Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Gaussian processes modifier adaptation with uncertain inputs for distributed learning and optimization of wind farms [1]

Leif Erik Andersson *, **Eric Christopher Bradford** *, **Lars Imsland** *,

* *Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7491 Trondheim, Norway. (e-mail: {lars.imsland, eric.bradford, leif.e.andersson}@ntnu.no)*

Abstract: A modifier adaptation scheme based on Gaussian processes is presented to optimize the control inputs of a wind farm. Often an approximate model of the wind farm is available, however due to the high complexity of the process plant-model mismatch is prevalent. For example the mechanics of wakes is not well-understood, which may have a profound impact on the power production of wind farms. Therefore, Gaussian process (GP) regression is exploited to account for this deviation. A distributed learning approach is used to learn the plant-model mismatch of each individual turbine considering explicitly the uncertainty of the uncontrolled inputs, like the wind direction. Afterwards, a distributed optimization scheme using alternating direction method of multipliers is applied to iteratively attain the wind farm optimum despite the presence of plant-model mismatch.

1. INTRODUCTION

In the last few decades wind energy production has grown rapidly. Nowadays about 5% of the global energy production is supplied by wind (Veers et al., 2019). The development is driven by high renewable energy targets in, e.g. the US, Europe, and China, but also by competitive costs of wind energy compared to conventional fossil-fueled power plants. Most wind turbines are deployed in wind farms, since it reduces deployment costs and maintenance costs (Fleming et al., 2016). On the other hand, wakes may develop behind turbines, which reduce flow velocity and increase turbulence intensity (Barthelmie et al., 2009). The overall energy production of turbines grouped in a wind farm, therefore, decreases due to the reduced flow velocity in comparison to the same number of individual turbines Steinbuch et al. (1988), and load variations of turbine parts increase due to the heightened turbulence intensity.

Several studies on wind farms show that operating the turbines at their individual optimal operating points leads to suboptimal performance (Steinbuch et al., 1988; Knudsen et al., 2015). Therefore, it is generally accepted that coordinated control of the turbines in a wind farm has the potential to reduce the levelized cost of wind energy (Boersma et al., 2019).

The two most common approaches for wind farm control are power de-rating control, e.g. Corten and Schaak (2003); Rotea (2014); Munters and Meyers (2016); Duc et al. (2019) and wake steering control, e.g. Medici (2005); Wagenaar et al. (2012); Park et al. (2013); Gebraad et al. (2016). The idea behind power de-rating is to operate the upwind turbine sub-optimally by changing the blade pitch and generator torque. Consequently, the upwind turbine causes a smaller velocity reduction of the wake. This can be exploited by the downwind turbine to increase its energy production. The target net effect is an increase in the overall energy product and a decrease in possible loads. For wake steering a yaw offset is applied to the upstream turbine to deflect the wake away from the downwind turbine.

The impact of wakes on the turbines in the plant and its interactions with the surrounding atmospheric boundary layer is still not well understood, which is a major challenge in the design of wind farm controllers. In model-based wind farm control usually simple engineering models are used to estimate the velocity deficit in the wake (Jensen, 1983; Bastankhah and Porté-Agel, 2016). The models often calculate the steady-state situation and can represent the general behavior of wakes (Barthelmie et al., 2013; Annoni et al., 2014). Nonetheless, the complex dynamics of wakes are just approximated by these models and a plant-model mismatch exists. Several model-free optimization methods were investigated to drive the plant iteratively to the plant optimum (Marden et al., 2013; Gebraad and Van Wingerden, 2014; Johnson and Fritsch, 2012; Ciri et al., 2017). These model-free optimization methods have however slow rates of convergence.

Another approach to improve the model is the two-step approach, which consists of repeated parameter estimation and optimization (Chen and Joseph, 1987; Darby et al., 2011). For wind farm control such an approach is, for example, proposed in Doekemeijer et al. (2019). However, the approach cannot guarantee plant optimality upon convergence if the model is structurally wrong (Marchetti et al., 2016). On the other hand, modifier adaptation (MA) is a real-time optimization (RTO) method that corrects the cost and constraint functions of the optimization problem directly, and reaches, under suitable assumptions, true plant optimality upon convergence (Marchetti et al.,

[1] This work was supported by OPWIND, RCN project no. 268044.

2009).
In this article we extend the previously proposed combination of MA and Gaussian processes (GP) regression for wind farm optimization (Andersson and Imsland, 2020; Andersson et al., 2020). Firstly, to identify the plant-model mismatch we not only consider wind turbine control inputs, but in addition consider uncontrolled inputs to the plant, such as wind direction or wind velocity. Secondly, GP regression is extended to account for uncertainties in the inputs explicitly. Lastly, each wind turbine is optimized individually and coordinated using a distributed optimization algorithm to determine the plant optimum.
The article is organized as follows. In section 2 a brief introduction of the optimization problem, the MA approach, the GP regression and its extension to consider uncertain inputs is given. In section 3 the distributed learning and optimization of the modifier-adaptation approach with Gaussian processes (MA-GP) is presented. The performance of the approach is illustrated on a numerical example in section 4. The article ends with a conclusion.

2. PROBLEM FORMULATION AND PRELIMINARIES

A steady-state plant performance subject to constraints is usually optimized using an approximate model. Consequently, plant-model mismatch exists, which can lead to sub-optimal performance. The standard MA approach takes advantage of available measurements to compensate for plant-model mismatch and adapts the model-based optimization problem to reach plan optimality. It applies zeroth- and first-order correction terms to the cost and constraint functions to match the necessary condition of optimality upon convergence (Marchetti et al., 2009). The first-order correction terms require the estimation of plant gradients at each RTO iteration, which is experimentally expensive and the main bottleneck of the MA implementation in practice (Marchetti et al., 2016).

2.1 Gaussian processes

In this section we give a brief outline of GP regression for our purposes, for more information refer to Rasmussen and Williams (2006). GP regression aims to identify an unknown function $f : \mathbb{R}^{n_u} \to \mathbb{R}$ from data. Let the noisy observation of $f(\cdot)$ be given by:

$$y = f(\mathbf{u}) + \nu \tag{1}$$

where the input $\mathbf{u}$ is assumed to follow a multivariate Gaussian distribution with mean $\bar{\mathbf{u}}$ and covariance $\mathbf{\Sigma_u}$, $\mathbf{u} \sim \mathcal{N}(\bar{\mathbf{u}}, \mathbf{\Sigma_u})$, and the value $f(\cdot)$ is perturbed by Gaussian noise ν with zero mean and variance σ_ν^2, $\nu \sim \mathcal{N}(0, \sigma_\nu^2)$.
We assume $f(\cdot)$ to follow a GP with a zero mean function and the squared-exponential (SE) covariance function. The choice of the mean and covariance functions assume certain smoothness and continuity properties of the underlying function (Snelson and Ghahramani, 2006). The SE covariance function can be expressed as follows:

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

where σ_f^2 is the covariance magnitude and
$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1^2, \ldots, \lambda_{n_u}^2)$ is a scaling matrix.
While output noise ν is easily accounted for in the GP formulation, the input noise on $\mathbf{u}$ is much more difficult to deal with. Nonetheless, if ignored it can lead to significantly worse GP predictions. The *proper* way to handle input noise is to integrate it out, which is however computationally expensive (Goldberg et al., 1998). Instead, we apply an approach first proposed in Dallaire et al. (2009), which suggests applying the expected SE. This covariance function can be stated as:

$$\begin{aligned}\hat{k}_n\left((\mathbf{u}_i, \mathbf{\Sigma_u}), (\mathbf{u}_j, \mathbf{\Sigma_u})\right) &= \int\int k(\mathbf{u}_i, \mathbf{u}_j) p(\mathbf{u}_i) p(\mathbf{u}_j) d\mathbf{u}_i d\mathbf{u}_j \\ &= \hat{\sigma}_{n_f}^2 \exp\left(-\frac{1}{2}(\mathbf{u}_i - \mathbf{u}_j)^T \hat{\mathbf{\Lambda}}_n^{-1}(\mathbf{u}_i - \mathbf{u}_j)\right)\end{aligned} \tag{3}$$

where $\hat{\sigma}_{n_f}^2 = \sigma_f^2 |\mathbf{I} + 2\hat{\mathbf{\Lambda}}_n^{-1}\mathbf{\Sigma_u}|^{-\frac{1}{2}}$ and $\hat{\mathbf{\Lambda}}_n^{-1} = (\mathbf{\Lambda} + 2\mathbf{\Sigma_u})^{-1}$. In this $\mathbf{u}_i$ and $\mathbf{u}_j$ are the mean values of the inputs.
The main idea is then to use the covariance function in (3) instead of the one stated in (2), which alleviates the bias resulting from noisy inputs. Assume we are given a training dataset $\mathcal{D} = \{\mathbf{U}, \mathbf{Y}\}$ of size M consisting of M input vectors $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_M]^\mathsf{T}$ and corresponding observations $\mathbf{y} = [y_1, \ldots, y_M]^\mathsf{T}$ according to (1). From the GP distribution the data then follows a joint multivariate Gaussian distribution, which can be stated as:

$$p(\mathbf{y}|\mathbf{U}) = \mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma_\nu^2\mathbf{I}), \quad K_{ij} = \hat{k}_n\left((\mathbf{u}_i, \mathbf{\Sigma_u}), (\mathbf{u}_j, \mathbf{\Sigma_u})\right) \tag{4}$$

Note that the actual mean values of the input data are unknown, and instead we use the values in $\mathbf{U}$ as proposed in McHutchon and Rasmussen (2011).
The hyperparameters $\boldsymbol{\psi} := [\sigma_f, \sigma_\nu, \lambda_1, \ldots, \lambda_{n_u}]^T$ are commonly unknown and hence need to be inferred from data. Most papers maximize the marginal likelihood to obtain these, the performance of which has however been shown to be strongly dependent on the assumptions made. In our case due to the input noise the GP assumptions do not hold exactly and in this case, it is more reasonable to use the less common leave-one-out (LOO) log-likelihood instead (Sundararajan and Keerthi, 2000). Ignoring constant terms and factors, this can be stated as:

$$\mathcal{L}_{\mathrm{LOO}}(\mathcal{D}, \boldsymbol{\psi}) = \sum_{i=1}^{M}\left(-\log\sigma_i^2 - \frac{(y_i - \mu_i)^2}{\sigma_i}\right) \tag{5}$$

where $\mu_i = y_i - \frac{[\mathbf{K}^{-1}\mathbf{y}]_i}{K_{ii}^{-1}}$ and $\sigma_i^2 = \frac{1}{K_{ii}^{-1}}$.
The required maximum likelihood estimate is then given by $\hat{\boldsymbol{\psi}} \in \arg\max_{\boldsymbol{\psi}} \mathcal{L}_{\mathrm{LOO}}(\mathcal{D}, \boldsymbol{\psi})$.
Next we require the predictive distribution of $f(\mathbf{u})$ at an arbitrary input $\mathbf{u}$, which can be found by the conditional distribution of $f(\mathbf{u})$ on the data distribution $p(\mathbf{y}|\mathbf{U})$. From the GP assumption this has a closed-form solution and can be stated as:

$$f(\mathbf{u})|\mathcal{D}, \hat{\boldsymbol{\psi}} \sim \mathcal{N}(\mu_{\mathrm{GP}}(\mathbf{u}; \mathcal{D}, \hat{\boldsymbol{\psi}}), \sigma_{\mathrm{GP}}^2(\mathbf{u}; \mathcal{D}, \hat{\boldsymbol{\psi}})) \tag{6}$$

$$\mu_{\mathrm{GP}}(\mathbf{u}; \mathcal{D}, \hat{\boldsymbol{\psi}}) = \mathbf{k}^T(\mathbf{u})(\mathbf{K} + \sigma_\nu^2\mathbf{I})^{-1}\mathbf{y} \tag{7}$$

$$\sigma_{\mathrm{GP}}^2(\mathbf{u}; \mathcal{D}, \hat{\boldsymbol{\psi}})) = \sigma_f^2 - \mathbf{k}^T(\mathbf{u})(\mathbf{K} + \sigma_\nu^2\mathbf{I})^{-1}\mathbf{k}(\mathbf{u}) \tag{8}$$

where $\mu_{\mathrm{GP}}(\mathbf{u}; \mathcal{D}, \hat{\boldsymbol{\psi}})$ can be seen as the GP prediction at $\mathbf{u}$ and $\sigma_{\mathrm{GP}}^2(\mathbf{u}; \mathcal{D}, \hat{\boldsymbol{\psi}})$ as a corresponding measure of uncertainty to this prediction.

3. METHODOLOGY

3.1 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 de Avila Ferreira et al. (2018). The idea is to replace the zeroth- and first-order modifiers of the cost and constraints in the traditional MA with GP regression terms. Combining MA and GP for wind farm optimization was introduced in Andersson and Imsland (2020).
In context of this article the power production of the wind farm is maximized without constrained functions. The optimization problem of the MA scheme with GP modifiers becomes

$$\hat{\mathbf{u}}^*_{k+1} = \arg\max_{\mathbf{u}} \sum_{i}^{N} \phi_i(\mathbf{u}_i) + \mu^{\phi_p-\phi}_{\mathrm{GP},k}(\mathbf{u}; \mathcal{D}, \hat{\boldsymbol{\psi}}) \tag{9a}$$

$$\mathbf{u} \in \mathcal{U}, \tag{9b}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ denote the plant input variables; ϕ_i and $\mathbf{u}_i \in \mathbb{R}^{n_{u,i}}$ are the power production and the inputs to the turbines of the plant; μ_{GP} is the plant-model mismatch of the cost functions represented by a GP and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ is the control domain, e.g. box constraints on the control inputs. The training set of the GP consist of the controlled and uncontrolled inputs and the mismatch in power production between plant and approximate model.
Like the original MA scheme, the optimal input of (9) may be filtered to reduce the step-size and help stabilize the MA-GP scheme (del Rio-Chanona et al., 2019). The whole MA-GP scheme is presented in Algorithm 1.
In Algorithm 1 the hyperparameters are updated if *HypOpt* is true. *HypOpt* is a user-defined condition, which allows to update the hyperparameter. The extrema are to update the hyperparameters each iteration or never. The hyperparameter update is usually the computational bottleneck of the MA-GP algorithm. Since for large data sets it can be expected that the hyperparameter do not change much from one iteration to the next, it is reasonable to update the hyperparameters less frequent.

Algorithm 1: Basic MA-GP scheme

Initialisation: GP regression model $\mu^{\phi_p-\phi}_{\mathrm{GP}}$ and hyperparameter $\boldsymbol{\psi}$ optimization with data sets $\mathcal{D}$; MA-GP optimization at optimal operation point of the approximate model $\mathbf{u}_0$.
for $k = 0,1,\ldots$ **do**
  Solve modified optimization problem (9) ;
  Filter new operating point $\mathbf{u}_{k+1}$;
  Evaluate approximate model at new operating point $\mathbf{u}_{k+1}$;
  Obtain measurements of cost function $\boldsymbol{\phi}(\mathbf{u}_{k+1})$;
  Update the data sets $\mathcal{D}^{k+1}$ using the measured controlled and uncontrolled inputs as input and the difference of cost function of approximate model and plant as output;
  if *HypOpt* **then**
    Update hyperparameters $\hat{\boldsymbol{\psi}}$ with new data set $\mathcal{D}^{k+1}$;
  end
  Update GP regression terms $\mu^{\phi_p-\phi}_{\mathrm{GP}}$ with data set $\mathcal{D}^{k+1}$ and hyperparameters $\hat{\boldsymbol{\psi}}$;
end

3.2 Distributed learning and optimization

The distributed learning approach was first proposed in Andersson et al. (2020). Instead of learning the plant-model mismatch of the entire plant the plant-model mismatch of N agents, e.g. N wind turbines, is learned before solving the RTO problem (Fig. 1). Consequently, N GPs must be trained instead of one for the cost function. The advantage of the distributed learning approach is a better scaling for large wind farms and the need of a considerable smaller training set. Moreover, a priori knowledge about the interconnection between agents can be included, e.g. exclude control inputs of downwind turbines in the GP training of an upwind turbine. The additional computational work to train N GPs can be parallelized.

Instead of combining the N separate GP models in the optimization as done in Andersson et al. (2020) it is possible to use a distributed optimization approach to obtain the required centralized solution. The advantage of distributed optimization is that each agent can solve its own optimization problem in less time than the it takes the full optimization considering all wind turbines simultaneously.
In this article the global consensus alternating direction method of multipliers (ADMM) is used to solve iteratively the distributed optimization problem, which was recently applied to wind farm optimization by Annoni et al. (2018) The ADMM algorithm was originally developed in the 1970s (Gabay and Mercier, 1976) and it was revitalized by the review article by Boyd et al. (2011). The distributed optimization solves the same problem as described previously in section 3.1. Consequently, Algorithm 1 still applies. However, the GP models of each of the wind turbines must be updated individually.

4. CASE STUDY

4.1 Set-up

On a wind farm with three wind turbines in a row the proposed real time optimization architecture is tested. A detailed description of the models used can be found in Andersson and Imsland (2020). The wind turbines are modelled with the actuator disk theory (Burton et al., 2011), which couples the power and thrust coefficients. The wake, which causes the interaction between turbines, is modelled with the Gaussian model proposed by Bastankhah and Porté-Agel (2016). In this model the three-dimensional steady-state far wake velocity is assumed to be Gaussian distributed.
The control inputs of the wind farms are the yaw angles γ_i and thrust coefficients $C_{T,i}$ of the three turbines and the objective is to maximize the power production $Ptot = \sum_i P_i$. The control inputs are constrained by box constraints with

$$0 \leq C_{T,i} \leq 0.95, \quad \text{and} \quad 0^\circ \leq \gamma_i \leq 40^\circ. \tag{10}$$

In addition, the wind direction is considered as an uncontrolled input to the wind farm, which is included as an input in the GPs. The wind direction is sampled in the

[2] 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`

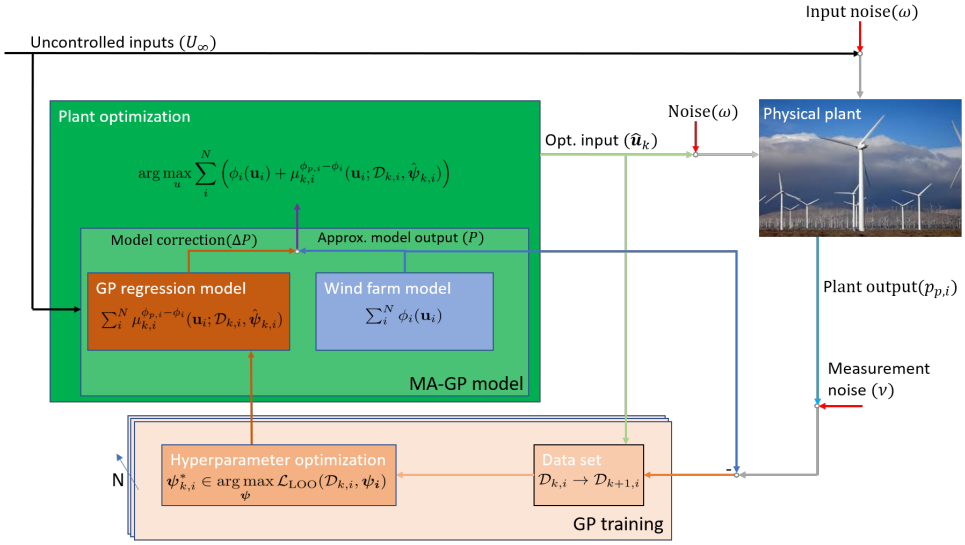


Fig. 1. The basic idea of the MA-GP scheme with distributed learning for a wind farm. The GP regression models create an input-output map of the control and uncontrolled inputs to the plant-model mismatch of each turbine. In the MA-GP model the GP regression models are used to correct the output of the approximate model. This MA-GP model is used in the optimization to compute optimal control inputs for the wind farm. The difference between the measured and estimated output of plant and model, respectively, are used to update the GP regression model. The uncontrolled and controlled inputs as well as the measured output are corrupted by noise. [2]

section ±20°, where for 0° the turbine row is aligned with the wind.

The plant is modelled with the same set-up, but several parameters in the actuator disc and flow model were changed (Andersson and Imsland, 2020). The relative error Θ between plant and model is 6%, where Θ is defined as

$$\Theta = \frac{P_p^* - P_m}{P_p^*}, \tag{11}$$

where P_p^* and P_m are the optimal power production of the plant and approximate model, respectively. Consequently, a plant-model mismatch exists, which should be corrected by the MA-GP method.

A training set with 300 training points is created using Latin hypercube sampling. The controlled inputs are perturbed with an insignificant amount of noise, while the uncontrolled input (wind direction) is perturbed with Gaussian noise with zero mean and standard deviation of 4°. The noisy observations are the power production P_i at each turbine, which are perturbed by Gaussian noise with zero mean and a standard deviation of 5000 W. Hence, the process has a relatively large input noise and a small output noise. The LOO log-likelihood is optimized using a multi-start optimization with 25 different initial values to find the initial hyperparameters of the GPs for the training set.

The root-mean square error of the difference between estimated power production by the model and actual power production by the plant is used to evaluate the initial model fit:

$$\Gamma = \sqrt{\frac{1}{M}\sum_i^M (P_m - P_p)}, \tag{12}$$

where M is the size of the test set. Hypercube sampling is used to create a test set with 10 000 data points.

Four different approaches are compared:

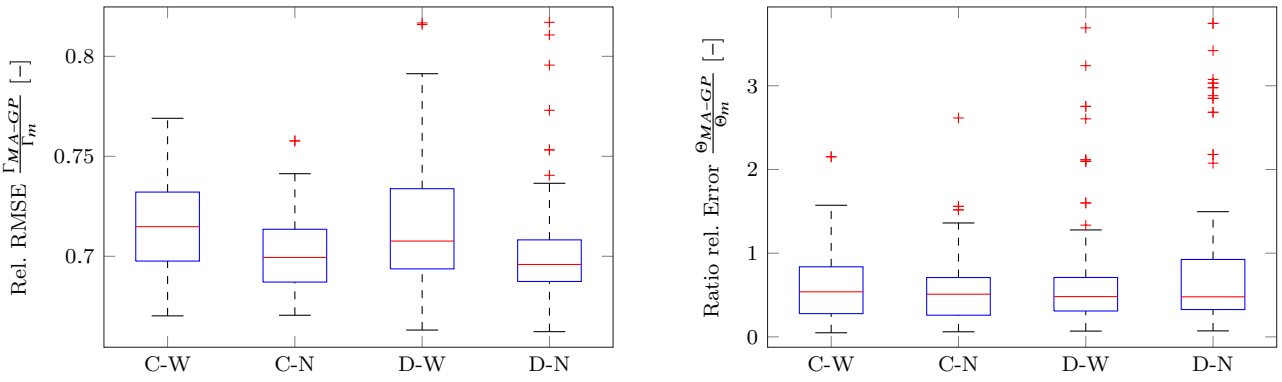
- MA-GP with distributed optimization without explicit consideration of noise (D-W),
- MA-GP with distributed optimization with explicit consideration of noise (D-N),
- MA-GP with centralized optimization without explicit consideration of noise (C-W),
- MA-GP with centralized optimization with explicit consideration of noise (C-N).

The method is evaluated on 100 Monte Carlo simulations. In each run the optimization is performed for 25 iterations. In each of these iterations the data set is updated with a new noisy data point and the hyperparameters of the GPs are re-optimized using the LOO log-likelihood.

4.2 Results

The result of the four different approaches are compared in Fig. 2. The root-mean-square error (RMSE) of the model fit of the MA-GP approach after the initial training (Fig. 2a) indicates an improvement in comparison to the approximate model. The median RMSE of all four MA-GP approaches is at about 70 % of the RMSE of the approximate model. The median and variance of the RMSE decreases slightly if the noise is considered explicitly. A slight improvement of the C-N and D-N approaches can also be observed in the optimization (Fig. 2b). The median error reduction compared to the approximate model of the C-W approach is about 46 % while the C-N approach achieves a reduction of about 49 %. The D-W and D-N approaches achieve a median error reduction of about 51.9 % and 52.1 %, respectively.

The distributed approaches tend to have outliers (about 10 % with the current set-up). This is the main difference between central and distributed approach. The tendency to outliers of the distributed approaches may have several causes:



(a) Relative RMSE of model fit after the initial training.

(b) Ratio of relative error after the optimization.

Fig. 2. The RMSE error of the model fit of the corrected model and the approximate model after the initial training and the ratio of the relative error Θ between the MA-GP model and the approximate model. A number smaller than one indicates superior performance of the MA-GP model.

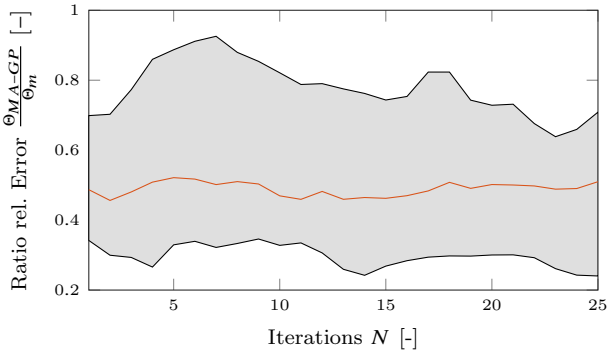


Fig. 3. The progression of the optimization error in the 25 iterations of the C-N approach. The red line gives the median and the edges of the gray region indicate the 25th and 75th percentiles.

- The problem is not convex while the ADMM algorithm is designed for convex problems.
- The ADMM algorithm was stopped after 100 iterations, which is not always sufficient to converge to the optimum.

The progression of the optimization error over the 25 iterations for the C-N approach is shown in Fig. 3. It can be observed that the median error does not decrease over the 25 iterations. However, the 75th percentiles decrease over the iterations. Filtering the results of the RTO iteration causes the smaller 75th percentile error at the beginning of the optimization. Similar observations can also be made for the other MA-GP approaches. It is not surprising that the improvement of the median error in the 25 iterations is negligible. Every new data point only adds a small amount of information to the initial data set of 300 points.
Robustness and a larger improvement can be achieved by increasing the number of data points in the training set. The C-W approach, for example, achieves a relative RMSE and optimization error after 25 iterations of 77 % and 64 %, respectively, using an initial training set of 150 data points. The same approach reaches a relative RMSE and optimization error of 71 % and 54 % using 300 data points. Moreover, the number of outliers is reduced with the larger data set.

Theoretically, the GP regression model could identify the power production of the individual wind turbines directly (assuming no prior knowledge of the model). Interestingly, including the approximate model in the MA-GP approach improves the performance. The RMSE of the identified model without using the approximate model is only slightly larger than when including the approximate model. In contrast, the optimization error increases significantly. Most of the time the algorithm is not able to converge to the actual plant optimum. It is an indication for local optima. Their existence could perhaps be reduced by constraints on the hyperparameter values. Nonetheless, this observation indicates more robustness in the optimization of the wind farm if the approximate model is included in the MA-GP algorithm.

5. CONCLUSION AND FUTURE WORK

In this article a new MA-GP scheme is proposed with uncertain inputs and outputs. This scheme considers noisy inputs explicitly to alleviate bias in the GP regression models. Moreover, the plant-model mismatch is distributively learnt and the wind farm distributively optimized. In the numerical test case, it was shown that the method to explicitly consider noisy inputs proposed in the article does only slightly improve the performance of the MA-GP approach in comparison to the ordinary MA-GP approach. It shows that the ordinary MA-GP handles noisy inputs relatively well and can be used for wind farm optimization. Still the in the article proposed approach to handle noisy inputs is simple to implement and comes without additional costs in the optimization and should, therefore, be considered as an attractive supplement.
The distributed learning improves the performance of the MA-GP approach significantly (Andersson et al., 2020) and enables the use of distributed optimization. In fact, the wind farm investigated in the numerical example is small, therefore distributed optimization is not required. However, in a larger wind farm distributed optimization may be simpler to use and may reduce computational time. In addition, the distributed approach may be well justified if a more physical approach to create the individual turbine models is used. Currently, the control and uncontrolled inputs of the wind farm are the inputs of each turbine model. However, the wind flow represents the actual physical connection between turbines, which should

be considered in future work as inputs of the individual turbine models in the MA-GP approach. This may reduce the inputs of the GP model and possibly allow to reduce the required training set to significantly reduce the computational demand for larger wind farms.

REFERENCES

Andersson, L.E. and Imsland, L. (2020). Real-time optimization of wind farms using modifier adaptation and machine learning. *Wind Energy Science Discussions*, 2020, 1–17.

Andersson, L.E., Bradford, E.C., and Imsland, L. (2020). Distributed learning and wind farm optimization with gaussian processes. In *American Control Conference (ACC), 2020 - [accepted]*.

Annoni, J., Seiler, P., Johnson, K., Fleming, P., and Gebraad, P. (2014). Evaluating wake models for wind farm control. In *2014 American Control Conference*, 2517–2523.

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

Barthelmie, R.J., Hansen, K.S., and Pryor, S.C. (2013). Meteorological controls on wind turbine wakes. *Proceedings of the IEEE*, 101(4), 1010–1019.

Barthelmie, R.J., Hansen, K., Frandsen, S.T., Rathmann, O., Schepers, J., Schlez, W., Phillips, J., Rados, K., Zervos, A., Politis, E., et al. (2009). Modelling and measuring flow and wind turbine wakes in large wind farms offshore. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 12(5), 431–444.

Bastankhah, M. and Porté-Agel, F. (2016). Experimental and theoretical study of wind turbine wakes in yawed conditions. *Journal of Fluid Mechanics*, 806, 506–541.

Boersma, S., Doekemeijer, B., Siniscalchi-Minna, S., and van Wingerden, J. (2019). A constrained wind farm controller providing secondary frequency regulation: An les study. *Renewable energy*, 134, 639–652.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1), 1–122.

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

Chen, C.Y. and Joseph, B. (1987). On-line optimization using a two-phase approach: An application study. *Industrial & engineering chemistry research*, 26(9), 1924–1930.

Ciri, U., Rotea, M.A., and Leonardi, S. (2017). Nested extremum seeking control for wind farm power optimization. In *American Control Conference (ACC), 2017*, 25–30. IEEE.

Corten, G. and Schaak, P. (2003). Heat and flux: Increase of wind farm production by reduction of the axial induction. In *Proceedings of the European Wind Energy Conference*.

Dallaire, P., Besse, C., and Chaib-Draa, B. (2009). Learning gaussian process models from uncertain data. In *International Conference on Neural Information Processing*, 433–440. Springer.

Darby, M.L., Nikolaou, M., Jones, J., and Nicholson, D. (2011). Rto: An overview and assessment of current practice. *Journal of Process Control*, 21(6), 874–884.

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

del Rio-Chanona, E.A., Graciano, J.E.A., Bradford, E., and Chachuat, B. (2019). Modifier-Adaptation Schemes Employing Gaussian Processes and Trust Regions for Real-Time Optimization. *IFAC-PapersOnLine*, 52(1), 52–57.

Doekemeijer, B.M., Van Wingerden, J.W., and Fleming, P.A. (2019). A tutorial on the synthesis and validation of a closed-loop wind farm controller using a steady-state surrogate model. In *2019 American Control Conference (ACC)*, 2825–2836. IEEE.

Duc, T., Coupiac, O., Girard, N., Giebel, G., and Göçmen, T. (2019). Local turbulence parameterization improves the jensen wake model and its implementation for power optimization of an operating wind farm. *Wind Energy Science*, 4(2), 287–302.

Fleming, P.A., Ning, A., Gebraad, P.M., and Dykes, K. (2016). Wind plant system engineering through optimization of layout and yaw control. *Wind Energy*, 19(2), 329–344.

Gabay, D. and Mercier, B. (1976). A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & mathematics with applications*, 2(1), 17–40.

Gebraad, P.M. and Van Wingerden, J. (2014). A control-oriented dynamic model for wakes in wind plants. In *Journal of Physics: Conference Series*, volume 524, 012186. IOP Publishing.

Gebraad, P., Teeuwisse, F., Van Wingerden, J., Fleming, P.A., Ruben, S., Marden, J., and 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), 95–114.

Goldberg, P.W., Williams, C.K., and Bishop, C.M. (1998). Regression with input-dependent noise: A gaussian process treatment. In *Advances in neural information processing systems*, 493–499.

Jensen, N.O. (1983). A note on wind generator interaction.

Johnson, K.E. and Fritsch, G. (2012). Assessment of extremum seeking control for wind farm energy production. *Wind Engineering*, 36(6), 701–715.

Knudsen, T., Bak, T., and Svenstrup, M. (2015). Survey of wind farm control - power and fatigue optimization. *Wind Energy*, 18(8), 1333–1351.

Marchetti, A., Chachuat, B., and Bonvin, D. (2009). Modifier-adaptation methodology for real-time optimization. *Industrial & engineering chemistry research*, 48(13), 6022–6033.

Marchetti, A.G., François, G., Faulwasser, T., and Bonvin, D. (2016). Modifier adaptation for real-time optimization – methods and applications. *Processes*, 4(4), 55.

Marden, J.R., Ruben, S.D., and Pao, L.Y. (2013). A model-free approach to wind farm control using game theoretic methods. *IEEE Transactions on Control Systems Technology*, 21(4), 1207–1214.

McHutchon, A. and Rasmussen, C.E. (2011). Gaussian process training with input noise. In *Advances in Neural Information Processing Systems*, 1341–1349.

Medici, D. (2005). *Experimental studies of wind turbine wakes: power optimisation and meandering*. Ph.D. thesis, KTH.

Munters, W. and Meyers, J. (2016). Effect of wind turbine response time on optimal dynamic induction control of wind farms. In *Journal of Physics: Conference Series*, volume 753, 052007. IOP Publishing.

Park, J., Kwon, S., and Law, K.H. (2013). Wind farm power maximization based on a cooperative static game approach. In *Active and Passive Smart Structures and Integrated Systems 2013*, volume 8688, 86880R. International Society for Optics and Photonics.

Rasmussen, C.E. and Williams, C.K. (2006). Gaussian processes for machine learning. *the MIT Press*, 2(3), 4.

Rotea, M.A. (2014). Dynamic programming framework for wind power maximization. *IFAC Proceedings Volumes*, 47(3), 3639–3644.

Snelson, E. and Ghahramani, Z. (2006). Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, 1257–1264.

Steinbuch, M., De Boer, W., Bosgra, O., Peeters, S., and Ploeg, J. (1988). Optimal control of wind power plants. *Journal of Wind Engineering and Industrial Aerodynamics*, 27(1-3), 237–246.

Sundararajan, S. and Keerthi, S.S. (2000). Predictive app roaches for choosing hyperparameters in gaussian processes. In *Advances in neural information processing systems*, 631–637.

Veers, P., Dykes, K., Lantz, E., Barth, S., and et al. (2019). Grand challenges in the science of wind energy. *Science*, 366(6464).

Wagenaar, J., Machielse, L., and Schepers, J. (2012). Controlling wind in ECN's scaled wind farm. *Proc. Europe Premier Wind Energy Event*, 685–694.