Time-shifted cost function design for more efficient dynamic wind farm flow control

Becker, Marcus; Allaerts, Dries; Van Wingerden, Jan Willem

DOI
10.1109/CCTA60707.2024.10666535

Publication date
2024

Document Version
Final published version

Published in
Proceedings of the IEEE Conference on Control Technology and Applications, CCTA 2024

2024 IEEE Conference on Control Technology and Applications (CCTA)
August 21-23, 2024. Newcastle upon Tyne, UK

Time-shifted cost function design for more efficient dynamic wind farm flow control*

Marcus Becker[1], Dries Allaerts[2] and Jan-Willem van Wingerden[1]

2024 IEEE Conference on Control Technology and Applications (CCTA) | 979-8-3503-7094-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/CCTA60707.2024.10666535

***Abstract*— Dynamic wind farm flow control is the art and science to maximize the energy yield of large wind farms. In this paper we will address the problem of large time delays between control actions of the different turbines in the farm and the delayed impact on the downstream turbines. We propose and show how a time-shifted cost function approach can render the receding horizon optimization problem more efficient and can mitigate the unavoidable turn-pike effect. We further show how the resulting setup can be used to break the optimization problem apart into several smaller optimization tasks to reduce the computational load. We demonstrate that the proposed changes do allow an economic model predictive control strategy to engage into collaborative wind farm control for long term gains, while a more traditional cost function approach leads to greedy turbine behavior. As a result, we take a crucial step towards a mature implementation of dynamic model based wind farm flow control.**

I. Introduction

Wind farms form an integral part in the renewable energy mix of the future. They combine the necessary infrastructure with an allocated location for kinetic wind power extraction. In this context a conflict arises: The more wind turbines are placed in the same area, the more they influence one another. As one turbine extracts the kinetic energy from the wind, it leaves a wake of lower wind speed behind it. This leads to power losses at turbines operating in the wake of an upstream turbine [1].

The goal of wind farm flow control is to mitigate these wake losses by changing the turbine state and by extend the wake shape. One such control strategy is wake steering, where the rotor plane is intentionally misaligned with the main wind direction [2], [3]. As a result, the wake is deflected and can be steered away from downstream turbines. Steady-state control approaches do exist for this strategy, and have been found effective in field experiments [4]. These approaches do neglect the wake dynamics by design, so the question remains if a dynamic model-based control approach can improve upon the results of a steady state control strategy. While dynamic control-oriented wind farm control models have been proposed [5], [6] there are limited to no results available on how these models can outperform existing control algorithms.

*This work is part of the research program "Robust closed-loop wake steering for large densely spaced wind farms" with project number 17512, which is (partly) financed by the Dutch Research Council (NWO).

[1]Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands marcus.becker@tudelft.nl

[2]Department of Flow Physics and Technology, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands.

One central barrier for dynamic wind farm flow control are the time delays between the control actions and the effect on downstream turbines. The delays make it necessary to simulate the wind farm at least until the point at which the upstream control changes arrive at the downstream turbine and often more time is needed to escape the turnpike effect [7], [8]. The turnpike effect arises when optimization parameters have effects which lie beyond the prediction horizon of the simulation. In the case of dynamic wind farm flow control, this usually leads to a greedy control of the upstream turbine: As it needs to sacrifice some of its power to enable the gains of a downstream turbine, it chooses to maximise its own power again at the end of the time horizon. The resulting disadvantages at the downstream turbine lie beyond the time horizon and are not captured by the cost function anymore. In practice this means that a part of the solution found by the optimization problem has to be disregarded. Workarounds do exist: One way to reduce the impact is to fixate the last variables of the action horizon [9], or to attach large costs to moving them. Faced with a similar issue, [10] uses a time shifted signal to identify the impact of a dithering signal in a model-free control approach.

We propose to extend the approach of using a time shifted signal, and to dynamically choose which turbines to consider for the cost function based on the time it takes for control actions to propagate downstream. This way, the cost function is reduced to the impact of the control actions at the upstream turbine synchronised with the impact at the downstream turbine(s).

The main contributions of this paper are threefold:

- We propose a new time-shifted cost function for dynamic wind farm control to mitigate the turnpike effect.
- We propose a novel clustering algorithm to break down the control problem in smaller uncoupled optimization problems.
- We demonstrate the concept with a proof-of-concept simulation study.

This paper is structured as follows: Section II describes the methodology used. In Section III, the main results are presented. Finally, in Section IV the conclusions will be drawn.

II. Methodology

We first describe the general problem setup in Sec. II-A. In Sec. II-B we derive the time delays between the turbines. Using these delays we deduce a way to optimize the turbine behaviour across a given action horizon in Sec. II-C. Sec. II-D comments on the choice of the prediction

and action horizon length and on means to further reduce the computational effort.

A. General setup

The receding horizon problem as we use it consists out of an action horizon of τ_{ah} time steps, and a prediction horizon of τ_{ph} time steps. Fig. 1 (a) depicts this time line. For each time step in the action horizon and for each turbine we consider a optimization variable $\theta_{i,k}$, where $i \in [1, n_T]$ is the index of a turbine (e.g. T_1) and $k \in [1, \tau_{\text{ah}}]$ is a time step. In our problem, these optimization variables directly relate to the turbine yaw angle γ in degrees. Therefore, they are also constrained: The absolute difference between two consecutive parameters may not be larger than the turbine yaw rate limit allows. In Fig. 1 (a) this is indicated by the triangular areas behind the trajectory. Fig. 1 (b) shows a sketch of a possible reaction of the power generated by a two turbine wind farm, T_1 and T_2, to the yaw angle of T_1: T_1 initially sacrifices power by changing its yaw angle. Meanwhile T_2 produces very little power as it is negatively affected by the wake of T_1. As the change propagates through the wake and arrives at T_2, the downstream turbine is less impacted by the wake and the power increases.

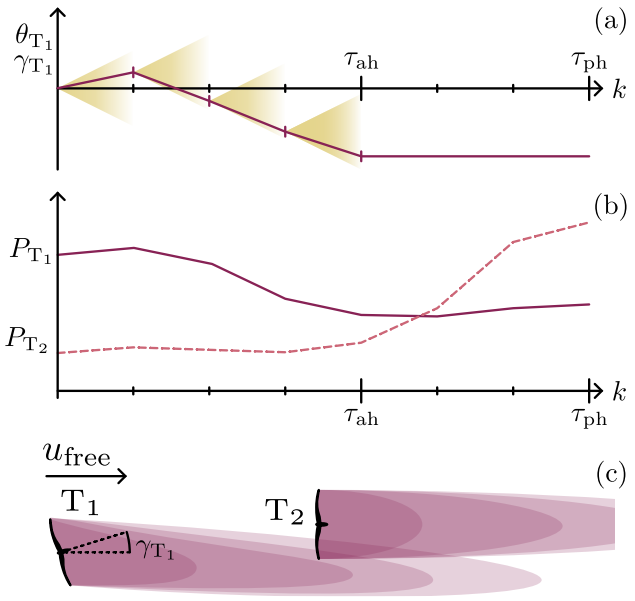

Fig. 1. A receding horizon example where the optimization parameters θ_{T_1} of turbine T1, here representative for the yaw angle γ_{T_1}, are adapted in (a) from the current time step until τ_{ah}. The effect of these changes can be observed over τ_{ph} time steps in (b). The power of T1 suffers due to the yaw angle change, but the power of T2 increases after some delay. The wind turbines and wakes are sketched in (c) .

B. Time delay estimation

In order to determine when a downstream turbine will be affected by the control signal, we need to approximate the advection speed. This is the speed with which the changes in the flow field propagate downstream. Two predominant approaches exist to determine the advection speed: (i) the use of a constant fraction of the free wind speed or (ii) the use of a wake model and the resulting knowledge of the effective wind speed in the wake. The latter method is used for instance by [11] and comes at the computational cost of solving the related model equations. The approach of a constant fraction of the free wind speed has been numerically derived and used by [10] and [12], and has been experimentally studied by [13]. In this work we assume $u_{\text{adv}} = 0.7 \cdot u_{\text{free}}$, following [10] as they are focusing on the same turbine-to-turbine interactions relevant to control. The time offset between two turbines is therefore calculated by

$$t_{i \to j} = \frac{x_{\text{T}_i \to \text{T}_j}}{0.7 \cdot u_{\text{free}}}, \tag{1}$$

where $t_{i \to j}$ describes the time it takes for changes at the rotor plane of T_i to arrive at T_j. The downwind distance from T_i to T_j is denoted by $x_{\text{T}_i \to \text{T}_j}$. Since we consider discrete time steps, we round $t_{i \to j}$ to the closest previous time step:

$$\tau_{i \to j} = \left\lfloor \frac{t_{i \to j}}{\Delta t} \right\rfloor . \tag{2}$$

Equally important to the downstream distance of the turbine is the crossstream distance due to the wake expansion. In this work we disregard turbines with a larger crossstream distance than ± 2 D, where D stands for the turbine diameter.

C. Cost function assembly

Our goal is to move from a single cost function to multiple smaller cost functions: To this end we initially consider all optimization parameters to be independent. Then, if two or more parameters affect the same system output, they become part of a set. For each set of optimization parameters we derive one cost function that combines their impact on the system outputs. This process is described in greater detail in the following paragraphs.

a) Basic formulation: Each optimization parameter $\theta_{i,k}$ belongs to a turbine T_i with $i \in [1, n_T]$ and a time step $k \in [1, \tau_{\text{ah}}]$. The output of the turbine, its power, is denoted as $P_{T_i}(k)$. As time marches, each $\theta_{i,k}$ has an initial effect on $P_{T_i}(k)$, but then delayed with $\tau_{i \to j}$ time steps also on the downstream turbine T_j and thus $P_{T_j}(k + \tau_{i \to j})$. To bring this relation into matrix form we define the vectors $\boldsymbol{P}$ and $\boldsymbol{\theta}$:

$$\boldsymbol{P} = \begin{bmatrix} P_{\text{T}_1}(1) \\ P_{\text{T}_1}(2) \\ \vdots \\ P_{\text{T}_{n_T}}(\tau_{\text{ph}}) \end{bmatrix}, \quad \boldsymbol{\theta} = \begin{bmatrix} \theta_{1,1} \\ \theta_{1,2} \\ \vdots \\ \theta_{n_T, \tau_{\text{ah}}} \end{bmatrix} \tag{3}$$

We then define a matrix $\boldsymbol{T}^*$. The (i, j)-th element of $\boldsymbol{T}^*$ is 1 if the i-th element of $\boldsymbol{\theta}$ affects the j-th element of $\boldsymbol{P}$ and 0 otherwise. Fig. 2(a) depicts an example how this matrix could look like for a four turbine wind farm.

b) Combine coupled optimization parameters: If one column of $\boldsymbol{T}^*$ has two or more 1-entries, multiple optimization parameters affect this output. This means that we cannot consider these optimization parameters as independent and need to optimize them together. We combine the entire rows of the affected parameters by a logical OR operation and

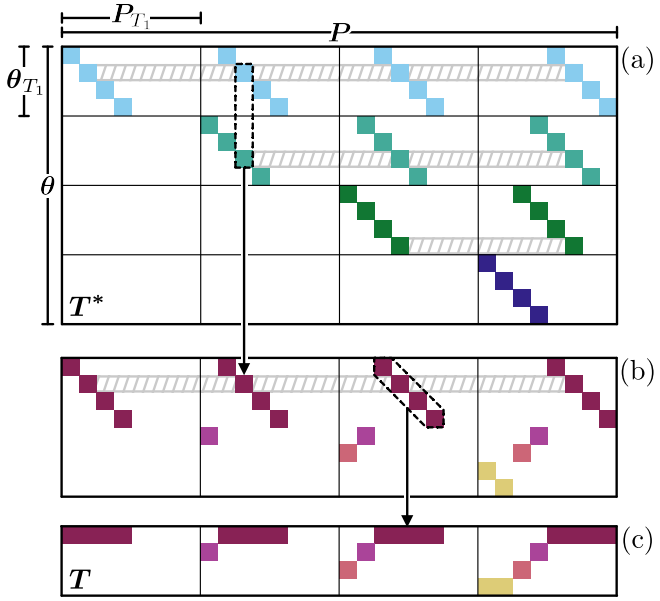

Fig. 2. (a) The relational matrix $\boldsymbol{T}^*$ between the turbine outputs $\boldsymbol{P}$ (columns) and the optimization parameters $\boldsymbol{\theta}$ (rows). The combination of linked optimization parameters leads to (b), indicated by one example. Combining consecutive outputs then leads to $\boldsymbol{T}$, depicted in (c), with one highlighted example.

combine the parameters into a new set $\mathcal{O}_i$. Fig. 2(b) shows which rows of (a) have been combined.

c) Connect consecutive sets: The previous step combined all parameters that need to be connected. The following step is not required but further reduces the number of optimization parameter sets for convenience: If two sets, $\mathcal{O}_i$ and $\mathcal{O}_j$, optimize consecutive-in-time parameter sets they can be combined: $\mathcal{O}_i^* = \mathcal{O}_i \cup \mathcal{O}_j$, where $\mathcal{O}_i^*$ denotes the updated set. In practice this means that consecutive time steps are connected to a single, longer time series. This does change how many optimization parameters are in one set, but it does not change how many turbines are considered in one optimization. The condition for the corresponding operation in the relational matrix is a diagonal similarity of the rows. See Fig. 2(b) where diagonally equal rows share the same color. Their combination leads to the matrix $\boldsymbol{T}$, depicted in Fig. 2(c).

d) Cost function derivation: Each row of $\boldsymbol{T}$ describes a way to map $\boldsymbol{P}$ to a number of smaller optimization problems that each optimize a subset of $\boldsymbol{\theta}$. In our case, this is the relation between the power generated by the turbines and the yaw angle of the turbines. The resulting cost function is the negative sum of the power generated:

$$\min_{\boldsymbol{\theta}_i} J_i(\boldsymbol{\theta}_i) = -\boldsymbol{t}_i \boldsymbol{P}(\boldsymbol{\theta}_i) \tag{4}$$

where i relates to optimization problem i, $\boldsymbol{\theta}_i$ denotes the optimization parameters in $\mathcal{O}_i$ and $\boldsymbol{t}_i$ is the i-th row of $\boldsymbol{T}$. In this particular case the cost of actuation has an intrinsic negative impact on the power, which is why we do not include it again as a negative term in the cost function.

e) Simulation order & clusters: Eq. (4) is a general description of all optimization problems to solve. What it does not offer is an order in which the problems need to be solved. For our system, we have to ensure that the yaw rate is below a given limit. This means that we depend on an initial yaw angle at the start of our optimization problem in order to deduct the consecutive ones. This dependency can be derived from $\boldsymbol{T}$. Fig 3 shows as an example how the optimization problem in the first row relies on previously deducted results for T_2, T_3, and T_4 in row 2. Row 2 further relies on row 3 and row 3 on row 4. Therefore, the solve-order of the optimization problems is $\boldsymbol{\theta}_4 \rightarrow \boldsymbol{\theta}_3 \rightarrow \boldsymbol{\theta}_2 \rightarrow \boldsymbol{\theta}_1$. Two things are worth noting: first, to solve optimization 4 we only need to simulate two time steps. This means that we can reduce the number of optimization parameters, the number of simulated turbines and also the time. Secondly, in a larger wind farms independent solving graphs appear, as not all turbines do influence each other. This offers to reduce the computational cost by splitting the optimization in smaller, independent wind farm clusters.[1]

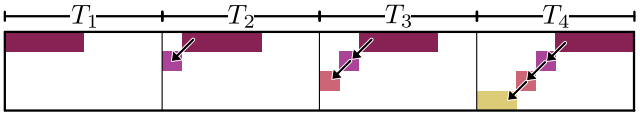

Fig. 3. The arrows indicate which row depends on the results of another row. In this case row 1 depends on 2, which depends on 3, which depends on 4. As a result, we need to solve optimization problem $4 \rightarrow 3 \rightarrow 2 \rightarrow 1$. The blocks indicate which part relates to which turbine. In order to solve row 4, we only need to evaluate T_4.

D. Choosing τ_{ah} and τ_{ph}

One of the central motivations of this work was to avoid the turnpike effect. The time shifted cost function approach introduced in Sec. II-C does a first step towards that goal by only combining turbine outputs that have an effect on the optimization parameters. But a poor choice of the prediction and action horizon will lead to a turnpike effect nevertheless. For instance, if we chose $\tau_{\text{ph}} = \tau_{\text{ah}} = 1$ the effect of the control action will never reach a downstream turbine and the turbines will behave greedy. To avoid this we use the results from Sec. II-B and set a minimum for τ_{ph}:

$$\tau_{\text{ph}} \geq \tau_{\text{ah}} + \max_{i,j \in [1,n_T]} \left\{ \tau_{\text{T}_i \rightarrow \text{T}_j} \right\} . \tag{5}$$

It may be valid to set a $\tau_{\text{ph,max}}$ which violates Eq. (5). The entries of the diagonal block in $\boldsymbol{T}$ are then shifted beyond τ_{ph} and are neglected. The violation could be based on an assumed maximum wake length or to lower the computational cost. Another measure to reduce computational cost would be to not simulate turbines at times when their output is not used, see the zero-columns in Fig. 2(a-d). In that case we alter Eq. (5) to only simulate the turbine j for as long as needed by other turbines:

$$\tau_{\text{ph},j} \geq \tau_{\text{ah}} + \max_{i \in [1,n_T]} \left\{ \tau_{\text{T}_i \rightarrow \text{T}_j} \right\} . \tag{6}$$

[1]The Matlab code to generate $\boldsymbol{T}$ and other derivatives is publicly available with the DOI 10.4121/54cfbca7-243e-4a27-af2d-74cdd91471b2

Note that it might still be necessary to simulate the wakes of the turbines. A third variation would be to define individual action horizons for all turbines, based on a given τ_{ph}:

$$\tau_{\text{ah},i} \leq \tau_{\text{ph}} - \max_{j \in [1, n_T]} \left\{ \tau_{\text{T}_i \to \text{T}_j} \right\} . \tag{7}$$

This can be especially advantageous for systems where changes propagate through the domain, e.g. in the case of a wind speed change or gust. A front row turbine might have a short action horizon, because it can only react to changes in the wind, while downstream turbines can take the wind history of the front row turbine as prediction into account.

III. Results

In this section we describe the simulation case along with the wind farm flow control setup in Sec. III-A. The results of the case are presented and discussed in Sec. III-B.

A. Simulation

We use the *Flow Redirection and Induction Dynamics* (FLORIDyn) framework [14] as eMPC model to evaluate the cost function and to carry out the reference simulation. The FLORIDyn framework has been designed to simulate wind turbine wake dynamics at a low computational cost, and is aimed to be used in a model based closed-loop wind farm flow control strategy. To test the proposed cost function design, we carry out three simulations:

1) a baseline simulation in which the turbines perfectly track the wind direction to maximise their own power generated,
2) a "naive" economic model predictive control (eMPC) setup where FLORIDyn is used to optimize the power generated over a given prediction horizon,
3) an eMPC setup with the same optimization setup but with the proposed cost function structure.

The difference between approach 2) and 3) isolates the effect of the proposed cost function.

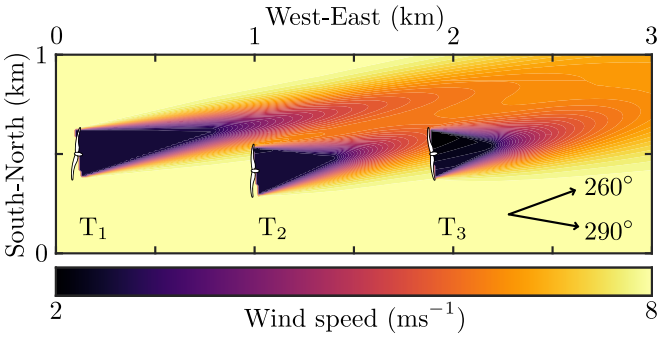

Fig. 4. The three turbine wind farm during the wind direction change. The turbines are placed with a 5 D distance along the West-East axis and the centre turbine is placed −0.5 D on the South-North axis. The initial wind direction is 260 deg, which gradually changes after 1000 s to 290 deg.

We use a three DTU 10-MW reference turbine [15] wind farm, depicted in Fig. 4. The synthetic wind direction signal is set to maintain a steady wind direction of 260 deg for the first 1000 s, followed by a constant change to 290 deg over the following 1000 s. The 290 deg are maintained for the future. This setup does encourage both, individual power maximisation of unwaked turbines and collaborative wind farm flow control to maximise the wind farm performance. In addition, the wind direction change forces changes in the yaw set points of the turbines. The simulation is carried out with a free wind speed of 8 ms^{-1}, an ambient wake model intrinsic turbulence intensity level of 6 %, and with shear, using the power-law and a coefficient of 0.28 [16].

The simulation is discretized in 5 s steps. We arbitrarily choose $\tau_{\text{ah}} = 6 \rightarrow 30$ s for all three turbines, which results in 18 optimization variables. The prediction horizon for both eMPC strategies is calculated based on Eq. (5). Every control iteration, the first two steps of τ_{ah} are applied and the receding horizon is pushed forward by 10 s. The optimization is constrained by a maximum yaw misalignment by ±30 deg and a maximum allowed rate of change of 0.3 deg s^{-1}. The optimization is solved by the interior point algorithm for constrained minimization problems [17]. Both eMPC strategies are initialized with a yaw misalignment of 0 deg and have knowledge of the future development of the wind direction. This assumption is made for simplicity, similar to the work in related publications, see [18], [19].

B. Simulation results and discussion

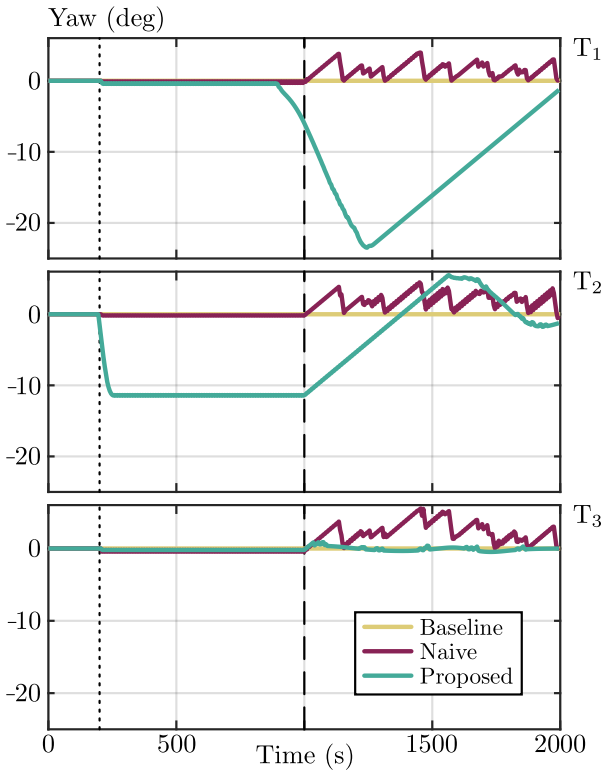

Fig. 5. Yaw angle of T1-T3 during the simulation for the baseline simulation, the "naive" eMPC strategy, and eMPC strategy with the proposed cost function structure. The dotted line indicates the activation of the controller, the dashed line the start of the wind direction change.

The yaw trajectories of all three turbines are depicted in Fig. 5, where the yaw misalignment is calculated as the

difference between the main wind direction and the turbine orientation. The baseline controller tracks the wind direction perfectly and shows a yaw misalignment of 0 deg at all times.

The "naive" eMPC strategy initially uses its degrees of freedom to drive the yaw misalignment to 0 deg. It maintains a yaw misalignment of 0 deg up until the wind direction changes. In the second half of the simulation, the controller engages into yaw steering. The reason is that it tries to avoid power losses in the near future: During the cost function evaluation the controller is aware that the wind direction will continue to change. However, the short action horizon only allows it to move during the initial 6 steps of the prediction horizon. As a result, the controller moves to a future wind direction to be more aligned in the near future.

The eMPC strategy with the proposed cost function does engage into yaw steering and misalignes T2 with the main wind direction. This steers the wake away from the downstream turbine T3 and allows an increase in the total power generated by the wind farm of 0.17 MW. Towards the wind direction transient, the controller preemptively steers T1 to avoid wake interaction at a later stage, which results in large power gains during the initial transient. Turbine T1 remains yawed for a longer time but then recovers its position and reduces its misalignment with the wind direction.

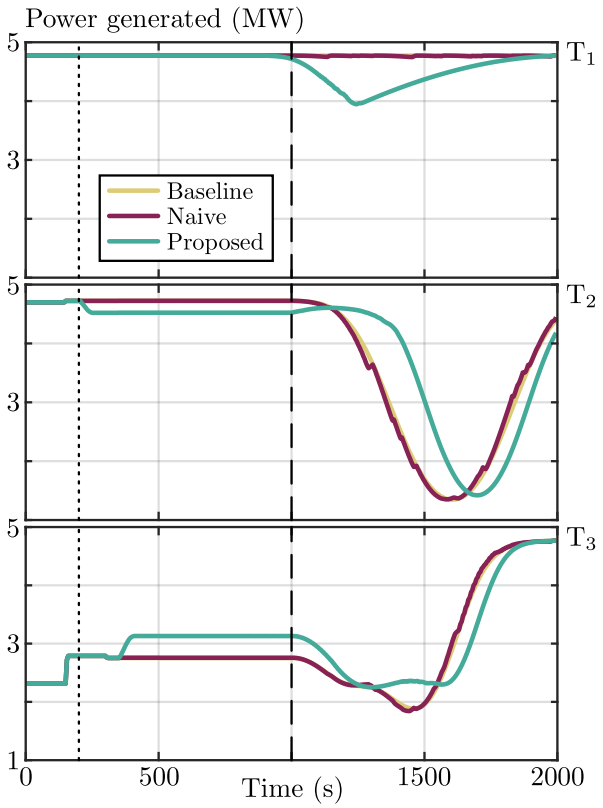

Fig. 6. The power generated by the three turbines in comparison. The dotted line indicates the activation of the controller, the dashed line the start of the wind direction change.

Fig. 6 depicts the generated power by each turbine over time. The baseline and "naive" eMPC strategy show almost identical behaviour, as expected from the yaw angle trajectories. However the data of the eMPC strategy with the proposed cost function does show that the strategy creates gains at the downstream turbines: During the steady state part of the simulation, T2 elevates the power generated by T3, which is the result of the wake steering depicted in Fig. 5. During the initial wind direction change, the yaw steering efforts of T1 show an effect at both T2 and T3. While the power for the two other controllers reduces as the wake interaction increases, the proposed controller is able to offset the wind speed reduction. As a joint effort of T1 and T2, the power of T3 never drops as low as it does in the baseline case or with the "naive" eMPC strategy.

The wind farm efficiency during the simulation is shown in Fig. 7. Once the controller is activated, the proposed eMPC strategy is willing to sacrifice efficiency, and therefore power, to gain efficiency in the long term. The proposed strategy outperforms the baseline as a result, after the effects of the induced yaw angle changes have propagated downstream. As observed in the yaw angle trajectories, the "naive" eMPC strategy does not engage in yaw steering and misses the opportunity to outperform the baseline. During the wind direction transient, the proposed method manages to offset the large drop in farm efficiency, connected to the inevitable wake passing. It is here, where the proposed method shows the largest gains, at its peak 11 % over the baseline. But as the change in wind direction progresses, the wind turbines need to recover their alignment with the main wind direction, something which the baseline has already done. Here, up to 7.6 % wind farm efficiency is sacrificed compared to the baseline.

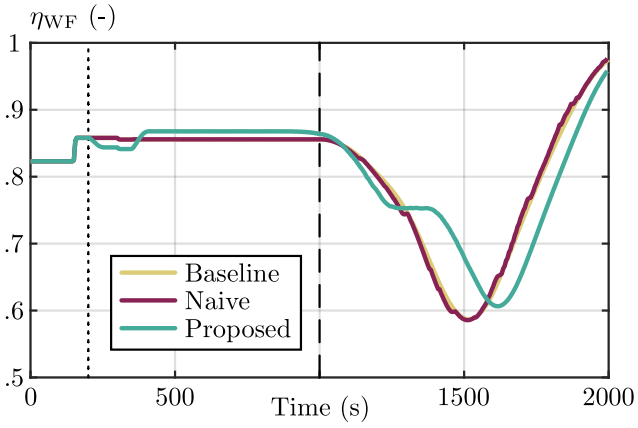

Fig. 7. Wind farm efficiency η_{WF} calculated by total power generated divided by power of the farm without wake effects and misalignment. The dotted line indicates the activation of the controller, the dashed line the start of the wind direction change. The plot contains wind farm start up effects prior to the controller activation.

In summary, the proposed cost function construction allows the eMPC framework to utilise the given degrees of freedom to engage into collaborative wind farm flow control for long term gains. With a more traditional cost function, the optimization result becomes greedy and only tries to minimise short term losses due to yaw misalignment.

IV. Conclusion

In this paper we present a novel method that is able to automatically restructure the optimization of a dynamic eMPC setup for wind farm flow control. It detects sub-optimization problems based on the inherent time delays in the wind farm and the spatial distance of wind turbines. The method further returns which problems can be solved in parallel and which need to be solved sequentially. Based on these factors, new cost functions are constructed which allow an eMPC framework with a short action horizon to optimize for long term wind farm gains in both steady state and dynamic conditions. The alternative, traditional, implementation of the cost function leads to greedy control behaviour, unable to perform collaborative control.

Future work should investigate the robustness of the proposed framework, mainly towards the assumptions around the advection speed of the wake and the flow preview. The solver and an efficient use of the optimization parameters are two additional aspects not discussed in this work, which are essential to a successful eMPC design for wind farm control.

References

[1] J. Meyers, C. Bottasso, K. Dykes, *et al.*, "Wind farm flow control: Prospects and challenges," *Wind Energy Science*, vol. 7, no. 6, pp. 2271–2306, 2022.

[2] M. F. Howland, S. K. Lele, and J. O. Dabiri, "Wind farm power optimization through wake steering," *Proceedings of the National Academy of Sciences*, vol. 116, no. 29, pp. 14 495–14 500, 2019.

[3] P. M. O. Gebraad, F. W. Teeuwisse, J. W. van Wingerden, *et al.*, "Wind plant power optimization through yaw control using a parametric model for wake effects—a CFD simulation study," *Wind Energy*, vol. 19, no. 1, pp. 95–114, 2016.

[4] P. Fleming, J. Annoni, J. J. Shah, *et al.*, "Field test of wake steering at an offshore wind farm," *Wind Energy Science*, vol. 2, no. 1, pp. 229–239, 2017.

[5] M. Becker, B. Ritter, B. Doekemeijer, *et al.*, "The revised FLORIDyn model: Implementation of heterogeneous flow and the Gaussian wake," *Wind Energy Science Discussions*, vol. 2022, pp. 1–25, 2022.

[6] S. Boersma, B. Doekemeijer, M. Vali, J. Meyers, and J.-W. van Wingerden, "A control-oriented dynamic wind farm model: WFSim," *Wind Energy Science*, vol. 3, no. 1, pp. 75–95, 2018.

[7] M. J. van den Broek, D. De Tavernier, B. Sanderse, and J.-W. van Wingerden, "Adjoint optimisation for wind farm flow control with a free-vortex wake model," *Renewable Energy*, vol. 201, pp. 752–765, 2022.

[8] R. Dorfman, *Application of linear programming to the theory of the firm: including an analysis of monopolistic firms by non-linear programming*. Univ of California Press, 2022.

[9] M. J. van den Broek, M. Becker, B. Sanderse, and J.-W. van Wingerden, "Dynamic wind farm flow control using free-vortex wake models," *Wind Energy Science Discussions*, vol. 2023, pp. 1–28, 2023.

[10] U. Ciri, M. A. Rotea, and S. Leonardi, "Model-free control of wind farms: A comparative study between individual and coordinated extremum seeking," *Renewable Energy*, vol. 113, pp. 1033–1045, Dec. 2017.

[11] M. Lejeune, M. Moens, and P. Chatelain, "A Meandering-Capturing Wake Model Coupled to Rotor-Based Flow-Sensing for Operational Wind Farm Flow Prediction," *Frontiers in Energy Research*, vol. 10, p. 884 068, Jul. 2022.

[12] S. J. Andersen, J. N. Sørensen, and R. F. Mikkelsen, "Turbulence and entrainment length scales in large wind farms," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 375, no. 2091, p. 20 160 107, Apr. 2017.

[13] S. Macrì, T. Duc, A. Leroy, N. Girard, and S. Aubrun, "Experimental analysis of time delays in wind turbine wake interactions," *Journal of Physics: Conference Series*, vol. 1618, no. 6, p. 062 058, Sep. 2020.

[14] M. Becker, D. Allaerts, and J. W. van Wingerden, "FLORIDyn - A dynamic and flexible framework for real-time wind farm control," *Journal of Physics: Conference Series*, vol. 2265, no. 3, p. 032 103, May 2022.

[15] C. Bak, F. Zahle, R. Bitsche, *et al.*, "The DTU 10-MW reference wind turbine," in *Danish wind power research 2013*, 2013.

[16] S. Emeis, *Wind Energy Meteorology* (Green Energy and Technology). Cham: Springer International Publishing, 2018.

[17] *MATLAB - Optimization Toolbox Version 9.5*, The MathWorks Inc. Natick, Massachusetts, 2023.

[18] M. J. Van Den Broek, M. Becker, B. Sanderse, and J.-W. Van Wingerden, "Dynamic wind farm flow control using free-vortex wake models," *Wind Energ. Sci. Discuss.*, Sep. 2023.

[19] B. A. M. Sengers, A. Rott, E. Simley, M. Sinner, G. Steinfeld, and M. Kühn, "Increased power gains from wake steering control using preview wind direction information," *Wind Energy Science*, vol. 8, no. 11, pp. 1693–1710, Nov. 2023.