Received: 26 April 2021 | Revised: 17 December 2021 | Accepted: 6 January 2022

DOI: 10.1002/oca.2870

SPECIAL ISSUE ARTICLE

WILEY

Economic nonlinear model predictive control of fatigue—Formulation and application to wind turbine control

Stefan Loew[1,2] | **Dragan Obradovic**[2] | **Carlo L. Bottasso**[1]

[1]Wind Energy Institute, Technical University of Munich, Garching, Germany

[2]Technology, Siemens AG, Munich, Germany

Correspondence
Stefan Loew, Wind Energy Institute, Technical University of Munich, Boltzmannstrasse 15, 85748 Garching, Germany.
Email: stefan.h.loew@tum.de

Abstract

In this article, the estimation of fatigue is implemented in the cost function of a gradient-based model predictive controller (MPC). This is a challenging problem, because calculating fatigue leads to a non-standard and discontinuous cost function. Based on a brief previous publication, in the present work the method is derived, explained, and assessed in detail. The key enablers of the proposed method are a sequential implementation of MPC, the periodic substitution of discontinuous aspects of the cost function by linear functions, and the assumption of a sufficiently infrequent switching of this substitution. Fatigue cost gradients are obtained efficiently by automatic differentiation. The method is implemented in an economic nonlinear model predictive controller (ENMPC), which optimally balances revenue and fatigue cost. This novel formulation is applicable to a very wide range of domains, and it is demonstrated here on the control of a wind turbine. The proposed ENMPC is fully parameterized by physical and monetary variables, and outperforms a conventional ENMPC based on the literature. The method is assessed by considering various metrics, including frequency spectra, damage estimation, switching, and gradient dynamics, which together provide useful insight into its main characteristics and an initial assessment of its performance.

KEYWORDS

economic control, fatigue, model predictive control, wind turbine

1 | INTRODUCTION

1.1 | Motivation

The phenomenon of fatigue is characterized as damage caused by cyclic loading. Fatigue failure of a component typically occurs after a large number of cycles even at rather low amplitudes and mean load values, well below the admissible ultimate ones.[1]

The present research is part of the project "MISTRALWIND—Monitoring and Inspection of Structures At Large Wind Turbines", funded by the German Federal Ministry for Economic Affairs and Energy.

TABLE 1 Cyclic fatigue in different domains

Domain	Relevant load state	Unit	Examples
Moderately loaded mechanical structures (elastic deformation)	Mechanical stress	Pa	Towers and blades of wind turbines[2,3]
Highly loaded mechanical structures (plastic deformation)	Mechanical strain	-	Jet engine rotors[4]
Power semiconductors (thermo-mechanical variations)	Temperature	K	Insulated-gate bipolar transistors (IGBT) in power converters[5,6]
Battery energy storage systems (loss of active material by charging and discharging)	State of charge (SoC)	%	Lithium-ion cells[7,8]

Fatigue is relevant to a variety of domains, where it is driven by different system states, as shown in Table 1. The present work considers the structural fatigue of a wind turbine tower. Thus, without loss of generality, mechanical stress σ is considered as the fatigue-driving quantity throughout this work.

Fatigue has a large impact on the investment and operating costs of devices, and thus on economic profit. However, unfortunately fatigue damage cannot in general be completely avoided, and can only be *mitigated*, for example by active controls or passive design solutions. Model predictive control (MPC) enables the economically optimal management of systems by using predictions of their future response,[9,10] including stress at crucial spots in the device structure. Rainflow counting (RFC) is the standard algorithm used for the decomposition of stress time-series in the estimation of fatigue damage. It should be noticed that RFC is typically a post-processing activity: given a measured or simulated signal, RFC can be used to decompose it in a way that is amenable to the estimation of fatigue-induced damage. Unfortunately, however, RFC is a discontinuous branching algorithm, which is not per-se directly usable within an optimization context, including MPC.[2] Furthermore, RFC requires complete stress trajectories as input, and thus cannot be cast into standard stage or terminal cost functions. Consequently, previous work on MPC for fatigue reduction always has avoided using RFC, as presented in the following.

1.2 Previous work

Conventional MPC formulations for fatigue found in the literature can be classified according to the use of an indirect or a direct fatigue metric.

With indirect fatigue metrics, instead of actual fatigue damage, only a damage-related value is considered and optimized, as illustrated in the left part of Figure 1. Two main families of methods have been described:

- The *penalization of deflection rate* is a common approach for considering fatigue in MPC. In Gros et al.[11] and Evans et al.,[12] this method is implemented for wind turbine towers by penalizing the deflection rate at the tower top. Since deflection correlates with stress, this can be interpreted as damping of stress oscillations. This method does not consider stress cycles per se, and thus cannot establish a direct link to fatigue. However, since this method has served as a common basis for comparison among several publications, it will be used here as a baseline reference under the label "Tower-tip-velocity penalization" (TTVP).
- In *spectral methods*, damage is approximated by empirical functionals depending on spectral moments of the predicted stress signal.[2] A severe limitation of this approach is the inherent assumption of a narrow-banded Gaussian process, which is often violated in practical cases.

In contrast, direct fatigue metrics evaluate actual fatigue damage, which can be readily converted to monetary fatigue cost, as illustrated in the right part of Figure 1. The main approaches within this family are as follows:

- By the *serialization of stress cycles*,[3] a simplified fatigue metric is calculated in continuous time within the control horizon. Here, stress cycles are detected by sign changes in the stress signal, similarly to the *Simple Range Counting*

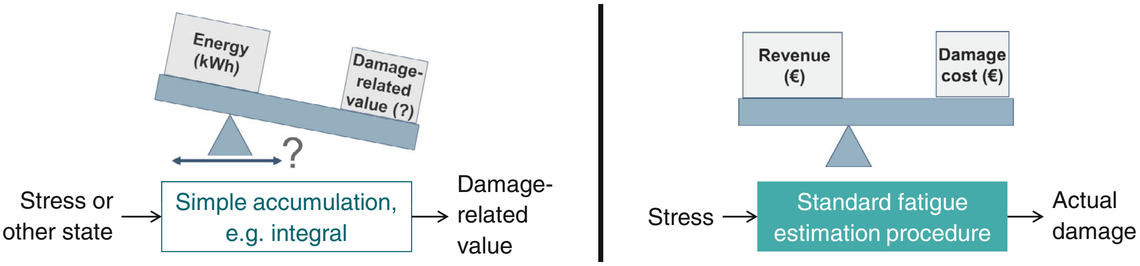

FIGURE 1 Left: Indirect fatigue metric. Right: Direct fatigue metric[13]

method of ASTM.[14] This serialization is an extreme simplification of RFC, because nesting of stress cycles is not accounted for.

- In Barradas-Berglind et al.,[15] *hysteresis operators* are used to adapt parameters of a cost function in the MPC. However, this cost function essentially penalizes only deflection rates, similarly to the TTVP approach. Additionally, the authors state that the resulting control problem is hard.
- In Collet et al.,[16] *fixed-point iterations* are used to tune a parametric nonlinear relationship of stress variance to actual fatigue at each MPC step. Two advantages of this approach are a solid relation to fatigue and the combination of proven numerical methods. The main disadvantage relates to the numerous required fixed-point iterations, with a consequently high computational cost.[16]
- In Luna et al.,[17] a surrogate *Artificial Neural Network* is trained based on damage results from a large number of stress time series. This approach seems to be very promising in terms of correct damage estimation. However, it involves a high a priori effort for the training of the surrogate model, as well as a significantly increased computational load in the MPC.[17]

In summary, all available indirect or direct fatigue metrics either do not accurately approximate fatigue damage, or result in high computational load. Since RFC is the de facto standard for fatigue estimation, a direct and non-simplified implementation of RFC in MPC is desirable and promises best results. To the authors' knowledge, the problem of rigorously implementing RFC within MPC was first solved by the same authors in Loew et al.,[18] using a novel method termed *Direct Online Rainflow Counting (DORFC)*. However, in Loew et al.,[18] only the core idea of the method was mentioned with a minimum level of detail, and brief simulation results were shown.

1.3 | Contribution

These gaps are closed in the present work. For the first time, a comprehensive presentation of DORFC is provided. The discussion covers the obstacles to the inclusion of RFC into MPC, the solution strategy, the required assumptions, and the inherent limitations of the resulting algorithm. Furthermore, mathematical and algorithmic details are provided here to facilitate a rapid implementation by interested readers. Finally, for the first time several interesting characteristics of DORFC are demonstrated with reference to a complex wind turbine control problem. These characteristics span over the domains of economic behavior, dynamic behavior, online damage estimation capabilities, switching behavior, and gradient dynamics.

1.4 | Outline

The standard procedure for off-line fatigue estimation is presented in Section 2. Next, Section 3 discusses an implementation of nonlinear MPC (NMPC), which is suitable for fatigue control. Section 4 provides a detailed explanation of the DORFC formulation. In order to give practical insights into DORFC, the method is applied to a wind turbine control problem in Section 5. The DORFC MPC is compared to a conventional MPC in Section 6, leading to an extensive demonstration and quantification of its performance. Finally, Section 7 presents the main conclusions and findings, and directions for further research.

2 | FATIGUE DAMAGE ESTIMATION

2.1 | Overview and governing quantities

Fatigue impact of a given stress time-series $\sigma(k)$ is characterized primarily by the amplitudes $\sigma_{\mathrm{a},c}$ of the N_c stress cycles c that the signal contains. Methods for identification of these stress cycles are presented in Section 2.2. For an overview, the cycle-governing quantities are collected in Table 2.

In addition to the amplitudes, stress mean values $\sigma_{\mathrm{m},c}$ of the cycles have to be considered, since typically a positive stress mean increases fatigue, whereas a negative stress mean has the opposite effect. In fatigue estimation, the frequencies and actual shapes of stress cycles are neglected. Therefore, all reviewed methods for cycle identification include the reduction of the stress time histories to a series of peaks and valleys (extrema). The calculation of fatigue damage is then based on the identified cycles, as shown in Section 2.3.

2.2 | Cycle identification

Cycle identification from stress time-series is straightforward if, for example, a simple sinusoid is analyzed. There, amplitude, mean value, and number of cycles are obvious. However, realistic stress trajectories are usually highly complex and contain stress cycles that can be nested.

In the ASTM-E1049 standard,[14] various methods are presented for the identification of stress cycles. These include methods for serial counting of stress-level crossings, stress extrema and stress ranges. For repetitive stress patterns, the *Reservoir Method* is a sophisticated and intuitive approach.[19] For general non-repetitive stress patterns, the *Rainflow Counting* algorithm (RFC) is considered as the most sophisticated method. Thus, RFC is chosen for the present work and is expressed as

$$[\sigma_{\mathrm{a},c}, \sigma_{\mathrm{m},c}, k_{\mathrm{max},c}, k_{\mathrm{min},c}, w_c] = RFC(\sigma(k)), \tag{1}$$

where the weight is set as $w_c = 1$ if a full stress cycle is detected, whereas $w_c = 0.5$ in the case of a half cycle. Full cycles are cosines of a full period, while half cycles are cosines of only a half period. Half cycles therefore represent either a rising or falling transient.

As shown in more detail in Appendix A, RFC contains algorithmic branches and loops, resulting in a discontinuous output behavior. Furthermore, the number N_c of identified cycles is not known before execution, but can only be bounded by the number of extrema.

2.3 | Damage estimation

After the identification of stress cycles, the total damage of a stress time-series can be computed as shown by the standard damage program reported in Figure 2a.

The effect of the cycle stress mean on fatigue is considered by correcting the stress amplitude to an equivalent stress via the *Goodman equation*[20(p. 184)]:

$$\sigma_{\mathrm{eq},c} = \sigma_{\mathrm{a},c} \frac{R_\mathrm{m}}{R_\mathrm{m} - \sigma_{\mathrm{m},c}}, \tag{2}$$

TABLE 2 Converted outputs of RFC for stress cycles c

Quantity	Variable	Unit
Stress amplitude	$\sigma_{\mathrm{a},c}$	Pa
Stress mean	$\sigma_{\mathrm{m},c}$	Pa
Sample index of cycle maximum	$k_{\mathrm{max,c}}$	-
Sample index of cycle minimum	$k_{\mathrm{min,c}}$	-
Weight	w_c	-

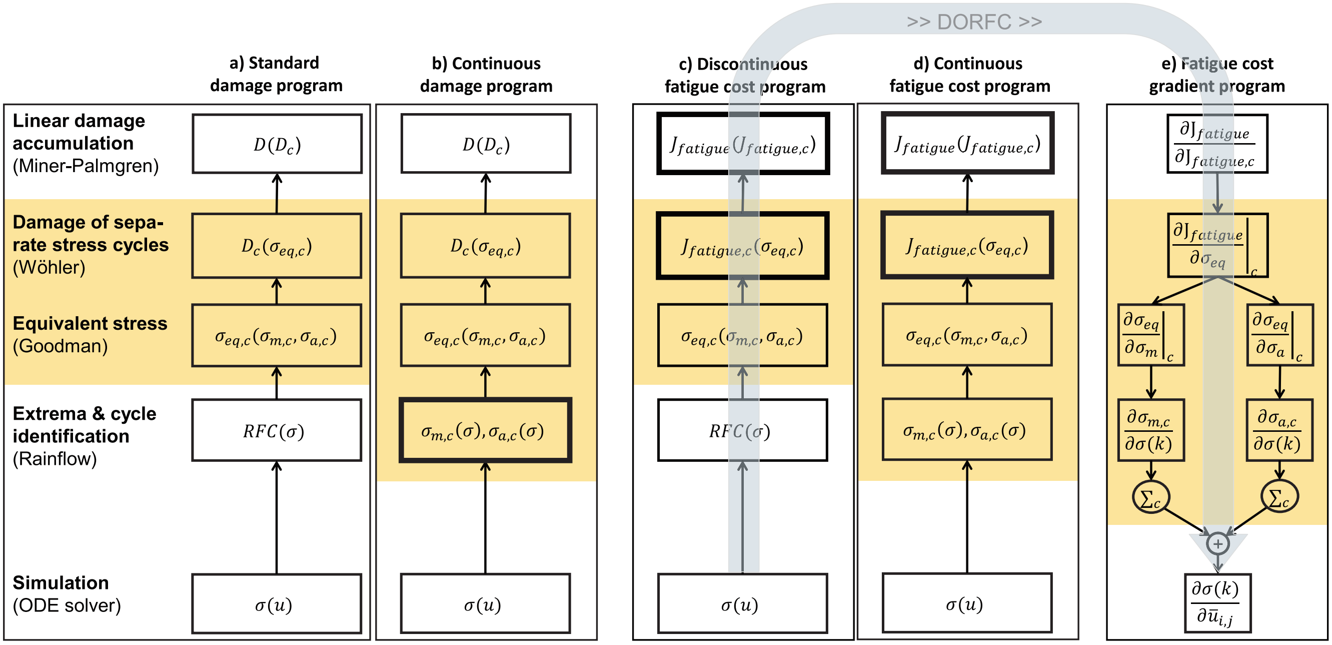
>> DORFC >>
a) Standard damage program
b) Continuous damage program
c) Discontinuous fatigue cost program
d) Continuous fatigue cost program
e) Fatigue cost gradient program
Linear damage accumulation (Miner-Palmgren)
Damage of separate stress cycles (Wöhler)
Equivalent stress (Goodman)
Extrema & cycle identification (Rainflow)
Simulation (ODE solver)
$D(D_c)$
$D_c(\sigma_{eq,c})$
$\sigma_{eq,c}(\sigma_{m,c},\sigma_{a,c})$
$RFC(\sigma)$
$\sigma(u)$
$\sigma_{m,c}(\sigma),\sigma_{a,c}(\sigma)$
$J_{fatigue}(J_{fatigue,c})$
$J_{fatigue,c}(\sigma_{eq,c})$
$\frac{\partial J_{fatigue}}{\partial J_{fatigue,c}}$
$\left.\frac{\partial J_{fatigue}}{\partial \sigma_{eq}}\right|_c$
$\left.\frac{\partial \sigma_{eq}}{\partial \sigma_m}\right|_c$
$\left.\frac{\partial \sigma_{eq}}{\partial \sigma_a}\right|_c$
$\frac{\partial \sigma_{m,c}}{\partial \sigma(k)}$
$\frac{\partial \sigma_{a,c}}{\partial \sigma(k)}$
$\sum_c$
$\sum_c$
$+$
$\frac{\partial \sigma(k)}{\partial \bar{u}_{i,j}}$

FIGURE 2 Standard damage program (a). Substitution of $RFC(\sigma)$ by linear continuous functions (16) yields the continuous damage program (b). Substitution of the damage functions D_c by fatigue cost functions $J_{\text{fatigue},c}$ yields the fatigue cost programs (c,d). Reverse automatic differentiation of (d) yields the fatigue cost gradient program (e). For all programs: Separate calculation for the individual stress cycles (colored background) and joint calculation (elsewhere). The grey arrow indicates the calculation flow of DORFC, where a forward execution of the "Discontinuous fatigue cost program" (c) is followed by a reverse execution of the "Fatigue cost gradient program" (e)

where R_m denotes the ultimate tensile stress.

Woehler curves, which can be obtained by ad hoc material experiments, provide the number N_fail of cycles to failure for specific equivalent stresses. Consequently, fatigue damage of one cycle is obtained by the reciprocal to the number N_fail of cycles to failure, that is,

$$D_{\text{fatigue},c} = \frac{1}{N_\text{fail}(\sigma_{\text{eq},c})}. \tag{3}$$

Assuming linear and time-invariant damage accumulation, the *Miner-Palmgren Rule*[21] is used to compute the total fatigue damage of the stress time series as

$$D_\text{fatigue} = \sum_c D_{\text{fatigue},c}. \tag{4}$$

3 | NMPC FORMULATION AND IMPLEMENTATION

The original NMPC problem is an optimal control problem over an infinite time horizon. Because this problem usually cannot be solved in closed form, an approximative solution is obtained by optimization over a shorter, moving horizon $[t_0, t_\text{end}]$.

3.1 | Dynamic system

The present work considers a continuous nonlinear system

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t)), \tag{5}$$

comprising the system states $\boldsymbol{x}(t)$, the control variables $\boldsymbol{u}(t)$, and external disturbances $\boldsymbol{d}(t)$. Fatigue is caused by stress $\sigma(t)$, which can be one of the states or outputs of the system.

3.2 | Economic formulation of revenue and fatigue

In order to provide meaningful optimal solutions, the concept of *Economic NMPC* (ENMPC) is pursued in the present work. An interesting discussion on economic nonlinear model predictive controller (ENMPC) can be found in Gros et al.[11] The core idea of ENMPC is the direct implementation of high-level operational objectives in the cost function. In many cases this is superior to the tracking of state trajectories often used in standard NMPC.

For the present application, the economic cost function J is expressed as

$$J(\boldsymbol{x}) = -J_{\text{revenue}}(\boldsymbol{x}) + \alpha_{\text{damage}} J_{\text{fatigue}}(\boldsymbol{x}). \tag{6}$$

This figure of merit is used to trade-off the maximization of revenue J_{revenue} and the minimization of fatigue cost J_{fatigue} by the controller. A weighting factor α_{damage} can be used to influence this trade-off. Revenue is formulated as a stage cost term

$$J_{\text{revenue}} = \int_{t_0}^{t_{\text{end}}} \dot{J}_{\text{revenue}}(\boldsymbol{x}(t))\,\mathrm{d}t. \tag{7}$$

The formulation of fatigue cost will be presented in Section 4.3.

3.3 | Formulation of the nonlinear program

The NMPC problem is transformed to a nonlinear program (NLP) by discretizing the time-continuous control variables $\boldsymbol{u}(t)$ into piecewise-constant optimization variables $\overline{u}_{i,j}$. Here, the type of control variable is determined by $i \in N_{\text{in}}$ and the control interval by $j \in N_u$. The number of control inputs is denoted by N_{in} and the number of control samples by $N_u = T_{\text{horizon}}/T_{\text{cntrl}}$, with horizon length $T_{\text{horizon}} = t_{\text{end}} - t_0$ and controller sample-time T_{cntrl}.

The NLP is defined by the minimization

$$\min_{\overline{u}} J(\boldsymbol{x}), \tag{8}$$

of the cost function (6) subject to the system dynamics (5), box constraints on the optimization variables

$$\boldsymbol{u}_{\min} \leq \overline{\boldsymbol{u}} \leq \boldsymbol{u}_{\max}, \tag{9}$$

and nonlinear inequality constraints

$$\boldsymbol{c}_{\text{ineq}}(\overline{\boldsymbol{x}}) = \begin{pmatrix} \overline{\boldsymbol{x}} - \boldsymbol{x}_{\max} \\ -\overline{\boldsymbol{x}} + \boldsymbol{x}_{\min} \end{pmatrix} \leq \boldsymbol{0}, \tag{10}$$

on the system states $\overline{x}_{k,j}$, which are sampled at the control intervals $j \in N_u$. Here, the individual states are represented by $k \in N_x$ with the total number of states N_x.

The above hard nonlinear inequality constraints are sufficient in a setting without noise or plant-model mismatch, as true for the exemplary application of the present work. In the case of significant noise, soft constraints have to be employed to ensure feasibility.[22]

In the present work, the *Sequential Approach* is applied, where the dynamics are solved in a single-shooting fashion before the optimization step. This results in a small-size optimization problem.[23] The simulated dynamics are expressed as

$$\dot{\boldsymbol{X}} = \left[\dot{\boldsymbol{x}}, \dot{J}_{\text{revenue}}, \frac{\mathrm{d}\dot{\boldsymbol{x}}}{\mathrm{d}\overline{\boldsymbol{u}}}, \frac{\mathrm{d}\dot{J}_{\text{revenue}}}{\mathrm{d}\overline{\boldsymbol{u}}}\right]^T, \tag{11}$$

and involve four components: the ODEs of the reduced plant system $\dot{\boldsymbol{x}}$, given by (5), and of the revenue term $\dot{J}_{\text{revenue}}$, given by (7), as well as the variational differential equations (VDEs) for the state gradients $\frac{\mathrm{d}\dot{\boldsymbol{x}}}{\mathrm{d}\overline{\boldsymbol{u}}}$ and the revenue gradients $\frac{\mathrm{d}\dot{J}_{\text{revenue}}}{\mathrm{d}\overline{\boldsymbol{u}}}$. Note that the VDEs make up the major portion of the simulated dynamics since, for example, for the state gradients, each state has to be derived w.r.t. each individual control sample, and thus the dimension is $\frac{\mathrm{d}\dot{\boldsymbol{x}}}{\mathrm{d}\overline{\boldsymbol{u}}} \in \mathbb{R}^{N_x \cdot N_{\text{in}} \cdot N_u}$. More details on the present formulation are provided in Löw et al.[24]

3.4 | Real-time implementation

Sequential Quadratic Programming (SQP) is a common method for solving NLPs. Here, the NLP is sequentially approximated by QPs, which is repeated until full convergence and can be very time-consuming. Thus, the present implementation is based on the *Real-Time Iteration* (RTI),[25] which is inspired by SQP but only solves one *Quadratic Programming* (QP) problem at each MPC step, as shown in Figure 3. RTI is based on the assumption that the optimization problems vary slowly over time. Thus, the QPs in subsequent MPC steps approximate an SQP.

Solving only one QP per MPC step implies that full convergence is in general not achieved. However, starting from a good initialization at the first MPC step, each new operating point is close to the previous one, and each new optimization solution exhibits only limited suboptimality (see Diehl et al.[26] and Gros et al.[25] for further details).

Each QP problem is formulated at the current operating point $(\boldsymbol{x}^*, \overline{\boldsymbol{u}}^*, \boldsymbol{d}^*)$, where the cost is approximated by a quadratic function as

$$\min_{\Delta\overline{\boldsymbol{u}}} \frac{1}{2} \Delta\overline{\boldsymbol{u}}^T \boldsymbol{H}^* \Delta\overline{\boldsymbol{u}} + (\boldsymbol{g}^*)^T \Delta\overline{\boldsymbol{u}}. \tag{12}$$

Here $\boldsymbol{H}^*$ is the Hessian, and $\boldsymbol{g}^*$ is the gradient of the cost function w.r.t. the optimization variables $\overline{\boldsymbol{u}}$. Each QP yields an update

$$\overline{\boldsymbol{u}}_{\text{new}} = \overline{\boldsymbol{u}}^* + \Delta\overline{\boldsymbol{u}}, \tag{13}$$

of the optimization variables.

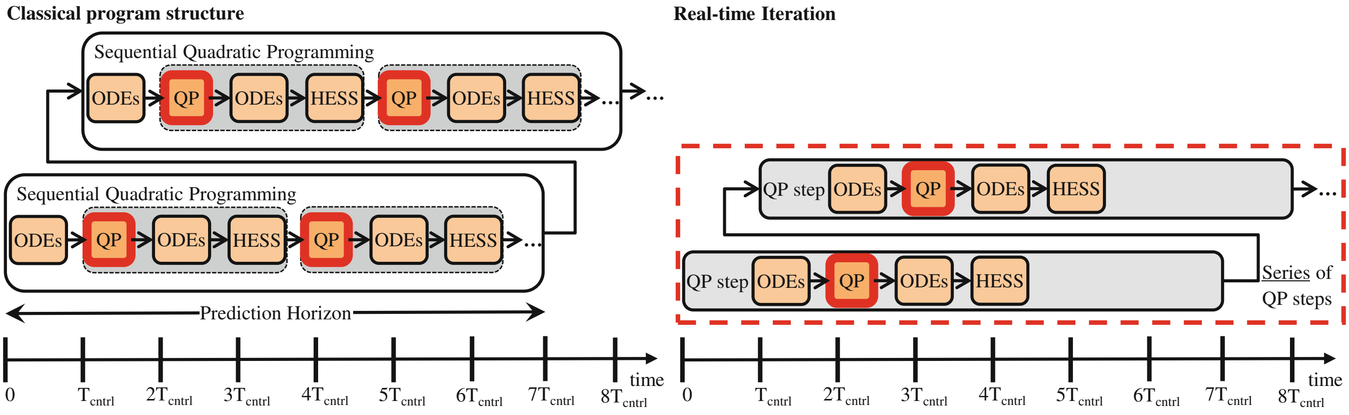

FIGURE 3 Comparison of the classical MPC program structure and the RTI method. HESS, update of the Hessian matrix; ODEs, numerical integration of the dynamics; QP, solution of the QP problem

In the present work, the Hessian $\boldsymbol{H}^*$ of the QP problem is approximated at each control step according to the *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) method.[23,27] The BFGS method only requires the current optimization variables and cost gradients, and the Hessian does not have to be explicitly calculated.

The gradients of the cost function (6) w.r.t. all optimization variables are written as

$$\boldsymbol{g}^* = \frac{\mathrm{d}J^*}{\mathrm{d}\overline{\boldsymbol{u}}^*} = \boldsymbol{g}^*_{\text{revenue}} + \boldsymbol{g}^*_{\text{fatigue}}, \tag{14}$$

and are obtained separately for revenue and fatigue cost. The gradients for the cumulative revenue stage cost write

$$\boldsymbol{g}^*_{\text{revenue}} = \frac{\mathrm{d}J^*_{\text{revenue}}(t_{\text{end}})}{\mathrm{d}\overline{\boldsymbol{u}}^*}, \tag{15}$$

and are only required at the end $t = t_{\text{end}}$ of the prediction horizon. These gradients are obtained via the corresponding VDE in (11). The formulation of the fatigue cost gradients will be presented in Section 4.4.

For the present implementation, the algorithms for solving the dynamics and the QP will be stated in Section 5.3.

4 | DIRECT ONLINE RAINFLOW COUNTING

4.1 | Obstacles

According to the literature, the following obstacles stand in the way of a direct implementation of RFC in MPC:

Obstacle 1: RFC requires the entire stress trajectory over the prediction horizon as an input,[3] and thus cannot be cast into standard stage or terminal cost functions. Stage costs would comprise a summation of state-samples or a time-integral of state-trajectories over the prediction horizon. Terminal costs would be defined as functions of the state samples at the end of the prediction horizon.[28]

Obstacle 2: It does not seem possible to derive analytical expressions of fatigue cost gradients $\boldsymbol{g}^*_{\text{fatigue}}$, due to the RFC algorithm not having a "closed mathematical form"[2,29-31]

Obstacle 3: RFC is a highly nonlinear algorithm,[15] which exhibits discontinuous outputs because of its branches and loops (see Section 2.2).

4.2 | Solution strategy

The above mentioned obstacles can be overcome by the combined application of three principles, which are labeled here "Separate", "Substitute", and "Switch seldom":

"Separate": Obstacle 1, which amounts to a "non-standard cost function", can be overcome if in the MPC implementation the solution of the ODEs and of the QP are separated and serialized, as visualized in the upper part of Figure 4. This is enabled in the present MPC implementation by exploiting the following properties of the *Sequential Approach* (see Section 3.3):

- Trajectories of states x and state gradients $\mathrm{d}x_l(k)/\mathrm{d}\overline{u}_{i,j}$ are obtained by an ODE solver using single-shooting, and thus are available before the QP solution (see Section 3.3).
- Cost gradients $\mathrm{d}J_{\text{fatigue}}(t_{\text{end}})/\mathrm{d}\overline{\boldsymbol{u}}$ need to be determined only at the end of the prediction horizon, but not over the prediction horizon itself (see Section 3.4).

Via the insertion of code in between the QP steps shown in the lower part of Figure 4, the rainflow algorithm is executed *(directly)* without approximations and *(online)* within the MPC. This justifies the use of the terms *Direct Online* in the DORFC name of the algorithm.

"Substitute": Obstacle 2 refers arguably to an absence of a "closed mathematical form", which is questioned in the present work. As shown in Figure A1 in the Appendix, the algorithmic loops in RFC are always executed a finite number of times. Thus, RFC can actually be classified as a "closed" algorithm.

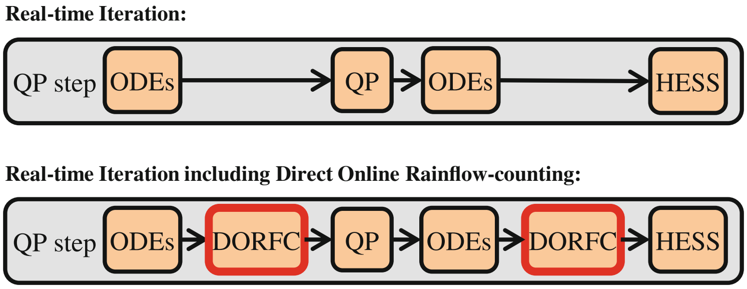

FIGURE 4 Integration of DORFC into the QP steps of the RTI method. HESS, update of Hessian Matrix; ODEs, numerical integration of the ordinary differential equations; QP, solution of a QP problem

Instead, in the present work, the argument of a non-closed form is interpreted to originate from the fact that the discrete *execution structure* of the RFC algorithm is highly dependent on the input, and thus is not known a priori.

Definition (Execution structure): *The execution structure is composed of the set of consecutive branch decisions.*

The input-dependency of the execution structure originates from two properties of RFC:

1. The input stress time-series is reduced to extrema, which clearly depend on the stress time-series itself.
2. The branches and loops in the cycle-counting procedure depend on the locations and values of these extrema.

The input-dependent execution structure of the RFC algorithm still poses a very challenging obstacle, and probably still blocks any attempt at deriving gradients of $RFC(\sigma)$. However, for many practical applications (including the one shown in Section 6.4), it is reasonable to hypothesize that:

- The execution structure only changes gradually over time.
- Consequently, the execution structure—and thus also the output *cycle structure*—can be assumed to remain fixed within each MPC step.

Definition (Cycle structure): *The cycle structure is composed of the number of cycles N_c, cycle weights w_c, and cycle samples $(k_{\mathrm{max},c}, k_{\mathrm{min},c})$.*

The assumption of a fixed structure allows for the substitution of $RFC(\sigma)$ by continuous expressions, and the transformation from a discontinuous to a continuous (and differentiable) damage program, as shown in Figure 2.

The substitution is performed for the stress mean $\sigma_{\mathrm{m},c}$ and stress amplitude $\sigma_{\mathrm{a},c}$ for separate cycles (see Table 2), by using their related maximum and minimum stress samples, which leads to

$$\sigma_{\mathrm{m},c} = \frac{\sigma(\hat{k}_{\mathrm{max},c}) + \sigma(\hat{k}_{\mathrm{min},c})}{2} \quad \text{and} \quad \sigma_{\mathrm{a},c} = \frac{\sigma(\hat{k}_{\mathrm{max},c}) - \sigma(\hat{k}_{\mathrm{min},c})}{2}. \tag{16}$$

These expressions are continuous and linear, and are assumed to be locally valid for one MPC step.

To conclude, each execution of DORFC consists first of the evaluation of the “Discontinuous fatigue cost program” based on RFC, and second of the evaluation of the “Fatigue cost gradient program” based on the continuous substitutes, as shown in Algorithm 1. The computation steps of the fatigue cost program will be specified in Section 4.3, and of the fatigue cost gradient program in Section 4.4.

“Switch seldom”: Obstacle 3, which refers to the “discontinuous output” of the algorithm, is especially relevant for the present work, since the RTI method is used. In fact, RTI requires the cost function and its gradients to only change mildly for successive MPC-steps. This can be achieved if:

Algorithm 1. MPC using DORFC

```
Input: measurement of system states, initial guess of control variables
Output: update of control variables
while true do
    1: Receive new initial states from measurement
    2: ODEs: Solve dynamics (11) via single-shooting,using current initial states and control trajectories
    3: DORFC(1): Evaluate "Discont. fatigue cost program", using rainflow algorithm (1) on stress trajectory
    4: DORFC(2): Evaluate "Fatigue cost gradient program", using continuous stubstitutes (16)
    5: QP: Set up and solve QP
    6: Apply new control variables to plant
    7: ODEs + DORFC: Execute steps 2: to 4:,using current initial states and new control trajectories
    8: HESS: Update Hessian matrix via BFGS method,using old and new control trajectories and cost gradients
end
```

- The sample-time T_{cntrl} of the controller is appropriately low w.r.t. the time constants of the system dynamics (5).
- The discontinuities in RFC switch sufficiently seldom.

This requirement of seldom switches can be linked to the concept of “average dwell-time” in switching cost functionals.[32] In a nutshell, this concept implies that an MPC with switching cost function can remain stable if, on average, there is a sufficient time period between the switching events. Since this requirement only holds “on average”, a longer time period without switches may be followed by a time phase of frequent switches. A more precise theoretical analysis is out of scope of the present paper. However, studies using an exemplary system in Section 6 show that indeed changes in the execution structure of RFC happen infrequently enough over the simulation time.

4.3 | Continuous fatigue cost program

In order to obtain monetary cost functions in the ENMPC, the damage programs are transformed into fatigue cost programs as shown in Figure 2 (programs (a) into (c), and (b) into (d)). This transformation requires a continuous expression for the cost $J_{\text{fatigue},c}$ of separate stress cycles c w.r.t. the equivalent stress $\sigma_{\text{eq},c}$. The following criteria are proposed for this analytical expression:

- A zero equivalent stress should result in a zero cost, that is,

$$J_{\text{fatigue},c}(\sigma_{\text{eq},c} = 0) = 0. \tag{17}$$

- There are no healing effects in standard fatigue models. Thus, negative fatigue costs should never be obtained for all possible values of the equivalent stress, that is,

$$J_{\text{fatigue},c}(\sigma_{\text{eq},c}) \geq 0, \tag{18}$$

  for every $\sigma_{\text{eq},c} \in \mathbb{R}_0^+$. Notice that the *Goodman equation* (2) does not produce negative equivalent stresses, since the stress mean $\sigma_{\text{m},c} < R_{\text{m}}$ must not exceed the ultimate tensile stress limit.
- Due to the use of gradient-based optimization, the expression has to be continuously differentiable on $\sigma_{\text{eq},c} \in \mathbb{R}_0^+$.
- Due to the use of a QP formulation, the expression has to be convex, that is,

$$J_{\text{fatigue},c}(\lambda\sigma_{\text{eq},c1} + (1-\lambda)\sigma_{\text{eq},c2}) \leq \lambda J_{\text{fatigue},c}(\sigma_{\text{eq},c1}) + (1-\lambda)J_{\text{fatigue},c}(\sigma_{\text{eq},c2}), \tag{19}$$

  for every $\lambda \in [0, 1]$ and $\sigma_{\text{eq},c1}, \sigma_{\text{eq},c2} \in \mathbb{R}_0^+$.

Polynomials are one class of analytical expressions that fulfills the above mentioned criteria, leading to the following expression for the cost of a cycle:

$$J_{\text{fatigue},c}(\sigma_{\text{eq},c}) = w_c \sum_{n=2}^{N} a_n \; \sigma_{\text{eq},c}^{n} \tag{20}$$

with a minimum order of $n \geq 2$ and non-negative coefficients $a_n \geq 0$. The cost is weighted by $w_c = 1$ if a full stress cycle is detected, and by $w_c = 0.5$ in the case of a half cycle. The polynomial order and coefficients can be adjusted to various economic fatigue models, which for example can comprise:

- Initial capital cost;
- Opportunity cost of lost remaining lifetime;
- Safety-margin for a guaranteed lifetime.

For the mechanical case, these cost models should be based on a relationship between applied stress and cycles-to-failure, such as the one provided by Woehler curves.

Following the linear damage accumulation approach by *Miner Palmgren*,[21] the total fatigue cost of a specific stress time series is obtained by the linear accumulation of fatigue cost of all detected cycles c, that is,

$$J_{\text{fatigue}} = \sum_{c} J_{\text{fatigue},c}. \tag{21}$$

4.4 | Fatigue cost gradient program

Differentiation of the continuous fatigue cost program yields the fatigue cost gradient program (Figure 2, program (d) into (e)), which can be written as

$$\frac{\mathrm{d}J_{\text{fatigue}}}{\mathrm{d}\overline{u}_{i,j}} = \frac{\partial J_{\text{fatigue}}}{\partial J_{\text{fatigue},c}} \operatorname{diag}\left(\frac{\partial J_{\text{fatigue},c}}{\partial \sigma_{\text{eq},c}}\right) \left[\operatorname{diag}\left(\frac{\partial \sigma_{\text{eq},c}}{\partial \sigma_{\text{m},c}}\right) \frac{\partial \sigma_{\text{m},c}}{\partial \sigma(k)} + \operatorname{diag}\left(\frac{\partial \sigma_{\text{eq},c}}{\partial \sigma_{\text{a},c}}\right) \frac{\partial \sigma_{\text{a},c}}{\partial \sigma(k)}\right] \frac{\mathrm{d}\sigma(k)}{\mathrm{d}\overline{u}_{i,j}} \tag{22}$$

with Jacobians $\frac{\mathrm{d}J_{\text{fatigue}}}{\mathrm{d}\overline{u}_{i,j}} \in \mathbb{R}^{N_{\text{in}} \cdot N_u}$, $\frac{\partial J_{\text{fatigue}}}{\partial J_{\text{fatigue},c}} \in \mathbb{R}^{N_c}$, $\frac{\partial J_{\text{fatigue},c}}{\partial \sigma_{\text{eq},c}} \in \mathbb{R}^{N_c}$, $\frac{\partial \sigma_{\text{eq},c}}{\partial \sigma_{\text{m},c}} \in \mathbb{R}^{N_c}$, $\frac{\partial \sigma_{\text{eq},c}}{\partial \sigma_{\text{a},c}} \in \mathbb{R}^{N_c}$, $\frac{\partial \sigma_{\text{m},c}}{\partial \sigma(k)} \in \mathbb{R}^{N_c \times N_u}$, $\frac{\partial \sigma_{\text{a},c}}{\partial \sigma(k)} \in \mathbb{R}^{N_c \times N_u}$ and $\frac{\mathrm{d}\sigma(k)}{\mathrm{d}\overline{u}_{i,j}} \in \mathbb{R}^{N_u \times N_{\text{in}} \cdot N_u}$.

Gradients of functions and algorithms ("programs") generally can be obtained by numerical differentiation, symbolic differentiation, or automatic differentiation (AD). Numerical gradients of nonlinear programs, for example, via finite differences, can be expensive and may result in loss of accuracy.

Symbolic gradients, on the other hand, are also unsuitable for the present problem: either a large number of symbolic gradients would have to be pre-computed offline for all possible execution structures of the rainflow algorithm, or the symbolic gradients would have to be composed online by computer algebra. The latter method would need to be executed at each controller step, which can be expensive for real-time applications.

In contrast, AD obtains exact numeric gradients online in a step-wise manner by following the chain rule. The forward execution of the original program (Figure 2c) determines which steps are taken, that is, which execution structure of RFC can be currently observed. Using the numerical results of each step of the program, gradients are obtained either by *Forward AD* or by *Reverse AD*. Simply put, these two modes stem from different applications of the associative and distributive law, and differ in efficiency depending on the problem. In the present work the fatigue program only has one output, represented by total fatigue cost J_{fatigue}, and a large number of input optimization variables $\overline{u}_{i,j}$. In such situations, *Reverse AD* is typically more efficient.[33] Accordingly, calculations are executed top-down as shown in the "Fatigue cost gradient program" in Figure 2e.

AD requires a library of gradients either for elementary mathematical operations or, as in the present case, for the chain elements of the differentiated program (22). In the following, analytical gradients for all chain elements are presented:

- Since linear damage accumulation (21) is used, gradients of total fatigue cost w.r.t. to fatigue cost of separate stress cycles result in

$$\frac{\partial J_{\text{fatigue}}}{\partial J_{\text{fatigue},c}} = 1. \tag{23}$$

- The gradient of fatigue cost of separate cycles (20) w.r.t. equivalent stress is obtained analytically and yields

$$\frac{\partial J_{\text{fatigue},c}}{\partial \sigma_{\text{eq},c}} = w_c \sum_{n=2}^{N} n\, a_n\, \sigma_{\text{eq},c}^{n-1}. \tag{24}$$

  A gradient w.r.t. weight w_c is not required, since w_c is part of the cycle structure and thus fixed for the current MPC step.
- Partial differentiation of the *Goodman equation* (2) yields the gradient of equivalent stress w.r.t. stress mean

$$\frac{\partial \sigma_{\text{eq},c}}{\partial \sigma_{\text{m},c}} = \sigma_{\text{a},c} \frac{R_{\text{m}}}{(\sigma_{\text{m},c} - R_{\text{m}})^2} \quad \text{and stress amplitude} \quad \frac{\partial \sigma_{\text{eq},c}}{\partial \sigma_{\text{a},c}} = \frac{R_{\text{m}}}{R_{\text{m}} - \sigma_{\text{m},c}}. \tag{25}$$

- The continuous replacement functions (16) enable the direct calculation of the gradient of mean value

$$\frac{\partial \sigma_{\text{m},c}}{\partial \sigma(k)} = \begin{cases} \frac{1}{2} & \text{if } k = k_{\text{max},c} \\ \frac{1}{2} & \text{if } k = k_{\text{min},c} \\ 0 & \text{otherwise} \end{cases} \quad \text{and amplitude value} \quad \frac{\partial \sigma_{\text{a},c}}{\partial \sigma(k)} = \begin{cases} \frac{1}{2} & \text{if } k = k_{\text{max},c} \\ -\frac{1}{2} & \text{if } k = k_{\text{min},c} \\ 0 & \text{otherwise} \end{cases}, \tag{26}$$

  w.r.t. the stress trajectory samples. It should be noted that each row of the resulting Jacobians will contain two non-zero entries, since each stress cycle c is defined by two stress samples. For instance, for $c = 3$ identified stress cycles and $k = 6$ stress samples, the Jacobian $\partial \boldsymbol{\sigma}_{\text{a}} / \partial \sigma$ will take the following form:

$$\begin{pmatrix} \cdot & -\frac{1}{2} & \cdot & \frac{1}{2} & \cdot & \cdot \\ \frac{1}{2} & -\frac{1}{2} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \frac{1}{2} & \cdot & -\frac{1}{2} \end{pmatrix}.$$

- The stress gradients $\frac{\mathrm{d}\sigma(k)}{\mathrm{d}\bar{u}_{i,j}} \subset \frac{\mathrm{d}x_l(k)}{\mathrm{d}\bar{u}_{i,j}}$ w.r.t. the optimization variables are subsets of the state gradients (Section 3.3), and thus are obtained by numerical integration.

5 | EXEMPLARY APPLICATION

Performance of MPC with DORFC is evaluated using the reference wind turbine developed within *IEA Wind Task 37*, as sketched in Figure 5. A detailed presentation of the model is provided in Appendix B.

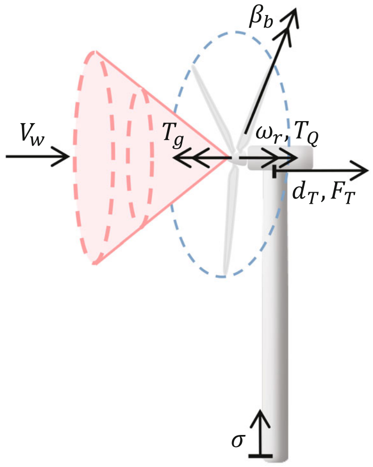

FIGURE 5 Sketch of the LiDAR-equipped wind turbine model with ambient wind speed V_{w}, generator torque T_{g}, rotational speed of the rotor ω_{r}, aerodynamic torque T_{Q}, displacement of tower tip d_{T}, aerodynamic thrust force F_{T}, collective blade pitch angle β_{b} and tensile stress at tower root σ

5.1 | Idealizations

The present work is focused on the assessment of the key characteristics of DORFC. In order to limit the complexity of the analysis, the following idealizations are used:

- The simulation (plant) model is the same as the internal MPC model, that is, there is no model mismatch.
- All system states (B1) are assumed to be measured exactly and without noise.
- An exact prediction of the wind is assumed within the MPC horizon.
- The MPC is assumed to operate without execution delays. This implies that acquired measurement signals are synchronous with output control signals.
- No mechanical and electrical energy-losses are considered.

Particularly the first three hypotheses neglect noise, biases, and uncertainties, thereby eliminating the need for a state estimator and soft constraints in the MPC. A more realistic and complete simulation and controller setup without the above idealizations is part of ongoing research.

5.2 | Cost functions

The primary objective of wind turbine operation below rated wind speed is to maximize the extraction of energy from wind, while respecting various system constraints. This is mainly achieved by adjusting the generator torque, while the variation of blade pitch angle typically is low or zero. Above rated wind speed, the goal is to track the rated electrical power, mainly by varying the pitch angle. However, perfect tracking is not possible due to the variability of the exciting wind. For instance, for a drop of wind speed, the pitch angle has to follow as rapidly as possible in order not to excessively affect energy extraction. This behavior can be achieved either by a power tracking cost function,[34] or by the combination of energy maximization and a power upper constraint. The latter was chosen in the present work. Consequently, the cost function (6) of power maximization and fatigue cost minimization is universally effective over all wind speed regions, from cut-in to cut-out wind speed.

The interaction of the rotor with the wind induces a thrust force that —among other effects— excites fore-aft oscillations of the tower. These oscillations result in cyclic stress and thus in fatigue at the tower base. To mitigate this effect, the turbine controller can be designed to achieve a second objective, that is, the minimization of fatigue. Clearly, fatigue damage occurs at other wind turbine components, and may drive their design. For simplicity, however, the present work only considers fatigue at tower base.

In the following, the above mentioned objectives are cast into the economic cost function (6) for the MPC formulations TTVP and DORFC:

(1) For TTVP, energy-based cost terms are considered. Therefore, following Gros et al.,[11] revenue is expressed as an integral of extracted aerodynamic power

$$J_{\mathrm{revenue}} = \int_{t_0}^{t_{\mathrm{end}}} P_{\mathrm{A}}(t)\,\mathrm{d}t. \tag{27}$$

Fatigue cost is represented by the average kinetic energy of tower oscillations, which is obtained as

$$J_{\mathrm{fatigue}} = \int_{t_0}^{t_{\mathrm{end}}} \frac{1}{2T_{\mathrm{horizon}}} m_{\mathrm{T}} \dot{d}_{\mathrm{T}}^{2}\,\mathrm{d}t, \tag{28}$$

using the lumped mass of the tower m_{T} and a normalization by the horizon length T_{horizon}.

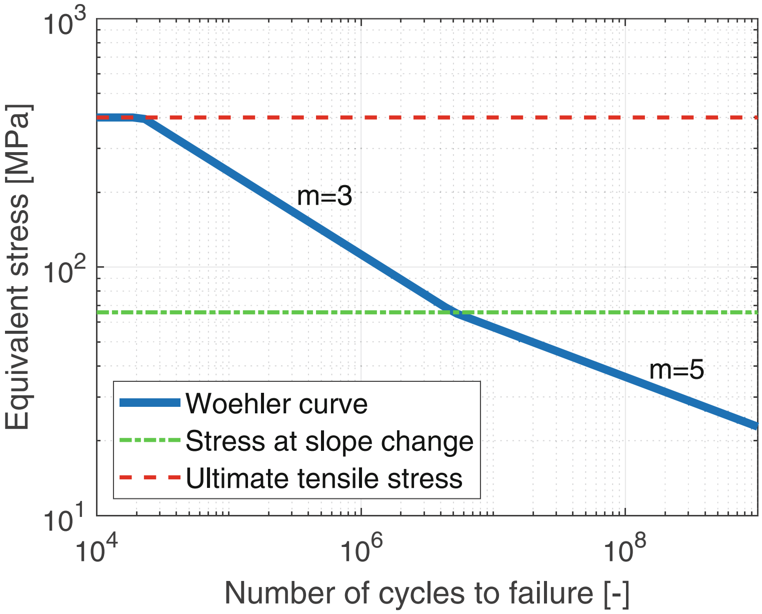

FIGURE 6 Woehler curve and characteristic stress levels in double-logarithmic representation

(2) For DORFC, monetary cost terms are considered. Thus, revenue cost within the horizon is obtained as

$$J_{\text{revenue}} = \int_{t_0}^{t_{\text{end}}} \frac{P_A(t) J_{\text{elec,kWh}}}{1000 \cdot 3600}\, \mathrm{d}t, \tag{29}$$

for a fixed electricity price of $J_{\text{elec,kWh}} = 0.1$ €/kWh. By the term at the denominator, the electricity price is converted to SI-units [€/(Ws)].

Fatigue cost is based on the Woehler curves

$$\log_{10}(\sigma_{\text{eq},c}) = -\frac{1}{m}(\log_{10} N_{\text{fail}} - \log_{10} N_{slo}) + \log_{10} \sigma_{\text{eq},slo}, \tag{30}$$

which are a function of the number of cycles to failure N_{fail}. The specific curve for the present wind turbine tower is obtained by the Woehler parameters of Table 3, and it is displayed in Figure 6. The damage model is obtained from the Woehler curve shown in (3). Using this damage model and an exemplary initial capital cost (ICC) $J_{\text{machine}} = 4 \cdot 10^6$ €, the Woehler-based fatigue cost of separate stress cycles can be calculated to be

$$\tilde{J}_{\text{fatigue},c}(\sigma_{\text{eq},c}) = J_{\text{machine}} \frac{1}{N_{\text{fail}}(\sigma_{\text{eq},c})}, \tag{31}$$

as shown in Figure 7.

TABLE 3 Material properties of the wind turbine tower steel

Quantity	Unit	Value
Yield stress R_p	MPa	355
Ultimate tensile stress R_{m}	MPa	400
Stress $\sigma_{\text{eq},slo}$ at slope change of Woehler curve	MPa	65.7
Number N_{slo} of cycles at slope change	-	$5 \cdot 10^6$
Exponent m for high-stress part of Woehler curve	-	3
Exponent m for low-stress part of Woehler curve	-	5

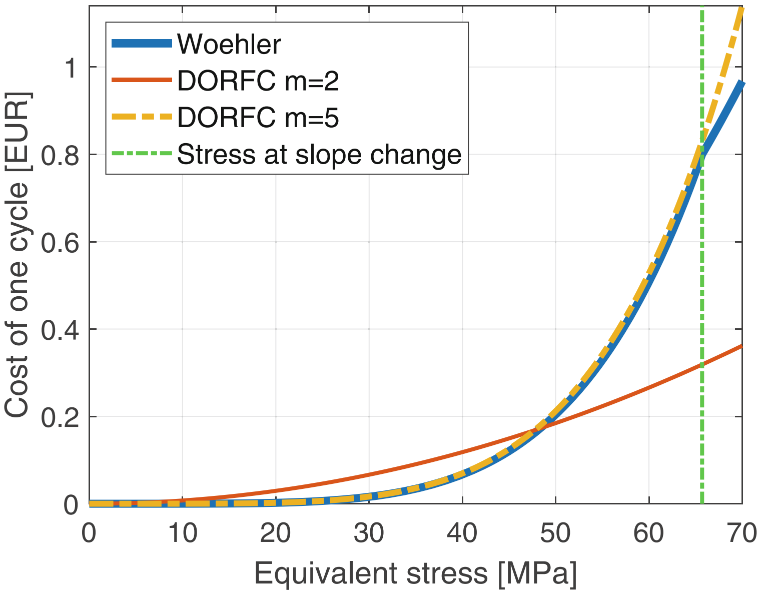

FIGURE 7 Cost of one stress cycle based on Woehler data, for approximations with exponent $m = 2$ and $m = 5$

For the definition of the MPC problem, a polynomial fatigue cost function according to (20) is required. An approximation by a 5th-order monomial is used here, resulting in the expression

$$J_{\text{fatigue},c}(\sigma_{\text{eq},c}) = w_c \; a_5 \sigma^5_{\text{eq},c} \tag{32}$$

with coefficient $a_5 = 6.79 \cdot 10^{-10}$. Due to this parameterization, the approximation is always equal to or above the original Woehler-based fatigue cost function, as shown in Figure 7. The corresponding MPC formulation is termed in the following DORFC-5. Due to the use of QPs, a second cost function based on a 2nd-order monomial is also tested, resulting in the expression

$$J_{\text{fatigue},c}(\sigma_{\text{eq},c}) = w_c \; a_2 \sigma^2_{\text{eq},c} \tag{33}$$

with coefficient $a_2 = 7.38 \cdot 10^{-5}$. Using this value of the coefficient, the *gradient* of (33) matches the gradient of the Woehler-based fatigue cost function at the frequently occurring equivalent stress level of $\sigma_{\text{eq},c} = 35$ MPa. The corresponding MPC formulation is termed in the following DORFC-2.

5.3 | Controller and simulation settings

For the present work, an in-house developed MPC implementation is used, where the dynamics are solved by a 4th-order Runge–Kutta integrator, and the QPs by the interior-point algorithm of `quadprog`.[35] A controller sample-time of $T_{\text{cntrl}} = 0.2$ s, a fixed integrator step-time of $T_{\text{integr}} = 0.005$ s and a prediction horizon of $T_{\text{horizon}} = 8$ s are set. This horizon length was chosen to obtain several stress cycles within the prediction horizon. Considering that Lidars are capable of scanning at distances of 200 m and more in front of the turbine,[36,37] 10 s long wind predictions are available without the need for extrapolation, even at high wind speeds of 20 m/s.

The number of control variables within the horizon is $N_u = T_{\text{horizon}}/T_{\text{cntrl}} = 40$, since the control horizon equals the prediction horizon. The present implementation is not capable of real-time execution with these settings, but achieves real-time performance with half the horizon length on a standard workstation equipped with an Intel® Core i7-6820HQ CPU and a 16 GB DDR4 RAM.[18] Each individual simulation is conducted using 645 s of turbulent hub-height wind data, where the first 30 s and the last 15 s are neglected in the post-processing phase.

Since the controller and plant models are identical in the present implementation and all measurements are exact and without delays, it was verified that all MPCs are able to almost exactly eliminate all tower oscillations when a smooth rotor-equivalent wind speed is used as input. To avoid this situation, the more turbulent hub-height wind speed signal is used, which essentially amounts to the introduction of a stochastic disturbance in the controller.

6 | RESULTS AND DISCUSSION

Three different MPC formulations are applied to the exemplary wind turbine problem: the conventional TTVP, and the two versions DORFC-2 and DORFC-5 of the novel method, respectively with an approximate and a physically correct fatigue exponent of $m = 2$ and $m = 5$.

For the strongly convex cost function of DORFC-5, the MPC initially exhibited a noisy behavior and deteriorated performance. This effect was eliminated by reducing the step length of the QP output from the standard $\Delta\bar{\boldsymbol{u}}$ to $0.3\Delta\bar{\boldsymbol{u}}$.

6.1 | Economic controller behavior

The tuning of the fatigue weight α_{damage} and the economic behavior over different reference wind speeds has already been discussed in Loew et al.[18] Therefore, only a brief summary is provided here.

The resulting fatigue cost $J_{\text{fatigue,real}}$ of the plant is obtained by post-processing via offline RFC. Fatigue damage is converted into fatigue cost by multiplying by an ICC of $4 \cdot 10^6$ €. Energy capture is converted into revenue by multiplying (after appropriate unit conversion) by an electricity price of 0.1 €/kWh. Profit is revenue minus fatigue cost.

6.1.1 | Performance as a function of tuning weights

The present turbine has a rated wind speed of $V_{\text{w,rated}} = 9.8$ m/s. Tuning of the controller was obtained by considering Design Load Case (DLC) 1.1[38] at a wind speed of 9 m/s. This wind speed was chosen because it is close to the rated one, and thus ensures frequent transitions between the partial and the full load operating regions.

As shown in Figure 8, maximum profit is reached for TTVP at a fatigue weight of $\alpha_{\text{damage}} = 2000$, for DORFC-2 at $\alpha_{\text{damage}} = 1$, and for DORFC-5 at $\alpha_{\text{damage}} = 10$. It should be noted that the weights for DORFC are in the order of $O(1)$, demonstrating that the monetary cost formulation renders tuning easier or almost unnecessary.

6.1.2 | Performance as a function of wind speed

As shown in Figure 9, DORFC reaches higher revenue and profit for almost all wind speeds. However, just as in Loew et al.,[18] these advantages remain below 1%. The reason mainly lies in the perfect match of the MPC-internal and plant models used in the present simulation setting. Because of this, all MPC formulations are able to suppress tower oscillations without significantly reducing energy capture. Furthermore, the low tower stress oscillations lead to a fatigue cost that is about 3 orders of magnitude smaller than the revenue.

Both DORFC formulations exhibit a similar profit behavior. However, this is achieved by different strategies. While DORFC-2 increases revenue only to a moderate extent and decreases fatigue cost by up to 10%, DORFC-5 tolerates significantly higher fatigue in order to maximize revenue.

Regarding actuator usage, DORFC-2 exceeds TTVP in numerous velocity ranges, while DORFC-5 decreases generator torque travel by about 20% and blade pitch travel by about 10% on average.

6.2 | Dynamic controller behavior

For the MPC without fatigue penalization, there are severe oscillations in the tower root stress, as shown in Figure 10. When the conventional cost formulation of TTVP is used, a good damping of turbine tower oscillations is achieved. By using DORFC, an unusual further modification of the stress trajectory is achieved: in fact, plateaus in the stress trajectory are visible in Figure 10, for example, at $t = 82$ s and $t = 89$ s. Although these plateaus only result in a minor reduction of stress cycle amplitudes, the effect on fatigue damage is strong due to the superlinear relation of stress to fatigue. These

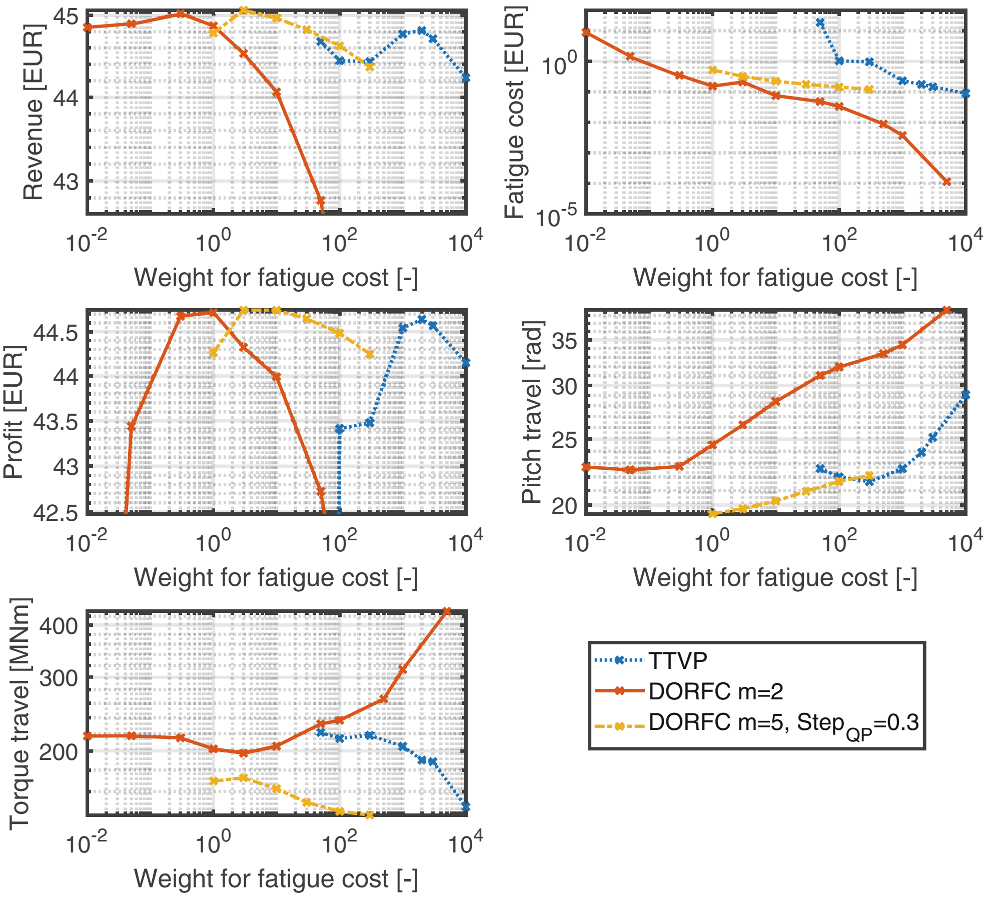

FIGURE 8 Variation of fatigue weights for three different MPC formulations. Each node represents an average over 12 simulations, each with a different turbulent seed

plateaus stem from the clear focus of DORFC on the reduction of stress amplitudes, without unnecessarily spending control effort on the damping of high-frequency oscillations.

In order to verify this statement, the amplitude spectra of the stress at tower root are compared for different controllers in Figure 11. For DORFC, different fatigue exponents m and fatigue weights α_{damage} are presented. Each spectrum is based on an average over 12 Fourier-transformed simulation outputs of 600 s of length each. For this comparison, the base configuration of DORFC-2 is re-tuned to a fatigue weight of $\alpha_{\text{damage}} = 0.8$, in order to achieve identical fatigue cost as TTVP.

For the MPC without fatigue penalization, a clear resonance peak is visible at the first fore-aft eigenfrequency $f_{\text{eigen,T,1}} = 0.277$ Hz of the tower, which is represented by the vertical dash-dotted line in Figure 11. All MPC formulations equipped with TTVP or DORFC are able to eliminate this peak.

However, despite TTVP and the base configuration of DORFC resulting in the same fatigue cost, their spectral behavior is quite different. In fact, TTVP exhibits a deep notch-filter-like decrease of amplitude around the first eigenfrequency of the tower. TTVP focuses on this frequency range, since it contains much stress content at high frequencies and thus at high stress rates, which are explicitly penalized in the cost function. In contrast, DORFC-2 and DORFC-5 exhibit a broader frequency range of moderate amplitude decrease. This behavior is even more pronounced in the additional spectrum plotted in the same figure for a higher fatigue weight $\alpha_{\text{damage}} = 10$ and DORFC-2. This demonstrates that DORFC is focused on the reduction of stress amplitudes, irrespective of the frequency where they occur. This conclusion is in line with the above statement regarding the origin of stress plateaus.

To complete the picture of spectral behavior, it is discussed how the controllable frequency range is limited by the controller settings (see Section 5.3):

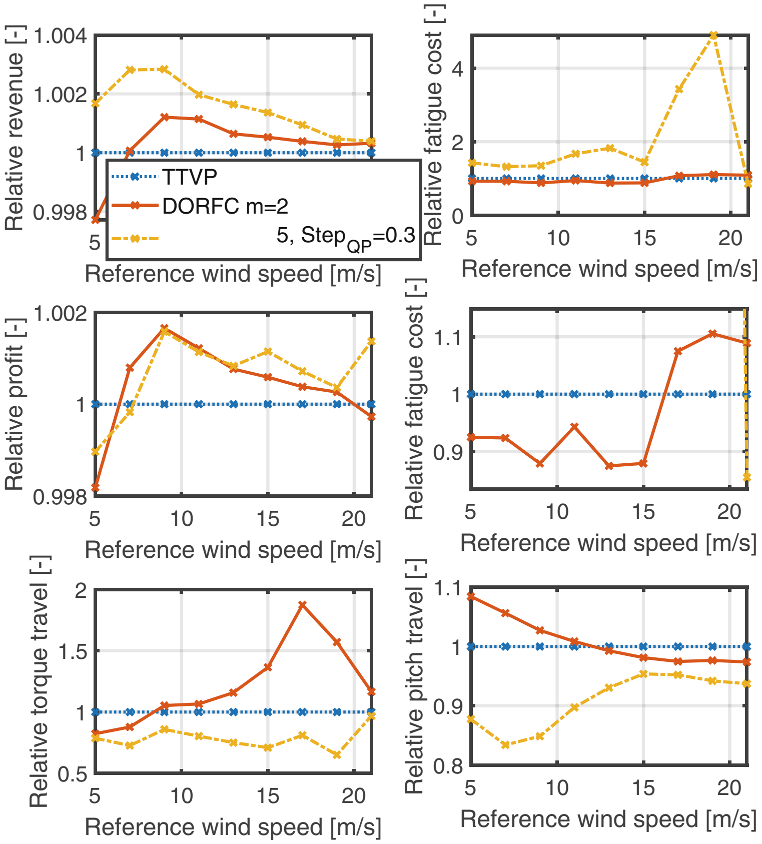

FIGURE 9 Performance as a function of wind speed for three different MPC formulations. All quantities are normalized w.r.t. the TTVP-based MPC. Middle row, right: zoomed version of fatigue cost plot. Each node represents an average over 12 simulations, each with a different turbulent seed DORFC m=2

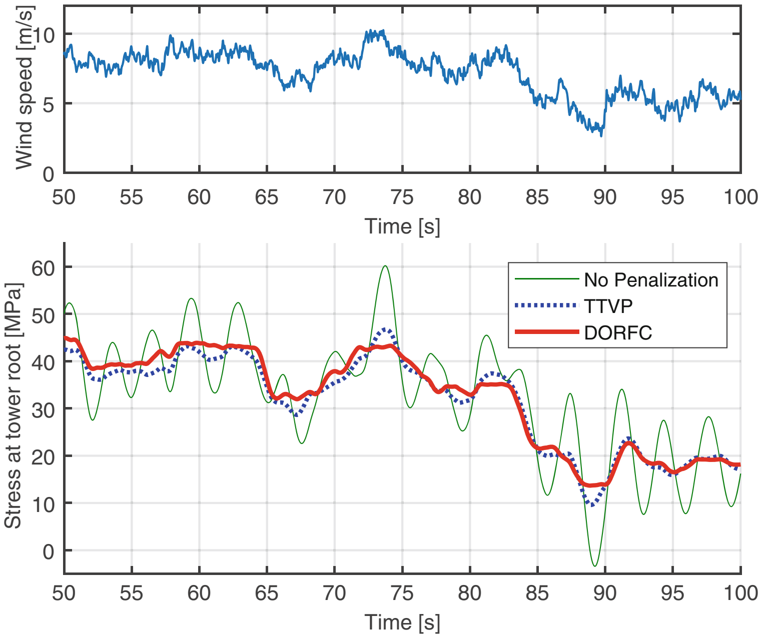

FIGURE 10 Comparison of the dynamic behavior for MPC without the penalization of fatigue, using TTVP and DORFC-2

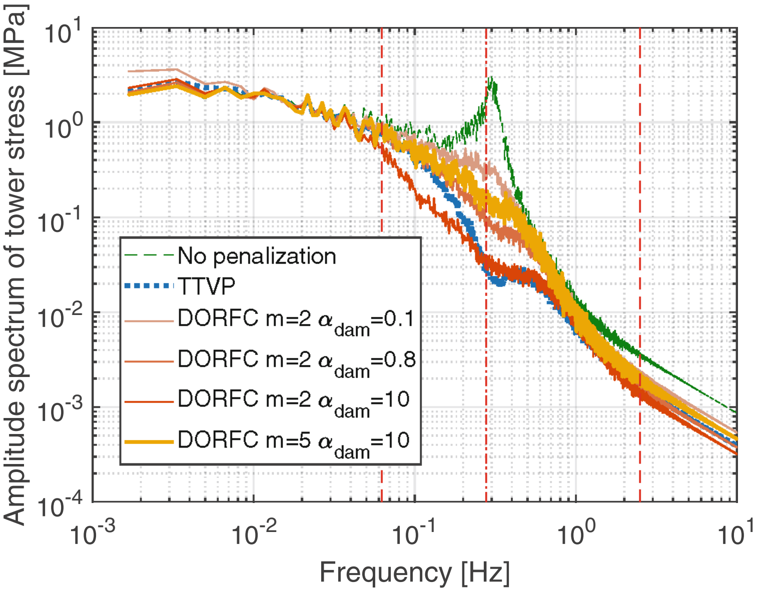

FIGURE 11 Amplitude spectrum of stress at tower root. DORFC-2 ($m = 2$) is re-tuned to $\alpha_{\text{damage}} = 0.8$ to achieve equal fatigue cost as TTVP. For comparison, results for a lower ($\alpha_{\text{damage}} = 0.1$) and a higher ($\alpha_{\text{damage}} = 10$) tuning of DORFC-2s are provided. DORFC-5 ($m = 5$) is only represented by its profit-optimal tuning. Vertical dashed lines: limits of controllable frequency range. Vertical dash-dotted line: 1st eigenfrequency of tower

- Lower frequency limit: The length $T_{\text{horizon}} = 8$ s of the prediction horizon limits the longest controllable *half* cycle to a length of 8 s. Thus, the minimum controllable frequency of a *full* cycle (double length) can be estimated to $f_{\text{control,min}} = 1/(2T_{\text{horizon}}) = 6.25 \cdot 10^{-2}$ Hz. This limit is represented by the left vertical dashed line in Figure 11.
- Upper frequency limit: Considering *Shannon theorem*,[39] the controller sample time of $T_{\text{cntrl}} = 0.2$ s results in a maximum controllable frequency of $f_{\text{control,max}} = 1/(2T_{\text{cntrl}}) = 2.5$ Hz. This limit is represented by the right vertical dashed line in Figure 11.

6.3 | Validation of damage estimation

For a verification of the online damage estimation capability of DORFC, its online fatigue estimate $\bar{J}_{\text{fatigue,estimated}}$ is compared to an a posteriori fatigue estimate $J_{\text{fatigue,real}}$, which is obtained via the original RFC and represents the fatigue accumulated over the entire simulation. Similarly, the online estimate $\bar{J}_{\text{fatigue,estimated}}$ is obtained via an accumulation over the simulation time $[T_{\text{sim,start}}, T_{\text{sim,end}}]$ of 600 s and is computed as

$$\bar{J}_{\text{fatigue,estimated}} = \sum_{t_{\text{cntrl}}=T_{\text{sim,start}}}^{T_{\text{sim,end}}} \frac{1}{N_u} J_{\text{fatigue}}(t_{\text{cntrl}}, t_{\text{end}}). \tag{34}$$

Here, $J_{\text{fatigue}}(t_{\text{cntrl}}, t_{\text{end}})$ is the predicted fatigue cost (21) at a controller execution instance t_{cntrl} within the simulation time and at the end of the respective prediction horizon t_{end}. These individual fatigue costs are normalized by the number N_u of control intervals within the MPC horizon. This normalization is required, since the predicted fatigue cost $J_{\text{fatigue}}(t_{\text{cntrl}}, t_{\text{end}})$ represents fatigue over the entire horizon T_{horizon}, while the accumulation is over simulation time and thus requires only the average fatigue value over one controller execution step $T_{\text{cntrl}} = \frac{T_{\text{horizon}}}{N_u}$. In the following, the ratio $\bar{J}_{\text{fatigue,estimated}}/J_{\text{fatigue,real}}$ of estimated to real fatigue cost is assessed for variations of wind speed and fatigue weight.

6.3.1 | Variation of wind speed

Figure 12 shows the fatigue estimation ratio $\bar{J}_{\text{fatigue,estimated}}/J_{\text{fatigue,real}}$ over more than 200 simulations at different wind speeds. For both formulations DORFC-2 and -5, the estimation ratio increases with wind speed and spans about 2 orders of magnitude. The ideal ratio of 1 is met for DORFC-2 at low reference wind speeds, and for DORFC-5 at high reference wind speeds.

This deserves some attention, as estimation ratios of 1 or more are not to be expected. As shown in Loew et al.,[13] a significant portion of stress cycles has a time range that is longer than the present MPC horizon of 8 s. Therefore, since the MPC only sees a part of the cycle stress amplitude, it should *under*estimate their fatigue impact. Two mechanisms of overestimation may compensate this effect:

- One reason for the high estimation ratio may be the presence of strong oscillations in the predicted stress signal, when an abrupt excitation (e.g., by a fast wind gust) enters the prediction horizon "from the right-hand side" at $t = T_{\text{end}}$. These strong oscillations in reality do not occur at the turbine (plant), since within the subsequent QP steps the MPC generates suitable control signals to attenuate the excitation until it reaches the beginning of the horizon at $t = t_0$. Thus, large cycles that only occur in the prediction lead to an overestimation of fatigue cost.
- Another reason, which only applies to DORFC-2, may be the approximation of fatigue cost by a monomial of exponent $m = 2$, as shown in Figure 7. This function overestimates fatigue cost for short cycles of low equivalent stress, and underestimates fatigue cost for long cycles. Thus, fatigue cost is overestimated for the rather small cycles that are visible in the prediction horizon.

6.3.2 | Variation of fatigue weight

The fatigue weight α_{damage} is influencing only indirectly the computation of fatigue $\bar{J}_{\text{fatigue,estimated}}$ in the MPC. Via the optimization, a higher fatigue weight typically leads to smaller stress cycles, and thus also affects the estimation ratio. Consequently, the question arises if fatigue weight tuning implicitly aims for a certain estimation ratio. Figure 13 shows the fatigue estimation ratio $\bar{J}_{\text{fatigue,estimated}}/J_{\text{fatigue,real}}$ for different fatigue weights from the fatigue cost tuning setting of Section 6.1.

The first observation is that for DORFC-2, the estimation ratio remains constant over various orders of magnitude of fatigue weight, and reaches a local maximum at the optimum weight of $\alpha_{\text{damage}} = 1$. In contrast, DORFC-5 exhibits a higher slope and no extremum at the optimum weight of $\alpha_{\text{damage}} = 10$. The second observation is that the estimation ratio at optimum fatigue weights deviates significantly from the value of 1 for both formulations. From these observations, it may be concluded that optimum tuning does not imply meeting a certain or ideal estimation ratio.

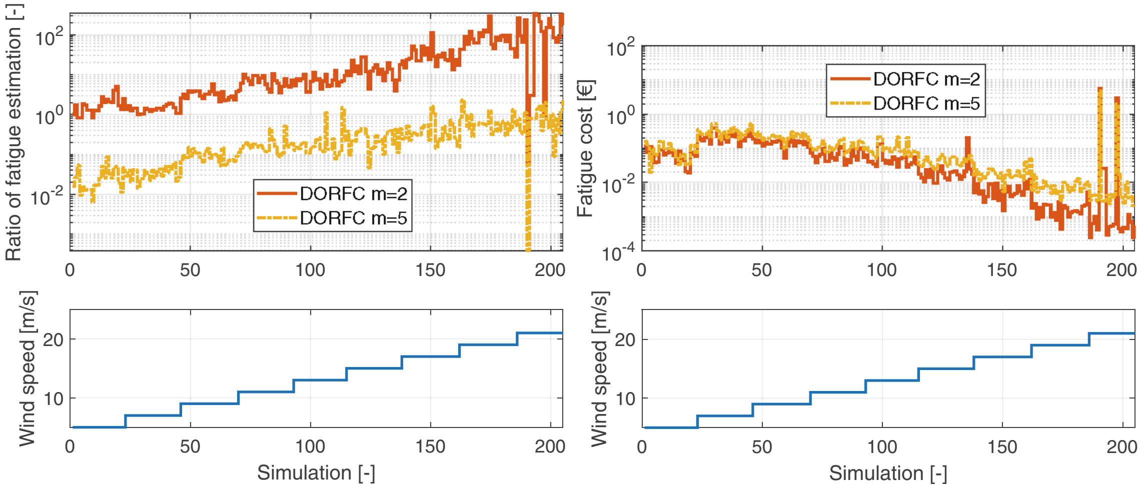

FIGURE 12 Ratio of fatigue cost estimation and fatigue cost over more than 200 individual simulations at different wind speeds

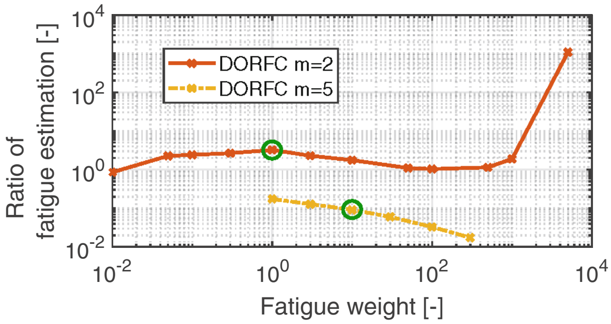

FIGURE 13 Ratio of fatigue cost estimation over fatigue weights at fixed wind speed. The green circles mark the optimum tuning weights. Each node represents an average over 12 simulations

6.4 | Seldom switching and continuous substitution

As shown in Section 4.2, the rainflow algorithm $RFC(\sigma)$ is substituted by the continuous expressions (16) for the calculation of gradients in DORFC. The underlying assumption is that the execution structure of RFC, and thus the output cycle structure, switches only seldom. In the following, the validity of this assumption is assessed *over subsequent* MPC steps and *within individual* MPC steps. The former is required for the applicability of the RTI approach.

For the present assessment, the controller sample time was extended to $T_{\text{cntrl}} = 0.25$ s and the prediction horizon was extended to $T_{\text{horizon}} = 15$ s, in order to accommodate even more stress cycles.

6.4.1 | Assumption of seldom switching cycle structure

The most crucial aspect of the cycle structure is the sample indices of maxima $k_{\max,c}$ and minima $k_{\min,c}$ of detected cycles. Figure 14 shows these indices for the three cycles of highest fatigue cost for every execution of DORFC over an operation period of 10 s. As shown in Figure 4, for each MPC step there are two evaluations of DORFC that occur simultaneously. In the present visualization, every second evaluation result of DORFC is shifted in time by one half controller sample time, which leads to a virtual DORFC sample time of $T_{\text{cntrl}}/2 = 0.125$ s.

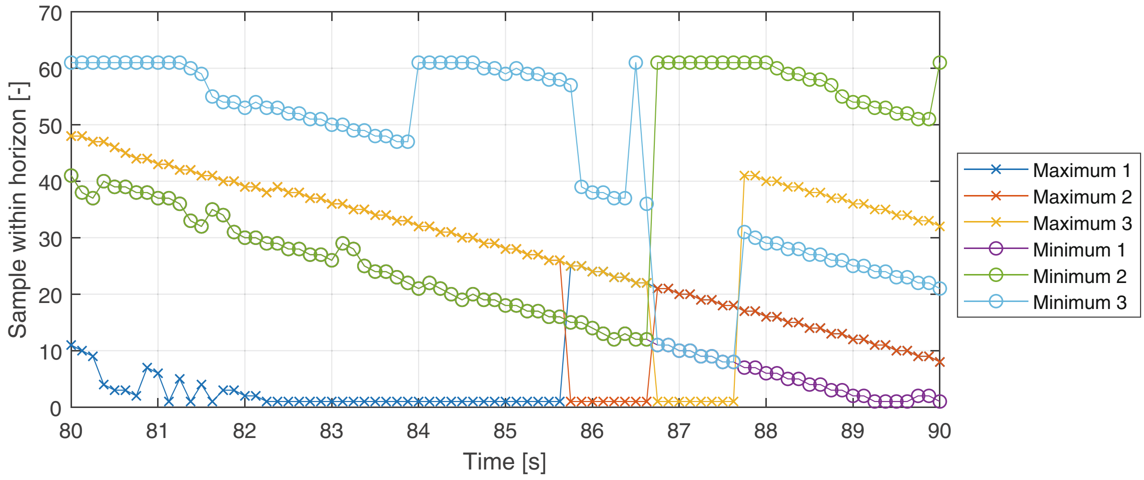

FIGURE 14 Sample positions of maxima $k_{\max,c}$ and minima $k_{\min,c}$ of the three most damaging stress cycles over an operation time of 10 s. The cycles are sorted in a descending order w.r.t. damage cost. Samples are displayed for each execution of DORFC that occurs every $T_{\text{cntrl}}/2 = 0.125$ s. The prediction horizon comprises 61 samples, and thus the vertical axis is limited accordingly. The corresponding stress output in these 10 s of operation time can be seen in Figure 10

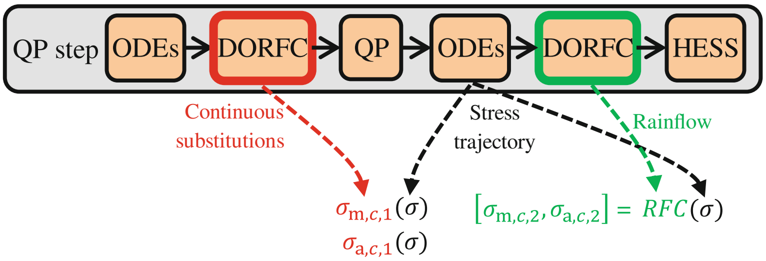

FIGURE 15 Setup for the validation of continuous substitution

Within the initial four seconds, the positions of the detected cycles approximatively follow the evolution of operation time, which is an expected behavior. This corresponds to a shift towards the beginning of the prediction horizon, and thus a decrease of the respective sample indices by one for every MPC step. Correspondingly, the slopes are approximatively equal to −4 samples/s. Another expected result is the null slope at the upper and lower borders of the horizon, where new cycles enter or exit the horizon. For example, at $k = 1$ the exit of the initially most damaging cycle from the prediction horizon can be observed by a switch from Maximum 1 over Maximum 2 and Maximum 3 to disappearance.

Continuity of the cycle identification is interrupted by only a few switching events, especially for the period 85 s < t < 88 s. However, it can be observed that maxima and minima maintain their respective nature, and are only assigned to different cycles. Thus, the directive to the optimization algorithm to decrease or increase the corresponding stress values will not change.

6.4.2 | Substitution by continuous functions

In order to check the validity of the continuous substitution within one MPC step, the outputs of the continuous expressions before the QP step are compared to the outputs of RFC after the QP step, as shown in Figure 15.

In order to focus on the change in cycle structure, the comparison is based on a common stress trajectory from the *second* ODE evaluation within the MPC step. Thus, this stress trajectory is obtained using the same initial states but updated control variables $\overline{u}_{i,j}$ w.r.t. the first evaluation of DORFC. This stress trajectory is then input into the continuous expressions ($\sigma_{m,c,1}, \sigma_{a,c,1}$) created before the QP step, and to the rainflow algorithm executed after the QP step.

Figure 16 shows that the ratios between the outputs of the continuous expressions and RFC are typically very close to the ideal value of 1. All amplitude and stress mean results are deviating by less than 3%, except for one sample at $t = 81.5$ s.

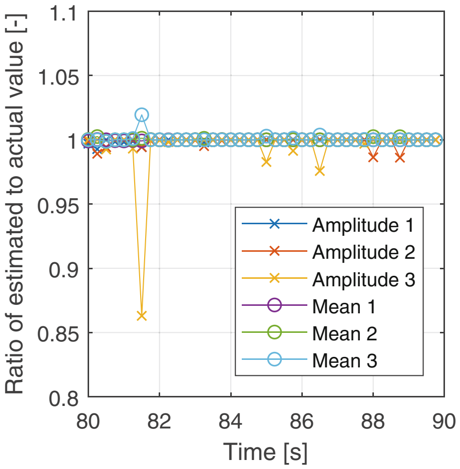

FIGURE 16 Ratio of output of continuous substitutes and output of actual rainflow counting. Ratios are displayed for amplitude and stress mean values of the three most damaging cycles over an operation time of 10 s. The cycles are sorted in a descending order w.r.t. damage cost. Samples are displayed for each MPC step occurring every $T_{\text{cntrl}} = 0.25$ s

At this instance, stress amplitudes are slightly underestimated, whereas stress mean values are slightly overestimated by the continuous expressions.

In conclusion, the use of continuous substitutes of the rainflow algorithm is considered as justified and amply acceptable, at least as far as the present application is concerned.

6.5 | Fatigue cost gradients

The results discussed so far show good performance of the MPC equipped with DORFC. However, switches in the fatigue cost gradient program can lead to excessive changes of the optimization problem for successive MPC steps. These changes can lead to convergence problems for the optimization algorithm, since RTI does not treat an approximate SQP in this case. Therefore, in the following, the continuity of the fatigue cost gradient is assessed, and the computation of the fatigue cost gradient is validated.

6.5.1 | Assessment of the fatigue cost gradient dynamics

Figure 17 shows time-series of the fatigue cost gradient $\mathrm{d}J_{\text{fatigue}}/\mathrm{d}\bar{u}_{1,1}$ w.r.t. to pitch angle ($i = 1$) in the first control interval of the prediction horizon ($j = 1$). Due to its position at the beginning of the prediction horizon, there is a high influence of this pitch control variable on the response of the system behavior, and thus on fatigue cost.

For TTVP, almost sinusoidal oscillations of the fatigue cost gradient can be observed, at a frequency similar to the first eigenfrequency $f_{\text{eigen,T,1}}$ of the tower.

For DORFC, significant variations in the fatigue cost gradient can be observed at higher frequencies. High-frequency content of the fatigue cost gradient signal could in principle stem from the discontinuity of the RFC algorithm. However, for the assessed time interval, the detected switchings in cycle structure visible in Figure 14 do not seem to correlate with the variations in the fatigue cost gradient signal.

Another reason for the presence of high frequencies in the fatigue cost gradient can be its inherent static nonlinearities (24) and (25). The output signal of a highly nonlinear function can contain a wider-band response than the input,[40] and especially it may contain higher harmonics.[41]

Part of future research will focus on the reduction of this high frequency content.

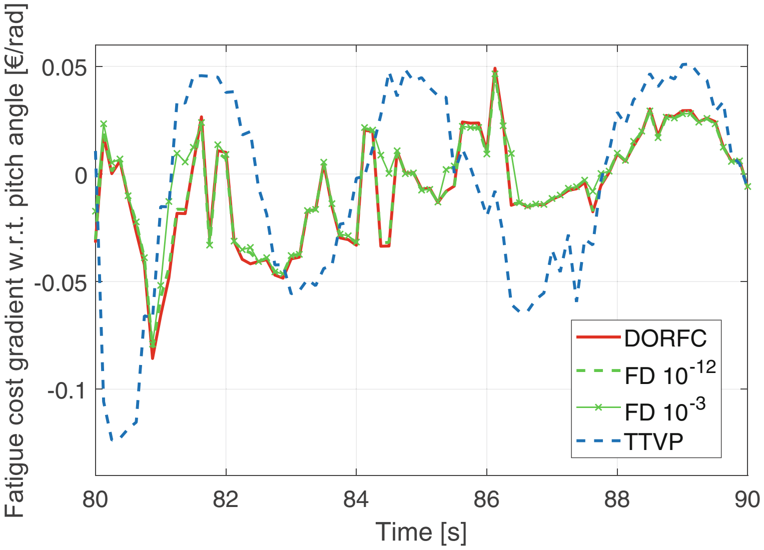

FIGURE 17 Gradients $\mathrm{d}J_{\text{fatigue}}/\mathrm{d}\bar{u}_{1,1}$ of fatigue cost w.r.t. pitch angle in the first control interval of the prediction horizon. Comparison of gradients obtained by automatic differentiation within DORFC, by finite differences (FD) using a step size of 10^{-12} or 10^{-3} within DORFC, and from TTVP. The corresponding stress output in these 10 s of operation time are reported in Figure 10

6.5.2 | Validation of the fatigue cost gradient computation

For validating the AD procedure within DORFC, gradients of fatigue cost are calculated by FD, as shown in Figure 17. AD and FD are in good agreement if a fixed FD step-size of $\delta\overline{u}_{1,1} = 10^{-12}$ rad is chosen. This step-size is sufficiently small, because the control variable of pitch angle usually varies in the order of $\delta\overline{u}_{1,1} = [10^{-2}, 10^{-1}]$ rad.

If a larger FD step-size of $\delta\overline{u}_{1,1} = 10^{-3}$ rad is chosen, some information is lost in comparison to the locally exact AD output of DORFC. This can be seen for example, at $t = 81$ s and $t = 84.5$ s.

7 | CONCLUSION AND OUTLOOK

The novel fatigue cost formulation Direct Online Rainflow Counting represents a fundamental advance in the control of fatigue. In the present work, for the first time, a comprehensive explanation and assessment of this formulation have been provided.

Three obstacles have been identified in the literature, that seemed to prevent the direct implementation of fatigue estimation in MPC:

- Non-standard cost function;
- Non-closed form of the algorithm;
- High nonlinearity due to discontinuities.

To overcome these obstacles, the new DORFC algorithm has been proposed, which works by

- Separating the solution of system dynamics and optimization problem, in order to allow for the “surgical” inclusion of specialized code for the calculation of gradients;
- Substituting the discontinuous algorithm by linear functions at each MPC step;
- Switching the discontinuities only seldom, by choosing a sufficiently low controller sampling time, and by ensuring a sufficiently long “average dwell-time” between switching events.

The present work has also discussed how to embed DORFC within an MPC framework, how to define appropriate cost models, and how to compute the fatigue cost gradients.

Application to a wind turbine control problem has shown that the novel DORFC MPC can achieve slightly higher profit at significantly lower actuator usage in comparison to a conventional MPC. Additionally, finding an initial guess for the tuning weight is much easier for DORFC.

A comparison of the dynamic response has shown that DORFC is able to reduce stress amplitudes by introducing plateaus in the stress trajectory. Frequency spectra have shown that DORFC beneficially focuses on the reduction of stress amplitudes, irrespective of the frequency where they occur.

An assessment of online damage estimation capabilities in DORFC MPC has revealed that estimation ratios are at meaningful levels, and that optimal cost tuning does not seem to imply meeting a certain estimation ratio.

An investigation into the discontinuous behavior of the cost function has shown that switching events happen sufficiently seldom. Furthermore, high-frequency behavior in the fatigue cost gradient do not seem to correlate to switching. Finally, the correct implementation of the fatigue cost gradient has been verified by the FD method.

Further investigations on the method of DORFC are planned, and in particular the focus will be directed to:

- Investigating the effects of uncertainties and robustness especially by considering the presence of noise in the measurements and of a plant-model mismatch;
- The simultaneous consideration of fatigue cost for multiple components of the turbine;
- The application to other devices, as for example batteries and power semiconductors. As a first step, a derived formulation from DORFC has been applied successfully to a complex hybrid energy system comprising a wind turbine and a battery energy storage system.[42]

ACKNOWLEDGMENTS

The authors would like to thank the German Federal Ministry for Economic Affairs and Energy for funding the *MISTRALWIND* project. Furthermore, the authors would like to thank Elisabeth Gruber for her careful review of the manuscript.

NOMENCLATURE

Quantity	**Explanation**
$a_{\square}$	Coefficient of fatigue cost function
$\boldsymbol{d}$	Vector of external disturbances
d_{T} (m)	Deflection of turbine tower
f_{control} (Hz)	Sample-frequency of controller
$f_{\text{eigen, T, 1}}$ (Hz)	First fore-aft eigenfrequency of turbine tower
$\boldsymbol{g}^{*}$	Gradient vector at operating point
k	Control samples within horizon
$k_{\text{end}\,,c}$	Sample index of cycle end
$k_{\text{max}\,,c}$	Sample index of cycle maximum
$k_{\text{min}\,,c}$	Sample index of cycle minimum
$k_{\text{start}\,,c}$	Sample index of cycle start
m_{T} (kg)	Lumped mass of turbine tower
t_{cntrl} (s)	Time-variable of controller sampling
$\boldsymbol{u}$	Vector of control variables
$\overline{\boldsymbol{u}}$	Vector of piece-wise constant optimization variables
w_{c}	Weight of one cycle
$\boldsymbol{x}$	State vector
D_{fatigue}	Total fatigue damage
$D_{\text{fatigue}\,,c}$	Fatigue damage of one cycle
$\boldsymbol{F}$	System of differential equations
$\boldsymbol{H}^{*}$	Hessian matrix at operating point
J	Cost function
N_{c}	Number of stress cycles identified by the rainflow algorithm
N_{fail}	Number of cycles to failure
N_{in}	Number of control inputs
N_{u}	Number of control variables within horizon
N_{x}	Number of states
R_{m} (Pa)	Ultimate tensile stress
T_{cntrl} (s)	Controller sample-time
T_{g} (Nm)	Torque of turbine generator
T_{horizon} (s)	Length of prediction horizon
T_{integr} (s)	Integrator sample-time
V_{w} (m/s)	Wind speed
α_{damage}	Weighing factor for cost function
β_{b} (rad)	Collective pitch angle of blades
σ(Pa)	Stress signal
$\sigma_{\mathrm{a},c}$ (Pa)	Stress amplitude of one cycle
$\sigma_{\mathrm{eq},c}$ (Pa)	Equivalent stress of one cycle
$\sigma_{\mathrm{m},c}$ (Pa)	Stress mean of one cycle
$\sigma_{\mathrm{r},c}$ (Pa)	Stress range of one cycle
ω_{r} (rad/s)	Rotational speed of turbine rotor
$\hat{a}$	Quantity fixed for one MPC-step
$\overline{a}$	Quantity sampled on the control intervals of the prediction horizon

Abbreviation	Explanation
AD	Automatic differentiation
DORFC	Direct Online Rainflow Counting
ENMPC	Economic nonlinear MPC
FD	Finite differences
ICC	Initial capital cost
LiDAR	Light detection and ranging
MPC	Model predictive control
NMPC	Nonlinear model predictive control
ODE	Ordinary differential equation
QP	Quadratic programming
RFC	Rainflow counting algorithm
RTI	Real-time iteration
SQP	Sequential quadratic programming
TTVP	Tower-tip-velocity-penalization
VDE	Variational differential equation

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Stefan Loew https://orcid.org/0000-0002-3342-6548

REFERENCES

1. Suresh S. *Fatigue of Materials*. 2nd ed. Cambridge University Press; 1998.
2. Barradas-Berglind JJ, Wisniewski R. Representation of fatigue for wind turbine control. *Wind Energy*. 2016;19(12):2189-2203. doi:10.1002/we.1975
3. Sanchez H, Escobet T, Puig V, Odgaard PF. Health-aware model predictive control of wind turbines using fatigue prognosis. *IFAC-PapersOnLine*. 2015;48(21):1363-1368. doi:10.1016/j.ifacol.2015.09.715
4. Sorenson GR, Clemett HR. Low-cycle fatigue in small turbines. *Exper Mech*. 1962;2(12):353-358. doi:10.1007/BF02325591
5. GopiReddy L , Tolbert LM, Ozpineci B , Pinto JOP. Rainflow algorithm based lifetime estimation of power semiconductors in utility applications. Proceedings of the 2014 IEEE Applied Power Electronics Conference and Exposition - APEC 2014; 2014:2293-2299.
6. Chen Z, Gao F, Yang C, Peng T, Zhou L. Converter lifetime modeling based on online rainflow counting algorithm. Proceedings of the 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE); 2019:1743-1748.
7. Barré A, Deguilhem B, Grolleau S, Gérard M, Suard F, Riu D. A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *J Power Sources*. 2013;241:680-689. doi:10.1016/j.jpowsour.2013.05.040
8. Shi Y, Xu B, Tan Y, Zhang B. A convex cycle-based degradation model for battery energy storage planning and operation. Proceedings of the 2018 Annual American Control Conference (ACC); 2018:4590-4596.
9. Findeisen R & Allgöwer F An introduction to nonlinear model predictive control. Proceedings of the 21st Benelux Meeting on Systems and Control; 2002:1-23.
10. Henson MA. Nonlinear model predictive control: current status and future directions. *Comput Chem Eng*. 1998;23(2):187-202. doi:10.1016/S0098-1354(98)00260-9
11. Gros S, Schild A. Real-time economic nonlinear model predictive control for wind turbine control. *Int J Control*. 2017;90(12):2799-2812. doi:10.1080/00207179.2016.1266514
12. Evans MA, Cannon M, Kouvaritakis B. Robust MPC tower damping for variable speed wind turbines. *IEEE Trans Control Syst Technol*. 2015;23(1):290-296. doi:10.1109/TCST.2014.2310513
13. Loew S, Obradovic D, Bottasso CL. Model predictive control of wind turbine fatigue via online rainflow-counting on stress history and prediction. *J Phys Conf Ser*. 2020;1618:22041. doi:10.1088/1742-6596/1618/2/022041
14. ASTM. Standard practices for cycle counting in fatigue analysis; 1985.
15. de Jesus Barradas-Berglind J, Wisniewski R, Soltani M. Fatigue damage estimation and data-based control for wind turbines. *IET Control Theory Appl*. 2015;9(7):1042-1050. doi:10.1049/iet-cta.2014.0730
16. Collet D, Alamir M, Di Domenico D, Sabiron G. Non quadratic smooth model of fatigue for optimal fatigue-oriented individual pitch control. *J Phys Conf Ser*. 2020;1618:022004. doi:10.1088/1742-6596/1618/2/022004
17. Luna J, Falkenberg O, Gros S, Schild A. Wind turbine fatigue reduction based on economic-tracking NMPC with direct ANN fatigue estimation. *Renew Energy*. 2020;147:1632-1641. doi:10.1016/j.renene.2019.09.092

18. Loew S, Obradovic D, Bottasso CL. Direct online rainflow-counting and indirect fatigue penalization methods for model predictive control. Proceedings of the 2019 18th European Control Conference (ECC); 2019:3371-3376; IEEE.
19. British Standards Institute. Steel, concrete and composite bridges. Code of practice for fatigue; 1980.
20. Betriebsfestigkeit EH. *Verfahren und Daten zur Bauteilberechnung*. Vol 3. korrigierte und erg. ed. VDI-Buch, Springer; 2006.
21. Miner MA. Cumulative damage in fatigue. *J Appl Mech*. 1945;12(3):159-164.
22. Rawlings JB, Mayne DQ, Diehl MM. *Model Predictive Control: Theory, Computation, and Design*. 2nd ed. Nob Hill Publishing; 2017.
23. Altmannshofer S. Fast suboptimal nonlinear model predictive control of an inverted pendulum. *IFAC Proc Vol*. 2012;45(17):442-447. doi:10.3182/20120823-5-NL-3013.00051
24. Löw S, Obradovic D. Real-time implementation of nonlinear model predictive control for mechatronic systems using a hybrid model. Proceedings of the 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE); 2018:164-167. doi:10.17560/atp.v60i09.2359
25. Gros S, Zanon M, Quirynen R, Bemporad A, Diehl M. From linear to nonlinear MPC: bridging the gap via the real-time iteration. *Int J Control*. 2016;93(1):62-80. doi:10.1080/00207179.2016.1222553
26. Diehl M, Bock H, Schlöder J. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM J Control Optim*. 2005;43:1714-1736. doi:10.1137/S0363012902400713
27. Papageorgiou M, Leibold M, Buss M. *Optimierung: Statische, dynamische, stochastische Verfahren für die Anwendung*. 4th ed. Springer Vieweg; 2015.
28. Grüne L, Pannek J. *Nonlinear Model Predictive Control: Theory and Algorithms*. Communications and Control Engineering. 2nd ed. Springer International Publishing; 2017.
29. Hammerum K. *A Fatigue Approach to Wind Turbine Control*. Master's thesis. Technical University of DenmarkDK-2800 Kgs, Lyngby, Denmark; 2006.
30. Barradas-Berglind JJ, Jayawardhana B, Wisniewski R. Wind turbine control with active damage reduction through energy dissipation. Proceedings of the 2016 American Control Conference (ACC); 2016.
31. Xu B, Zhao J, Zheng T, Litvinov E, Kirschen DS. Factoring the cycle aging cost of batteries participating in electricity markets. *IEEE Trans Power Syst*. 2018;33(2):2248-2259. doi:10.1109/TPWRS.2017.2733339
32. Müller MA, Allgöwer F. Improving performance in model predictive control: switching cost functionals under average dwell-time. *Automatica*. 2012;48(2):402-409. doi:10.1016/j.automatica.2011.11.005
33. Al Seyab RK. *Nonlinear Model Predictive Control Using Automatic Differentiation*. Ph.D. dissertation. Bedfordshire, UK: Cranfield University; 2007.
34. Schlipf D, Schlipf DJ, Kühn M. Nonlinear model predictive control of wind turbines using LIDAR. *Wind Energy*. 2013;16(7):1107-1129. doi:10.1002/we.1533
35. The MathWorks Inc. Quadratic programming algorithms; 2019.
36. Schlipf D, Fürst H, Raach S, Haizmann F. Systems engineering for lidar-assisted control: a sequential approach. *J Phys Conf Ser*. 2018;1102:012014. doi:10.1088/1742-6596/1102/1/012014
37. Simley E, Pao LY, Frehlich R, Jonkman B, Kelley N. Analysis of light detection and ranging wind speed measurements for wind turbine control. *Wind Energy*. 2014;17(3):413-433. doi:10.1002/we.1584
38. International Electrotechnical Commission. IEC 61400-1 Ed.3: wind turbines - Part 1: design requirements; 2005.
39. Lunze J. *Regelungstechnik*. 8th ed. Springer-Lehrbuch, Springer Vieweg; 2014.
40. Wise G, Traganitis A, Thomas J. The effect of a memoryless nonlinearity on the spectrum of a random process. *IEEE Trans Inf Theory*. 1977;23(1):84-89. doi:10.1109/TIT.1977.1055658
41. Chatterjee A. *A Brief Introduction to Nonlinear Vibrations: Lecture Notes*. Bangalore: Indian Institute of Science; 2009.
42. Anand A, Loew S, Bottasso CL. Economic control of hybrid energy systems composed of wind turbine and battery. Proceedings of the 2021 European Control Conference (ECC); 2021;2565-2572; IEEE.
43. The MathWorks Inc. Rainflow counts for fatigue analysis; 2018.
44. Bottasso CL, Croce A, Savini B, Sirchi W, Trainelli L. Aero-servo-elastic modeling and control of wind turbines using finite-element multibody procedures. *Multibody Syst Dyn*. 2006;16(3):291-308. doi:10.1007/s11044-006-9027-1

How to cite this article: Loew S, Obradovic D, Bottasso CL. Economic nonlinear model predictive control of fatigue—Formulation and application to wind turbine control. *Optim Control Appl Meth*. 2023;44(2):647-676. doi:10.1002/oca.2870

APPENDIX A. RAINFLOW ALGORITHM

A.1 Algorithmic flow

A flowchart of RFC is displayed in Figure A1. At the beginning of the algorithm, RFC receives as input a stress trajectory and extracts its reversals (extrema). Throughout the algorithm, reversals are read consecutively from left to right. Each new reversal is stored in an operational memory. From this memory, cycles are identified based on a triplet of reversals. The rainflow algorithm contains four main loops. *Loop 1* initiates the reading of a new reversal sample, if less than three reversals are in the operational memory. *Loop 2* initiates the reading of a new reversal if, based on the current operational memory, no cycle could be identified. *Loop 3* and *Loop 4* initiate the subsequent check for a cycle in the current operational

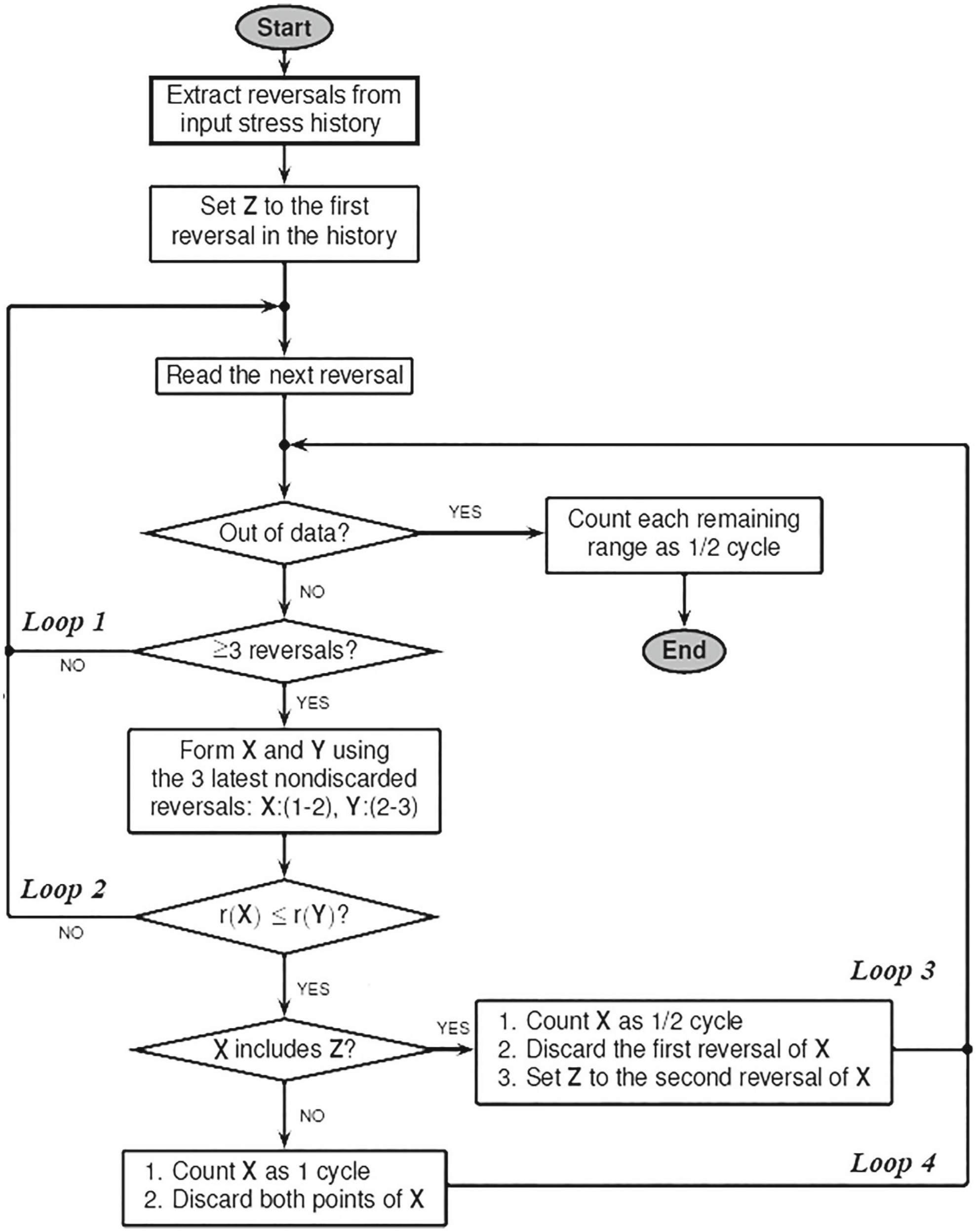
Start
Extract reversals from input stress history
Set Z to the first reversal in the history
Read the next reversal
Out of data?
YES
Count each remaining range as 1/2 cycle
End
NO
Loop 1
NO
≥3 reversals?
YES
Form X and Y using the 3 latest nondiscarded reversals: X:(1-2), Y:(2-3)
Loop 2
NO
r(X) ≤ r(Y)?
YES
Loop 3
X includes Z?
YES
1. Count X as 1/2 cycle
2. Discard the first reversal of X
3. Set Z to the second reversal of X
NO
1. Count X as 1 cycle
2. Discard both points of X
Loop 4

FIGURE A1 Flowchart of the MATLAB-implementation `rainflow()` of the *Three-Point Algorithm* (simplified from the Mathworks[43]). Here, stress extrema are called "reversals". The range $r(\boldsymbol{X}) = |X(2) - X(1)|$ of a stress value pair $\boldsymbol{X}$ is the absolute value of the difference between both stresses

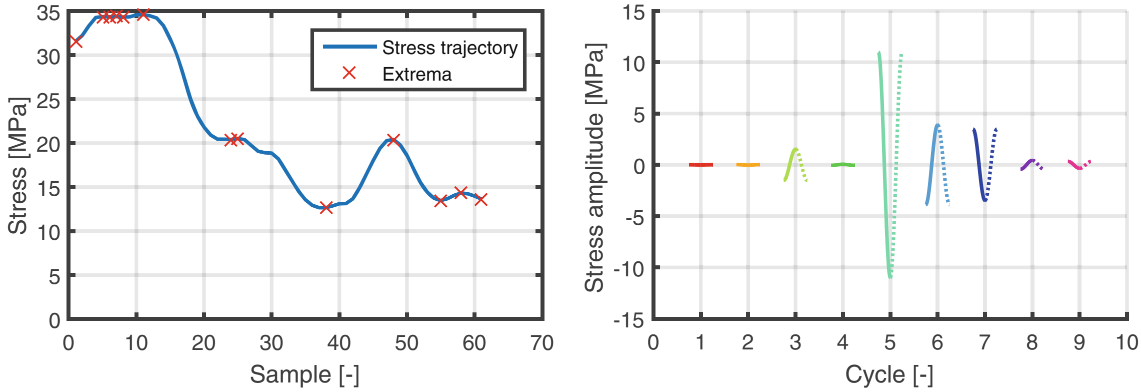

FIGURE A2 Left: Stress trajectory and identified extrema. Right: identified stress cycles. The cosines are flipped on the horizontal axis according to their appearance in the stress trajectory. All cycles are displayed with uniform period time. For the identified half cycles, the second half of the cosine is dotted, whereas full cycles are shown entirely in solid lines

TABLE A1 Outputs of the Three-Point Algorithm for the exemplary stress trajectory of Figure A2

c	$\sigma_{r,c}$ (MPa)	$\sigma_{m,c}$ (MPa)	$k_{start,c}$	$k_{end,c}$	w_c
1	0.02	34.37	5	6	1
2	0.06	34.36	7	8	1
3	3.05	33.09	1	11	0.5
4	0.11	20.47	24	25	1
5	21.96	23.63	11	38	0.5
6	7.76	16.53	38	48	0.5
7	6.95	16.94	48	55	0.5
8	0.86	13.90	55	58	0.5
9	0.68	13.99	58	61	0.5

memory and are triggered after identification of a half or full cycle, respectively. A more comprehensive explanation of the algorithm can be found in the MathWorks.[43]

A.2 Example

The left part of Figure A2 shows an exemplary stress trajectory and its extrema. Using this series of extrema, the Three-Point Algorithm identifies three minor full cycles and six half cycles as shown in the right part of Figure A2 and in Table A1. The half cycle $c = 5$ has the most impact on fatigue and contains the nested full cycle $c = 4$.

APPENDIX B. WIND TURBINE MODEL

B.1 System quantities and optimization variables

The states of the wind turbine are collected in the state vector

$$\boldsymbol{x}(t) = \left(\omega_r(t), d_T(t), \dot{d}_T(t), \beta_b(t), \dot{\beta}_b(t), T_g(t)\right)^T, \tag{B1}$$

which is composed of the rotational speed of the rotor $\omega_r(t)$, the fore-aft deflection of the tower tip $d_T(t)$ and its time derivative, the collective pitch angle of the blades $\beta_b(t)$ and its time derivative, and the generator torque $T_g(t)$.

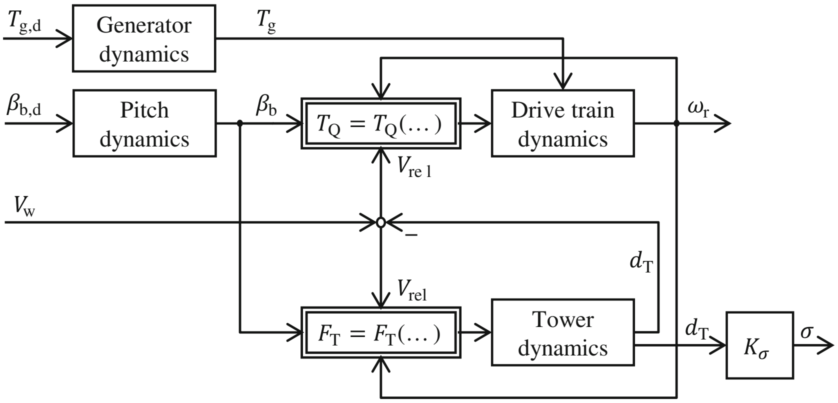

FIGURE B1 Coupled dynamics of wind turbine drive train and tower

Stress at tower root is not a state of this simplified model. However, it is assumed that the stress at tower root is linearly dependent on the tower tip deflection $d_T(t)$, resulting in $\sigma(t) = K_\sigma d_T(t)$. The control variables for the subordinate controllers of the corresponding actuators are $\boldsymbol{u}(t) = \left(\beta_{b,d}(t), T_{g,d}(t)\right)^T$ where $\beta_{b,d}(t)$ is the desired blade pitch angle and $T_{g,d}(t)$ is the desired generator torque. The control variables are defined as piecewise-constant time functions $\overline{u}_{1,j} = \overline{\beta}_{b,d,j}$ and $\overline{u}_{2,j} = \overline{T}_{g,d,j}$. This leads to the MPC optimization variable vector $\overline{\boldsymbol{u}} = \left(\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2\right)^T$. Excitement and disturbance $d(t) = V_w(t)$ of the system stem from the ambient wind speed $V_w(t)$, which is assumed to be perfectly measured by a forward-looking LiDAR.

B.2 System dynamics

The generator and pitch actuator dynamics are modeled as linear first order (PT_1) and second order systems (PT_2), respectively. Nonlinearity in the system model stems from the aerodynamic torque

$$T_Q(t) = T_Q(\omega_r(t), V_{rel}(t), \beta_b(t)), \tag{B2}$$

and the aerodynamic thrust force

$$F_T(t) = F_T(\omega_r(t), V_{rel}(t), \beta_b(t)). \tag{B3}$$

The aerodynamic loads T_Q and F_T are modeled via feed-forward neural networks, trained on samples at stationary operating points calculated by the aeroelastic simulator *Cp-Lambda*,[44] as presented in Löw et al.[24] The aerodynamic loads depend on the rotor-relative wind speed

$$V_{rel}(t) = V_w(t) - \dot{d}_T(t), \tag{B4}$$

where the external rotor-effective ambient wind speed $V_w(t)$ is combined with the tower tip speed $\dot{d}_T(t)$. In turn, torque and thrust affect the drive train dynamics

$$\dot{\omega}_r(t) = \frac{1}{J_{tot}}\left(T_Q(t) - T_g(t)\right), \tag{B5}$$

and the tower fore-aft dynamics

$$\ddot{d}_T(t) = \frac{1}{m_T}\left(F_T(t) - c_T\dot{d}_T(t) - k_T d_T(t)\right), \tag{B6}$$

as shown in Figure B1. The model parameters include the lumped inertia J_{tot} of rotor and drive train, and the lumped mass m_T, damping c_T and stiffness k_T of the tower.