WIND ENERGY SCIENCE

Open Access

Iterative feedback tuning of wind turbine controllers

Edwin van Solingen, Sebastiaan Paul Mulders, and Jan-Willem van Wingerden

Delft Center for Systems and Control, Faculty of Mechanical Engineering, Delft University of Technology,
Mekelweg 2, 2628 CD Delft, the Netherlands

Correspondence to: Sebastiaan Paul Mulders (s.p.mulders@tudelft.nl)

Abstract. Traditionally, wind turbine controllers are designed using first principles or linearized or identified models. The aim of this paper is to show that with an automated, online, and model-free tuning strategy, wind turbine control performance can be significantly increased. For this purpose, iterative feedback tuning (IFT) is applied to two different turbine controllers: drivetrain damping and collective pitch control. The results, obtained by high-fidelity simulations using the NREL 5MW wind turbine, indicate significant performance improvements over baseline controllers, which were designed using classical loop-shaping techniques. It is concluded that iterative feedback tuning of turbine controllers has the potential to become a valuable tool to improve wind turbine performance.

1 Introduction

The control system plays a crucial role in the operation of wind turbines (van Kuik et al., 2016). Without properly tuned control loops, the turbine does not extract the maximum amount of energy from the wind and loads might not be mitigated optimally. Typically, wind turbine controllers are designed using linearized models obtained from wind turbine software packages (Bossanyi and Witcher, 2009). The linearized models approximate the nonlinear wind turbine dynamics in the vicinity of selected operating points for which the controller is designed. To obtain a controller that performs satisfactorily across the different operating conditions, gain-scheduling techniques are often used. When necessary, the controllers can be fine tuned by connecting them to the nonlinear wind turbine model using high-fidelity software packages.

Several factors detriment the controller performance when implemented on the actual turbine. First, the controller is designed upon the basis of models. This directly implies that there will be modeling errors and, hence, differences with the actual turbine. Second, every turbine will be different from the specifications due to, for instance, manufacturing errors and imperfections (van der Veen et al., 2013a). Third, due to environmental differences such as varying soil dynamics

throughout a wind plant, the dynamics of wind turbines vary per turbine (Lombardi et al., 2013; Abhinav and Saha, 2015). Finally, due to wear and tear, dynamics will change over time. All these factors impact the (controller) performance of wind turbines and should be addressed during commissioning and periodically during the lifetime of a wind turbine.

A wind turbine manufacturer has several opportunities to overcome the aforementioned issues. One of these is by applying system identification techniques (Hjalmarsson, 2005; van der Veen et al., 2013a, b; La Cava et al., 2016). With system identification, the dynamics of a wind turbine subsystem are obtained by exciting a wind turbine input and measuring the response thereof, which often yields a more realistic model. By using the identified models as a basis for the control design, performance can be increased. The drawback of this approach is that it might be time consuming to obtain the dynamics for all operating conditions, after which the controller needs to be redesigned.

In the past decade, research has been conducted on developing load-reducing controllers that "learn" the optimal controller settings online (van Wingerden et al., 2011; Houtzager et al., 2013; Navalkar et al., 2014, 2015; Xiao et al., 2016). These controllers are typically scheduled on basis functions and thereby mainly target the periodic wind turbine loading. By minimizing a cost function with respect to the con-

troller parameters, the optimal (linear) combination of the basis functions can be obtained online. The controllers have been successfully demonstrated in both simulation studies as well as experimental wind tunnel testing. However, the main drawback of the majority of the approaches is that the resulting controller operates in feedforward. This means that mainly the deterministic loads are targeted, while stochastic loads remain roughly unaffected.

Another strategy is to tune the controllers offline by using the previously mentioned models and use this as a starting point for an automated online tuning algorithm (which can be run during commissioning and periodically over the turbine lifetime for instance). One such algorithm is given by iterative feedback tuning (IFT) (Hjalmarsson et al., 1998; Hjalmarsson, 2002). With IFT, the parameters of a fixed-structure controller are iteratively optimized by carrying out two or more experiments on the closed-loop system, with which estimates of the gradient with respect to the controller parameters are obtained. Iteratively updating the controller parameters by using the gradient estimates then minimizes a user-defined cost function. The key advantage of IFT is that detailed knowledge of the (wind turbine) system is not needed, while the main requirement is that the initial closed-loop system is stable, implying that it can be directly used for optimization of wind turbine controllers. IFT has been successfully applied to various application fields, including mechatronics (Al Mamun et al., 2007; Heertjes et al., 2016), robotics (Liu et al., 2011), process industry (Lequin, 1997), and recently wind turbines (Navalkar and van Wingerden, 2015).

In this paper, IFT is used to optimize the performance of an active drivetrain damper and of reference tracking using collective pitch control (CPC). The main contribution is therefore to show the (practical) application of IFT to existing wind turbines. This paper also contributes in showing how IFT can be applied to systems that have multiple controllers in the loop. Moreover, it is shown that IFT can be applied to systems that have a reference input with a static offset by performing an additional experiment. Finally, the impact of several practical considerations is shown, including the experiment length, signal-to-noise ratio, and convergence speed.

This paper is organized as follows. In the next section, the details of the IFT algorithm and the closed-loop experiments that are required to obtain gradient estimates are given. Subsequently, in Sect. 3, short descriptions of the NREL 5MW wind turbine and the high-fidelity simulation environment are given. An overview of the control system is also provided. In Sects. 4 and 5, the IFT algorithm is used to optimize the performance of an active drivetrain damper and the reference tracking of the rotor speed using CPC. Finally, conclusions are drawn in Sect. 6.

2   Iterative feedback tuning theory

This section introduces the IFT method. First, the main details of IFT are given in Sect. 2.1, followed by the analysis of IFT for systems with a reference input containing an offset in Sect. 2.3.

2.1   IFT introduction

The basic rationale of IFT (Hjalmarsson et al., 1998; Hjalmarsson, 2002) is to minimize a cost function, given for instance by the quadratic expression

$$J(\rho) = \frac{1}{2N} \sum_{k=1}^{N} E\left[(y(k,\rho) - r(k))^2 + \lambda u(k,\rho)^2 \right] \qquad (1)$$

in an iterative manner. The cost function $J(\rho)$ in Eq. (1) depends on the (tunable) controller parameters ρ, the squared error between the output $y(k,\rho)$ and the reference input $r(k)$ (possibly prefiltered with a system reference model), and the squared input signal of the system $u(k,\rho)$, where k indicates the time instance. The cost function is divided by 2 times the number of data samples N, and involves the expectation $E[\cdot]$ due to noise. Using a gradient search of the type

$$\rho_{i+1} = \rho_i - \gamma_i R_i^{-1} \frac{\partial J}{\partial \rho}(\rho_i), \qquad (2)$$

where i is the iteration number, $\partial J / \partial \rho(\rho_i)$ the gradient of the cost function Eq. (1), R_i a positive definite matrix (e.g., the Hessian of Eq. 1), an unconstrained optimization problem is obtained. Note that the objective function can be non-convex, and will in such a case converge to a local minimum.

It is clear that minimizing Eq. (1) boils down to computing the gradient $\partial J / \partial \rho(\rho_i)$ and Hessian R_i at every iteration. Previous studies (e.g., refer to Hjalmarsson et al., 1998; Hjalmarsson, 2002) have shown that these quantities can be obtained from the closed-loop system by conducting a number of experiments. To see this, first consider the partial derivative of the cost function $J(\rho)$ with respect to the controller parameter vector ρ:

$$\frac{\partial J}{\partial \rho}(\rho) = \frac{1}{N} \sum_{k=1}^{N} E\left[(y(k,\rho) - r(k)) \frac{\partial y}{\partial \rho}(k,\rho) \right.$$
$$\left. + \lambda u(k,\rho) \frac{\partial u}{\partial \rho}(k,\rho) \right], \qquad (3)$$

which involves the signals $\partial y / \partial \rho(k,\rho)$ and $\partial u / \partial \rho(k,\rho)$. Thus, in each iteration the following signals are required:

1. the signals $r(k)$, $y(k,\rho_i)$, and $u(k,\rho_i)$;

2. the gradients $\partial y / \partial \rho(k,\rho_i)$ and $\partial u / \partial \rho(k,\rho_i)$;

3. unbiased estimates of the products $(y(k,\rho_i) - r(k))\partial y / \partial \rho(k,\rho_i)$ and $u(k,\rho_i)\partial u / \partial \rho(k,\rho_i)$.

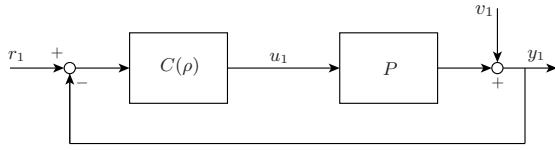**Figure 1.** Block diagram of a closed-loop system.

Figure 2. Block diagram of the closed-loop gradient experiment.

The signals of the first requirement can be obtained from running a closed-loop experiment as in Fig. 1.

Obtaining the signals of the latter two requirements is slightly more involved. In order to derive the required gradients and unbiased estimates, consider the block scheme of the closed-loop system in Fig. 1. From the block scheme, it is readily observed that

$$y_1(k, \rho) = P u_1(k, \rho) + v_1(k), \tag{4}$$

$$u_1(k, \rho) = C(\rho)(r_1(k) - y_1(k, \rho)), \tag{5}$$

where P is the plant, $C(\rho)$ the fixed-structure parameterized controller, and v_1 the measurement noise. Taking the partial derivatives of the latter signals with respect to the controller parameters ρ gives

$$\frac{\partial y_1}{\partial \rho}(k, \rho) = P \frac{\partial u_1}{\partial \rho}(k, \rho), \tag{6}$$

$$\frac{\partial u_1}{\partial \rho}(k, \rho) = \frac{\partial C}{\partial \rho}(\rho)(r_1(k) - y_1(k, \rho))$$
$$- C(\rho)\frac{\partial y_1}{\partial \rho}(k, \rho). \tag{7}$$

Substitution of Eq. (7) in Eq. (6) yields

$$\frac{\partial y_1}{\partial \rho}(k, \rho) = (I + PC(\rho))^{-1} P \frac{\partial C}{\partial \rho}(\rho)(r_1(k) - y_1(k, \rho))$$
$$= S(\rho) P \frac{\partial C}{\partial \rho}(\rho)(r_1(k) - y_1(k, \rho)), \tag{8}$$

where $S(\rho) = (I + PC(\rho))^{-1}$ is the sensitivity function. The gradient in Eq. (8) can be obtained by injecting $r_1(k) - y_1(k, \rho)$ at the process input $u_2(k, \rho)$ according to Fig. 2. This experiment is the so-called *gradient* experiment. Notice that Fig. 2 includes a scaled injection signal with a factor F at the process input as well as a factor $1/F$ at the output, which will become clear in the following paragraphs; for now assume $F = 1$. It should also be noted that the gradient $\widehat{\partial y}/\partial \rho(k, \rho)$ obtained through Fig. 2 is an estimate of Eq. (8) because it is contaminated with noise $v_2(k)$, i.e.,

$$\frac{\widehat{\partial y}}{\partial \rho}(k, \rho) = S(\rho) P \frac{\partial C}{\partial \rho}(\rho)(r_1(k) - y_1(k, \rho))$$
$$+ S(\rho)\frac{\partial C}{\partial \rho}(\rho)v_2. \tag{9}$$

Finally, note that the subscript indicates the experiment number and should not be confused with the iteration number i.
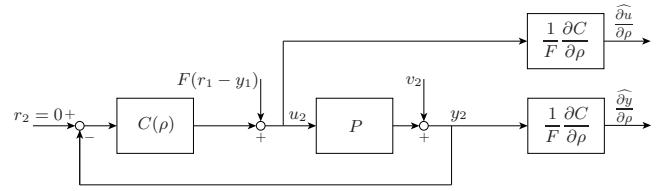
Performing the gradient experiment as in Fig. 2 avoids the need of an inverse of the controller, which will become clear in the next section.

The input gradient signal $\widehat{\partial u}/\partial \rho(k, \rho)$ can be obtained in a similar way (refer to Hjalmarsson et al., 1998, for a derivation). It can be derived that

$$\frac{\widehat{\partial u}}{\partial \rho}(k, \rho) = S(\rho)\frac{\partial C}{\partial \rho}(\rho)(r_1(k) - y_1(k, \rho))$$
$$- S(\rho)C(\rho)\frac{\partial C}{\partial \rho}(\rho)v_2, \tag{10}$$

which means that the gradient can be obtained from Fig. 2 by multiplying the process input $u_2(k, \rho)$ with $\partial C/\partial \rho$ (or by $1/F \cdot \partial C/\partial \rho$ when the scaling factor F is applied). Again, this gradient is an estimate because it is contaminated with $v_2(k)$ and is therefore denoted by $\widehat{\partial u}/\partial \rho(k, \rho_i)$.

Under some mild assumptions of the noise properties (zero-mean noise and the noise should be uncorrelated in each experiment), it can be shown (Hjalmarsson, 2002) that $(y_1(k, \rho_i) - r_1(k))\widehat{\partial y}/\partial \rho(k, \rho_i)$ and $u_1(k, \rho_i)\widehat{\partial u}/\partial \rho(k, \rho_i)$ are unbiased estimates.

The matrix R in Eq. (2) is often replaced by an approximation of the Hessian. When $y(k, \rho) - r(k)$ is small, the Gauss-Newton direction

$$R = \frac{1}{N}\sum_{k=1}^{N}\left[\frac{\widehat{\partial y}}{\partial \rho}(k, \rho)\frac{\widehat{\partial y}}{\partial \rho}^{\mathrm{T}}(k, \rho) + \lambda \frac{\widehat{\partial u}}{\partial \rho}(k, \rho)\frac{\widehat{\partial u}}{\partial \rho}^{\mathrm{T}}(k, \rho)\right] \tag{11}$$

can be a suitable choice (Hjalmarsson, 2002). However, the approximated Hessian in Eq. (11) will be biased because of the disturbances. Typically, this will slow down the convergence of the algorithm. In Solari and Gevers (2004), it is shown that an unbiased estimate of the Hessian can be obtained on the basis of two additional closed-loop experiments. In this paper, however, it was found that the approximated Hessian Eq. (11) performed sufficiently well and was therefore the preferred choice.

2.2 Improving the signal-to-noise ratio

In order to improve the signal-to-noise ratio in the gradient experiment, it is suggested to scale the injection signal with a factor F (Hjalmarsson, 2002) as shown in Fig. 2. Consequently, the output $\widehat{\partial y}/\partial \rho(k, \rho)$ should be divided by the same scaling factor. For a discussion on the optimal choice of F, refer to Hjalmarsson and Gevers (1997).

In the later sections, the gain F needs to not only be used to improve the signal-to-noise ratio but also to appropriately scale the input injection signal $r_1(k) - y_1(k, \rho)$. In certain cases, the reference $r(k)$ and/or measured signals $y(k, \rho)$ can be orders of magnitude larger (or smaller) than the process input $u(k, \rho)$, which means that injection of $r_1(k) - y_1(k, \rho)$ with large amplitude would lead to an undesired input. Hence, F can be used to scale this signal to the desired input level.

2.3 IFT for systems with offset in reference inputs

In the previous subsection, the controller C was subject to IFT, where it was assumed that $r(k)$ could be set to zero. In this subsection, the details of iteratively optimizing controller C when operating at a non-zero reference input $r(k)$ are analyzed. Specifically, the details in this part are tailored to the case of optimizing the CPC, for instance. In the CPC loop, the generator speed is held close to the rated generator speed. Thus, the reference input can be used to generate a step response. Then, the reference $r(k)$ can be written as $r(k) = r_o + r_n(k)$, where $r_o \neq 0$ is the constant offset (e.g., the rated generator speed) and r_n the reference step. Similar to the previous section, the offset r_o will contaminate the gradient signals and therefore a third experiment is required to remove this contamination.

The experiments to obtain unbiased gradient estimates are as follows (Hjalmarsson, 2002). First, an experiment is carried out where a step change $r_1(k) = r_o + r_n(k)$ is applied to the closed-loop system according to Fig. 1. Recording the process input $u_1(k, \rho)$ and the output $y_1(k, \rho)$,

$$y_1(k, \rho) = S(\rho)\big(PC(\rho)(r_o + r_n(k)) + v_1(k)\big), \quad (12)$$
$$u_1(k, \rho) = S(\rho)C(\rho)\big(r_o + r_n(k) - v_1(k)\big), \quad (13)$$

gives the first set of signals. Second, an experiment identical to the first experiment is carried out, except that there is not a step change in the reference; hence, $r_2(k) = r_0$. This yields the second set of signals:

$$y_2(k, \rho) = S(\rho)\big(PC(\rho)r_o + v_2(k)\big), \quad (14)$$
$$u_2(k, \rho) = S(\rho)C(\rho)\big(r_o - v_2(k)\big). \quad (15)$$

The third experiment is the experiment where the recorded signals of the first and second experiments are used to obtain an estimate of the gradient. The control configuration for the gradient experiment is shown in Fig. 3, where it is seen that the reference input equals r_o. For this configuration, the gradient estimate signals read (also see Appendix A)

$$\widehat{\frac{\partial y}{\partial \rho}}(k, \rho) = \frac{1}{F}\frac{\partial C}{\partial \rho}(\rho)\bigg(S(\rho)\big[PF(r_1(k) - y_1(k, \rho))$$
$$+ PC(\rho)r_o + v_3(k)\big] - y_2(k, \rho)\bigg), \quad (16)$$

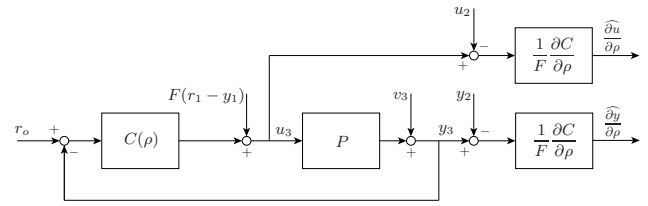**Figure 3.** Block diagram of modified closed-loop gradient experiment for reference inputs with offsets.

$$\widehat{\frac{\partial u}{\partial \rho}}(k, \rho) = \frac{1}{F}\frac{\partial C}{\partial \rho}(\rho)\bigg(S(\rho)\big[F(r_1(k) - y_1(k, \rho))$$
$$+ C(\rho)r_o - C(\rho)v_3(k)\big] - u_2(k, \rho)\bigg), \quad (17)$$

from which it can be seen that both noise and the reference offset r_o perturb the estimate. Fortunately, the offset r_o in the gradient estimates can be removed by using Eqs. (14) and (15) during the gradient experiment (see Fig. 3).

It is noted that the first and second experiments are nearly identical, but they are both required in order for $(y_1(k, \rho_i) - r_1(k))\widehat{\partial y}/\partial \rho(k, \rho_i)$ and $u_1(k, \rho_i)\widehat{\partial u}/\partial \rho(k, \rho_i)$ to be unbiased (Hjalmarsson, 2002) under the same assumptions as before.

2.4 IFT for systems with multiple controllers

The IFT method discussed in the previous part cannot directly be applied to control systems where multiple (decoupled) controllers are working on the same input signal. Consider in this case, for example, the torque controller and the drivetrain controller. The torque controller regulates the generator torque in such a way that the rotor speed provides maximum power extraction from the wind. At the same time, a drivetrain damper adds a small torque ripple on the regulated generator torque in order to reduce drivetrain oscillations (which will be further explained in Sect. 4). Thus, in such a case the reference signal $r(k)$ in Fig. 2 cannot be set to zero. Moreover, the torque and damping controllers might have some interaction such that biased estimates are obtained. In order to apply IFT to this kind of control system, it is shown that a similar procedure as for the case with reference offsets can be used.

The basic control system analyzed in this section is shown in Fig. 4. The control system consists of controllers C_1 and $C_2(\rho)$, where the latter is subject to optimization. Controller C_1 works on the error between a reference $r(k)$ and the measured output $y(k, \rho)$, while $C_2(\rho)$ works on the output $y(k, \rho)$ filtered by $\mathcal{H}(s)$. The role of the high-pass filter will become clear at a later point. The controller $C_2(\rho)$ is implemented with positive feedback without loss of generality.

The first experiment is identical to the single controller case as described in the previous paragraphs: the closed-loop
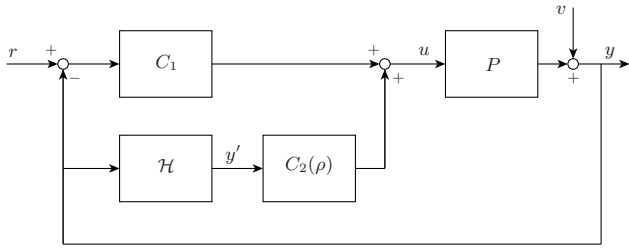
Figure 4. Block diagram of the closed-loop system, which involves multiple controllers (representing the scenario of torque control and drivetrain damping).

Figure 5. Block diagram of the gradient experiment for the multiple controller scenario. The controller C_1 is fixed and controller $C_2(\rho)$ is subject to IFT. The signals $\mathcal{H}y_1(k, \rho)$, $y_2(k, \rho)$, and $u_2(k, \rho)$ are obtained from two other experiments and are required to obtain unbiased gradient estimates.

system in Fig. 4 is used to obtain the following signals:

$$y_1(k, \rho) = S(\rho)\big(PC_1r_1(k) + v_1(k)\big), \tag{18}$$

$$u_1(k, \rho) = S(\rho)\big(C_1r_1(k) - C_1v_1(k) + C_2(\rho)\mathcal{H}v_1(k)\big), \tag{19}$$

where $S = (I + PC_1 - PC_2(\rho)\mathcal{H})^{-1}$ is the sensitivity function. The signals in Eqs. (18)–(19) determine the cost function given in Eq. (1).

The gradient signal related to the output $y_1(k, \rho)$ is obtained as follows. First note that

$$y(k, \rho) = Pu(k, \rho) + v(k) \tag{20}$$

such that the partial derivative thereof is given by

$$\frac{\partial y}{\partial \rho}(k, \rho) = P\frac{\partial u}{\partial \rho}(k, \rho). \tag{21}$$

Then, note that the process input $u(k, \rho)$ in Fig. 4,

$$u(k, \rho) = C_1(r(k) - y(k, \rho)) + C_2(\rho))\mathcal{H}y(k, \rho), \tag{22}$$

has partial derivative

$$\frac{\partial u}{\partial \rho}(k, \rho) = -C_1\frac{\partial y}{\partial \rho}(k, \rho) + \frac{\partial C_2}{\partial \rho}\mathcal{H}y(k, \rho)$$
$$+ C_2(\rho)\mathcal{H}\frac{\partial y}{\partial \rho}(k, \rho). \tag{23}$$

Substituting Eq. (23) in Eq. (21) and manipulating it yields

$$\frac{\partial y}{\partial \rho}(k, \rho) = S(\rho)P\frac{\partial C_2}{\partial \rho}\mathcal{H}y(k, \rho). \tag{24}$$

Similarly, the gradient related to the input can be found to be (see Appendix B)

$$\frac{\partial u}{\partial \rho}(k, \rho) = S(\rho)\frac{\partial C_2}{\partial \rho}(\rho)\mathcal{H}y(k, \rho). \tag{25}$$

Thus, the signals in Eqs. (24)–(25) can be obtained by injecting the high-pass filtered output $\mathcal{H}y(k, \rho)$ or simply $y'(k, \rho)$
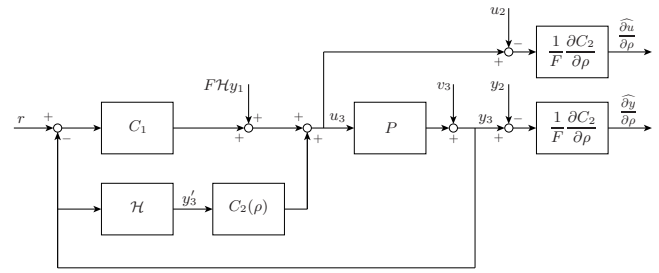
at the process input $u(k, \rho)$ in a gradient experiment according to Fig. 5. Doing so, one then obtains the following gradient estimates:

$$\widehat{\frac{\partial y}{\partial \rho}}(k, \rho) = \frac{1}{F}\frac{\partial C_2}{\partial \rho}\Big(S(\rho)\big[PC_1r(k) + FP\mathcal{H}y_1(k, \rho)$$
$$+ v_3(k)\big] - y_2(k, \rho)\Big), \tag{26}$$

$$\widehat{\frac{\partial u}{\partial \rho}}(k, \rho) = \frac{1}{F}\frac{\partial C_2}{\partial \rho}\Big(S(\rho)\big[C_1r(k) - C_1v_3(k) + C_2(\rho)\mathcal{H}v_3(k)$$
$$+ F\mathcal{H}y_1(k, \rho)\big] - u_2(k, \rho)\Big). \tag{27}$$

Compare the equations above with Eqs. (24)–(25) and it can be observed that the estimates are biased. Now, by running an experiment identical to the first experiment, one acquires $y_2(k, \rho)$ and $u_2(k, \rho)$ identical to Eqs. (18)–(19). Subtracting these, as shown in Fig. 5, cancels the undesired terms in the gradient signals in Eqs. (26)–(27) such that the desired gradient is obtained, which is only perturbed by noise terms.

The high-pass filter $\mathcal{H}$ is incorporated for practical consideration. In the gradient experiment, the signal $F\mathcal{H}y_1(k, \rho)$ is injected into the system. In the case of drivetrain damping, this would mean injecting the measured generator speed $y_1(k, \rho)$ with a scaling factor F. As the generator speed during operating is larger than zero, this would imply inserting a step change in the demanded generator torque. High-pass filtering the measured speed causes the injection of a signal that varies around zero.

2.5 Including cost function weights

This section shortly discusses weighting filters in the cost function. The cost function in Eq. (1) can be modified to include time and frequency weights. First, consider the following so-called zero-weighting mask of the cost function (Lequin et al., 1999, 2003)

$$J(\rho) = \frac{1}{2N}\sum_{k=N_0}^{N}E\Big[(y(k, \rho) - r(k))^2 + \lambda u(k, \rho)^2\Big], \tag{28}$$

where the index of the summation starts at $N_0 > 1$. The motivation for this mask, often used for step response tuning, is as follows. Typically, the main objective of step response tuning is to move the system quickly from one point to another. The settling time is an important parameter in this context. Without zero-weighting mask, the controller is tuned such that the reference is as closely matched as possible, while in practice one does not care too much about the trajectory to the new reference value, as long as the overshoot is not too large. By zero weighting the transient trajectory in the cost function, the algorithm tries to achieve a fast settling time.

The cost function can also incorporate frequency weights L_y and L_u

$$J(\rho) = \frac{1}{2N} \sum_{k=N_0}^{N} E\big[(L_y(y(k, \rho) - r(k)))^2 + \lambda(L_u u(k, \rho))^2\big], \tag{29}$$

which filter the error $y(k, \rho) - r(k)$ and input $u(k)$ accordingly. With the frequency weights, one can emphasize or suppress frequency bands in the cost function. For example, in the case of drivetrain damping, the measured output signal will not only be composed of the drivetrain resonance frequency but also of many other disturbances with various frequency components. Thus, filtering the drivetrain frequencies in the cost function makes sure the controller will focus on the filtered frequencies.

The cost function Eq. (29) requires filtering of $y(k, \rho) - r(k)$ and $u(k, \rho)$ with L_y and L_u, respectively. Moreover, the derivative of Eq. (29) with respect to the control parameters

$$\frac{\partial J}{\partial \rho}(\rho) = \frac{1}{N} \sum_{k=1}^{N} E\left[L_y(y(k, \rho) - r(k))\frac{\partial y}{\partial \rho}(k, \rho) + \lambda L_u u(k, \rho)\frac{\partial u}{\partial \rho}(k, \rho)\right] \tag{30}$$

then involves the filters L_y and L_u as well. Thus, this also requires the gradient signals to be passed through the frequency filters.

2.6 Stability and convergence

The iterative optimization of the controller parameters ρ described in the previous paragraphs can have robustness issues. This is caused by the fact that, typically, there are no guarantees that the controller remains stable during the iteration. For that reason a number of articles (Heertjes et al., 2016; Procházka et al., 2005; de Bruyne and Kammer, 1999; Veres and Hjalmarsson, 2002) have appeared on the subject of including stability constraints for the optimization procedure. These algorithms are not considered here because no stability issues are encountered in this study. However, a safety measure that can be taken is lowering the parameter update step size γ_i, but for this a trade-off between stability

Table 1. NREL 5MW wind turbine description (Bossanyi and Witcher, 2009).

Description	Value
Rated power	5 MW
Number of blades	3
Rotor diameter	126 m
Orientation	Upwind
Hub height	90 m
Gearbox ratio	97
Rated wind speed	11.3 m s^{-1}
Rated rotor speed	12.1 rpm
Rated generator torque	43.093 kNm
Drivetrain natural frequency	10.49 rad s^{-1}

and convergence rate has to be made. A technique to improve convergence is to adjust the step size γ_i after each iteration, based on a norm of the cost function gradient $\partial J / \partial \rho(\rho_i)$.

3 Wind turbine and simulation environment

In this study, the NREL 5MW reference wind turbine model is used. The model does not represent an existing wind turbine but is considered to reflect typical commercial wind turbines of similar ratings. The turbine has three blades in an upwind rotor configuration, a rotor diameter of 126 m, and it reaches the rated power output of 5 MW at a wind speed of 12.1 m s^{-1}. A summary of the most important parameters is listed in Table 1. The wind turbine is a variable-speed variable-pitch machine, such that the rotor speed in below-rated wind is regulated by means of the generator torque, and in above-rated wind by collectively pitching the blades. The natural frequency and damping ratio of the drivetrain dynamics, which are important for the drivetrain damper that will be designed in Sect. 4, are $\omega_r = 10.49$ rad s^{-1} and $\zeta = 0.0217$, respectively.

The relevant control scheme for the scope of this study is given in Fig. 6. The measured generator speed Ω_{gen} forms the input for the controllers. The torque controller provides a demanded generator torque set point T_{trq} to regulate the rotor speed in below-rated wind, and drivetrain damping can be achieved by superimposing T_{trq} with a small torque ripple T_{dtd}, such that the final demanded generator torque is given by T_{gen}. In above-rated conditions, the generator speed is held constant at $\Omega_{ref} = \Omega_{rated}$ by means of the collective blade pitch set point θ generated by the CPC. The drivetrain damper and the CPC will be optimized using IFT in Sects. 4 and 5.

The software package GH Bladed 4.20 (Garrad Hassan, 2015; Garrad Hassan & Partners Ltd, 2017a, b), which is a certified and widely used wind turbine design software package in the industry, is used to simulate the behavior of the wind turbine in response to a supplied wind field. To do so, the structural model of the turbine is modeled by a multi-
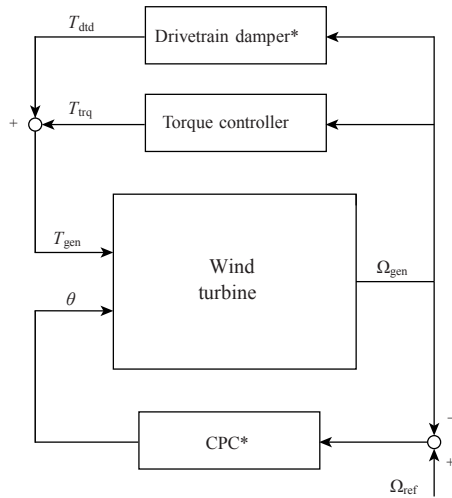
Figure 6. Overview of the relevant controller configuration in this study. The starred controllers will be the subject for IFT.

body approach combined with a modal representation of the flexible components. The rotor aerodynamics are modeled by combining blade and momentum theory. Initial (baseline) controllers will be constructed using classical loop shaping techniques on linear models of the different subsystems, which are obtained from Bladed using the built in linearization tool. The controllers will be subsequently implemented in the NREL 5MW Bladed wind turbine model and optimized using the IFT algorithm, as Bladed allows for testing of new control algorithms by compiling an external controller designed in MATLAB Simulink (Mathworks, 2013), for example, to a dynamic-like library (DLL) file (Houtzager, 2013). The DLL file is then used during the calculations to obtain the closed-loop dynamics of the wind turbine. The controllers are discretized using the Tustin approximation and are operated at a sampling time of 0.01 s. Note that some of the considered subsystem dynamics are active in a lower frequency range, such that a reduced sampling time would be sufficient. However, in this work all controllers are designed to operate at the sampling frequency, as commercial turbines typically operate at this frequency. The data in the log files are recorded with a sampling time of 0.05 s. The interested reader is referred to the theory manual of Bladed (Garrad Hassan & Partners Ltd, 2017b) for details on the calculation methods.

4 IFT of a drivetrain damper

Wind turbines that have a geared drivetrain are known to have a lightly damped drivetrain mode. Subjected to turbulent wind, the rotor speed will vary despite the speed regulation by torque control and CPC. The rotor speed variations cause the drivetrain mode to be excited, which can lead to oscillations in the drivetrain. In order to prevent this, a driv-

etrain damping controller is typically included in the control system (e.g., see Fig. 6). This controller adds a small torque ripple at the drivetrain frequency (Bossanyi, 2000) to the demanded generator torque of the torque controller. Doing so will dramatically reduce the drivetrain oscillations. Several studies in the past have considered the design of drivetrain damping control, e.g., refer to Bossanyi (2000); Dixit and Suryanarayanan (2005); Fleming et al. (2011, 2013); Wright et al. (2011).

In this section, the IFT algorithm (see. Sect. 2) is applied to the optimization of the drivetrain damper parameters. In the next paragraph, the controller parameterization is given, as well as the classical design approach. Then, a parameter study is carried out to visualize the cost function. Subsequently, the controller is iteratively optimized for realistic loading scenarios as well as for different algorithm settings. Finally, the results of this case study are discussed.

4.1 Baseline damping controller design

Typically, the drivetrain controller is chosen to be a bandpass filter of the form (Bossanyi, 2000; Bossanyi and Witcher, 2009)

$$C_{dtd}(s) = \frac{2K\omega\zeta s}{s^2 + 2\zeta\omega s + \omega^2}, \tag{31}$$

where K is the bandpass gain, ω is the drivetrain frequency, and ζ influences the damping. With a (linearized) model of the drivetrain dynamics G_{dtd}, this bandpass filter can be tuned to damp the drivetrain oscillations. To this end, a linearized model of the drivetrain dynamics is obtained from Bladed using the built-in linearization tool. With this model, the drivetrain damper is designed using classical loop-shaping techniques (Skogestad and Postlethwaite, 2006). The resulting Bode diagrams of the drivetrain dynamics, the open loop (controller times drivetrain dynamics), and the closed loop are shown in Fig. 7. From the open-loop dynamics shown in the Bode diagram, it is recognized that the bandpass gain K can theoretically be increased infinitely (i.e., because the phase plot never crosses $-180°$); however, in practice the controller bandwidth is limited by actuator and communication constraints and higher-order unmodeled dynamics. Hence, during the iterative optimization, the controller cannot become unstable (for positive values of the controller parameters). The bandpass filter that is used in Fig. 7 has $K = 2500$, $\zeta = 0.3$, and $\omega = \omega_d = 10.49 \, \text{rad s}^{-1}$ and is considered as the baseline controller. This filter has been experimentally verified to yield satisfactory performance.

4.2 Parameter study for drivetrain damping

Before the IFT algorithm was implemented, first a parameter study using the linearized drivetrain dynamics was carried out. In order to gain understanding of the optimization
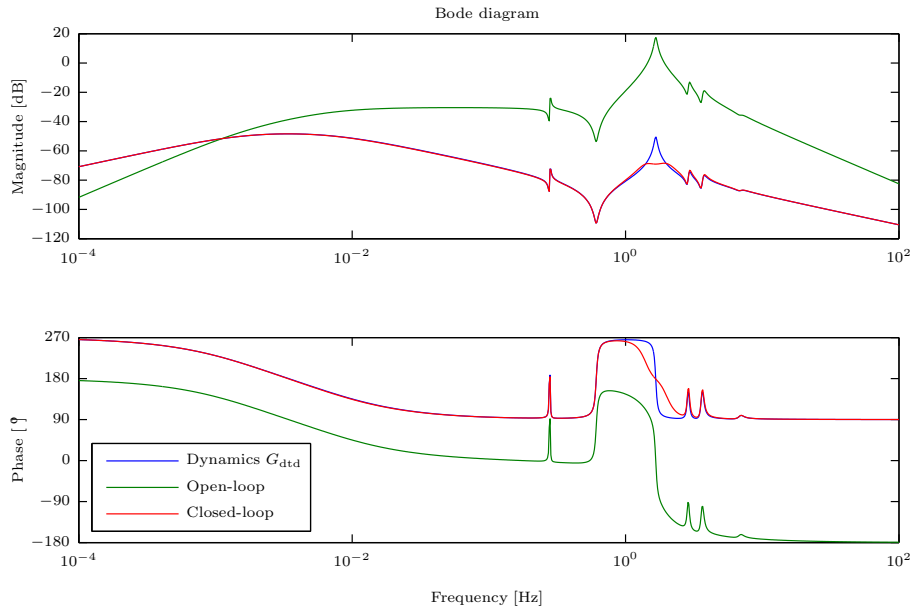
Figure 7. Bode diagrams of the drivetrain dynamics, the open-loop (controller times drivetrain dynamics), and the closed-loop system.

problem, the closed-loop system, including the drivetrain dynamics G_{dtd} and the bandpass filter C_{dtd}, is simulated for a range of parameters (K, ω, ζ). A disturbance signal at the drivetrain frequency is injected at the output. The cost function is taken as Eq. (1), i.e., both the output and the input are weighted. The cost function for combinations of K and ω is shown in Fig. 8. The results indicate that the cost function has a large area where it is almost optimal and which becomes wider for increasing ζ. This means that for this problem many combinations of K and ω exist that give almost identical results. Moreover, this also indicates that the parameters are likely to quickly converge to almost minimum cost values, but they will slowly converge to the optimum value. It also shows that the baseline damping filter with $\zeta = 0.3$ could be improved (at least for the simulation case) by decreasing the damping ζ of the filter. Finally, the cost function plot shows that the phase of the controlled system can be adjusted by increasing or decreasing the frequency ω and at the same time increasing or decreasing the bandpass gain K to maintain practically the same compensation performance.

4.3   IFT of drivetrain damping

The input of the drivetrain damping controller Eq. (31) is taken to be the measured generator speed. The generator speed signal consists of many frequency components arising from the interaction between the wind turbine system and the wind. For drivetrain damping, one is only interested in the frequencies close to the drivetrain frequency. Hence, the controller scheme is taken to be identical to Fig. 4, where C_1 is the torque controller and C_2 is the drivetrain damper. The input to the system is $u = T_{gen}$ and the output is $y = \Omega_{gen}$.

The measured generator speed signal Ω_{gen} is passed through the high-pass filter

$$\mathcal{H}(s) = \frac{s^2}{s^2 + 2\zeta_{\mathcal{H}}\omega_{\mathcal{H}}s + \omega_{\mathcal{H}}^2}, \tag{32}$$

where $\omega_{\mathcal{H}} = 0.63\,\mathrm{rad\,s^{-1}}$ is the cut-off frequency and $\zeta_H = 0.7$ determines the damping of the filter. In order to make sure that the cost function is dominated by the drivetrain excitations, the output component of the cost function Eq. (1) is filtered. The weighting filter

$$L_y(s) = \frac{\omega_r s}{s^2 + 2\zeta_r\omega_r s + \omega_r^2}, \tag{33}$$

with $\omega_r = 10.49\,\mathrm{rad\,s^{-1}}$ at the drivetrain frequency and $\zeta_r = 0.1$, is effectively an inverted notch filter passing through the frequencies close to the drivetrain frequency and attenuating other frequencies. The input component of the cost function is also filtered, as one in this case is only interested in the high-frequency part of the generator torque set point T_{gen}. Therefore, the input component is also filtered with the high-pass filter Eq. (32). The cost function then becomes

$$\begin{aligned} J_{dtd}(\rho) = \frac{1}{2N} \sum_{k=1}^{N} E\big[&(L_y\mathcal{H}\Omega_{gen}(k,\rho))^2 \\ &+ \lambda\mathcal{H}T_{gen}(k,\rho)^2\big], \end{aligned} \tag{34}$$

which thus consists of the measured response of the generator speed Ω_{gen} filtered by Eqs. (32) and (33) and the demanded generator torque T_{gen} filtered by Eq. (32). These filters are also included in the gradient experiment configuration in Fig. 5 just before the controller derivatives.
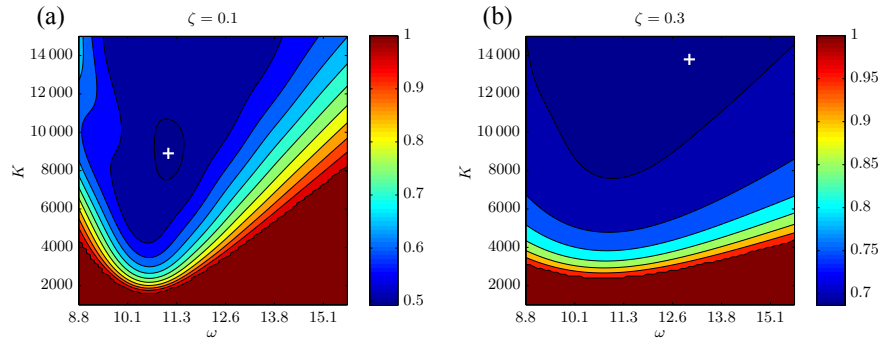
Figure 8. Close-up of the optimal parameter combinations for the drivetrain excitation problem with $\zeta = 0.1$ (left) and $\zeta = 0.3$ (right). The results are thresholded and normalized for clarity. The results show large areas where the cost function is almost optimal. The minimum values of the cost function for each case are indicated by the white + marker.

The experiments of IFT for the iterative drivetrain controller optimization are according to Sect. 2.4 as follows:

First experiment (T: 0–20 s) In the first experiment, the closed-loop system is operated for 20 s (i.e., $N = 2000$ samples), with C_1 and C_2 identical to Fig. 4. The output Ω_{gen} is filtered by Eqs. (32) and (33), and the input T_{gen} is filtered by Eq. (32).

Second experiment (T: 20–40 s) The second experiment is identical to the first experiment.

Third experiment (T: 40–60 s) In the third experiment, the filtered output $\mathcal{H}\Omega_{gen}$ from the first experiment is added to the torque set point T_{gen} and the recorded signals from the second experiment are subtracted as shown in Fig. 5.

After the third experiment at $T = 63$ s, the controller parameters ρ are updated according to Eq. (33), and before the next iteration is started, 7 s is used to let the transients disappear that are caused by the controller parameter update. Thus, a full iteration takes 70 s. In the results, the IFT algorithm is also adjusted to collect $N = 1000$ and $N = 3000$ samples (10 and 30 s, respectively) per experiment. In these cases the experiment lengths are adjusted accordingly and the iterations therefore take 40 and 100 s in total, respectively. In all cases, the first iteration starts at $T = 30$ s after the start of the simulation, such that all initial transients have disappeared.

The generator torque set point T_{dtd} provided by the IFT algorithm is limited to ± 1.8 kNm. Moreover, the generator torque set point is also limited in the rate of change, i.e., a maximum rate of ± 20 kNm s^{-1} is allowed.

4.4 Results

The results for IFT of the drivetrain damper are subdivided into four parts, each covering a different aspect of the IFT results.

4.4.1 General analysis of results

In the first part, the results for a number of general settings of IFT are analyzed. The wind field considered here has a mean wind speed of 14 m s^{-1} and a turbulence intensity of 10 %. The number of data points considered is $N = 2000$, corresponding to 20 s of simulated time (recall that the controllers run at a sampling time of 0.01 s) and the parameter update step size of Eq. (2) is set to $\gamma = 0.3$. The adjustable signal-to-noise ratio parameter F is set to 2000, which was experimentally found to provide a good trade-off between the signal-to-noise ratio and the amplitude of the injected signal.

First, the convergence of the cost function and controller parameters for different optimization cases are considered. In total four cases are considered:

1. The baseline was considered. In the baseline case the controller parameters are held constant and are given for reference.

2. ρ_{base} was considered. In the second case, the initial parameters $\rho_{0,base}$ are equal to the baseline parameters (i.e., $K = 2500$, $\omega = 10.49$ rad s^{-1}, and $\zeta = 0.3$), from which IFT procedure is started. The input weighting is chosen to be $\lambda = 5 \times 10^{-7}$.

3. $\rho_{base,\lambda}$ was considered. The third case is identical to the second case except that the input weighting is smaller, i.e., $\lambda = 2 \times 10^{-7}$.

4. ρ_{subopt} was considered. In the final case, the initial parameters $\rho_{0,subopt}$ are chosen to be far from the baseline case: $K = 1000$, $\omega = 6.28$ rad s^{-1}, and $\zeta = 0.4$. The input weighting is identical to the second case.

The results of the comparison are shown in Fig. 9. From the three plots related to the cost, it is observed that the baseline controller is already rather close to optimal controller performance. The controller with suboptimal initial controller parameters converges towards the other controller
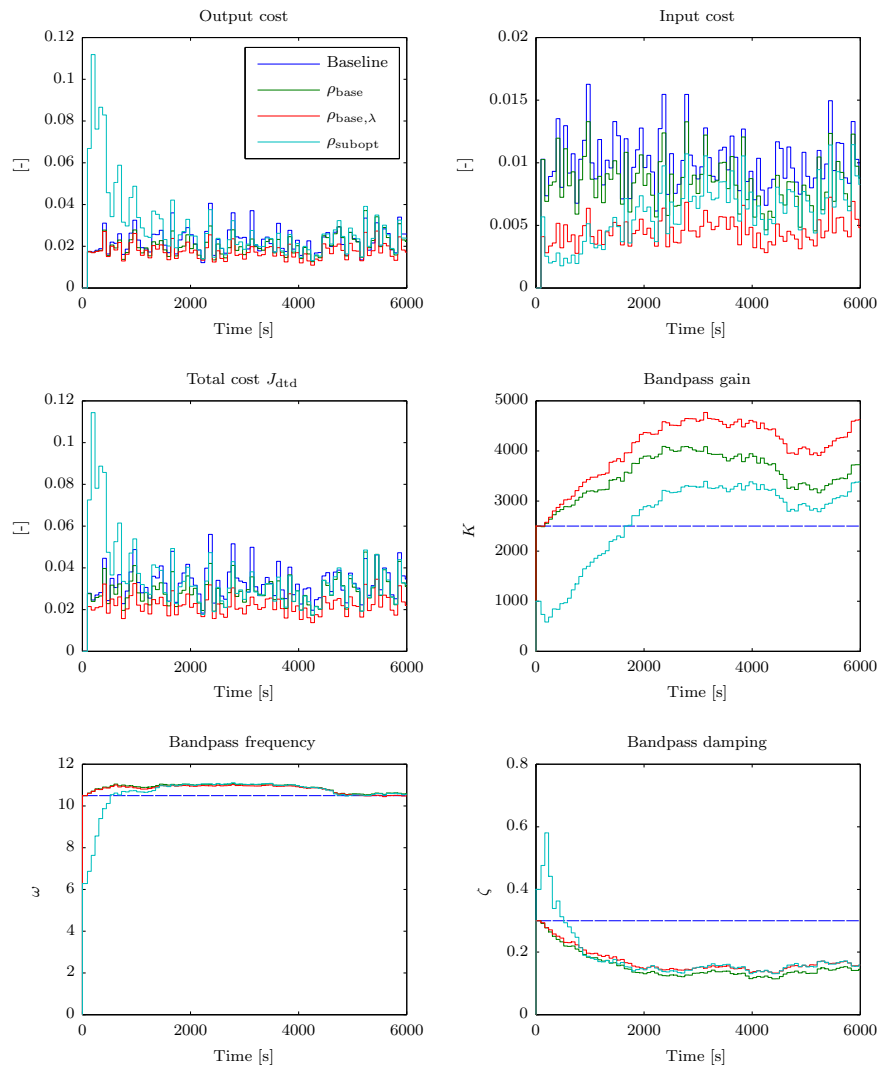
Figure 9. Comparison of three drivetrain torsional damping controllers, two of which are subject to IFT. The baseline controller parameters are kept constant and the result is shown as reference. The other three cases involve IFT, where the torsional damper is optimized starting from different initial conditions. Results shown are obtained with $\gamma = 0.3$, $F = 2000$, and turbulent wind with mean speed $14\,\mathrm{m\,s^{-1}}$ and $10\,\%$ turbulence intensity.

cases in roughly 10 iterations.[1] It is observed that the optimization of the controller parameters converges in a way that it varies around the steady-state result. The variance of the trajectory is correlated to the turbulence intensity used.

The main performance improvement is observed in the generator torque effort. It can be observed that the bandpass gain K and bandpass damping ζ of the baseline controller should, respectively, be increased and decreased in order to improve the performance. The bandpass frequency ω remains close to the drivetrain resonance frequency ω_d during the optimization. The influence of the lower input weighting λ in the case of $\rho_{\mathrm{base},\lambda}$ is clearly seen in the bandpass trajectory.

With the controller parameters obtained in the last iteration of the second case, a comparison is made with the baseline case in a normal design load case according to IEC (2005). The frequency spectra of the demanded generator torque T_gen and the resulting generator speed Ω_gen are shown in Fig. 10. From the spectrum plot of Ω_gen it can be observed that the IFT-optimized controller yields a higher damping around the drivetrain frequency ω_d. The optimized parameterization also slightly increases the frequency contents around the drivetrain mode. The demanded generator torque T_gen displays a similar result. There is slightly more energy concentrated at and around the drivetrain frequency, while at low and high frequencies the energy is reduced.

---

[1]To support the statement of convergence, an extended optimization is performed and results are included in Appendix C, Fig. C1.
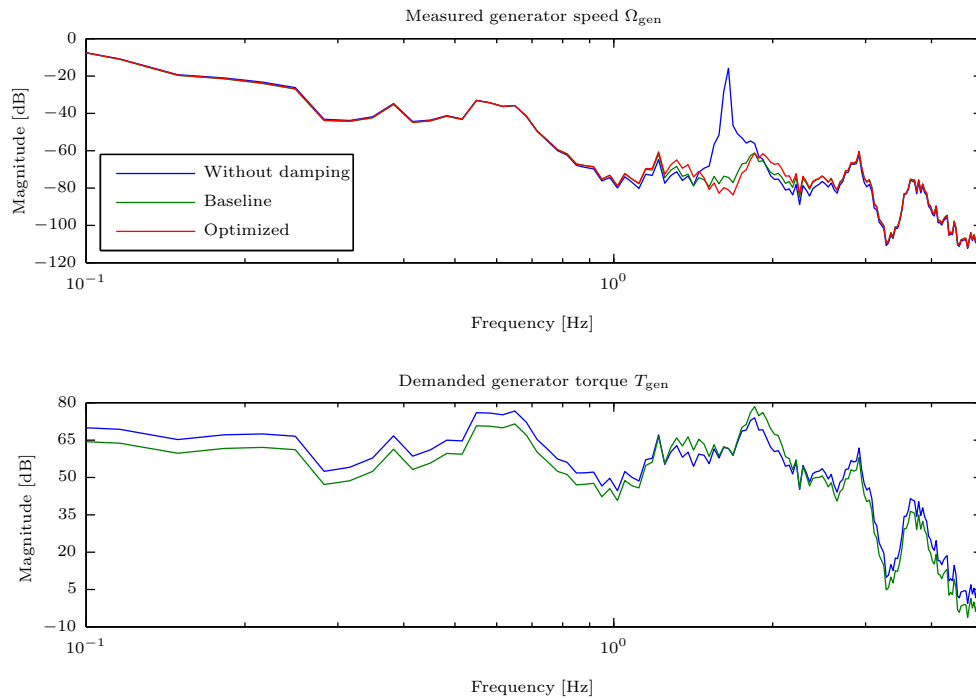
Figure 10. Comparison of the baseline and IFT-optimized drivetrain controller performance.

4.4.2 Impact of F on results

In this paragraph, the influence of the scaling factor F on the IFT performance is investigated. The IFT algorithm is carried out on the drivetrain damper using different values of F. The initial parameters are identical to the baseline and the step size is chosen to be $\gamma = 0.3$ and the input weight to be $\lambda = 5 \times 10^{-7}$. The number of data samples collected is again set to $N = 2000$, corresponding to experiment lengths of 20 s. The turbulent wind field is identical to that in the previous paragraphs. The results for three cases where F is varied between $F = 1000$ and $F = 2000$ are shown in Fig. 11.

The first thing that can be observed from the cost function plot is that the cost function slightly reduces with an increase in the scaling factor F. The effect of F on the optimization is more recognized in the convergence trajectories of the bandpass gain K and the bandpass damping coefficient ζ. Clearly, by increasing the scaling factor F, the convergence rate increases. This is also what one could expect from reasoning because the excitation of the system increases with F, as can also be seen from the lower plot in Fig. 11. The choice of F seems to be independent, at least for the considered values, of the final values to which the controller parameters converge. Thus, when applying IFT, the scaling factor should be carefully chosen such that the choice for F in that sense becomes a trade-off between the convergence rate and maximum allowed magnitude of the injected signal.

4.4.3 Varying experiment length N

In the previous results, the length of the experiments in each iteration was $N = 2000$ (20 s). Here, the iterative optimization results are compared for experiment lengths of 10, 20, and 30 s. The step size and input weight are set to $\gamma = 0.3$ and $\lambda = 5 \times 10^{-7}$, and the scaling factor is set to $F = 2000$. The results are shown in Fig. 12.

It is observed that the experiment length has a clear influence on the variance of the obtained results, both for the cost function plots as well as the controller parameters. The results also indicate that for the drivetrain damping case the experiment lengths do not dramatically change the optimization outcomes. It seems that for the $N = 1000$ case the load reduction performance is slightly better than in the other cases. Similarly, the $N = 3000$ case seems to result in the lowest generator torque effort to reduce the drivetrain oscillations. Moreover, for this case the parameters remain rather close to the original baseline parameters. It can be argued that the $N = 2000$ case provides the best trade-off between the variance, performance, and convergence time.

4.4.4 Varying wind conditions

In the final part of this case study, IFT is also applied to a wind speed of $10 \, \mathrm{m \, s^{-1}}$ at below-rated operating conditions. Although the need for an active drivetrain damper in below-rated varying wind conditions is strongly turbine dependent, this case study will show the ability of the algorithm to optimize controllers while turbine power control is active. For
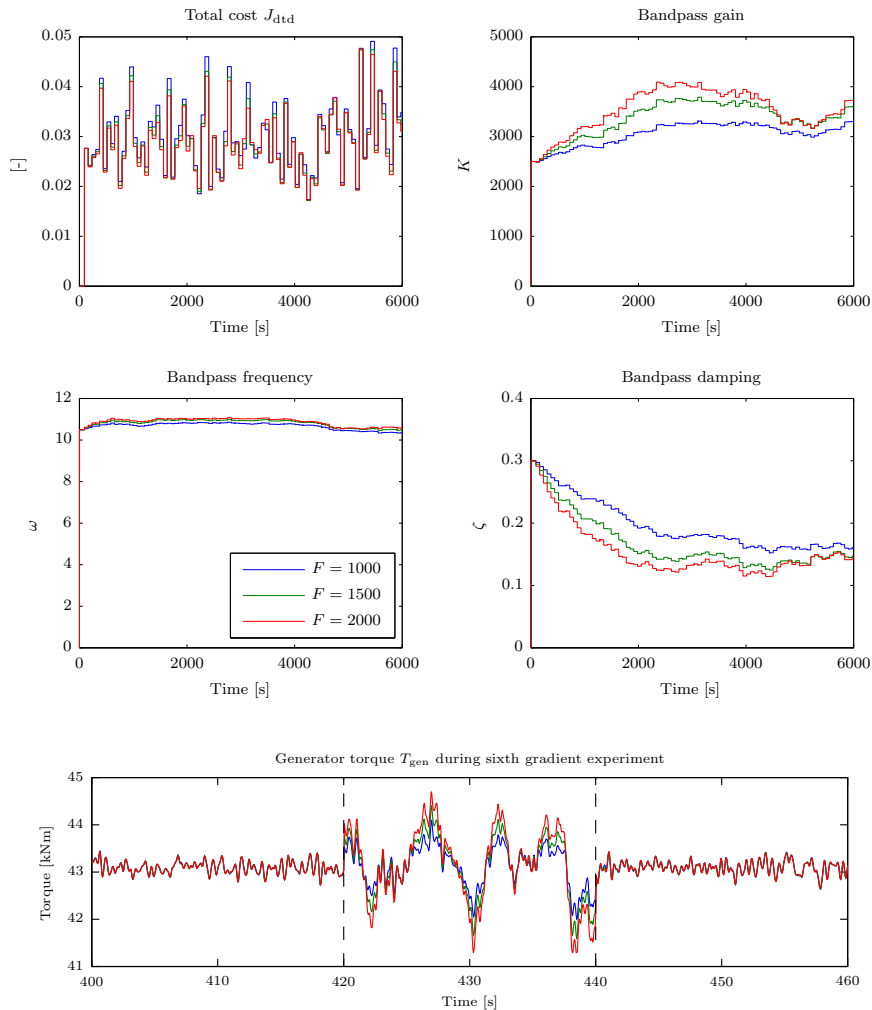
Figure 11. Comparison of three different signal-to-noise ratio scaling factors F. Results are shown for $\gamma = 0.3$, $\lambda = 5 \times 10^{-7}$, and turbulent wind with mean speed $14\,\mathrm{m\,s^{-1}}$ and $10\,\%$ turbulence intensity.

comparison, the $14\,\mathrm{m\,s^{-1}}$ wind case is also shown. The step size and input weight are once more set to $\gamma = 0.3$ and $\lambda = 5 \times 10^{-7}$, and the scaling factor is chosen to be $F = 3000$ for the $10\,\mathrm{m\,s^{-1}}$ wind and $F = 2000$ for the $14\,\mathrm{m\,s^{-1}}$ wind. The experiment length is kept at $N = 2000$ samples. The results are displayed in Fig. 13.

It is observed in the plots that the convergence trajectories for both wind speeds are very similar. The $14\,\mathrm{m\,s^{-1}}$ turbulent wind case excites the drivetrain mode more than the $10\,\mathrm{m\,s^{-1}}$ wind case as is suggested by the increased cost, which thus also requires a higher input energy. The cost function plots show a number of iterations with a clearly higher input cost. This is caused by sudden wind speed changes to which the torque controller responds. Although the input in the cost function is high-pass filtered, these sudden changes remain partly in the input signal. On such occasions, the controller parameters also display relatively large changes. This could be overcome by increasing the cut-off frequency of the high-

pass filter that is used to filter the input signal in the cost function or by increasing the experiment length such that these effects are averaged (i.e., the experiment length acts as low-pass filter on the results).

5   IFT of CPC

This section presents the results of the IFT algorithm applied to step tuning of the CPC. In the next paragraph, first the controller structure is given, including the details of the IFT method thereof. In the subsequent section the optimization results for several cases are discussed.

5.1   CPC design and IFT implementation

The CPC of a wind turbine generally consists of a proportional integral (PI) controller cascaded with some filters that prevent undesired contributions in the pitch signal. The full
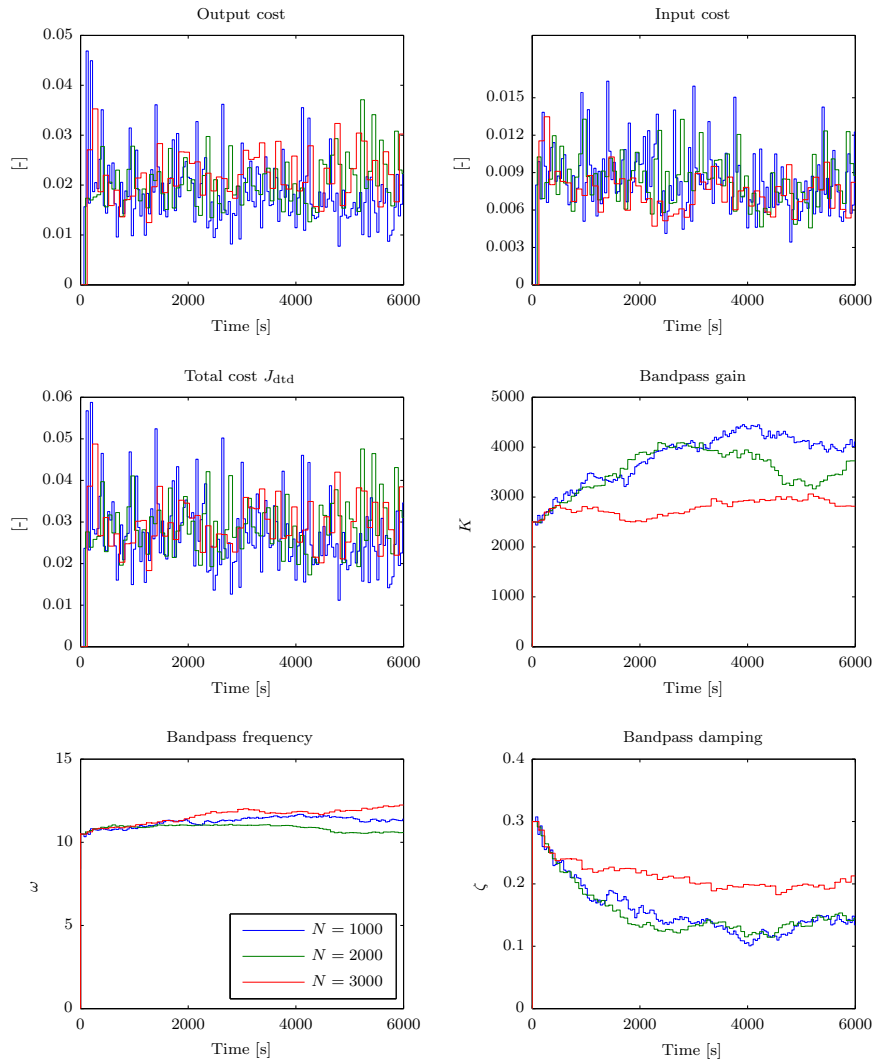
Figure 12. Comparison of IFT performance for different number N of collected data samples. Results shown are obtained with $\gamma = 0.3$, $\lambda = 5 \times 10^{-7}$, $F = 2000$, and turbulent wind with mean speed $14\,\mathrm{m\,s^{-1}}$ and $10\,\%$ turbulence intensity.

controller of the CPC loop is given by

$$C_{cpc} = \left(K_p + \frac{K_i}{s} \right) \times \underbrace{\frac{s^2 + 2\zeta_{z,3P}\omega_{3P}s + \omega_{3P}^2}{s^2 + 2\zeta_{p,3P}\omega_{3P}s + \omega_{3P}^2}}_{\text{Notch at 3P frequency}}$$

$$\times \underbrace{\frac{s^2 + 2\zeta_{z,d}\omega_d s + \omega_d^2}{s^2 + 2\zeta_{p,d}\omega_d s + \omega_d^2}}_{\text{Notch at dtr frequency}} \times \underbrace{\frac{\omega_{LP}^2}{s^2 + 2\zeta_{p,LP}\omega_{LP}s + \omega_{LP}^2}}_{\text{Low−pass filter}}, \qquad (35)$$

where the filter coefficients are listed in Table 2. The CPC thus includes a notch filter that prevents it from working on the 3P (3 times per revolution) frequency present in the generator speed signal, similarly a notch filter that prevents the controller from reacting to the drivetrain frequency component and a low-pass filter that removes all frequencies above 1.6 Hz are included. The controller takes the measured

generator speed Ω_{gen} as input signal and outputs a demanded collective pitch signal θ; see Fig. 6.

The IFT algorithm is applied so as to optimize the step response tracking of the controller in Eq. (35). Since CPC is aimed at disturbance rejection rather than reference tracking, this might seem like an odd optimization objective. However, for this single-input single-output case they both target the same sensitivity. Therefore, the frequency-dependent performance is dependent on the excitation signal. Since the power spectrum of a step could be considered as a weighted version of a wind spectrum, with a higher emphasis on the low-frequency components, meaningful results can be obtained using this approach. This can be further improved by using a reference signal with a power spectrum similar to that of a wind signal. The advantage of the approach followed is that the reference signal is user defined and the wind is uncontrollable. The step change is imposed in the rated genera-
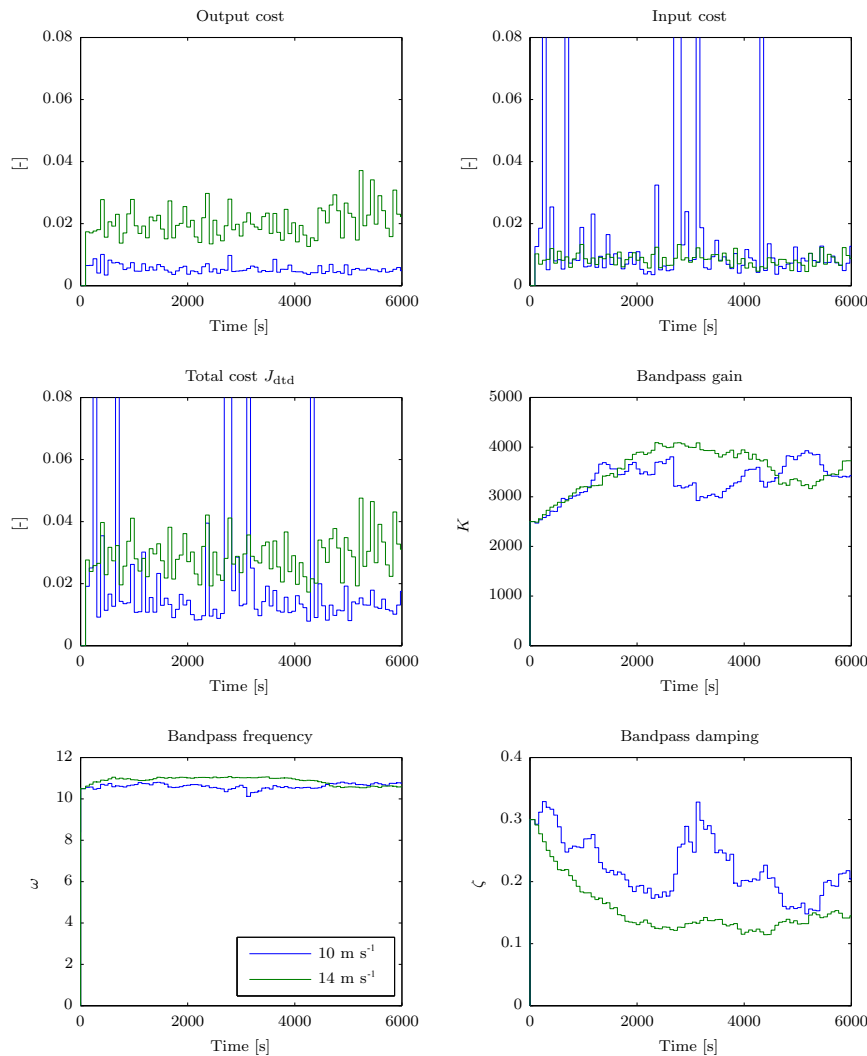
Figure 13. Comparison of IFT performance for turbulent wind with mean wind speeds 10 and 14 m s^{-1}, both with 10 % turbulence intensity. Results shown are obtained with $\gamma = 0.3$ and $\lambda = 5 \times 10^{-7}$.

tor speed signal, i.e., $\Omega_{\text{ref}} = \Omega_{\text{rated}} - \Omega_{\text{step}}$. A negative step change Ω_{step} is applied to prevent the turbine from going into overspeed. The generator speed signal has a constant offset $r_o = \Omega_{\text{rated}}$ and therefore the experiments according to Sect. 2.3 and Figs. 1 and 3 are used:

First experiment (T: 20–40 s) In the first experiment, the step change $\Omega_{\text{ref}} = \Omega_{\text{rated}} - \Omega_{\text{step}}$ in the reference input is applied for 20 s. The generator speed response Ω_{gen} and the high-pass-filtered process input θ are recorded.

Second experiment (T: 0–20 s) In the second experiment, the closed-loop system is operated with $\Omega_{\text{ref}} = \Omega_{\text{rated}}$. The output Ω_{gen} and process input θ are recorded and used in the next experiment.

Third experiment (T: 65–85 s) In the third experiment, the gradients are obtained by operating the closed-loop

system with $\Omega_{\text{ref}} = \Omega_{\text{rated}}$, injecting the error signal $F(\Omega_{\text{ref}} - \Omega_{\text{gen}})$ from the second experiment at the process input θ, using the recorded signals of the first experiment at the process input of the system according to Sect. 2.3, and filtering with controller derivatives (including the high-pass filter for the input gradient).

The time between the second and third experiments is required to make sure all oscillations due to the step change have disappeared. At $T = 86$ s, the controller parameters are updated. Then, after 34 s during which the transients due to the gradient experiment and the controller update have disappeared, at $T = 120$ s, the next iteration is started. The first iteration starts at $T = 30$ s. During the optimization, the maximum pitch rate $\dot{\theta}$ is limited to $\pm 8°$ s^{-1}. Notice that the order of the first and second experiments have been reversed during the optimization in comparison to Sect. 2.3.

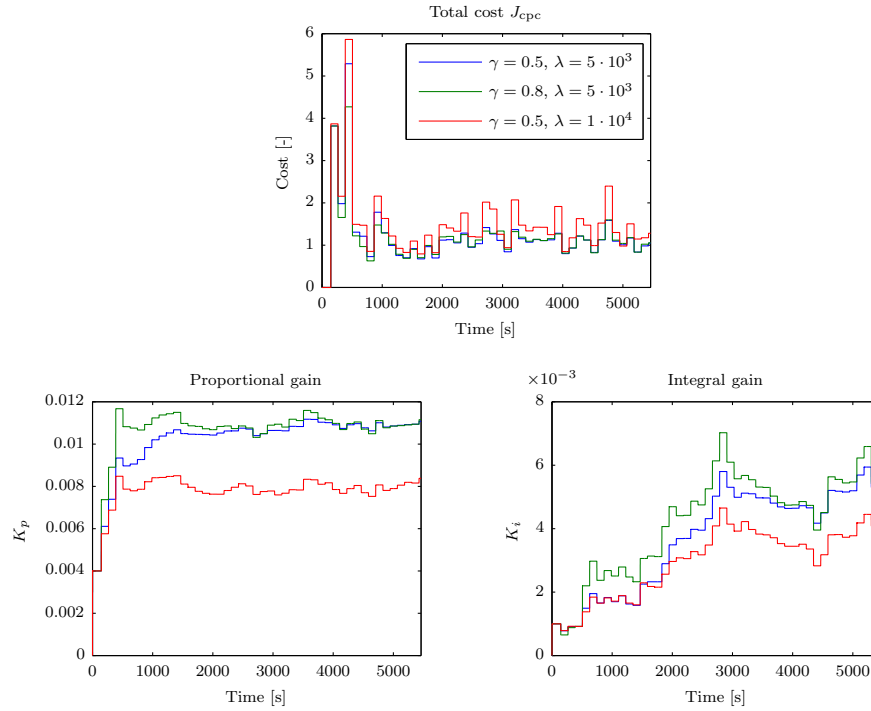**Figure 14.** Reference step tuning using IFT of CPC at a wind speed of $14\,\mathrm{m\,s^{-1}}$ with $4\,\%$ turbulence intensity. The results show three different cases where the step size γ and the input weighting λ in the cost function are varied.
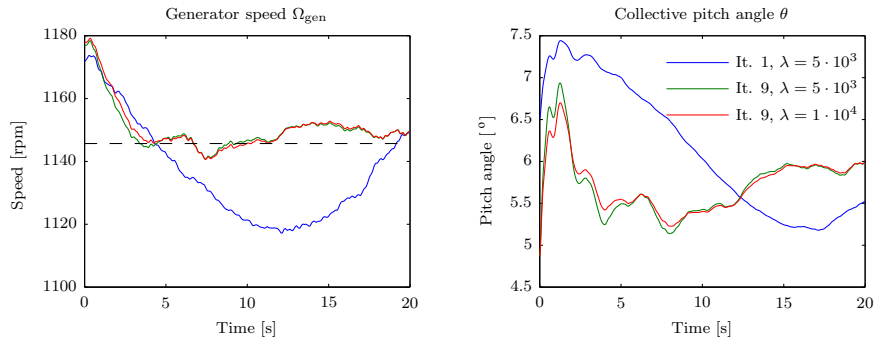

Figure 15. Comparison of the generator speed step response during the first and ninth iterations, and for different input weighting factors λ. Results shown are obtained for a wind speed of $14\,\mathrm{m\,s^{-1}}$ with $4\,\%$ turbulence intensity.

The cost function is chosen as

$$
\begin{aligned}
J_{\mathrm{cpc}}(\rho) = \frac{1}{2N} \sum_{k=1}^{N} E\big[& (\Omega_{\mathrm{gen}}(k,\rho) - \Omega_{\mathrm{ref}}(k))^2 \\
& + \lambda L_\theta \theta(k,\rho)^2 \big],
\end{aligned}
\tag{36}
$$

where the input weighting factor λ and the step size γ in the parameter update rule Eq. (2) are both considered for different values. Note that the input signal in the cost function is high-pass filtered by L_θ, which is identical to Eq. (32). The high-pass filter L_θ is required because the optimization procedure should focus on the dynamic pitch response rather

than on the static pitch offset required to maintain the rated generator speed.

5.2 Results of IFT for CPC

The first result is obtained by optimizing the CPC for a turbulent wind field with a mean wind speed of $14\,\mathrm{m\,s^{-1}}$ and turbulence intensity of $4\,\%$. The step change for this result is chosen to be $\Omega_{\mathrm{step}} = 30\,\mathrm{rpm}$. Moreover, the adjustable signal-to-noise ratio gain F is set to 0.02. The initial PI controller values are $K_{\mathrm{p}} = 4 \times 10^{-3}$ and $K_{\mathrm{i}} = 1 \times 10^{-3}$. The results for three cases with varying input weighting λ and step size γ are shown in Fig. 14.

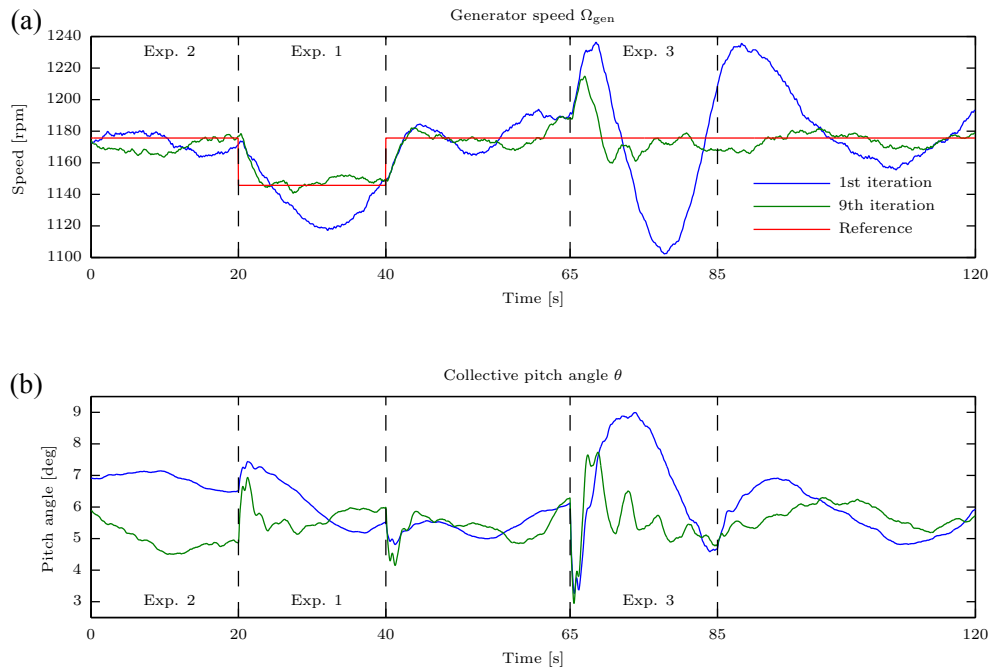**Figure 16.** Comparison of responses during the first and ninth iterations. Results shown are obtained for a wind speed of $14 \, \mathrm{m \, s^{-1}}$ with 4 % turbulence intensity. The measured generator speed and reference input are shown in the upper plot, and the lower plot displays the pitch response.

Table 2. Filter parameters of the different controller components.

Description	Symbol	Value
Low-pass filter		
Low-pass filter frequency	ω_{LP}	$10.05 \, \mathrm{rad \, s^{-1}}$
Low-pass filter damping	$\zeta_{p,\mathrm{LP}}$	0.7
Notch filter at 3P frequency		
Notch filter frequency	$\omega_{3\mathrm{P}}$	$3.77 \, \mathrm{rad \, s^{-1}}$
Notch filter damping zero	$\zeta_{z,3\mathrm{LP}}$	0.0015
Notch filter damping pole	$\zeta_{p,3\mathrm{LP}}$	0.15
Notch filter at drivetrain frequency		
Notch filter frequency	ω_{d}	$10.49 \, \mathrm{rad \, s^{-1}}$
Notch filter damping zero	$\zeta_{z,\mathrm{d}}$	0.002
Notch filter damping pole	$\zeta_{p,\mathrm{d}}$	0.2

The trajectory of the cost function values and the controller parameters are shown in Fig. 14. Note that the performance objective in this work is expressed by the cost function $J(\rho)$, and minimization of this function is considered as improved performance. Another method for controller performance comparison is the use of Pareto curves (Odgaard et al., 2015a, b), but this technique will not be considered.

It can be observed that the initial controller parameters were suboptimal and they converge in a few iterations to a much better performance. The results also show that the con-

vergence of the parameters behave differently. The proportional gain K_p converges to its final value after four iterations. Conversely, the integral controller gain K_i slowly increases to larger values. The difference in trajectory can be explained due to the fact that in the step response the proportional gain is more dominant. In order to make the integral controller parameter more dominant, one could increase the experiment length N.

The effect of the gains γ and λ on the convergence trajectories are apparent in Fig. 14. It is observed that the convergence of the proportional gain K_p for the cases where $\lambda = 5 \times 10^3$ is faster due to the higher step size γ. The effect of increasing the weight λ in the cost function is also clear from Fig. 14: the parameters converge to smaller values, which is expected.

In Fig. 15, three different step responses are shown. The blue graph displays the response of the initial controller to the step change. As can be observed, this response is sloppy and after nine iterations has improved to a decent response. The increased weight on the input cost yields a step response where less pitch duty is required with only a very limited loss of tracking performance. In Fig. 16, the generator speed response and the collective pitch angle for a full iteration during the first and ninth iterations are displayed.

The final results involve a comparison between IFT of CPC for two different wind speeds: 14 and $18 \, \mathrm{m \, s^{-1}}$ with 4 % turbulence intensity. It is generally known that the control authority of CPC increases when the blades are fur-

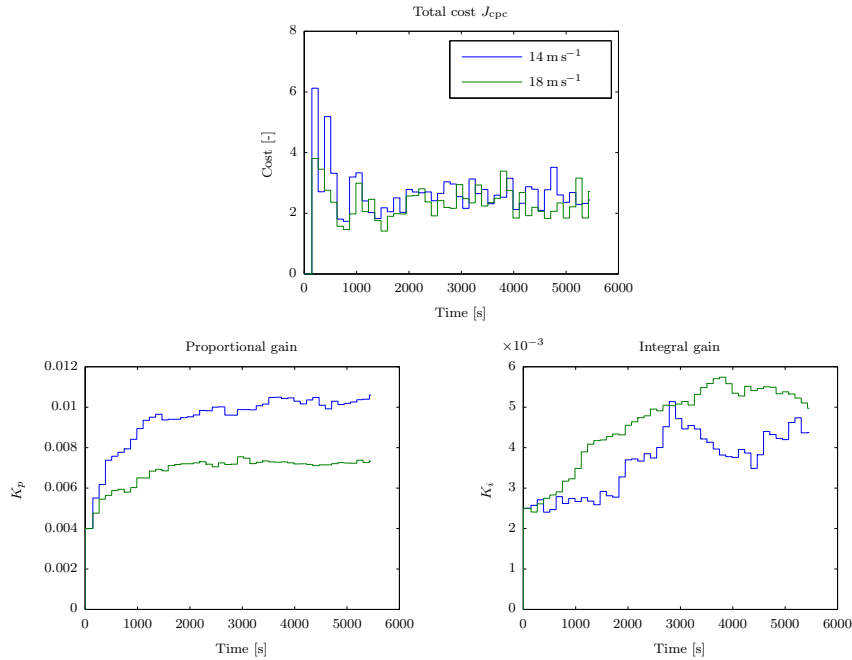**Figure 17.** Reference step tuning of CPC using IFT of CPC at a wind speed of $14 \, \mathrm{m \, s^{-1}}$ with 4 % turbulence intensity.
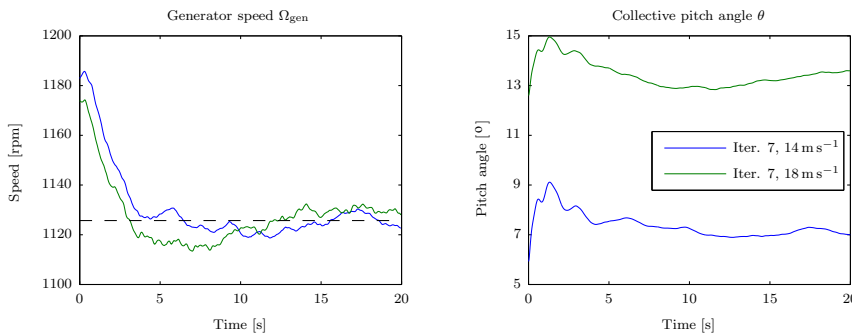

Figure 18. Comparison of the CPC step responses for 14 and $18 \, \mathrm{m \, s^{-1}}$ with 4 % turbulence intensity during the seventh iteration.

ther pitched from the wind, which means that less pitch effort is required to keep the rotor speed close to rated. The IFT tuning results also display this behavior. In Fig. 17 it can be seen that the proportional gain for $18 \, \mathrm{m \, s^{-1}}$ is roughly two-thirds of the value compared to the $14 \, \mathrm{m \, s^{-1}}$ case. Conversely, the integrator gain K_i is somewhat higher for the $18 \, \mathrm{m \, s^{-1}}$ wind speed. The cost function converges to a comparable result. The step responses shown for the seventh iteration are also rather similar.

6 Conclusions

In this paper, IFT controllers for wind turbines have been developed. The typical controller configurations used for wind turbine control require three closed-loop experiments to be carried out. With the data that are collected during these experiments, it has been shown that IFT can be successfully applied. The results indicate that starting the optimization from a baseline controller with decent performance can already improve the performance within a few iterations. It has also been shown that IFT can be applied to both disturbance rejection and reference tracking control for wind turbines. This is demonstrated by means of optimizing the drivetrain damping controller and the CPC. The methodology could similarly be applied to improve fore–aft and/or side–side tower damping performance. Finally, it is argued that IFT could be a valuable tool with which the performance of wind turbine controllers can be improved without the need of system identification.

Appendix A: Derivation of input gradient for Sect. 2.3

Before the derivation of the input gradient is given, first recall that the control system in Sect. 2.3 is according to Fig. 1, and therefore the sensitivity function $S(\rho)$ and complementary sensitivity function $T(\rho)$ are given by

$$S(\rho) = (I + PC(\rho))^{-1}, \tag{A1}$$

$$T(\rho) = (I + PC(\rho))^{-1} PC(\rho). \tag{A2}$$

Then, note that in Fig. 1 the input $u(k, \rho)$ is determined by

$$u(k, \rho) = S(\rho)C(\rho)r_o - S(\rho)C(\rho)v(k), \tag{A3}$$

where $r(k)$ has been replaced by r_o. The gradient of Eq. (A3) with respect to ρ is derived as

$$\frac{\partial u}{\partial \rho}(k, \rho) = \left(\frac{\partial S}{\partial \rho} C(\rho) + S(\rho) \frac{\partial C}{\partial \rho}(\rho) \right) r_o$$
$$- \left(\frac{\partial S}{\partial \rho} C(\rho) + S(\rho) \frac{\partial C}{\partial \rho}(\rho) \right) v(k). \tag{A4}$$

The derivative of the sensitivity function $S(\rho)$ equals

$$\frac{\partial S}{\partial \rho} = -(I + PC(\rho))^{-2} P \frac{\partial C}{\partial \rho}(\rho) = -S^2 P \frac{\partial C}{\partial \rho}(\rho). \tag{A5}$$

Substituting for the latter sensitivity derivative in Eq. (A4) yields

$$\frac{\partial u}{\partial \rho}(k, \rho) = S(\rho) \frac{\partial C}{\partial \rho}(\rho) \big(r_o - T(\rho) r_o$$
$$+ T(\rho) v(k) - v(k) \big). \tag{A6}$$

Realizing that $(T(\rho) - I)v(k) = -S(\rho)v(k)$ and that $T(\rho)r_o + S(\rho)v(k) = y(k, \rho)$ gives the input gradient

$$\frac{\partial u}{\partial \rho}(k, \rho) = S(\rho) \frac{\partial C}{\partial \rho}(\rho)(r_o - y(k, \rho)). \tag{A7}$$

Appendix B: Derivation of input gradient for Sect. 2.4

The derivation of the input gradient in Sect. 2.4 is similar to the derivation in Appendix A. Here, the sensitivity $S(\rho)$ and complementary sensitivity $T(\rho)$ are as follows

$$S(\rho) = (I + PC_1 - PC_2(\rho)\mathcal{H})^{-1}, \tag{B1}$$

$$T(\rho) = (I + PC_1 - PC_2(\rho)\mathcal{H})^{-1} PC_1. \tag{B2}$$

In Fig. 4, the input $u(k, \rho)$ equals

$$u(k, \rho) = S(\rho)C_1 r(k) + S(\rho)C_2(\rho)\mathcal{H}v(k) - S(\rho)C_1 v(k), \tag{B3}$$

such that the gradient can be derived to be

$$\frac{\partial u}{\partial \rho}(k, \rho) = \frac{\partial S}{\partial \rho} C_1 r(k) + \left(\frac{\partial S}{\partial \rho} C_2(\rho)\mathcal{H} + S(\rho) \frac{\partial C_2}{\partial \rho} \mathcal{H} \right.$$
$$\left. - \frac{\partial S}{\partial \rho} C_1 \right) v(k), \tag{B4}$$

where

$$\frac{\partial S}{\partial \rho} = (I + PC_1 - PC_2(\rho)\mathcal{H})^{-2} P\mathcal{H} \frac{\partial C_2}{\partial \rho}. \tag{B5}$$

With the derivative of the sensitivity function, the input gradient becomes

$$\frac{\partial u}{\partial \rho}(k, \rho) = S(\rho) \frac{\partial C_2}{\partial \rho} \mathcal{H} \Big(S(\rho)PC_1 r(k) + \big(S(\rho)PC_2(\rho)\mathcal{H}$$
$$- S(\rho)PC_1 + I \big) v(k) \Big). \tag{B6}$$

Realizing that $\big(S(\rho)PC_2(\rho)\mathcal{H} - S(\rho)PC_1 + I \big) v(k) = S(\rho)v(k)$ yields the final result

$$\frac{\partial u}{\partial \rho}(k, \rho) = S(\rho) \frac{\partial C_2}{\partial \rho} \mathcal{H} \Big(T(\rho)r(k) + S(\rho)v(k) \Big)$$
$$= S(\rho) \frac{\partial C_2}{\partial \rho} \mathcal{H} y(k, \rho) = S(\rho) \frac{\partial C_2}{\partial \rho} y'(k, \rho). \tag{B7}$$

Appendix C: Extended optimization simulation to show convergence
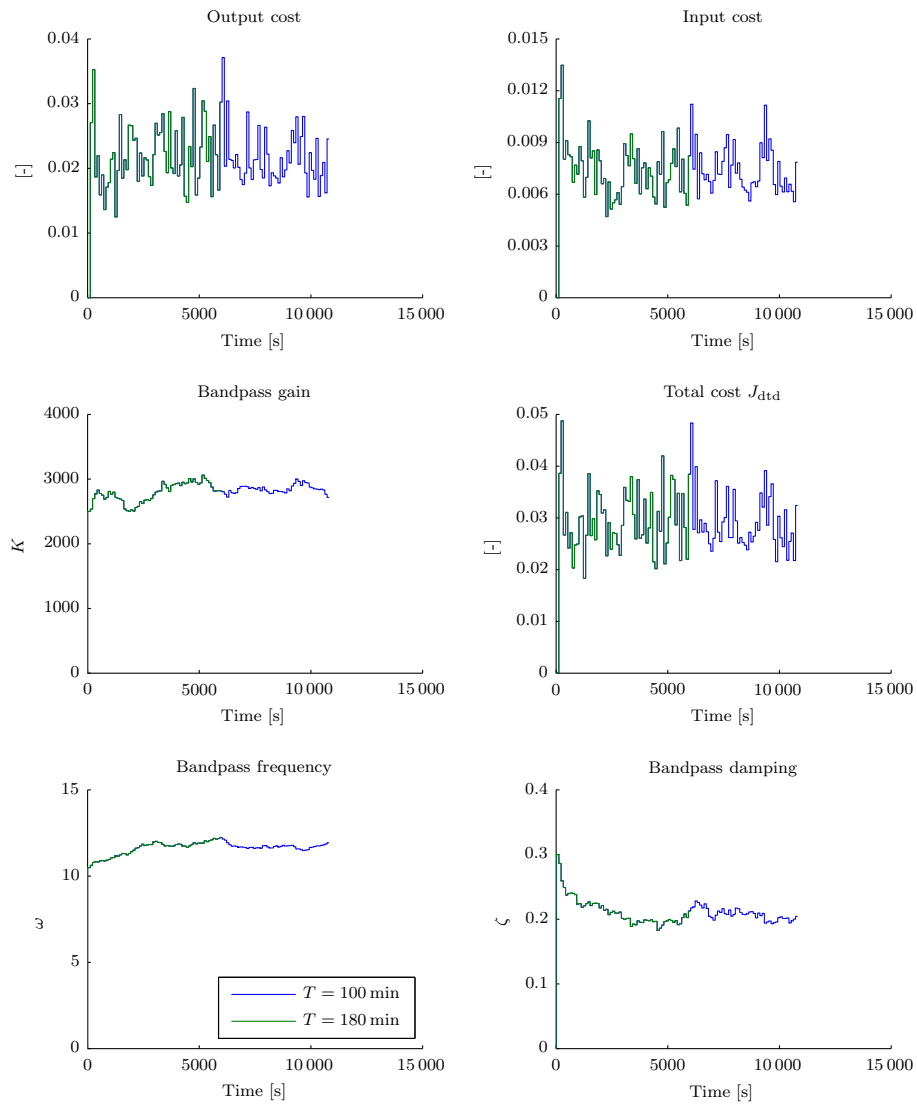

Figure C1. Comparison of IFT performance for different simulation lengths. The simulation settings used are a wind speed of $14\,\mathrm{m\,s^{-1}}$, a turbulence intensity of $10\,\%$, step size $\gamma = 0.3$, input weighting factor $\lambda = 5 \times 10^{-7}$, and scaling factor $F = 2000$.

References

Abhinav, K. A. and Saha, N.: Dynamic Analysis of an Offshore Wind Turbine Including Soil Effects, Procedia Engineering, 116, 32–39, 8th International Conference on Asian and Pacific Coasts (APAC 2015), 2015.

Al Mamun, A., Ho, W. Y., Wang, W. E., and Lee, T. H.: Iterative Feedback Tuning (IFT) of Hard Disk Drive Head Positioning Servomechanism, in: Industrial Electronics Society, 2007, IECON 2007, 33rd Annual Conference of the IEEE, 769–774, 2007.

Bossanyi, E. A.: The design of closed loop controllers for wind turbines, Wind Energy, 3, 149–163, 2000.

Bossanyi, E. A. and Witcher, D.: Controller for 5MW reference turbine, Tech. Rep. Controller for 5MW reference turbine, UP-WIND, 2009.

de Bruyne, F. and Kammer, L. C.: Iterative feedback tuning with guaranteed stability, in: American Control Conference, 1999, Proceedings of the 1999, Vol. 5, 3317–3321, 1999.

Dixit, A. and Suryanarayanan, S.: Towards Pitch-Scheduled Drive Train Damping in Variable-Speed, Horizontal-Axis Large Wind Turbines, in: Decision and Control, 2005 and 2005 European Control Conference, CDC-ECC'05, 44th IEEE Conference, 1295–1300, 2005.

Fleming, P. A., van Wingerden, J. W., and Wright, A. D.: Comparing State-Space Multivariable Controls to Multi-SISO Controls for Load Reduction of Drivetrain-Coupled Modes on Wind Turbines through Field-Testing, AIAA Wind Energy Symposium, AIAA, Reston, 2011.

Fleming, P. A., van Wingerden, J. W., Scholbrock, A. K., van der Veen, G., and Wright, A. D.: Field testing a wind turbine drivetrain/tower damper using advanced design and validation techniques, in: American Control Conference (ACC 2013), IEEE, New York, 2227–2234, 2013.

Garrad Hassan: Bladed 4.20, available at: http://www.gl-garradhassan.com, last access: 1 December 2015.

Garrad Hassan & Partners Ltd: Bladed User Manual, version 4.2, 2017a.

Garrad Hassan & Partners Ltd: Bladed Theory Manual, version 4.2, 2017b.

Heertjes, M. F., Van der Velden, B., and Oomen, T.: Constrained Iterative Feedback Tuning for Robust Control of a Wafer Stage System, IEEE T. Contr. Syst. T., 24, 56–66, 2016.

Hjalmarsson, H.: Iterative feedback tuning – an overview, Int. J. Adapt. Control, 16, 373–395, 2002.

Hjalmarsson, H.: From experiment design to closed-loop control, Automatica, 41, 393–438, 2005.

Hjalmarsson, H. and Gevers, M.: Frequency domain expressions of the accuracy of a model free control design scheme, in: 11th IFAC Symposium on System Identification, Vol. 1, 135–140, 1997.

Hjalmarsson, H., Gevers, M., Gunnarsson, S., and Lequin, O.: Iterative feedback tuning: theory and applications, IEEE Control Systems, 18, 26–41, 1998.

Houtzager, I.: External Controller Design Tool for GH Bladed and Simulink, available at: http://www.dcsc.tudelft.nl/~datadriven/discon/, last access: June 2013.

Houtzager, I., van Wingerden, J. W., and Verhaegen, M.: Wind turbine load reduction by rejecting the periodic load disturbances, Wind Energy, 16, 235–256, 2013.

IEC: IEC61400-1. Wind Turbines – Part 1: Design Requirements, Edn. 3, New York, 2005.

La Cava, W., Danai, K., Spector, L., Fleming, P., Wright, A., and Lackner, M.: Automatic identification of wind turbine models using evolutionary multiobjective optimization, Renew. Energ., 87, 892–902, 2016.

Lequin, O.: Optimal closed-loop PID tuning in the process industry with the "iterative feedback tuning" scheme, in: Control Conference (ECC), IEEE, New York, 3931–3936, 1997.

Lequin, O., Gevers, M., and Triest, L.: Optimizing the settling time with Iterative Feedback Tuning, in: Proc. 14th IFAC World Congress, Beijing, 1999.

Lequin, O., Gevers, M., Mossberg, M., Bosmans, E., and Triest, L.: Iterative feedback tuning of PID parameters: comparison with classical tuning rules, Control Engineering Practice, 11, 1023–1033, 2003.

Liu, D., McDaid, A. J., Aw, K. C., and Xie, S. Q.: Position control of an Ionic Polymer Metal Composite actuated rotary joint using Iterative Feedback Tuning, Mechatronics, 21, 315–328, 2011.

Lombardi, D., Bhattacharya, S., and Wood, D. M.: Dynamic soil-structure interaction of monopile supported wind turbines in cohesive soil, Soil Dyn. Earthq. Eng., 49, 165–180, 2013.

Mathworks: Simulink, available at: http://www.mathworks.com (last access: 1 August 2016), 2013.

Navalkar, S. T. and van Wingerden, J. W.: Iterative Feedback Tuning of an {LPV} Feedforward Controller for Wind Turbine Load Alleviation*, IFAC-PapersOnLine, 48, 207–212, 2015.

Navalkar, S. T., van Wingerden, J. W., van Solingen, E., Oomen, T., Pasterkamp, E., and van Kuik, G. A. M.: Subspace predictive repetitive control to mitigate periodic loads on large scale wind turbines, Mechatronics, 24, 916–925, 2014.

Navalkar, S. T., van Solingen, E., and van Wingerden, J. W.: Wind Tunnel Testing of Subspace Predictive Repetitive Control for Variable Pitch Wind Turbines, IEEE T. Contr. Syst. T., 23, 2101–2116, doi:10.1109/TCST.2015.2399452, 2015.

Odgaard, P. F., Knudsen, T., Overgaard, A., Steffensen, H., and Jørgensen, M.: Importance of dynamic inflow in model predictive control of wind turbines, IFAC-PapersOnLine, 48, 90–95, 2015a.

Odgaard, P. F., Knudsen, T., Wisniewski, R., and Bak, T.: Optimized control strategy for over loaded offshore wind turbines, Proceedings of Ewea Offshore 2015, EWEA, Brussels, Belgium, 2015b.

Procházka, H., Gevers, M., Anderson, B. D. O., and Ferrera, C.: Iterative Feedback Tuning for robust controller design and optimization, in: Proceedings of the Conference on Decision and Control, and the European Control Conference, IEEE, New York, 2005.

Skogestad, S. and Postlethwaite, I.: Multivariable feedback control: analysis and design, John Wiley & Sons, Chicester, 2006.

Solari, G. and Gevers, M.: Unbiased estimation of the Hessian for iterative feedback tuning (IFT), in: Decision and Control, 2004, CDC, 43rd IEEE Conference, Vol. 2, 1759–1760, 2004.

van der Veen, G. J., van Wingerden, J. W., Fleming, P. A., Scholbrock, A. K., and Verhaegen, M.: Global data-driven modeling of wind turbines in the presence of turbulence, Control Eng. Pract., 21, 441–454, 2013a.

van der Veen, G. J., van Wingerden, J. W., and Verhaegen, M.: Global Identification of Wind Turbines Using a Hammerstein Identification Method, IEEE T. Contr. Syst. T., 21, 1471–1478, 2013b.

van Kuik, G. A. M., Peinke, J., Nijssen, R., Lekou, D., Mann, J., Sørensen, J. N., Ferreira, C., van Wingerden, J. W., Schlipf, D., Gebraad, P., Polinder, H., Abrahamsen, A., van Bussel, G. J. W., Sørensen, J. D., Tavner, P., Bottasso, C. L., Muskulus, M., Matha, D., Lindeboom, H. J., Degraer, S., Kramer, O., Lehnhoff, S., Sonnenschein, M., Sørensen, P. E., Künneke, R. W., Morthorst, P. E., and Skytte, K.: Long-term research challenges in wind energy – a research agenda by the European Academy of Wind Energy, Wind Energ. Sci., 1, 1–39, doi:10.5194/wes-1-1-2016, 2016.

van Solingen, E., Mulders, S., and van Wingerden, J. W.: Source code and simulation data of iterative feedback tuning of wind turbine controllers, doi:10.5281/zenodo.345978, 2017.

van Wingerden, J. W., Hulskamp, A., Barlas, T., Houtzager, I., Bersee, H., van Kuik, G., and Verhaegen, M.: Two-Degree-of-Freedom Active Vibration Control of a Prototyped Smart Rotor, IEEE T. Contr. Syst. T., 19, 284–296, doi:10.1109/TCST.2010.2051810, 2011.

Veres, S. and Hjalmarsson, H.: Tuning for robustness and performance using iterative feedback tuning, in: Decision and Control, 2002, Proceedings of the 41st IEEE Conference, Vol. 4, 4682–4687, 2002.

Wright, A. D., Fleming, P., and van Wingerden, J. W.: Refinements and Tests of an Advanced Controller to Mitigate Fatigue Loads in the Controls Advanced Research Turbine, in: 49th AIAA Aerospace Sciences Meeting, Orlando, Florida, USA, 2011.

Xiao, Y., Li, Y., and Rotea, M.: Experimental Evaluation of Extremum Seeking Based Region-2 Controller for CART3 Wind Turbine, in: AIAA 34th Wind Energy Symposium, AIAA, Reston, 2016.