment of wind turbine component loads under yaw-offset conditions. *Wind Energy Science*, 3(1).
- Doekemeijer, B. M., Boersma, S., Fleming, & van Wingerden, J.-W. (2019). Wind power modelling. *The Institution of Engineering and Technology (IET)*.
- Doekemeijer, B. M., Boersma, S., Pao, L. Y., Knudsen, T., & van Wingerden, J.-W. (2018). Online model calibration for a simplified les model in pursuit of real-time closed-loop wind farm control. *Wind Energy Science Discussions, 2018*.
- Gebraad, P., Teeuwisse, F., Van Wingerden, J., Fleming, P. A., Ruben, S., Marden, J., & Pao, L. (2016). Wind plant power optimization through yaw control using a parametric model for wake effects—a cfd simulation study. *Wind Energy*, 19(1).
- Gebraad, P., & Van Wingerden, J. (2015). Maximum power-point tracking control for wind farms. *Wind Energy*, 18(3).
- Gebraad, P. M. O. (2014). *Data-driven wind plant control*. Delft University of Technology.
- Heinen, Pauwels, Reniers, Smit, & Zaky. (2018). Wake steering demonstration model. *BEP TU Delft*.
- Jonas Mockus, V. T., & Zilinskas, A. (1978). The application of bayesian methods for seeking the extremum. *Elsevier, Amsterdam*.
- Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4).
- McLeod, M., Osborne, M. A., & Roberts, S. J. (2018). Optimization, fast and slow: optimally switching between local and bayesian optimization. *arXiv preprint arXiv:1805.08610*.
- Pao, L. Y., & Johnson, K. E. (2011). Control of wind turbines. *IEEE Control Systems*, 31(2).
- Park, J., & Law, K. H. (2015). A bayesian optimization approach for wind farm power maximization. In *Smart sensor phenomena, technology, networks, and systems integration 2015* (Vol. 9436).
- Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. MIT Press.
- Stein, M. L. (1999). *Interpolation of spatial data*. Springer Science & Business Media.

Appendix WORK IN PROGRESS

A GP and BO mathematics

A.1 Gaussian Processes mathematics

To start, there is an input interval on which the desired function will be evaluated. This interval needs to be defined as sample points, a vector of n input data points on this interval will be made, this vector we call

$$xs$$

Next, the initial i input points on which to evaluate will be made. This vector we call

$$xm$$

These vectors are placed behind one another to create the vector

$$x = [xm; xs]$$

The evaluations of xm will also be made into the vector

$$fm$$

from the vector x, the initial Gaussian process will be made. First a difference matrix is created, this matrix is

$$Diff = [X - X']$$

Where X is an n+i by n+i matrix with the vector x repeated n times and X' is an n+i by n+i matrix with the vector x' repeated n times. Next, the covariance matrix is produced by applying the covariance function, k, element-wise on the difference matrix.

K=k(Diff). This covariance function or kernel, contains some (mostly 3) hyper parameters. The most common covariance function is the squared exponential function:

function The hyper parameters as well as the covariance function itself can be adjusted for better results, but more on that later. The K matrix is then divided in four sub-matrices,

$$K = \begin{bmatrix} K_{mm} & K_{ms} \\ K_{sm} & K_{ss} \end{bmatrix}$$

Where K_{mm} , K_{ms} , K_{sm} , K_{ss} are i by i, i by n, n by i and n by n respectively. The following matrix operations will be executed to create the posterior mean and standard deviation:

$$\begin{aligned} mpost &= K_{sm}/(K_{mm}) \times fm' \\ Spost &= K_{ss} - K_{sm}/(K_{mm}) \times K_{ms} \\ spost &= \sqrt{\text{trace}(Spost)} \end{aligned}$$

Then, the next evaluation will be chosen, making the xm and fm vector one digit longer and the posterior mean and variance will be made again.

A.2 Covariance functions and hyper parameters

As mentioned earlier, there are many covariance functions to be used, and they all fit different objectives. The basic function is that of the squared exponential:

$$K(X - X') = s_{fm} + l_f^2 e^{-\frac{(X-X')^2}{2*l_x^2}}$$

It has s_{fm} , l_f and l_x as hyper parameters

A.3 Bayesian optimisation mathematics

In BO, the next point to evaluate for the GP, is not randomly chosen, but calculated. An acquisition function is a function made from the posterior mean and variance outputs of the GP, along with optional hyper parameters. The maximum of the acquisition function then is the point at which the function will be evaluated next. Bayesian optimisation can be explorative or exploitative. Exploitative means that the acquisition will generate maximums near the highest evaluated points so far, where an explorative optimisation will generate maximums at the points with the highest uncertainty.

A.4 list of used acquisition functions

The mean is denoted as μ (mu) and the variance as σ (sigma). The various acquisition functions are built out of these two variables and sometimes an additional hyper parameter κ (kappa). The three most common acquisition functions are the Upper Confidence Bound (UCB):

$$UCB = \mu + \kappa * \sigma$$

the Expected Improvement:

$$EI = (f^* - \mu) * \Phi\left(\frac{f^* - \mu}{\sigma}\right) + \sigma * \phi\left(\frac{f^* - \mu}{\sigma}\right)$$

(Where f^* is the max of the mean and Φ and ϕ are the CDF and PDF standard normal distributions respectively) and the Prediction of Improvement:

$$PI = \Phi\left(\frac{f^* - \mu}{\sigma}\right)$$

B Detailed description of the scaled wind farm

B.1 The wind tunnel

The wind tunnel used is a customised model FlowVis (flow visualisation) from the company Smart Blade. In figure 6 there is a picture of the wind tunnel, along with a legend. The Wind tunnel allows for manually controlling the wind speed. The fog installation that was manufactured by FlowVis does not produce enough fog to fill the scaled-up tunnel.



Figure 6: The FlowVis wind tunnel

More information about the wind tunnel mechanism can be found on the manufacturer's website <https://www.smart-blade.com/windtunnel-des/>.

B.1.1 Wind field

To gain insight in the reliability of measurements in the tunnel, the wind speed across the inlet is measured using a Testo 445 anemometer. Measurements are taken in an evenly spaced 4x4 grid over the inlet surface for two different settings of the wind speed. The results can be found in APPENDIX B. The mean wind speed found for setting 1 is 5.1m/s . The mean wind speed found for setting 2 is 7.1m/s .

B.2 The wind farm

B.2.1 The wind farm set-up

The turbines

To make the wind tunnel into a scaled wind farm, several adjustments have been made by Heinen et al. (2018). Instead of an air foil, which is normally the object in a Smart Blade wind tunnel, three small wind turbines have been placed inside. The wind turbines consist of an aluminium tower on top of which a dc motor is placed. On the axis of this motor, the 3D printed turbine blades are attached. More information about the design of the blades can be found in the paper ‘A new miniature Wind Turbine for Wind Tunnel Experiments’ by Bastankhah & Porté-Agel (2017). Figure 7 shows this wind turbine design.

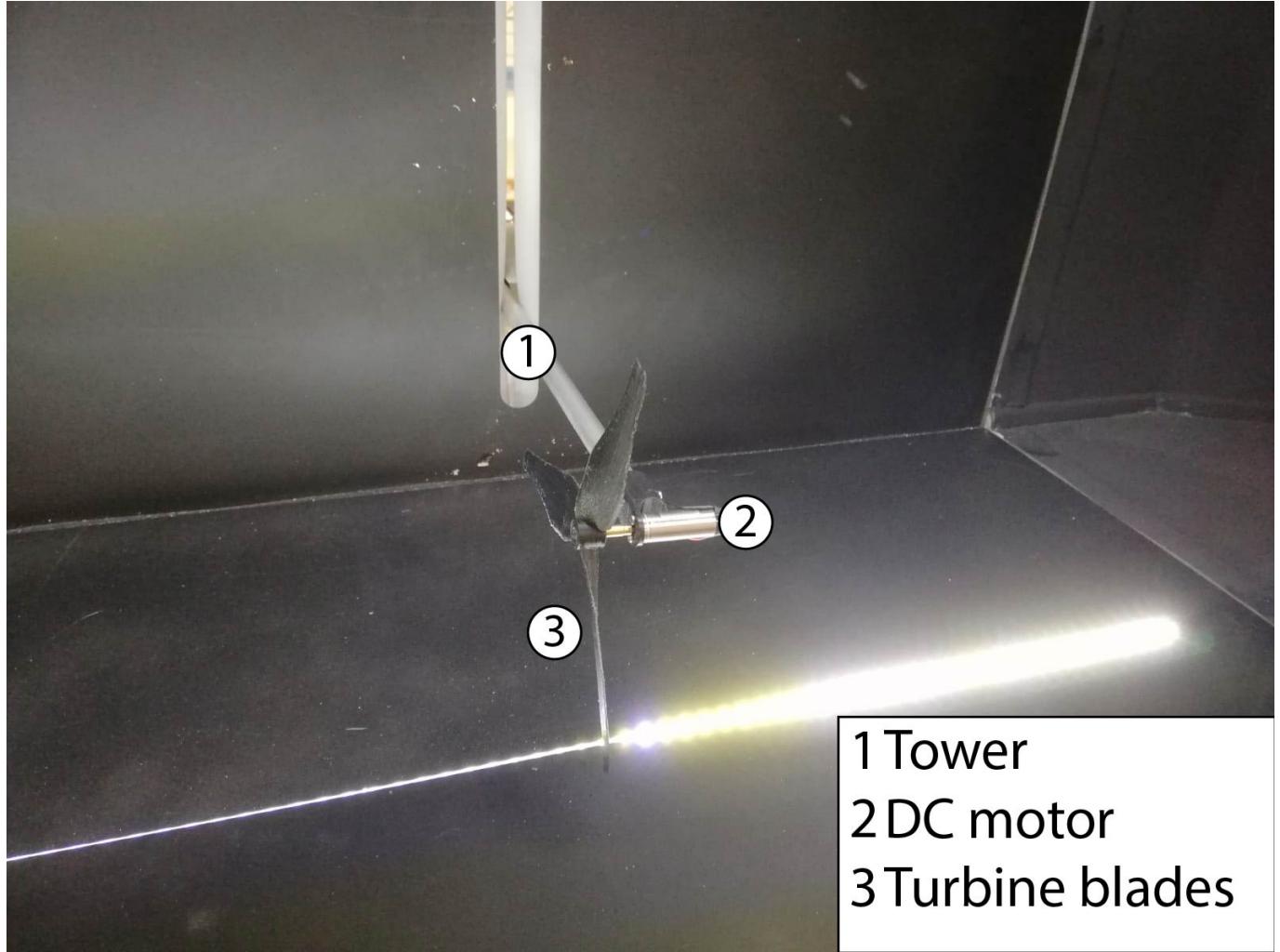


Figure 7: Turbine detail

Translation

To simulate a change in wind direction, the scaled wind turbines allow for vertical translation. This is realised by a translation construction that is fixed to the side of the wind tunnel (figure 3). This consists of two slider rods with linear bearings per wind turbine. A NEMA17 stepper motor controls the translation of the scaled turbines and a TowerPro SG92R micro-servomotor controls the yaw rotation of the turbine. By moving the first turbine up and the second one down or vice versa, a virtual change in wind direction can be made. The distance between the turbines changes from 50 cm to 51 cm, so this change is treated as negligible. This creates a maximum wind angle of 13 degrees.

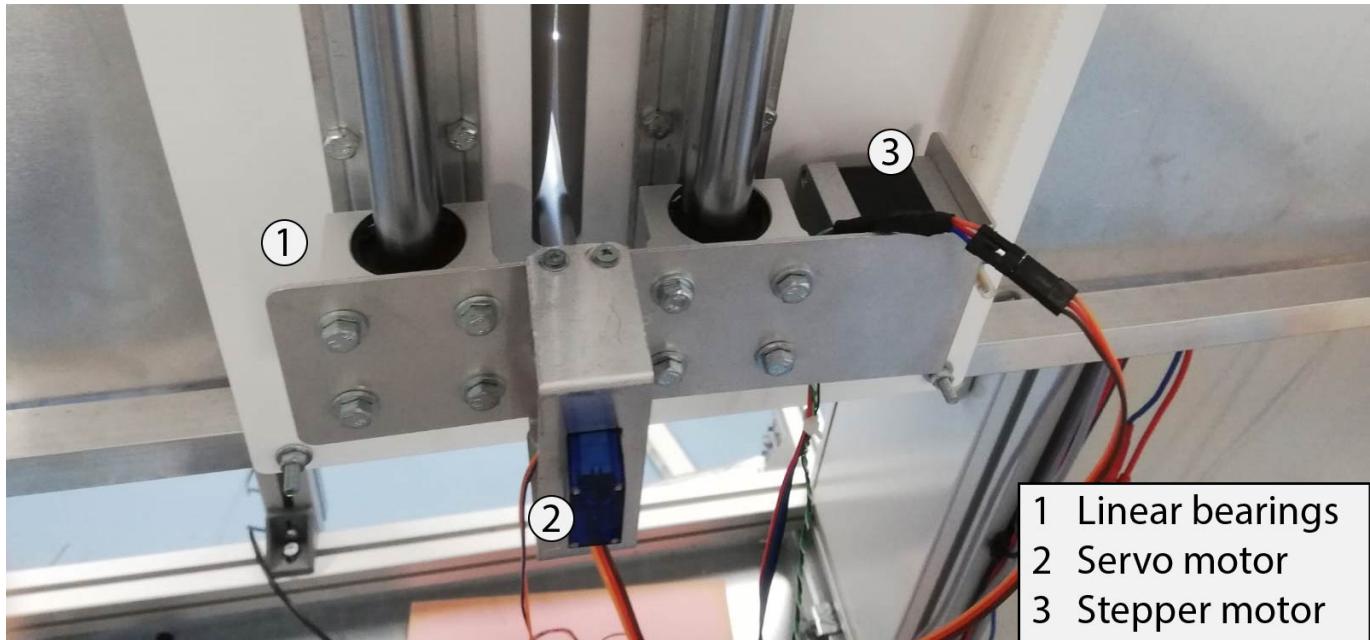


Figure 8: Translation detail

Actuators

Stepper motors NEMA17 stepper motors DRV8825 stepper drivers Half steps: steps of 0.9° Servo motors TowerPro SG92R They seem to have a deviation of approximately 3° .

Sensors

DC motors The DC motors are connected to the Escon drivers through a + and a - cable. The Escon drivers can determine the RPM of the DC motors, by scaling the voltage with the speed constant: 2710 RPM/V. This RPM is only an indication, and cannot be used for exact measurements. Since the motors are equipped with the ENX 10 EASY 1024IMP encoders, the RPM can be determined more exactly, for quantitative measurements and control loop.

DCX 10 L Ø10 mm, 4.5V, Precious Metal Brushes CLL, sintered sleeve bearings ENX 10 EASY 1024IMP encoders

Controllers

Arduino Mega Escon 36/2 for the DC motors DRV8825 for the stepper motors

Arduino board

The following picture illustrates how the connections on the Arduino board are made:

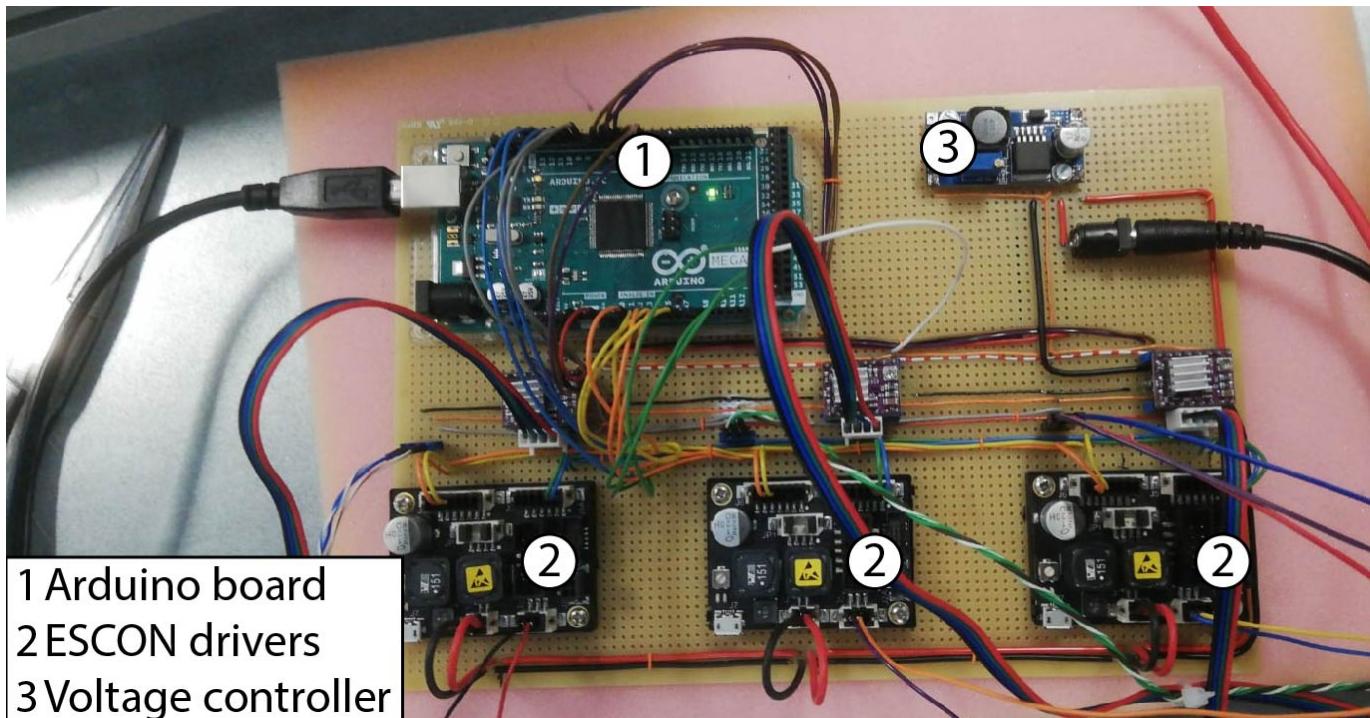


Figure 9: Arduino connections

B.2.2 Settling time and standard deviation

The grid searches in this research require a large amount of iterations so it is desired to keep the measurement time per iteration as low as possible while making sure the output reaches its final value to at least an allowable error. There will be two types of inputs used in this research: setting torque then determining power output to determine the constant value K, and setting the yaw angles 1, 2, and 3 then again determining power output. Both control inputs will be given as step functions, thus the step response of both inputs is determined. The step responses are plotted in figure 10 and 11. The settling time is determined to be 8.7363 seconds for a torque input and 7.0676 seconds for a yaw input.[PLOT GRAPH]

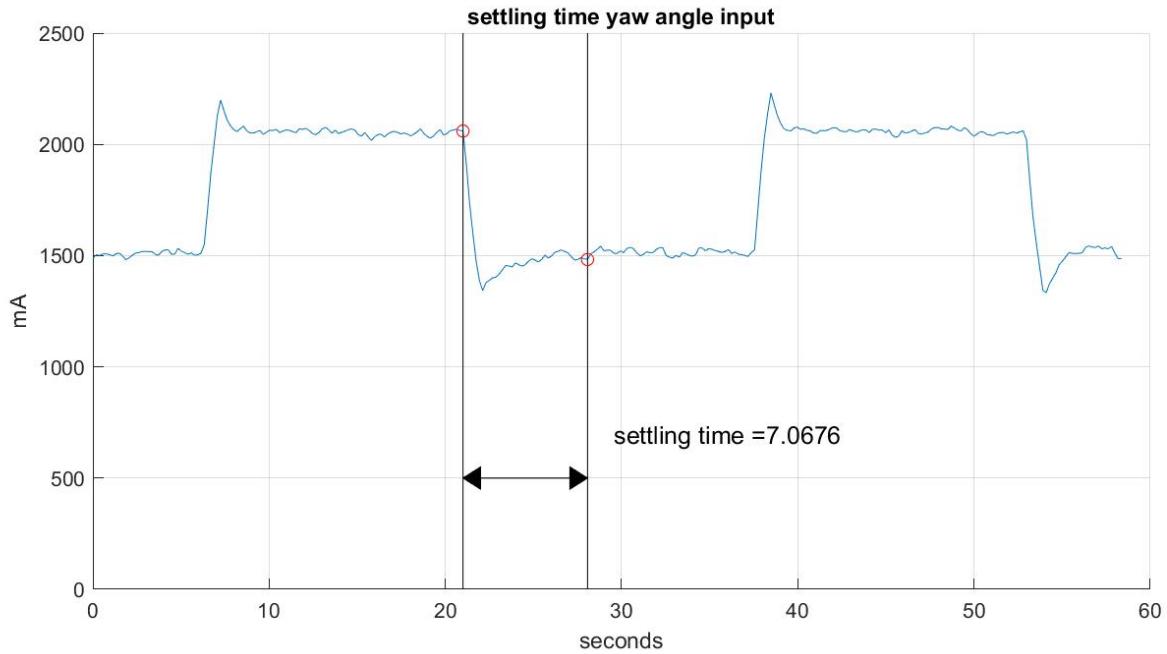


Figure 10: Settling time yaw

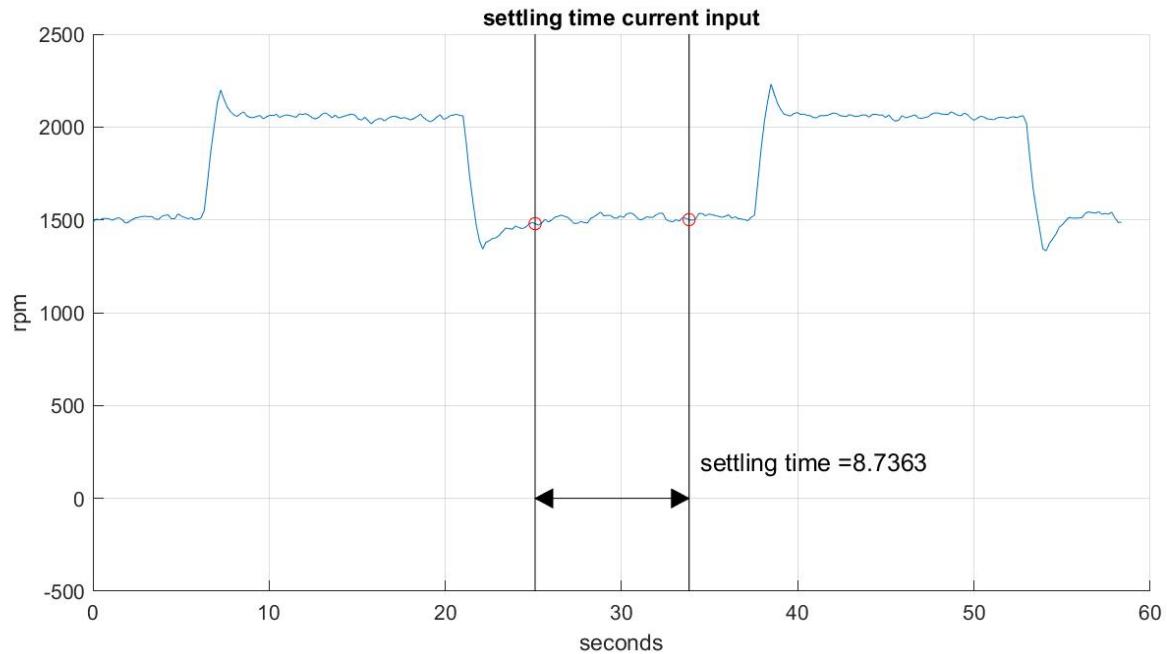


Figure 11: Settling time current

There are two main factors for errors in the power output measurements: fluctuations in wind speed due to turbulence, and inaccuracies in the hardware. To account for these errors, experiment results are averaged on the Arduino over multiple measurements before being sent to Matlab. An experiment is performed to determine the relationship between measurement time and measurement accuracy in order to make a trade-off. The number of measurements, measured power, measurement time and standard deviation can be found in TABLE 1.

| No. of meas's. | Meas. power [W] | Meas. time [s] | Standard deviation [W] |
|----------------|-----------------|----------------|------------------------|
| 10 | 0.2341 | 0.11 | 0.0044 |
| 100 | 0.1997 | 0.27 | 0.0033 |
| 1000 | 0.1959 | 1.89 | 0.0027 |

It is decided to average over 100 measurements per iteration, as this provides a satisfactory low standard deviation combined with reasonable measurement time. This setting is applied to 100 measurements over 30 minutes, resulting in a mean of 0.2011 W with a standard deviation of 0.0035 W.

B.2.3 Torque control

To maximise the individual turbine power output P_{mech} , the torque T of the generators is controlled. The goal is to keep the ratio of the blade tip speed and the wind speed U_w , called the blade tip speed ratio at the optimal value λ^{opt} . A control loop is permanently running on the Arduino, measuring the angular velocities Ω of the rotors and setting the torque of the generators.

The following formula can be found by matching the fundamental relations for mechanical and aerodynamic power, taking into account air density ρ , rotor diameter D , power coefficient C_P and the gearbox ratio G :

$$T^{opt} = \left(\frac{\pi \rho (\frac{D}{2})^5 C_p^{opt}}{2 (\lambda^{opt} G)^3} \right) \omega^2 \quad (7)$$

All the parameters can be summarised in constant k :

$$= k \cdot \omega^2 \quad (8)$$

Torque is related to current with another constant. The total relation used for the feedback loop can be captured by:

$$I^{opt} = K \cdot \omega^2 \quad (9)$$

K can be found by filling in the individual variables, or can be determined empirically. In this research, the latter method is used. A property of the value K is that it is constant for different wind speeds and it will be approximately the same for turbines 1, 2 and 3.

The factor K is found by running a series of measurements, applying generator current and measuring power output, increasing from 0 until the torque resulting from the current matches the torque resulting from the wind acting on the rotor. The optimal current for this wind condition is now known, and value K can be found from:

$$K = \frac{I^{opt}}{\omega^2} \quad (10)$$

K is computed to be $2.8 \cdot 10^{-6} As^2 rad^{-2}$ and the torque feedback loop using this factor K will be used for all of this research.

B.2.4 Power coefficient

The power measurements of the turbines can be used as a metric for the fidelity of the scaled wind farm, by evaluating the turbine power coefficient C_P . According to Bastankhah & Porté-Agel (2017), the scaled turbines are designed to have a C_P between 0.3 and 0.4 for wind speeds between 3 and 8 m/s. Using the equations for mechanical and aerodynamic power, taking into account air density ρ , it can be found that:

$$C_P = \frac{P_{mech}}{\frac{1}{2} \rho \pi (\frac{D}{2})^2 U_w^3} \quad (11)$$

Filling in the mechanical power of turbine 1 with greedy control settings for wind angle $\psi = 0$, air density $\rho = 1.225$, rotor diameter $D = 0.14m$, $V = 5.1m/s$ for wind speed setting 1 and $V = 7.1m/s$ for wind speed setting 2 in the C_P formula:

$$C_P = \frac{P_{mech}}{\frac{1}{2} \rho \pi (\frac{D}{2})^2 V^3} \quad (12)$$

gives $C_P = \dots$ for setting 1 and $C_P = \dots$ for setting 2.

B.2.5 Grid Search

The amount of iterations increases exponentially with the amount of control inputs. The total experiment time T can be calculated using equation 13, where γ_{range} is the range of the yaw angles, γ_{step} is the step size, n is the number of inputs, t_s is the settling time and t_m is the measurement time.

$$T = \left(\frac{\gamma_{range}}{\gamma_{step}} + 1 \right)^n * (t_s + t_m) \quad (13)$$

C Relevant data output

C.1 Wind tunnel data

C.1.1 Feedback loop for torque control

PLOT GRAFIEK MET K-SEARCH

C.2 Greedy control

PLOT GRAFIEK MET OP DE X-AS WINDHOEK EN OP DE Y-AS TOTALE POWER OUTPUT. POWEROUTPUT FOR WINDSETTING 1 EN WINDSETTING 2 KUNNEN IN DEZELFDE GRAFIEK MET LABEL

C.2.1 Grid search

The results can be found in [figures Ex,y,z,..]. Figure [x?] shows the power production of the three turbines, for different yaw angles of the turbine furthest upstream, while the other turbines are kept constant at 0° . The maximum outputs of the grid searches are summarised in figure [reference to figure (BO, GS and Greedy comparison) with staafdiagram x: wind angle, y: power production greedy, power production max. Grid Search]

The grid search is used as a comparison for yaw misalignment by BO.

C.2.2 FLORIS

tabel: Optimal yaw angles computed by floris grafiek: power increase per hoek

C.2.3 Bayesian Optimisation

The Bayesian Optimisation algorithm is used to control the wind farm for wind degrees $\psi = [-12, 12]$ with a step size of $\psi_{step} = 4^\circ$ and wind speed setting 1 and 2. The algorithm is run for 100 iterations. At the 50th, 75th and 100th iteration, the yaw angle recommendation $\gamma_{1,2,3}$ and corresponding power output is recorded.

C.3 SOWFA data

C.4 Greedy control

As a benchmark, an experiment is performed to evaluate the total power production of the wind farm with greedy control settings, $\gamma_{1,2,3} = 0$, for wind degrees $\psi = [0, 8]$ with a step size of $\psi_{step} = 4^\circ$ and wind speed $V = 8m/s$.

C.5 Model-based feed-forward control

Using the dimensions as described in 2.2.3 and a wind speed of $V = 8m/s$, the recommended yaw settings for $\gamma_{1,2}$ are computed in FLORIS for wind degrees $\psi = [-0, 8]$ with a step size of $\psi_{step} = 4^\circ$. They are then evaluated in SOWFA. The computed yaw angles and the found powers can be found in APPENDIX E.