Distributed learning for wind farm optimization with Gaussian processes*

Leif Erik Andersson[1], Eric Christopher Bradford[1] and Lars Imsland[1]

Abstract— **This article investigates optimization of wind farms using a modifier adaptation scheme based on Gaussian processes. In this scheme measurements are used to identify plant-model mismatch using Gaussian process regression, which are then used to find the optimal plant control inputs. However, for systems with many agents and a large control input space, the identification of the input-output map of the plant is challenging. Therefore, the paper proposes a distributed learning approach, in which sub-parts of the plant are identified with individual GP regression models. Afterwards, all of these are used to build a model of the overall plant-model mismatch, which is then used in the optimization. In the wind farm case the sub-parts are the individual turbines. The distributed learning approach clearly outperforms the original central learning approach in numerical illustrations of wind farm test cases.**

I. Introduction

The wind energy production in the last few decades has grown rapidly. The development is driven by high renewable energy targets in the US, Europe, and China, where wind energy will be a significant factor in achieving these goals [1]. On the other hand wind energy has become competitive, with the cost of energy (CoE) significantly reduced, to levels comparable to those of conventional fossil-fueled power plants [2]. The deployment of wind turbines in a farm reduces the overall CoE due to, e.g. reduced deployment costs of turbines and electrical grid, reduced maintenance costs and smaller land use [3]. However, it also decreases the power production compared to the same amount of individual turbines [4], and increases the turbulence intensity with depth in the farm causing larger load fluctuations on the turbines [5].

Currently, turbines in a wind farm are typically operated at their individual optimal operating point, even though it is well established and shown in several studies that this leads to suboptimal performance, e.g. [4], [6]–[9]. Consequently, a wind farm control design coordinating the wind turbines and considering the aerodynamic interactions through wakes between turbines has great potential to increase to overall performance of the farm.

Two common approaches for wake control are axial induction control, e.g. [10]–[15] and wake steering control, e.g. [16]–[20]. In axial induction control the blade pitch and generator torque are controlled operating the upwind turbine sub-optimally causing a smaller wake deficit and increasing e.g. the overall energy production of multiple turbines. In wake steering control the yaw angle of the upwind turbine is changed to deflect the wake and improve the performance of multiple turbines.

Many wind farm control approaches are model-based. Consequently, models of the wind farm that have a low computational cost while describing the necessary dynamics to sufficient accuracy, are required. These engineering wake models often estimate the steady-state situation for a given wind speed and direction assuming the same wind inflow characteristics for the entire farm. Even though engineering models are simple, it was shown that they can estimate the wind speed deficit in the far wake region [21]–[23]. Moreover, new models are developed to further enhance wind farm control [24]–[26]. Nevertheless, a plant-model mismatch still exist since the engineering models can only roughly approximate the complex dynamics of the wake [27]. Hence, purely model-based optimization may be unable to reach plant optimum.

Real-time optimization (RTO) methods adapt the model-based optimization problem using process measurements to drive the plant to optimal performance while guaranteeing constraint satisfaction [28]. The most intuitive RTO strategy is the two-step approach, which consist of repeated parameter estimation and optimization [29], [30]. However, this method cannot guaranty plant optimality upon convergence if the model is structurally mismatched [28]. In contrast, modifier adaptation (MA) is an RTO method that corrects the cost and constraint functions of the optimization problem directly and reaches, under suitable assumptions, plant optimality upon convergence [31]. It requires, however, the estimation of the plant gradients [32].

In this article the recently proposed combination of MA and Gaussian process (GP) regression [32], [33] is applied to wind farm optimization. GP is a probabilistic, non-parametric modelling technique well known in the machine learning community [34]. In the context of MA the modifiers are replaced with GP regression functions that estimate the plant-model mismatch. Bayesian Optimization, which is a purely data driven approach using GP regression models, is used in [20]. In contrast, the approach proposed here exploits prior knowledge given by a model of the system and just corrects the model. Moreover, a distributed learning strategy is proposed, which considerably improves the identification of the input-output map by the GP regression model for large wind farms.

The article is organized as follows: In Section II an overview over the optimization problem, the MA approach and GP

*This work was supported by OPWIND: Operational Control for Wind Power Plants, Norwegian Research Council Grant No. 268044.

[1] Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7491 Trondheim, Norway. {lars.imsland, eric.bradford, leif.e.andersson}@ntnu.no

regression is given, followed by an introduction of the distributed learning MA-GP approach in Section III. The performance of the approach is illustrated in Section IV. The article ends with a conclusion.

II. Problem formulation and preliminaries

The power production optimization problem of the steady-state wind farm can be formulated as:

$$\mathbf{u}_p^* = \arg\min_{\mathbf{u}} P_p(\mathbf{u}) := \sum_i^N p_{p,i}(\mathbf{u}_i) \tag{1a}$$

$$\mathbf{u} \in \mathcal{U}, \quad \mathbf{u} = [u_1^T, u_2^T, \ldots, u_N^T]^T, \tag{1b}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ denote the plant input variables; $\mathbf{u}_i \in \mathbb{R}^{n_{u,i}}$ are the inputs to the turbines of the plant; $P_p : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}$ is the cost function to be minimized, which is the power production in this article; the power production at each agent is also the output $\mathbf{y}_{p,i} \in \mathbb{R}^{n_{y,i}}$ of the plant; and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ is the control domain, e.g. box constraints on the control inputs.
A wind farm consist of N interconnected wind turbines, which contribute to the objective function. The inputs $\mathbf{u}_i$ to each turbine are the thrust coefficient $C_{T,i}$ and the yaw angle γ_i. Inequality constraints are omitted in the formulation, but can be easily included.
The formulation (1) assumes that P_p of $\mathbf{u}$ is known perfectly. However, in any practical application the exact input-output map of the plant is unknown and instead a model of the system is exploited for the optimization:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}} P(\mathbf{u}) := \sum_i^N p_i(\mathbf{u}_i) \tag{2a}$$

$$\mathbf{u} \in \mathcal{U}, \quad \mathbf{u} = [u_1^T, u_2^T, \ldots, u_N^T]^T, \tag{2b}$$

where Φ refers to the quantities in (1) as output of the model. RTO takes advantage of the available measurements to compensate for plant-model mismatch and adapt the model-based problem (2) to reach plant optimality.
MA adds for our optimization problem a first-order correction term to the cost function to match the necessary conditions of optimality upon convergence [31]. Iteratively the following modified optimization problem is solved:

$$\hat{\mathbf{u}}_{k+1}^* = \arg\min_{\mathbf{u}} P(\mathbf{u}) + (\lambda_k^p)^T \mathbf{u} \tag{3a}$$

$$\mathbf{u} \in \mathcal{U}, \tag{3b}$$

where $\hat{\mathbf{u}}_{k+1}^*$ is the optimal solution at iteration $k+1$, and λ_k^ϕ is the first-order modifier for the cost. The correction term is given by:

$$(\lambda_k^p)^T := \frac{\partial P_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial P}{\partial \mathbf{u}}(\mathbf{u}_k), \tag{4}$$

It is recommended to filter the input update $\hat{\mathbf{u}}_{k+1}^*$ to avoid excessive correction and reduce sensitivity to noise [28]:

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \mathbf{L}(\hat{\mathbf{u}}_{k+1} - \mathbf{u}_k), \tag{5}$$

with $\mathbf{L} = \text{diag}(l_1, \ldots, l_{n_u})$, $l_i \in (0, 1]$ where l_i may be reduced to help stabilize the iterations.
The MA scheme requires the estimation of the plant gradients at each RTO iteration, which is experimentally expensive and the main bottleneck for the MA implementation [28]. In this article a wind farm without inequality constraints is considered. Inequality constraints can be easily added to the MA approach. In addition, other cost functions, e.g. including turbine loads can be incorporated.

A. Gaussian processes

In this section GP regression is briefly introduced. GPs are based on kernel methods that aim to describe the unknown function $f : \mathbb{R}^{n_u} \to \mathbb{R}$ using the training set $\mathcal{D}$ consisting of M input vectors $\mathbf{U} = \{\mathbf{u}_j\}_{j=1}^M$ and corresponding noisy outputs $\mathbf{y} = \{y_j\}_{j=1}^M$ [34]. We assume that the noise is additive, independent and Gaussian such that the relationship between function $f(\mathbf{u})$ and the observed noisy outputs y are given by $y_j = f(\mathbf{u}_j) + \nu_j$, where $\nu_j \sim \mathcal{N}(0, \sigma_\nu^2)$ is the variance of the noise [35]. Placing a zero mean Gaussian process prior on the latent function $f(\mathbf{u}_j)$ we get a multivariate Gaussian distribution on a finite subset of latent variables [36]. In particular the function values behave according to $p(\mathbf{f}|\mathbf{U}) = \mathcal{N}(\mathbf{0}, \mathbf{K})$, where $\mathbf{f} = [f_1, \ldots, f_m]^T$ is a vector of latent function values, $f_j = f(\mathbf{u}_j)$ and $\mathbf{K}$ is a covariance matrix.
The covariance matrix is constructed from a covariance function, $K_{ij} = k(\mathbf{u}_i, \mathbf{u}_j)$, which express some prior notion of smoothness of the underlying function [36]. In this work, we use the auto relevance determination (ARD) squared exponential (SE) covariance function [34]

$$k(\mathbf{u}_i, \mathbf{u}_j) = \sigma_n \exp\left(-\frac{1}{2}(\mathbf{u}_i - \mathbf{u}_j)^T \mathbf{\Lambda}(\mathbf{u}_i - \mathbf{u}_j)\right), \tag{6}$$

where σ_n is the covariance magnitude and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_{n_u})$ is a scaling matrix.
The predictive distribution is the marginal of the normalized joint prior times the likelihood. Since both factors in this integral are Gaussian, the integral can be evaluated in closed form. The distribution at an arbitrary input point $\mathbf{u}$ conditioned on the observed data and hyperparameters Ψ has the mean

$$\mu(u, \mathcal{D}, \Psi) = \mathbf{k}_{\mathbf{u}}^T (\mathbf{K} + \sigma_\nu^2 \mathbf{I})^{-1} \mathbf{y}, \tag{7}$$

and variance

$$\nu(u, \mathcal{D}, \Psi) = \mathbf{k}_{\mathbf{uu}} - \mathbf{k}_{\mathbf{u}}^T (\mathbf{K} + \sigma_\nu^2 \mathbf{I})^{-1} \mathbf{k}_{\mathbf{u}}), \tag{8}$$

where $[\mathbf{k}_{\mathbf{u}}]_m = k(\mathbf{u}_m, \mathbf{u})$ and $\mathbf{k}_{\mathbf{uu}} = k(\mathbf{u}, \mathbf{u})$. The GP is a non-parametric model. The training data are explicitly required at the test time to construct the predictive distribution. For the above expression a matrix of size $m \times m$ must be inverted, which prohibits large data sets.
Essential for a good performance of the GP is the choice of the unknown hyperparameters $\Psi := [\sigma_n, \sigma_\nu, \lambda_1, \ldots, \lambda_{n_u}]$, which are usually inferred from the log marginal likelihood $\ln p(\mathbf{y}|\mathbf{U})$:

$$\mathcal{L}(\Psi) = -\frac{1}{2}\mathbf{y}^T (\mathbf{K} + \sigma_\nu^2 \mathbf{I})^{-1} \mathbf{y} - \frac{1}{2} \ln |\mathbf{K} + \sigma_\nu^2 \mathbf{I}| - \frac{m}{2} \ln 2\pi, \tag{9}$$

with the maximisation problem

$$\Psi^* = \max_{\Psi} \mathcal{L}(\Psi). \tag{10}$$

III. Methodology

A. Modifier Adaptation with Gaussian processes

The use of GPs in a MA approach to overcome the limitation of estimating the plant gradients was first proposed by [32]. The idea is to replace the first-order modifier of the cost in (3) with GP regression term

$$\mu^{P_p - P}(\mathbf{u}, \mathcal{D}, \Psi). \tag{11}$$

The new optimization problem of the MA scheme with GP modifiers (MA-GP) is

$$\hat{\mathbf{u}}_{k+1} = \arg\min_{\mathbf{u}} P(\mathbf{u}) + \mu_k^{P_p - P}(\mathbf{u}, \mathcal{D}_k, \Psi_k) \tag{12a}$$

$$\mathbf{u} \in \mathcal{U}, \tag{12b}$$

where the plant-model mismatch of cost function is modelled by $\mu^{P_p - P}$. Similar to the original MA scheme the optimal input of (12) may be filtered with (5) to reduce the step-size and help stabilize the MA-GP scheme [33]. The whole MA-GP scheme is presented in Algorithm 1.

It is not strictly necessary and for large input dimensions and data sets this may even be impractical to update the hyperparameters Ψ of the GP in every iteration. In Algorithm 1 *HypOpt* represents a condition when to update the hyperparameters.

Moreover, to avoid overfitting and numerical difficulties in constructing the GPs caution has to be taken when updating the data sets $\mathcal{D}$. [32] recommended a limited number of historical records in the data set $\mathcal{D}$, which can be achieved by either using only N nearest-neighbors to the new operating point $\mathbf{u}_{k+1}$ or rejecting or substituting the current iterate $\mathbf{u}_{k+1}$ if it is within a given radius to an existing point in the data sets $\mathcal{D}$ [33].

Algorithm 1: Basic MA-GP scheme [33]

```
Initialisation: GP regression model $(\mathcal{GP})^{P_p - P}$, and
                hyperparameters $\Psi^0$ optimization with
                data sets $\mathcal{D}_0$
for $k = 0,1,\ldots$ do
    Solve modified optimization problem (12);
    Filter new operating point $\mathbf{u}_{k+1}$ with (5);
    Obtain measurements of cost function $P_p(\mathbf{u}_{k+1})$;
    Update the data set $\mathcal{D}_{k+1}$ with measurements ;
    Update GP regression term $(\mathcal{GP})^{P_p - P}$ with data
      set $\mathcal{D}_{k+1}$ ;
    if HypOpt then
        Update hyperparameters $\Psi_{k+1}$ with new data
          set $\mathcal{D}_{k+1}$;
    end
end
```

B. Distributed learning for plant-model mismatch

Let us assume we like to optimize a distributed system consisting of N agents, which interact with each other and contribute to the overall objective function Φ, e.g. a wind farm with N wind turbines. In the basic MA-GP approach the overall objective function is modified with a GP regression model of the plant-model mismatch. However, in problems with, e.g. large input dimensions, the correct identification of the regression model may be difficult. In such a situation we propose to learn the plant-model mismatch of each agent individually before solving the RTO problem.

The MA-GP scheme with distributed learning becomes

$$\hat{\mathbf{u}}_{k+1} = \arg\min_{\mathbf{u}} P(\mathbf{u}) + \sum_{i}^{N} \mu_{k,i}^{p_{p,i} - p_i}(\mathbf{u}_i, \mathcal{D}_i, \Psi_i) \tag{13a}$$

$$\mathbf{u} \in \mathcal{U}, \quad \mathbf{u} = [u_1^T, u_2^T, \ldots, u_N^T]^T, \tag{13b}$$

where $\mu_{k,i}^{p_{p,i} - p_i}$ are the mean predictions of the GP regression terms of the plant-model mismatch of the individual turbines. The optimal input of (13) may again be filtered with (5). In fact, the only difference to the basic MA-GP scheme is the GP training, which consist of N GPs (Fig. 1).

The inputs $\mathbf{u}_i$ used to identify the individual turbines may be a subset of the plant inputs because either from the system structure it is known that the excluded inputs do not influence the turbine, e.g. control inputs of downstream turbines do not influence the output of upstream turbines, or only limited communication between turbines is allowed, e.g. only nearest-neighbor communication. The former does not influence the quality of the model while for the latter parts of the explanatory variables are neglected, which usually decreases the performance of the model.

For the distributed learning approach N GPs have to be trained instead of one for the cost function. This is a considerable amount of additional computational work, which, however, can be parallelized. Distributed learning can also be applied to constraints if present and the constraints have a similar form as the cost function presented. Indeed, not every global constraint is suitable for distributed learning.

IV. Case Study

In this section we apply the basic MA-GP scheme and the MA-GP scheme with distributed learning to a wind farm. The wind turbines are represented using the actuator disc theory to couple the power and thrust coefficient, C_P and C_T [37], [38]

$$C_P = 4a(1-a)^2, \tag{14}$$

$$C_T = 4a(1-a), \tag{15}$$

where a is the axial induction factor. The axial induction factor indicates the ratio of wind velocity reduction at the turbine compared to the upstream wind velocity. The steady-state power of each turbine under yaw misalignment is given by [39]

$$P = \frac{1}{2}\rho A C_P \cos\gamma^{\kappa} v^3, \tag{16}$$

where A is the rotor area, ρ the air density and κ a correction factor and v is the wind velocity. In the actuator disc theory

[1] The wind farm picture is by Erik Wilde from Berkeley, CA, USA `https://www.flickr.com/photos/dret/24110028330/`, *Wind turbines in southern California 2016*, `https://creativecommons.org/licenses/by-sa/2.0/legalcode`

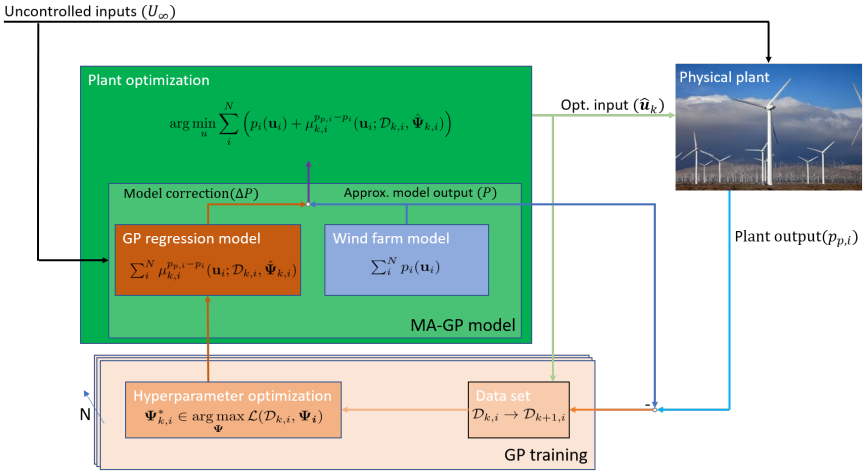

Uncontrolled inputs (U_∞)
Plant optimization
$\arg\min_u \sum_i^N \left(p_i(\mathbf{u}_i) + \mu_{k,i}^{p_{p,i}-p_i}(\mathbf{u}_i; \mathcal{D}_{k,i}, \hat{\mathbf{\Psi}}_{k,i})\right)$
Model correction(ΔP)
Approx. model output (P)
GP regression model
$\sum_i^N \mu_{k,i}^{p_{p,i}-p_i}(\mathbf{u}_i; \mathcal{D}_{k,i}, \hat{\mathbf{\Psi}}_{k,i})$
Wind farm model
$\sum_i^N p_i(\mathbf{u}_i)$
MA-GP model
Physical plant
Opt. input ($\hat{\boldsymbol{u}}_k$)
Plant output($p_{p,i}$)
N
Hyperparameter optimization
$\mathbf{\Psi}_{k,i}^* \in \arg\max_{\mathbf{\Psi}} \mathcal{L}(\mathcal{D}_{k,i}, \mathbf{\Psi}_i)$
Data set
$\mathcal{D}_{k,i} \rightarrow \mathcal{D}_{k+1,i}$
-
GP training

Fig. 1: The basic idea of the distributed MA-GP scheme for a wind farm [1].

$\kappa = 3$ [37]. However, based on large-eddy simulations, the turbine power yaw misalignment has been shown to match the output when $\kappa = 1.88$ for the NREL 5MW turbine [40], which is used in the article.

The Gaussian model proposed by [41] and extended for yaw misalignment by [42] is used to model the aerodynamic interactions between turbines. The three-dimensional far wake velocity $\bar{v}(x, y, z)$ of a yawed turbine can be calculated with

$$\frac{\bar{v}(x,y,z)}{\bar{v}_\infty} = 1 - Ce^{-0.5((y-\delta)/\sigma_y)^2} e^{-0.5((z-z_h)/\sigma_z)^2}, \quad (17a)$$

$$C = 1 - \sqrt{1 - \frac{C_T \cos\gamma}{8(\sigma_y \sigma_z / d^2)}}, \quad (17b)$$

where $\bar{v}_\infty$ is the upwind velocity, C_T is the thrust coefficient, γ is the yaw angles, which is assumed positive in clockwise direction, z_h is the tower height, δ is the wake deflection, and σ_y and σ_z are the wake widths in lateral and vertical directions, respectively. For more details about the model the reader is referred to [38], [42].

The control inputs $\mathbf{u}$ of the wind farm are the yaw γ_i and thrust coefficients $C_{T,i}$ of the N turbines. Consequently, a farm has in total $2N$ control inputs. The objective is maximizing the power production $P_{tot} = \sum_i P_i$ of the wind farm. Only box constraints on the control inputs, $C_{T,i} \in [0, 0.95]$, $\gamma_i \in [0, {}^2/_9\pi]$, are implemented. The yaw misalignment is constraint to clock-wise rotations. The Gaussian wake model, which is used to represent the model and plant in this article, is symmetric. Therefore, the algorithm would not be able to differentiate between clockwise and counter-clockwise rotations. Training on more more accurate data would prevent this behavior [43].

In the following the basic MA-GP scheme and the MA-GP scheme with distributed learning are compared on wind farms with different complexity. For the turbine dimensions the NREL 5-MW wind turbine is used [44]. The plant is modelled with the Gaussian model presented, while for the model in the MA-GP scheme several parameters of the actuator disc and wake model were changed resulting in a nonlinear behaviour of the plant-model mismatch. Both the plant and the MA-GP model still represent a behaviour of a wind farm.

The identified model (model plus correction) is compared to the plant model on a test set. The root-mean square error (RMSE) is used as performance index

$$\Gamma = \sqrt{\frac{1}{M} \sum_i^M (P_m - P_p)^2}, \quad (18)$$

where P_m and P_p are the corrected model and plant power production and M is the size of the test set. The performance of the overall MA-GP approach is given by the percentage of relative error in power production

$$\Theta = 100 \frac{P_p^* - \hat{P}}{P_p^*}, \quad (19)$$

where P_p^* and $\hat{P}$ are the optimal power production of the plant and MA-GP approach, respectively.

A. Three turbine case

In this case a row of three turbines is simulated. The turbines face the wind and the spacing between turbines is $5D$, where D is the turbine diameter. Three cases, a zero noise, small noise and large noise case, are tested with differently large initial training data sets. The small noise has a standard deviation of about 0.5 % to 0.8 % and the large noise of about 5 % to 8 % with respect to the nominal power signal of each turbine. The same noise is applied to the central and distributed approach. In case of the central learning approach the sum of the three measurement noises of the single turbines is taken to get the overall noise of the power signal. The optimization and training is run for 100 iterations for both the centralised training and distributed training approach.

The simulation results are summarized in Tab. I. It can be seen that the performance of the distributed learning approach is superior to the central learning approach. However,

TABLE I: The root-mean square error Γ_{N0} and Γ_{N100} of the identified model on the same test set after initial training and 100 iterations; and the relative performance error Θ of the MA-GP approach after 100 iterations. The MA-GP updates hyperparameters and data set in each iteration. The initial training set consist of either N= 60, 120 or 180 operation points, and the standard deviation of the noise is either σ_0 = 0 W, σ_1 = 5 kW or σ_2 = 50 kW.

		N	Γ_{N0} [$\cdot 10^4$]	Γ_{N100} [$\cdot 10^4$]	Θ [%]
Central	σ_0	60	2.57	2.17	0.016
		120	1.67	1.27	0.019
		180	0.98	0.95	0.012
	σ_1	60	3.00	2.72	0.180
		120	1.90	1.67	0.044
		180	1.57	1.31	0.032
	σ_2	60	7.01	6.34	1.450
		120	5.64	5.12	0.965
		180	5.92	4.40	0.734
Distributed	σ_0	60	1.89	1.62	0.045
		120	1.04	0.95	0.020
		180	0.73	0.70	0.013
	σ_1	60	1.87	1.71	0.071
		120	1.21	1.13	0.033
		180	0.94	0.91	0.027
	σ_2	60	6.05	5.30	0.822
		120	4.09	3.90	0.450
		180	3.55	3.51	0.372

for the noise-free case the optimization error is slightly better even though the RMSE error of the identified model is larger. Noise increases the RMSE and the optimization error. More initial training data points are extremely beneficial for the large noise cases.

Including the new operation points of each iteration in the GP data set is beneficial and reduces the RMSE. It has to be noted that the 100 new operation points contain small amount of information since the algorithm approaches the optimal operation point within a few iterations and afterwards stays there. In fact, it also shows that the algorithm is relatively robust to overfitting. However, we observed that in many test cases the smallest optimization error was reached within 6 to 20 iterations[2] and afterwards the error slightly deteriorated.

In addition, the performance of the approach was tested if only the operation points are added to the GP data set without updating the hyperparameter and the case without updating data set and hyperparameters. For the former case only slight changes in the RMSE and optimization error are observed. However, for the large noise case with small initial data set the RMSE and optimization error increases drastically. The hyperparameters, especially the noise term, are not well identified with the small data set and cause large overfitting problems when new almost identical operation points with different outputs are added. Not updating the GP data set causes usually a larger RMSE at the end of the 100 iterations. The optimization error changes slightly compared to the case where both data set and hyperparmeter are updated. For larger initial data sets the optimization error sometimes even decreases.

B. Ten turbine case

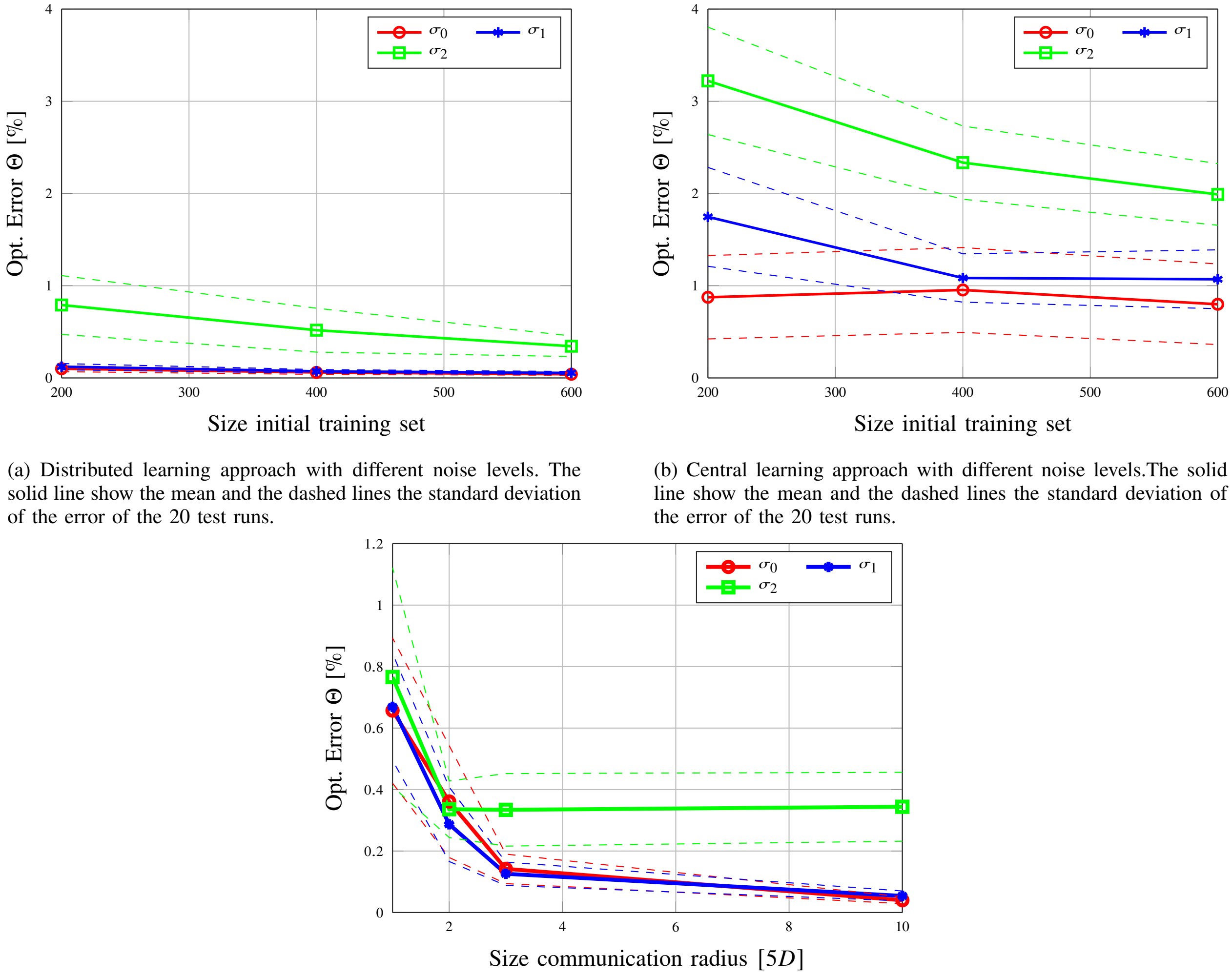
In this case a row of ten turbines facing the wind is simulated. The spacing is $5D$, where D is the turbine diameter. The same noise classes as in the previous test case are used. The MA-GP algorithm is run for 25 iterations and 20 runs are performed of each test case.

The results of different test runs are summarized in Fig. 2, where the mean error and standard deviation of the 20 test runs of each test case are shown. Again, a larger initial training set improves usually the performance of the MA-GP approach. For the noise-free central learning approach the test case with a training set of 200 data points converges to a smaller error than the case with 400 data points. This is an exception but was already observed in the three turbine test case. The optimization results with a small initial training set have usually the largest variance. Consequently, their performance is stronger dependent of the data in the training sets. However, the variance of the optimization error for the noise-free central learning approach is almost independent of the training set size.

Comparing Fig. 2a and 2b we notice that the distributed learning approach clearly outperforms the central learning approach. Even the large noise cases of the distributed learning approach show better performance than the noise-free central learning approach. In fact, the performance of the central learning approach decreases strongly under influence of noise while the distributed learning approach is less affected by noise. The performance of the small noise case σ_1 and the noise-free case are almost identical indicating robustness of the approach. Moreover, the error of the distributed learning approach increases only slightly from the three to the ten turbine case while a significant increase is observed for the central learning approach. It indicates that the input-output map between control inputs and total power production becomes more difficult to identify as the input space increases. The sensitivities between control inputs and total power production are small especially considering that, e.g. derating and yawing a turbine decreases the power production of the turbine itself but increases the power production of the downwind turbines. The information of power production of the individual turbines, which is used in the distributed learning approach, improves the identification of the input-output map, and makes the approach more suitable for large wind farms.

Limited communication decreases the performance of the MA-GP approach for the noise-free and small noise cases (Fig. 2c). This is not surprising since part of the energy flow in the plant is neglected and cannot be explained by the model. For the cases with large noise σ_2, on the other hand, the limited communication case considering turbines only in a radius of two times the turbine spacing (two upwind and two downwind turbines - in total the control inputs of five turbines are considered in the model) converges to a very similar error as the full communication case. The influence of turbines further away is within the noise range and cannot be correctly identified by the GP-model. Therefore, neglecting

[2] Faster convergence can be achieved with a larger filter constant, which was set to $k_i = 0.4$.



(a) Distributed learning approach with different noise levels. The solid line show the mean and the dashed lines the standard deviation of the error of the 20 test runs.

(b) Central learning approach with different noise levels.The solid line show the mean and the dashed lines the standard deviation of the error of the 20 test runs.

(c) Influence of communication radius on the optimization error with an intial training set size of 600 data points. The x-axis indicates of how many upwind turbines the control inputs are used in the learning of the individual GPs.

Fig. 2: The optimization error Θ in dependency of the size of the initial training set for the ten turbine test case. Red color and round marker indicate case with noise σ_0, blue color and asterisk marker indicate case with noise σ_1, and green color and square marker indicate case with noise σ_2.

them is reasonable and may be positive for the overall performance of the MA-GP approach. Still the RMSE is first similar to the full communication case when turbines in the radius of three times the turbine spacing are used.

Indeed, for other sizes of the initial training set the cases with large noise and with communication in a radius of three times the turbine spacing usually showed at least similar or even better performance than the full communication case. In contrast, the root-mean square error Γ_{N25} of the full communication case is in all cases smaller than with limited communication.

In general, smaller computational times for the optimizations of the hyperparameters or finding new operation points were not observed for the limited in comparison to the full communication case.

V. Conclusion

This article proposes the use of the MA-GP approach for wind farm control. The approach connects real-time optimization with machine learning. A distributed learning approach is proposed in which the objective functions of the individual turbines are learned prior to the optimization of the plant inputs. This approach uses more information about the plant than the central learning approach and clearly outperforms it. The benefits of the distributed learning approach become increasingly important for plants with many turbines and large control input spaces. The better performance comes with the cost of identifying as many GP regression models as turbines in the plant in comparison to one GP regression model for the central learning approach. Indeed, the distributed learning process can be completely parallelized, which negates this drawback.

References

[1] J. Annoni, P. Gebraad, and P. Seiler, “Wind farm flow modeling using an input-output reduced-order model,” in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 506–512.

[2] IRENA, “Renewable power generation costs in 2017,” 2018.

[3] P. Fleming, A. Ning, P. M. Gebraad, and K. Dykes, “Wind plant system engineering through optimization of layout and yaw control,” *Wind Energy*, vol. 19, no. 2, pp. 329–344, 2016.

[4] M. Steinbuch, W. De Boer, O. Bosgra, S. Peeters, and J. Ploeg, “Optimal control of wind power plants,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 27, no. 1-3, pp. 237–246, 1988.

[5] S. Frandsen, “Turbulence and turbulence-generated structural loading in wind turbine clusters,” Ph.D. dissertation, 2007, risø-R-1188(EN).

[6] K. E. Johnson and N. Thomas, “Wind farm control: Addressing the aerodynamic interaction among wind turbines,” in *American Control Conference, 2009. ACC’09.* IEEE, 2009, pp. 2104–2109.

[7] R. J. Barthelmie, S. C. Pryor, S. T. Frandsen, K. S. Hansen, J. Schepers, K. Rados, W. Schlez, A. Neubert, L. Jensen, and S. Neckelmann, “Quantifying the impact of wind turbine wakes on power output at offshore wind farms,” *Journal of Atmospheric and Oceanic Technology*, vol. 27, no. 8, pp. 1302–1317, 2010.

[8] K. S. Hansen, R. J. Barthelmie, L. E. Jensen, and A. Sommer, “The impact of turbulence intensity and atmospheric stability on power deficits due to wind turbine wakes at horns rev wind farm,” *Wind Energy*, vol. 15, no. 1, pp. 183–196, 2012.

[9] T. Knudsen, T. Bak, and M. Svenstrup, “Survey of wind farm control - power and fatigue optimization,” *Wind Energy*, vol. 18, no. 8, pp. 1333–1351, 2015.

[10] M. A. Rotea, “Dynamic programming framework for wind power maximization,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3639–3644, 2014.

[11] J. Annoni, P. M. Gebraad, A. K. Scholbrock, P. A. Fleming, and J.-W. v. Wingerden, “Analysis of axial-induction-based wind plant control using an engineering and a high-order wind plant model,” *Wind Energy*, vol. 19, no. 6, pp. 1135–1150, 2016.

[12] T. Horvat, V. Spudić, and M. Baotić, “Quasi-stationary optimal control for wind farm with closely spaced turbines,” in *MIPRO, 2012 Proceedings of the 35th International Convention.* IEEE, 2012, pp. 829–834.

[13] E. Bossanyi and T. Jorge, “Optimisation of wind plant sector management for energy and loads,” in *Control Conference (ECC), 2016 European.* IEEE, 2016, pp. 922–927.

[14] M. Vali, V. Petrović, S. Boersma, J.-W. van Wingerden, L. Y. Pao, and M. Kühn, “Model predictive active power control of waked wind farms,” in *2018 Annual American Control Conference (ACC).* IEEE, 2018, pp. 707–714.

[15] W. Munters and J. Meyers, “Effect of wind turbine response time on optimal dynamic induction control of wind farms,” in *Journal of Physics: Conference Series*, vol. 753, no. 5. IOP Publishing, 2016, p. 052007.

[16] J. Park, S. Kwon, and K. H. Law, “Wind farm power maximization based on a cooperative static game approach,” in *Active and Passive Smart Structures and Integrated Systems 2013*, vol. 8688. International Society for Optics and Photonics, 2013, p. 86880R.

[17] P. M. Gebraad and J. Van Wingerden, “A control-oriented dynamic model for wakes in wind plants,” in *Journal of Physics: Conference Series*, vol. 524, no. 1. IOP Publishing, 2014, p. 012186.

[18] J. J. Thomas, P. M. Gebraad, and A. Ning, “Improving the floris wind plant model for compatibility with gradient-based optimization,” *Wind Engineering*, vol. 41, no. 5, pp. 313–329, 2017.

[19] W. Munters and J. Meyers, “Dynamic strategies for yaw and induction control of wind farms based on large-eddy simulation and optimization,” *Energies*, vol. 11, no. 1, p. 177, 2018.

[20] J. Park, S.-D. Kwon, and K. Law, “A data-driven, cooperative approach for wind farm control: a wind tunnel experimentation,” *Energies*, vol. 10, no. 7, p. 852, 2017.

[21] R. J. Barthelmie, K. S. Hansen, and S. C. Pryor, “Meteorological controls on wind turbine wakes,” *Proceedings of the IEEE*, vol. 101, no. 4, pp. 1010–1019, 2013.

[22] J. Annoni, P. Seiler, K. Johnson, P. Fleming, and P. Gebraad, “Evaluating wake models for wind farm control,” in *2014 American Control Conference*, June 2014, pp. 2517–2523.

[23] P.-E. Réthoré, K. S. Hansen, R. Barthelmie, S. Pryor, G. Sieros, J. Prospathopoulos, J. Palma, V. Gomes, G. Schepers, P. Stuart, *et al.*, “Benchmarking of wind farm scale wake models in the eera-dtoc project,” in *International Conference on aerodynamics of Offshore Wind Energy Systems and wakes (ICOWES 2013).* Technical University of Denmark (DTU), 2013.

[24] M. Bastankhah and F. Porté-Agel, “Wind tunnel study of the wind turbine interaction with a boundary-layer flow: Upwind region, turbine performance, and wake region,” *Physics of Fluids*, vol. 29, no. 6, p. 065105, 2017.

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

[26] L. A. Martínez-Tossas, J. Annoni, P. A. Fleming, and M. J. Churchfield, “The aerodynamics of the curled wake: A simplified model in view of flow control,” *Wind Energy Science Discussions*, vol. 2018, pp. 1–17, 2018.

[27] P. Fleming, J. Annoni, M. Churchfield, L. A. Martinez-Tossas, K. Gruchalla, M. Lawson, and P. Moriarty, “A simulation study demonstrating the importance of large-scale trailing vortices in wake steering,” *Wind Energy Science*, vol. 3, no. 1, pp. 243–255, 2018.

[28] A. G. Marchetti, G. François, T. Faulwasser, and D. Bonvin, “Modifier adaptation for real-time optimization – methods and applications,” *Processes*, vol. 4, no. 4, p. 55, 2016.

[29] C. Y. Chen and B. Joseph, “On-line optimization using a two-phase approach: An application study,” *Industrial & engineering chemistry research*, vol. 26, no. 9, pp. 1924–1930, 1987.

[30] M. L. Darby, M. Nikolaou, J. Jones, and D. Nicholson, “Rto: An overview and assessment of current practice,” *Journal of Process Control*, vol. 21, no. 6, pp. 874–884, 2011.

[31] A. Marchetti, B. Chachuat, and D. Bonvin, “Modifier-adaptation methodology for real-time optimization,” *Industrial & engineering chemistry research*, vol. 48, no. 13, pp. 6022–6033, 2009.

[32] T. de Avila Ferreira, H. A. Shukla, T. Faulwasser, C. N. Jones, and D. Bonvin, “Real-time optimization of uncertain process systems via modifier adaptation and gaussian processes,” in *2018 European Control Conference (ECC).* IEEE, 2018, pp. 465–470.

[33] E. A. del Rio-Chanona, J. E. A. Graciano, E. Bradford, and B. Chachuat, “Modifier-Adaptation Schemes Employing Gaussian Processes and Trust Regions for Real-Time Optimization,” *IFAC-PapersOnLine*, vol. 52, no. 1, pp. 52–57, 2019.

[34] C. E. Rasmussen and C. K. Williams, “Gaussian processes for machine learning,” *the MIT Press*, vol. 2, no. 3, p. 4, 2006.

[35] J. Quiñonero-Candela and C. E. Rasmussen, “A unifying view of sparse approximate gaussian process regression,” *Journal of Machine Learning Research*, vol. 6, no. Dec, pp. 1939–1959, 2005.

[36] E. Snelson and Z. Ghahramani, “Sparse gaussian processes using pseudo-inputs,” in *Advances in neural information processing systems*, 2006, pp. 1257–1264.

[37] T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi, *Wind energy handbook.* John Wiley & Sons, 2011.

[38] J. Annoni, C. Bay, T. Taylor, L. Pao, P. Fleming, and K. Johnson, “Efficient optimization of large wind farms for real-time control,” in *2018 Annual American Control Conference (ACC).* IEEE, 2018, pp. 6200–6205.

[39] P. Gebraad, F. Teeuwisse, J. Van Wingerden, P. A. Fleming, S. Ruben, J. Marden, and L. Pao, “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.

[40] J. Annoni, P. Fleming, A. Scholbrock, J. Roadman, S. Dana, C. Adcock, F. Porte-Agel, S. Raach, F. Haizmann, and D. Schlipf, “Analysis of control-oriented wake modeling tools using lidar field results,” *Wind Energy Science*, vol. 3, no. 2, pp. 819–831, 2018.

[41] M. Bastankhah and F. Porté-Agel, “A new analytical model for wind-turbine wakes,” *Renewable Energy*, vol. 70, pp. 116–123, 2014.

[42] ——, “Experimental and theoretical study of wind turbine wakes in yawed conditions,” *Journal of Fluid Mechanics*, vol. 806, pp. 506–541, 2016.

[43] L. E. Andersson, B. Doekemeijer, D. van der Hoek, and L. Imsland, “Adaptation of engineering wake models using gaussian process regression and high-fidelity simulation data,” in *TORQUE 2020 - [accepted]*, 2020.

[44] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, “Definition of a 5-mw reference wind turbine for offshore system development,” *National Renewable Energy Laboratory, Golden, CO, Technical Report No. NREL/TP-500-38060*, 2009.