Technische Universität München

TUM School of Engineering and Design

Model Predictive Control of Fatigue

Stefan Herbert Löw

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz:	Prof. Dr.-Ing. Christian Große
Prüfer der Dissertation:	1. Prof. Dr. Carlo L. Bottasso
	2. Prof. Dr. Kathryn Johnson

Die Dissertation wurde am 04.11.2021 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 17.02.2022 angenommen.

Technische Universität München
TUM School of Engineering and Design
Lehrstuhl für Windenergie
Boltzmannstraße 15
D-85748 Garching bei München
Germany
Tel.: +49 (0) 89 / 289 – 16681
Fax.: +49 (0) 89 / 289 – 16611
Email: info@wind.tum.de
Web: www.wind.mw.tum.de

Abstract

The vision of this dissertation is that devices are aware of their fatigue behavior, and utilize this information to optimally balance revenue and fatigue. Based on this vision, the online estimation of fatigue is improved and novel formulations for Model Predictive Control (MPC) of fatigue are developed. Additionally, the effect of fatigue reduction is demonstrated in the integrated wind turbine design scenario.

Online estimation of fatigue provides insight, which points in time and which actions have a decisive impact on the total fatigue level. Previous algorithms for fatigue estimation provide only value-discontinuous updates of fatigue at discrete time instances or at discrete events. This gap is closed in the present dissertation, where a novel one-step value-continuous algorithm for fatigue estimation is developed, implemented and successfully tested.

Model Predictive Control of fatigue typically involves the minimization of fatigue via a cost function. Conventional fatigue cost formulations from the literature either are not accurate enough or result in high computational load. None of these fatigue cost formulations exhibits a direct implementation of the Rainflow counting algorithm, which, however, is the de facto standard for fatigue estimation. Furthermore, since fatigue is a long-term effect, the historic stress has to be considered, while typical MPCs optimize their control actions only based on the predicted states.

These challenges are overcome in the present work, where four novel fatigue cost formulations are developed, implemented and successfully tested. The corresponding MPCs have a more realistic perception of the current fatigue situation, and have an increased capability of controlling fatigue. From the algorithmic point of view, the added computational load by the fatigue cost is minor compared to the MPC execution, which facilitates the application in various domains. Differences among these four novel fatigue cost formulations are related to simpler implementation into standard MPC frameworks or higher generality. Simpler implementation is enabled by the externalization of the fatigue estimation from the MPC. Higher generality is enabled by casting the fatigue estimation into a hybrid dynamical system.

A glimpse of the broad applicability of the novel MPC formulations is provided by successfully controlling a high-fidelity wind turbine model, a complex battery model, and a challenging hybrid energy system comprising both device models. Particularly for the wind turbine, a comprehensive controller toolchain is set up, comprising a Lidar (Light detection and ranging) simulator, Lidar data processing, and a Moving Horizon Estimator. In a wind turbine simulation setup with turbulent inflow, the novel MPC shows superior performance over a conventional MPC particularly for lower quality of wind prediction and for shorter prediction horizons. In a battery simulation setup targeting a realistic primary frequency control market, the novel MPC outperforms state-of-the-art controllers by smartly balancing power mismatch penalty and battery cyclic aging. In a hybrid energy system simulation, the novel MPC simultaneously controls wind turbine tower fatigue and battery cyclic aging, while ensuring a defined power supply to the grid.

ACKNOWLEDGMENTS

I would like to thank my academic supervisor Prof. Carlo L. Bottasso for his trust that I could be a suitable PhD candidate. I particularly will remember your remarkably close and encouraging supervision, as well as your solid compass for scientific ethics. Especially during the pandemic, you were also concerned about our private situation and family well-being. Over the years, life at the chair benefited a lot from your initiatives for social events as beergarden and Oktoberfest visits, Panettone tastings, and adventurous sledging competitions.

I want thank Prof. Kathryn Johnson for co-supervising this dissertation and for the helpful feedback regarding my work. Further thanks go to Prof. Christian U. Große for our fruitful cooperation in the Mistralwind project, and for taking the chair position of the dissertation committee.

My Siemens supervisor Dragan has generously devoted much of his time to guide my research and to share his rich set of experience. Particularly, I will remember our open brainstorming sessions, your optimal control view on life, your daily happiness at work, your enthusiasm, and your well-sorted music recommendations. "Hvala vam"!

My Siemens colleague Andrei has initiated and supported our research work on battery storage, and has been a very good "babo" of the Energy Systems Automation crew. "Mulțumesc"!

Our Siemens research group head Georg has constantly endeavored to provide an excellent research environment for our PhD projects. Being an expert for sales and sails, you have always safely steered our crew through the stormy company waters. Thank you very much!

Special thanks go to our Siemens team and management assistants Evelyn and Sabrina for their constant help in various aspects, and for always having an open ear.

I would like to thank my Siemens colleague Martin for supporting my research intermezzo in wind farm control, and for integrating me into the SysFlex project team.

It has been a big luck to meet my Siemens and TUM colleague Abhinav, and to have the opportunity to work with him. One of the best aspects is, that we do not only share scientific interests, but also many personal values.

I would like to thank my colleague Helena for the fun co-working, and for communicating many chair aspects which I could have missed due to my "second life" at Siemens. I will particularly remember your research vision and Eurovision.

At the TUM Wind Energy Institute, I was very lucky to share time with Alberto, Arturo, Bastian, Bruno, Carlo S., Chengyu, Doruk, Elli, Filippo, Franz, Irene, Jesse, Johannes, Manos, Marta, Pietro, Robert, Robin, and Sivateja. Special thanks go to the ones who patiently helped me at the very beginning of my PhD project, to my office mates, to the Thursday-sports-and-Friday-pizza team, to the Irish roadtrip crew, and particularly to Captain Cork. During the years, we created countless good memories!

In our Siemens research group, I was very lucky to have so many experienced, motivated and sociable colleagues. Especially our retreat and lunch breaks have been very enjoyable. It has been a particular pleasure to share the experiences and joys of the PhD time with Amer, Armin, Bernd, Florian, Philipp, Sarah, Simon, Sissi, and Vincent. I hope for more common experiences in the future!

I want to thank all project partners of the Mistralwind, SysFlex and Flamingo projects for the nice collaboration and the enriching meetings.

I want to acknowledge the organizations WindForS, European Academy of Wind Energy, Faculty Graduate Center Mechanical Engineering at TUM, sowento GmbH (particularly Martin Koch, Steffen Raach, and Prof. David Schlipf), as well as the PhD Network and the University Relations department of Siemens (especially Fabian Rhein).

From my Bachelor studies at the Regensburg University of Applied Sciences and my Master studies at the FAU Erlangen-Nuremberg, I particularly would like to thank Prof. Borchsenius for sparking my interest in software development, Johannes Meyer for teaching methods of scientific work, as well as Prof. Schneider, Prof. Roppenecker, Prof. Deutscher, Johannes Popp, and Andreas Klotzek for teaching the beauty of control engineering.

Being a solid base in my life, I would like to deeply thank my flatmates and friends, especially Gregor, Matthias and Philipp. Your long-lasting support, advice, and care have helped me in countless circumstances and make life great. It is a pleasure to have you by my side!

I would like to express my deepest thanks to the five people who represent my home, irrespective of where I currently live. This thanks goes to my mum for her continuous support and care, to my dad for teaching me many aspects of life, as well as to Veronika and Regina, who are the best sisters I could imagine. Finally, this thanks goes to my partner Lissy. With you, each common library visit became a holiday, each paper submission a celebration, each pandemic lockdown a retreat, and the last few years of dissertation one long summer season.

"What barrier is there that love cannot break?"

Mohandas Karamchand "Mahatma" Gandhi

Contents

CHAPTER 1

Introduction

1.1 Motivation

1.1.1 Why renewable energy transition?

The fight against human-induced climate change requires the reduction of greenhouse gas emissions, which is often headlined "decarbonization". In the energy sector, this goal is pursued by the transition to renewable energy. Besides the energy sector, also buildings, industry and transport are major emitters of greenhouse gases, and need to be decarbonized [1]. In order to supplement their hydrocarbon input, even more renewable energy is required. Thus, the transition to renewable energy can be considered as the core task in the fight against the human-induced climate change.

Human-induced climate change is an *unequivocal* reality according to [2]. However, for human individuals the full picture of climate change can be hard to grasp since it occurs on large scale in terms of time (decades) and space (global), as shown in Fig. 1.1. Consequently, it is also worth exploring, which potential co-benefits can be caused by the renewable energy transition in further domains like health, economy, security, politics and society. Each of these high-level co-benefits is related to concrete benefits of human individuals, and is likely to be experienced sooner and more local than climate change. In this regard, note that Fig. 1.1 shows only one possible version of temporal and spacial arrangement.

In the following subsections, the high-level co-benefits, their potentially overall positive effects on the human individuals, but also potential drawbacks are presented in the presumed temporal order of Fig. 1.1. Instead of claiming a comprehensive assessment, the goal is to stimulate discussions, and ideally to provide to each reader an individual motivation for the potentially biggest project of humanity to this date: the renewable energy transition.

Conservation of earth surface & prevention of radiation

Conventional fossil and nuclear energy lead to massive surface mining for lignite coal [3], poisoning of drinking water with fracking residuals [4, 5], or the contamination of land due to nuclear accidents as in Chernobyl (Ukraine) or Fukushima (Japan). However, it should be mentioned that also certain renewables like hydro and geothermal energy can consume significant amounts of land.

Often, this consumption of land comes along with the permanent loss of home for the local population, who is forced to migrate. For instance, as of 2018, the lignite mining in Germany has forced the resettlement of 120,000 people [3]. In contrast, wind energy and photovoltaics are geologically less invasive, do not contaminate land, and thus enable people to *keep their home*.

Conservation of air quality

Throughout wide parts of the world, the air is polluted by particulate matter, ozone, nitrogen dioxide, and sulfur dioxide. Indeed, 91% of the world's population are affected by excessive pollution, which increases their risk of stroke, heart disease, lung cancer, and respiratory diseases. Ultimately, ambient

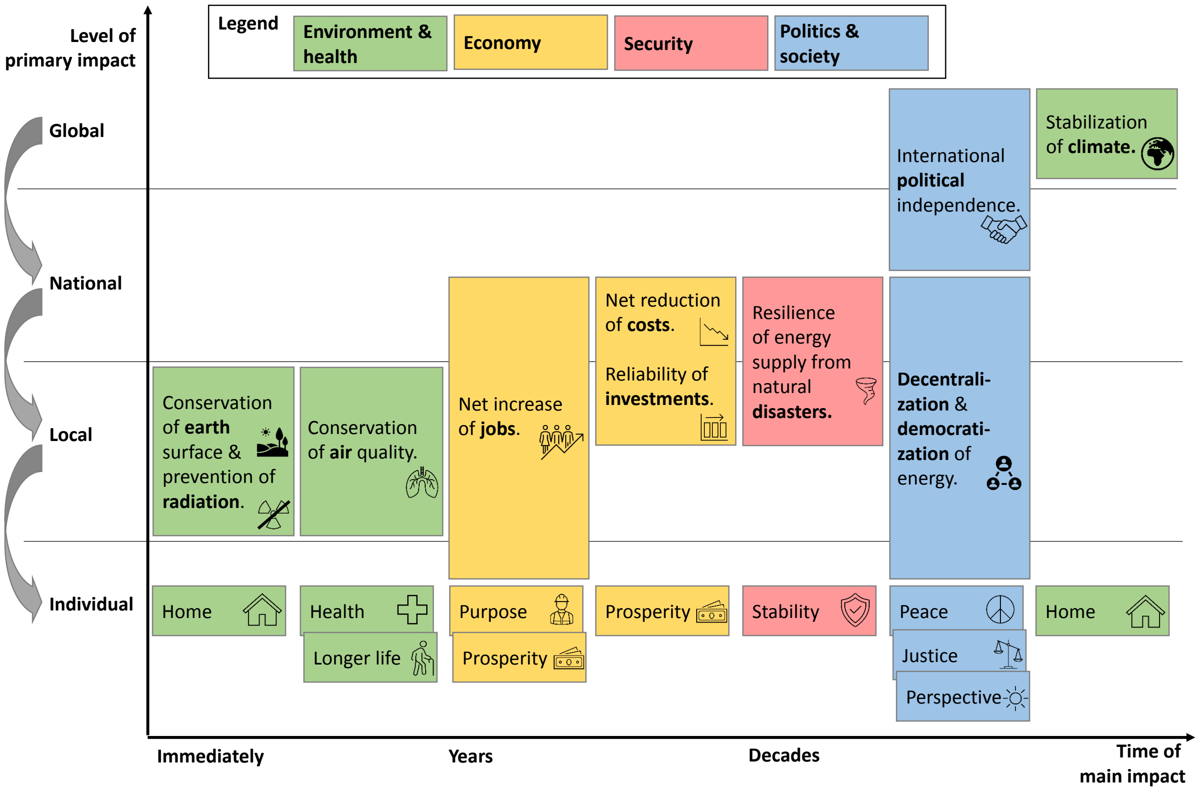


Figure 1.1: Potential benefits of the renewable energy transition. Colors: different domains. The box of each high-level benefit is positioned according to its presumed time of *main* impact and level of *primary* impact. Note that, clearly, a *primarily* national phenomenon can spread to the international and local levels at a secondary stage. The small boxes in the "Individual" row represent the derived effects to the human individuals.

air pollution kills 4.2 million people worldwide per year. Even in the EU, despite the enforcement of air quality standards, in average each citizen looses 8.6 months of lifetime. [6]

Since the energy sector is one of the key emitters for the above mentioned substances, the transition to renewable energy will *protect health* and *extend lives* of people in former proximity of fossil power stations.

Net increase of jobs

In the context of the existing energy sector, the phase-out of fossil energy clearly results in job cuts, which, however, can be compensated by retirement, retraining and reallocation of workers [3]. For the example of Germany, during the years 2005-2019, net employment in the electricity generation sector actually rose by 4.8% [7], while the share of renewables quadrupled from 10% to 40% [8]. In the context of new investments, studies estimate that $1 million spent in fossil energy creates 2.65 full-time-equivalent (FTE) jobs in the short-to-medium term, while the same amount in renewable energy creates 7.49 FTE jobs [9]. To conclude, investments shifted from fossil to renewable energy will provide extra employment, and consequently *purpose* and *prosperity*, to the people.

Net reduction of costs

As shown in Fig. 1.2, the Levelized Cost of Energy (LCOE) of coal energy could not be decreased in the past decade. One reason is that its LCOE is significantly dictated by fuel price, instead of plant

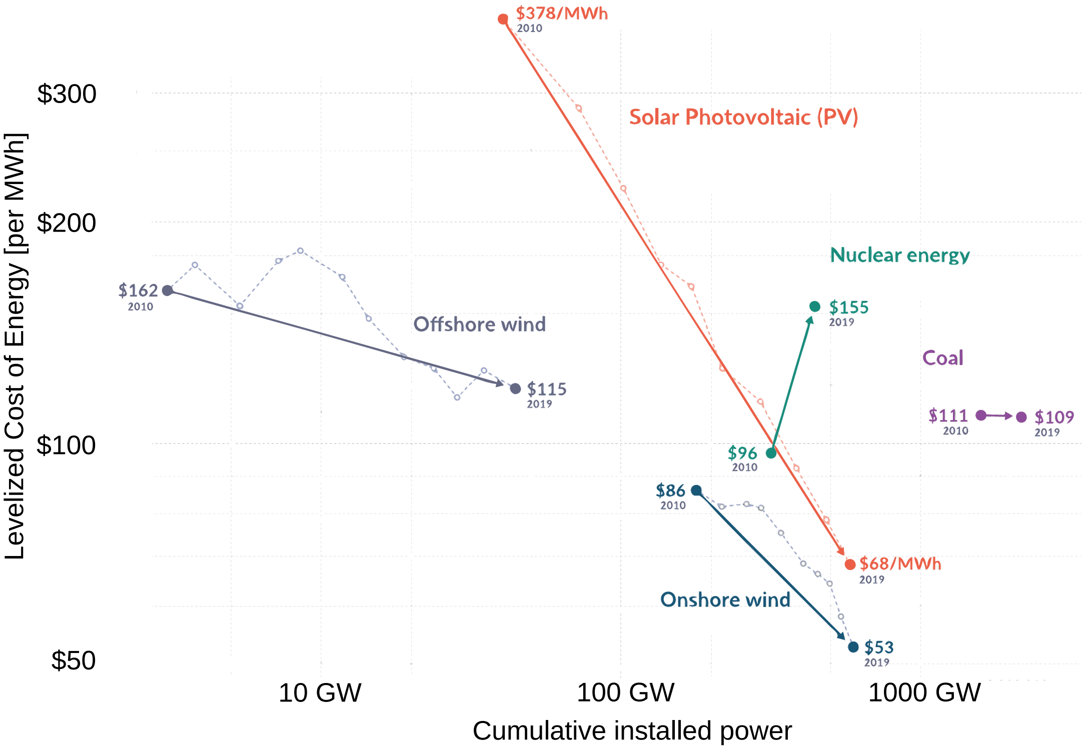


Figure 1.2: Double-logarithmic plot of LCOE [$/MWh] over cumulative installed power worldwide [MW] (modified from [10]).

investment cost. Consequently, like for every fossil energy, there is an LCOE barrier which cannot be overcome by innovation. Furthermore, due to the very likely increase of carbon prices, e.g. via emissions trading systems, coal energy will even become more expensive in the upcoming years. The LCOE of nuclear energy already increased, due to more costly safety requirements. In contrast, LCOEs of onshore wind energy (-70%) and solar photovoltaics (-89%) impressively decreased, and reached a competitive level. In the future, the decline of their LCOE is expected to continue due to technological advances and the economy of scale. [10]

Additionally, the prices for Li-ion battery packs, which help balancing the intermittent renewable supply and the demand, decreased by 89% within one decade. In the upcoming years, this trend is expected to persist as well. [11]

To conclude, a renewable energy system has the potential to provide cheaper electricity, and thus to *increase prosperity* of the individual people.

Reliability of investments

To date, a significant amount of money is still invested in fossil energy, despite these investments being in danger of becoming stranded. A worst-case study has shown the following: There is an enormous risk of investing in fossil projects under the hypothesis that the agreed climate goals would not be enforced by new laws. If those laws then are put in place after all, the demand for fossil energy would decrease. If then, low-price fossil energy suppliers "out-sell" their energy, high-price fossil-fuel assets of up to $12 trillion could end up as stranded. This financial volume corresponds to more than 15% of global gross domestic product, and actually could cause severe regional or global financial crises. [12]

Thus, the shift of investments from fossil to renewable mitigates wide-ranging financial risks, which ultimately *protects prosperity* of the individual people.

Resilience of energy supply from natural disasters

By their nature, a renewable energy system is more granular and decentralized than the conventional energy system. This offers the opportunity of locally aggregating power production units into microgrids, which can be connected to a national grid. In case of a natural disasters, where transmission lines may be cut or important power plants may be shut down, an advanced microgrid can disconnect from the national grid, black-start, and provide a self-sufficient local energy supply [13].

Thus, a smart management of renewable power units can increase the resilience of energy systems, and can *secure stability* of the consumers' lifestyle.

Decentralization & democratization of energy

In a conventional centralized energy system, the profits from energy supply are also centralized and primarily concentrated at the government and few corporations [14]. This can promote rent-seeking and corruption, and thus harm the political process [15]. In contrast, a renewable energy system typically is more decentralized, and its profits are "democratized" towards communities or even individual households. In Germany in the year 2016, private citizens, farmers and municipalities owned 44.9% of the installed renewable power capacity [16]. Also for the upcoming years, especially for photovoltaics (60%) and onshore wind energy (40%), end users and municipal utilities are expected to be key renewable investors in Germany [17].

An even higher impact of energy democratization can be expected for developing countries with regional lacks of electrification. Here, decentralized renewable energy and microgrids offer the "opportunity to leapfrog, not only fossil fuels, but, to some extent, the need for a centralized electricity grid", and thus facilitate electrification [14].

To conclude, the renewable energy transition can *promote political and economic justice*, and *provide perspective* to people who receive electricity supply for the first time.

International political independence & peace

At least 80% of the world's population lives in countries that are net importers of fossil fuels [14], and thus heavily depend on few other countries. Furthermore, fossil fuels often have to be transported through energy choke points like certain sea routes and channels, which are vulnerable to disruption by aggressive countries or criminals. Regarding the relationship to these producing or transmitting countries, the dependence can be used as a bilateral weapon and restrict political freedom. In contrast, in a renewable energy economy, many countries will be energy-independent and only trade technology (national scenario), or at least build up a diverse energy supplier base of neighboring countries (continental scenario) [18].

Apart from restricted political freedom, even more severe conflicts are likely to be started, maintained and extended due to the dependence on fossil fuels [19]. Oil and natural gas production is robustly correlated with outbreaks of civil war [20]. The long-term continuation of such wars is frequently financed by fossil fuel export. For instance, in 2015, the terrorist group "Islamic State of Iraq and Syria" had an estimated income of $500 million by oil sales [21]. Additionally, the presence of fossil fuels can involve further countries in such conflicts. A study based on 69 countries which had a civil war between 1945 and 1999 shows an increasing probability for third-party country military interference if the conflict country has higher oil reserves and if the third-party country has a stronger import dependence regarding oil [22]. Actually, national militaries are in a self-exciting dilemma since contemporary war machinery almost entirely runs on oil-based fuel. As a consequence, governments and military leaders consider oil supply as national security interest, and have to deploy more armed forces to support their fossil companies in controlling resources [23]. This shall be underlined by the following quotation:

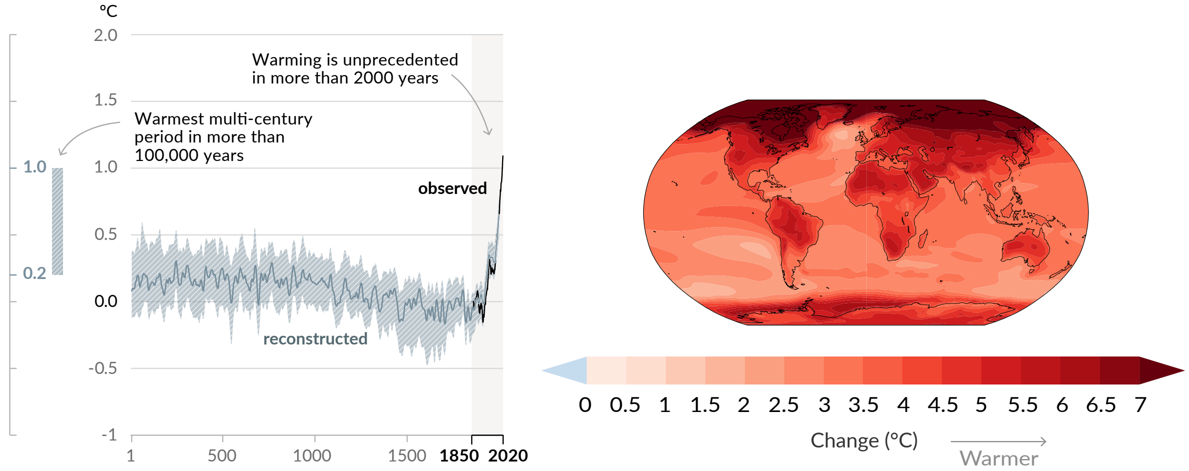


Figure 1.3: Left: Change in global surface temperature (decadal average) as reconstructed (1-2000) and observed (1850-2020); Right: Simulated change of annual mean surface temperature for 4 °C global warming relative to end of 19th century. [2]

> **Quotation (John McCain, USA, Republican party, former presidential candidate, [24]):**
> *"My friends, I will have an energy policy that we will be talking about, which will eliminate our dependence on oil from the Middle East that will prevent us from having ever to send our young men and women into conflict again in the Middle East."*

To conclude, the renewable energy transition has the potential to generate more sovereign and confident states, to eliminate some causes of wars, and finally to *promote peace* for the citizens.

Stabilization of climate

Studies report at high confidence that the atmospheric CO_2 concentration was higher in 2019 than it has even been for at least 2 million years [2]. As a consequence, in the second decade of the 21st century, the global surface temperature was already 1.09°C higher than at the end of the 19th century, as shown in Fig. 1.3 (left). If the trend of increasing greenhouse gas emissions is continued, a scenario with a mean temperature increase of 4°C by the year 2100 will be very likely. As shown in Fig. 1.3 (right), temperature increase over land is even higher and particularly pronounced around the north pole. This mean temperature increase already would have severe effects on humans, animals and vegetation. However, particularly extreme heat events and heavy precipitation events are expected to have even worse impact. These events already have become more frequent and intense, and are mainly caused by human-induced climate change [2]. Further effects of human-induced climate change on nature are the retreat of glaciers and the Arctic sea ice, the acidification of the surface open ocean, and the rise of the global mean sea level (0.2m by now). Since many of these effects are irreversible for the upcoming centuries and millenia, the reduction of greenhouse gas emissions by the renewable energy transition is mandatory in order to *save home* of the people.

1.1.2 Why wind energy & battery energy storage?

As shown in Fig. 1.2, particularly onshore wind energy has already reached a high installed power, has been scaled significantly, and has reached a very low LCOE. Thus, it is a main backbone of the renewable energy transition. Since onshore installation space is very limited, offshore wind energy will very likely gain importance in the next decades.

Due to the intermittent nature of most renewables, electrical energy storage is crucial for balancing supply and demand. Particularly Li-ion battery energy storage systems can absorb and release electrical energy very fast. Furthermore, they have gained technological maturity, can be scaled well, and have reached a viable price range.

1.1.3 Why control of wind turbines & batteries?

The operation of wind turbines or batteries is a complex task with often adversarial goals, which only can be handled via sophisticated automatic controllers. Put simple, wind turbines are operated with the aims of extracting maximum power in the partial load region, limiting power and rotor speed in the full load region, limiting extreme deflections, and reducing fatigue [25]. Batteries are operated with the aims of following a reference power, limiting State of Charge and temperature, and reducing calendric and cyclic aging [26]. Note that, in essence, the term "cyclic aging" is a battery-related synonym for "fatigue", and thus only the latter term will be used in the following.

1.1.4 Why control of fatigue?

Fatigue of wind turbines and batteries corresponds to the damage of device integrity via cyclic loading. Originally, the controller of the device has to make sure, that sufficient integrity is maintained until the planned end-of-life. Beyond that, controllers with improved fatigue mitigation capabilities can be utilized in two scenarios [27]:

- Retrofit scenario: If the fatigue rate of a device in the operation phase is further reduced by upgrading the controller, the device lifetime can be extended.
- Integrated design scenario: If the fatigue rate of a device in the design phase is further reduced by integrating a novel controller, less material can be specified, performance can be increased, or design lifetime can be extended.

In suitable market settings, both scenarios will translate to a reduction of LCOE, and ultimately to higher profit of the operator. In the present dissertation, both scenarios are targeted.

1.2 Research topics & innovative content

Based on the fatigue mitigation scenarios of Section 1.1.4, the present dissertation pursues the following vision:

> **Vision:** *Devices are aware of their fatigue behavior, and utilize this information to optimally balance revenue and fatigue.*

Based on this vision, the mission is derived, which comprises two research topics:

1. Improve the online *estimation of fatigue*.

2. Develop novel formulations for more accurate *Model Predictive Control of fatigue*.

These research topics contribute to both fatigue mitigation scenarios of Section 1.1.4. Particularly the integrated design scenario is investigated in **Papers 10** and **11** for wind turbines. These papers have not been led by the author of this dissertation. Consequently, the papers are shown in Chapter 5, but their content is neither included in the research topics of the present section, nor in the methods of Chapter 2.

Estimation and control of fatigue can be applied to various domains. The applications for the present dissertation stem from the renewable energy sector: wind turbines, battery energy storage systems, and hybrid systems combining both. In the following, for each research topic, the background is provided and the innovative content of the present dissertation is presented. Parts of the following text within this chapter are excerpts of the papers of Chapters 3 and 4.

1.2.1 Research topic 1: Estimation of fatigue

Background: The current value of accumulated fatigue provides information about the remaining useful lifetime of the device.

Since fatigue is caused by cyclic application of stress, fatigue estimation typically involves the decomposition of the input stress trajectory to stress cycles. Cycle identification is straightforward if, e.g., a simple sinusoid is analysed. There, amplitudes, mean values and number of cycles are obvious. However, realistic stress trajectories usually are highly complex and contain stress cycles that can be nested *("nested cycles")*. Additionally, half and full cycles can be present. The most widely accepted algorithm for cycle identification from complex trajectories is the Rainflow(-counting) algorithm (RFC) [28].

At a real machine, fatigue estimation is based on stress which can be derived from measurements or obtained from a co-simulation. In a simulation, stress can be obtained directly from the model. For both settings, total fatigue can be estimated batch-wise offline, or gradually online.

Online estimation of fatigue provides insight, which points in time and which actions have a decisive impact on the total fatigue level. In theory, at each online update step, a full batch-wise offline estimation could be performed. However, this would require storing the entire stress history since start of operation, and analyzing this high amount of data at every estimation step. Consequently, the batch-wise approach in an online setting is too expensive in terms of memory and computing power.

Instead, cumulative algorithms have emerged, which store and evaluate only the *residue*, a condensed set of historic stress samples. However, these cumulative algorithms provide only value-discontinuous updates of fatigue at discrete time instances or at discrete events:

- Online updates at discrete *time instances* are implemented in the moving window approach of Heinrich et al [29]. Here, the algorithm periodically gathers stress samples for a defined time range, identifies full stress cycles from the residue and this new stress set via the Rainflow algorithm, and stores the remaining half cycles in the residue. Here, the value-discontinuity stems from attributing only full cycles to the total fatigue value.

- Online updates at discrete *events* are implemented in the algorithm of Musallam et al [30]. Here, an own cycle identification algorithm is implemented, which however is derived from the original Rainflow algorithm. Instead of one residue set, two buffers for historic stress minima and maxima are utilized. For each of both buffers, a separate cycle identification algorithm is run, which is event-triggered by the appearance of a new minimum or maximum, respectively. Here, the value-discontinuity stems from only attributing completed full or half cycles to the total fatigue value.

For a very granular insight and for control purposes, a continuous update of fatigue is desirable.

Innovative content: This gap is closed in the present work, where a novel one-step value-continuous algorithm for fatigue estimation is developed, implemented and successfully tested. The differences with respect to the above algorithms are summarized in Table 1.1. In contrast to the above time- or event-triggered algorithms, the novel algorithm is executed for each new available stress sample, which can be provided at arbitrarily small time steps. In contrast to the above value-discontinuous fatigue updates for completed cycles, the novel algorithm also attributes the fatigue impact of ongoing incomplete cycles to the total fatigue value. Consequently, a smooth value-continuous fatigue update algorithm is achieved, which enables accurate monitoring of fatigue evolution, and serves as a useful basis for controller formulations.

Table 1.1: Comparison of previous and novel fatigue estimation algorithms

Property	**Heinrich et al. [29]**	**Musallam et al. [30]**	**Novel algorithm**
Triggered by	time	new extremum	new stress sample
Fatigue updated by	completed full cycle	completed full or half cycle	completed or ongoing full or half cycle
Value-continuous fatigue update	-	-	X
Original Rainflow algorithm utilized	X	-	X
Number of residue sets / buffers	1	2	1

In detail, the novel algorithm merges the residue set and the new stress sample, and identifies full cycles and incomplete half cycles via the Rainflow algorithm. Only those stress samples, which belong to half cycles, are stored in the residue. The fatigue damage of the full cycles is added to a dedicated fatigue state. This fatigue state represents the condensed fatigue impact of all historic full cycles. Based on this fatigue state and the fatigue of new half cycles, the total fatigue is updated.

In accordance with physics, practical tests reveal that this total fatigue value is increasing as long as the stress is oscillating. The total fatigue value remains constant for time periods where the stress is constant. Compared to an offline batch-wise fatigue estimation, the novel algorithm exhibits exactly the same end value of fatigue, and thus zero loss of information.

The novel online fatigue estimation algorithm is developed and implemented in **Paper 5**, serves as the basis for a novel controller formulation in the same paper, and is utilized to simulate fatigue of a battery energy storage system in **Papers 8** and **9**.

1.2.2 Research topic 2: Model Predictive Control of fatigue

Background: Model Predictive Control (MPC) enables the economically optimal management of systems by using predictions of their future response, including stress at crucial spots in the device structure. 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 Fig. 1.4. Two approaches within this class are as follows:

- *Stress rate penalization* is a common approach for considering fatigue in MPC. In Gros et al [31] and Evans et al [32], 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 a damping of stress oscillations. There is no consideration of stress cycles per se, and thus there is no direct link to fatigue.

- In *spectral methods*, damage is approximated by empirical functionals depending on spectral moments of the predicted stress signal [33]. A severe limitation of this approach is the inherent assumption of a narrow-banded Gaussian process, which is often violated in practical cases.

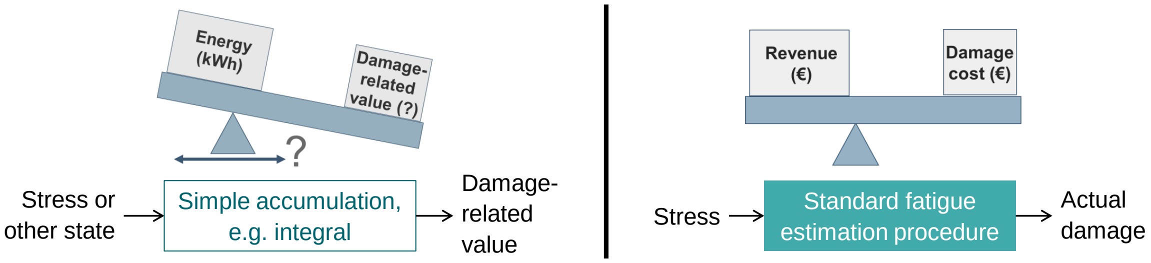


Figure 1.4: Left: Indirect fatigue metric. Right: Direct fatigue metric. [34]

In contrast, direct fatigue metrics evaluate actual damage, which can be readily converted to monetary fatigue cost, as illustrated in the right part of Fig. 1.4. The main approaches within this class are as follows:

- By a *piecewise-affine (PWA) dynamical system* [35, 36], a simplified fatigue metric is calculated in continuous time within the control horizon. Here, pseudo stress cycles are detected by sign changes in the stress signal, similarly to the *Simple Range Counting* method of ASTM [28]. This method represents an extreme simplification of RFC, because nesting of stress cycles is not accounted for.

- In Barradas-Berglind et al [37], *hysteresis operators* are used to adapt parameters of a cost function in the MPC. However, this cost function essentially penalizes deflection rates, similarly to the TTVP approach. Additionally, the control problem is reported to become significantly hard to solve.

- In Collet et al [38], *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 [38].

- In Luna et al [39], 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 [39].

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. Unfortunately, however, there are two properties of fatigue estimation which pose challenges regarding this implementation, as visualized in Fig. 1.5. First, Rainflow is a discontinuous branching algorithm, which is not per-se directly usable within a gradient-based optimization context, including MPC. Second, since fatigue is a long-term effect, the historic stress has to be considered, while typical MPCs optimize their control actions only based on the predicted states.

Innovative content: These challenges are overcome in the present work, where four novel Rainflow-based fatigue cost formulations are developed, implemented and successfully tested. Via these formulations, for the first time, the standard fatigue estimation procedure is implemented in an MPC. This implies that the controller has a more realistic perception of the current fatigue situation, and has increased capability of controlling fatigue. This proximity to the standard fatigue estimation procedure implies that the novel formulations are more transparent than, for instance, the Artificial Neural

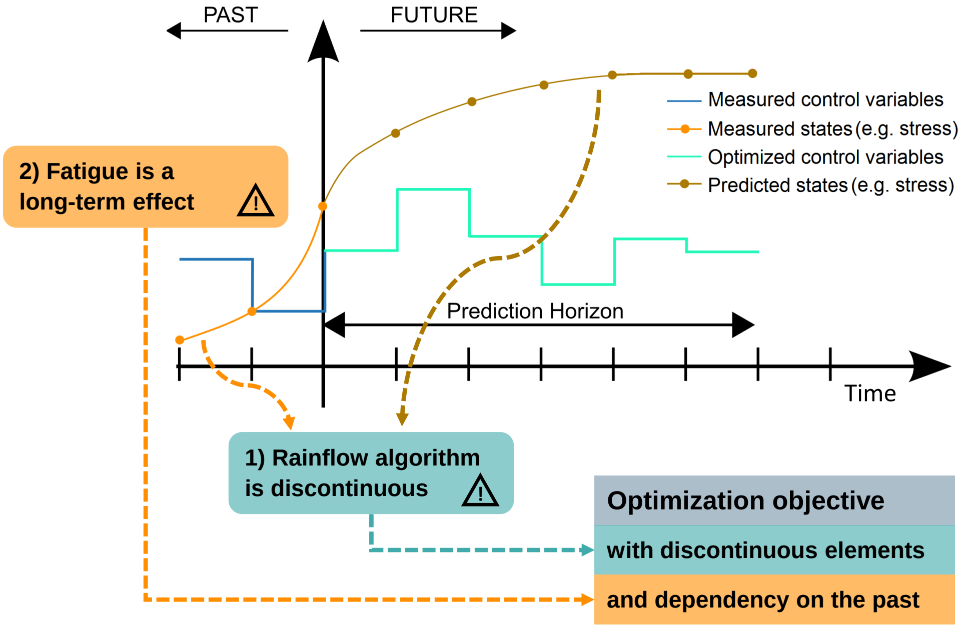

PAST
FUTURE
Measured control variables
Measured states (e.g. stress)
Optimized control variables
Predicted states (e.g. stress)
2) Fatigue is a long-term effect ⚠
Prediction Horizon
Time
1) Rainflow algorithm is discontinuous ⚠
Optimization objective
with discontinuous elements
and dependency on the past

Figure 1.5: Two properties of fatigue estimation which pose challenges to Model Predictive Control of fatigue (modified from [40]).

Network formulation of Luna et al [39]. From the algorithmic point of view, the added computational load by the fatigue cost is minor compared to the MPC execution, which facilitates the application in various domains. Considering these disruptive improvements, it can be argued that the four novel formulations of the present dissertation start a new family of fatigue cost formulations. Consequently, the way of controlling dynamic systems by these novel fatigue cost functions has been successfully protected by a worldwide patent [41].

In detail, the common solution of all novel formulations to the discontinuity of the Rainflow algorithm is to fix its result for one MPC step and employ a continuous proxy of the cycle identification result in the cost function. In the algorithm, after setting up the proxy, the gradients can be computed and the optimization problem can be solved.

The common solution of all novel formulations to the long-term nature of fatigue is to employ and update a residue set, and consider it in the preparation of the cost function.

Differences of the formulations particularly lie in the point where the Rainflow algorithm and the residue update are executed, and in the design of the above mentioned proxy. In the following, details on these differences and their effects on the practical implementation are explained for the individual formulations:

1. In the formulation "Direct Online Rainflow Counting" (DORFC), the Rainflow algorithm is executed within the MPC algorithm. At this point, the residue information can be utilized and updated. The fatigue proxy in the cost function comprises individual functions for each stress cycle. Since this formulation requires manipulation of the MPC algorithm and comprises a non-standard cost function, it has been implemented in an in-house MPC framework instead of a public standard MPC framework. The formulation is developed and tested extensively with a wind turbine model in **Papers 2** and **3**.

2. In the formulation "Parametric Online Rainflow Counting" (PORFC), the Rainflow algorithm and residue update are taken out (externalized) of the core MPC, but still executed at each MPC step. The results of fatigue estimation then are fed to the MPC via parameters. The fatigue

proxy within the MPC is a standard *economic* cost function. These modifications with respect to DORFC beneficially enable employing many standard MPC frameworks, while retaining a correct estimation of fatigue. This formulation is developed in **Paper 4** and extended by the residue concept from **Paper 5** in **Paper 6**. Further, PORFC is applied successfully to a variety of challenging applications comprising a high-fidelity wind turbine simulator in **Papers 6** and **7**, a Li-ion battery storage system in **Paper 8**, and a hybrid energy system in **Paper 9**.

3. The formulation "Tracking Online Rainflow Counting" (TORFC) is implemented identically to PORFC with the only differences that the parameters are further condensed and, instead of an economic cost function, a more conventional *tracking* cost function is utilized. This has the advantages of supporting even more MPC frameworks, and facilitating convergence of the optimization. A disadvantage is that part of the fatigue information is lost in the above mentioned condensing of the parameters. This formulation is developed in **Paper 4** and tested with a wind turbine model there.

4. In a fourth formulation, the Rainflow algorithm and residue update are incorporated deeply in the MPC-internal system model via a dedicated hybrid dynamical system, which directly computes fatigue. A separate hybrid dynamical system, but with the same structure, provides a continuous proxy for the fatigue cost gradient computation. Unfortunately, these hybrid dynamical systems impede the utilization of standard MPC frameworks. Instead, the advantages of this formulation have mainly been identified in the academic domain as yet. Casting fatigue estimation in the general framework of a hybrid dynamical system supports the understanding of fatigue control processes. Furthermore, by deriving successfully the above three formulations from the present one, they have been linked to a common basis, and their assumptions have been analyzed in a unified manner. This formulation is developed and successfully tested with a wind turbine model in **Paper 5**.

Finally, the existing literature as well as the formulation and application results of the present dissertation will enable the discussion of the following high-level questions in Section 6.1:

- How far can wind turbine fatigue be reduced by control, and how much is needed?
- How can cyclic aging control influence operational profit of batteries, and when is it applicable?
- How do the simulation setups influence the results?
- How and to which level of accuracy do the novel formulations represent fatigue?

1.3 Publications

Within this publication-based dissertation, eleven publications are included and referred to. In the following, an overview of the publication structure is presented, and their references are listed.

1.3.1 Overview of publications

Figure 1.6 shows a schematic overview of all publications (**Papers 1-11**). Here, the publications are grouped with respect to their main type of research work: development of novel *formulations*, *applications* of these formulations, and assessment of *integrated design* approaches. This grouping is coherent with the Chapters 3-5, where the individual papers are presented.

The publications led by the author of this dissertation are highlighted with lighter background. The co-authored papers contain significant scientific contributions, but are not led by the author. Published publications are shown with a solid edge, while a dotted edge indicates the review status. Three journal publications have emerged as extensions of conference publications, which is indicated by a solid arrow pointing to the journal publication. In all cases, the journal publications contain substantial extra content regarding formulations, more challenging test environments, or extended assessments. Finally, the formulations of two publications have been combined (indicated by the circle), and applied in four other publications (indicated by dashed arrows) .

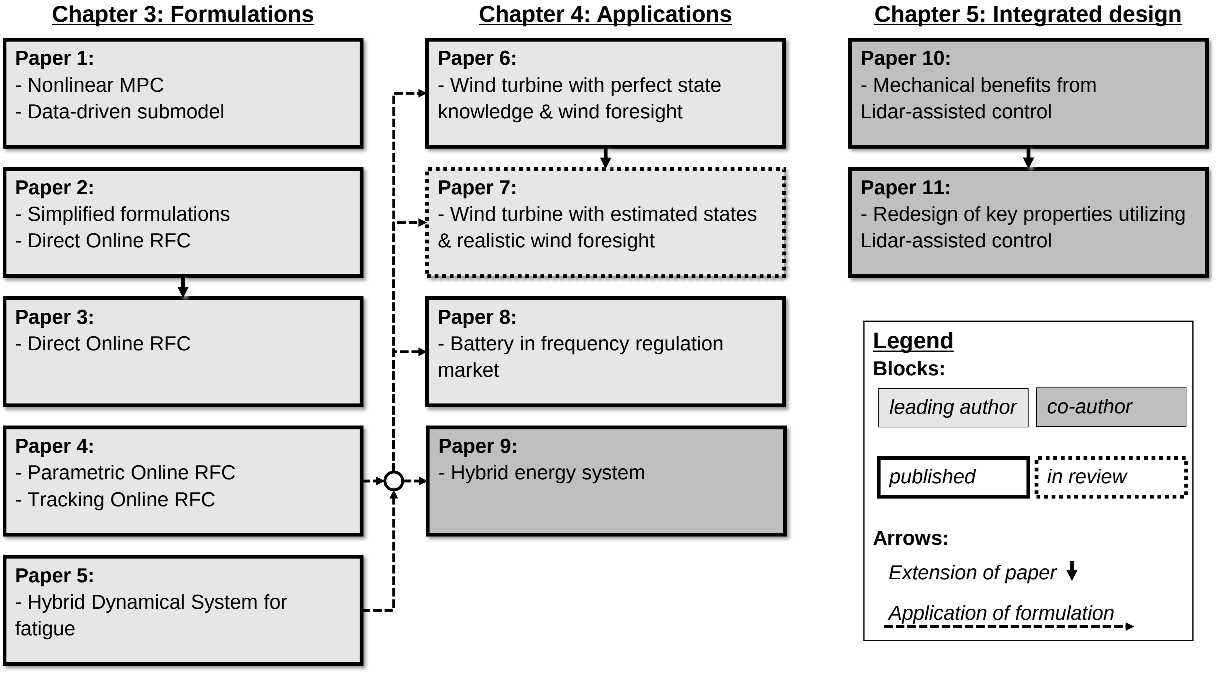

Chapter 3: Formulations
Chapter 4: Applications
Chapter 5: Integrated design
Paper 1:
- Nonlinear MPC
- Data-driven submodel
Paper 2:
- Simplified formulations
- Direct Online RFC
Paper 3:
- Direct Online RFC
Paper 4:
- Parametric Online RFC
- Tracking Online RFC
Paper 5:
- Hybrid Dynamical System for fatigue
Paper 6:
- Wind turbine with perfect state knowledge & wind foresight
Paper 7:
- Wind turbine with estimated states & realistic wind foresight
Paper 8:
- Battery in frequency regulation market
Paper 9:
- Hybrid energy system
Paper 10:
- Mechanical benefits from Lidar-assisted control
Paper 11:
- Redesign of key properties utilizing Lidar-assisted control
Legend
Blocks:
leading author
co-author
published
in review
Arrows:
Extension of paper
Application of formulation

Figure 1.6: Schematic overview of publications presented within this dissertation.

In **Paper 1**, a Nonlinear Model Predictive Control (NMPC) implementation is presented for hybrid system models which comprise physics-based dynamics and data-driven static behavior. This NMPC implementation serves as the basis for **Paper 2**, where simplified fatigue cost formulations and the idea for Direct Online Rainflow Counting (DORFC) are presented. Extending this publication, **Paper 3** provides a comprehensive derivation of DORFC and an extensive assessment of its capabilities and properties. For wider applicability with standard MPC frameworks, in **Paper 4**, the fatigue cost formulations Parametric (PORFC) and Tracking Online Rainflow Counting (TORFC) are derived from DORFC. A generalization of DORFC, PORFC and TORFC is provided in **Paper 5**, where the fatigue estimation process is cast into the framework of a hybrid dynamical system. Furthermore, the formulation of the one-step value-continuous fatigue estimation algorithm is developed in the same paper.

The concept of stress history via the residue is taken from **Paper 5** and added to the PORFC formulation from **Paper 4**. This combined formulation serves as the basis for all four application papers, since PORFC is conveniently applicable but still very close to the original fatigue estimation process. In **Paper 6**, the resulting novel MPC is applied to a high-fidelity wind turbine simulator, and compared to a conventional MPC and a conventional PID controller. This paper comprises the assumptions that all required wind turbine states are measured and also the incoming wind is perfectly known. These assumptions are dropped in **Paper 7**, which extends the latter paper by employing a Moving Horizon Estimator and a realistic Lidar simulator, and by performing a much broader assessment. The wide applicability of the present formulations is emphasized in **Paper 8**, where PORFC is controlling the cyclic aging of a battery energy storage system. Going further, in **Paper 9**, a hybrid energy system comprising one wind turbine and a battery system is effectively controlled.

The benefits of the novel MPCs can be exploited in both the retrofit and the integrated design scenario, as shown in Section 1.1.4. Particularly the integrated design scenario is pursued in **Papers 10** and **11**; however utilizing a Lidar-assisted conventional controller, instead of an MPC. In **Paper 10**, a design method is presented to rapidly assess the mechanical benefits of Lidar-assisted control at the first stages of wind turbine design. This work is expanded in **Paper 11**, where the controller upgrade is assessed in combination with turbine redesign options, and where the economic feasibility is analyzed more deeply. By revealing the potential of fatigue reduction for wind turbines, **Papers 10** and **11** also support the motivation for **Papers 1** - **7**, and put them into a new perspective.

1.3.2 List of publications

The following publications have been included into this thesis. All eleven publications have been submitted to peer-reviewed conferences or journals that are either Scopus- or Web-of-Science-listed. Ten of these publications have been published already, while one is in the review process. In the individual sections of Chapters 3-5, for each publication a summary is provided, and the contribution of the author of this dissertation is stated. As each publisher has granted a reprint permit, a copy of each paper is also included in each section.

Publications led by the author:

- **Paper 1**: S. Loew and D. Obradovic, “Real-time implementation of nonlinear model predictive control for mechatronic systems using a hybrid model,” in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, 2018, pp. 164–167.
  Reprinted in: S. Loew and D. Obradovic, “Real-time nonlinear model predictive control: Predictive control for mechatronic systems using a hybrid model,” *atp magazin*, no. 08, pp. 46–52, 2018

- **Paper 2**: S. Loew, D. Obradovic, and C. L. Bottasso, “Direct online rainflow-counting and indirect fatigue penalization methods for model predictive control,” in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3371–3376. doi:10.23919/ECC.2019.8795911

- **Paper 3**: S. Loew, D. Obradovic, and C. L. Bottasso, “Economic nonlinear model predictive control of fatigue—formulation and application to wind turbine control,” *Optimal Control Applications and Methods*, 2022. doi:10.1002/oca.2870

- **Paper 4**: S. Loew, D. Obradovic, A. Anand, and A. Szabo, “Stage cost formulations of online rainflow-counting for model predictive control of fatigue,” in *2020 European Control Conference (ECC)*, 2020, pp. 475–482. doi:10.23919/ECC51009.2020.9143939

- **Paper 5**: S. Loew and D. Obradovic, “Formulation of fatigue dynamics as hybrid dynamical system for model predictive control,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6616–6623, 2020. doi:10.1016/j.ifacol.2020.12.080

- **Paper 6**: S. Loew, D. Obradovic, and C. L. Bottasso, “Model predictive control of wind turbine fatigue via online rainflow-counting on stress history and prediction,” *Journal of Physics: Conference Series*, vol. 1618, p. 22041, 2020. doi:10.1088/1742-6596/1618/2/022041

- **Paper 7 (in review)**: S. Loew and C. L. Bottasso, “Lidar-assisted model predictive control of wind turbine fatigue via online rainflow-counting considering stress history,” *Wind Energy Science Discussions*, vol. 2021, pp. 1–31, 2021. doi:10.5194/wes-2021-119

- **Paper 8 (publication led in equal amount by Abhinav Anand)**: S. Loew, A. Anand, and A. Szabo, “Economic model predictive control of li–ion battery cyclic aging via online rainflow–analysis,” *Energy Storage*, 2021. doi:10.1002/est2.228

Publications with significant scientific contribution by the author:

- **Paper 9** : A. Anand, S. Loew, and C. L. Bottasso, “Economic control of hybrid energy systems composed of wind turbine and battery,” in *2021 European Control Conference (ECC)*, 2021, pp. 2565–2572. doi:10.23919/ ECC54610.2021.9654911

- **Paper 10**: H. Canet, S. Löw, and C. L. Bottasso, “Lidar-assisted control in wind turbine design: Where are the potential benefits?” *Journal of Physics: Conference Series*, vol. 1618, p. 042020, 2020. doi:10.1088/1742-6596/1618/4/042020

- **Paper 11**: H. Canet, S. Loew, and C. L. Bottasso, “What are the benefits of lidar-assisted control in the design of a wind turbine?” *Wind Energy Science*, vol. 6, no. 5, pp. 1325–1340, 2021. [Online]. Available: https://wes.copernicus.org/articles/6/1325/2021/. doi:10.5194/wes-6-1325-2021

CHAPTER 2

Methods

This chapter provides a summary of existing and novel methods which contribute to the research topics of Section 1.2. As this is a cumulative dissertation, these methods have been published previously in the papers of Chapters 3 and 4. Consequently, this chapter contains also excerpts of these publications. At the end of the chapter, the utilized system models are presented for completeness.

2.1 Estimation of fatigue

In order to perform model-based control of fatigue, a suitable model of fatigue estimation is required. As a basis, the phenomenon of fatigue is defined, and the widely used Rainflow algorithm is presented, which is a key ingredient for accurate fatigue estimation. Next, the need for long analysis time frames is stressed, and the existing solution of a residue is provided. Finally, a novel value-continuous fatigue estimation model is introduced which will serve as a basis for control design in Section 2.2.

2.1.1 Definition of fatigue

Fatigue is damage of a material caused by cyclic application of stress. Without loss of information, the fatigue impact of a given stress-trajectory can be analyzed solely based on its extrema or "reversals". This implies that the shape and contained frequencies of the original continuous stress trajectory are considered to be irrelevant for fatigue estimation [37]. Therefore, the fatigue impact of such a reversal sequence is fully determined by its contained individual stress cycles. Each stress cycle can be represented by a cosine function. A stress trajectory typically contains full cycles, which are cosines of a full period, and half cycles, which are cosines of only a half period, as shown in Fig. 2.1.

Half cycles therefore represent either a rising or falling transient. Instead of storing three (full cycle) or two (half cycle) stress samples, it is common to store two stress samples and a weight, which

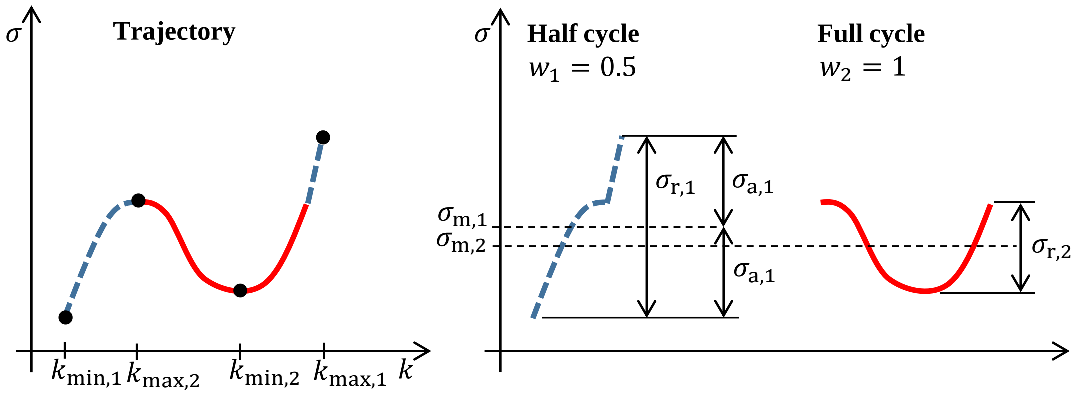


Figure 2.1: Stress trajectory which comprises one full cycle (red) and one rising half cycle (blue-dashed).

is valued $w_c = 1$ (full cycle) or $w_c = 0.5$ (half cycle). The two stress samples can be the cycle stress maximum and minimum, or the stress amplitude $\sigma_{\mathrm{a},c}$ and mean $\sigma_{\mathrm{m},c}$. Instead of stress amplitude, stress range $\sigma_{\mathrm{r},c} = 2\sigma_{\mathrm{a},c}$ is frequently used as well.

Typically, fatigue impact of a stress cycle mainly correlates with its stress amplitude. As a secondary factor, a positive stress mean increases and a negative stress mean decreases the fatigue impact. Quantitatively, this *mean stress effect* is considered by correcting the stress amplitude to an equivalent stress via the *Goodman equation* [53] (p. 184)

$$\sigma_{\mathrm{eq},c} = \sigma_{\mathrm{a},c} \frac{R_\mathrm{m}}{R_\mathrm{m} - \sigma_{\mathrm{m},c}} \tag{2.1}$$

where R_m denotes the ultimate tensile stress. Consequently, equivalent stress is used to calculate the number of cycles to failure $N_c = f_{\mathrm{SN}}^{-1}(\sigma_{\mathrm{eq},c})$ via the inverse S-N or *"Woehler"* curve, which typically has a piecewise definition over the stress-axis. Fatigue damage of a given stress cycle $D_{\mathrm{fatigue},c} = 1/N_c$ is obtained by the reciprocal of the number of cycles to failure. Assuming linear and time-invariant damage accumulation, the *Miner-Palmgren Rule* [54] is used to compute the total fatigue damage of the stress time series as

$$D_{\mathrm{fatigue}} = \sum_c D_{\mathrm{fatigue},c}. \tag{2.2}$$

2.1.2 Cycle identification via the Rainflow algorithm

Cycle identification is straightforward if, e.g., a simple sinusoid is analysed. There, amplitudes, mean values and number of cycles are obvious. However, realistic stress trajectories usually are highly complex and contain stress cycles that can be nested *("nested cycles")*. Additionally, half and full cycles can be present, as stated above. The most widely accepted algorithm for cycle identification from complex trajectories is the Rainflow(-counting) algorithm (RFC) [28].

The Rainflow algorithm contains algorithmic branches and loops. Thus, a crucial property of the Rainflow algorithm is its discontinuous output behavior. Furthermore, the number N_c of identified cycles is not known before execution, but bounded by the number of extrema. The characteristics of the identified cycles that are output by the Rainflow algorithm

$$[\sigma_{\mathrm{a},c}, \sigma_{\mathrm{m},c}, k_{\mathrm{max},c}, k_{\mathrm{min},c}, w_c] = RFC(\sigma(k)) \tag{2.3}$$

for each cycle c are 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 $k_{\mathrm{min},c}$ [-], and cycle weight $w_c[-]$.

2.1.3 Long time frames & residue

Originally, Rainflow analysis is performed batch-wise on the entire stress history. However, in [29] it is shown that Rainflow analysis also can be performed on a moving window. Furthermore, fatigue is a long-term effect, and correct evaluation requires knowledge about the entire stress history. As an extreme example, the very first stress sample after commissioning of the machine can form a stress cycle with the current stress sample several years later. These *transition cycles* grow over a long period of time and can reach high stress amplitudes with dominating fatigue impact [55]. Since transition cycles per definition have not been closed yet, they appear as half cycles in the Rainflow analysis. Thus, to account for transition cycles in the moving-window algorithm, the corresponding half-cycle stress samples are buffered in the so-called *residue* [56].

Definition (Residue): *The residue denotes a set of stress samples that occured in the past and have not formed full cycles as yet.*

Depending on the stress signal, a high number of samples can be accumulated in the residue. Highest possible dimensions of the residue vector result from diverging and converging stress time series because they result in a very high number of half cycles [56]. However, long-term diverging series are unrealistic because unstable machine behavior typically is counteracted by the controller or an emergency shutdown. Long-term converging series are irrelevant, since very low-amplitude cycles can be neglected without significant error in fatigue estimation. Concluding, the dimension N_{res} of the residue vector is finite and in practical tests remained well below 100.

2.1.4 Value-continuous fatigue estimation

Motivation

The moving-window Rainflow analysis of [29] results in periodic updates of identified full cycles. This corresponds to a value-discrete update of fatigue cost. However, gradient-based optimization in MPC requires value-*continuous* behavior of cost functions. Therefore, also the dynamics of fatigue cost need to be expressed in a continuous fashion. Additionally, just like the plant states, fatigue cost should be updated on arbitrarily small timesteps unlike the larger time periods of the moving-window algorithm.

Solution

Both goals are achieved by the following two novel steps of enhancement from **Paper 5**:

As the first step, the moving-window algorithm is adapted to a one-step cycle identification which is shown in Alg. 1. The advantage is that this algorithm does not pause until a new string of stress samples is available, but directly provides an update of full cycles with each new stress sample.

Algorithm 1 One-step Rainflow-analysis

```
Input: existing stress string $\boldsymbol{\sigma}_{\text{exist}}$; periodic update of a scalar new stress sample $\sigma$
Output: full cycles
Initialization: stress residue $\boldsymbol{x}_{\text{res}} = \boldsymbol{\sigma}_{\text{exist}}$
while true do
    Merge residue $\boldsymbol{x}_{\text{res}}$ and new stress sample $\sigma$
    Extract full cycles from merged set using Rainflow algorithm; store residue in $\boldsymbol{x}_{\text{res}}$
end
```

As the second step, Alg. 1 is enhanced by the additional consideration of half cycles which start "growing" already due to infitesimally small variations in stress. This consideration results in continuous output of the computation of fatigue cost which is appended to the algorithm, like stated in Alg. 2.

These enhancements do not introduce further assumptions. Thus, like shown in Fig. 2.2, this continuous fatigue cost calculation provides the same output like a batch evaluation over the entire stress trajectory. As expected, the fatigue cost is monotonously increasing, and thus the fatigue cost rate is never negative.

Application

Algorithm 2 is applied to simulate cyclic aging within the battery plant models of **Papers 8** and **9**, and will be the basis for the MPC-internal fatigue model of Section 2.2.7.

Algorithm 2 One-step value-continuous fatigue cost estimation

```
Input: existing stress string $\boldsymbol{\sigma}_{\text{exist}}$; update of a scalar new stress sample $\sigma(k+1)$ at each step $k$
Output: fatigue cost $J_{\text{fatigue}}(k)$
Initialization: zero total fatigue cost $J_{\text{fatigue}}(0) = 0$ and fatigue cost $x_{\text{fatigue,FC}}(0) = 0$ of full cycles; stress residue $\boldsymbol{x}_{\text{res}}(0) = \boldsymbol{\sigma}_{\text{exist}}$, $k = 0$
while true do
    Merge residue $\boldsymbol{x}_{\text{res}}(k)$ and new stress sample $\sigma(k+1)$
    Extract full and half cycles from merged set using Rainflow algorithm; store residue in $\boldsymbol{x}_{\text{res}}(k+1)$
    Calculate fatigue cost based on full cycles; add result to $x_{\text{fatigue,FC}}(k)$ to obtain $x_{\text{fatigue,FC}}(k+1)$
    Calculate fatigue cost based on full and half cycles; add result to $x_{\text{fatigue,FC}}(k)$ to obtain $J_{\text{fatigue}}(k+1)$
    $k = k + 1$
end
```

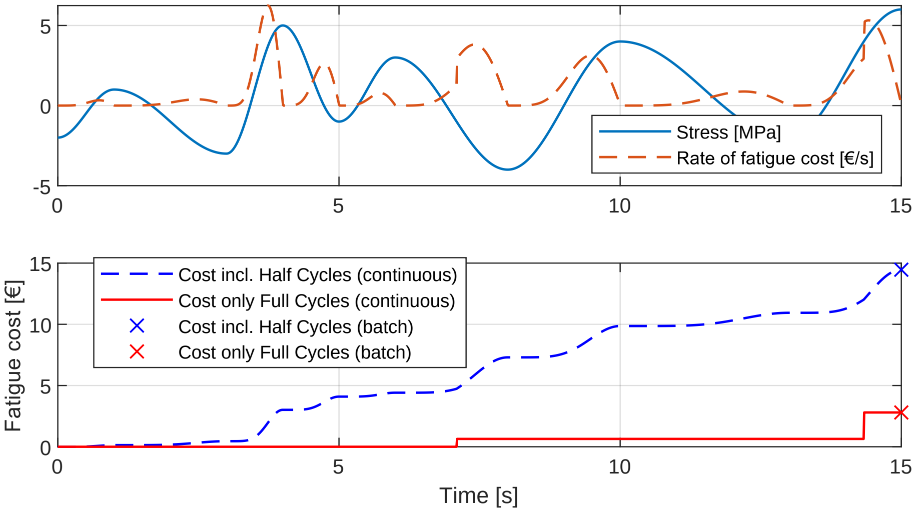


Figure 2.2: Top: Exemplary input stress trajectory, and fatigue cost rate. Bottom: value-continuous and batch estimation using only full cycles or full+half cycles.

2.2 Model Predictive Control of fatigue

For various technical devices, an important task of operation is the economic balancing of revenue and fatigue cost. This is a classical optimal control problem (OCP). Model Predictive Control (MPC) enables the solution of this OCP on a moving horizon while respecting constraints on system states and inputs [57]. For reference, important variants and properties of MPC are provided in Appendix A. In a generic form, the optimization problem for control of fatigue is

$$\min_{\bar{u}} -J_{\text{revenue}}(\boldsymbol{x}) + \alpha_{\text{damage}} J_{\text{fatigue}}(\cdot) \tag{2.4a}$$

subject to

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t)) \tag{2.4b}$$

$$\boldsymbol{x}(t_0) = \tilde{\boldsymbol{x}} \tag{2.4c}$$

$$\boldsymbol{G}(\bar{\boldsymbol{x}}) = \mathbf{0} \tag{2.4d}$$

$$\boldsymbol{H}(\bar{\boldsymbol{x}}) \leq \mathbf{0} \tag{2.4e}$$

$$\boldsymbol{u}_{\min} \leq \bar{\boldsymbol{u}} \leq \boldsymbol{u}_{\max} \tag{2.4f}$$

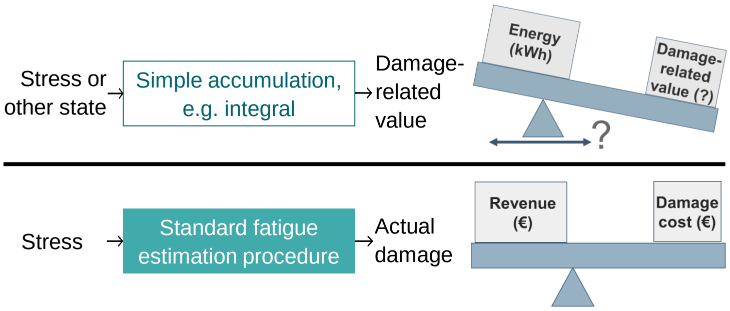


Figure 2.3: Top: indirect fatigue metric. Bottom: direct fatigue metric.

where revenue J_{revenue} is balanced with fatigue cost J_{fatigue} by the weight α_{damage}. The continuous nonlinear system dynamics $\boldsymbol{F}$ depend on the system states $\boldsymbol{x}(t)$, the control variables $\boldsymbol{u}(t)$ and the external disturbances $\boldsymbol{d}(t)$. Fatigue is caused by stress $\sigma(t)$, which can be one of the states or outputs of the system. Practical representations of system dynamics can be found for a wind turbine in Section 2.3.4, and for a battery energy storage system in Section 2.3.5.

At each MPC step, the states are initialized by their measurements or estimates $\tilde{\boldsymbol{x}}$ which are obtained from the plant. The state samples $\bar{\boldsymbol{x}}$ at the control intervals are constrained by equality $\boldsymbol{G}$ and inequality constraint sets $\boldsymbol{H}$. Finally, the control variable samples $\bar{\boldsymbol{u}}$ are limited by lower $\boldsymbol{u}_{\min}$ and upper bounds $\boldsymbol{u}_{\max}$.

The revenue cost function typically is a stage cost of the states $\boldsymbol{x}$. In contrast, the fatigue cost function can take more complex forms. An overview of existing fatigue cost functions from the literature is provided for wind turbines in **Paper 3** and for battery energy storage systems in **Paper 8**. Fatigue costs can be based either on an *indirect* or *direct* fatigue metric, as shown in Fig. 2.3. For indirect fatigue metrics, only a damage-related value is optimized and thus the link to economics is difficult. In contrast, direct fatigue metrics return actual damage, which can be converted to monetary fatigue cost. As stated in **Papers 3** and **8**, the existing direct fatigue cost functions in the literature either are inaccurate or computationally expensive. Accurate fatigue estimation is enabled by the Rainflow algorithm (see Section 2.1.2) which, however, is discontinuous and thus cannot be differentiated. Thus, in essence, the goal is to find direct, accurate, computationally cheap and differentiable fatigue cost functions J_{fatigue}.

In the following, three existing simplified fatigue costs will be presented for reference. Later, four novel fatigue cost formulations will be presented which fulfill the above goal to a very high degree.

2.2.1 Simplification by stress rate penalization

The quadratic penalization of stress rate

$$J_{\text{fatigue,TTVP}} = \int_{t_0}^{t_{\text{end}}} \dot{\sigma}(t)^2 \mathrm{d}t \tag{2.5}$$

is a common approach for considering fatigue in MPC. In Gros et al [31] and Evans et al [32], 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 a sort of damping of stress oscillations. There is no consideration of stress cycles per se, and thus there is no direct link to fatigue. However, since this cost function is the most common basis for comparison in the literature, it will be used for the same purpose under the label "Tower Tip Velocity Penalization" (TTVP) in **Papers 2-7**.

2.2.2 Simplification by stress amplitude penalization

In the standard fatigue estimation process, stress amplitudes of separate cycles are considered as key fatigue drivers (see Section 2.1.1). A constant stress trajectory results in zero fatigue. Thus, fatigue damage can be penalized approximately by penalizing stress amplitudes $\sigma_\mathrm{a}(k) = \bar{\sigma}(k) - \hat{\sigma}_\mathrm{m}$ w.r.t. a stress mean $\hat{\sigma}_\mathrm{m}$ over the entire MPC horizon. The notation $\bar{(\cdot)}$ means sampled on the N_u control intervals of the prediction horizon, while $\hat{(\cdot)}$ means fixed for one MPC step. The resulting fatigue cost function

$$J_\text{fatigue,AP} = \sum_{k=1}^{N_u} \sigma_\mathrm{a}(k)^2 \tag{2.6}$$

is a sum of squares of stress amplitudes, and is utilized in **Paper 2**.

2.2.3 Simplification by piecewise-affine cycle identification

In the MPC formulation of [35], cycle identification is approximated via a piecewise-affine dynamical system. Therefore, this formulation is referenced as "PWA" in the present work. This formulation corresponds to the *Simple Range Counting* of [28] which is a strong simplification of the Rainflow analysis. In short, the fatigue cost function

$$J_\text{fatigue,PWA} = \sum_{k=1}^{N_u} \boldsymbol{\sigma}_\mathrm{r}(k)^2 \tag{2.7}$$

is a sum of squares of stress ranges $\boldsymbol{\sigma}_\mathrm{r} = (\sigma_\text{r,rise}(k), \sigma_\text{r,fall}(k))^T$ which are treated separately for rising and falling half cycles. These stress ranges are obtained by the piecewise-affine dynamical system

$$\boldsymbol{\sigma}_\mathrm{r}(k+1) = \boldsymbol{A}(k)\boldsymbol{\sigma}_\mathrm{r}(k) + \boldsymbol{b}(k)\,|\sigma(k+1) - \sigma(k)| \tag{2.8}$$

where the system matrix $\boldsymbol{A}(k)$ and the input vector $\boldsymbol{b}(k)$ are switched at each stress extremum. More details can be found in **Paper 8**. In essence, the stress ranges are reset to zero at each stress extremum and grow with stress evolution until the next extremum.

2.2.4 Incorporation into the MPC algorithm

As mentioned in Section 2.1.2, the Rainflow algorithm is the de facto standard for fatigue estimation. Thus, a direct and non-simplified implementation of the Rainflow algorithm in MPC is desirable and promises best results.

Obstacles

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

■ **Obstacle 1:** The Rainflow algorithm requires the entire stress trajectory over the prediction horizon as an input [36], and thus cannot be cast into standard stage or terminal cost functions (see Appendix A.5).

■ **Obstacle 2:** It does not seem possible to derive analytical expressions of fatigue cost gradients $\boldsymbol{g}^*_\text{fatigue}$, due to the Rainflow algorithm not having a "closed mathematical form" [33, 58–60].

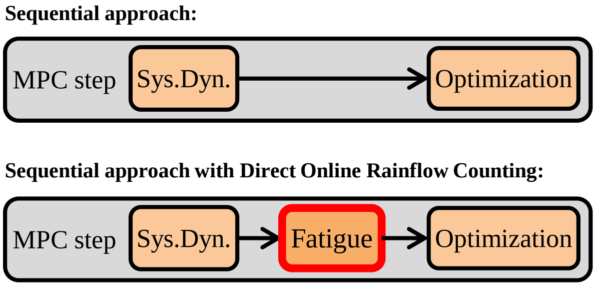
■ **Obstacle 3:** The Rainflow algorithm is a highly nonlinear algorithm [37], which exhibits discontinuous outputs because of its branches and loops (see Section 2.1.2).

Solution

In **Papers 2** and **3**, the above mentioned obstacles are 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. This applies if the *Sequential approach* is pursued, as explained in Appendix A.3 and visualized in the upper part of Fig. 2.4.

Via the insertion of code in between these two steps, as shown in the lower part of Fig. 2.4, 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.


Sequential approach:

Sequential approach with Direct Online Rainflow Counting:

Figure 2.4: Integration of DORFC into the Sequential MPC approach.

■ **"Substitute":** Obstacle 2 refers arguably to the absence of a "closed mathematical form", which is questioned in the present work. As shown in **Paper 3**, the algorithmic loops in the Rainflow algorithm are always executed a finite number of times. Thus, Rainflow can actually be classified as a "closed" algorithm. 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 Rainflow 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-dependent execution structure of the Rainflow 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 **Paper 3**), 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 $(k_{\max,c}, k_{\min,c})$.*

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 Fig. 2.5.

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

$$\sigma_{\mathrm{m},c} = \frac{\sigma(\hat{k}_{\max,c}) + \sigma(\hat{k}_{\min,c})}{2} \quad \text{and} \quad \sigma_{\mathrm{a},c} = \frac{\sigma(\hat{k}_{\max,c}) - \sigma(\hat{k}_{\min,c})}{2}. \tag{2.9}$$

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

To conclude, each execution of DORFC consists firstly of the evaluation of the "Discontinuous fatigue cost program" based on the Rainflow algorithm, and secondly of the evaluation of the "Fatigue cost gradient program" based on the continuous substitutes, as shown in Alg. 3. The computation steps of the fatigue cost program and of the fatigue cost gradient program will be specified in subsequent subsections.

Algorithm 3 MPC using DORFC

```
Input: measurement of system states, initial guess of control variables
Output: update of control variables
while true do
    Receive new initial states from measurement
    Solve system dynamics via single-shooting
    Evaluate "Discontinuous fatigue cost program", using the Rainflow algorithm (2.3)
    Evaluate "Fatigue cost gradient program", using the continuous stubstitutes (2.9)
    Set up and solve QP
    Apply new control variables to plant
    Execute steps 2: to 4:, using current initial states and new control trajectories
    Update Hessian matrix via BFGS method, using old and new control trajectories and cost gradients
end
```

■ **"Switch seldom":** Obstacle 3, which refers to the "discontinuous output" of the algorithm, is especially relevant for the present work, since the Real-time Iteration (RTI) approach is applied (see Appendix A.2). In fact, RTI requires the cost function and its gradients to only change mildly for successive MPC steps. This can be achieved if:

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

This requirement of seldom switches can be linked to the concept of "average dwell-time" in switching cost functionals [61]. 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 work. However, studies using an exemplary system in **Paper 3** show that indeed changes in the execution structure of the Rainflow algorithm happen infrequently enough over the simulation time.

Continuous Fatigue Cost Program

The fatigue cost of one cycle is a polynomial

$$J_{\text{fatigue},c}(\sigma_{\text{eq},c}) = w_c \sum_{n=2}^{N} a_n \, \sigma_{\text{eq},c}^n \tag{2.10}$$

with a minimum monomial 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 derived from material and economic data, as shown in **Paper 3**.

Following the Linear Damage Accumulation approach by *Miner & Palmgren* [54], the total fatigue cost of a specific stress time series is obtained by the linear accumulation of fatigue cost of all N_c detected cycles c, i.e.

$$J_{\text{fatigue,DORFC}} = \sum_{c=1}^{N_c} J_{\text{fatigue},c}. \tag{2.11}$$

Fatigue Cost Gradient Program

Differentiation of the continuous fatigue cost program yields the fatigue cost gradient program (Fig. 2.5 *d)* to *e)*), which can be written as

$$\frac{\mathrm{d}J_{\text{fatigue,DORFC}}}{\mathrm{d}\bar{u}_{i,j}} = \frac{\partial J_{\text{fatigue,DORFC}}}{\partial J_{\text{fatigue},c}} \operatorname{diag}\left(\frac{\partial J_{\text{fatigue},c}}{\partial \sigma_{\text{eq},c}}\right) \left[\operatorname{diag}\left(\frac{\partial \sigma_{\text{eq},c}}{\partial \sigma_{\text{m},c}}\right) \frac{\partial \sigma_{\text{m},c}}{\partial \sigma(k)} + \operatorname{diag}\left(\frac{\partial \sigma_{\text{eq},c}}{\partial \sigma_{\text{a},c}}\right) \frac{\partial \sigma_{\text{a},c}}{\partial \sigma(k)}\right] \frac{\mathrm{d}\sigma(k)}{\mathrm{d}\bar{u}_{i,j}}. \tag{2.12}$$

Automatic Differentiation (AD) obtains exact numeric gradients online in a step-wise manner by following the chain rule. The forward execution of the original program (Fig. 2.5*c)*) determines which steps are taken, i.e. which execution structure of the Rainflow algorithm can be currently observed.

In the present work the fatigue program has only one output, represented by total fatigue cost $J_{\text{fatigue,DORFC}}$, and a large number of input optimization variables $\bar{u}_{i,j}$. In such situations, *Reverse AD* is more efficient [62]. Accordingly, calculations are executed top-down as shown in the "Fatigue cost gradient program" in Fig. 2.5*e)*. Numerical values of each chain element of the differentiated program (2.12) are computed by manually coded symbolic gradients.

Noteworthy, the gradient of the continuous substitutes (2.9) w.r.t. the stress trajectory samples yields

$$\frac{\partial \sigma_{\text{m},c}}{\partial \sigma(k)} = \begin{cases} \frac{1}{2} \text{ if } k = k_{\text{max},c} \\ \frac{1}{2} \text{ if } k = k_{\text{min},c} \\ 0 \text{ otherwise} \end{cases} \quad \text{and} \quad \frac{\partial \sigma_{\text{a},c}}{\partial \sigma(k)} = \begin{cases} \frac{1}{2} \text{ if } k = k_{\text{max},c} \\ -\frac{1}{2} \text{ if } k = k_{\text{min},c} \\ 0 \text{ otherwise.} \end{cases} \tag{2.13}$$

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 $\partial\sigma_{\text{a}}/\partial\sigma$ will take the following form: $\begin{pmatrix} \cdot & -\frac{1}{2} & \cdot & \frac{1}{2} & \cdot & \cdot \\ \frac{1}{2} & -\frac{1}{2} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \frac{1}{2} & \cdot & -\frac{1}{2} \end{pmatrix}$.

2.2.5 Externalization from the MPC algorithm via an Economic Stage Cost

Challenges

Direct Online Rainflow Counting (DORFC) exhibits the following challenges w.r.t. formal analysis and implementation in standard MPC-frameworks:

■ **Challenge 1:** This Rainflow-based definition of fatigue cost is a discontinuous function of all time-samples of stress within the prediction horizon. Thus, neither the concept of stage cost nor of terminal cost applies. Formal proofs of stability and recursive feasibility usually are researched for those standard concepts ([63], [64], [65] p.112 ff.). Consequently, many standard MPC-frameworks like `ACADO Toolkit` [66] as well only allow for stage and terminal costs.

■ **Challenge 2:** The structure of the fatigue cost function is redefined at every MPC step. This fact as well poses difficulties concerning formal proofs. Additionally, such redefinitions are not allowed for in many standard MPC-frameworks.

■ **Challenge 3:** DORFC requires a valid and continuous stress trajectory during the preparation phase of the Quadratic Program (QP). In case of a Multiple-shooting [64] MPC-framework, the state

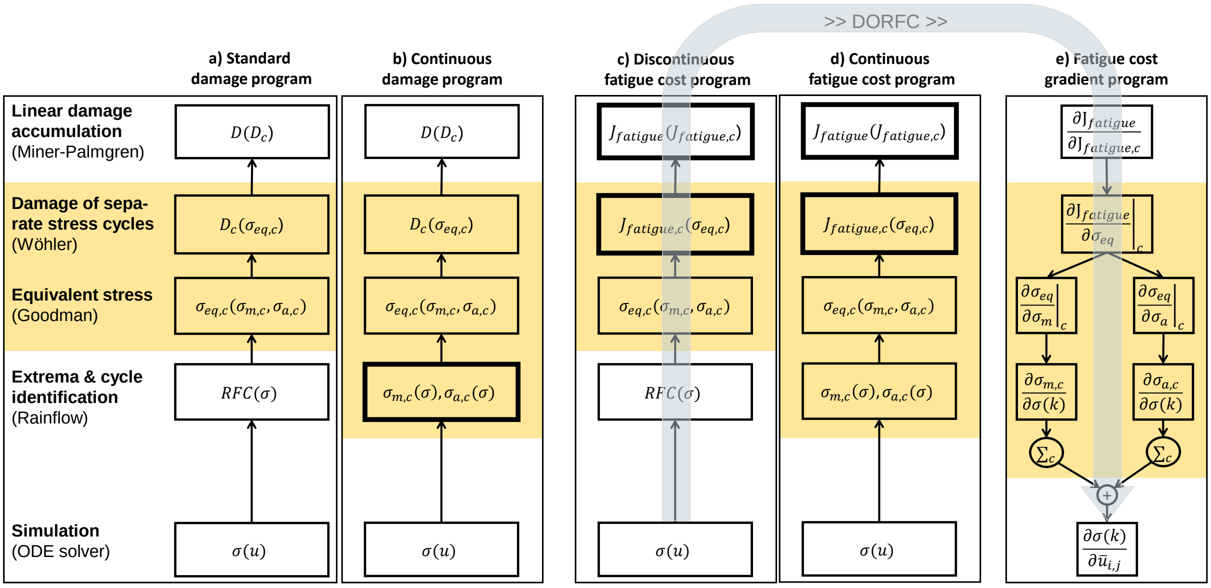

>> DORFC >>
a) Standard damage program
b) Continuous damage program
c) Discontinuous fatigue cost program
d) Continuous fatigue cost program
e) Fatigue cost gradient program
Linear damage accumulation (Miner-Palmgren)
Damage of separate stress cycles (Wöhler)
Equivalent stress (Goodman)
Extrema & cycle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

Figure 2.5: Standard damage program *a)*. Substitution of $RFC(\sigma)$ by linear continuous functions (2.9) 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)* and *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)*.

trajectories can exhibit discontinuities after the preparation phase of the MPC. As a consequence, a Rainflow-analysis of such a discontinuous stress trajectory might exhibit unrealistic extra cycles, and the definition of the fatigue cost function would be wrong.
■ **Challenge 4:** In DORFC, fatigue cost is evaluated solely based on the states in the prediction horizon. However, fatigue is a long-term effect, and correct evaluation requires knowledge about the entire stress history.

Solution

The above Challenges 1 and 2 are overcome by the principle "Decouple", which refers to decoupling the fatigue cost in time in order to achieve a standard Economic Stage Cost formulation with time-varying parameters (see Appendix A.4). Challenge 3 is overcome by the principle "Externalize", which refers to performing the fatigue estimation externally and prior to each MPC execution in order to generate these time-varying parameters, as shown in Fig. 2.6. Challenge 4 is overcome by the principle "Memorize", which refers to memorizing the past stress via the residue from Section 2.1.3.

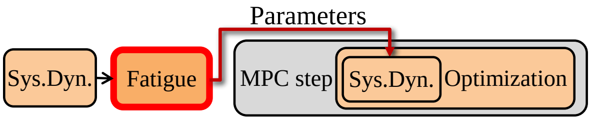


Figure 2.6: Externalization of fatigue estimation from the Simultaneous MPC approach.

■ **"Decouple":** Inserting (2.9) into the Goodman equation (2.1) and the latter into fatigue cost of individual cycles (2.10), total accumulated fatigue cost (2.11) can be expressed by

$$J_{\text{fatigue,DORFC}} = \sum_{c=1}^{N_c} J_{\text{fatigue},c}\left(\sigma(\hat{k}_{\text{max},c}), \sigma(\hat{k}_{\text{min},c}), \hat{w}_c\right). \tag{2.14}$$

This equation highlights that the fatigue cost of each cycle c depends on two stress samples whose positions in the prediction horizon are determined by $\hat{k}_{\text{max},c}$ and $\hat{k}_{\text{min},c}$. These positions and the cycle weights $\hat{w}_c$ are outputs of the Rainflow-algorithm.

Consequently, the fatigue cost of individual cycles (2.14) is split into two halves

$$J_{\text{fatigue,DORFC}} \approx \sum_{c=1}^{N_c}\left(\frac{1}{2}J_{\text{fatigue},c}\left(\sigma(\hat{k}_{\text{max},c}), \hat{\sigma}(\hat{k}_{\text{min},c}), \hat{w}_c\right) + \frac{1}{2}J_{\text{fatigue},c}\left(\hat{\sigma}(\hat{k}_{\text{max},c}), \sigma(\hat{k}_{\text{min},c}), \hat{w}_c\right)\right) \tag{2.15}$$

where for each term only one stress value σ remains variable and the complementary stress value $\hat{\sigma}$ becomes a parameter. For instance for the first cost term, stress mean now is calculated by

$$\sigma_{\text{m},c} = \frac{\sigma(\hat{k}_{\text{max},c}) + \hat{\sigma}(\hat{k}_{\text{min},c})}{2} \quad \text{and stress amplitude by} \quad \sigma_{\text{a},c} = \frac{\sigma(\hat{k}_{\text{max},c}) - \hat{\sigma}(\hat{k}_{\text{min},c})}{2}. \tag{2.16}$$

Based on practical considerations in **Paper 4**, this is reformulated to

$$J_{\text{fatigue,DORFC}} \approx \sum_{c=1}^{N_c}\left(\frac{1}{2}J_{\text{fatigue},c}\left(\sigma(\hat{k}_{\text{max},c}), \hat{\sigma}_{\text{m},c}, \hat{w}_c\right) + \frac{1}{2}J_{\text{fatigue},c}\left(\sigma(\hat{k}_{\text{min},c}), \hat{\sigma}_{\text{m},c}, \hat{w}_c\right)\right) \tag{2.17}$$

where the mean stress value instead of the complementary stress values is a parameter. For both cost terms, stress amplitude now is calculated by

$$\sigma_{\text{a},c} = \left|\sigma(\hat{k}_{\text{max/min},c}) - \hat{\sigma}_{\text{m},c}\right|. \tag{2.18}$$

By the preceding derivation, the fatigue cost calculation was decoupled in time which is shown in Fig. 2.7 (right) for an exemplary case of four stress cycles.

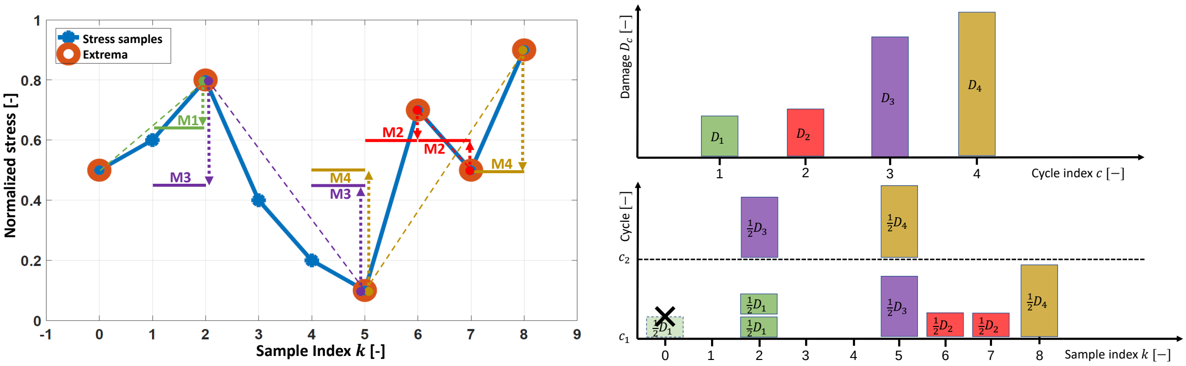


Figure 2.7: Left: Initial stress trajectory (blue), its extrema (orange circles), generated time-varying mean stresses (solid green, red, purple, yellow) and optimization goals (dotted arrows) for PORFC. Right: Corresponding distribution of damage over the prediction horizon. Both figures are modified from [67].

Thus, this formulation of a summation over cycles c with each two temporal terms can be converted to the stage cost formulation

$$J_{\text{fatigue,DORFC}} \approx \sum_{k=1}^{N_u}\left(J_{\text{fatigue},c}\left(\sigma(k), \hat{\sigma}_{\text{m},c1}(k), \hat{w}_{c1}(k)\right) + J_{\text{fatigue},c}\left(\sigma(k), \hat{\sigma}_{\text{m},c2}(k), \hat{w}_{c2}(k)\right)\right) \tag{2.19}$$

of a summation over samples k with each two cycle terms. Only two terms for two cycles $c1$ and $c2$ are needed, since each stress sample contributes to maximum two stress cycles [68]. Since only stress extrema contribute to stress cycles, the parameters of mean stress $\left(\hat{\sigma}_{\mathrm{m},c1}(k), \hat{\sigma}_{\mathrm{m},c2}(k)\right)$ and cycle weight $(\hat{w}_{c1}(k), \hat{w}_{c2}(k))$ are nonzero only for those samples within the horizon. This is shown in Fig. 2.7 (left).

For the present MPC implementation, a time-continuous stage cost function is needed. Thus, the summation-type formulation finally is approximated by the integral-type formulation

$$J_{\text{fatigue,PORFC}}\left(\sigma, \bar{\boldsymbol{p}}\right) = \\ \frac{1}{T_{\text{cntrl}}} \int_{t_0}^{t_{\text{end}}} \left(J_{\text{fatigue},c}\left(\sigma(t), \hat{\bar{\sigma}}_{\mathrm{m},c1}(t), \hat{\bar{w}}_{c1}(t)\right) + J_{\text{fatigue},c}\left(\sigma(t), \hat{\bar{\sigma}}_{\mathrm{m},c2}(t), \hat{\bar{w}}_{c2}(t)\right)\right) \mathrm{d}t \quad (2.20)$$

where the parameters

$$\bar{\boldsymbol{p}} = \left(\hat{\bar{\sigma}}_{\mathrm{m},c1}, \hat{\bar{\sigma}}_{\mathrm{m},c2}, \hat{\bar{w}}_{c1}, \hat{\bar{w}}_{c2}\right) \quad (2.21)$$

are defined piecewise constant over the control intervals of the prediction horizon. The factor $\frac{1}{T_{\text{cntrl}}}$ is prepended to approximate the discrete sum (2.19) by the present time-integral.

■ **"Externalize":** The PORFC formulation requires the parameters $\bar{\boldsymbol{p}}$ to be available before each MPC execution. This computation has to be based on the same stress trajectory like the one which is simulated within the MPC. This is achieved by performing the fatigue estimation based on a forward pre-simulation, as shown in Alg. 4 and Fig. 2.9.

■ **"Memorize":** Before executing the Rainflow analysis, the pre-simulated stress trajectory is merged with the residue, as shown in Alg. 4 and Fig. 2.9. Later on, the residue is updated based on the results of the Rainflow analysis.

Algorithm 4 MPC using Parametric Online Rainflow-counting (PORFC)

```
Input: measurement of system states, initial guess of control variables
Output: update of control variables
while true do
    Pre-simulation: The reduced plant model is simulated over the prediction horizon using the latest measured states as initial values. Relevant result is a stress prediction, as visualized in Fig. 2.8 (right).
    Merge: The residue is merged with the stress prediction.
    Rainflow: The Rainflow algorithm is used to identify stress cycles over this merged trajectory. Consequently, according to the principle "Switch seldom" from DORFC, it is assumed that the cycle structure does not change within the upcoming optimization run. As shown in Fig. 2.8, this assumption implies that the controllable extrema in the prediction horizon only can be shifted vertically by the optimization.
    Residue update: The measured initial value $\sigma(t_0)$ is added to the residue. Furthermore, the results of the Rainflow analysis are used to discard certain samples of the residue. Further details are stated in Paper 6.
    Time-varying parameters: Information from the Rainflow analysis is used to set the time-varying parameters, which are forwarded to the cost function of the MPC.
    Optimization/MPC: In the cost function of the MPC, the parameters are used to time-continuously calculate fatigue cost over the horizon and accumulate it via integration. Finally, the optimization problem is solved and the resulting control variables are applied to the plant.
end
```

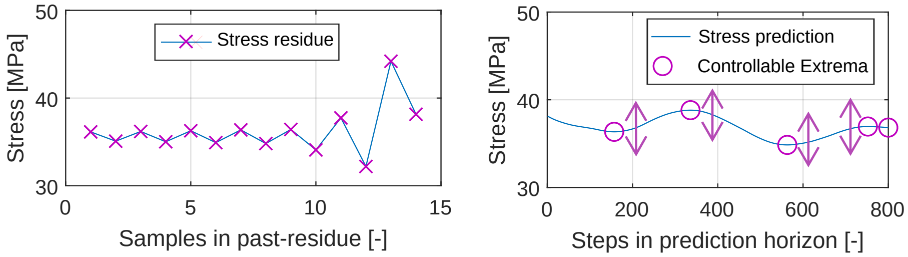


Figure 2.8: Left: Stress residue from the past. Right: Stress prediction in the future.

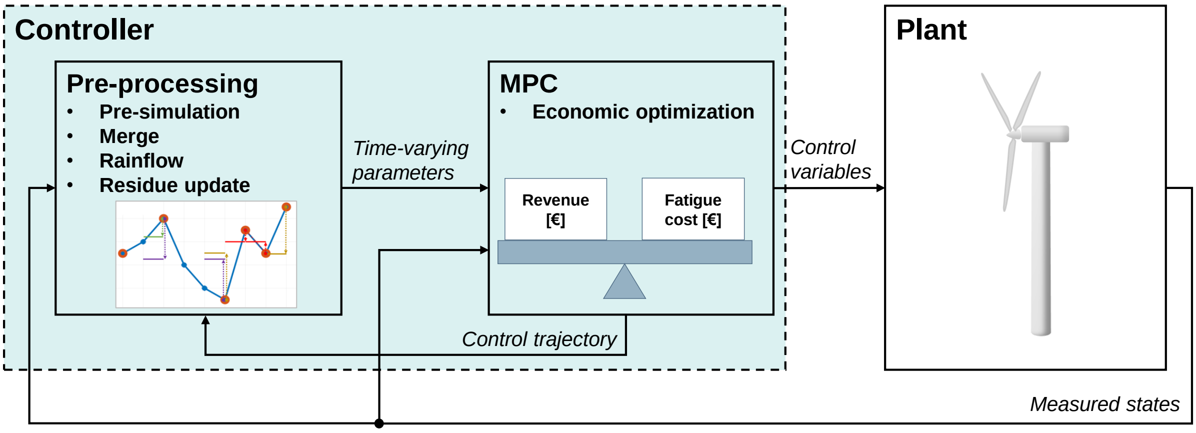


Figure 2.9: Externalization of fatigue estimation from the MPC via PORFC.

Application

The formulation of PORFC is developed in **Paper 4** and extended by the residue concept from **Paper 5** in **Paper 6**. Further, PORFC is applied to a high-fidelity wind turbine simulator in **Paper 6** and **Paper 7**, to a Li-ion battery storage system in **Paper 8**, and to a hybrid energy system in **Paper 9**.

2.2.6 Externalization from the MPC algorithm via a Tracking Stage Cost

Motivation

The Economic Stage Cost formulation (2.20) effectively corresponds to a pointwise penalization of deviation of stress w.r.t. parametric mean stress values $\hat{\hat{\sigma}}_{\mathrm{m},c1}$ and $\hat{\hat{\sigma}}_{\mathrm{m},c2}$. This formulation already is very close to a Tracking MPC formulation (see Appendix A.4), which typically is characterized by simpler implementation and improved convergence.

Solution

In the following, a Tracking Stage Cost is achieved by "Condensing" the time-varying parameters, and by "Matching" the cost function, as shown in **Paper 4**. The new formulation will be referred to as *Tracking Online Rainflow-counting* (TORFC).

■ **"Condense":** In TORFC, stress mean becomes the tracking reference for the system state of stress. In contrary to PORFC, at each sample, only a scalar stress mean can be defined. Thus, in case of two stress mean values in PORFC, those two values have to be condensed to a scalar value for TORFC. One straightforward and fairly meaningful way of condensing them seems to be a definition of the tracking reference

$$\hat{\bar{\sigma}}_{\text{ref}} = \frac{J_{\text{fatigue},c1}\,\hat{\bar{\sigma}}_{\text{m},c1} + J_{\text{fatigue},c2}\,\hat{\bar{\sigma}}_{\text{m},c2}}{J_{\text{fatigue},c1} + J_{\text{fatigue},c2}} \tag{2.22}$$

by a fatigue-cost-weighted average. Analogously to stress mean, also the cycle weights are condensed to a scalar cycle weight of

$$\hat{\bar{w}}_{c,\text{ref}} = \frac{J_{\text{fatigue},c1}\,\hat{\bar{w}}_{c1} + J_{\text{fatigue},c2}\,\hat{\bar{w}}_{c2}}{J_{\text{fatigue},c1} + J_{\text{fatigue},c2}}. \tag{2.23}$$

Analogously to PORFC (2.18), the stress amplitude

$$\sigma_{\text{a}}(t) = \left|\sigma(t) - \hat{\bar{\sigma}}_{\text{ref}}(t)\right| \tag{2.24}$$

can be represented by the stress tracking deviation.

Since mean stress is the reference now, the variable mean stress effect (2.1) degenerates to a fixed linear amplification

$$\sigma_{\text{eq}} = \sigma_{\text{a}}\,\hat{\bar{w}}_{\text{m}} \tag{2.25}$$

of stress amplitude by the factor

$$\hat{\bar{w}}_{\text{m}} = \frac{R_{\text{m}}}{R_{\text{m}} - \hat{\bar{\sigma}}_{\text{ref}}}. \tag{2.26}$$

■ **"Match":** In Tracking MPC, generic stage costs

$$J_{\text{Tracking}}\left(\sigma, \hat{\bar{\sigma}}_{\text{ref}}, \hat{\bar{w}}_{\text{T}}\right) = \int_{t_0}^{t_{\text{end}}} \hat{\bar{w}}_{\text{T}}(t)\,\left(\sigma(t) - \hat{\bar{\sigma}}_{\text{ref}}(t)\right)^2 \mathrm{d}t \tag{2.27}$$

involve a quadratic penalization of deviations from the state reference $\hat{\bar{\sigma}}_{\text{ref}}$ which can be weighted by the time-varying tracking weight $\hat{\bar{w}}_{\text{T}}$. Matching the present fatigue cost formulation (2.10) to this pattern only allows for an exponent of $m = 2$ in the fatigue cost function

$$J_{\text{fatigue},c}(\sigma_{\text{eq}}, \hat{\bar{w}}_{c,\text{ref}}) = \hat{\bar{w}}_{c,\text{ref}}\,a_2\,\sigma_{\text{eq}}^2 \tag{2.28}$$

of individual stress cycles. The coefficient a_2 can be tuned to approximate the original cost function (2.10) in a desired range of equivalent stress σ_{eq}.

Inserting (2.24) into (2.25) into (2.28) leads to the fatigue cost

$$J_{\text{fatigue},c}(\sigma, \hat{\bar{\sigma}}_{\text{ref}}, \hat{\bar{w}}_{c,\text{ref}}) = \hat{\bar{w}}_{c,\text{ref}}\,a_2\,\hat{\bar{w}}_{\text{m}}^2\,\left(\sigma(t) - \hat{\bar{\sigma}}_{\text{ref}}(t)\right)^2 \tag{2.29}$$

for individual stress samples. Analogously to (2.20), this leads to a total fatigue cost of

$$J_{\text{fatigue,TORFC}}\left(\sigma, \hat{\bar{\sigma}}_{\text{ref}}, \hat{\bar{w}}_{c,\text{ref}}\right) = \frac{1}{T_{\text{cntrl}}}\int_{t_0}^{t_{\text{end}}} J_{\text{fatigue},c}(\sigma, \hat{\bar{\sigma}}_{\text{ref}}, \hat{\bar{w}}_{c,\text{ref}})\mathrm{d}t \tag{2.30}$$

which finally is matched with the generic Tracking Stage Cost (2.27) to

$$J_{\text{fatigue,TORFC}}\left(\sigma, \hat{\bar{\sigma}}_{\text{ref}}, \hat{\bar{w}}_{\text{T}}\right) = \int_{t_0}^{t_{\text{end}}} \hat{\bar{w}}_{\text{T}}(t)\,\left(\sigma(t) - \hat{\bar{\sigma}}_{\text{ref}}(t)\right)^2 \mathrm{d}t \tag{2.31}$$

with the tracking weight

$$\hat{\bar{w}}_{\text{T}}(t) = \frac{\hat{\bar{w}}_{c,\text{ref}}\,a_2\,\hat{\bar{w}}_{\text{m}}^2}{T_{\text{cntrl}}}. \tag{2.32}$$

Implementation

TORFC is implemented in MPC similarly to PORFC also based on the principles "Externalize" and "Memorize" (see Section 2.2.5). The only difference lies in the time-varying parameters (2.21) which in the formulation of TORFC are replaced by the time-varying tracking references (2.22) and weights (2.32).

2.2.7 Generalization & incorporation into the MPC-internal model

Motivation

The previous formulations of fatigue estimation in MPC have been practically motivated and contained workarounds. In order to gain a deeper understanding and to put those formulations into perspective, the following more abstract questions are raised and solved:

■ **Question 1:** In DORFC, PORFC and TORFC, the fatigue dynamics are treated separately from the remaining MPC-internal system model. However, treating fatigue as part of the system model (see Fig. 2.10) seems natural and promises a more generic formulation. Thus, is it possible to include fatigue estimation including residue into the MPC-internal system model?

■ **Question 2:** An assessment in **Paper 4** shows that the PORFC fatigue cost function actually penalizes stress samples based on other stress samples which lie not only in the past but also in the future. This corresponds to *acausal* behavior. However, other approaches for fatigue control in the literature use causal surrogate models for fatigue dynamics [31, 37, 39]. Thus, can correct fatigue control also be achieved by a cost formulation with causal properties?

■ **Question 3:** In PORFC, a standard stage cost formulation is achieved by decoupling fatigue cost in time, and externalizing its estimation from the MPC. However, are standard stage or terminal cost formulations without those workarounds possible as well?

■ **Question 4:** As with the previous fatigue cost formulations, gradient-based MPC should be used. Thus, can gradients be computed from the novel fatigue cost formulation?

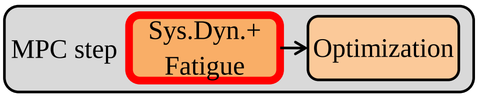


Figure 2.10: Incorporation of fatigue estimation into the MPC via a hybrid dynamical system (red block).

Solution strategy

Question 1 is solved by utilizing the one-step fatigue estimation including residue of Alg. 2. However, this is an algorithm, and reformulation to a known class of dynamical systems is desired. Since fatigue estimation comprises physical value-continuous states and algorithmic discrete states, it can be "Divided" into suitable subsystems and represented by a hybrid dynamical system. A background on hybrid dynamical systems is provided in **Paper 5**.

Via the residue, full information about the prior stress evolution is present in the state of the hybrid system. Thus, at any time within the prediction horizon, the fatigue accumulated up to this point can be computed only based on the instantaneous hybrid states. This corresponds to causal behavior, and solves Question 2. Note: Since the system is utilized in an MPC, there will be a distinction between time instances in the controllable prediction horizon (*prediction*) and time instances before execution of the current MPC step (*past*).

When the MPC-internal model simulation reaches the end of the prediction horizon, total fatigue can be computed solely based on the terminal states. Thus, the fatigue penalization can be "Matched" to a standard terminal cost which solves Question 3.

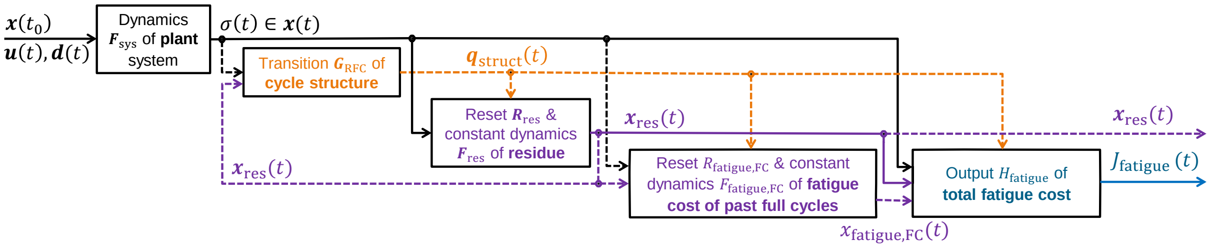

$\boldsymbol{x}(t_0)$
$\boldsymbol{u}(t), \boldsymbol{d}(t)$
Dynamics $\boldsymbol{F}_{\text{sys}}$ of plant system
$\sigma(t) \in \boldsymbol{x}(t)$
Transition $\boldsymbol{G}_{\text{RFC}}$ of cycle structure
$\boldsymbol{q}_{\text{struct}}(t)$
Reset $\boldsymbol{R}_{\text{res}}$ & constant dynamics $\boldsymbol{F}_{\text{res}}$ of residue
$\boldsymbol{x}_{\text{res}}(t)$
$\boldsymbol{x}_{\text{res}}(t)$
$\boldsymbol{x}_{\text{res}}(t)$
Reset $R_{\text{fatigue,FC}}$ & constant dynamics $F_{\text{fatigue,FC}}$ of fatigue cost of past full cycles
$x_{\text{fatigue,FC}}(t)$
Output H_{fatigue} of total fatigue cost
$J_{\text{fatigue}}(t)$

Figure 2.11: Structure of the hybrid dynamical system for plant dynamics and fatigue cost evaluation. Black = value-continuous dynamics, Orange = transition map, Purple = reset map, Blue = output function. Dashed lines = paths which are neglected at fatigue cost gradient calculation.

Originally, computation of fatigue cost gradients from this hybrid dynamical system is not straightforward due to its discontinuous subsystems and its complex structure. However, the task can be eased: the discontinuities are avoided by the principle "Switch seldom" of Section 2.2.4; the complexity of the gradient computation is further reduced by "Neglecting" subsystems which are independent from the control variables. Thus, gradients can be computed with reasonable effort, and Question 4 is solved.

Formulation of fatigue dynamics

■ **"Divide":** In order to represent Alg. 2 as a hybrid dynamical system, it is divided into suitable actions which are: the simulation of plant dynamics, extrema & cycle identification, update of residue, update of fatigue cost of past full cycles, and output of total accumulated fatigue cost. Consequently, these actions are represented by subsystems of the hybrid dynamical system, as visualized in Fig. 2.11. Interestingly, all types of mappings of hybrid dynamical systems are utilized; namely several continuous differential equations, a transition map, three reset maps, and one output function. Additionally, there are value-discrete and value-continuous state sets of variable dimensions. In the following, the main concepts are presented, while the mathematical details can be found in **Paper 5**.

- **Plant:** The time-continuous evolution of value-continuous plant states $\boldsymbol{x}(t)$ is obtained by the set $\boldsymbol{F}_{\text{sys}}$ of differential equations from (2.4b).
- **Cycle identification:** According to Alg. 2, cycle identification by the Rainflow algorithm is based on a stress string which is a concatenation of the residual stress samples and the new stress sample. This string is input to the transition map $\boldsymbol{G}_{\text{RFC}}$ which instantaneously updates the value-discrete structural states $\boldsymbol{q}_{\text{struct}}$. The structural states comprise cycle weights w_c, and sample indices of cycle maxima $k_{\max,c}$ and minima $k_{\min,c}$.
- **Residue:** As per the definition in Section 2.1.3, the residue emerges at τ which lies before the current evaluation time $\tau < t$. Particularly in the present case, this time frame stretches over the uncontrollable past $\tau \leq t_0$ and the controllable prediction $t_0 < \tau < t$. Thus, also the respective parts of the residue are either uncontrollable or controllable, and should be distinguished. Consequently, the value-continuous residue states

$$\boldsymbol{x}_{\text{res}} = \left(\{\boldsymbol{x}_{\text{res,past}}, \boldsymbol{x}_{\text{res,pred}}\}\right) = \begin{pmatrix} (\sigma_{\text{res,past},1}, \sigma_{\text{res,past},2}, \ldots, \sigma_{\text{res,past},N_{\text{past}}}) \\ (\sigma_{\text{res,pred},1}, \sigma_{\text{res,pred},2}, \ldots, \sigma_{\text{res,pred},N_{\text{pred}}}) \end{pmatrix}^T \tag{2.33}$$

  comprise of these two parts and are updated by the reset map

$$\boldsymbol{x}_{\text{res}}(t) = \boldsymbol{R}_{\text{res}}\left(\boldsymbol{q}_{\text{struct}}(t^-), \boldsymbol{x}_{\text{res}}(t^-), \sigma(t^-), t\right). \tag{2.34}$$

  The numbers of past N_{past} and predicted N_{pred} residue samples - and thus the dimensions - are variable. The reset map comprises two sub-maps $\boldsymbol{R}_{\text{res,past}}$ and $\boldsymbol{R}_{\text{res,pred}}$ for treatment of past-

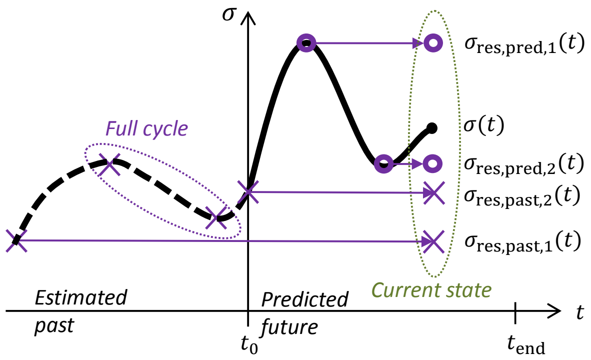

σ
Full cycle
$\sigma_{\text{res,pred,1}}(t)$
$\sigma(t)$
$\sigma_{\text{res,pred,2}}(t)$
$\sigma_{\text{res,past,2}}(t)$
$\sigma_{\text{res,past,1}}(t)$
Current state
Estimated past
Predicted future
t
t_0
t_{end}

Figure 2.12: Generation of past-residue $\boldsymbol{x}_{\text{res,past}}$ and prediction-residue $\boldsymbol{x}_{\text{res,pred}}$ from estimated past and predicted future stress evolution. Both past extrema which form a full cycle are excluded from the residue.

and prediction-residue, respectively. Furthermore, the reset map is accompanied by a constant continuous dynamics $\dot{\boldsymbol{x}}_{\text{res}}(t) = \boldsymbol{F}_{\text{res}} = \boldsymbol{0}$ in order to memorize the states in between of the reset events. Figure 2.12 shows the sampling of the past and predicted stress trajectory via reset maps to obtain the residue states.

At the beginning $t = t_0$ of the *predicted future*, the *past*-residue is updated by appending the new estimated stress sample $\tilde{\sigma}(t_0)$ and by forgetting samples related to full cycles, as shown in Fig. 2.12. Within the *predicted future* $t > t_0$, the past-residue is held constant.

In contrast, the *prediction*-residue is cleared at the beginning of the *predicted future* in order to forget the outdated information from the previous MPC step. During the progression of the *predicted future*, each emerging stress extremum is stored there.

- **Fatigue cost of past full cycles:** The fatigue cost $x_{\text{fatigue,FC}}(t)$ of full cycles which occurred in the *estimated past* $(t \leq t_0)$ is updated by the reset map

$$x_{\text{fatigue,FC}}(t) = R_{\text{fatigue,FC}}\left(\boldsymbol{q}_{\text{struct}}(t^-), \boldsymbol{x}_{\text{res}}(t^-), \sigma(t^-), x_{\text{fatigue,FC}}(t^-)\right) \tag{2.35}$$

  at the beginning of the *predicted future*. This reset map as well is accompanied by a constant continuous dynamics in order to memorize $x_{\text{fatigue,FC}}(t)$ within the *predicted future* $t > t_0$.

- **Fatigue cost:** Total accumulated fatigue cost $J_{\text{fatigue}}(t)$ is obtained by the value-continuous output function H_{fatigue}.

■ **"Match":** For the MPC, only the fatigue cost is relevant which is added within the prediction horizon. Added fatigue

$$J_{\text{fatigue,Hybrid}} = J_{\text{fatigue}}(t_{\text{end}}) - J_{\text{fatigue}}(t_0) \tag{2.36}$$

is computed by subtracting the initial fatigue cost from the terminal fatigue cost. This essentially matches fatigue cost to a standard terminal cost.

Derivation of fatigue cost gradient

In order to compute the gradient $\frac{\mathrm{d}J_{\text{fatigue,Hybrid}}(t)}{\mathrm{d}\bar{\boldsymbol{u}}}$ of fatigue cost w.r.t. the control variables, according to the chain rule, all mappings of the hybrid dynamical system have to be differentiated. For the plant subsystem, the gradient $\frac{\mathrm{d}\sigma(t)}{\mathrm{d}\bar{\boldsymbol{u}}}$ is obtained from Variational Differential Equations.

■ **"Switch seldom":** In contrast, the value-discrete transition map $\boldsymbol{G}_{\text{RFC}}$ cannot be differentiated. Therefore, the principle "Switch seldom" of Section 2.2.4 is applied which assumes that the update of control variables by the optimization algorithm only leads to mild variations in the state trajectories within one MPC step. Therefore, the cycle structure $\boldsymbol{q}_{\text{struct}}$ can be assumed as "frozen" and thus invariant w.r.t. the controls within one MPC step. Consequently, all paths are neglected for gradient calculation which are input or output to the transition map $\boldsymbol{G}_{\text{RFC}}$, as shown in Fig. 2.11. This also simplifies the gradient formulation for the downstream submodels of residue, fatigue cost of past full cycles, and total fatigue cost.

For instance, now the reset-map $\boldsymbol{R}_{\text{res,pred}}$ for the prediction-residue does not have to be differentiated w.r.t. the structural states $\boldsymbol{q}_{\text{struct}}$. Thus, the gradients $\frac{\mathrm{d}\boldsymbol{x}_{\text{res,pred}}(t)}{\mathrm{d}\bar{\boldsymbol{u}}}$ of the prediction-residue can simply be updated by a reset map $\boldsymbol{R}_{\text{res,pred},\bar{\text{u}}}$ of similar design to $\boldsymbol{R}_{\text{res,pred}}$ itself. The difference is that now gradient samples $\frac{\mathrm{d}\sigma(t)}{\mathrm{d}\bar{\boldsymbol{u}}}$ instead of stress samples are reset. A commonality is that these events of the gradient reset map $\boldsymbol{R}_{\text{res,pred},\bar{\text{u}}}$ are also triggered by the "frozen" structural states $\boldsymbol{q}_{\text{struct}}$ and thus coincide with the events in $\boldsymbol{R}_{\text{res,pred}}$.

■ **"Neglect":** Further simplification of gradient computation stems from the fact that states in the *estimated past* are independent from the future control variables. Consequently, no gradient is required for the reset map $\boldsymbol{R}_{\text{res,past}}$ of the past-residue, or for the reset map $R_{\text{fatigue,FC}}$ of the fatigue cost of past full cycles.

Thanks to these assumptions and neglections, the total fatigue cost gradient calculation boils down to the approximation

$$\frac{\mathrm{d}J_{\text{fatigue,Hybrid}}(t)}{\mathrm{d}\bar{\boldsymbol{u}}} = \frac{\partial H_{\text{fatigue}}(t)}{\partial\sigma}\frac{\mathrm{d}\sigma(t)}{\mathrm{d}\bar{\boldsymbol{u}}} + \sum_{l=1}^{N_{\text{pred}}}\frac{\partial H_{\text{fatigue}}(t)}{\partial\sigma_{\text{res,pred},l}}\frac{\mathrm{d}\sigma_{\text{res,pred},l}(t)}{\mathrm{d}\bar{\boldsymbol{u}}} \tag{2.37}$$

where the gradients of the output function H_{fatigue} w.r.t. the stress values are obtained by Automatic Differentiation.

2.3 Utilized system models

2.3.1 Wind model

For turbulent simulations with the high-fidelity wind turbine model in **Papers 6** and **7**, a stochastic, full-field wind model is employed. The model is executed offline, i.e. before the simulation, in the software `TurbSim` from the National Renewable Energy Laboratory (NREL) [69]. Here, for each time step, a two-dimensional vertical rectangular array of three-dimensional wind velocity vectors is generated, as shown in Fig. 2.13. The spectral properties of the wind fields are based on the IEC Kaimal model [70].

2.3.2 Lidar model

In **Paper 7**, the model of a pulsed Lidar with ten range gates and four beams is employed, as shown in Fig. 2.14. The model is executed in the Lidar simulator from `sowento GmbH` which generates Lidar wind estimates offline, and thus independently from the wind turbine simulation suite [71]. Considered physical effects are: limitation to line-of-sight wind speeds, spatial averaging via a Gaussian range weighting function, discrete scanning, and "unfrozen" wind evolution. Particularly the wind evolution can be parameterized by an exponential decay constant; here, a higher value results in higher variation of the wind during its convection towards the rotor. Finally, the spatially distributed measurements are converted to rotor-effective wind speed by wind field reconstruction [71, 72].

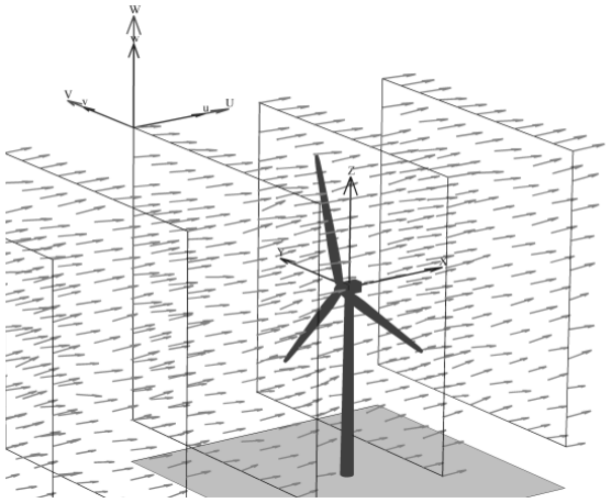


Figure 2.13: Turbulent wind field created by `TurbSim` [69]

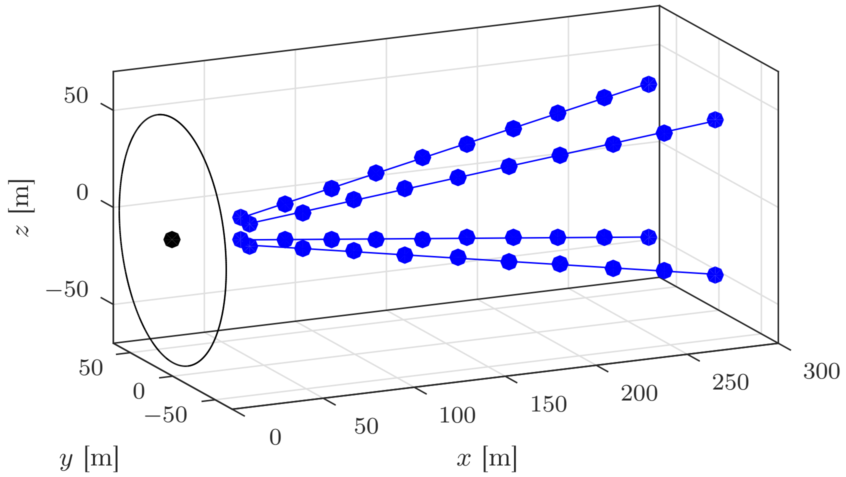


Figure 2.14: Left: rotor disc. Right: scanning points of the Lidar model in the upwind volume [27].

2.3.3 High-fidelity wind turbine model

The 3-bladed NREL 5MW onshore wind turbine model in the aero-elastic simulator `OpenFAST` is utilized in **Papers 6** and **7**.

The aerodynamics are represented by the classical Blade Element Momentum (BEM) theory. Here, the Momentum theory describes the interaction of the rotor disc with the stream tubes. The Blade Element theory describes the interaction of the flow with each blade section [72].

The structural elasto-dynamics are represented by the most significant mode shapes. The amplitude of each mode shape is defined by a second-order differential equation. Finally, the amplitudes of the independent modes are superimposed.

The active structural degrees of freedom (DOF) are stated in Tab. 2.1. Each of these DOFs is represented by 2 states. On top of those, 2 states for the collective blade pitch actuator and 1 state for the generator torque are added which results in a total of 33 states.

Table 2.1: Number of Degrees of Freedom for the NREL 5MW onshore turbine.

Component	Number of instances	DOFs	Total number of DOFs
Blades	3	2 flap-wise, 1 edge-wise	9
Tower	1	2 fore-aft, 2 side-side	4
Rotor	1	1 rotation	1
Drive-train	1	1 torsion	1
Total	-	-	**15**

2.3.4 Reduced wind turbine model

In all wind turbine MPCs of the present work, a standard reduced wind turbine model is utilized [73,74]. Its six states

$$\boldsymbol{x}(t) = \left(\omega_{\mathrm{r}}(t), d_{\mathrm{T}}(t), \dot{d}_{\mathrm{T}}(t), \beta_{\mathrm{b}}(t), \dot{\beta}_{\mathrm{b}}(t), T_{\mathrm{g}}(t)\right)^T \tag{2.38}$$

comprise the rotational speed of the rotor $\omega_{\mathrm{r}}(t)$, the fore-aft deflection of the tower tip $d_{\mathrm{T}}(t)$ and its time derivative, the collective pitch angle of the blades $\beta_{\mathrm{b}}(t)$ and its time derivative, and the generator torque $T_{\mathrm{g}}(t)$. Figure 2.15 provides an overview of the system dynamics.

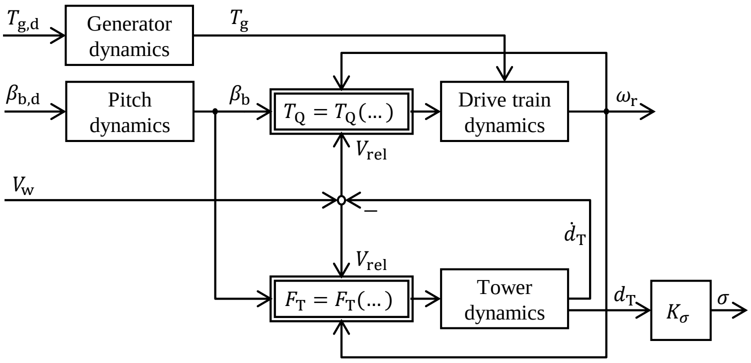


Figure 2.15: Coupled dynamics of wind turbine drive train and tower.

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 $d_{\mathrm{T}}(t)$, resulting in $\sigma(t) = K_\sigma d_{\mathrm{T}}(t)$. The control variables of the model correspond to the demands $\boldsymbol{u}(t) = \left(\beta_{\mathrm{b,d}}(t), T_{\mathrm{g,d}}(t)\right)^T$ for the subordinate actuator controllers. Here, $\beta_{\mathrm{b,d}}(t)$ is the demanded blade pitch angle and $T_{\mathrm{g,d}}(t)$ is the demanded generator torque. Excitement and disturbance $d(t) = V_{\mathrm{w}}(t)$ of the system stem from the ambient wind speed $V_{\mathrm{w}}(t)$, which is predicted for example by a Lidar scanner.

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

$$T_{\mathrm{Q}} = T_{\mathrm{Q}}(\omega_{\mathrm{r}}(t), V_{\mathrm{rel}}(t), \beta_{\mathrm{b}}(t)) \tag{2.39}$$

and the aerodynamic thrust force

$$F_{\mathrm{T}} = F_{\mathrm{T}}(\omega_{\mathrm{r}}(t), V_{\mathrm{rel}}(t), \beta_{\mathrm{b}}(t)). \tag{2.40}$$

These loads depend on the rotor-relative wind speed

$$V_{\mathrm{rel}}(t) = V_{\mathrm{w}}(t) - \dot{d}_{\mathrm{T}}(t) \tag{2.41}$$

where the external rotor-effective ambient wind speed $V_{\mathrm{w}}(t)$ is combined with the tower tip speed $\dot{d}_{\mathrm{T}}(t)$.

2.3.5 Li-ion battery model

In **Papers 8** and **9**, the plant is represented by a grey-box Li-ion battery model, which involves first-principle models and interpolated lookup-tables. For the MPC, the lookup-tables are substituted by continuous data-fitted static maps, which introduces a minor model mismatch. The Li-ion battery model comprises several sub-models, capturing the electrical, thermal and aging behavior as shown in Fig. 2.16. These sub-models will be presented in the following.

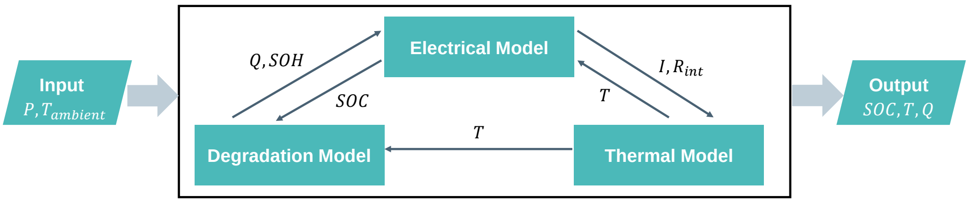


Figure 2.16: Process flow of the Battery model

Electrical model

A simple electrical equivalent circuit is chosen [75]. Important variables are charge/discharge power P, Open Circuit Voltage U_{ocv}, internal resistance R_{int}, current I and terminal voltage U_{terminal}. The states considered are State of Charge SOC, State of Health SOH and cell temperature T. State of Charge $SOC = \frac{Q_{\text{cont}}}{Q}$ $[-]$ is defined as the ratio of instantaneous charge *content* Q_{cont} with respect to the instantaneous charge *capacity* Q of the battery. Thus, the rate of SOC is directly proportional to the current, and can be expressed as

$$\frac{\mathrm{d}SOC}{\mathrm{d}t} = -\frac{I}{Q} \quad \left[\frac{1}{\mathrm{s}}\right]. \tag{2.42}$$

By convention, current is considered positive for the discharging case and negative for the charging case.

State of Health $SOH = \frac{Q}{Q_{\text{nominal}}}$ $[-]$ is defined as the ratio of instantaneous charge capacity Q with respect to the nominal charge capacity of the battery Q_{nominal} at the time of production. Charge capacity and thereby State of Health are decreased by aging, as it will be detailed later in this section.

Thermal model

The thermal model captures the dynamic behavior of cell temperature T (in [K]) due to internal heating and external cooling. The most commonly used thermal model

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{1}{C_{\mathrm{H}}}\left(I^2 R_{\text{int}} - C_{\mathrm{R}}(T - T_{\text{ambient}})\right) \quad \left[\frac{\mathrm{K}}{\mathrm{s}}\right] \tag{2.43}$$

is based on a lumped heat capacitance with the parameters Heat Capacity C_{H} (in [J/K]) and Cooling Rate C_{R} (in [J/(Ks)]). Ambient temperature T_{ambient} is an external input [76].

Degradation model

The degradation model captures the loss in charge capacity and the increase of internal resistance over time and usage.

Charge capacity: Typically, the charge capacity decreases over the lifetime of the battery. By convention, the amount of loss of charge capacity is assigned to two quantities: calendric capacity loss Q_{cal} which occurs intrinsically at any time, and cyclic capacity loss Q_{cyc} which occurs during usage of the battery. Consequently, the instantaneous charge capacity

$$Q = Q_{\text{nominal}} - (Q_{\text{cal}} + Q_{\text{cyc}})$$

can be expressed by the aforementioned quantities. The battery is considered to be at the end of its service life when its capacity Q has decreased to a predefined end-of-life capacity.

Calendric Aging: Calendric aging refers to the permanent loss in battery capacity, independently of usage. The corresponding rate of calendric capacity loss

$$\frac{\mathrm{d}Q_{\text{cal}}}{\mathrm{d}t}(t, SOC, T) = A_{\text{cal}} f_{Q\text{cal}}(t) f_{Q\text{cal}}(SOC) f_{Q\text{cal}}(T) \qquad \left[\frac{\text{As}}{\text{s}}\right] \tag{2.44}$$

depends on time since production t, and on the additional stress factors of State of Charge SOC and cell temperature T.

The time-variance is modeled by

$$f_{Q\text{cal}}(t) = \frac{1}{2\sqrt{t}} \approx \frac{1}{2\sqrt{t + t_{\text{shift}}}} \qquad \left[\frac{1}{\sqrt{s}}\right] \tag{2.45}$$

where the shift time t_{shift} can be used to account for the previous operation time duration of an already aged battery. The constant A_{cal} is derived from experimental data.

The temperature effect is represented by the Arrhenius law

$$f_{Q\text{cal}}(T) = A_{\text{arrhenius}} \exp\left(\frac{-E_{\text{activation}}}{R_{\text{gas}} T}\right) \qquad [-] \tag{2.46}$$

where $A_{\text{arrhenius}}$ is a model parameter obtained by experiments, $E_{\text{activation}}$ is the activation energy of the cell, and R_{gas} is the Universal gas constant. The State of Charge effect

$$f_{Q\text{cal}}(SOC) = K_1 + K_2 SOC \qquad [-] \tag{2.47}$$

is represented by a linear fit of experimental data via the parameters K_1 and K_2.

Cyclic aging: Cyclic aging refers to the loss of battery capacity during dynamic operation by repeated charging and discharging. The major stress factor for cyclic aging is the absolute range of magnitude of the individual SOC cycles, also referenced as "Depth of Discharge" DOD [77,78]. The most accepted algorithm for cycle identification from SOC trajectories is the well-known *Rainflow* algorithm [28]. Thus, the dynamics of cyclic aging can be represented by the one-step value-continuous Algorithm 2.

CHAPTER 3

Formulations

3.1 Paper 1: Formulation of Nonlinear MPC comprising data-driven submodels

3.1.1 Summary

An MPC implementation is presented for hybrid system models which comprise physics-based differential equations and data-driven static nonlinearities. For the present example of a wind turbine, these nonlinearities represent the aerodynamic torque and thrust of the rotor, and are required for the accurate prediction of fatigue. This MPC implementation will be used in **Papers 2**, **3** and **5**.

First, a symbolic math implementation of the wind turbine system dynamics is presented. Lookup tables of the nonlinearities are obtained from high-fidelity simulations. After training Feedforward Neural Networks (NN) with this numerical lookup data, a symbolic version of them can be integrated into the above mentioned symbolic differential equations.

Second, a symbolic math implementation of the state and cost gradient dynamics for the MPC is presented. Instead of "brute force" differentiating the entire hybrid model w.r.t. the control variables, a "divide and conquer" approach is pursued where hand-coded symbolic gradients of the NN are inserted after differentiating the remaining simpler differential equations. This "divide and conquer" approach beneficially reduces computational effort during MPC setup and online operation.

For validation at different wind speeds, the stationary operating points of this MPC in the loop with the same simplified model are compared to the stationary operating points of an LQR in the loop with the high-fidelity model. The results show very good agreement for wind speeds which have been part of the NN training set. Outside of this set, the results differ significantly which indicates low extrapolation capabilities of the NNs.

3.1.2 Contribution

The author of this dissertation extended the implementation, formalized the approach, performed the simulations, and wrote the paper. DO developed the approach and the initial implementation, and supervised the work. All authors provided important input to this research work through discussions, feedback and by improving the manuscript.

3.1.3 Reference

This work has been peer-reviewed and published at the *14th IEEE CASE* conference [42]. Subsequently, the work has been reprinted without modifications in the *atp magazin* journal, as included in this dissertation: S. Loew and D. Obradovic, “Real-time nonlinear model predictive control: Predictive control for mechatronic systems using a hybrid model,” *atp magazin*, no. 08, pp. 46–52, 2018.
Note: Due to a typesetting error, the numbers of Figure 1 and 2 have been exchanged in the respective figure captions.

Real-time Nonlinear Model Predictive Control

Predictive control for mechatronic systems using a hybrid model

Nonlinear Model Predictive Control (NMPC) is a control method for the implementation of advanced controller behavior. The present work shows the symbolic mathematical implementation of a mechatronic system model containing aerodynamic nonlinearities modeled by Feedforward Neural Networks. Gradients for the optimization are obtained efficiently by exploiting the feedforward property of the neural networks and symbolic computation. Current research on the implementation of damage metrics into the cost function is outlined. In order to achieve real-time capability, *Real-time Iteration* is used. The research is part of the project "MISTRALWIND – Monitoring and Inspection of Structures At Large Wind Turbines", funded by the German Federal Ministry of Economic Affairs and Energy.

KEYWORDS Model Predictive Control / NMPC / Real-time Iteration / Neural Networks / Hybrid model / Wind turbine

Echtzeit-Implementierung von NMPC –
Prädiktive Regelung für mechatronische Systeme unter Verwendung eines hybriden Modells

Nichtlineare Modellprädiktive Regelung (NMPC) ist eine aufstrebende Methode zur Umsetzung von komplexem Reglerverhalten. Die vorliegende Arbeit zeigt die symbolische Modellierung eines mechatronischen Systems unter Einbeziehung von aerodynamischen Nichtlinearitäten durch Neuronale Netze. Durch die Feedforward-Eigenschaft der Neuronalen Netze sowie Computeralgebra können die Gradienten für die Optimierung effizient berechnet werden. Aktuelle Forschungstätigkeit zur Einbindung von Materialermüdungsmetrik in die Kostenfunktion der Regelung wird kurz dargestellt. Um einen echtzeitfähigen Regler zu erreichen, wird die Methode Real-time Iteration verwendet.

SCHLAGWÖRTER Modellprädiktive Regelung / NMPC / Real-time Iteration / Neuronale Netze / Hybrides Modell / Windturbine

PEER Review 14.06.2018

STEFAN LOEW, DRAGAN OBRADOVIC, Siemens

The *MISTRALWIND* project aims to develop new monitoring, inspection and control methods for structures of large wind turbines. One of the work packages of Siemens is the development of a controller which reduces damage to structural parts and adapts to the remaining useful lifetime. Taking into account future wind predictions using lidar [1], a Nonlinear Model Predictive Controller (NMPC) is developed. This paper starts with an outline of the dynamic system. Afterwards, modeling of the aerodynamic nonlinearities by Feedforward Neural Networks is described. Furthermore, the applied strategies for the symbolic setup of the NMPC problem and the solution method using *Real-Time Iteration* are described.

1. WIND TURBINE SYTEM

A. States, control variables and optimization variables

Generally in MPC, future plant behavior is predicted by a simplified model of the plant. The states of the wind turbine are reduced to the state vector

$$\boldsymbol{x}(t) = \begin{pmatrix} \omega_r(t) \\ d_T(t) \\ \dot{d}_T(t) \\ \beta_b(t) \\ \dot{\beta}_b(t) \\ T_g(t) \end{pmatrix} \tag{1}$$

with 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 $T_g(t)$.

The control variables for subordinate controllers of the corresponding actuators are

$$\mathbf{u}(t) = \begin{pmatrix} \beta_{b,d}(t) \\ T_{g,d}(t) \end{pmatrix} \tag{2}$$

where $\beta_{b,d}(t)$ is the desired pitch angle of blades and $T_{g,d}(t)$ is the desired generator torque. The control variables are defined as piecewise-constant time functions

$$\bar{u}_{1,j} = \bar{\beta}_{b,d,j} \tag{3}$$

and

$$\bar{u}_{2,j} = \bar{T}_{g,d,j}\,. \tag{4}$$

For the MPC this leads to the optimization variables vector

$$\bar{\mathbf{u}} = \begin{pmatrix} \bar{\mathbf{u}}_1 \\ \bar{\mathbf{u}}_2 \end{pmatrix} \tag{5}$$

B. Analysis of system dynamics

The closed loop systems of the actuators are modeled as linear second and first order systems, respectively. Nonlinearity in the system model stems from the aerodynamic torque coefficient

$$C_Q = C_Q(\omega_r(t), V_{rel}(t), \beta_b(t)) \tag{6}$$

and the aerodynamic thrust coefficient

$$C_T = C_T\big(\omega_r(t), V_{rel}(t), \beta_b(t)\big). \tag{7}$$

These coefficients depend on the relative wind speed

$$V_{rel}(t) = V_w(t) - \dot{d}_T(t) \tag{8}$$

where the external variable rotor averaged ambient velocity of wind $V_w(t)$ is superimposed by velocity $\dot{d}_T(t)$ at the tower tip. These coefficients also couple the drive train dynamics

$$\dot{\omega}_r(t) = \frac{1}{J_{tot}}\Big(\frac{1}{2}\rho V^2{}_{rel} ARC_Q - T_g\Big) \tag{9}$$

and the tower fore-aft dynamics

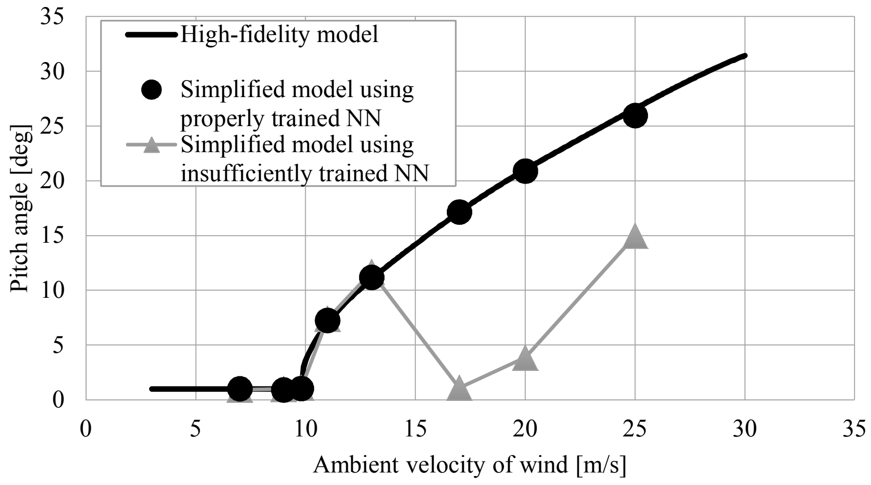


FIGURE 1: Comparison of stationary pitching behavior of baseline-controlled high-fidelity model and MPC-controlled simplified model

$$\ddot{d}_T = \frac{1}{m_T}\left(\frac{1}{2}\rho V^2{}_{rel} A C_T - c_T \dot{d}_T - k_T d_T\right) \qquad (10)$$

as shown in Fig. 1. The fixed parameters are lumped inertia of the drive train J_{tot}, air density ρ, rotor area A and rotor radius R. The mass m_T , damping coefficient d_T and stiffness k_T stem from a lumped model of tower fore-aft movement. Here, the aerodynamic coefficients C_Q and C_T are calculated at stationary operating points using the high-fidelity simulator *Cp-Lambda*. The entire simplified model

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\boldsymbol{x}(t), \mathbf{u}(t), V_w(t)) \qquad (11)$$

is a nonlinear hybrid model in which fundamental dynamic models are combined with empirical models [2].

2. MODELING OF LOOKUP DATA BY NEURAL NETWORKS

Lookup data can be included in control models, e.g. by high-order polynomials [3], cubic B-splines [4] or Feedforward Neural Networks. Since Feedforward Neural Networks (NN) globally approximate functions, usually fewer parameters are necessary. Although trained by numerical data, the inputs of Neural Networks can be defined symbolically. The outputs are then symbolic functions which can be inserted into other symbolic functions. In this work, the lookup data of the static aerodynamic nonlinearities (6) and (7) are modeled by Neural Networks and inserted into the symbolic representation of the simplified model (11). The Neural Network modeling is demonstrated for the aerodynamic torque. The aerodynamic thrust is treated analogously.

A. Inputs, Outputs and Training Data

A suitable input vector for the proposed Neural Network is

$$\mathbf{u}_{NN} = \begin{pmatrix} \omega_r \\ V_{rel} \\ \beta_b \end{pmatrix}. \qquad (12)$$

Rotational speed ω_r and relative wind velocity V_{rel} should not be merged to one input like tip-speed-ratio

$$\lambda = \frac{\omega_r R_r}{V_{rel}} \qquad (13)$$

where R_r is the radius of the wind turbine rotor. The reason is that the angle of attack and thus the aerodynamic coefficients of modern wind turbines can show significant direct dependency on wind velocity. This is due to bending and twisting of blades with increasing thrust force as passive thrust compensation.

Instead of the aerodynamic torque coefficient C_Q (6), the resulting torque T_Q is defined as the output

$$y_{NN} = T_Q = \frac{1}{2}\rho V^2{}_{rel} A R C_Q \tag{14}$$

for the Neural Network. This results in more compact equations if gradients of the system dynamics (11) have to be derived.

For the present case, Neural Networks exhibit very low extrapolation capabilities. In Fig. 2 the stationary behavior of the simplified model (11) controlled by an MPC is compared to a high-fidelity model in *Cp-Lambda* controlled by a Linear Quadratic Regulator (LQR). If the training data for the simplified model only covers an insufficient range $3\ m/s < V < 13\ m/s$ of ambient wind velocity, completely wrong stationary points are reached if higher wind velocities are applied, for example. If the training data set is extended to a proper range, the stationary operating points match the ones of the high-fidelity model very well.

The Neural Network only should be trained using data samples which are within the permitted operation range. For this, negative aerodynamic torques and forces are discarded, as well as aerodynamic torques exceeding a defined limit.

B. Extraction of gradients

Another advantage of Feedforward Neural Network modeling is the straightforward extraction of output gradients $\frac{\partial T_Q(\omega_r, V_{rel}, \beta_b)}{\partial \omega_r}$, $\frac{\partial T_Q(\omega_r, V_{rel}, \beta_b)}{\partial V_{rel}}$ and $\frac{\partial T_Q(\omega_r, V_{rel}, \beta_b)}{\partial \beta_b}$ w.r.t the inputs. Using the analytic derivative of the activation function, the gradients can be obtained by applying the chain rule.

3. NMPC METHOD AND REAL-TIME ITERATION

A. Method

The original problem is an Optimal Control Problem over an infinite horizon. Because this problem usually cannot be solved, an approximate solution is obtained by optimization over shorter, moving horizons. The length of the horizon is chosen taking into account the time constant of the turbine drive train and the tower.

The present optimization problem is defined by the cost function

$$J = J(\mathbf{x}, \mathbf{u}, V_w)\,, \tag{15}$$

box constraints on the optimization variables

$$\mathbf{u}_{min} \le \mathbf{u}(t) \le \mathbf{u}_{max}\,, \tag{16}$$

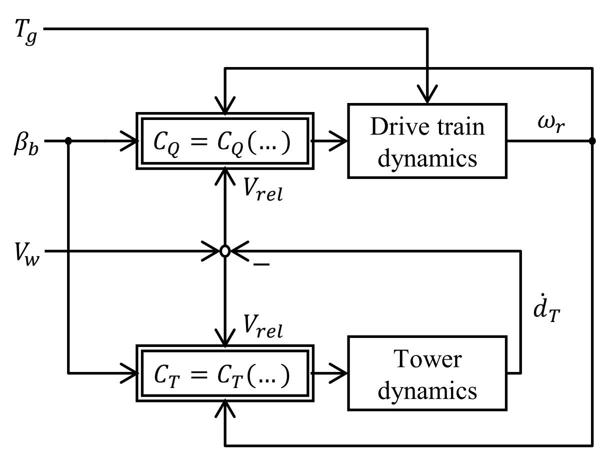


FIGURE 2: Coupled dynamics of wind turbine drive train and tower

box constraints on the rate of change of the optimization variables

$$\dot{\mathbf{u}}_{min} \le \dot{\mathbf{u}}(t) \le \dot{\mathbf{u}}_{max}\,, \tag{17}$$

and nonlinear inequality constraints on the system states

$$\mathrm{c}_{\mathrm{ineq}}\big(\mathbf{x}(t)\big) \le \begin{pmatrix} \mathbf{x}(t) - \mathbf{x}_{max} \\ -\mathbf{x}(t) + \mathrm{x}_{min} \end{pmatrix} \le 0\,. \tag{18}$$

If the future system states are defined as free variables (e.g. in the *Multiple-shooting approach*), additional constraints are needed in order to satisfy the dynamic behavior of the system [5]. This *Simultaneous approach* is advantageous for highly nonlinear systems and if parallelization is intended. Main disadvantages are the increased optimization effort and the chance of an unfeasible simulation. Alternatively, in the *Sequential approach* the system dynamics model is used explicitly in calculating the future states; in this case no additional free variables and constraints are needed [6]. The *Sequential approach* is applied in the present work.

B. Cost Function

In order to provide maximum freedom to the optimization algorithm in finding optimum solutions, the concept of *Economic NMPC* (ENMPC) is pursued. An interesting discussion on ENMPC can be found in [7]. For maximization of generated electrical energy [7]

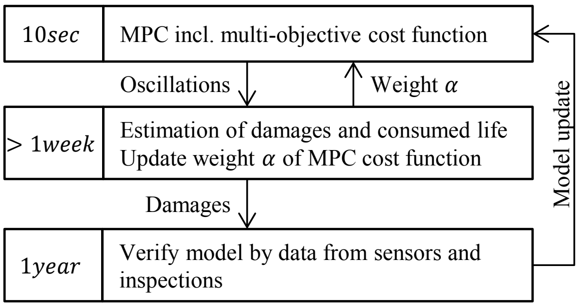

10sec
MPC incl. multi-objective cost function
Oscillations
Weight α
> 1week
Estimation of damages and consumed life
Update weight α of MPC cost function
Damages
1year
Verify model by data from sensors and inspections
Model update

FIGURE 3: Monitoring and control of structural damage at three different time stages

propose the indirect costing of power $P_A(t)$ extracted from the wind. Direct costing of electrical power $P_E(t)$ can lead to greedy extraction of rotor inertial energy, called the *turnpike* effect [7]. This behavior was also observed in the present work.

One objective in the *MISTRALWIND* project is the implementation of a tower damage metric into the cost function. This is challenging since damage is a long term effect. A three-stage approach is proposed as shown in Fig. 3 where the damage term in the cost function is adaptively weighted by α. Various cost terms $\prod_{damage}$ including tower tip velocity [7], thrust force, rainflow counting and frequency based methods are investigated currently. First simulations involving tower tip velocity show promising behavior.

In the present MPC implementation all terms of the cost function are of integral type in time. Thus the definition of the time derivative of the cost function

$$\dot{J}(t) = -P_A\big(\mathbf{x}(t), V_w(t)\big) + \alpha \prod\nolimits_{damage}(\mathbf{x}(t), \mathbf{u}(t)) \tag{19}$$

by integration over the MPC horizon $[t_0, t_{end}]$ yields the cost function

$$J = \int_{t_0}^{t_{end}} \dot{J}(t)\, dt \, .$$

C. Real-time iteration

Instead of solving one *Sequential Quadratic Programming* (SQP) problem for each new optimization horizon, *Real-time Iteration* (RTI) is applied, where only one *Quadratic Programming* (QP) problem is solved at each control step as shown in Fig. 4. RTI is based on the assumption of only mildly varying optimization problems throughout subsequent MPC-steps [8]. In the present wind turbine application this can be assumed to be fulfilled if the sampling time of the MPC is appropriately low, considering system dynamics and worst case wind variations. Each QP problem is formulated at the current operating point $(\mathbf{x}^*, \bar{\mathbf{u}}^*, V_w^*)$. The cost function is approximated quadratically as

$$\min_{\Delta\bar{\mathbf{u}}} \frac{1}{2} \Delta\bar{\mathbf{u}}^T \mathbf{H}^* \Delta\bar{\mathbf{u}} + (\mathbf{g}^*)^T \Delta\bar{\mathbf{u}} \, , \tag{21}$$

where $\mathbf{H}^*$ is the Hessian and $\mathbf{g}^*$ is the gradient of the cost function with respect to the optimization variables $\bar{\mathbf{u}}$. The Hessian $\mathbf{H}$ of the QP problem is updated at each control step in accordance with the *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) method ([6], [9]). Since the cost function (20) is of integral type, the gradients of the cost function with respect to all optimization variables have to be evaluated at $t=t_{end}$ of the MPC horizon. For a specific operating point this yields

$$\mathbf{g}^* = \frac{dJ^*(t_{end})}{d\bar{\mathbf{u}}^*} \, . \tag{22}$$

D. Cost and State Gradients with respect to optimization variables

Since the cost function is of integral type, the time derivative of the cost function $\dot{J}(t) = \frac{dJ}{dt}$ is already defined (19). Thus using Schwarz's theorem [10]

$$\frac{d}{dt}\left(\frac{dJ(t)}{d\bar{\mathbf{u}}}\right) = \frac{d}{d\bar{\mathbf{u}}}\left(\frac{dJ(t)}{dt}\right) = \frac{d\dot{J}(t)}{d\bar{\mathbf{u}}} \tag{23}$$

the gradient at the end of a horizon can be determined by numerical integration. Using the chain rule, this leads to $N_{Ju}=i{\cdot}j$ cost gradient differential equations

$$\frac{d\dot{J}(t)}{d\bar{u}_{i,j}} = \sum\nolimits_l \frac{\partial \dot{J}(t)}{\partial x_l(t)} \frac{dx_l(t)}{d\bar{u}_{i,j}} + \sum\nolimits_p \frac{\partial \dot{J}(t)}{\partial u_p(t)} \frac{du_p(t)}{d\bar{u}_{i,j}} \, . \tag{24}$$

In the preparation phase of the MPC problem, the functional derivatives $\frac{\partial \dot{J}(t)}{\partial x_l(t)}$ and $\frac{\partial \dot{J}(t)}{\partial u_p(t)}$ are determined efficiently by symbolic differentiation of the cost dynamics $\dot{J}(t)$ which will be described in Section 3-E. Considering piecewise-constant optimization variables $\bar{\mathbf{u}}$ gives the following determination of the gradient

$$\frac{du_p(t)}{d\bar{u}_{i,j}} = \begin{cases} 1 \text{ if } (j-1)T_{cntrl} \leq t \leq jT_{cntrl} \text{ and } i = p \\ \text{0 otherwise.} \end{cases} \tag{25}$$

Finally the cost gradient differential equations in (24) require the state gradients with respect to the optimization variables. Analogously to the cost gradient, by using Schwarz's theorem

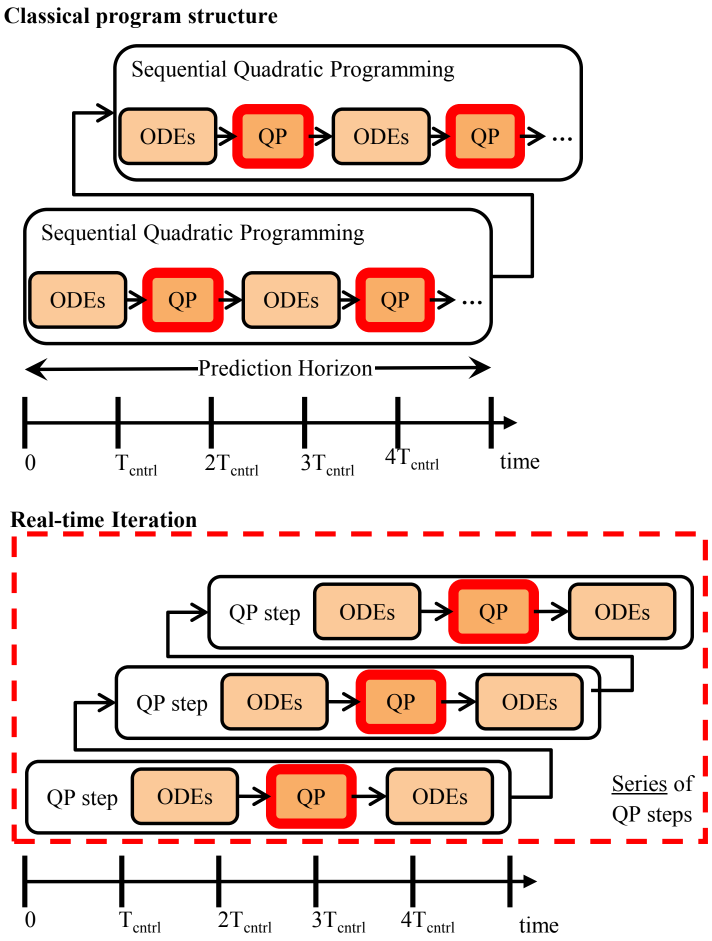


FIGURE 4: Comparison of classical program structure and the method of Realtime Iteration. The blocks "ODEs" symbolize numerical integration of the differential equation and the blocks "QP" symbolize the solution of a QP problem.

ACKNOWLEDGMENTS

We would like to thank the German Federal Ministry for Economic Affairs and Energy for funding the *MISTRALWIND* project. We would also like to thank the research partners at Wind Energy Institute of TUM and at the Department of Aerospace Engineering at Politecnico di Milano for their advice and their support with the *Cp-Lambda* simulator.

$$\frac{d}{dt}\left(\frac{d\mathbf{x}(t)}{d\bar{\mathbf{u}}}\right)=\frac{d}{d\bar{\mathbf{u}}}\left(\frac{d\mathbf{x}(t)}{dt}\right)=\frac{d\dot{\mathbf{x}}(t)}{d\bar{\mathbf{u}}} \qquad (26)$$

the state gradients can be determined by numerical integration. Using the chain rule and (11), this leads to $N_{xu}=k{\cdot}i{\cdot}j{\cdot}$ state gradient differential equations

$$\frac{d\dot{x}_k(t)}{d\bar{u}_{i,j}}=\sum\nolimits_l\frac{\partial F_k(t)}{\partial x_l(t)}\frac{dx_l(t)}{d\bar{u}_{i,j}}+\sum\nolimits_p\frac{\partial F_k(t)}{\partial u_p(t)}\frac{du_p(t)}{d\bar{u}_{i,j}}. \qquad (27)$$

The gradients $\frac{dx_l(t)}{d\bar{u}_{i,j}}$ are the states of these ODEs and introduce full coupling of the state gradient differential equations. The functional derivatives $\frac{\partial F_k(t)}{\partial x_l(t)}$ and $\frac{\partial F_k(t)}{\partial u_p(t)}$ are again determined by symbolic differentiation. The state gradients are additionally used to linearize the nonlinear inequality constraints (18) of the states for the QP.

E. Gradient dynamics by Symbolic Differentiation

Functional derivatives of cost (19) and states (11) ODEs with respect to states and control variables can be obtained by numerical differentiation, automatic differentiation and symbolic differentiation [11].

Symbolic differentiation of extensive nonlinear equations can lead to high computational effort at the setup of the MPC-problem. In the present case, the aerodynamic coefficients (6) and (7) introduce most of complexity. By initially defining the aerodynamic coefficients as functionals, only compact ODEs have to be differentiated symbolically. Afterwards the functionals and their gradients can be substituted by symbolic expressions explicitly provided by the Neural Networks (see Section 2).

REFERENCES

[1] Bottasso, C. L., Pizzinelli, P., Riboldi, C. E. D., & Tasca, L. (2014). LiDAR-enabled model predictive control of wind turbines with real-time capabilities. *Renewable Energy*, 71, 442-452..

[2] Henson, M. A. (1998). Nonlinear model predictive control: current status and future directions. *Computers & Chemical Engineering, 23*(2), 187-202.S. Gros, “An economic nmpc formulation for wind turbine control,” in 52nd IEEE Conference on Decision and Control, 2013, pp. 1001–1006.

[3] Gros, S., Quirynen, R., & Diehl, M. (2014, December). An improved real-time economic NMPC scheme for Wind Turbine control using spline-interpolated aerodynamic coefficients. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on* (pp. 935-940). IEEE.

[4] Findeisen, R., & Allgöwer, F. (2002). An introduction to nonlinear model predictive control. In 2002 *21st Benelux Meeting on Systems and Control on* (pp. 1–23).

[5] Altmannshofer, S. (2012). Fast suboptimal nonlinear model predictive control of an inverted pendulum. In IFAC Proceedings Volumes vol. 45, no. 17, 2012 *IFAC Nonlinear Model Predictive Control Conference International Federation of Automatic Control on* (pp. 442–447). IFAC.

[6] Gros, S., & Schild, A. (2017). Real-time economic nonlinear model predictive control for wind turbine control. *International Journal of Control, 90*(12), 2799-2812.

[7] Gros, S., Zanon, M., Quirynen, R., Bemporad, A., & Diehl, M. (2016). From linear to nonlinear MPC: bridging the gap via the real-time iteration. *International Journal of Control*, 1-19..

[8] Papageorgiou, M., Leibold, M., & Buss, M. (2015). *Optimierung: Statische, dynamische, stochastische Verfahren für die Anwendung*. Berlin, Heidelberg: Springer Vieweg.

[9] Hackbusch, W., & Zeidler, E. (Eds.). (2013). *Springer-Taschenbuch der Mathematik*. Springer Spektrum.

[10] Al Seyab, R. K. (2007). Nonlinear model predictive control using automatic differentiation. Abgerufen von: https://core.ac.uk/download/pdf/138164.pdf

AUTHORS

STEFAN LOEW (born 1992) is PhD student in the research group “Autonomous Systems & Control” at Siemens Corporate Technology and at the Wind Energy Institute of TUM in Munich. His research focuses on Economic Nonlinear Model Predictive Control of mechatronic systems. Specific topics are control of mechanical fatigue and control under uncertainty.

Siemens AG,
Otto-Hahn-Ring 6,
81739 Munich,
Phone +49 (0) 89 636 63 32 75,
e-Mail: Loew.Stefan@siemens.com

Dr. **DRAGAN OBRADOVIC** (born 1957) holds a position of Senior Research Scientist in the Corporate Technology division of Siemens AG. His current research focuses, among others, on control and optimization of wind energy systems and on autonomously guided vehicles in logistic applications.

Siemens AG,
Otto-Hahn-Ring 6,
81739 Munich,
Phone +49 (0) 173 999 28 91,
e-Mail: Dragan.Obradovic@siemens.com

3.2 Paper 2: Simplification of fatigue & concept of incorporation into the MPC algorithm

3.2.1 Summary

The idea for a completely novel formulation is raised which directly incorporates the standard fatigue estimation process in the cost function of an Economic Nonlinear Model Predictive Controller. The key feature is the substitution of the discontinuous Rainflow(-counting) algorithm by continuous expressions. This substitution is performed once per MPC step and can be fixed in the meantime as supported by practical observations. The formulation called "Direct Online Rainflow-counting" (DORFC) is presented briefly, implemented in the MPC of **Paper 1**, and compared to two indirect fatigue cost formulations.

The first indirect fatigue cost function is very common and corresponds to a stage cost of the squared stress rate. For the wind turbine tower example, stress rate at the tower bottom is highly correlated with the deflection rate of the tower tip, and thus is called "Tower-Tip-Velocity Penalization" (TTVP) here.

The second indirect fatigue cost function is based on the facts that in the standard fatigue estimation process stress cycle amplitudes are considered as key fatigue drivers, and that a constant stress trajectory results in zero fatigue. Thus, fatigue damage can be penalized approximately by a stage cost of squared stress differences from the stress mean over the prediction horizon. This formulation is called "Amplitude Penalization" (AP) here.

In turbulent simulations, the dynamic behavior of DORFC exhibits characteristic plateaus in the stress trajectory which stem from the direct objective of reducing the amplitudes of stress cycles. TTVP and AP exhibit fair economic results while the latter seems to be less sensitive to reference wind velocity. DORFC exhibits best profit due to higher total revenue while accepting higher damage of the turbine tower. Furthermore, in contrary to the indirect formulations, DORFC can be fully parameterized by physical and economical quantities which simplifies the commissioning.

3.2.2 Contribution

The author of this dissertation developed the core idea of Direct Online Rainflow Counting, performed the implementation, executed and analyzed the simulations, and wrote the paper. DO and CLB put the formulation in the right context, recommended simulation cases, and supervised the work. All authors provided important input to this research work through discussions, feedback and by improving the manuscript.

3.2.3 Reference

S. Loew, D. Obradovic, and C. L. Bottasso, “Direct online rainflow-counting and indirect fatigue penalization methods for model predictive control,” in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3371–3376. doi:10.23919/ECC.2019.8795911 © 2019 IEEE. Reprinted with permission.

2019 18th European Control Conference (ECC)
Napoli, Italy, June 25-28, 2019

Direct Online Rainflow-counting and Indirect Fatigue Penalization Methods for Model Predictive Control*

Stefan Loew[1], Dragan Obradovic[2], *Member, IEEE*, Carlo L. Bottasso[3]

***Abstract*— The standard material fatigue estimation process is implemented in an Economic Nonlinear Model Predictive Controller. The process comprises cycle identification, mean-stress effect, Wöhler curves and Linear Damage Accumulation. For cycle identification, this novel implementation uses a standard Rainflow-algorithm which is substituted periodically by linear functions based on its output. The implementation is compared to two straightforward methods for fatigue penalization using a wind turbine model and exhibits higher profit over the entire range of tested reference wind velocities.**

I. Introduction

Fatigue is damage of a material caused by cyclic application of mechanical stress. Failure occurs after a large number of e.g. $10^1 - 10^8$ stress cycles at rather low stress amplitudes. Stress levels may be smaller by dimensions than the ultimate tensile stress where already a single stress peak leads to failure of the considered machine part. Fatigue has large impact on the operating costs of devices and thus control of fatigue is used to increase the total economic profit.

Model Predictive Controllers (MPC) enable optimal control of many devices by using predictions of future system excitation ([1], [2]). Based on these input predictions, stress time-series at crucial spots in the device structure can be predicted. Rainflow-counting (RFC) is the standard method for decomposition of stress time-series for fatigue estimation. Until recently, RFC could not be implemented in MPC [3] and only could be used for post-processing of measured and simulated data. In the preliminary study [4], RFC is implemented in MPC by the method of *Direct Online Rainflow-counting* (DORFC).

In the present paper, DORFC is applied to a wind turbine model using turbulent wind of different reference wind velocities. DORFC is compared to two conventional methods which here are called *Tower-Tip-Velocity-Penalization* (TTVP) and *Amplitude Penalization* (AP). Previous applications of MPC to wind turbines can be found in [5], [6] and [7].

This paper is organized as follows. In Section II, the conventional fatigue estimation process is presented. In Section III, three online implementations of fatigue penalization and their gradients are described. In Section IV, the exemplary wind turbine system and setup of the MPC is presented. In Section V, the three implementations are compared w.r.t. tuning effort as well as economic and dynamic behavior. Finally, Section VI summarizes results and gives an outlook to future work.

II. Fatigue Estimation

A. Relevant Quantities

Fatigue impact of a given stress time-series is characterized primarily by the amplitude and number of contained stress cycles. Mean stress of cycles has to be considered, since it alters fatigue impact of a cycle of a specific stress amplitude. In current fatigue estimation procedures, frequencies and actual shapes of stress cycles are neglected.

B. Cycle Identification

Rainflow-counting (RFC) is considered as most sophisticated method for the identification of stress cycles c from general non-repetitive stress time-series. RFC is based on the work of Matsuishi & Endo [8] and standardized in ASTM-E1049 [9]. This standard as well is the basis for the implementation `rainflow()` in MATLAB [10]. The outputs for identified stress cycles of RFC can be converted to the quantities of Tab. I where the value of weight equals $w_c = 1$ if a full stress cycle is detected and equals $w_c = 0.5$ if a half cycle is detected.

TABLE I: Converted outputs of RFC for stress cycles c

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

C. Fatigue Cost Estimation

After the identification of stress cycles c, damage D of the stress time-series $\sigma(\mathbf{u})$ can be computed, like shown in Fig. 1 (left). For brevity of notation, only the dependency on the control variables $\mathbf{u}$ is stated.

*The depicted research is part of the project 'MISTRALWIND - Monitoring and Inspection of Structures At Large Wind Turbines' and funded by the German Federal Ministry of Economic Affairs and Energy.

[1] Stefan Loew is with Siemens Corporate Technology, Otto-Hahn-Ring 6, 81739 Munich, Germany and the Wind Energy Institute of Technical University of Munich, Boltzmannstrasse 15, 85748 Garching, Germany. `loew.stefan@siemens.com`

[2] Dr. Dragan Obradovic is with Siemens Corporate Technology, Otto-Hahn-Ring 6, 81739 Munich, Germany. `dragan.obradovic@siemens.com`

[3] Prof. Carlo L. Bottasso is head of the Wind Energy Institute of Technical University of Munich, Boltzmannstrasse 15, 85748 Garching, Germany. `carlo.bottasso@tum.de`

978-3-907144-00-8 ©2019 EUCA

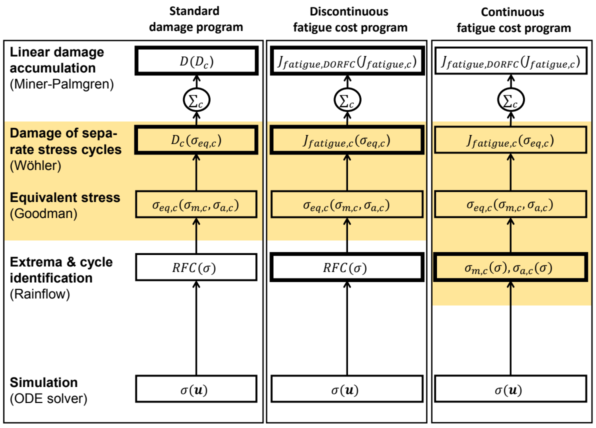

Standard damage program
Discontinuous fatigue cost program
Continuous fatigue cost program
Linear damage accumulation (Miner-Palmgren)
Damage of separate stress cycles (Wöhler)
Equivalent stress (Goodman)
Extrema & cycle identification (Rainflow)
Simulation (ODE solver)
$D(D_c)$
$J_{fatigue,DORFC}(J_{fatigue,c})$
$J_{fatigue,DORFC}(J_{fatigue,c})$
Σ_c
Σ_c
Σ_c
$D_c(\sigma_{eq,c})$
$J_{fatigue,c}(\sigma_{eq,c})$
$J_{fatigue,c}(\sigma_{eq,c})$
$\sigma_{eq,c}(\sigma_{m,c}, \sigma_{a,c})$
$\sigma_{eq,c}(\sigma_{m,c}, \sigma_{a,c})$
$\sigma_{eq,c}(\sigma_{m,c}, \sigma_{a,c})$
$RFC(\sigma)$
$RFC(\sigma)$
$\sigma_{m,c}(\sigma), \sigma_{a,c}(\sigma)$
$\sigma(\boldsymbol{u})$
$\sigma(\boldsymbol{u})$
$\sigma(\boldsymbol{u})$

Fig. 1: Original damage program (left). Substitution of damage D by fatigue cost $J_{fatigue}$ (middle). Substitution of $RFC(\sigma)$ by linear continuous functions yields the continuous fatigue cost program (right). Parallel calculation for separate cycles (colored) and joint calculation (elsewhere).

1) Equivalent stress: The effect of mean stress of cycles on fatigue is considered by a correction of amplitude stress to an equivalent stress via the *Goodman equation* [11]

$$\sigma_{eq,c} = \sigma_{a,c} \frac{R_m}{R_m - \sigma_{m,c}} \tag{1}$$

where R_m denotes ultimate tensile stress.

2) Damage of separate stress cycles: Wöhler curves $\sigma_{eq,c}(N_c)$ which can be obtained by material experiments provide the equivalent stress of a cycle for a specific number of cycles to failure. Fatigue damage of one cycle is obtained by the reciprocal

$$D_c = \frac{1}{N_c(\sigma_{eq,c})} \tag{2}$$

of the number N_c of cycles to failure which is the inverse of the Wöhler function.

3) Linear damage accumulation: Following the *Miner-Palmgren* rule [12], total fatigue damage is obtained by linear accumulation

$$D = \sum_c D_c \tag{3}$$

of damage of all detected cycles.

III. Fatigue Cost Functions for ENMPC

This chapter deals with the online implementation of fatigue metrics and their gradients in ENMPC. Two straightforward methods (TTVP, AP) and one novel method (DORFC) are presented.

A. Tower-Tip-Velocity Penalization (TTVP)

A common approach of implementing fatigue in MPC is a cost function term which quadratically penalizes stress rate. The present work follows [7], where this is implemented for a wind turbine tower by penalizing deflection rate $\dot{d}_T$ at the tower top. In the present work, this is implemented by averaged kinetic energy of tower oscillation which is obtained by

$$J_{fatigue,TTVP} = \int_{t_0}^{t_{end}} \frac{1}{2T_{horizon}} m_T \dot{d}_T^2 dt \tag{4}$$

using the lumped mass of the tower m_T and a normalization by the horizon length $T_{horizon} = t_{end} - t_0$.

ODEs for gradients $\frac{dJ_{fatigue,TTVP}}{d\bar{u}_{i,j}}$ of this integral fatigue cost function w.r.t. the optimization variables $\bar{u}_{i,j}$ are set-up symbolically using Schwarz's theorem [13] and solved by numerical integration (see [14]).

B. Amplitude Penalization (AP)

In the standard fatigue estimation process, stress amplitudes of separate cycles are considered as key fatigue-drivers (see Section II-A). A constant stress trajectory results in zero fatigue. Thus, fatigue damage can be penalized approximatively by penalizing stress amplitudes

$$\sigma_a(k) = \bar{\sigma}(k) - \sigma_m \tag{5}$$

w.r.t. a stress mean

$$\sigma_m = \frac{\sum_{k=1}^{N_u} \bar{\sigma}(k)}{N_u} \tag{6}$$

over the entire MPC horizon. The variable N_u denotes the number of control intervals within the MPC horizon and $\bar{\sigma}(k)$ denotes stress samples at the control intervals.

The fatigue cost function

$$J_{fatigue,AP} = \sum_{k=1}^{N_u} \sigma_a(k)^2 \tag{7}$$

is a sum of squares of stress amplitudes. Gradients $\frac{dJ_{fatigue,AP}}{d\bar{u}_{i,j}}$ of total fatigue cost are obtained by applying the chain rule.

C. Direct Online Rainflow-counting (DORFC)

The method DORFC is based on the standard fatigue estimation process of Section II applied to the entire MPC horizon. In order to enable monetary cost terms, the damage program of Fig. 1 (left) is transformed to the monetary fatigue cost program in Fig. 1 (middle). This requires a monetary analytical expression for the cost $J_{fatigue,c}$ of separate stress cycles c w.r.t. equivalent stress $\sigma_{eq,c}$. In [4] physical and mathematical requirements for this expression are defined. One analytical expression which fulfills these requirements is a polynomial

$$J_{fatigue,c}(\sigma_{eq,c}) = w_c \left(a_2 \ \sigma_{eq,c}^2 + a_4 \ \sigma_{eq,c}^4\right) \tag{8}$$

of even exponents. The coefficients a_2 and a_4 can be based on fatigue damage relations (2) and additional monetary damage cost models. Analogously to fatigue damage (3), fatigue cost is accumulated linearly by

$$J_{fatigue,DORFC} = \sum_c J_{fatigue,c}. \tag{9}$$

In [4] a direct implementation of the Rainflow-algorithm in MPC is introduced which overcomes obstacles for gradient calculation. Key feature is the substitution of the discontinuous RFC algorithm $RFC(\sigma)$ by continuous expressions, like shown in Fig. 1.

The implementation is based on the practical observation with the present wind turbine model that the structure of identified cycles does not vary strongly for most subsequent MPC steps. This structure includes number of cycles $\max(c)$, cycle weights w_c and cycle samples $(k_{max,c}, k_{min,c})$. Thus, this structure can be assumed to be fixed during one optimization step in MPC. Consequently, according to Table I, only mean $\sigma_{m,c}$ and amplitude stresses $\sigma_{a,c}$ remain variable. Continuous expressions of mean stress

$$\sigma_{m,c} = \frac{\bar{\sigma}(k_{max,c}) + \bar{\sigma}(k_{min,c})}{2} \tag{10}$$

and amplitude stress

$$\sigma_{a,c} = \frac{\bar{\sigma}(k_{max,c}) - \bar{\sigma}(k_{min,c})}{2} \tag{11}$$

are obtained using their related maximum and minimum stress samples. The positions $k_{max,c}$ and $k_{min,c}$ of these stress samples within the MPC horizon are obtained from the RFC algorithm.

Expressions (10) and (11) provide a continuous substitute to the Rainflow-algorithm $RFC(\sigma)$ in Fig. 1 (middle) which yields the continuous fatigue cost program in Fig. 1 (right). Gradients $\frac{dJ_{fatigue,DORFC}}{d\bar{u}_{i,j}}$ of total fatigue cost w.r.t. the optimization variables $\bar{u}_{i,j}$ then can be obtained by applying the chain rule.

IV. Exemplary System and Controller

The exemplary three-bladed variable-speed $3.6MW$ wind turbine model is assumed to be equipped with wind prediction e.g. by a LiDAR, like shown in Fig. 2. Perfect foresight of the rotor-equivalent velocity V_w of wind is assumed in this work. Wind induces thrust force on the rotor which excites fore-aft oscillations in the tower. These oscillations result in severe cyclic stress and thus in fatigue at the tower root. This wind turbine model is used as internal controller model and as simulation model.

A. System Quantities

The wind turbine is modeled as an explicit nonlinear system

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}(t), \mathbf{u}(t), d(t)). \tag{12}$$

The state vector

$$\mathbf{x}(t) = \begin{pmatrix} \omega_r(t) \\ d_T(t) \\ \dot{d}_T(t) \\ \beta_b(t) \\ \dot{\beta}_b(t) \\ T_g(t) \end{pmatrix} \tag{13}$$

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 $T_g(t)$. Stress at tower root is modeled by a linear dependency

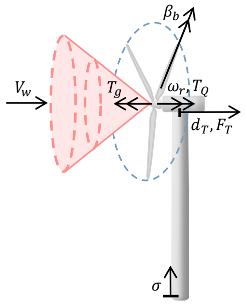


Fig. 2: Sketch of the LiDAR-equipped wind turbine model with rotor-equivalent velocity V_w of wind, generator torque T_g, rotational speed of the rotor ω_r, aerodynamic torque T_Q, displacement of tower tip d_T, aerodynamic thrust force F_T, collective blade pitch angle β_b and tensile stress at tower root σ.

$$\sigma(t) = K_\sigma d_T(t) \tag{14}$$

to tower tip deflection $d_T(t)$ and thus is an output of the system. The control variables for subordinate controllers of the corresponding actuators are

$$\mathbf{u}(t) = \begin{pmatrix} \beta_{b,d}(t) \\ T_{g,d}(t) \end{pmatrix} \tag{15}$$

where $\beta_{b,d}(t)$ is the desired pitch angle of blades and $T_{g,d}(t)$ is the desired generator torque. Excitement and disturbance

$$d(t) = V_w(t) \tag{16}$$

of the system stem from velocity $V_w(t)$ of wind. Details on the system dynamics and idealizations are given in [14].

B. Setup of Control Problem

The original problem of MPC is an Optimal Control Problem over an infinite horizon. Because this problem usually cannot be solved, an approximative solution is obtained by optimization over shorter, moving horizons $[t_0, t_{end}]$. In the present case, the time-continuous control variables $u_i(t)$ become the piecewise-constant optimization variables $\bar{u}_{i,j}$, where the type of control variable is determined by i and the control interval by j.

In order to provide maximum freedom to the optimization algorithm in finding optimal solutions, the concept of *Economic NMPC* (ENMPC) is pursued [7]. An economic cost function

$$J = -J_{revenue}(\mathbf{x}) + \alpha_{damage} J_{fatigue}(\mathbf{x}) \tag{17}$$

is used to trade-off maximization of revenue $J_{revenue}$ and minimization of fatigue cost $J_{fatigue}$ by the controller. A weighting-factor α_{damage} can be used to influence this trade-off. The optimization problem is subject to

$$\mathbf{u}_{min} \leq \bar{\mathbf{u}} \leq \mathbf{u}_{max}, \tag{18}$$

$$\dot{\mathbf{u}}_{min} \leq \dot{\bar{\mathbf{u}}} \leq \dot{\mathbf{u}}_{max} \tag{19}$$

$$\text{and } \mathbf{c_{ineq}}(\bar{\mathbf{x}}) = \begin{pmatrix} \bar{\mathbf{x}} - \mathbf{x}_{max} \\ -\bar{\mathbf{x}} + \mathbf{x}_{min} \end{pmatrix} \leq \mathbf{0}, \tag{20}$$

where the system states $\bar{\mathbf{x}}$ are sampled at the control intervals.

This nonlinear program is solved by the *Sequential approach* where the system $\left[\dot{\mathbf{x}}, \dot{J}_{integral}, \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{u}}, \frac{\partial \dot{J}_{integral}}{\partial \mathbf{u}}\right]^T$ of ordinary differential equations (ODEs) is solved in a Single-shooting manner [1]. The integral cost function $J_{integral}$ comprises both cost terms for TTVP and only the revenue term for AP and DORFC (see Section IV-C). The solutions of the ODEs are used for preparation of a Quadratic Programming (QP) problem and for Hessian update via the *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) method ([15], [16]). Following the method of *Real-time Iteration* [17], only one QP is solved per MPC step. More details on the present implementation are described in [14].

C. Terms of Cost Function

For TTVP and AP, following [7], revenue is represented by extracted energy from the wind which is obtained by

$$J_{revenue,TTVP/AP} = \int_{t_0}^{t_{end}} P_A(t)dt \tag{21}$$

using extracted aerodynamic power P_A. For TTVP, fatigue cost is represented by (4) and a pareto-optimal tuning weight of $\alpha_{damage,TTVP} = 4200$ has been determined by simulations. For AP, fatigue cost is represented by (7) and the corresponding pareto-optimal tuning weight is $\alpha_{damage,AP} = 2520$. Note that using this weight, stresses $\bar{\sigma}$ need to be in the unit $[MPa]$.

For DORFC, monetary cost terms are applied. Thus, revenue within the horizon is obtained by

$$J_{revenue,DORFC} = \int_{t_0}^{t_{end}} P_A(t) J_{elec,kWh} \frac{1}{1000 \cdot 3600} dt \tag{22}$$

for a fixed electricity price of $J_{elec,kWh} = 0.1 \frac{€}{kWh}$. By the denominator, the electricity price is converted to SI-units $[\frac{€}{Ws}]$. For DORFC, the monetary fatigue cost term (9) requires an expression for cost $J_{fatigue,c}$ of separate stress cycles. This expression is obtained by fitting the coefficients in (8) to Wöhler parameters of Tab. II and an exemplary machine cost of $J_{machine} = 4 \cdot 10^6 €$. This yields the polynomial cost curve in Fig. 3.

It needs to be stated that this cost curve is not valid for equivalent stresses close to ultimate tensile stress since cost of one cycle should approach total machine cost there. In the present work this is not crucial since equivalent stresses remain far below these levels. Since both cost function terms for DORFC are on the same monetary basis, the tuning weight can be set to $\alpha_{damage,DORFC} = 1$.

TABLE II: Material properties of steel of the wind turbine tower.

Quantity	Unit	Value
Yield stress R_p	$[MPa]$	355
Ultimate tensile stress R_m	$[MPa]$	400
Number N_{slo} of cycles at slope change (Wöhler)	$[-]$	$5 \cdot 10^6$
Stress $\sigma_{eq,slo}$ at slope change (Wöhler)	$[MPa]$	65.7
Slope k of first part of Wöhler curve	$[-]$	3
Slope k of second part of Wöhler curve	$[-]$	5
Number of cycles at fatigue endurance	$[-]$	$8 \cdot 10^6$

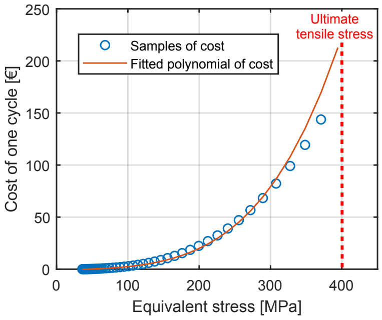


Fig. 3: Curve fitting of fatigue cost $J_{fatigue,c}(\sigma_{eq,c})$ of separate stress cycles.

D. Controller and Simulation Settings

For the MPC, a controller sample-time of $T_{cntrl} = 0.2s$, $N_u = 20$ control steps per prediction horizon and a fixed integrator step-time of $T_{integr} = 0.005s$ are used. The MPC is real-time feasible in this case. Length of the prediction horizon was chosen to $T_{horizon} = 4s$ to contain at least one full stress cycle at the first eigenfrequency of the wind turbine tower. Simulations in the present work are conducted using rotor-averaged IEC 61400 Class A turbulent wind data $V_w(t)$ of different reference velocities $V_{w,ref}$. These time series of $3600s$ each are generated by *TurbSim*.

V. Results of Application & Discussion

A. Tuning of Cost Function

Like mentioned in Section IV-C, determination of profit-optimal tuning weights α_{damage} for TTVP and AP requires extensive simulations. Since there is no common economic basis between revenue and fatigue cost function terms, even guessing suitable dimensions for the tuning weights is not intuitive. In contrary, DORFC can be applied immediately, only by setting electricity price, material parameters and the cost of machine failure.

B. Economic Controller Behavior

Behavior of absolute revenue in Fig. 4a is similar for all of the three methods. Towards high reference velocities of wind, revenue settles since the wind turbine is operated at rated conditions for an increasing portion of time. Like shown in Fig. 4b, TTVP and AP exhibit significantly lower revenue by 1% to 2% normalized to revenue of DORFC.

This especially is pronounced for below-rated reference wind velocities $V_{w,ref} < 9.8m/s$.

Absolute monetary fatigue cost in Fig. 4c is very low in comparison to revenue for all reference velocities. This as well was observed from simulations of another turbine at the same wind conditions. Relatively high fatigue costs occur at reference wind velocities close to rated wind velocity. This is influenced by the sharp maximum of thrust force at the transition from region II $1/2$ to region III and strong gradients of thrust force w.r.t. wind velocity in proximity to this peak. Thus, following fluctuations of wind velocity results in highly dynamic excitation of the turbine tower.

Like shown in [4], for the present wind turbine the Pareto optimum for TTVP occurs at low fatigue costs but as well at low revenue. For AP, the pareto-optimal weighting factor results in medium fatigue cost, like shown in Fig. 4c and Fig. 4d. For all reference wind velocities, the ENMPC using DORFC operates the turbine tower most aggressively and results in highest fatigue cost.

Profit of operation here is defined by the difference between revenue and fatigue cost. Normalized profit

$$Profit_{norm}(V_{w,ref}) = \frac{Revenue(V_{w,ref}) - CostFatigue(V_{w,ref})}{Revenue_{DORFC}(V_{w,ref}) - CostFatigue_{DORFC}(V_{w,ref})} \tag{23}$$

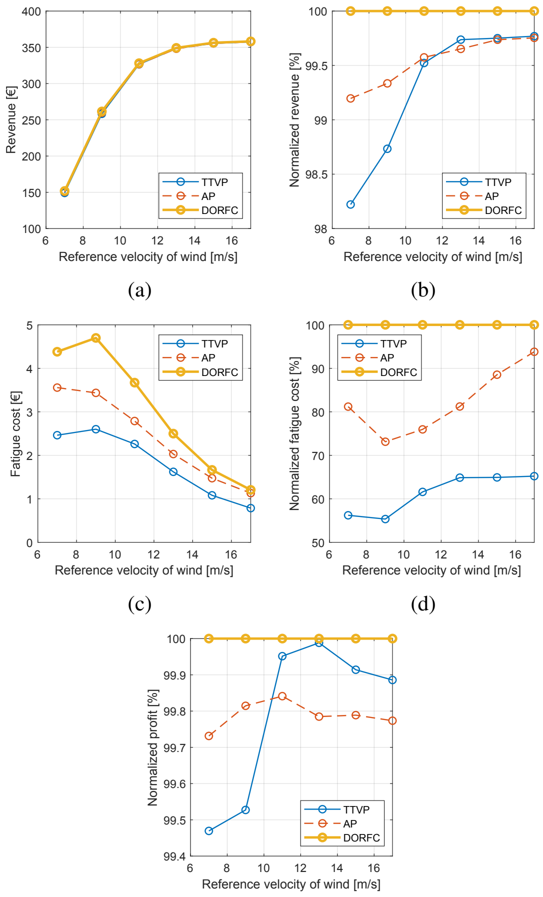
for a specific reference wind velocity in this case is defined by a normalization to the respective profit of DORFC.

Like shown in Fig. 4e, TTVP looses up to 0.5% of profit compared to DORFC. AP looses almost 0.3% for a reference wind velocity of $V_{w,ref} = 7m/s$. The profit of AP seems to be less sensitive to wind velocity than it is the case with TTVP. In general, significant benefits of using DORFC can be expected especially for below-rated wind velocities since there is more freedom in shifting power extraction in time. It can be concluded for the present application that DORFC economically correctly accepts higher fatigue damage in the tower in exchange to higher energy extraction from the wind in comparison to the other methods. Additionally, note that in contrary to TTVP and AP, the weight $\alpha_{damage,DORFC} = 1$ is not optimized.

C. Dynamic Controller Behavior

Fig. 5 shows the dynamic behavior of the wind turbine for turbulent wind of a reference velocity $V_{w,ref} = 9m/s$. This reference velocity results in high fatigue cost for all methods, like shown Section V-B.

TTVP very well smoothens the effect of turbulent wind on angular velocity of the rotor. AP and especially DORFC stronger follow the wind. For DORFC, this as well results in higher amplitudes in tower root stress which can be expected considering the fatigue analysis in Section V-B. However, this stress trajectory shows plateaus which as well are observed in [4]. While strongly following the wind, part of the damage is reduced by cutting the peaks of the stress cycles.



(a) (b) (c) (d) (e)

Fig. 4: Economic comparison of Tower-Tip-Velocity-Penalization (TTVP, weighting factor $\alpha_{damage} = 4200$), Amplitude Penalization (AP, $\alpha_{damage} = 2520$) and using Direct Online Rainflow-counting (DORFC, $\alpha_{damage} = 1$). a) Revenue by sold electric energy. b) Normalized revenue. c) Fatigue cost by stress oscillations at tower root. d) Normalized fatigue cost. e) Normalized profit.

Due to slightly higher amplitudes in generator torque and higher variability in angular velocity of the rotor, power quality is expected to be lower for DORFC in comparison to TTVP and AP. On the other hand, DORFC shows least use of the pitch actuators. This is highly beneficial since fatigue in the blades and in the pitch bearings is a further key cost driver which has to be considered.

VI. Conclusion & Outlook

Implementing the standard material fatigue estimation process in MPC via Direct Online Rainflow-counting exhibits practical advantages because the cost function only needs to be parameterized by physical and economical quantities. No extensive simulations are necessary to find suitable tuning weights like it is the case with Tower-Tip-

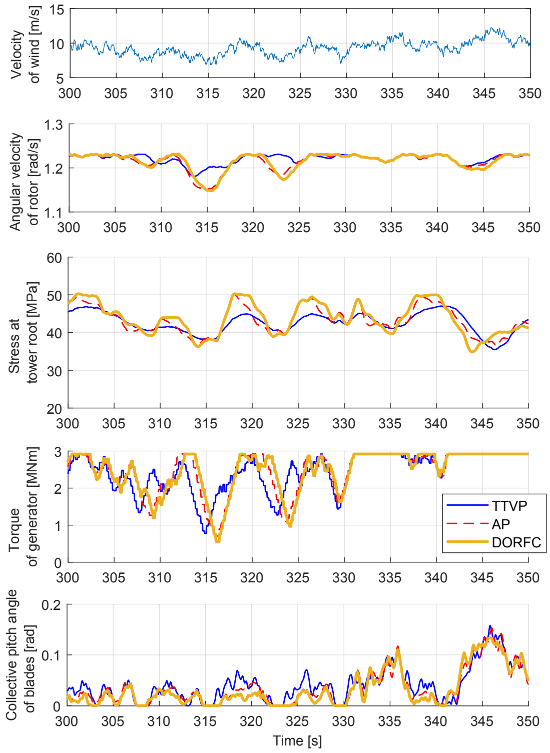


Fig. 5: Comparison of dynamic behavior using Tower-Tip-Velocity-Penalization (TTVP, weighting factor $\alpha_{damage} = 4200$), Amplitude Penalization (AP, $\alpha_{damage} = 2520$) and Direct Online Rainflow-counting (DORFC, $\alpha_{damage} = 1$) for a reference velocity $V_{w,ref} = 9m/s$.

Velocity-Penalization and Amplitude Penalization. Tower-Tip-Velocity-Penalization and Amplitude Penalization exhibit fair economic results while the latter seems to be less sensitive to reference wind velocity. Direct Online Rainflow-counting exhibits best profit due to higher total revenue while accepting higher damage of the turbine tower.

Dynamic behavior of DORFC exhibits characteristic plateaus in the stress trajectory which stem from the direct objective of reducing the amplitudes of stress cycles. Power quality of AP and DORFC seems to be lower than for TTVP. On the other hand, DORFC advantageously requires less pitch activity for the present case.

The present work is to be considered as a preliminary study, and further investigations are planned. Future work will be directed to:

- Including stress information from past measurements
- Research on uncertainty and robustness
- Application of DORFC within the *Simultaneous approach* (Multiple Shooting)
- Higher-sophisticated and business-oriented fatigue cost models
- Simultaneous consideration of fatigue cost for multiple wind turbine parts
- Realistic wind turbine controller tests using a high-fidelity wind turbine simulator
- Comparison of DORFC to further fatigue penalization methods
- Application of DORFC to other fatigue-driven systems.

ACKNOWLEDGMENT

The authors would like to thank the German Federal Ministry for Economic Affairs and Energy for funding the *MISTRALWIND* project.

REFERENCES

[1] R. Findeisen and F. Allgöwer, “An introduction to nonlinear model predictive control,” in *21st Benelux Meeting on Systems and Control*, 2002, pp. 1–23.

[2] M. A. Henson, “Nonlinear model predictive control: Current status and future directions,” *Computers & Chemical Engineering*, vol. 23, no. 2, pp. 187–202, 1998.

[3] J. J. Barradas-Berglind and R. Wisniewski, “Representation of fatigue for wind turbine control,” *Wind Energy*, vol. 19, no. 12, pp. 2189–2203, 2016.

[4] S. Löw, D. Obradovic, and C. L. Bottasso, “Model predictive control of fatigue by direct online rainflow-counting,” *In process of submission to IEEE Transactions on Control Systems Technology*, 2019.

[5] C. L. Bottasso, P. Pizzinelli, C. Riboldi, and L. Tasca, “Lidar-enabled model predictive control of wind turbines with real-time capabilities,” *Renewable Energy*, vol. 71, pp. 442–452, 2014.

[6] S. Raach, D. Schlipf, F. Sandner, D. Matha, and P. W. Cheng, “Nonlinear model predictive control of floating wind turbines with individual pitch control: 2014 american control conference,” in *2014 American Control Conference (ACC)*, 2014.

[7] S. Gros and A. Schild, “Real-time economic nonlinear model predictive control for wind turbine control: International journal of control,” *International Journal of Control*, pp. 1–14, 2017.

[8] M. Matsuishi and T. Endo, “Fatigue of metals subjected to varying stress,” *Japan Society of Mechanical Engineers, Fukuoka, Japan*, vol. 68, no. 2, pp. 37–40, 1968.

[9] ASTM, “Standard practices for cycle counting in fatigue analysis,” West Conshohocken, PA, 1985. [Online]. Available: https://cds.cern.ch/record/1463271

[10] The MathWorks Inc., “Rainflow counts for fatigue analysis,” 2018. [Online]. Available: https://de.mathworks.com/help/signal/ref/rainflow.html

[11] K. R. Chandran, “A physical model and constitutive equations for complete characterization of s-n fatigue behavior of metals,” *Acta Materialia*, vol. 121, pp. 85–103, 2016.

[12] M. A. Miner, “Culmulative damage in fatigue,” *Journal of Applied Mechanics*, pp. 159–164, 1945.

[13] E. Zeidler, G. Grosche, V. Ziegler, D. Ziegler, and I. N. Bronštejn, Eds., *Springer-Taschenbuch der Mathematik*, 3rd ed. Wiesbaden: Springer, 2013.

[14] S. Löw and D. Obradovic, “Real-time implementation of nonlinear model predictive control for mechatronic systems using a hybrid model,” *atp edition*, vol. 60, no. 08, pp. 46–53, 2018.

[15] S. Altmannshofer, “Fast suboptimal nonlinear model predictive control of an inverted pendulum,” *IFAC Proceedings Volumes*, vol. 45, no. 17, pp. 442–447, 2012.

[16] M. Papageorgiou, M. Leibold, and M. Buss, *Optimierung: Statische, dynamische, stochastische Verfahren für die Anwendung*, 4th ed. Springer Vieweg, 2015.

[17] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, “From linear to nonlinear mpc: Bridging the gap via the real-time iteration,” *International Journal of Control*, pp. 1–19, 2016.

3.3 Paper 3: Incorporation of fatigue into the MPC algorithm

3.3.1 Summary

The novel method "Direct Online Rainflow-counting" (DORFC) has been presented only with a minimum level of detail in **Paper 2**. The present paper closes this gap.

A literature review reveals that all indirect [32, 33] and direct methods [36–39] for the online implementation of fatigue in MPC either do not accurately approximate fatigue damage, or result in high computational load. Furthermore, the following obstacles are reported to stand in the way of direct implementation of RFC in MPC:

1. RFC requires the entire stress trajectory over the prediction horizon, and thus cannot be cast into standard stage or terminal cost functions;
2. It does not seem possible to derive analytical expressions of fatigue cost gradients, due to the RFC algorithm not having a "closed mathematical form";
3. RFC is a highly nonlinear algorithm, which exhibits discontinuous outputs because of its branches and loops.

These obstacles are overcome by the combined application of the following principles:

1. Separate: In the MPC implementation, the fatigue cost algorithm is inserted in between of the ODE solution and the QP solution, which are separated according to the *Sequential approach*;
2. Substitute: For the purpose of gradient computation, RFC as part of the fatigue cost algorithm is substituted by continuous linear expressions. These expressions are assumed to be locally valid for one MPC step;
3. Switch seldom: This local validity is supported by numerical studies using the wind turbine system which show that switches in the execution structure of RFC happen sufficiently seldom over the simulation time.

The numerical studies reveal the following four extra insights w.r.t. **Paper 2**: In the frequency space, DORFC beneficially focuses on the reduction of stress amplitudes, irrespective of the frequency where they occur. Online fatigue estimation ratios are at meaningful levels, and optimal cost tuning does not seem to imply meeting a certain estimation ratio. In the cost function, switching events happen sufficiently seldom. The fatigue cost gradients obtained by DORFC match with the ones obtained by the Finite Difference method.

3.3.2 Contribution

The author of this dissertation developed the core idea of Direct Online Rainflow Counting, formalized the approach, performed the implementation, and executed and analyzed the simulations. The author of this dissertation and CLB wrote the paper. DO and CLB put the formulation in the right context, recommended simulation cases, and supervised the work. All authors provided important input to this research work through discussions, feedback and by improving the manuscript.

3.3.3 Reference

S. Loew, D. Obradovic, and C. L. Bottasso, “Economic nonlinear model predictive control of fatigue—formulation and application to wind turbine control,” *Optimal Control Applications and Methods*, 2022. doi:10.1002/oca.2870

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*

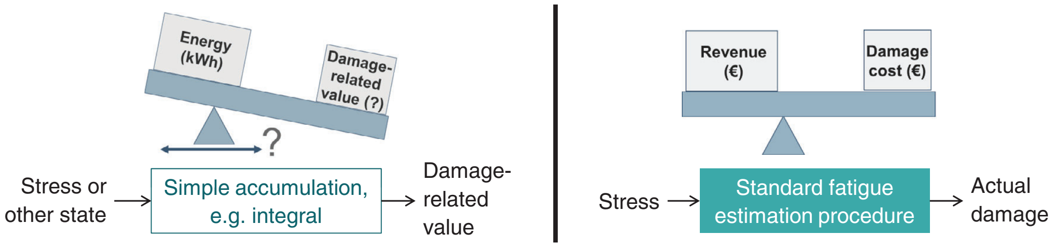


FIGURE 1 Left: Indirect fatigue metric. Right: Direct fatigue metric[13]

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]

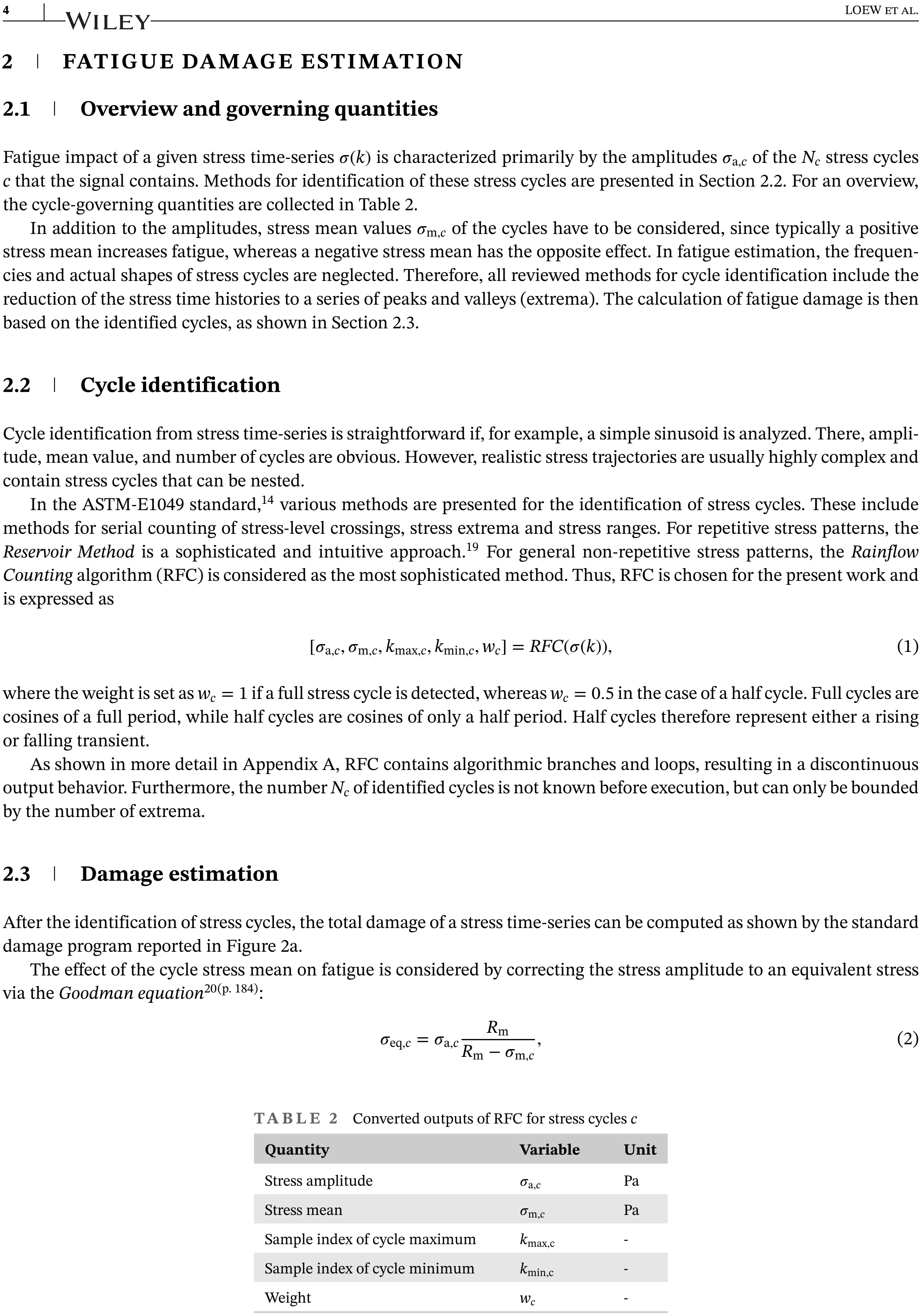
In summary, all available indirect or direct fatigue metrics either do not accurately approximate fatigue damage, or result in high computational load. Since RFC is the de facto standard for fatigue estimation, a direct and non-simplified implementation of RFC in MPC is desirable and promises best results. 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	-

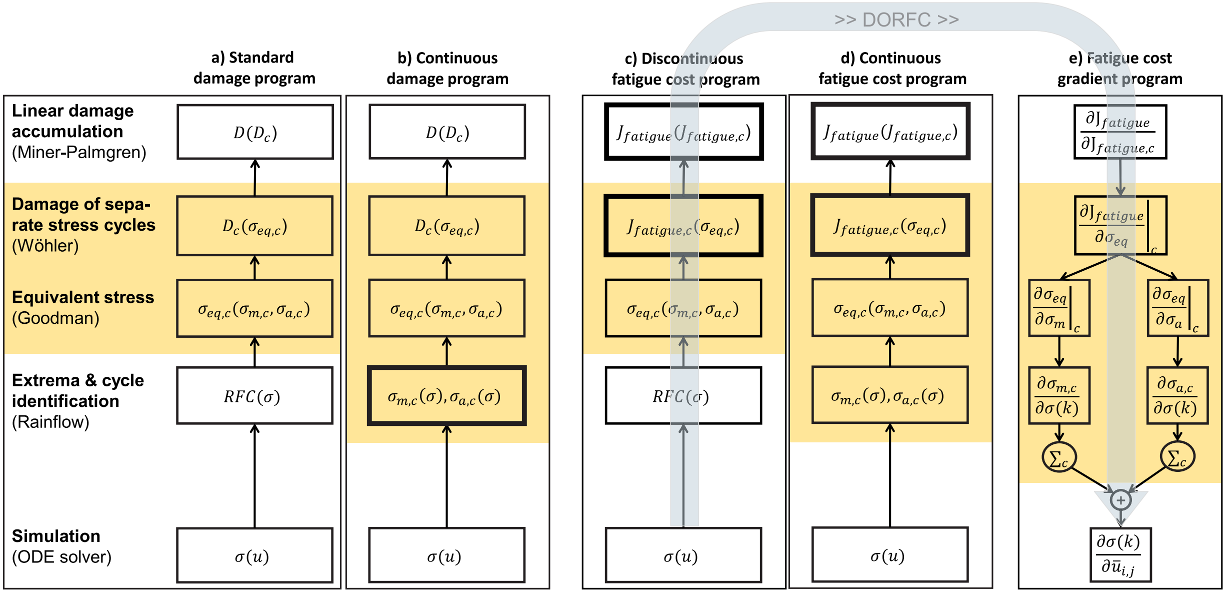

>> DORFC >>
a) Standard damage program
b) Continuous damage program
c) Discontinuous fatigue cost program
d) Continuous fatigue cost program
e) Fatigue cost gradient program
Linear damage accumulation (Miner-Palmgren)
Damage of separate stress cycles (Wöhler)
Equivalent stress (Goodman)
Extrema & cycle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

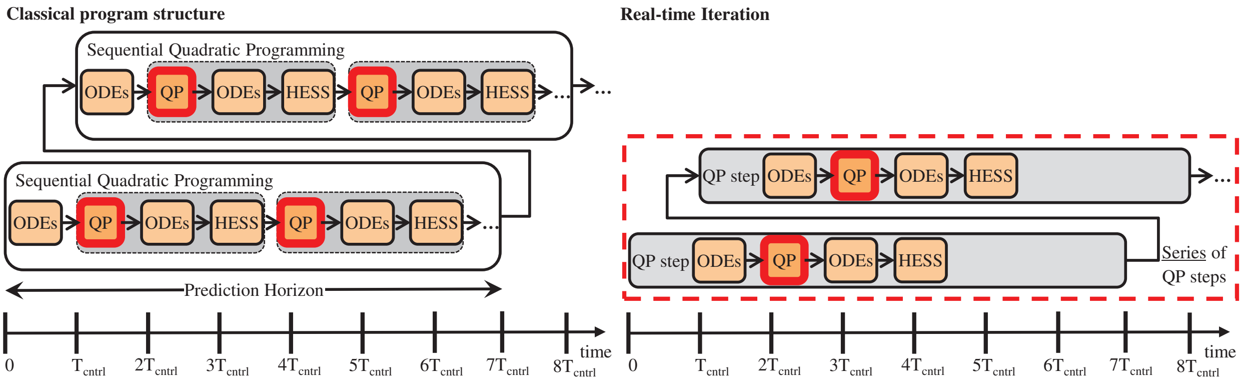


FIGURE 3 Comparison of the classical MPC program structure and the RTI method. HESS, update of the Hessian matrix; ODEs, numerical integration of the dynamics; QP, solution of the QP problem

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.

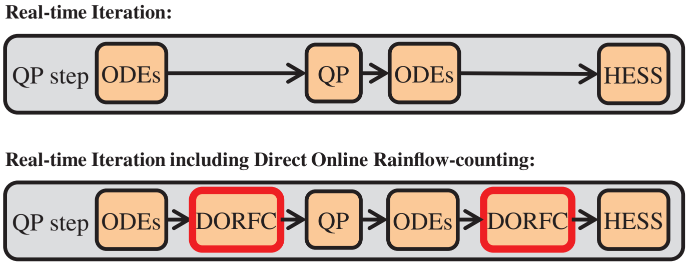


FIGURE 4 Integration of DORFC into the QP steps of the RTI method. HESS, update of Hessian Matrix; ODEs, numerical integration of the ordinary differential equations; QP, solution of a QP problem

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*(σ). 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 $(k_{\max,c}, k_{\min,c})$.*

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

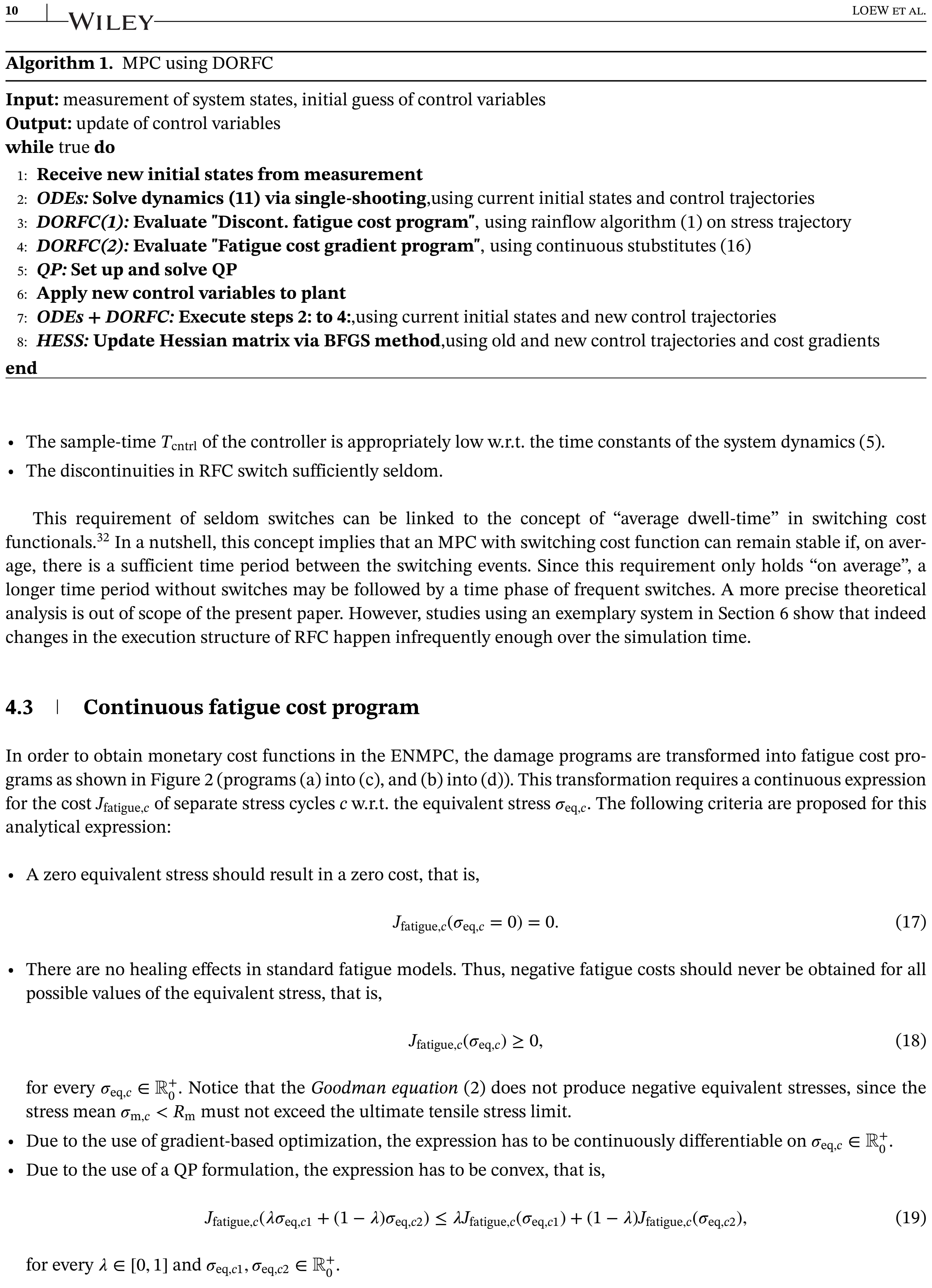
The substitution is performed for the stress mean $\sigma_{\mathrm{m},c}$ and stress amplitude $\sigma_{\mathrm{a},c}$ for separate cycles (see Table 2), by using their related maximum and minimum stress samples, which leads to

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

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 $\partial \boldsymbol{\sigma}_{\text{a}} / \partial \boldsymbol{\sigma}$ 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.

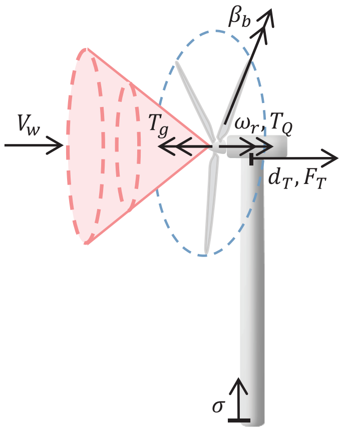


FIGURE 5 Sketch of the LiDAR-equipped wind turbine model with ambient wind speed V_{w}, generator torque T_{g}, rotational speed of the rotor ω_{r}, aerodynamic torque T_{Q}, displacement of tower tip d_{T}, aerodynamic thrust force F_{T}, collective blade pitch angle β_{b} and tensile stress at tower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

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

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

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

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

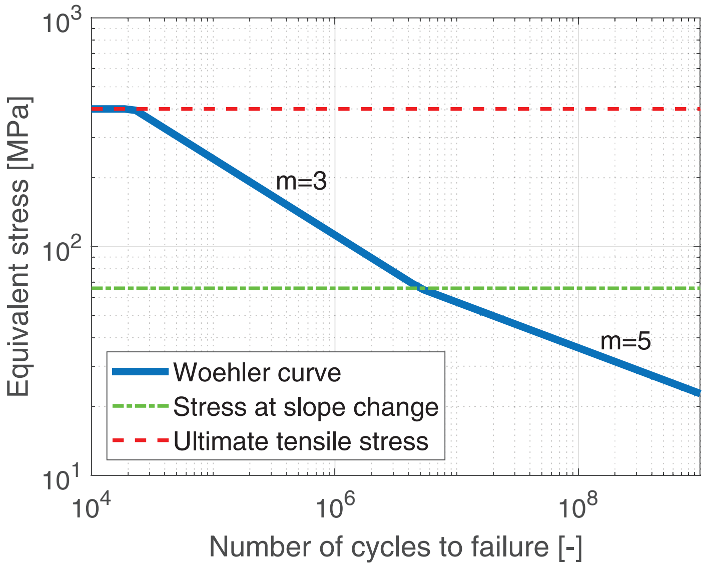


FIGURE 6 Woehler curve and characteristic stress levels in double-logarithmic representation

(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

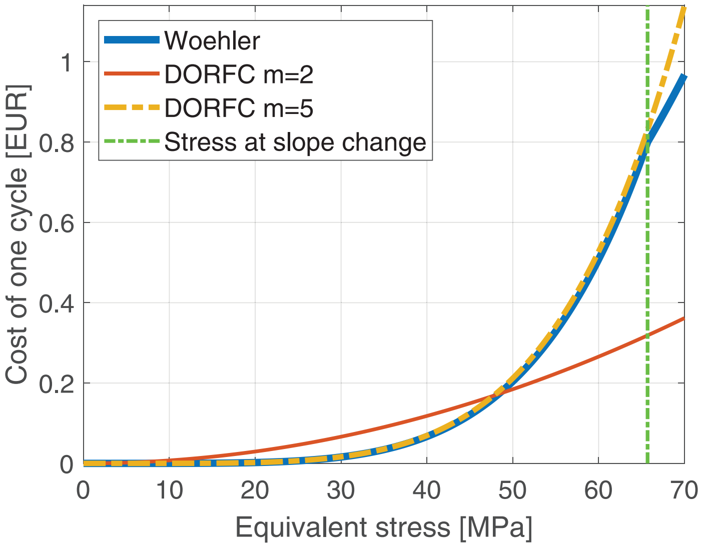


FIGURE 7 Cost of one stress cycle based on Woehler data, for approximations with exponent $m = 2$ and $m = 5$

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

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

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

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

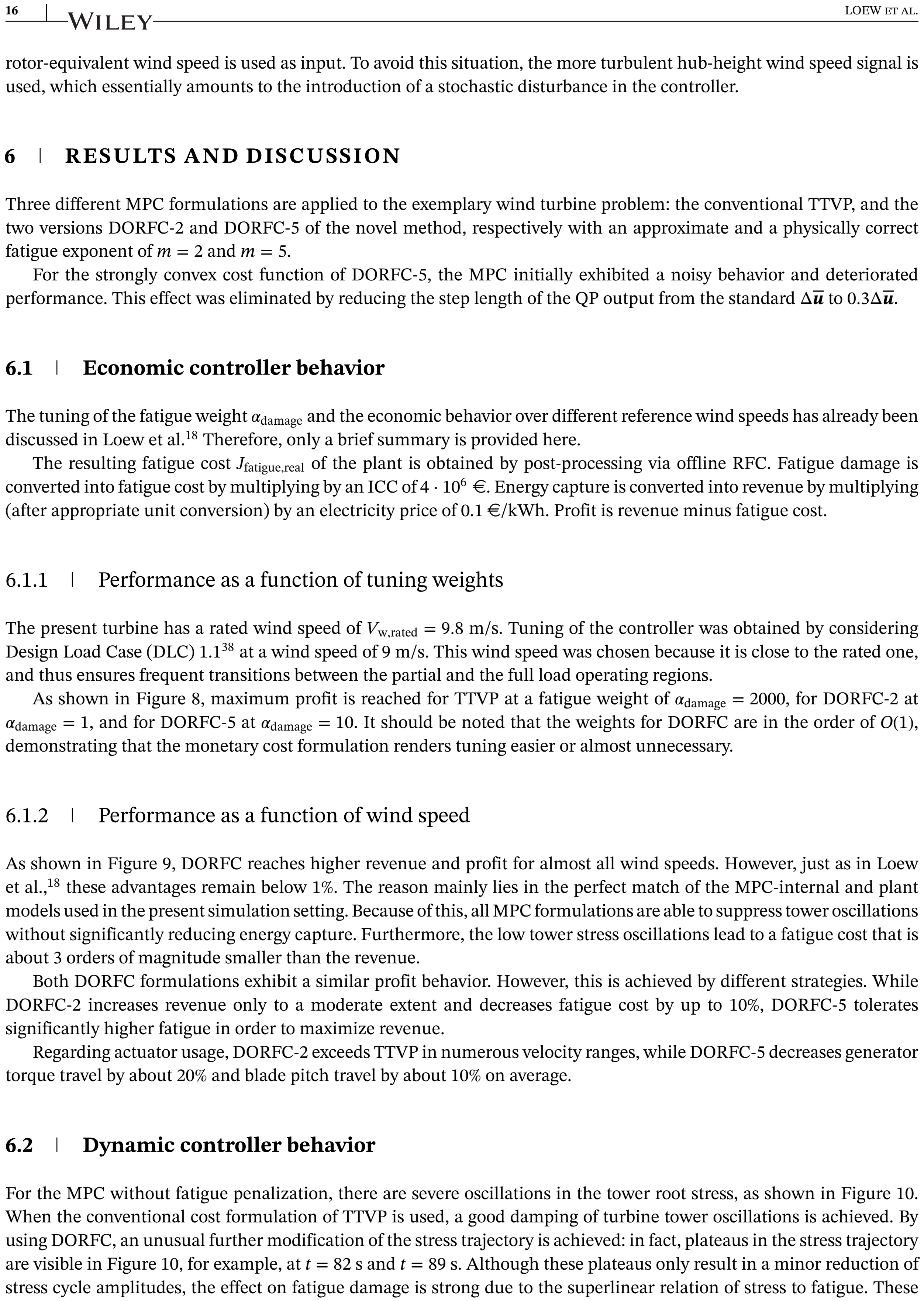
with coefficient $a_2 = 7.38 \cdot 10^{-5}$. Using this value of the coefficient, the *gradient* of (33) matches the gradient of the Woehler-based fatigue cost function at the frequently occurring equivalent stress level of $\sigma_{\text{eq},c} = 35$ MPa. The corresponding MPC formulation is termed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

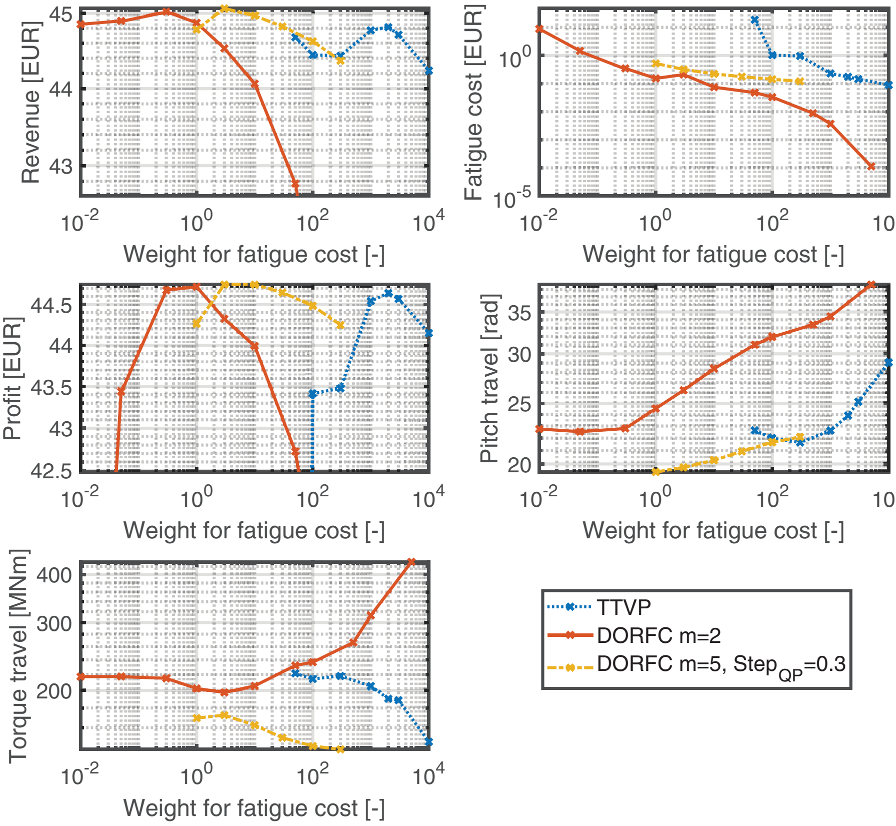


FIGURE 8 Variation of fatigue weights for three different MPC formulations. Each node represents an average over 12 simulations, each with a different turbulent seed

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):

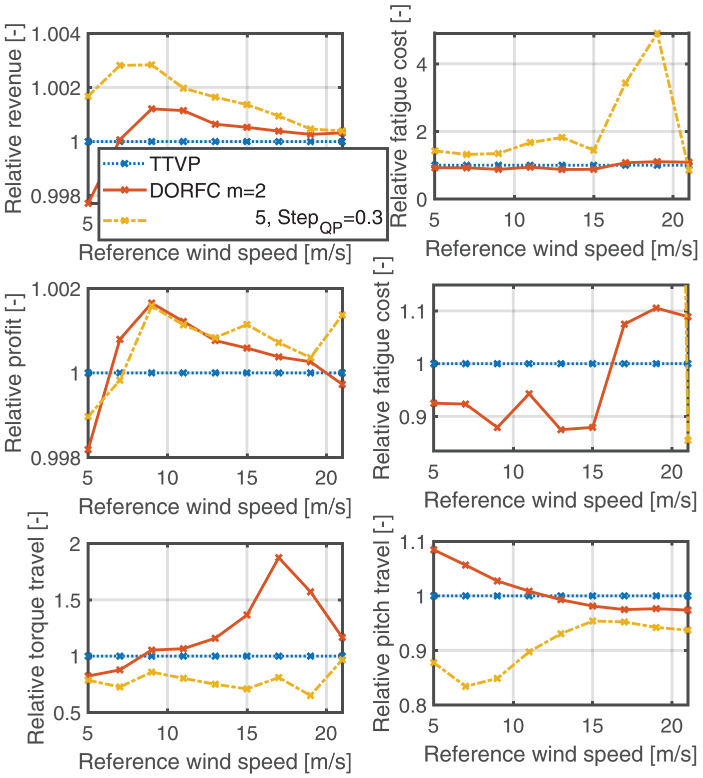


FIGURE 9 Performance as a function of wind speed for three different MPC formulations. All quantities are normalized w.r.t. the TTVP-based MPC. Middle row, right: zoomed version of fatigue cost plot. Each node represents an average over 12 simulations, each with a different turbulent seed DORFC m=2

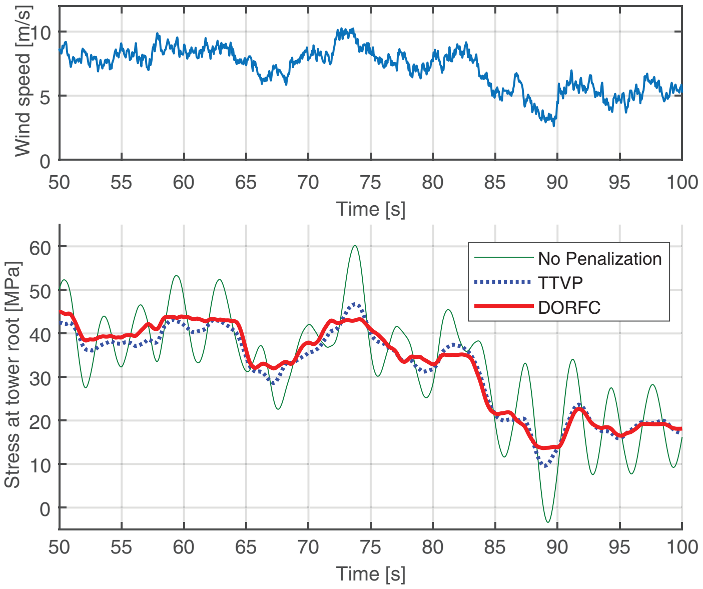


FIGURE 10 Comparison of the dynamic behavior for MPC without the penalization of fatigue, using TTVP and DORFC-2

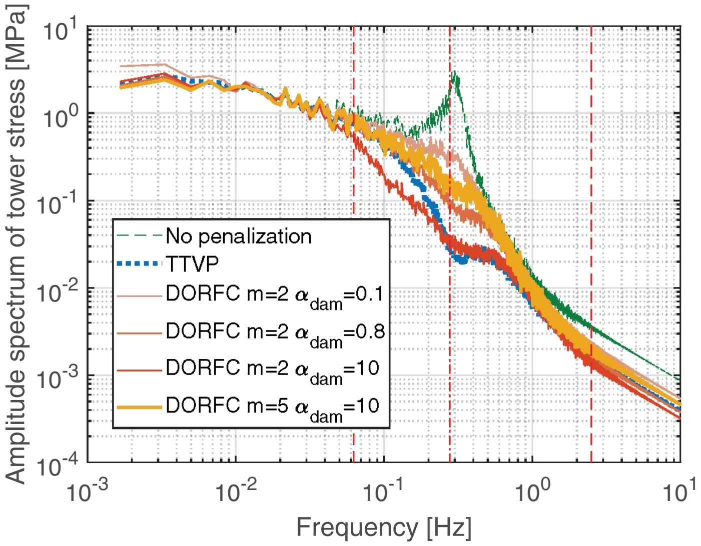


FIGURE 11 Amplitude spectrum of stress at tower root. DORFC-2 ($m = 2$) is re-tuned to $\alpha_{\text{damage}} = 0.8$ to achieve equal fatigue cost as TTVP. For comparison, results for a lower ($\alpha_{\text{damage}} = 0.1$) and a higher ($\alpha_{\text{damage}} = 10$) tuning of DORFC-2s are provided. 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.

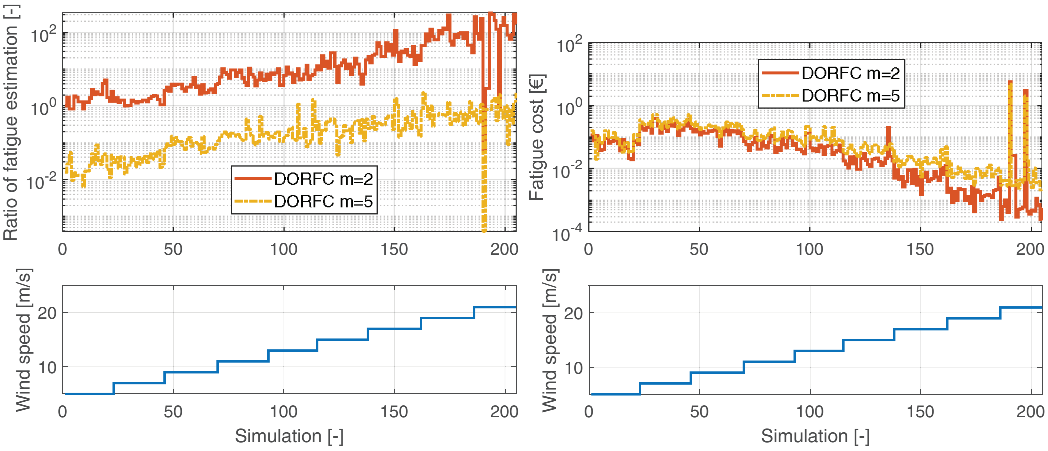


FIGURE 12 Ratio of fatigue cost estimation and fatigue cost over more than 200 individual simulations at different wind speeds

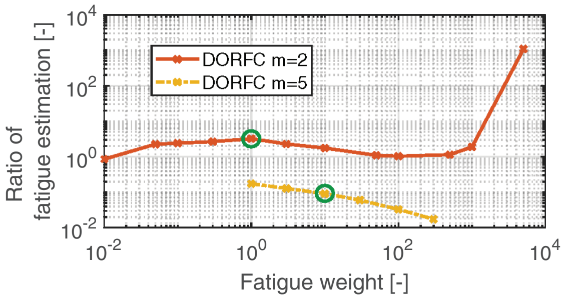


FIGURE 13 Ratio of fatigue cost estimation over fatigue weights at fixed wind speed. The green circles mark the optimum tuning weights. Each node represents an average over 12 simulations

6.4 | Seldom switching and continuous substitution

As shown in Section 4.2, the rainflow algorithm *RFC*(σ) 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.

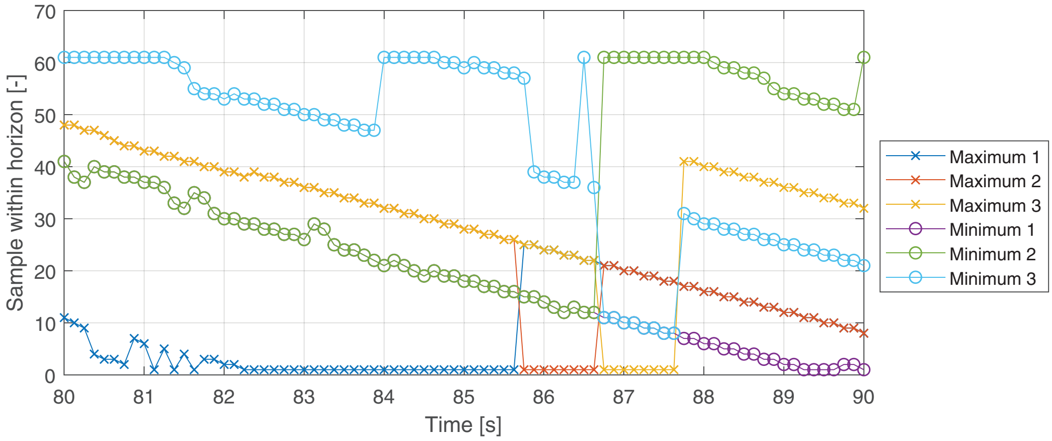


FIGURE 14 Sample positions of maxima $k_{\max,c}$ and minima $k_{\min,c}$ of the three most damaging stress cycles over an operation time of 10 s. The cycles are sorted in a descending order w.r.t. damage cost. Samples are displayed for each execution of DORFC that occurs every $T_{\text{cntrl}}/2 = 0.125$ 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

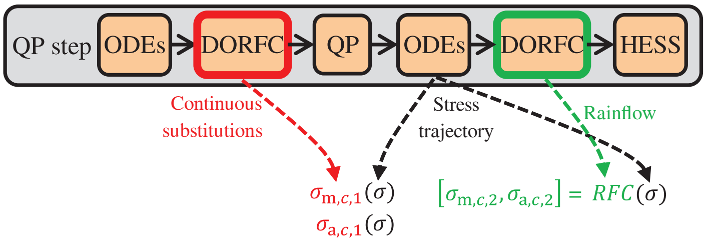


FIGURE 15 Setup for the validation of continuous substitution

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 $85\text{ s} < t < 88\text{ s}$. 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 $(\sigma_{\mathrm{m,c,1}}, \sigma_{\mathrm{a,c,1}})$ 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.

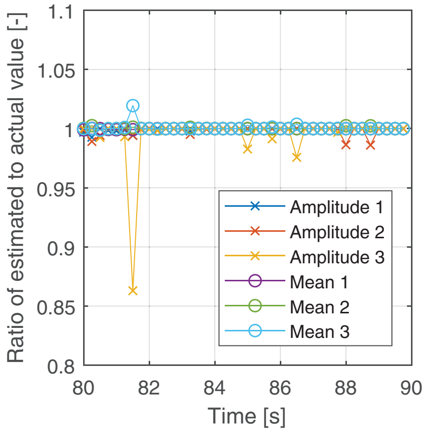


FIGURE 16 Ratio of output of continuous substitutes and output of actual rainflow counting. Ratios are displayed for amplitude and stress mean values of the three most damaging cycles over an operation time of 10 s. The cycles are sorted in a descending order w.r.t. damage cost. Samples are displayed 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 $\mathrm{d}J_{\mathrm{fatigue}}/\mathrm{d}\overline{u}_{1,1}$ 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 $f_{\mathrm{eigen,T,1}}$ 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.

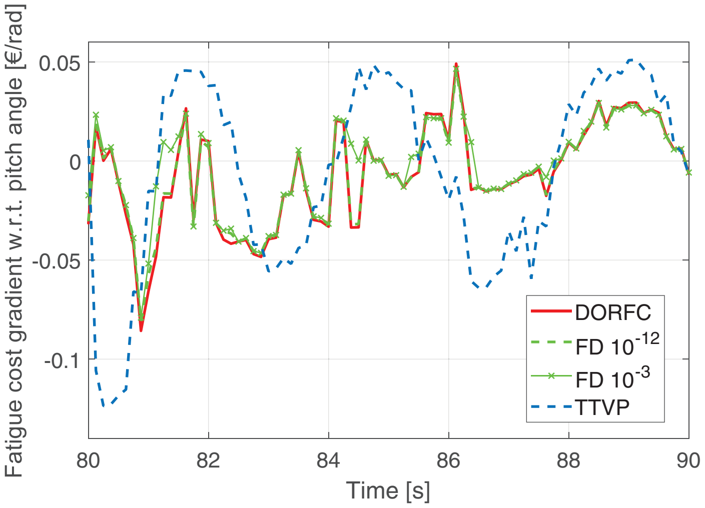


FIGURE 17 Gradients $\mathrm{d}J_{\mathrm{fatigue}}/\mathrm{d}\overline{u}_{1,1}$ of fatigue cost w.r.t. pitch angle in the first control interval of the prediction horizon. Comparison of gradients obtained by automatic differentiation within DORFC, by finite differences (FD) using a step size of 10^{-12} 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:

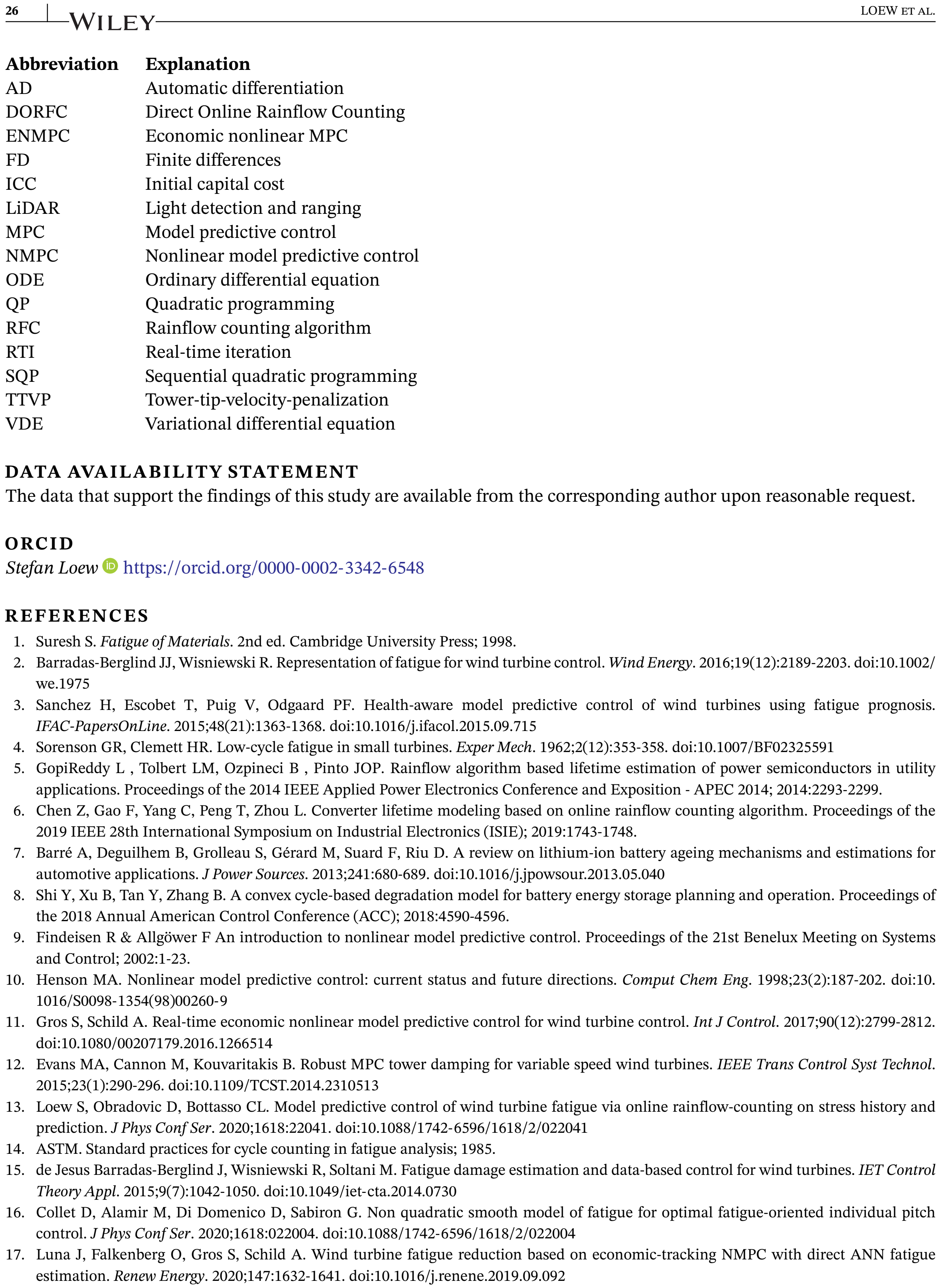
- Investigating the effects of uncertainties and robustness especially by considering the presence of noise in the measurements and of a plant-model mismatch;
- The simultaneous consideration of fatigue cost for multiple components of the turbine;
- The application to other devices, as for example batteries 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
$\sigma_{\text{a},c}$ (Pa)	Stress amplitude of one cycle
$\sigma_{\text{eq},c}$ (Pa)	Equivalent stress of one cycle
$\sigma_{\text{m},c}$ (Pa)	Stress mean of one cycle
$\sigma_{\text{r},c}$ (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*. 2022;1-30. 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

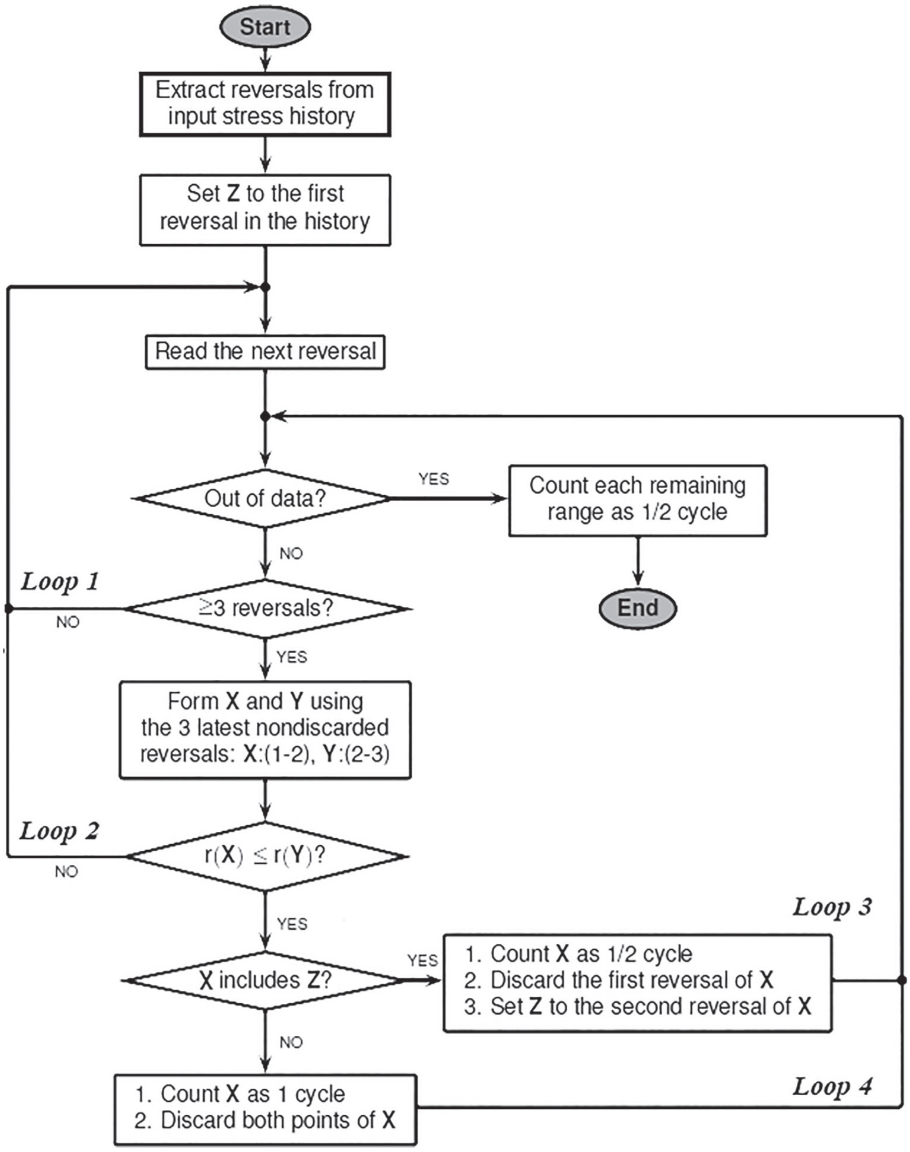

Start
Extract reversals from input stress history
Set Z to the first reversal in the history
Read the next reversal
Out of data?
YES
Count each remaining range as 1/2 cycle
End
NO
Loop 1
NO
≥3 reversals?
YES
Form X and Y using the 3 latest nondiscarded reversals: X:(1-2), Y:(2-3)
Loop 2
NO
r(X) ≤ r(Y)?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

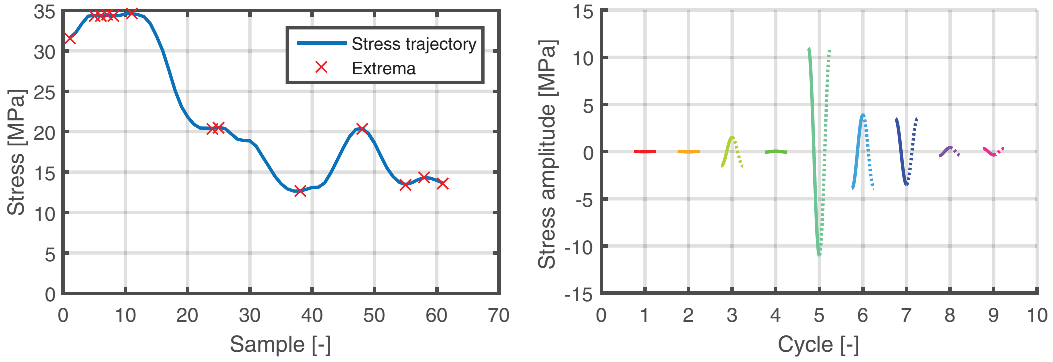


FIGURE A2 Left: Stress trajectory and identified extrema. Right: identified stress cycles. The cosines are flipped on the horizontal axis according to their appearance in the stress trajectory. All cycles are displayed with uniform period time. For the identified half cycles, the second half of the cosine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

$$\boldsymbol{x}(t) = \left(\omega_{\mathrm{r}}(t), d_{\mathrm{T}}(t), \dot{d}_{\mathrm{T}}(t), \beta_{\mathrm{b}}(t), \dot{\beta}_{\mathrm{b}}(t), T_{\mathrm{g}}(t)\right)^{T}, \tag{B1}$$

which is composed of the rotational speed of the rotor $\omega_{\mathrm{r}}(t)$, the fore-aft deflection of the tower tip $d_{\mathrm{T}}(t)$ and its time derivative, the collective pitch angle of the blades $\beta_{\mathrm{b}}(t)$ and its time derivative, and the generator torque $T_{\mathrm{g}}(t)$.

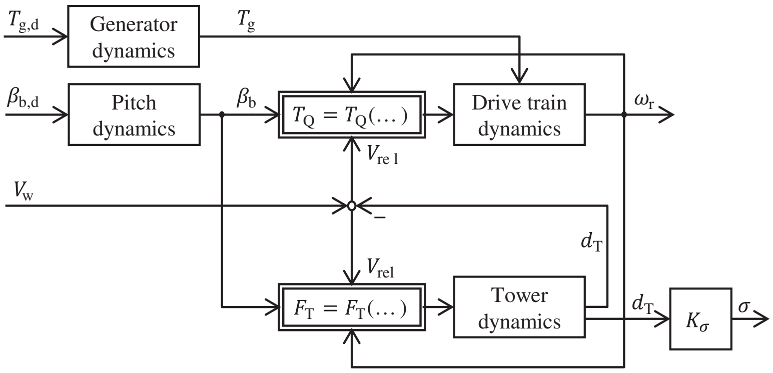


FIGURE B1 Coupled dynamics of wind turbine drive train and tower

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 $d_{\mathrm{T}}(t)$, resulting in $\sigma(t) = K_\sigma d_{\mathrm{T}}(t)$. The control variables for the subordinate controllers of the corresponding actuators are $\boldsymbol{u}(t) = \left(\beta_{\mathrm{b,d}}(t), T_{\mathrm{g,d}}(t)\right)^T$ where $\beta_{\mathrm{b,d}}(t)$ is the desired blade pitch angle and $T_{\mathrm{g,d}}(t)$ is the desired generator torque. The control variables are defined as piecewise-constant time functions $\overline{u}_{1,j} = \overline{\beta}_{\mathrm{b,d},j}$ and $\overline{u}_{2,j} = \overline{T}_{\mathrm{g,d},j}$. 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 $d(t) = V_{\mathrm{w}}(t)$ of the system stem from the ambient wind speed $V_{\mathrm{w}}(t)$, 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

$$T_{\mathrm{Q}}(t) = T_{\mathrm{Q}}(\omega_{\mathrm{r}}(t), V_{\mathrm{rel}}(t), \beta_{\mathrm{b}}(t)), \tag{B2}$$

and the aerodynamic thrust force

$$F_{\mathrm{T}}(t) = F_{\mathrm{T}}(\omega_{\mathrm{r}}(t), V_{\mathrm{rel}}(t), \beta_{\mathrm{b}}(t)). \tag{B3}$$

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

$$V_{\mathrm{rel}}(t) = V_{\mathrm{w}}(t) - \dot{d}_{\mathrm{T}}(t), \tag{B4}$$

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

$$\dot{\omega}_{\mathrm{r}}(t) = \frac{1}{J_{\mathrm{tot}}}\left(T_{\mathrm{Q}}(t) - T_{\mathrm{g}}(t)\right), \tag{B5}$$

and the tower fore-aft dynamics

$$\ddot{d}_{\mathrm{T}}(t) = \frac{1}{m_{\mathrm{T}}}\left(F_{\mathrm{T}}(t) - c_{\mathrm{T}}\dot{d}_{\mathrm{T}}(t) - k_{\mathrm{T}}d_{\mathrm{T}}(t)\right), \tag{B6}$$

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.

3.4 Paper 4: Externalization of fatigue from the MPC algorithm

3.4.1 Summary

In the formulation of DORFC in **Papers 2** and **3**, the fatigue cost computation is inserted in between of the ODE solution and the QP solution in the MPC algorithm. However, in contrast to the utilized in-house MPC implementation from **Paper 1**, most standard MPC frameworks like `ACADO Toolkit` impede such modifications, perform better with the *Simultaneous approach*, and only allow for standard stage and terminal cost functions. Furthermore, formal proofs of stability and recursive feasibility typically are researched for those standard cost functions only.

In order to overcome these limitations, the fatigue estimation is taken out (externalized) of the core MPC but still executed at each MPC step. The results of fatigue estimation then are fed to the MPC via time-varying parameters. Within the MPC, a standard economic stage cost formulation with time-varying parameters remains. Thus, the formulation will be referred to as "Parametric Online Rainflow-counting" (PORFC).

In DORFC, fatigue damage is a sum over damages from individual *stress cycles*. In PORFC, this is replaced by a sum over damages from individual *time samples* which motivates the use of time-varying parameters to transfer this information to the MPC optimization problem. The summation over time samples is achieved by splitting the damage of each cycle in two halves, and allocating those to the two time samples which define the stress cycle. The parameters are stress mean and cycle weight, and are calculated based on a pre-simulation which matches the internal simulation of the MPC.

To complete the picture, also a tracking stage cost is formulated. In a nutshell, here the parameters of PORFC are condensed to a time-varying stress reference and a tracking weight.

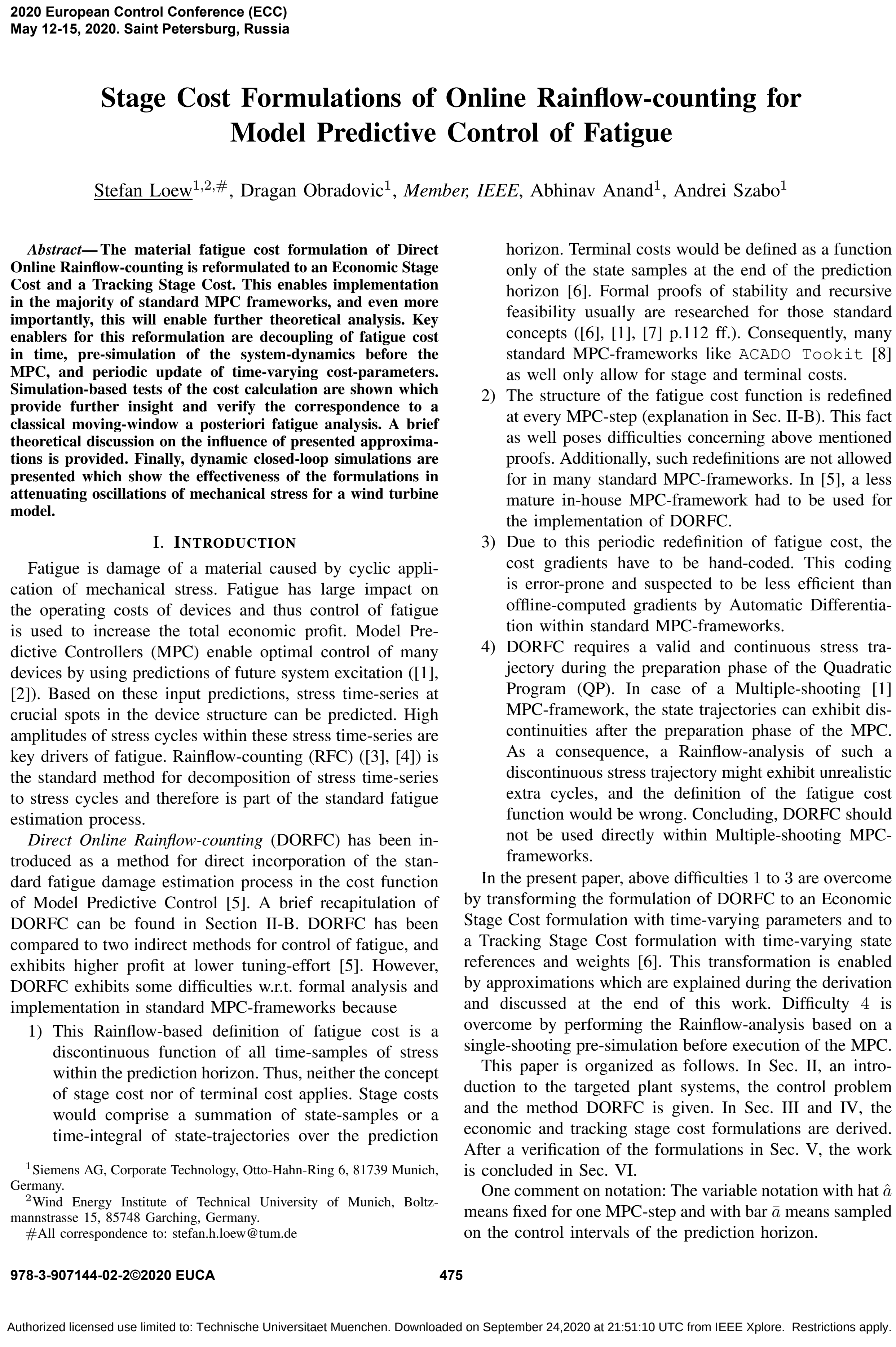
Simulations show that the accumulated stage costs compare very well to an a posteriori Rainflow analysis, exhibit ramps at samples which contribute to a cycle, and exhibit noteworthy acausal behavior. Furthermore, a closed-loop simulation shows the successful mitigation of highly damaging stress cycles. For certain successive MPC steps, jumps to different prediction trajectories are observed which are suspected to be caused by jumps into new local minima of the optimization problem.

3.4.2 Contribution

The author of this dissertation developed the core idea of the fatigue stage costs, performed the implementation, executed and analyzed the simulations, and wrote the paper. DO and AS supervised the work. All authors jointly formalized the idea, and provided important input to this research work through discussions, feedback and by improving the manuscript.

3.4.3 Reference

S. Loew, D. Obradovic, A. Anand, and A. Szabo, “Stage cost formulations of online rainflow-counting for model predictive control of fatigue,” in *2020 European Control Conference (ECC)*, 2020, pp. 475–482. doi:10.23919/ECC51009.2020.9143939 © 2020 IEEE. Reprinted with permission.



2020 European Control Conference (ECC)
May 12-15, 2020. Saint Petersburg, Russia

Stage Cost Formulations of Online Rainflow-counting for Model Predictive Control of Fatigue

Stefan Loew[1,2,#], Dragan Obradovic[1], *Member, IEEE*, Abhinav Anand[1], Andrei Szabo[1]

***Abstract*— The material fatigue cost formulation of Direct Online Rainflow-counting is reformulated to an Economic Stage Cost and a Tracking Stage Cost. This enables implementation in the majority of standard MPC frameworks, and even more importantly, this will enable further theoretical analysis. Key enablers for this reformulation are decoupling of fatigue cost in time, pre-simulation of the system-dynamics before the MPC, and periodic update of time-varying cost-parameters. Simulation-based tests of the cost calculation are shown which provide further insight and verify the correspondence to a classical moving-window a posteriori fatigue analysis. A brief theoretical discussion on the influence of presented approximations is provided. Finally, dynamic closed-loop simulations are presented which show the effectiveness of the formulations in attenuating oscillations of mechanical stress for a wind turbine model.**

I. Introduction

Fatigue is damage of a material caused by cyclic application of mechanical stress. Fatigue has large impact on the operating costs of devices and thus control of fatigue is used to increase the total economic profit. Model Predictive Controllers (MPC) enable optimal control of many devices by using predictions of future system excitation ([1], [2]). Based on these input predictions, stress time-series at crucial spots in the device structure can be predicted. High amplitudes of stress cycles within these stress time-series are key drivers of fatigue. Rainflow-counting (RFC) ([3], [4]) is the standard method for decomposition of stress time-series to stress cycles and therefore is part of the standard fatigue estimation process.

Direct Online Rainflow-counting (DORFC) has been introduced as a method for direct incorporation of the standard fatigue damage estimation process in the cost function of Model Predictive Control [5]. A brief recapitulation of DORFC can be found in Section II-B. DORFC has been compared to two indirect methods for control of fatigue, and exhibits higher profit at lower tuning-effort [5]. However, DORFC exhibits some difficulties w.r.t. formal analysis and implementation in standard MPC-frameworks because

1) This Rainflow-based definition of fatigue cost is a discontinuous function of all time-samples of stress within the prediction horizon. Thus, neither the concept of stage cost nor of terminal cost applies. 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 a function only of the state samples at the end of the prediction horizon [6]. Formal proofs of stability and recursive feasibility usually are researched for those standard concepts ([6], [1], [7] p.112 ff.). Consequently, many standard MPC-frameworks like `ACADO Tookit` [8] as well only allow for stage and terminal costs.
2) The structure of the fatigue cost function is redefined at every MPC-step (explanation in Sec. II-B). This fact as well poses difficulties concerning above mentioned proofs. Additionally, such redefinitions are not allowed for in many standard MPC-frameworks. In [5], a less mature in-house MPC-framework had to be used for the implementation of DORFC.
3) Due to this periodic redefinition of fatigue cost, the cost gradients have to be hand-coded. This coding is error-prone and suspected to be less efficient than offline-computed gradients by Automatic Differentiation within standard MPC-frameworks.
4) DORFC requires a valid and continuous stress trajectory during the preparation phase of the Quadratic Program (QP). In case of a Multiple-shooting [1] MPC-framework, the state trajectories can exhibit discontinuities after the preparation phase of the MPC. As a consequence, a Rainflow-analysis of such a discontinuous stress trajectory might exhibit unrealistic extra cycles, and the definition of the fatigue cost function would be wrong. Concluding, DORFC should not be used directly within Multiple-shooting MPC-frameworks.

In the present paper, above difficulties 1 to 3 are overcome by transforming the formulation of DORFC to an Economic Stage Cost formulation with time-varying parameters and to a Tracking Stage Cost formulation with time-varying state references and weights [6]. This transformation is enabled by approximations which are explained during the derivation and discussed at the end of this work. Difficulty 4 is overcome by performing the Rainflow-analysis based on a single-shooting pre-simulation before execution of the MPC.

This paper is organized as follows. In Sec. II, an introduction to the targeted plant systems, the control problem and the method DORFC is given. In Sec. III and IV, the economic and tracking stage cost formulations are derived. After a verification of the formulations in Sec. V, the work is concluded in Sec. VI.

One comment on notation: The variable notation with hat $\hat{a}$ means fixed for one MPC-step and with bar $\bar{a}$ means sampled on the control intervals of the prediction horizon.

[1] Siemens AG, Corporate Technology, Otto-Hahn-Ring 6, 81739 Munich, Germany.

[2] Wind Energy Institute of Technical University of Munich, Boltzmannstrasse 15, 85748 Garching, Germany.

[#] All correspondence to: stefan.h.loew@tum.de

978-3-907144-02-2©2020 EUCA

II. Fundamentals

A. Plant system and control problem

The present work can be applied to general continuous nonlinear systems

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t)), \tag{1}$$

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.

The NMPC problem is transformed to a Nonlinear Programming (NLP) problem by discretizing the time-continuous control variables $u_i(t)$ to piecewise-constant optimization variables $\bar{u}_{i,j}$. Here, individual control inputs to the plant system are determined by $i \in N_{in}$ and the respective control interval within the prediction horizon by $j \in N_u$. The number of control inputs is denoted by N_{in} and the number of control samples by $N_u = \frac{T_{horizon}}{T_{cntrl}}$ with horizon length $T_{horizon} = t_{end} - t_0$ and controller sample-time T_{cntrl}. This NLP is solved via *Real-time Iteration* [9] where one Quadratic Programming (QP) problem is solved per MPC-step and suboptimal control variables are applied to the plant.

B. Recapitulation of Direct Online Rainflow-counting

Direct Online Rainflow-counting incorporates cycle identification from a stress trajectory, mean-stress effect, Wöhler curves and Linear Damage Accumulation into the cost function of Model Predictive Control. An overview is given via Alg. 1.

Cycle identification is straightforward if e.g. a simple sinusoid is analysed. There, amplitudes, mean values and number of cycles are obvious. However, realistic stress trajectories usually are highly complex and contained stress cycles can be nested. Additionally, half and full cycles can be present. The most accepted algorithm for cycle identification from complex trajectories is the Rainflow(-counting) algorithm (RFC) which as well is used in DORFC. The outputs of RFC can be converted to the variables in Table I where the value of weight equals $w_c = 1$ if a full stress cycle is detected and equals $w_c = 0.5$ if a half cycle is detected.

TABLE I: Converted outputs of RFC for stress cycles c

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

The Rainflow-algorithm is based on reversals (extrema) of the stress trajectory and contains algorithmic branches. Thus, one property of the Rainflow-algorithm is its discontinuous output-behavior. Therefore, for gradient calculation in DORFC, at every MPC-step the Rainflow-algorithm is substituted by linear functions based on its outputs. This substitution is possible because the structure of identified cycles can be assumed to be fixed during one optimization step in MPC. This structure includes number of cycles $N_c = \max(c)$, cycle weights w_c and cycle samples $(k_{max,c}, k_{min,c})$. Consequently, only stress mean values $\sigma_{m,c}$ and stress amplitudes $\sigma_{a,c}$ remain variable. Continuous expressions of stress mean

$$\sigma_{m,c} = \frac{\sigma(\hat{k}_{max,c}) + \sigma(\hat{k}_{min,c})}{2} \tag{2}$$

and stress amplitude

$$\sigma_{a,c} = \frac{\sigma(\hat{k}_{max,c}) - \sigma(\hat{k}_{min,c})}{2} \tag{3}$$

are obtained using their related maximum and minimum stress samples. The positions $k_{max,c}$ and $k_{min,c}$ of these stress samples within the MPC horizon are obtained from the Rainflow-algorithm. Note: The sample indices are defined to $k \in [0, N_u]$ where $k = 0$ corresponds to $t = t_0$ and $k = N_u$ to $t = t_{end}$.

The effect of mean stress of cycles on fatigue is considered by a correction of stress amplitude to an equivalent stress via the *Goodman equation* [10] (p. 184)

$$\sigma_{eq,c} = \sigma_{a,c} \frac{R_m}{R_m - \sigma_{m,c}} \tag{4}$$

where R_m denotes ultimate tensile stress. Equivalent stress is converted to fatigue cost $J_{fatigue,c}$ of separate stress cycles c which can be expressed by an exponentiation

$$J_{fatigue,c}(\sigma_{eq,c}, \hat{w}_c) = \hat{w}_c \; a_m \; \sigma_{eq,c}^m \tag{5}$$

of the same order like the Woehler exponent for low stress amplitudes. For the present example, the exponent $m = 5$ is obtained from the material-specific Woehler-curve and the coefficient a_m is determined from *Initial Capital Cost* of the machine (see also [5]). Analogously to the standard procedure for fatigue damage, fatigue cost is accumulated linearly by

$$J_{fatigue,DORFC} = \sum_{c=1}^{N_c} J_{fatigue,c}. \tag{6}$$

III. Economic Stage Cost with Time-varying Parameters

A. Formulation

In order to fit the method of Direct Online Rainflow-counting into a classical MPC-formulation, this non-stage non-terminal fatigue cost formulation is transformed to an economic stage cost formulation with time-varying parameters. The presented method will be referred to as *Parametric Online Rainflow-counting* (PORFC).

Inserting (2) and (3) into the Goodman equation (4) and the latter into fatigue cost of individual cycles (5), total accumulated fatigue cost (6) can be expressed by

$$J_{fatigue,DORFC} = \sum_{c=1}^{N_c} J_{fatigue,c}(\sigma(\hat{k}_{max,c}), \sigma(\hat{k}_{min,c}), \hat{w}_c). \tag{7}$$

This equation highlights that the fatigue cost of each cycle c depends on two stress samples whose positions in the prediction horizon are determined by $\hat{k}_{max,c}$ and $\hat{k}_{min,c}$.

Algorithm 1 MPC using Direct Online Rainflow-counting (DORFC)

```
Input: Periodic measurement of system states, Initial guess of control variables
Output: Periodic update of control variables
while true do
    Receive new initial states from measurement
    Start preparation of QP by simulating state and state gradient dynamics via single-shooting
        Using current initial states and guess of control variables
    Evaluate Discontinuous Fatigue Cost program
        Based on stress trajectory, Using Rainflow-algorithm
    Evaluate Fatigue Cost Gradient program
        Using continuous stubstitutes (2) and (3)
    Finish preparation and solve QP
        Based on result of single-shooting and of fatigue cost gradient program
    Apply updated control variables to plant
end
```

These positions and the cycle weights $\hat{w}_c$ are outputs of the Rainflow-algorithm like shown in Tab. I. Consequently, this fatigue cost of individual cycles (7) is split w.r.t. both contributing stress samples

$$J_{fatigue,DORFC} \approx \sum_{c=1}^{N_c} \Bigg(\frac{1}{2} J_{fatigue,c}(\sigma(\hat{k}_{max,c}), \hat{\sigma}(\hat{k}_{min,c}), \hat{w}_c) + \frac{1}{2} J_{fatigue,c}(\hat{\sigma}(\hat{k}_{max,c}), \sigma(\hat{k}_{min,c}), \hat{w}_c) \Bigg) \tag{8}$$

where now for each term only one stress value σ remains variable and the complementary stress value $\hat{\sigma}$ becomes a parameter. E.g. for the first cost term, stress mean now is calculated by

$$\sigma_{m,c} = \frac{\sigma(\hat{k}_{max,c}) + \hat{\sigma}(\hat{k}_{min,c})}{2} \tag{9}$$

and stress amplitude by

$$\sigma_{a,c} = \frac{\sigma(\hat{k}_{max,c}) - \hat{\sigma}(\hat{k}_{min,c})}{2}. \tag{10}$$

However, this fatigue cost formulation unfavorably encourages the optimization algorithm to shift the variable stress value towards congruence with the fixed complementary stress value. Via the first cost term e.g., the variable $\sigma(\hat{k}_{max,c})$ is shifted towards the fixed complementary $\hat{\sigma}(\hat{k}_{min,c})$. However, conversely the same happens with the variable complementary stress sample $\sigma(\hat{k}_{min,c})$ via the second cost term. Thereby, within one QP-step the two samples can exchange their role as maximum or minimum sample, respectively. As a result, practical observations - like in Fig. 1 - exhibit stress trajectories which are alternating between vertically flipped realizations for consecutive optimization steps.

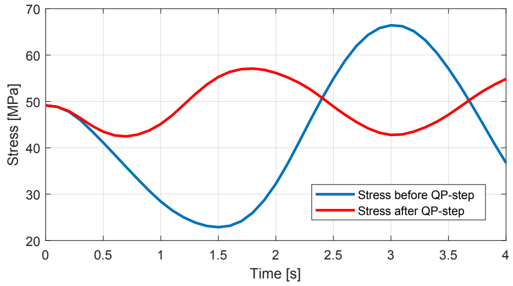


Fig. 1: PORFC using parametric complementary stress values: From blue to red graph, two local minima ($t = 1.5$s, $t = 4$s) and one local maximum ($t = 3$s) approximatively change their role. In the red graph, an additional minimum ($t = 0.7$s) appears.

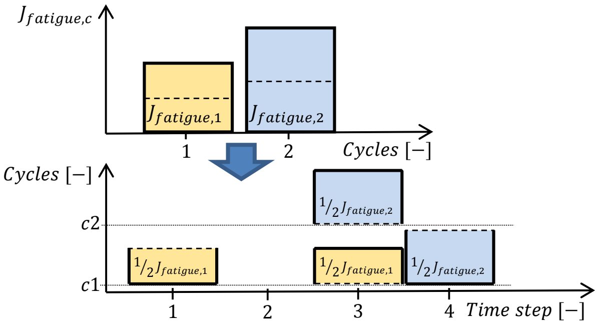


Fig. 2: Decoupling of stress cycles in time by assigning half of the fatigue cost to each contributing stress sample. For example, cycle 1 is defined by the two stress samples at time steps 1 and 3.

This phenomenon is counteracted by a formulation

$$J_{fatigue,DORFC} \approx \sum_{c=1}^{N_c} \Bigg(\frac{1}{2} J_{fatigue,c}(\sigma(\hat{k}_{max,c}), \hat{\sigma}_{m,c}, \hat{w}_c) + \frac{1}{2} J_{fatigue,c}(\sigma(\hat{k}_{min,c}), \hat{\sigma}_{m,c}, \hat{w}_c) \Bigg) \tag{11}$$

where the mean stress value instead of the complementary stress values is a parameter. For both cost terms, amplitude stress now is calculated by

$$\sigma_{a,c} = \left| \sigma(\hat{k}_{max/min,c}) - \hat{\sigma}_{m,c} \right|. \tag{12}$$

An unfavorable property of this formulation is that a reduction of mean stress cannot be encouraged anymore. However, via the Goodman equation (4) parametric mean stress still amplifies the influence of the related amplitude stress. Since formulation (11) adds an approximation, further studies on convergence of the original problem (8) may be worthwhile as well.

By the preceding derivation, the fatigue cost calculation was decoupled in time which is shown in Fig. 2 for an exemplary case of two stress cycles. Thus, this formulation of a summation over cycles c with each two temporal terms can be converted to the stage cost formulation

$$J_{fatigue,DORFC} \approx \sum_{k=1}^{N_u} \left(J_{fatigue,c}(\sigma(k), \hat{\sigma}_{m,c1}(k), \hat{w}_{c1}(k)) + J_{fatigue,c}(\sigma(k), \hat{\sigma}_{m,c2}(k), \hat{w}_{c2}(k))\right) \tag{13}$$

of a summation over samples k with each two cyclic terms. Only two terms for two cycles $c1$ and $c2$ are needed, since each stress sample contributes to maximum two stress cycles [11]. Since only stress extrema contribute to stress cycles, the parameters of mean stress ($\hat{\sigma}_{m,c1}(k)$, $\hat{\sigma}_{m,c2}(k)$) and cycle weight ($\hat{w}_{c1}(k)$, $\hat{w}_{c2}(k)$) are nonzero only for those samples within the horizon.

A special but regular case is a cycle where one of the two contributing stress samples is the initial stress sample ($k = 0$) of the horizon. Since the initial stress sample is a measured sample which cannot be modified, fatigue cost of the corresponding cycle can be solely manipulated via the complementary stress sample which lies in the controllable part ($k > 0$) of the horizon. Thus, in the stage cost, the entire fatigue cost contribution of this cycle has to be concentrated at the complementary sample index. This is accounted for by neglecting the splitting-factors $\frac{1}{2}$ of (11) and by conditionally halving the cycle weights $\hat{w}_{c1}(k)$ and $\hat{w}_{c2}(k)$ for (13) instead. For a given sample $k > 0$, a cycle weight is only halved if the complementary sample also lies in the controllable part ($k > 0$) of the horizon. This definition of cycle weights is summarized in Tab. II.

TABLE II: Cycle weight $\hat{w}_{c1}(k)$ or $\hat{w}_{c2}(k)$ at a given stress sample k which contributes to a specific cycle c

Idx of given sample Idx of complementary sample	$k > 0$
$k > 0$	$\frac{\hat{w}_c}{2}$
$k = 0$	$\hat{w}_c$

For the present MPC-implementation, a time-continuous stage cost function is needed. Thus, the summation-type formulation finally is approximated by the integral-type formulation

$$J_{fatigue,PORFC}\,(\sigma, \bar{\boldsymbol{p}}) = \frac{1}{T_{cntrl}} \int_{t_0}^{t_{end}} \big(J_{fatigue,c}(\sigma(t), \hat{\hat{\sigma}}_{m,c1}(t), \hat{\hat{w}}_{c1}(t)) + J_{fatigue,c}(\sigma(t), \hat{\hat{\sigma}}_{m,c2}(t), \hat{\hat{w}}_{c2}(t))\big)\, dt. \tag{14}$$

where the parameters

$$\bar{\boldsymbol{p}} = \begin{pmatrix} \hat{\hat{\sigma}}_{m,c1} \\ \hat{\hat{\sigma}}_{m,c2} \\ \hat{\hat{w}}_{c1} \\ \hat{\hat{w}}_{c2} \end{pmatrix} \tag{15}$$

are defined piecewise constant over the control intervals of the prediction horizon. The factor $\frac{1}{T_{cntrl}}$ is prepended to approximate the discrete sum (13) by the present time-integral.

B. Controller structure and NLP

Above presented method requires the calculation of the parameters $\bar{\boldsymbol{p}}$ at every controller step before the MPC is executed. Since these parameters are a substitute for stress cycle identification which otherwise would have to happen within the MPC, they have to be calculated based on the same stress trajectory like the one which is present within the MPC.

This is achieved by a pre-simulation *PRESIM* of the system model over the prediction horizon which matches the simulation within the MPC, and by a Rainflow-analysis *RFC* of the resultant stress trajectory $\boldsymbol{\sigma}_{sim}$, like shown in Fig. 3.

The entire procedure is given in Algorithm 2. Just like the *MPC*, *PRESIM* is initialized with the states $\boldsymbol{x}_{0,sim} = \boldsymbol{x}_{Plant}$ which are sampled at the *PLANT* via zero-order hold *ZOH*. Since shifting [9] is used within the MPC, the applied control trajectories for *PRESIM* as well are obtained by a *SHIFT* of the control trajectories from the previous MPC evaluation.

The pre-simulation is done in a forward-simulation-fashion, irrespective of whether single- or multiple-shooting is being used in the MPC. On the one hand, this introduces a mismatch of simulation methods in the latter case. On the other hand, the Rainflow-analysis is executed on pre-simulation output which is guaranteed continuous.

The resulting stress trajectory from *PRESIM* is input to the subsequent Rainflow-analysis *RFC*. Its outputs (Tab. I) can be mapped directly to the parameters (15).

After this update of parameters $\bar{\boldsymbol{p}}$, the *MPC* can be executed which is implemented as the following parametric Nonlinear Program. Its objective

$$\min_{\bar{u}} -J_{revenue}(\boldsymbol{x}) + \alpha_{damage} J_{fatigue,PORFC}\,(\sigma, \bar{\boldsymbol{p}}) \tag{16}$$

is maximization of the revenue stage cost $J_{revenue}$ and minimization of the fatigue stage cost $J_{fatigue,PORFC}$ (14) subject to the system dynamics (1), box constraints on the control variables

$$\boldsymbol{u}_{min} \leq \bar{\boldsymbol{u}} \leq \boldsymbol{u}_{max} \tag{17}$$

and nonlinear inequality constraints

$$\boldsymbol{c}_{ineq}(\bar{\boldsymbol{x}}) = \begin{pmatrix} \bar{\boldsymbol{x}} - \boldsymbol{x}_{max} \\ -\bar{\boldsymbol{x}} + \boldsymbol{x}_{min} \end{pmatrix} \leq \boldsymbol{0} \tag{18}$$

on the system states $\bar{\boldsymbol{x}}$ which are sampled at the control intervals.

IV. Tracking Stage Cost with Time-varying References and Weights

A. Formulation

The Economic Stage Cost formulation (14) effectively corresponds to a pointwise penalization of deviation of stress w.r.t. parametric mean stress values $\hat{\hat{\sigma}}_{m,c1}$ and $\hat{\hat{\sigma}}_{m,c2}$. This formulation already is very close to a Tracking MPC

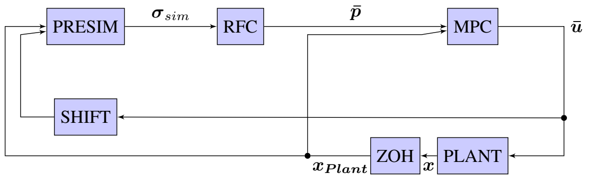


Fig. 3: Flowchart of Parametric Online Rainflow-counting. PRESIM = Pre-simulation over prediction horizon, RFC = Rainflow-counting, MPC = Model Predictive Controller, PLANT = Controller plant, ZOH = Zero-order hold.

Algorithm 2 MPC using Parametric Online Rainflow-counting (PORFC)

```
Input: Periodic measurement of system states, Initial guess of control variables
Output: Periodic update of control variables
while true do
    Receive new initial states from measurement
    PRESIM: Pre-simulate state dynamics by forward simulation
        Using current initial states and SHIFTed guess of control variables
    RFC: Evaluate Rainflow-algorithm and adapt time-variant fatigue cost parameters of MPC
    MPC: Prepare and Solve QP
        Using Multiple-shooting
    Apply updated control variables to PLANT
end
```

formulation. In the following, necessary modifications of the fatigue cost calculation are presented which achieve a reformulation to Tracking Stage Cost. The presented method will be referred to as *Tracking Online Rainflow-counting* (TORFC).

In TORFC, stress mean becomes the tracking reference for the system state of stress. In contrary to PORFC, at each sample, only a scalar stress mean can be defined. Thus, in case of two stress mean values in PORFC, those two values have to be reduced to a scalar value for TORFC. One straightforward and fairly meaningful way of combining them seems to be a definition of the tracking reference

$$\hat{\hat{\sigma}}_{ref} = \frac{J_{fatigue,c1}\ \hat{\hat{\sigma}}_{m,c1} + J_{fatigue,c2}\ \hat{\hat{\sigma}}_{m,c2}}{J_{fatigue,c1} + J_{fatigue,c2}}. \tag{19}$$

by a fatigue-cost-weighted average. Analogously to stress mean, also the cycle weights are combined to a scalar cycle weight of

$$\hat{\hat{w}}_{c,ref} = \frac{J_{fatigue,c1}\ \hat{\hat{w}}_{c1} + J_{fatigue,c2}\ \hat{\hat{w}}_{c2}}{J_{fatigue,c1} + J_{fatigue,c2}}. \tag{20}$$

Equivalently to PORFC (12), the stress amplitude

$$\sigma_a(t) = \left|\sigma(t) - \hat{\hat{\sigma}}_{ref}(t)\right|. \tag{21}$$

can be represented by the stress tracking deviation.

Since mean stress is the reference now, the controllable mean stress effect (4) degenerates to a fixed linear amplification

$$\sigma_{eq} = \sigma_{a,c}\hat{\hat{w}}_m \tag{22}$$

of amplitude stress by the factor

$$\hat{\hat{w}}_m = \frac{R_m}{R_m - \hat{\hat{\sigma}}_{ref}}. \tag{23}$$

In Tracking MPC, stage costs

$$J_{Tracking}\left(\sigma, \hat{\hat{\sigma}}_{ref}, \hat{\hat{w}}_T\right) = \int_{t_0}^{t_{end}} \hat{\hat{w}}_T(t)\ \left(\sigma(t) - \hat{\hat{\sigma}}_{ref}(t)\right)^2 dt. \tag{24}$$

involve a quadratic penalization of deviations from the state reference $\hat{\hat{\sigma}}_{ref}$ which can be weighted time-varying by the tracking weight $\hat{\hat{w}}_T$. Adaptation of the present fatigue cost formulation (5) to this pattern only allows for an exponent of $m = 2$ in the fatigue cost function

$$J_{fatigue,c}(\sigma_{eq}, \hat{\hat{w}}_{c,ref}) = \hat{\hat{w}}_{c,ref}\ a_2\ \sigma_{eq}^2 \tag{25}$$

of individual stress cycles. The coefficient a_2 can be used to adapt this tracking cost function to approximate absolute or derivative values of the original cost function (5) in a defined range of equivalent stress σ_{eq}.

Inserting (21) into (22) into (25) leads to fatigue cost

$$J_{fatigue,c}(\sigma, \hat{\hat{\sigma}}_{ref}, \hat{\hat{w}}_{c,ref}) = \hat{\hat{w}}_{c,ref}\ a_2\ \hat{\hat{w}}_m^2\ \left(\sigma(t) - \hat{\hat{\sigma}}_{ref}(t)\right)^2 \tag{26}$$

for individual stress samples. Considering (14), this leads to a total fatigue cost of

$$J_{fatigue,TORFC}\left(\sigma, \hat{\hat{\sigma}}_{ref}, \hat{\hat{w}}_{c,ref}\right) = \frac{1}{T_{cntrl}} \int_{t_0}^{t_{end}} J_{fatigue,c}(\sigma, \hat{\hat{\sigma}}_{ref}, \hat{\hat{w}}_{c,ref}) dt \tag{27}$$

which finally is reformulated to the standard Tracking Stage Cost formulation

$$J_{fatigue,TORFC}\left(\sigma, \hat{\hat{\sigma}}_{ref}, \hat{\hat{w}}_T\right) = \int_{t_0}^{t_{end}} \hat{\hat{w}}_T(t)\ \left(\sigma(t) - \hat{\hat{\sigma}}_{ref}(t)\right)^2 dt \tag{28}$$

with the tracking weight

$$\hat{\hat{w}}_T(t) = \frac{\hat{\hat{w}}_{c,ref}\ a_2\ \hat{\hat{w}}_m^2}{T_{cntrl}}. \tag{29}$$

B. Controller structure and NLP

TORFC is implemented in MPC similarly to PORFC which is described in Sec. III-B. This includes a Rainflow-analysis based on the output of a pre-simulation of the system states. The only difference lies in the time-varying parameters (15) which in the formulation of TORFC are replaced by the time-varying tracking references (19) and weights (29).

V. Verification & Discussion

The following analysis is presented mainly for the PORFC formulation, due to space limitations. However, in simulative tests, the TORFC formulation exhibits qualitatively comparable behavior to PORFC.

A. Exemplaric system & MPC setup

Above developed methods are tested using a simplified model of the NREL 5 MW wind turbine [12]. Details on this 6-state dynamical model can be found in [5]. This model is used as internal controller model and as plant model. The turbine is assumed to receive perfect foresight of the incoming rotor-equivalent wind velocity from a LiDAR. Wind induces thrust force on the rotor which excites fore-aft oscillations in the tower. These oscillations result in severe cyclic stress and thus in fatigue at the tower root. The turbine exhibits two control variables; generator torque and collective blade pitch angle. Typically, the latter is primarily used to manipulate thrust force and thus to reduce fatigue cost $J_{fatigue}$. However, the controller must balance this with the primary objective of maximizing energy output $J_{revenue}$.

The MPC is implemented in `ACADO Toolkit` using 40 control steps over a horizon length of 4s which leads to a controller sample time of 100ms. Since the concept of *Real-time Iteration* (RTI) [9] is used, only one QP-step is executed per MPC-step. The RTI-steps of the MPC are real-time-feasible in this configuration. Only at the initial MPC-step, a SQP of 50 QPs is executed. It is important to note that in an SQP, for each QP-step the time-varying parameters (15) have to be updated. Thus, at each QP-step, the pre-simulation *PRESIM* and Rainflow-analysis *RFC* have to be executed.

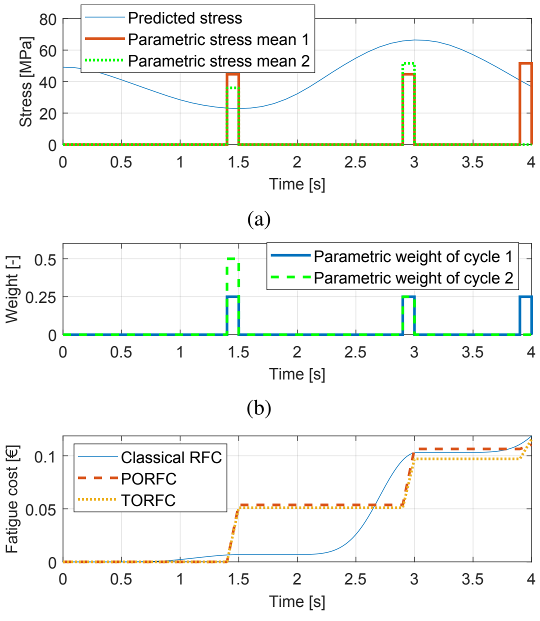
The internal integrator of the MPC as well as the integrators of the pre-simulation and of the plant are set-up with a fixed step-size of 5ms. The external variable of wind velocity only can be input at a rate of 100ms for *PRESIM* and *MPC*, but at a rate of 5ms for the *PLANT*.

B. Verification of fatigue cost calculation

In the following, insights into the calculation of fatigue cost (14) is given exemplarily for the case of PORFC; for TORFC the procedure is similar. The entire process can be traced in Fig. 4. The underlying stress trajectory in Fig. 4a is calculated within the pre-simulation and exhibits 4 local extrema: 2 minima and 2 maxima. The Rainflow-algorithm is applied to this stress trajectory, and its outputs are used to create two trajectories of stress mean values (Fig. 4a) and two trajectories of cycle weights (Fig. 4b), like described in Sec. III-B. Those variables can be nonzero at controllable stress extrema in the horizon ($t > t_0$). Both representations of each variable (mean & weight) are nonzero for samples which contribute to two stress cycles.

The accumulated PORFC fatigue cost within the prediction horizon is shown in Fig. 4c. Its trajectory exhibits ramps at each sample where the parameters and thus the fatigue stage cost are nonzero.

Additionally, Fig. 4c enables analysis of further properties of the herein defined MPC fatigue cost functions. Here, the P/TORFC fatigue cost functions are compared to an a posteriori fatigue cost evaluation method based on classic Rainflow-counting on a moving window [13]. The latter algorithm is evaluated at each point of the very fine simulation time grid. The plot of its output fatigue cost results in the blue trajectory. It can be seen that fatigue cost increases gradually in time with growing stress cycles and finally leads to a similar end value like PORFC and TORFC. However, obviously the fatigue cost values within the horizon differ drastically. The reason lies in the design of the cost functions for PORFC and TORFC, where cost of a stress cycle is allocated back in time to the stress sample which starts the cycle. For an example, the most damaging half cycle can be studied which occurs at $1.5\text{s} < t < 3\text{s}$. Half of its fatigue cost already causes a ramp in P/TORFC fatigue cost at $t = 1.5\text{s}$. The second half of its fatigue cost is visible at $t = 3\text{s}$. In general words, information about the future *stress* state influences an earlier *fatigue* state. Thus, from a system-theoretical point of view, these parametric cost functions show acausal behavior.



(a)

(b)

(c)

Fig. 4: Calculation of fatigue cost for PORFC based on *a)* predicted stress trajectory via parametric stress mean values and *b)* cycle weight. *c)* Comparison of resulting PORFC fatigue cost evolution to the ones of TORFC and of a classical moving-window Rainflow-analysis.

C. Verification of closed-loop dynamic behavior

Fig. 5 shows the behavior of stress over a simulation time of 30 seconds using an MPC with PORFC in closed loop with the plant. The effectiveness of the presented formulation is visible via the reduction of stress amplitude in comparison to the initially predicted trajectories. Since

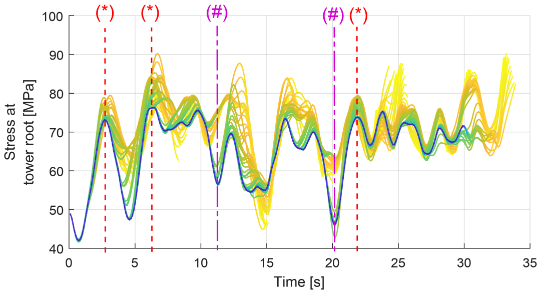


Fig. 5: Predicted stress at tower bottom for different successive MPC-steps over a simulation time of 30 seconds. Each predicted trajectory has an horizon length of 4 seconds and is colored from blue (beginning of horizon) to yellow (end of horizon). Vertical lines marked by (*) where reduction of stress amplitude and (#) where jumping-behavior occurs.

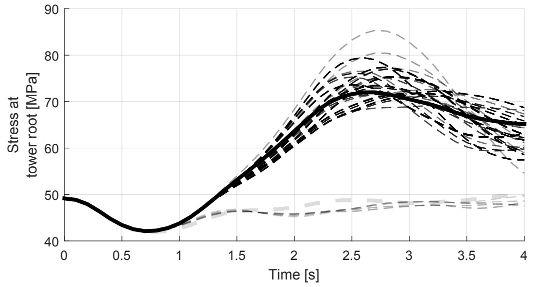


Fig. 6: Predicted stress at tower bottom for 50 different QP-steps of the initial SQP at the beginning of simulation. Light-grey bold-dashed line = first QP; varying-grey thin-dashed lines = further QPs, where darker color = later in SQP-iteration; black bold-solid line = last QP.

fatigue cost (5) increases superlinearly with stress amplitude, these reductions especially occur at big cycles like at $t = 3\text{s}$, $t = 7\text{s}$ and $t = 22\text{s}$ which are marked by *(*)*.

In contrary, at $t = 11\text{s}$ and $t = 20\text{s}$ which are marked by *(#)*, the opposite seems to occur. Here, the finally implemented stress amplitude is much higher than the initially predicted ones. This behavior is suspected to be due to a jump into new local minima for the subsequent optimization problems. Additionally to these observations in the time-marching *Real-time Iteration* context, similar behavior can be detected at a fixed-time SQP in Fig. 6. Here, the solution trajectory first converges towards a low-oscillatory trajectory but after 6 QPs jumps to a trajectory with a pronounced maximum of 85MPa at $t = 2.7\text{s}$. Starting from there, the optimization converges towards the bolt-solid trajectory with a more moderate cycle depth. Further studies are necessary to obtain a clear answer on this jumping into new minima.

D. Comparative discussion w.r.t. the underlying DORFC formulation

The reformulations of PORFC and TORFC introduce further approximations w.r.t. to the formulation of DORFC. Tab. III provides an overview of key quantities which appear in the individual fatigue cost terms of DORFC (5), PORFC (14) and TORFC (28). A main difference of PORFC w.r.t. DORFC lies in neglecting sample indices $k_{max,c}$ and $k_{min,c}$. This means that the MPC does not receive information about the *position* of the complementary stress sample. However, the MPC does not require this information for fatigue cost calculation since information about the *value* of the complementary stress sample is contained in the parametric mean stress $\hat{\bar{\sigma}}_{m,c1/2}$. A main difference of TORFC w.r.t. to DORFC and PORFC is the restriction to a quadratic penalization of stress amplitudes by $m = 2$.

TABLE III: Overview of key quantities in the respective fatigue cost terms and their treatment as variable (”V”) or fixed (”F”) during one optimization step, or constant (”C”) throughout operation.

Quantity	DORFC	PORFC	TORFC
Stress value 1	$\sigma(\hat{k}_{max,c})$ (V)	$\sigma(t)$ (V)	$\sigma(t)$ (V)
Stress value 2	$\sigma(\hat{k}_{min,c})$ (V)	$\hat{\bar{\sigma}}_{m,c1/2}(t)$ (F)	$\hat{\bar{\sigma}}_{ref}(t)$ (F)
Sample index 1	$\hat{k}_{max,c}$ (F)	-	-
Sample index 2	$\hat{k}_{min,c}$ (F)	-	-
Weight	$\hat{w}_c\, a_m$ (F)	$\hat{\bar{w}}_{c1/2}(t)\, a_m$ (F)	$\hat{\bar{w}}_T(t)$ (F)
Exponent	$m \geq 2$ (C)	$m \geq 2$ (C)	$m = 2$ (C)

In terms of total fatigue cost values for the entire prediction horizon, the approximations of the PORFC formulation do not introduce significant mismatch w.r.t. DORFC. Minor mismatch can be expected since in DORFC stress amplitudes are determined based on stress sampled at the coarse control grid while in PORFC stress amplitudes are determined based on numerical integration over control intervals which are ”activated” by the parameters (15).

In the calculation of fatigue cost gradients and Hessian, higher mismatch is suspected. Like indicated in Sec. III-A, the gradient of fatigue cost w.r.t. mean stress does not affect the optimization problem anymore since mean stress is a parameter in PORFC. Additionally, since the fatigue cost terms are decoupled in time, all off-diagonal terms of the second derivative $\frac{\partial^2 J_{fatigue,c1/2}}{\partial \sigma(k)^2}$ of fatigue cost w.r.t. separate stress samples are zero. However, further theoretical and numerical investigation is needed here.

VI. Conclusion & Outlook

Reformulation of the non-standard fatigue cost of Direct Online Rainflow-counting (DORFC) to an Economic and a Tracking Stage Cost is a significant advancement in terms of practical applicability and of theoretical interpretability of Online Rainflow-counting in MPC. Applicability within standard MPC frameworks has been shown via an implementation in `ACADO Toolkit`. It has been shown that the necessary Rainflow-analysis at each MPC-step can be externalized based on a pre-simulation which matches the internal simulation of the MPC. The results of this externalized Rainflow-analysis are inserted into the optimization problem via time-varying parameters (Economic), or time-varying tracking references and weights (Tracking).

It has been demonstrated that the accumulated stage costs compare very well to an a posteriori Rainflow-analysis, exhibit ramps at samples which contribute to a cycle, and exhibit noteworthy acausal behavior. Furthermore, a closed-loop simulation has shown the successful mitigation of highly damaging stress cycles. For certain successive MPC-steps, jumps to different prediction trajectories have been observed which are suspected to be caused by jumps into new local minima in the optimization problem. Especially this property will be investigated further in the future.

Additional work will be directed to:

- Including stress information from past measurements and far-future predictions in order to account for long-term transition cycles
- Further formalization of Online Rainflow-counting in MPC
- Comparison to fatigue cost approximations by Hysteresis Operators [14] and Artificial Neural Networks [15]
- Research on uncertainty and robustness
- Simultaneous consideration of fatigue cost for multiple parts of the plant
- Realistic controller tests using high-fidelity plant simulators
- Online-adaptation of horizon and control step size based on current error of fatigue cost estimation.

References

[1] R. Findeisen and F. Allgöwer, “An introduction to nonlinear model predictive control,” in *21st Benelux Meeting on Systems and Control*, 2002, pp. 1–23.

[2] M. A. Henson, “Nonlinear model predictive control: Current status and future directions,” *Computers & Chemical Engineering*, vol. 23, no. 2, pp. 187–202, 1998.

[3] ASTM, “Standard practices for cycle counting in fatigue analysis,” West Conshohocken, PA, 1985.

[4] The MathWorks Inc., “Rainflow counts for fatigue analysis,” 2018.

[5] S. Loew, D. Obradovic, and C. L. Bottasso, “Direct online rainflow-counting and indirect fatigue penalization methods for model predictive control,” in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3371–3376.

[6] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms*, 2nd ed., ser. Communications and Control Engineering. Springer International Publishing, 2017.

[7] J. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Pub., 2009.

[8] B. Houska, H. J. Ferreau, and M. Diehl, “Acado toolkit—an open–source framework for automatic control and dynamic optimization,” *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[9] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, “From linear to nonlinear mpc: Bridging the gap via the real-time iteration,” *International Journal of Control*, pp. 1–19, 2016.

[10] E. Haibach, *Betriebsfestigkeit: Verfahren und Daten zur Bauteilberechnung*, 3rd ed., ser. VDI-Buch. Berlin: Springer, 2006.

[11] Y. Shi, B. Xu, Y. Tan, and B. Zhang, “A convex cycle-based degradation model for battery energy storage planning and operation.”

[12] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, *Definition of a 5-MW Reference Wind Turbine for Offshore System Development: Technical Report NREL/TP-500-38060*. National Renewable Energy Laboratory, 2009.

[13] C. Heinrich, M. Khalil, K. Martynov, and U. WEVER, “Online remaining lifetime estimation for structures,” *Mechanical Systems and Signal Processing*, vol. 119, pp. 312–327, 2019.

[14] J. B. Berglind, R. Wisniewski, and M. Soltani, “Fatigue load modeling and control for wind turbines based on hysteresis operators,” in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 3721–3727.

[15] J. Luna, O. Falkenberg, S. Gros, and A. Schild, “Wind turbine fatigue reduction based on economic-tracking nmpc with direct ann fatigue estimation,” *Renewable Energy*, vol. 147, pp. 1632–1641, 2020.

3.5 Paper 5: Generalization & incorporation of fatigue into the MPC- internal system model

3.5.1 Summary

In the DORFC formulation of **Papers 2** and **3** fatigue estimation is incorporated as an intermediate step of the MPC algorithm. In the present paper, this incorporation is even intensified by formulating fatigue estimation as part of the MPC-internal system model.

As an open-loop basis, an existing moving-window value-discrete fatigue estimation algorithm is enhanced to a one-step value-continuous algorithm. Key enablers are the perpetual update of a *residue* set, and the consideration of full and half cycles. The residue set plays a central role as "condensed memory" by carrying along past stress extrema which have not contributed to full cycles as yet.

On top of this algorithm, the MPC-internal fatigue estimation model requires a distinction between time instances before execution of the current MPC step *(past)* and time instances in the prediction horizon *(prediction)*. The reason behind is that only the predicted states are controllable, and that they are deleted and re-simulated at each new MPC step. As system class, a hybrid dynamical system is chosen since it can represent physical value-continuous dynamics and algorithmic discontinuous mappings. The submodels are:

- The plant dynamics is provided by a set of value-continuous differential equations.
- The stress cycle identification (Rainflow algorithm) is represented by a transition map which updates the value-discrete *structural states* (cycle weights and sample indices of stress extrema).
- The past- and prediction-residues are updated by separate reset maps and held constant in the meantime. The reset maps are defined by a logic of appending and deleting samples which is governed by the structural states.
- The fatigue cost of past full cycles as well is updated by a reset map and held constant in the meantime. The reset map is defined by continuous fatigue cost functions, and by a logic governed by the structural states.
- The total fatigue cost is provided by continuous output functions which as well are adapted by a logic.

The gradients of total fatigue cost w.r.t. the control variables are formulated. This is simplified by the fact that past states are independent from the control variables, and by the assumption "Switch seldom" from **Paper 3**. Finally, the hybrid dynamical system is implemented in the cost function of an Economic Nonlinear MPC where it provides accurate fatigue estimation by efficiently memorizing past stress samples. By deriving previous fatigue cost formulations from the present hybrid dynamical system, the impressive generality of this approach is proven.

3.5.2 Contribution

The author of this dissertation developed the one-step value-continuous fatigue estimation algorithm and the hybrid dynamical systems, performed the implementation, executed and analyzed the simulations, and wrote the paper. DO put the formulation in the right context and supervised the work. Both authors jointly developed the idea for the generalization of fatigue estimation, and provided important input to this research work through discussions, feedback and by improving the manuscript.

3.5.3 Reference

S. Loew and D. Obradovic, “Formulation of fatigue dynamics as hybrid dynamical system for model predictive control,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6616–6623, 2020. doi:10.1016/j.ifacol.2020.12.080

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 53-2 (2020) 6616–6623

Formulation of Fatigue Dynamics as Hybrid Dynamical System for Model Predictive Control

Stefan Loew * **Dragan Obradovic** **

* *Wind Energy Institute of Technical University of Munich, Boltzmannstrasse 15, 85748 Garching, Germany, and Siemens AG, Corporate Technology, Otto-Hahn-Ring 6, 81739 Munich, Germany(e-mail: loew.stefan@siemens.com).*
** *Siemens AG, Corporate Technology, Otto-Hahn-Ring 6, 81739 Munich, Germany.*

Abstract: The standard fatigue estimation procedure is formulated as a hybrid dynamical system, which subsequently is utilized to calculate an economic terminal cost in MPC. This formulation is enabled by the development of a novel algorithm for continuous stress cycle identification. A second hybrid dynamical system is designed to provide fatigue cost gradients. The formulation turns out to be a powerful generalization of previous fatigue cost formulations, and additionally introduces consideration of past stress into the cost function. Presented closed-loop simulations using a wind turbine model provide insight into the subsystems of the hybrid dynamical system, and show the benefit of memorizing the past.

Keywords: Optimal Control, Hybrid Systems, Mechatronic Systems, Energy Systems

1. INTRODUCTION

Fatigue is damage of a material caused by cyclic application of mechanical stress. Fatigue has large impact on the operating costs of devices and thus control of fatigue is used to increase the total economic profit. Model Predictive Controllers (MPC) enable optimal control of many devices by using predictions of future system excitation (Findeisen and Allgöwer (2002)). Based on these input predictions, stress time-series at crucial spots in the device structure can be predicted. In *Direct Online Rainflow-counting* (DORFC, Loew et al. (2019)), these stress predictions are used for direct incorporation of the standard fatigue damage estimation procedure in the cost function of Model Predictive Control.

Questions:
Question 1) Since within the formulation of DORFC, fatigue cost is a discontinuous function of all time-samples of stress within the prediction horizon, neither the concept of stage cost nor of terminal cost applies (Grüne and Pannek (2017)). However, formal proofs of stability and recursive feasibility usually are commonly researched for those standard concepts (Grüne and Pannek (2017), Findeisen and Allgöwer (2002), Rawlings and Mayne (2009) p.112 ff.).
In Loew et al. (2020), stage cost formulations are achieved by some approximations, externalization of the fatigue cost evaluation from the MPC algorithm, and insertion of its results into the MPC via time-varying parameters. However, the question arises, if *direct stage or terminal cost formulations* without those additions are possible as well.

Question 2) In preceding methods (Loew et al. (2019), Loew et al. (2020)), fatigue cost is evaluated only based on the states in the prediction horizon. However, fatigue is a long-term effect, and correct evaluation requires knowledge about the entire stress history. As an extreme example, the very first stress sample after commissioning of the machine can form a stress cycle with the current stress sample several years later. Thus, the question arises, if *information about the stress history* can be included consistently into the predictive cost function.

Question 3) Taking up the previous question, storing the entire stress history requires a high amount of computational memory in the order of Gigabytes. Analyzing this high amount of data at every controller step requires excessive computing power. Thus, a smart method is required which enables *pertaining a reduced dataset* in the controller without losing information.

Question 4) Other approaches for fatigue control in literature use Markovian and causal surrogate models for fatigue dynamics (Luna et al. (2020), Gros and Schild (2017), Barradas-Berglind et al. (2015)). However, analysis in Loew et al. (2020) shows that cost functions are closer to the nature of the fatigue estimation process if stress state values are penalized based on their past *and* future evolution. In Loew et al. (2020), this property is implemented by above mentioned externalization of fatigue evaluation. However, the question arises, if this property also can be achieved by a standard cost formulation with *Markovian and causal properties.*

Contribution & Outline: In Sec. 2, *Questions 2)* and *3)* are solved by the development of a value-continuous fatigue cost calculation with residue handling. In Sec. 3,

10.1016/j.ifacol.2020.12.080

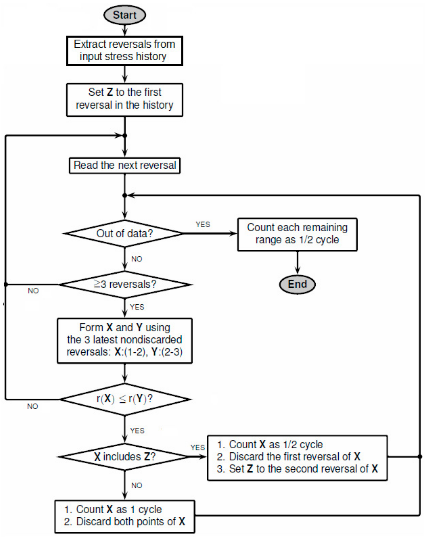

Start
Extract reversals from input stress history
Set Z to the first reversal in the history
Read the next reversal
Out of data?
YES
Count each remaining range as 1/2 cycle
End
NO
≥3 reversals?
NO
YES
Form X and Y using the 3 latest nondiscarded reversals: X:(1-2), Y:(2-3)
r(X) ≤ r(Y)?
NO
YES
X includes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

Fig. 1. Flowchart of the MATLAB-implementation `rainflow()` of the *Three-point Rainflow algorithm* (simplified from The MathWorks Inc. (2018)). 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.

fundamentals on delayed and hybrid systems are presented as a basis for further modeling. In Sec. 4, fatigue cost calculation is cast into a hybrid dynamical system which solves *Question 4)*. In Sec. 5, a link to previous formulations and an incorporation of the hybrid dynamical system into the terminal cost of an MPC is provided, which solves *Question 1)*. Sec. 6 provides first insight into this novel MPC implementation. Sec. 7 provides conclusion and outlook.

One comment on notation: The variable notation with bar $\bar{a}$ means sampled on the control intervals of the prediction horizon and with tilde $\tilde{a}$ means estimated from measurements.

2. FATIGUE ESTIMATION

2.1 Cycle identification and fatigue estimation

Rainflow-counting (RFC) (ASTM (1985), The MathWorks Inc. (2018)) is the standard method for decomposition of stress time-series to stress cycles and therefore is part of the standard fatigue estimation process. A flowchart of the Rainflow algorithm is displayed in Fig. 1. The Rainflow algorithm is based on reversals (extrema) of the stress trajectory, and contains algorithmic branches and loops. Thus, a crucial property of the Rainflow algorithm is its discontinuous output-behavior. Furthermore, the number N_c of identified cycles is variable, but bounded by the number of extrema. The outputs of RFC can be converted to the variables in Table 1 where the value of weight equals $w_c = 1$ if a full stress cycle is detected and equals $w_c = 0.5$ if a half cycle (half period) is detected.

Table 1. Converted outputs of RFC for stress cycles c

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

2.2 Moving-window cycle identification & Residue

Originally, Rainflow analysis is performed on the entire stress history. However, in Heinrich et al. (2019) it is shown that Rainflow analysis also can be performed on a moving window, like stated in Alg. 1.

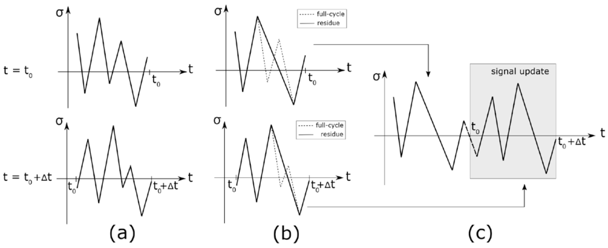


Fig. 2. Rainflow analysis on a moving window (Heinrich et al. (2019)).

In *Question 2)* of Sec. 1, the importance of considering the stress history in fatigue evaluation is pointed out. This is due to so called *transition cycles* which grow over a long period of time and can reach high stress amplitudes with dominating fatigue impact (Marsh et al. (2016)). Since transition cycles per definition have not been closed yet, they appear as half cycles in Rainflow analysis. Thus, to account for transition cycles in the moving-window algorithm, the corresponding half-cycle stress samples are carried along in the so-called *residue* (Köhler et al. (2012)).

Algorithm 1 Rainflow-analysis on a moving window; according to Heinrich et al. (2019)

```
Input: Existing stress string $\boldsymbol{\sigma}_{exist}$, Periodic update of new stress string $\boldsymbol{\sigma}_{new}$
Output: Full cycles
1: Extract full cycles from existing string ($t \leq t_0$) using Rainflow algorithm, Store residue in $\boldsymbol{x}_{res,1}$
   while true do
2:     Receive string $\boldsymbol{\sigma}_{new}$ of new stress samples obtained on $(t_0, t_0 + \Delta t]$
3:     Extract full cycles from $\boldsymbol{\sigma}_{new}$ using Rainflow algorithm, Store residue in $\boldsymbol{x}_{res,2}$
4:     Merge residues $\boldsymbol{x}_{res,1}$ and $\boldsymbol{x}_{res,2}$
5:     Extract full cycles from $\{\boldsymbol{x}_{res,1}, \boldsymbol{x}_{res,2}\}$ using Rainflow algorithm, Store residue in $\boldsymbol{x}_{res,1}$
   end
```

As an illustration, the Rainflow analysis on a moving window is shown in Fig. 2. At step *a)*, full cycles are extracted from an existing stress string which occurs at $t \leq t_0$. Additionally, a string of new stress samples ($t_0 < t \leq t_0 + \Delta t$) is obtained and full cycles are extracted from it as well. Step *b)* shows extracted full cycles and residues of both. Step *c)* shows the string which results from concatenation of both residues. Consequently, this string as well is analyzed for full cycles.

Depending on the stress signal, a high number of samples can be accumulated in the residue. Highest possible dimensions of the residue vector result from diverging and converging stress time series because they result in a very high number of half cycles (Köhler et al. (2012)). However, long-term diverging series are unrealistic because unstable machine behavior typically is counteracted by the controller or an emergency shutdown. Long-term converging series are irrelevant, since very low-amplitude cycles can be neglected without significant error in fatigue estimation. Concluding, the dimension N_{res} of the residue vector is finite and in practical tests remained well below 100.

2.3 One-step value-continuous cycle identification

The moving-window approach of Sec. 2.2 results in periodic updates of identified full cycles. This corresponds to a periodic value-discrete update of fatigue cost. However, gradient-based optimization in MPC requires value-*continuous* evolution of cost functions. Therefore, also the dynamics of fatigue cost needs to be expresses in a continuous fashion. Additionally, just like the plant states, fatigue cost should evolve on arbitrarily small timesteps unlike time periods Δt of Alg. 1. Both goals are achieved by the following two novel steps of enhancement:

As the first step, Alg. 1 is adapted to a one-step cycle identification which is shown in Alg. 2. The advantage is that this algorithm does not pause until a new string of stress samples is available, but directly provides an update of full cycles with each new stress sample.

Algorithm 2 One-step Rainflow-analysis

```
**Input:** Existing stress string $\boldsymbol{\sigma}_{exist}$, Periodic update of a scalar new stress sample $\sigma$
**Output:** Full cycles
**Initialization:** Stress residue $\boldsymbol{x}_{res} = \boldsymbol{\sigma}_{exist}$
**while** true **do**
    Merge residue $\boldsymbol{x}_{res}$ and new stress sample $\sigma$
    Extract full cycles from $\{\boldsymbol{x}_{res}, \sigma\}$ using Rainflow algorithm, Store residue in $\boldsymbol{x}_{res}$
**end**
```

As the second step, Alg. 2 is enhanced by additional consideration of half cycles which start "growing" already due to infitesimally small variations in stress. This consideration results in continuous output of the computation of fatigue cost which additionally is added to the algorithm, like stated in Alg. 3.

These enhancements do not introduce further assumptions. Thus, like shown in Fig. 3, this continuous fatigue cost calculation provides the same output like a batch evaluation over the entire stress trajectory. As expected, the fatigue cost rate never is negative, and thus fatigue cost is monotonously increasing. Concluding, Alg. 3 also solves *Question 3)* of Sec. 1 by condensing information of past full cycles in the scalar $x_{fatigue,FC}$, and by only memorizing stress samples which are extrema and have not contributed to full cycles yet.

3. SYSTEM CLASSES

In order to integrate the continuous fatigue cost estimation of Alg. 3 in an MPC, it has to be cast into a class of dynamical systems. A system class is required which allows for incorporation of past states, for dynamic variation of state dimensions, and for discontinuous updates of states. In the following, system classes from literature are assessed which seem promising with regard to these properties.

Algorithm 3 One-step value-continuous fatigue cost estimation

```
**Input:** Existing stress string $\boldsymbol{\sigma}_{exist}$, Periodic update of a scalar new stress sample $\sigma(k+1)$ at each step $k$
**Output:** Periodic update of fatigue cost $J_{fatigue}(k)$ at each step $k$
**Initialization:** Zero total fatigue cost $J_{fatigue}(0) = 0$ and fatigue cost $x_{fatigue,FC}(0) = 0$ of full cycles, Stress residue $\boldsymbol{x}_{res}(0) = \boldsymbol{\sigma}_{exist}$, $k = 0$
**while** true **do**
    Merge residue $\boldsymbol{x}_{res}(k)$ and new stress sample $\sigma(k+1)$
    Extract full and half cycles from $\{\boldsymbol{x}_{res}(k), \sigma(k+1)\}$ using Rainflow algorithm, Store residue in $\boldsymbol{x}_{res}(k+1)$
    Calculate fatigue cost based on full cycles, Add result to $x_{fatigue,FC}(k)$ to obtain $x_{fatigue,FC}(k+1)$
    Calculate fatigue cost based on full and half cycles, Add result to $x_{fatigue,FC}(k)$ to obtain $J_{fatigue}(k+1)$
    $k = k + 1$
**end**
```

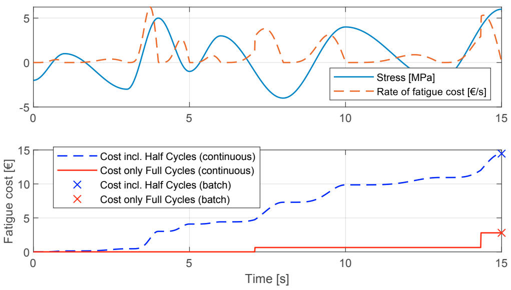


Fig. 3. Upper: Exemplary input stress trajectory, and fatigue cost rate obtained by Finite Differences from continuous fatigue cost including half cycles. Lower: continuous (one-step) and batch estimation using only full cycles or full+half cycles.

3.1 Delay differential equations

Delay differential equations (DDEs) are differential equations in which the state differential at a certain time depends on past state values (see Shampine and Thompson (2009)). However, discontinuous updates of states and their dimensions do not seem to be possible. Additionally, delays would have to increase with simulation time since the instances of residue-sampling are fixed in *absolute* time. Therefore, DDEs are *not utilized* in the present work.

3.2 Hybrid dynamical systems

Definition: A hybrid dynamical system is characterized by a coexistence of value-continuous and value-discrete dynamics (Aihara and Suzuki (2010)). They are able to describe physical mechanisms like impact and friction, and cyber-physical mechanisms like switching. Therefore for the present work, hybrid systems are highly suitable

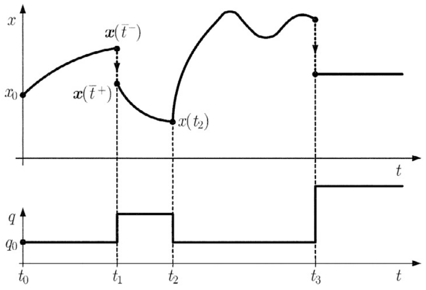


Fig. 4. Switching of dynamics at t_2 and resets of the value-continuous state x at t_1 and t_3, both triggered by the value-discrete state q (Lunze and Lamnabhi-Lagarrigue (2009)).

for modeling the interplay of physical value-continuous dynamics and algorithmic discontinuous mappings.

According to Aihara and Suzuki (2010) and Lunze and Lamnabhi-Lagarrigue (2009), hybrid dynamical systems can be described by the mappings

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}_{\boldsymbol{q}(t)}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{1a}$$
$$\boldsymbol{q}(t) = \boldsymbol{G}(\boldsymbol{q}(t^-), \boldsymbol{x}(t^-), \boldsymbol{u}(t)) \tag{1b}$$
$$\boldsymbol{x}(t) = \boldsymbol{R}(\boldsymbol{q}(t^-), \boldsymbol{x}(t^-), \boldsymbol{u}(t)) \tag{1c}$$
$$\boldsymbol{y}(t) = \boldsymbol{H}(\boldsymbol{q}(t), \boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{1d}$$

where

- $\boldsymbol{x}(t) \in \mathbb{R}^{N_x}$ are value-continuous states.
- $\boldsymbol{u}(t) \in \mathbb{R}^{N_{in}}$ are control variables.
- $\boldsymbol{q}(t) \in \mathbb{R}^{N_q}$ are value-discrete states.
- $t^- = \lim_{\epsilon \to 0} \tau - \epsilon$ denotes the time instant right before a discontinuous transition/reset event τ of states. t equivalently denotes the time instant after such an event.
- $\boldsymbol{y}(t) \in \mathbb{R}^{N_y}$ is the output of the system.
- $\boldsymbol{F}_{\boldsymbol{q}(t)}$ is a switched set of continuous differential equations. Which subsystem is active, is determined by the value-discrete states $\boldsymbol{q}(t)$.
- $\boldsymbol{G}$ is a discontinuous transition-map for the value-discrete states from $\boldsymbol{q}(t^-)$ to $\boldsymbol{q}(t)$.
- $\boldsymbol{R}$ is a discontinuous reset-map for jumps of the value-continuous states from $\boldsymbol{x}(t^-)$ to $\boldsymbol{x}(t)$.
- $\boldsymbol{H}$ is an output function.

An exemplary time-behavior of a simple hybrid dynamical system is shown in Fig. 4.

4. FATIGUE DYNAMICS AS HYBRID DYNAMICAL SYSTEM

4.1 Formulation of fatigue dynamics

In the following, plant dynamics and fatigue evolution of Alg. 3 will be cast into the framework of an hybrid dynamical system which was introduced in Sec. 3.2. Since the ultimate goal is utlization of this hybrid dynamical system in an MPC cost function, there will be a distinction between time instances in the controllable prediction horizon (*prediction*) and time instances before execution of the current MPC-step (*past*). The resulting subsystems stem from plant dynamics, extrema & cycle identification, update of residue, update of fatigue cost of past full cycles, and output of total accumulated fatigue cost. This is visualized in Fig. 5. Interestingly, all types of mappings of hybrid dynamical systems are utilized; namely several continuous differential equations, a transition-map, three reset maps, and one output function. Additionally, there are value-discrete and value-continuous state sets of variable dimensions.

Plant: The time-continuous evolution of value-continuous plant states $\boldsymbol{x}(t)$ is obtained by the set

$$\dot{\boldsymbol{x}} = \boldsymbol{F}_{sys}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t)) \tag{2}$$

of differential equations with control variables $\boldsymbol{u}(t)$, disturbances $\boldsymbol{d}(t)$ and initial states $\boldsymbol{x}(t_0) = \tilde{\boldsymbol{x}}(t_0)$. Fatigue is caused by stress $\sigma(t) \in \boldsymbol{x}(t)$ which is considered as one of the states of the system.

Cycle identification: According to Alg. 3, cycle identification by the Rainflow algorithm is based on a stress string which is a concatenation of the residual stress samples and the new stress sample. This string is input to the transition-map

$$\boldsymbol{q}_{struct}(t) = \boldsymbol{G}_{RFC}(\{\boldsymbol{x}_{res}(t), \sigma(t)\}) \tag{3}$$

which instantaneously updates the value-discrete structural states. The structural states are defined by

$$\boldsymbol{q}_{struct} = \begin{pmatrix} \boldsymbol{w} \\ \boldsymbol{k}_{max} \\ \boldsymbol{k}_{min} \end{pmatrix} = \begin{pmatrix} (w_1, w_2, ..., w_{N_c}) \\ (k_{max,1}, k_{max,2}, ..., k_{max,N_c}) \\ (k_{min,1}, k_{min,2}, ..., k_{min,N_c}) \end{pmatrix} \tag{4}$$

where the number N_c of identified cycles - and thus the dimensions - are variable. The structural states are cycle weights $\boldsymbol{w}$, sample indices $\boldsymbol{k}_{max}$ of cycle maxima and sample indices $\boldsymbol{k}_{min}$ of minima (compare to Tab. 1).

Residue: The update of the value-continuous residue states is obtained by the reset-map

$$\boldsymbol{x}_{res}(t) = \boldsymbol{R}_{res}(\boldsymbol{q}_{struct}(t^-), \boldsymbol{x}_{res}(t^-), \sigma(t^-), t) \tag{5}$$

with the residue states

$$\boldsymbol{x}_{res} = (\{\boldsymbol{x}_{res,past}, \boldsymbol{x}_{res,pred}\}) = \begin{pmatrix} (\sigma_{res,past,1}, \sigma_{res,past,2}, ..., \sigma_{res,past,N_{past}}) \\ (\sigma_{res,pred,1}, \sigma_{res,pred,2}, ..., \sigma_{res,pred,N_{pred}}) \end{pmatrix}^T \tag{6}$$

where the numbers of past N_{past} and predicted N_{pred} stress samples - and thus the dimensions - are variable. Since the reset-map depends on the structural states $\boldsymbol{q}_{struct}$, it is event-driven. Additionally, the reset-map is time-driven since the reset-behavior differs for the beginning $t = t_0$ and within $t_0 < t \leq t_{end}$ the horizon.

The reset-map comprises two sub-maps $\boldsymbol{R}_{res,past}$ and $\boldsymbol{R}_{res,pred}$ for treatment of past and predicted residue, respectively. Fig. 6 shows the sampling of past and prediction trajectory via reset-maps to obtain the residue states.

The update of the past-residue is obtained by the reset-map

$$\boldsymbol{x}_{res,past} = \boldsymbol{R}_{res,past} = \begin{cases} \{\sigma_{res,past,l}(t^-), \sigma(t^-)\} \quad \forall l \in \{k_{max,j}, k_{min,j}\} \\ \quad \forall j | \neg(w_j = 1 \wedge k_{max,j} \leq N_{past} \wedge k_{min,j} \leq N_{past}) \\ \quad \textbf{if } t = t_0 \\ \text{not active} \quad \textbf{if } t_0 < t \leq t_{end} \end{cases} \tag{7}$$

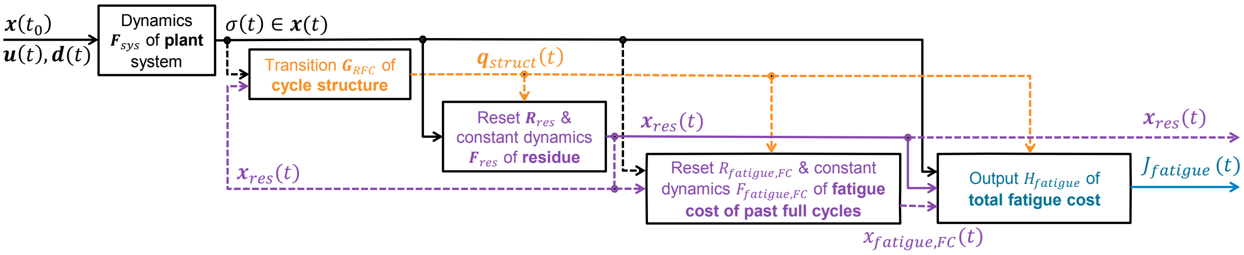

$x(t_0)$
$u(t), d(t)$
Dynamics F_{sys} of plant system
$\sigma(t) \in x(t)$
Transition G_{RFC} of cycle structure
$q_{struct}(t)$
Reset R_{res} & constant dynamics F_{res} of residue
$x_{res}(t)$
$x_{res}(t)$
$x_{res}(t)$
Reset $R_{fatigue,FC}$ & constant dynamics $F_{fatigue,FC}$ of fatigue cost of past full cycles
Output $H_{fatigue}$ of total fatigue cost
$J_{fatigue}(t)$
$x_{fatigue,FC}(t)$

Fig. 5. Structure of the hybrid dynamical system for plant dynamics and fatigue cost evaluation. Black = value-continuous dynamics, Orange = transition-map, Purple = reset-map, Blue = output function. Dashed lines = paths which are neglected at fatigue cost gradient calculation.

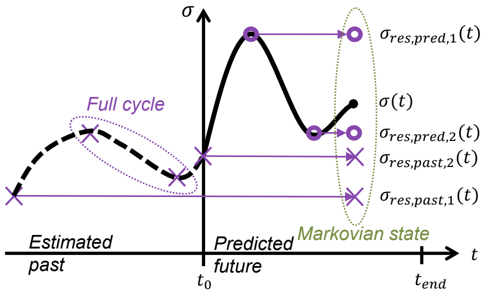


Fig. 6. Generation of past-residue $\boldsymbol{x}_{res,past}$ and prediction-residue $\boldsymbol{x}_{res,pred}$ from estimated past and predicted future stress evolution. Both past extrema which form a full cycle are excluded from the residue and their fatigue cost is added to the fatigue cost $x_{fatigue,FC}$ of past full cycles.

where the new estimated stress sample $\sigma(t^-)$ is appended to the previous past-residue $\sigma_{res,past,l}(t^-)$ at the beginning $t = t_0$ of the prediction horizon. From the previous past-residue, only samples are maintained which are extrema and thus contribute to stress cycles (index $l \in \{k_{max,j}, k_{min,j}\}$). This is further limited to extrema which do *not* contribute to full cycles ($w_j = 1$) which entirely occur in the past-residue ($k_{max,j} \leq N_{past} \wedge k_{min,j} \leq N_{past}$).

The update of the prediction-residue is obtained by the reset-map

$$\boldsymbol{x}_{res,pred} = \boldsymbol{R}_{res,pred} = \begin{cases} \emptyset \quad \textbf{if } t = t_0 \\ \{\sigma_{res,pred,l}(t^-), \sigma(t^-)\} \quad \forall l + N_{past} \in \{\boldsymbol{k}_{max}, \boldsymbol{k}_{min}\} \\ \quad \textbf{if } t_0 < t \leq t_{end} \end{cases} \tag{8}$$

where the new estimated stress sample is appended to the previous prediction-residue within the horizon $t_0 < t \leq t_{end}$. From the previous prediction-residue, all samples are maintained which contribute to stress cycles. At the beginning $t = t_0$ of a new MPC-step, the prediction-residue is initialized to an empty set.

In between of those resets, the residue states are held constant; which is represented by the continuous dynamics

$$\dot{\boldsymbol{x}}_{res}(t) = \boldsymbol{F}_{res} = \boldsymbol{0}. \tag{9}$$

Fatigue cost of past full cycles: The update of fatigue cost of full cycles which occurred at $t \leq t_0$ is obtained by the reset-map

$$\begin{aligned} &x_{fatigue,FC}(t) = \\ &\quad R_{fatigue,FC}(\boldsymbol{q}_{struct}(t^-), \boldsymbol{x}_{res}(t^-), \sigma(t^-), x_{fatigue,FC}(t^-)) = \\ &\begin{cases} \sum_j (w_j\ f_{fatigue,j}(\{\boldsymbol{x}_{res}(t^-), \sigma(t^-)\}, k_{max,j}, k_{min,j})) + \\ +x_{fatigue,FC}(t^-) \quad \forall l \in \{k_{max,j}, k_{min,j}\} \\ \quad \forall j | (w_j = 1 \wedge k_{max,j} \leq N_{past} \wedge k_{min,j} \leq N_{past}) \\ \quad \textbf{if } t = t_0 \\ \text{not active} \quad \textbf{if } t_0 < t \leq t_{end} \end{cases} \end{aligned} \tag{10}$$

which calculates a sum of convex fatigue cost functions $f_{fatigue}$ over all full cycles in the past residue (see Loew et al. (2019) for more details). Each fatigue cost function is composed by two stress samples out of the string $\{\boldsymbol{x}_{res}(t^-), \sigma(t^-)\}$; determined by the indices $k_{max,j}, k_{min,j}$. The summed fatigue cost is added to the previous version of the state $x_{fatigue,FC}(t^-)$. This state as well is accompanied by a constant continuous dynamics $\dot{x}_{fatigue,FC}(t) = F_{fatigue,FC} = 0$.

Fatigue cost: The continuous calculation of total accumulated fatigue cost is obtained by the output function

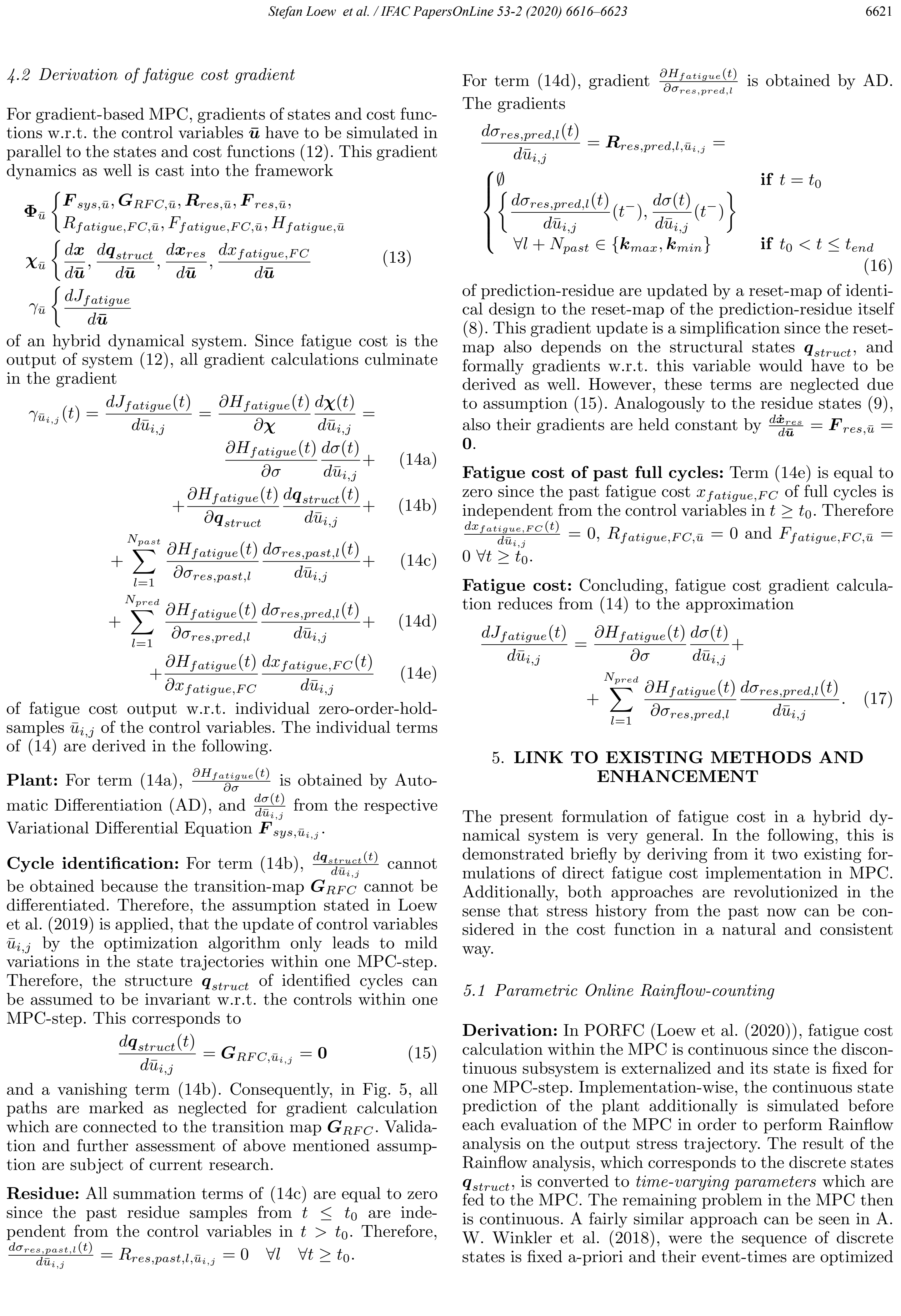
$$\begin{aligned} &J_{fatigue}(t) = y_{fatigue}(t) = \\ &\quad H_{fatigue}(\boldsymbol{q}_{struct}(t), \{\boldsymbol{x}_{res}(t), \sigma(t)\}, x_{fatigue,FC}(t)) = \\ &\quad \sum_c (w_c\ f_{fatigue,c}(\{\boldsymbol{x}_{res}(t), \sigma(t)\}, k_{max,c}, k_{min,c})) + \\ &\qquad + x_{fatigue,FC} \end{aligned} \tag{11}$$

which calculates a sum of fatigue cost over all identified full and half cycles c, and adds this to the fatigue cost of past full cycles $x_{fatigue,FC}$.

Summary: The total hybrid system Φ, its states χ and output γ are defined by

$$\boldsymbol{\Phi} \begin{cases} \boldsymbol{F}_{sys} \\ \boldsymbol{G}_{RFC} \\ \boldsymbol{R}_{res} \\ \boldsymbol{F}_{res} \\ R_{fatigue,FC} \\ F_{fatigue,FC} \\ H_{fatigue} \end{cases} \boldsymbol{\chi} \begin{cases} \boldsymbol{x} \\ \boldsymbol{q}_{struct} \\ \boldsymbol{x}_{res} \\ \\ x_{fatigue,FC} \\ \\ \\ \end{cases} \gamma \begin{cases} \\ \\ \\ \\ \\ \\ J_{fatigue}. \end{cases} \tag{12}$$

Since at any point of time the states $\boldsymbol{\chi}(t)$ contain all relevant information about the entire history, they form a *Markovian state.*



4.2 Derivation of fatigue cost gradient

For gradient-based MPC, gradients of states and cost functions w.r.t. the control variables $\bar{\boldsymbol{u}}$ have to be simulated in parallel to the states and cost functions (12). This gradient dynamics as well is cast into the framework

$$\begin{aligned} &\boldsymbol{\Phi}_{\bar{u}} \begin{cases} \boldsymbol{F}_{sys,\bar{u}}, \boldsymbol{G}_{RFC,\bar{u}}, \boldsymbol{R}_{res,\bar{u}}, \boldsymbol{F}_{res,\bar{u}}, \\ R_{fatigue,FC,\bar{u}}, F_{fatigue,FC,\bar{u}}, H_{fatigue,\bar{u}} \end{cases} \\ &\boldsymbol{\chi}_{\bar{u}} \begin{cases} \frac{d\boldsymbol{x}}{d\bar{\boldsymbol{u}}}, \frac{d\boldsymbol{q}_{struct}}{d\bar{\boldsymbol{u}}}, \frac{d\boldsymbol{x}_{res}}{d\bar{\boldsymbol{u}}}, \frac{dx_{fatigue,FC}}{d\bar{\boldsymbol{u}}} \end{cases} \\ &\gamma_{\bar{u}} \begin{cases} \frac{dJ_{fatigue}}{d\bar{\boldsymbol{u}}} \end{cases} \end{aligned} \tag{13}$$

of an hybrid dynamical system. Since fatigue cost is the output of system (12), all gradient calculations culminate in the gradient

$$\begin{aligned} \gamma_{\bar{u}_{i,j}}(t) = \frac{dJ_{fatigue}(t)}{d\bar{u}_{i,j}} &= \frac{\partial H_{fatigue}(t)}{\partial \boldsymbol{\chi}} \frac{d\boldsymbol{\chi}(t)}{d\bar{u}_{i,j}} = \\ & \frac{\partial H_{fatigue}(t)}{\partial \sigma} \frac{d\sigma(t)}{d\bar{u}_{i,j}} + && (14a) \\ & + \frac{\partial H_{fatigue}(t)}{\partial \boldsymbol{q}_{struct}} \frac{d\boldsymbol{q}_{struct}(t)}{d\bar{u}_{i,j}} + && (14b) \\ & + \sum_{l=1}^{N_{past}} \frac{\partial H_{fatigue}(t)}{\partial \sigma_{res,past,l}} \frac{d\sigma_{res,past,l}(t)}{d\bar{u}_{i,j}} + && (14c) \\ & + \sum_{l=1}^{N_{pred}} \frac{\partial H_{fatigue}(t)}{\partial \sigma_{res,pred,l}} \frac{d\sigma_{res,pred,l}(t)}{d\bar{u}_{i,j}} + && (14d) \\ & + \frac{\partial H_{fatigue}(t)}{\partial x_{fatigue,FC}} \frac{dx_{fatigue,FC}(t)}{d\bar{u}_{i,j}} && (14e) \end{aligned}$$

of fatigue cost output w.r.t. individual zero-order-hold-samples $\bar{u}_{i,j}$ of the control variables. The individual terms of (14) are derived in the following.

Plant: For term (14a), $\frac{\partial H_{fatigue}(t)}{\partial \sigma}$ is obtained by Automatic Differentiation (AD), and $\frac{d\sigma(t)}{d\bar{u}_{i,j}}$ from the respective Variational Differential Equation $\boldsymbol{F}_{sys,\bar{u}_{i,j}}$.

Cycle identification: For term (14b), $\frac{d\boldsymbol{q}_{struct}(t)}{d\bar{u}_{i,j}}$ cannot be obtained because the transition-map $\boldsymbol{G}_{RFC}$ cannot be differentiated. Therefore, the assumption stated in Loew et al. (2019) is applied, that the update of control variables $\bar{u}_{i,j}$ by the optimization algorithm only leads to mild variations in the state trajectories within one MPC-step. Therefore, the structure $\boldsymbol{q}_{struct}$ of identified cycles can be assumed to be invariant w.r.t. the controls within one MPC-step. This corresponds to

$$\frac{d\boldsymbol{q}_{struct}(t)}{d\bar{u}_{i,j}} = \boldsymbol{G}_{RFC,\bar{u}_{i,j}} = \boldsymbol{0} \tag{15}$$

and a vanishing term (14b). Consequently, in Fig. 5, all paths are marked as neglected for gradient calculation which are connected to the transition map $\boldsymbol{G}_{RFC}$. Validation and further assessment of above mentioned assumption are subject of current research.

Residue: All summation terms of (14c) are equal to zero since the past residue samples from $t \leq t_0$ are independent from the control variables in $t > t_0$. Therefore, $\frac{d\sigma_{res,past,l}(t)}{d\bar{u}_{i,j}} = R_{res,past,l,\bar{u}_{i,j}} = 0 \quad \forall l \quad \forall t \geq t_0$.

For term (14d), gradient $\frac{\partial H_{fatigue}(t)}{\partial \sigma_{res,pred,l}}$ is obtained by AD. The gradients

$$\frac{d\sigma_{res,pred,l}(t)}{d\bar{u}_{i,j}} = \boldsymbol{R}_{res,pred,l,\bar{u}_{i,j}} = \begin{cases} \emptyset & \textbf{if } t = t_0 \\ \left\{ \frac{d\sigma_{res,pred,l}(t)}{d\bar{u}_{i,j}}(t^-), \frac{d\sigma(t)}{d\bar{u}_{i,j}}(t^-) \right\} & \\ \forall l + N_{past} \in \{\boldsymbol{k}_{max}, \boldsymbol{k}_{min}\} & \textbf{if } t_0 < t \leq t_{end} \end{cases} \tag{16}$$

of prediction-residue are updated by a reset-map of identical design to the reset-map of the prediction-residue itself (8). This gradient update is a simplification since the reset-map also depends on the structural states $\boldsymbol{q}_{struct}$, and formally gradients w.r.t. this variable would have to be derived as well. However, these terms are neglected due to assumption (15). Analogously to the residue states (9), also their gradients are held constant by $\frac{d\dot{\boldsymbol{x}}_{res}}{d\bar{\boldsymbol{u}}} = \boldsymbol{F}_{res,\bar{u}} = \boldsymbol{0}$.

Fatigue cost of past full cycles: Term (14e) is equal to zero since the past fatigue cost $x_{fatigue,FC}$ of full cycles is independent from the control variables in $t \geq t_0$. Therefore $\frac{dx_{fatigue,FC}(t)}{d\bar{u}_{i,j}} = 0$, $R_{fatigue,FC,\bar{u}} = 0$ and $F_{fatigue,FC,\bar{u}} = 0 \ \forall t \geq t_0$.

Fatigue cost: Concluding, fatigue cost gradient calculation reduces from (14) to the approximation

$$\begin{aligned} \frac{dJ_{fatigue}(t)}{d\bar{u}_{i,j}} = & \frac{\partial H_{fatigue}(t)}{\partial \sigma} \frac{d\sigma(t)}{d\bar{u}_{i,j}} + \\ & + \sum_{l=1}^{N_{pred}} \frac{\partial H_{fatigue}(t)}{\partial \sigma_{res,pred,l}} \frac{d\sigma_{res,pred,l}(t)}{d\bar{u}_{i,j}}. \end{aligned} \tag{17}$$

5. LINK TO EXISTING METHODS AND ENHANCEMENT

The present formulation of fatigue cost in a hybrid dynamical system is very general. In the following, this is demonstrated briefly by deriving from it two existing formulations of direct fatigue cost implementation in MPC. Additionally, both approaches are revolutionized in the sense that stress history from the past now can be considered in the cost function in a natural and consistent way.

5.1 Parametric Online Rainflow-counting

Derivation: In PORFC (Loew et al. (2020)), fatigue cost calculation within the MPC is continuous since the discontinuous subsystem is externalized and its state is fixed for one MPC-step. Implementation-wise, the continuous state prediction of the plant additionally is simulated before each evaluation of the MPC in order to perform Rainflow analysis on the output stress trajectory. The result of the Rainflow analysis, which corresponds to the discrete states $\boldsymbol{q}_{struct}$, is converted to *time-varying parameters* which are fed to the MPC. The remaining problem in the MPC then is continuous. A fairly similar approach can be seen in A. W. Winkler et al. (2018), were the sequence of discrete states is fixed a-priori and their event-times are optimized

in a continuous and smooth MPC problem.
In comparison to the hybrid dynamical system (12), subsystems in PORFC are:

- Externalized from MPC: $\boldsymbol{G}_{RFC}$.
- Neglected: $\boldsymbol{R}_{res}, \boldsymbol{F}_{res}, R_{fatigue,FC}, F_{fatigue,FC}$ since Rainflow analysis is performed batch-wise on the entire stress prediction, and no past residue is taken into account.
- Remaining in MPC: $\boldsymbol{F}_{sys}, H_{fatigue}$

Enhancement: Following the hybrid dynamical system formulation, past stress is introduced to the fatigue calculation of PORFC by adding all neglected subsystems to above mentioned external pre-preparation phase of the MPC.

5.2 Direct Online Rainflow-counting

Derivation: In DORFC Loew et al. (2019), the entire discontinuous fatigue cost calculation is performed within the MPC. Therefore, calculation of fatigue cost and its gradients already are similar to the hybrid dynamical systems (12) and (13), respectively. The two main differences are:

- Fatigue cost is calculated in a batch fashion over the entire prediction horizon. Thus, compared to the hybrid dynamical system, the stress update $\sigma(t^-)$ is not a scalar but is the entire stress trajectory $(\sigma(t)\ \forall t \in [t_0, t_{end}])$ over the prediction horizon.
- A past residue is not taken into account.

Enhancement: The fatigue cost calculation of DORFC can be replaced by a terminal cost $\Delta J_{fatigue}$ utilizing the present hybrid dynamical system. This terminal cost represents the *added* fatigue cost within the prediction horizon. This formulation results in the following time-continuous Nonlinear Programming problem:

$$\min_{\bar{u}} -\int_{t_0}^{t_{end}} J_{revenue}(\boldsymbol{x}(t))dt + \Delta J_{fatigue}(\boldsymbol{\chi}(t_{end}), \boldsymbol{\chi}(t_0)) \quad (18a)$$

where:

$$\Delta J_{fatigue}(\boldsymbol{\chi}(t_{end}), \boldsymbol{\chi}(t_0)) = H_{fatigue}(\boldsymbol{\chi}(t_{end})) - H_{fatigue}(\boldsymbol{\chi}(t_0)) = H_{fatigue}(\boldsymbol{q}_{struct}(t_{end}), \{\boldsymbol{x}_{res}(t_{end}), \sigma(t_{end})\}, x_{fatigue,FC}(t_{end})) - H_{fatigue}(\boldsymbol{q}_{struct,0}, \{\boldsymbol{x}_{res,0}, \tilde{\sigma}\}, x_{fatigue,FC,0}) \quad (18b)$$

subject to:

$$\boldsymbol{\chi}(t) \mapsto \boldsymbol{\Phi}(\boldsymbol{\chi}(t)) \quad (18c)$$

$$\boldsymbol{\chi}(0) = (\tilde{\boldsymbol{x}}, \boldsymbol{q}_{struct,0}, \boldsymbol{x}_{res,0}, x_{fatigue,FC,0}) \quad (18d)$$

$$\begin{pmatrix} \boldsymbol{x} - \boldsymbol{x}_{max} \\ -\boldsymbol{x} + \boldsymbol{x}_{min} \end{pmatrix} \leq \boldsymbol{0} \quad (18e)$$

$$\boldsymbol{u}_{min} \leq \bar{\boldsymbol{u}} \leq \boldsymbol{u}_{max} \quad (18f)$$

If subsystems regarding past stress $\boldsymbol{R}_{res,past} = \boldsymbol{0}$ and past fatigue cost $R_{fatigue,FC} = 0$ are neglected, (18) returns similar results like the DORFC-formulation of Loew et al. (2019). If those subsystems are considered, DORFC is enhanced significantly since the MPC now can minimize stress cycles which start in the past and terminate in the prediction.

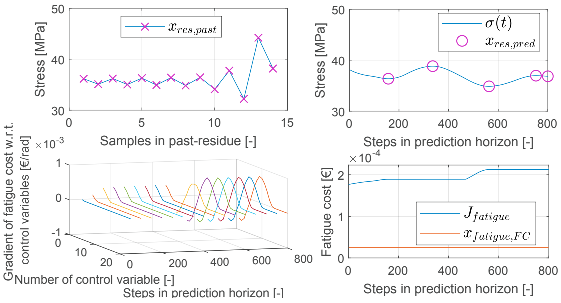


Fig. 7. *Upper left:* past-residue. *Upper right:* prediction trajectory $\sigma(t)$ and prediction-residue. *Lower right:* total fatigue cost and fatigue cost of full cycles over the prediction horizon. *Lower left:* gradients of fatigue cost w.r.t. to all samples $j = 1 ... N_u$ of one $i = 1$ control variable $\bar{u}_{i=1,j}$ in the prediction horizon. The gradient trajectory is given at every 50 out of 800 integrator steps in the prediction horizon.

6. DEMONSTRATION & VERIFICATION

The hybrid dynamical system formulation (18) is implemented in the Economic Nonlinear Model Predictive Controller of Loew et al. (2019) and tested with a wind turbine model. Goals for the MPC are maximization of revenue by harvested energy and minimization of structural fatigue at the root of the turbine tower. Important states are rotor speed and stress at the tower root. Control variables are generator torque and collective blade pitch angle. See Loew et al. (2019) for more details. Further examples of wind turbine control via MPC can be found in Luna et al. (2020), Gros and Schild (2017) and Barradas-Berglind et al. (2015). In the following, a brief insight and initial simulation results are provided.

Fig. 7 shows some of the hybrid states at a specific MPC-step. The controls are discretized zero-order-hold by $N_u = 20$ samples over the prediction horizon. The evolution of states is displayed on the integrator grid which has 800 steps per horizon. Fatigue cost $J_{fatigue}$ evolves continuously, like expected, but as well exhibits non-smoothness e.g. at step 480. This is due to the closing of a full cycle which is formed by $x_{res,pred,1}$ and $x_{res,pred,2}$, and the consequent continuation of a large half cycle which starts in the past-residue at $x_{res,past,13}$. This continuation as well results in a jump of the fatigue cost gradients. However, since (18b) is a terminal cost, only the fatigue cost gradients at the final integrator step 800 are used in the optimization. At the beginning of the horizon, the offset of $J_{fatigue}$ w.r.t. the fatigue cost of past full cycles $x_{fatigue,FC}$ equals the fatigue cost of all half cycles formed by the past-residue samples $\boldsymbol{x}_{res,past}$.

Fig. 8 shows first results from wind turbine simulations using the *DORFC* formulation from Loew et al. (2019) and the new *DORFC-R* formulation (18) including utilization of past-residue. Very generally, *DORFC-R* beneficially seems to result in smoother stress trajectories with less excursions. Especially in $82s < t < 100s$ a long-term flat behavior is observed, originating from the long-term stress memory due to past-residue $\boldsymbol{x}_{res,past}$. For the turbine rotor speed at $65s < t < 80s$ both formulations exhibit entirely

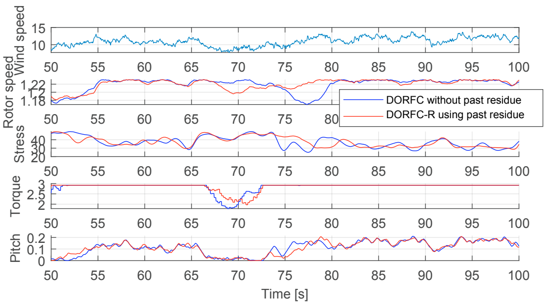


Fig. 8. Wind turbine simulation using the previous *DORFC* formulation without residue, and using the new hybrid dynamical system formulation *DORFC-R* with residue.

different strategies. Comprehensive analysis has to be done to judge which behavior is superior. In general, these very first results need to be extended by simulations which are planned for the future.

7. CONCLUSION & OUTLOOK

Within the present work, a novel algorithm for continuous fatigue cost calculation has been presented. Using this algorithm, it has been possible to cast the complete standard fatigue estimation procedure into the general framework of a hybrid dynamical system. For implementation in gradient-based MPC, a second hybrid dynamical system has been developed which provides fatigue cost gradients w.r.t. control variables. The hybrid dynamical system has been implemented in the cost function of an Economic Nonlinear MPC where it provides accurate fatigue estimation by efficiently memorizing stress samples which even might date back for several years. By deriving previous fatigue cost formulations from the present hybrid dynamical system, the impressing generality of this approach has been proven.

In the future, this new MPC implementation will be tested thoroughly by applying it to a high-fidelity wind turbine simulator and to a LiIon battery energy storage system. Furthermore, the generalization of fatigue cost functions to hybrid dynamical systems will allow for systematic theoretical analysis of stability (Müller and Allgöwer (2012)).

ACKNOWLEDGEMENTS

The authors would like to thank Prof. Carlo L. Bottasso for the fruitful discussions and his academic guidance. Further thanks goes to the Siemens-colleagues Mohamed Khalil, Armin Nurkanovic and Sebastian Albrecht.

REFERENCES

A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli (2018). Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. *IEEE Robotics and Automation Letters*, 3(3), 1560–1567. doi:10.1109/LRA.2018.2798285.

Aihara, K. and Suzuki, H. (2010). Theory of hybrid dynamical systems and its applications to biological and medical systems. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 368(1930), 4893–4914. doi:10.1098/rsta.2010.0237.

ASTM (1985). Standard practices for cycle counting in fatigue analysis.

Barradas-Berglind, J.d.J., Wisniewski, R., and Soltani, M. (2015). Fatigue damage estimation and data-based control for wind turbines. *IET Control Theory & Applications*, 9(7), 1042–1050. doi:10.1049/iet-cta.2014.0730.

Findeisen, R. and Allgöwer, F. (2002). An introduction to nonlinear model predictive control. In *21st Benelux Meeting on Systems and Control*, 1–23.

Gros, S. and Schild, A. (2017). Real-time economic nonlinear model predictive control for wind turbine control: International journal of control. *International Journal of Control*, 1–14. doi:10.1080/00207179.2016.1266514.

Grüne, L. and Pannek, J. (2017). *Nonlinear Model Predictive Control: Theory and Algorithms*. Communications and Control Engineering. Springer International Publishing, 2nd ed. 2017 edition.

Heinrich, C., Khalil, M., Martynov, K., and WEVER, U. (2019). Online remaining lifetime estimation for structures. *Mechanical Systems and Signal Processing*, 119, 312–327. doi:10.1016/j.ymssp.2018.09.028.

Köhler, M., Jenne, S., Pötter, K., and Zenner, H. (2012). *Zählverfahren und Lastannahme in der Betriebsfestigkeit*. Springer, Dordrecht.

Loew, S., Obradovic, D., Anand, A., and Szabo, A. (2020). Stage cost formulations of online rainflow-counting for model predictive control of fatigue. In *Accepted to European Control Conference 2020*.

Loew, S., Obradovic, D., and Bottasso, C.L. (2019). Direct online rainflow-counting and indirect fatigue penalization methods for model predictive control. In *2019 18th European Control Conference (ECC)*, 3371–3376. IEEE. doi:10.23919/ECC.2019.8795911.

Luna, J., Falkenberg, O., Gros, S., and Schild, A. (2020). Wind turbine fatigue reduction based on economic-tracking nmpc with direct ann fatigue estimation. *Renewable Energy*, 147, 1632–1641. doi:10.1016/j.renene.2019.09.092.

Lunze, J. and Lamnabhi-Lagarrigue, F. (2009). *Handbook of Hybrid Systems Control: Theory, Tools, Applications*. Cambridge University Press, Cambridge. doi:10.1017/CBO9780511807930.

Marsh, G., Wignall, C., Thies, P.R., Barltrop, N., Incecik, A., Venugopal, V., and Johanning, L. (2016). Review and application of rainflow residue processing techniques for accurate fatigue damage estimation. *International Journal of Fatigue*, 82, 757–765. doi:10.1016/j.ijfatigue.2015.10.007.

Müller, M.A. and Allgöwer, F. (2012). Improving performance in model predictive control: Switching cost functionals under average dwell-time. *Automatica*, 48(2), 402–409. doi:10.1016/j.automatica.2011.11.005.

Rawlings, J. and Mayne, D.Q. (2009). *Model Predictive Control: Theory and Design*. Nob Hill Pub.

Shampine, L. and Thompson, S. (2009). Delay differential equations. *Delay Differential Equations: Recent Advances and New Directions*. doi:10.1007/978-0-387-85595-0_9.

The MathWorks Inc. (2018). Rainflow counts for fatigue analysis.

CHAPTER 4

Applications

4.1 Paper 6: Wind turbine with perfect state knowledge & wind foresight

4.1.1 Summary

The economic stage cost with time-varying parameters (PORFC) from **Paper 4** is combined with the residue concept from **Paper 5**. Furthermore, an application-focused explanation for the wind energy community, and a thorough assessment of the MPC coupled with the high-fidelity wind turbine simulator `OpenFAST` is provided.

In order to demonstrate the need for the residue as "condensed memory", the concept of *Rainflow spectrum* is developed, which corresponds to a binning of Rainflow-identified cycles w.r.t. their period time. In comparison to the Fourier spectrum, for the present application the Rainflow spectrum shows more amplitude content at higher period times. Furthermore, the study indicates that - without residue - an MPC prediction horizon of at least 10s is required to see cycles with sufficient damage-equivalent effect.

The MPC formulation is implemented in the state-of-the-art MPC framework `acados`. In comparison to **Papers 1-5**, the optimization problem is extended by spline-interpolated aerodynamic nonlinearities, slack variables for recursive feasibility, and a penalization of pitch activity.

In order to retain ideal conditions for the present assessment, several degrees of freedom are deactivated in `OpenFAST`, all states are assumed as measurable, and perfect foresight of rotor-effective wind speed is provided. Aeroelastic simulations show that PORFC outperforms conventional PID- and MPC-controllers in terms of profit over the entire wind regime by up to 195% and 7.8%, respectively. Particularly above rated wind velocity, this is achieved by only minor extra pitch activity compared to the PID-controller. By adding the residue consideration to PORFC, about 1% of profit is added for low to medium wind speeds, while the pitch activity is equal or lower.

4.1.2 Contribution

The author of this dissertation combined the fatigue cost formulation with the concept of residue, performed the implementation, and executed and analyzed the simulations. The author of this dissertation and CLB conceptualized the research work and wrote the paper. DO and CLB supervised the work. All authors provided important input to this research work through discussions, feedback and by improving the manuscript.

4.1.3 Reference

S. Loew, D. Obradovic, and C. L. Bottasso, “Model predictive control of wind turbine fatigue via online rainflow-counting on stress history and prediction,” *Journal of Physics: Conference Series*, vol. 1618, p. 22041, 2020. doi:10.1088/1742-6596/1618/2/022041

The Science of Making Torque from Wind (TORQUE 2020) IOP Publishing
Journal of Physics: Conference Series **1618** (2020) 022041 doi:10.1088/1742-6596/1618/2/022041

Model predictive control of wind turbine fatigue via online rainflow-counting on stress history and prediction

S Loew[1,2], D Obradovic[1] and C L Bottasso[2]

[1]Siemens AG, Corporate Technology, Otto-Hahn-Ring 6, 81739 Munich, Germany.
[2]TU Munich, Wind Energy Institute, Boltzmannstrasse 15, 85748 Garching, Germany.

E-mail: `stefan.h.loew@tum.de`

Abstract. The standard fatigue estimation procedure is implemented in Model Predictive Control via externalization of the Rainflow algorithm from the optimization problem. Additionally, stress history is considered in a consistent manner by employing a so-called stress residue. The formulation is implemented in the state-of-the-art MPC framework acados and tested in closed-loop with the 5MW onshore turbine in OpenFAST. Simulation results indicate that the new formulation outperforms conventional PID- and MPC-controllers over the entire wind regime, and that the consideration of stress history is highly beneficial.

1. Introduction

Fatigue is damage of a material caused by cyclic application of mechanical stress. For wind turbines, fatigue has large impact on lifetime e.g. of tower, blades and drivetrain, and is a main design driver. Model Predictive Controllers (MPC) enable optimal control of turbines by utilizing predictions of incoming wind by a *Light detection and ranging* (Lidar) device [1, 2]. Based on these input predictions, stress time series at crucial spots in the turbine structure can be predicted. Rainflow-counting (RFC) is the standard method for the decomposition of stress time series for fatigue estimation. Until recently, RFC could not be implemented in MPC [3] and could only be used for post-processing of measured and simulated data. In [4], a MPC formulation was presented that allows for the externalization of the RFC evaluation from the MPC algorithm, and the inclusion of its results into the MPC via time-varying parameters. Therefore, this formulation is referred to as *Parametric Online Rainflow-counting (PORFC)*. PORFC allows for the direct incorporation of monetary fatigue in the cost function of MPC, and thus for a true economic balancing with revenue from generated electricity.

In PORFC, fatigue is calculated based on stress information from the prediction horizon of the MPC, which is in the order of a few seconds. However, fatigue is a long-term effect where stress cycles are usually defined on much longer time spans. Therefore in [5] a systematic incorporation of historic stress samples (”residue”) into the fatigue cost calculation of MPC was presented. In [4] and [5], these novel formulations were introduced in detail, but only preliminary closed-loop simulations were presented.

The main goal of the present work is to more thoroughly assess PORFC including stress history. Therefore, this paper is organized as follows. In Sec. 2, the phenomenon of fatigue and cycle identification are assessed. This analysis is the basis for an application-focused description of PORFC in Sec. 3. In Sec. 4, aeroelastic simulations over the full range of turbulent wind conditions are carried out. Furthermore, PORFC is compared to a common MPC formulation from the literature and to a conventional controller.

2. Assessment of fatigue estimation

2.1. Definition of fatigue

In the following, the phenomenon of fatigue is defined for conditions and assumptions that apply to the wind energy domain: mechanical fatigue, normal ambient temperatures, neglection of irreversible strain effects and invariance w.r.t. time. In this setting, fatigue is damage of a material caused by cyclic application of mechanical stress. Without loss of information, the fatigue impact of a given stress-trajectory can be analyzed solely based on its extrema or "reversals". This implies that the shape and contained frequencies of the original continuous stress trajectory are considered to be irrelevant for fatigue estimation [6]. Therefore, the fatigue impact of such a reversal sequence is fully determined by its contained individual stress cycles. Each stress cycle can be represented by a cosine function. A stress trajectory typically contains full cycles, which are cosines of a full period, and half cycles, which are cosines of only a half period. Half cycles therefore represent either a rising or falling transient. Instead of storing three (full cycle) or two (half cycle) stress samples, it is common to store two stress samples and a weight, which is valued $w_c = 1$ (full cycle) or $w_c = 0.5$ (half cycle). The two stress samples can be the cycle stress maximum and minimum, or the stress amplitude $\sigma_{a,c}$ and mean $\sigma_{m,c}$. Instead of stress amplitude, stress range $\sigma_{r,c} = 2\sigma_{a,c}$ is frequently used as well.

Typically, fatigue impact of a stress cycle mainly correlates with its stress amplitude: a positive stress mean increases and a negative stress mean decreases fatigue impact. Quantitatively, this *mean stress effect* is expressed by the Goodman equation [7] (p. 184) which leads to the equivalent stress $\sigma_{eq,c}$. Consequently, equivalent stress is used to calculate the number of cycles to failure

$$N_c = f_{SN}^{-1}(\sigma_{eq,c}) \tag{1}$$

via the inverse S-N or *"Woehler"* curve, which typically has a piecewise definition over the stress-axis. Fatigue damage of a given stress cycle $D_{\text{fatigue,c}} = 1/N_c$ is obtained by the reciprocal of the number of cycles to failure. Total damage of the given stress-trajectory is obtained by linear accumulation $D_{\text{fatigue}} = \sum_c D_{\text{fatigue,c}}$ of damages of individual stress cycles according to the *Miner-Palmgren-Rule* [8].

2.2. Cycle identification via the Rainflow algorithm

Cycle identification is straightforward if, e.g., a simple sinusoid is analysed. There, amplitudes, mean values and number of cycles are obvious. However, realistic stress trajectories usually are highly complex and contain stress cycles that can be nested *("nested cycles")*. Additionally, half and full cycles can be present, as stated above. The most widely accepted algorithm for cycle identification from complex trajectories is the Rainflow(-counting) algorithm (RFC) [9]. A flowchart of the Rainflow algorithm is displayed in Fig. 1.

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 memory and are triggered after identification of a

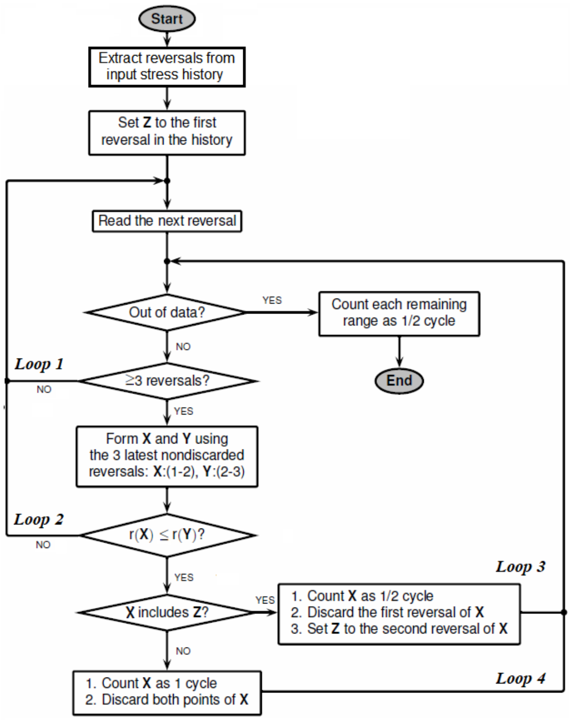

Start
Extract reversals from input stress history
Set **Z** to the first reversal in the history
Read the next reversal
Out of data?
YES
Count each remaining range as 1/2 cycle
End
NO
Loop 1
≥3 reversals?
NO
YES
Form **X** and **Y** using the 3 latest nondiscarded reversals: **X**:(1-2), **Y**:(2-3)
Loop 2
r(**X**) ≤ r(**Y**)?
NO
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 1. Flowchart of the MATLAB-implementation `rainflow()` of the *Three-point algorithm* (simplified from [10]). 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.

half or full cycle, respectively. A more comprehensive explanation of the algorithm can be found in [10].

As shown above, the Rainflow algorithm contains algorithmic branches and loops. Thus, a crucial property of the Rainflow algorithm is its discontinuous output behavior. Furthermore, the number N_c of identified cycles is not known before execution, but bounded by the number of extrema.

The characteristics of the identified cycles that are output by RFC for each cycle c are stress range $\sigma_{r,c}$ [Pa], stress mean $\sigma_{m,c}$ [Pa], sample index of cycle start $k_{start,c}$ [-], sample index of cycle end $k_{end,c}$ [-] and cycle weight $w_c[-]$. In the present work, these characteristics will be used in a converted form of stress amplitude $\sigma_{a,c}$ [Pa], stress mean $\sigma_{m,c}$ [Pa], sample index of cycle maximum $k_{max,c}$ [-], sample index of cycle minimum $k_{min,c}$ [-] and cycle weight $w_c[-]$.

2.3. Temporal range of cycle amplitudes and damage

It is important to note that stress cycles are not only caused by instantaneous oscillations, but also by long-term changes of deflection. This phenomenon is expressed by so-called *transition cycles* which grow over a long period of time and can reach high stress amplitudes with a dominating fatigue impact [11]. Since transition cycles, by definition, have not been closed yet, they appear as half cycles in the Rainflow analysis. In the wind turbine context, transition cycles are caused, e.g., by long-term evolution of mean wind velocity and can span across the turbulent and intra-day range (O(10s) to O(1day)). For the example of the turbine tower, a cycle can reach from turbine start-up (beginning of positive deflection) until shutdown (return to vertical orientation). To gain a quantitative impression of this long-term impact, in the following, an exemplary 620s tower base stress trajectory from a DLC 1.2 simulation is analyzed w.r.t. period time of contained stress cycles.

For the following analysis, the Rainflow cycles are binned w.r.t. their period time, and their amplitudes $\sigma_{a,i}$ within the respective bins are summed. This is done according to

$$f_{\mathrm{RF}}(T_i) = \sum_i \sigma_{a,i} \quad \forall i| \; T_i \leq t_{\mathrm{end,i}} - t_{\mathrm{start,i}} < T_i + \Delta T, \tag{2}$$

where T_i are discrete period times, $\Delta T = 0.005$s is the chosen bin size, and $(t_{\mathrm{start,i}}, t_{\mathrm{end,i}})$ denote start and end time of a cycle, respectively. The resulting trajectory $f_{\mathrm{RF}}(T)$ is called *"Rainflow spectrum"* in the following.

For comparison, the amplitude spectrum for the same stress trajectory is calculated by a Fast Fourier Transform (*"Fourier spectrum"*). Both spectra are cumulated over time and normalized w.r.t. their end value. As shown in Fig. 2, the cumulative Rainflow spectrum is lower than the cumulative Fourier spectrum for low period times because, for the former, open stress cycles are not closed when they are interrupted by nested cycles. Instead, an open cycle is continued after closing of a nested cycle and can grow into a larger cycle with longer period time. This translates into a higher amplitude spectrum at higher period times for the cumulated Rainflow spectrum.

Consequently, for each bin, damage is calculated based on the output of the Rainflow analysis. Damage is as well cumulated over period time and normalized w.r.t. its end value, as shown in Fig. 3. The following properties of the cumulative damage trajectory are worth mentioning:

- Damage remains close to zero until period times of $T = 3$s, which is close to the first fore-aft eigenfrequency of the turbine tower. In this set, only low-amplitude cycles are present. Due to the superlinear nature of the damage function, these cycles only cause minor damage.
- For analysis windows of about $T = 4$s, already cycles with a 40% damage-equivalent effect can be seen entirely within the window.
- Analysis windows of more than $T = 20$s would be necessary to see cycles with a 80% damage-equivalent effect within the window.

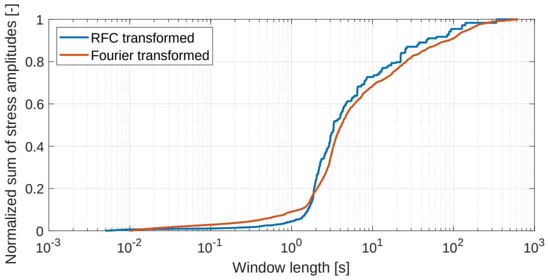


Figure 2. Normalized cumulative amplitude spectrum obtained via the Rainflow algorithm and a Fourier transformation, respectively.

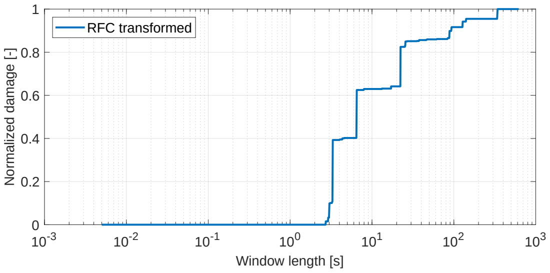


Figure 3. Normalized cumulative fatigue damage obtained via the Rainflow algorithm.

2.4. Batchwise cycle identification & residue

As demonstrated in Sec. 2.3, wind turbine stress trajectories can contain long-term cycles. Thus, the Rainflow analysis has to be carried out over the entire length of an available stress trajectory. For offline purposes, this mode is perfectly adequate. However, for online monitoring and control, a complete Rainflow analysis for each newly measured stress sample is computationally infeasible. As a solution, in [12] it is shown that Rainflow analysis also can be performed batchwise if a so-called *residue* is used for carrying along the half-cycle stress samples. Residue, therefore, denotes a set of stress samples that occured in the past and have not formed full cycles as yet.

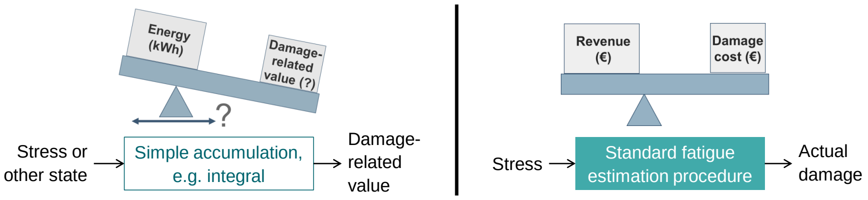


Figure 4. Left: Indirect fatigue metric. Right: Direct fatigue metric.

Depending on the stress signal, a high number of samples can be accumulated in the residue. The highest possible dimensions of the residue vector result from diverging and converging stress time series, because they result in a very high number of half cycles [13]. However, long-term diverging series are unrealistic, because unstable machine behavior typically is counteracted by the controller or an emergency shutdown. Long-term converging series are irrelevant, since very low-amplitude cycles can be discarded without significant errors in fatigue estimation. To conclude, the dimension N_{res} of the residue vector is finite and remained well below 100 in practical tests [5].

3. Fatigue in Model Predictive Control of wind turbines

Wind turbine tower base fatigue usually is implemented in MPC within the cost function. Common cost types in MPC are *Stage cost* and *Terminal cost*. Stage costs comprise a summation of state samples or a time integral of state trajectories over the prediction horizon, and are preferred for the present application. Terminal costs are defined as a function of the sole state samples at the end of the prediction horizon [14].

3.1. Indirect fatigue metrics in MPC

Several approaches reported in the literature involve indirect fatigue metrics [3], [15], [16]. However, indirect fatigue metrics have two main disadvantages:

- Instead of actual damage, only a damage-related value is obtained and optimized, as illustrated in Fig. 4.
- Indirect fatigue terms have different units from harvested energy. Thus, weighting both terms in the cost function is not straightforward.

The most common approach involves quadratic penalization of tower tip deflection rate $\dot{d}_T$. This also can be interpreted as a penalization of kinetic energy of the lumped tower mass m_T, averaged over the prediction horizon $T_{horizon}$. In the present work, therefore, the stage cost

$$J_{fatigue,TTVP} = \int_{t_0}^{t_{end}} \frac{1}{2T_{horizon}P_{\mathrm{g,max}}} m_T \dot{d}_T^2 \mathrm{d}t \qquad (3)$$

is used for comparison and referred to as *Tower tip velocity penalization (TTVP)*. An additional division by rated power $P_{\mathrm{g,max}}$ is used for scaling the cost, which is beneficial for optimization.

3.2. Direct fatigue metrics in MPC

In contrast to indirect fatigue metrics, direct fatigue metrics return actual damage, which can be readily converted to monetary fatigue cost, as visualized in Fig. 4. This conversion is achieved for instance by multiplication with Initial Capital Cost of the respective component or the entire turbine. Since harvested energy also can be converted to revenue by electricity price, the optimization algorithm can directly maximize profit.

As shown in Sec. 2, direct fatigue estimation involves the Rainflow algorithm. Implementation of RFC within a gradient-based optimization seemed impossible until now due to following obstacles:

- RFC is a function of all stress samples. Therefore, neither the concept of stage nor of terminal cost applies.
- RFC contains branches. Therefore, it exhibits discontinuous outputs and is not continuously differentiable.
- RFC contains ”while” loops, which lead to a changing function execution structure depending on the stress input.

Thus, in all known references, the Rainflow algorithm is approximated to some extent. In [17], a version of *Simple Range Counting* is applied, which is standardized in [9]. In [6], hysteresis operators are used to adapt parameters of a cost function in MPC. This cost function penalizes deflection rates, comparable to TTVP. In [18], damage estimation including standard RFC is performed on a large number of stress time series which are used to train a surrogate Artificial Neural Network. The latter seems to be very promising in terms of correct damage estimation. However, the approach involves a high a priori engineering effort, as well as a significantly increased computational load in the MPC [18].

Stress history is not included in any of these approaches. In [6], the hysteresis operators only have memory of damage evolution. Similarly, in [18], only the previous fatigue rate output of the ANN is memorized until the next evaluation.

3.3. Parametric Online Rainflow-counting - Concept

The above mentioned obstacles for a direct implementation of RFC in MPC are overcome by the method of *Parametric Online Rainflow-counting (PORFC)*. In PORFC, all discontinuous parts of the fatigue estimation procedure are carried out before each execution of the MPC algorithm, as shown in Fig. 5. Additionally, the stress history is incorporated via a residue which is inspired by the batchwise cycle identification in Sec. 2.4. The algorithmic workflow is as follows:

- **Simulation:** The reduced wind turbine model is simulated over the prediction horizon using the current measured states as initial values. Relevant result is a stress prediction, as visualized in Fig. 6 (right).
- **Merge:** The residue is merged with the stress prediction.
- **Rainflow:** The Rainflow algorithm is used to identify stress cycles over this merged trajectory. Consequently, it is assumed that the structure of identified cycles does not change within the upcoming optimization run. The term ”structure” denotes here positions ($k_{\mathrm{min,c}}$, $k_{\mathrm{max,c}}$) and weights (w_c) of cycles. As shown in Fig. 6, this assumption implies that the controllable extrema in the prediction horizon only can be shifted vertically by the optimization.
- **Residue update:** Stress cycles can be composed by stress samples only from residue or prediction, or by a combination of both. However, only controllable samples within the prediction horizon can be altered by the optimization. Especially the measured initial value at prediction step 0 cannot be altered and, therefore, is added to the residue. If a full cycle is detected entirely within the *residue*, both contributing values are discarded from the residue. The reason for this is that also in the future they will never anymore form a cycle with a sample from the prediction and, therefore, are irrelevant for the MPC.
- **Time-varying parameters:** Information from cycle identification is used to fill vectors of time-varying parameters, which are forwarded to the cost function of the MPC. Details on this step are provided in Sec. 3.4.

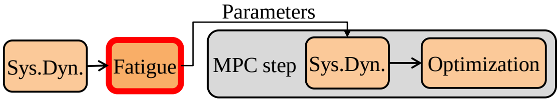


Figure 5. Externalization of fatigue estimation (Rainflow algorithm) from MPC.

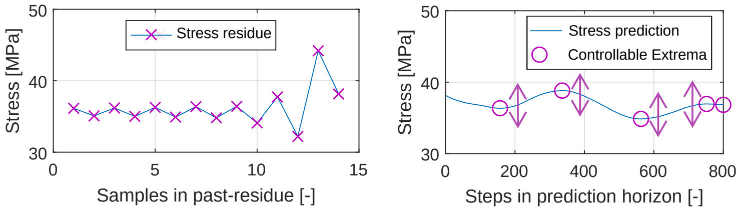


Figure 6. Left: Stress residue from the past. Right: Stress prediction in the future.

- **Optimization/MPC:** In the cost function of the MPC, the parameters are used to time-continuously calculate fatigue cost over the horizon and accumulate it via integration. Finally, the optimization problem is solved and the resulting control variables are applied to the wind turbine plant.

3.4. Parametric Online Rainflow-counting - Time-varying parameters & cost function

Distribution of damage over time: Since information from cycle identification is forwarded to the MPC via parameters, which are varying over the prediction horizon, the total fatigue damage has to be distributed over the prediction horizon, as visualized in Fig. 7 (right). Therefore, the damage of each stress cycle is split into two halves, which are allocated to the two contributing stress samples. For example, cycle 4 is formed by samples $k = 5$ and $k = 8$. Their fatigue cost terms therefore are allocated to these samples, as shown by the blocks in Fig. 7 (right). This example also shows an important property of the Rainflow algorithm, which identifies cycle 4 even though it is interrupted by the nested cycle 2. If, for a given stress sample, the complementary stress sample is not controllable (residue or initial value), all damage is allocated to the given sample. Here, this is the case for cycle one, where all damage is allocated to sample $k = 2$ since the complementary stress sample at $k = 0$ is not controllable.

Setup of the time-varying parameters: Figure 7 (left) visualizes the generation of the time-varying parameters. Since each stress extremum belongs to one or two stress cycles [19], the Rainflow algorithm provides one or two mean stresses per extremum. These mean stresses (M1 - M4) are considered as optimization- or tracking-goals for the current MPC-step. A more detailed derivation and explanation can be found in [4].

Cost function: Consequently, the fatigue cost term of PORFC is defined. One comment on notation: the variable notation with hat $\hat{a}$ means fixed for one MPC-step and with bar $\bar{a}$ means sampled on the control intervals of the prediction horizon.

The fatigue cost function is defined by an integral over two cost terms, each one representing one potential cycle contribution of a stress sample, i.e.:

$$J_{fatigue,PORFC}\left(\sigma, \bar{\boldsymbol{p}}\right) = \frac{1}{T_{cntrl}} \int_{t_0}^{t_{end}} \left(J_{fatigue,c}(\sigma(t), \hat{\bar{\sigma}}_{m,c1}(t), \hat{\bar{w}}_{c1}(t)) + J_{fatigue,c}(\sigma(t), \hat{\bar{\sigma}}_{m,c2}(t), \hat{\bar{w}}_{c2}(t))\right) \mathrm{d}t \quad [€]. \quad (4)$$

The cost terms are "switched on" by nonzero cycle weights $\hat{\bar{w}}_{c1/2}(t)$. Mean stresses $\hat{\bar{\sigma}}_{m,c1/2}(t)$

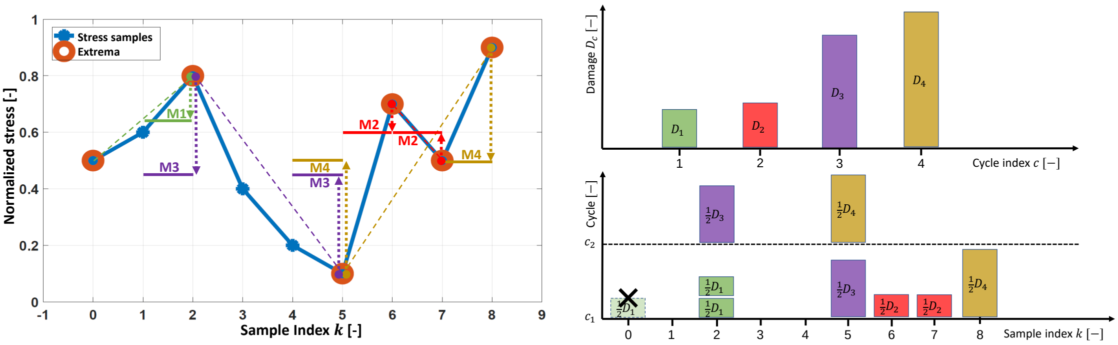


Figure 7. Left: Initial stress trajectory (blue), its extrema (orange circles), generated time-varying reference stresses (solid green, red, purple, yellow) and optimization goals (dotted arrows) for PORFC. Right: Corresponding distribution of damage over the prediction horizon. Both figures are modified from [20].

and cycle weights $\hat{\bar{w}}_{c1/2}(t)$ are collected in the parameter vector

$$\bar{\boldsymbol{p}} = \left(\hat{\bar{\sigma}}_{m,c1}, \hat{\bar{\sigma}}_{m,c2}, \hat{\bar{w}}_{c1}, \hat{\bar{w}}_{c2}\right), \tag{5}$$

which is defined as piecewise constant over the control intervals of the prediction horizon. The cost of individual cycles is defined by

$$J_{fatigue,c} = \hat{\bar{w}}_{c1/2}(t)\ a_m\ |\sigma(t) - \hat{\bar{\sigma}}_{m,c1/2}(t)|^m. \tag{6}$$

3.5. *Optimization problem for TTVP and PORFC*

The Economic MPC of a wind turbine is defined by the following optimization problem

$$\min_{\bar{\mathbf{u}},\bar{\mathbf{s}}} \left(-\alpha_{\text{revenue}} J_{\text{revenue}} + \alpha_{\text{fatigue}} J_{\text{fatigue}} + \int_{t_0}^{t_{end}} \left(10^{-2}|\dot{\beta}_b| + 10^7\ \bar{s}_\omega^2 + 10^7\ \bar{s}_P^2\right) \mathrm{d}t\right), \tag{7}$$

which maximizes the revenue J_{revenue} and minimizes the fatigue $J_{\text{fatigue,PORFC}}$, where α_{revenue} and α_{fatigue} are weighting factors. Instead of generator power, aerodynamic power is maximized by $J_{\text{revenue}} = \omega_r\, T_w(\omega_r, V_{rel}, \beta_b)$ to avoid a greedy extraction of rotor kinetic energy by the MPC (*turnpike effect*), as suggested by [15]. Furthermore, pitch travel $|\dot{\beta}_b|$, and slack variables for rotational speed $\bar{s}_\omega$ and generator power $\bar{s}_P$ are penalized (see their use in the constraints below). The optimization variables are the demanded pitch angle and torque rate $\bar{\mathbf{u}} = (\bar{\boldsymbol{\beta}}_{b,d}, \dot{\bar{\mathbf{T}}}_{g,d})$, and the slack variables $\bar{\mathbf{s}} = (\bar{\mathbf{s}}_\omega, \bar{\mathbf{s}}_P)$.

For both TTVP and PORFC, revenue is weighted by the current electricity price $\alpha_{\text{revenue,PORFC}} = p_{\text{elec}}$ [€/Ws] to match the monetary nature of (4). The fatigue weight remains free and will be determined later in this work.

The optimization problem is subject to:

- The system dynamics of a reduced turbine model $\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t))$ with six states: rotational speed of the rotor, tower tip deflection, tower tip velocity, pitch angle, pitch rate and generator torque. More details about the model are given in [21].
- Inequality constraints over the horizon, to keep rotational speed, tower deflection (yield strength), pitch angle, pitch rate, generator torque and generator power within their limits. In order to maintain feasibility of the optimization despite model uncertainties and spontaneous constraint violations, the constraints on rotational speed and generator power are augmented by slack variables, as suggested by [22].
- Box constraints on control and slack variables.

4. Simulation setup and results

4.1. Simulation setup

Plant model: The designed MPC formulations are tested with the NREL 5MW onshore reference turbine [23] in the aeroelastic simulator `OpenFAST`. The present preliminary study is focused on assessment of the formulations in ideal conditions. Therefore, exact measurements of the required turbine states are provided to the MPC. Furthermore, in `OpenFAST`, only the tower fore-aft and the rotor Degree of Freedom (DOF) are activated. Future work will be directed to extending the MPC-internal turbine model and finally enabling all DOFs in `OpenFAST`. All results in this work are mean values of 12 simulations (each with a different seed) of 600s length in DLC 1.2 with turbulence category A.

Lidar model: The Lidar-simulator of `OpenFAST` is used and set to provide exact pointwise wind measurements. These measurements are obtained via circular scan patterns of four different radii $r = \{0.2, 0.4, 0.6, 0.8\}R_{\text{rotor}}$ which are applied in alternating fashion. For each radius, 20 azimuthal stops are distributed equally (every 18 degrees). Inspired by [2], each measurement is weighted by the cubic of the scanning radius, since the associated area increases quadratically, and the stationary spanwise variation $\frac{\partial C_p}{\partial r}$ of the power coefficient increases approximately linearly with the scanning radius. Every 5 ms a new wind sample is obtained. Due to vertical wind shear, this raw wind prediction signal oscillates, an effect which is counteracted by applying a moving mean filter, whose window length corresponds to two full scans over all radii and azimuths. The longitudinal scanning distance is set to $x_{scan} = V_{w,ref}\, T_{\text{horizon}}$, where $V_{w,ref}$ is the reference wind velocity of a given DLC, and $T_{\text{horizon}} = 8\text{s}$ is the horizon length of the MPC. This value for the horizon length was chosen based on the findings in Sec. 2.3 which indicate, that cycles of at least 60% of damage-equivalent effect will be contained in the controllable prediction horizon.

MPC framework: The MPC is implemented in the state-of-the-art `acados` framework [24], using the interior-point solver `HPIPM` for the underlying Quadratic Programs (QP). The controller sample time is 100 ms. Maximum 5 QPs are solved per MPC step to ensure results close to convergence. The Hessian matrix is automatically convexified to account for possible numerical issues due to the highly non-standard cost formulation of PORFC.

Controller variants: In the following, performance of five MPC formulations and the baseline conventional controller (CC) from NREL [23] are compared. The MPCs involve the conventional formulation of Tower tip velocity penalization (TTVP, Sec. 3.1) and the novel formulation of Parametric Online Rainflow-counting (PORFC, Sec. 3.3). For PORFC, a fatigue exponent of $m = 5$ (see (6)) is used, which is present at low stress amplitudes in the actual S-N-curve of the tower material. This case is assessed in combination with (PORFC-5R) and without (PORFC-5) the use of residue. Since especially PORFC-5R will not lead to satisfactory results, additional formulations (PORFC-2R, PORFC-2) with fatigue exponent $m = 2$ are assessed which result in quadratic cost functions, and thus are more suitable for Quadratic Programming.

Performance indicators: Considered performance indicators are revenue (analog to energy), fatigue cost (based on a realistic piecewise defined S-N-curve, see Sec. 2.1), profit (revenue subtracted by fatigue cost), pitch travel and torque travel.

4.2. Simulation results for weight variation

As shown in Sec. 3.5, the fatigue weight α_{fatigue} is a free tuning variable. In contrast to the Online-Rainflow MPC setup in [25], in the present setup a choice of $\alpha_{\text{fatigue}} = 1$ does not seem to result in maximum profit. One reason is the additional presence of non-monetary cost terms next to revenue and fatigue in the MPC cost function (7). Another reason may be the approximating nature of PORFC, where fatigue is distributed over and decoupled in time (see Sec. 3.4).

For the purpose of weight-tuning, Fig. 8 shows results for variations of fatigue weight for all MPC setups and a reference wind velocity of 7 m/s. This wind velocity was chosen, since

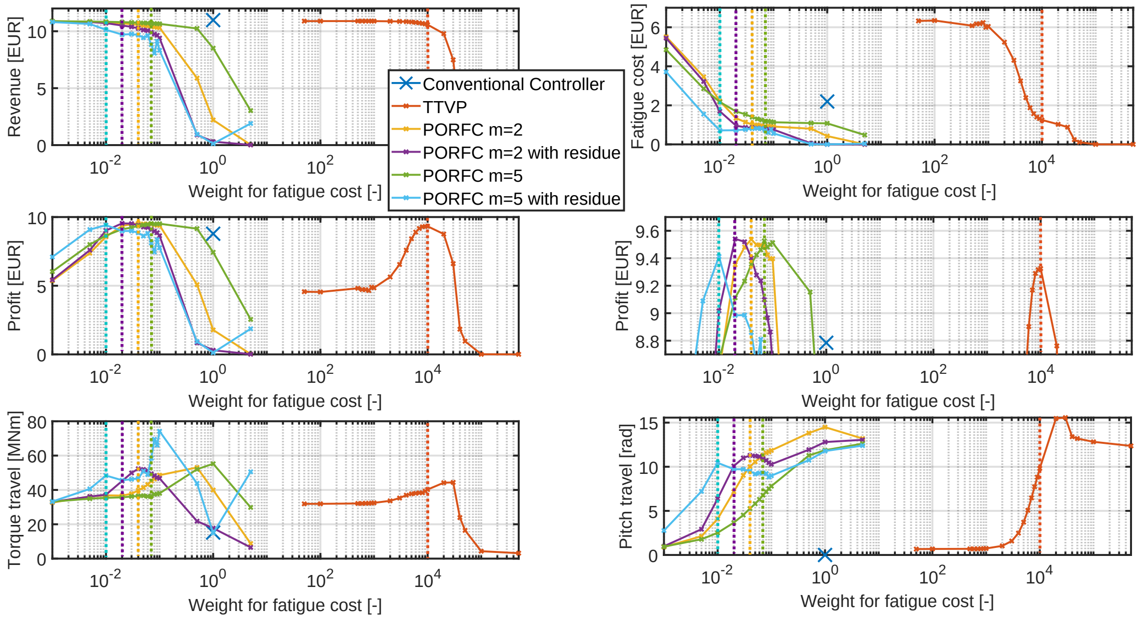


Figure 8. Variation of fatigue weights. Vertical dashed lines: location of profit-optimal weights. Middle row, right: Zoomed version of profit plot.

it is close to the mean annual wind velocity for the given hub height and an onshore site. Profit-optimality is the prime goal and achieved by the following weights: 10^4 (TTVP), $4 \cdot 10^{-2}$ (PORFC-2), $2 \cdot 10^{-2}$ (PORFC-2R), $7 \cdot 10^{-2}$ (PORFC-5) and 10^{-2} (PORFC-5R).

For very low fatigue weights, all MPCs (TTVP, PORFC) are able to achieve similar revenue that, however, is about 1% lower than the revenue of CC. The investigation of reasons for this behavior is part of ongoing research. With increasing weights, revenue is sacrificed to reduce fatigue cost. For all MPCs, the major drop in fatigue cost appears for lower weights than the drop of revenue. As a result, pronounced peaks in profit are visible. Optimum profit of TTVP is 6.2% higher than the profit of CC. The profit-peaks of PORFC-2, -2R and -5 are at almost equal levels and about 8.6% higher than the profit of CC. Unfortunately, the performance of PORFC-5R is slightly deteriorated, but still higher than the one of TTVP.

Width of the profit-peaks is suggested as an indicator for robustness of tuning. Width, e.g., can be measured by the weight range where profit is above a certain threshold, which here is defined as 95% of the individual maximum profit. In the present case, width is half an order of magnitude for TTVP, almost one order of magnitude for PORFC-2(R) and PORFC-5R, and substantially more than one order of magnitude for PORFC-5. To conclude, PORFC-5 is considered as the most robust against inaccurate tuning.

4.3. Simulation results for velocity variation

As a next step, the above-presented profit-optimal fatigue weights are fixed and the controllers are applied to a comprehensive range of reference wind velocities. The results are normalized w.r.t. TTVP and are presented in Fig. 9. Strikingly, all MPCs are extracting significantly less energy than CC at low wind velocities. Just like in [2], this is due to the MPCs' strategy of reducing fatigue via significantly decreasing rotor speed. Above rated wind velocity, the MPCs result in about 1% higher revenue than CC. An exception is PORFC-5R, which performs unreliably over the wind regime and is excluded from the remaining analysis.

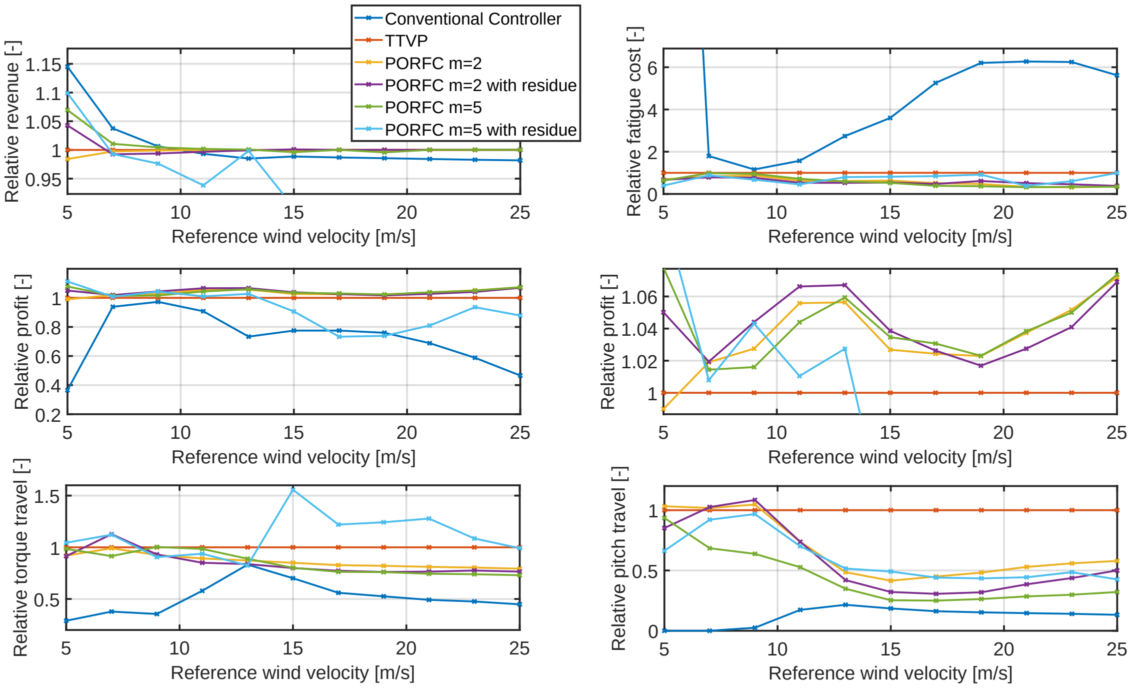


Figure 9. Variation of reference wind velocities. All resulting quantities are normalized w.r.t. the MPC using TTVP. Middle row, right: Zoomed version of profit plot.

TTVP exhibits between 14% (9m/s) and 98% (5m/s) less fatigue cost than CC. This is further improved by PORFC-5, which exhibits between 2% (7m/s) and 68% (21m/s) less fatigue cost than TTVP. As a result, the MPCs generally exhibit higher profit than CC, especially for very low wind velocity and above rated. Specifically TTVP exhibits between 2.8% (9m/s) and 174% (5m/s) more profit than CC. The advanced formulations of PORFC add further improvement, where, e.g., PORFC-5 exhibits up to 7.8% more profit than TTVP. The beneficial effect of residue can be observed from PORFC-2 to -2R, where significant profit is added for low to medium wind velocities, while pitch travel is equal or lower.

Pitch activity usually is a main concern for the control of fatigue. Indeed, for low wind velocities, the MPCs exhibit substantial pitch activity in comparison to almost none for CC. Above rated, PORFC reliefs the pitch system by about 50-75% in comparison to TTVP, and almost reaches the low levels of CC.

The deteriorated behavior of PORFC-5R cannot be fully explained yet. It may be caused by numerical difficulties, which arise from the approximation of the cost function of exponent $m = 5$ by a *Quadratic* Program in the optimization algorithm. This phenomenon may manifest itself with the use of residue, since here higher (long-term) stress amplitudes are detected which lie in a steep region of the damage function (6).

5. Conclusion & Outlook

In the present work, the MPC formulation of *Parametric Online Rainflow-counting* (PORFC) has been presented in an application-focused way. It has been highlighted how PORFC directly incorporates mechanical fatigue in predictive wind turbine control. A study on period times of tower base stress cycles has demonstrated that long observation windows are required to see cycles with sufficient damage-equivalent effect. As a solution for MPC, which typically is based on short prediction horizons, stress history can be considered in a consistent manner by carrying along a *residue*. Aeroelastic simulations show that PORFC outperforms conventional PID- and MPC-controllers in terms of profit over the entire wind regime. Particularly above rated wind velocity, this is achieved by a moderate amount of extra pitch activity compared to the PID-controller. Furthermore, a simple test has shown that all variants of PORFC are more robust against inaccurate tuning compared to the conventional MPC formulation of TTVP.

Future research concerning PORFC will be directed to monetary cost functions for actuator usage, application with a sophisticated Lidar-simulator, and further assessment of influencing factors like MPC settings and state uncertainty.

References

[1] Bottasso C L, Pizzinelli P, Riboldi C and Tasca L 2014 *Renewable Energy* **71** 442–452 ISSN 09601481
[2] Schlipf D, Schlipf D J and Kühn M 2013 *Wind Energy* **16** 1107–1129
[3] Barradas-Berglind J J and Wisniewski R 2016 *Wind Energy* **19** 2189–2203
[4] Loew S, Obradovic D, Anand A and Szabo A Stage cost formulations of online rainflow-counting for model predictive control of fatigue *Accepted to European Control Conference 2020*
[5] Loew S and Obradovic D 2020 Formulation of fatigue dynamics as hybrid automaton for model predictive control *Accepted for IFAC World Congress 2020 2020*
[6] Barradas-Berglind J d J, Wisniewski R and Soltani M 2015 *IET Control Theory & Applications* **9** 1042–1050 ISSN 1751-8644
[7] Haibach E 2006 *Betriebsfestigkeit: Verfahren und Daten zur Bauteilberechnung* 3rd ed VDI-Buch (Berlin: Springer) ISBN 9781280618024
[8] Miner M A 1945 *Journal of Applied Mechanics* 159–164
[9] ASTM 1985 Standard practices for cycle counting in fatigue analysis
[10] The MathWorks Inc 2018 Rainflow counts for fatigue analysis
[11] Marsh G, Wignall C, Thies P R, Barltrop N, Incecik A, Venugopal V and Johanning L 2016 *International Journal of Fatigue* **82** 757–765 ISSN 0142-1123
[12] Heinrich C, Khalil M, Martynov K and WEVER U 2019 *Mechanical Systems and Signal Processing* **119** 312–327 ISSN 0888-3270
[13] Köhler M, Jenne S, Pötter K and Zenner H 2012 *Zählverfahren und Lastannahme in der Betriebsfestigkeit* (Dordrecht: Springer) ISBN 978-3-642-13163-9
[14] Grüne L and Pannek J 2017 *Nonlinear Model Predictive Control: Theory and Algorithms* 2nd ed Communications and Control Engineering (Springer International Publishing) ISBN 978-3-319-46023-9
[15] Gros S and Schild A 2017 *International Journal of Control* 1–14 ISSN 0020-7179
[16] Evans M A, Cannon M and Kouvaritakis B 2015 *IEEE Transactions on Control Systems Technology* **23** 290–296 ISSN 1063-6536
[17] Sanchez H, Escobet T, Puig V and Odgaard P F 2015 *IFAC-PapersOnLine* **48** 1363–1368 ISSN 2405-8963
[18] Luna J, Falkenberg O, Gros S and Schild A 2020 *Renewable Energy* **147** 1632–1641 ISSN 09601481
[19] Shi Y, Xu B, Tan Y and Zhang B A convex cycle-based degradation model for battery energy storage planning and operation
[20] Anand A 2020 *Optimal Control of Battery Energy Storage System for Grid Integration of Wind Turbines* Master thesis TU Munich Munich
[21] Löw S and Obradovic D 2018 *atp edition* **60** 46–53 ISSN 2364-3137
[22] Gros S 2013 An economic nmpc formulation for wind turbine control *52nd IEEE Conference on Decision and Control* pp 1001–1006
[23] Jonkman J, Butterfield S, Musial W and Scott G 2009 *Definition of a 5-MW Reference Wind Turbine for Offshore System Development: Technical Report NREL/TP-500-38060* (National Renewable Energy Laboratory)
[24] Robin Verschueren, Gianluca Frison, Dimitris Kouzoupis, Niels van Duijkeren, Andrea Zanelli, Branimir Novoselnik, Jonathan Frey, Thivaharan Albin, Rien Quirynen and Moritz Diehl 2019 acados: a modular open-source framework for fast embedded optimal control arXiv
[25] Loew S, Obradovic D and Bottasso C L 2019 Direct online rainflow-counting and indirect fatigue penalization methods for model predictive control *2019 18th European Control Conference (ECC)* (IEEE) pp 3371–3376 ISBN 978-3-907144-00-8

4.2 Paper 7: Wind turbine with estimated states & realistic wind foresight

4.2.1 Summary

In **Paper 6**, the novel PORFC MPC formulation was simulated in a partially idealized setting where only a few degrees of freedom (DOF) in the high-fidelity plant turbine model were activated, and where full information about the incoming wind and about the turbine states was assumed. The present work closes these gaps by thoroughly assessing the formulation in a more realistic simulation scenario. Particularly, all DOFs of the plant are activated, a Moving Horizon Estimator (MHE) provides initial state estimates for the MPC, and a Lidar simulator is utilized to generate a realistically imperfect wind estimate.

A sequential tuning approach is employed for the Lidar simulator, Lidar data processing, MHE, and MPC. Here, key findings are:

- For the Lidar data processing, simple adaptive tuning laws are sufficient without further tuning.
- For the MHE, instead of accurately reconstructing the plant states, it is important to estimate only the low-frequency state information which can be handled by the MPC-internal model.
- For the MPC, no horizon length leads to the best performance in all scenarios.

Finally, extensive economic and dynamic simulation results in turbulent and gust wind settings reveal the following insights:

- In the "Default lidar" scenario with high prediction quality, PORFC-2R exhibits 2.5% higher profit than a conventional MPC and 30% higher profit than a conventional PID controller. Towards shorter horizons, especially the PORFC formulation with residue shows a higher profit stability than the conventional MPC.
- In the "High decay lidar" scenario with low prediction quality, PORFC-2R even increases its profit benefit over the conventional MPC to 5.1%, and still surpasses the conventional PID controller by 26%. This suggests that PORFC-2R is strong especially towards lower Lidar prediction quality.
- In the "Perfect prediction" scenario, both MPCs exhibit similar results. A comparison to a wind persistence setting shows that PORFC-2R benefits more from the availability of this prediction than the conventional MPC.
- In Extreme Operating Gust settings, both MPCs show similar pitching and tower dynamics.

4.2.2 Contribution

The author of this dissertation combined the fatigue cost formulation with the concept of residue, formulated the Moving Horizon Estimator, implemented and tuned the controller toolchain, and executed and analyzed the simulations. CLB supervised the work. Both authors jointly conceptualized the research work, and provided important input to this research work through discussions, feedback and by writing the paper.

4.2.3 Reference

S. Loew and C. L. Bottasso, “Lidar-assisted model predictive control of wind turbine fatigue via online rainflow-counting considering stress history,” *Wind Energy Science Discussions*, vol. 2021, pp. 1–31, 2021. doi:10.5194/wes-2021-119

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

Lidar-assisted model predictive control of wind turbine fatigue via online rainflow-counting considering stress history

Stefan Loew[1] and Carlo L. Bottasso[1]

[1]Wind Energy Institute, Technical University of Munich, 85748 Garching b. München, Germany

Correspondence: Stefan Loew (stefan.h.loew@tum.de)

Abstract. The formulation of Parametric Online Rainflow Counting implements the standard fatigue estimation process and a stress history in the cost function of a Model Predictive Controller. The formulation is tested in realistic simulation scenarios where the states are estimated by a Moving Horizon Estimator and the wind is predicted by a lidar simulator. The tuning procedure for the controller toolchain is carefully explained. In comparison to a conventional MPC in a turbulent wind setting, the novel formulation is especially superior with low lidar quality, benefits more from the availability of a wind prediction, and exhibits a more robust performance with shorter prediction horizons. A simulation excerpt with the novel formulation provides deeper insight into the update of the stress history and the fatigue cost parameters. Finally, in a deterministic gust setting, both the conventional and the novel MPC - despite their completely different fatigue cost - exhibit similar pitch behavior and tower oscillation.

1 Introduction

Fatigue is damage of a material caused by cyclic application of mechanical stress. For wind turbines, fatigue has a large impact on lifetime e.g. of tower, blades and drivetrain, and is a main design driver. Model Predictive Controllers (MPC) enable optimal control of turbines by utilizing predictions of the incoming wind by a *light detection and ranging* (lidar) device (Bottasso et al., 2014; Schlipf et al., 2013). Based on these input predictions, stress time series at crucial spots in the turbine structure can be predicted. Rainflow-counting (RFC) is the standard method for the decomposition of stress time series for fatigue estimation. Until recently, RFC could not be implemented in MPC (Barradas-Berglind and Wisniewski, 2016) and could only be used for post-processing of measured and simulated data. In Loew et al. (2020a), a MPC formulation was presented that allows for the externalization of the RFC evaluation from the MPC algorithm, and the inclusion of its results into the MPC via time-varying parameters. Therefore, this formulation is referred to as *Parametric Online Rainflow Counting (PORFC)*. PORFC allows for the direct incorporation of monetary fatigue in the cost function of MPC, and thus for a true economic balancing with revenue from generated electricity.

In PORFC, fatigue is calculated based on stress information from the prediction horizon of the MPC, which is in the order of a few seconds. However, fatigue is a long-term effect where stress cycles are usually defined on much longer time spans. Therefore, in Loew et al. (2020b) PORFC was combined with a systematic incorporation of historic stress samples ("residue").

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

In the same work, this formulation was simulated in an idealized setting where only a few degrees of freedom (DOF) in the plant turbine model were activated, and where full information about the incoming wind and the turbine states was assumed.

The main goal of the present work is to thoroughly assess the formulation in a more realistic simulation scenario. Particularly, all DOFs of the plant are activated, a Moving Horizon Estimator provides initial state estimates for the MPC, and a lidar simulator is utilized to generate a realistically imperfect wind estimate. The assessment is performed in several turbulent as well as deterministic gust scenarios.

This paper is organized as follows. In Sect. 2, the phenomenon of fatigue and cycle identification are reviewed. This analysis is the basis for an application-focused description of PORFC in Sect. 3. In Sect. 4, a Moving Horizon Estimator is formulated. In Sect. 5, the controller toolchain and the tuning of each of its elements are presented. Finally, PORFC is compared to a conventional MPC and to a conventional PID controller in the above mentioned simulation scenarios.

2 Review of fatigue estimation

In the following, fatigue is defined, cycle identification is explained, and the concept of *residue* is presented.

2.1 Definition of fatigue

In the following, the phenomenon of fatigue is defined for conditions and assumptions that apply to the wind energy domain: namely mechanical fatigue, normal ambient temperatures, neglection of irreversible strain effects and invariance with respect to time. In this setting, fatigue is damage of a material caused by cyclic application of mechanical stress. Without loss of information, the fatigue impact of a given stress-trajectory can be analyzed solely based on its extrema or "reversals". This implies that the shape and contained frequencies of the original continuous stress trajectory are considered to be irrelevant for fatigue estimation (Barradas-Berglind et al., 2015). Therefore, the fatigue impact of a reversal sequence is fully determined by its contained individual stress cycles. Each stress cycle can be represented by a cosine function. A stress trajectory typically contains full cycles, which are cosines of a full period, and half cycles, which are cosines of only a half period. Half cycles therefore represent either a rising or falling transient. Instead of storing three (full cycle) or two (half cycle) stress samples, it is common to store two stress samples and a weight, which is valued $w_c = 1$ (full cycle) or $w_c = 0.5$ (half cycle). The two stress samples can be the cycle stress maximum and minimum, or the stress amplitude $\sigma_{\mathrm{a},c}$ and mean $\sigma_{\mathrm{m},c}$. Instead of stress amplitude, stress range $\sigma_{\mathrm{r},c} = 2\sigma_{\mathrm{a},c}$ is frequently used as well.

Typically, fatigue impact of a stress cycle mainly correlates with its stress amplitude: a positive stress mean increases whereas a negative stress mean decreases fatigue impact. Quantitatively, this *mean stress effect* is expressed by the Goodman equation (Haibach, 2006) (p. 184), which leads to the equivalent stress $\sigma_{\mathrm{eq},c}$. Consequently, equivalent stress is used to calculate the number of cycles to failure

$$N_c = f_{\mathrm{SN}}^{-1}(\sigma_{\mathrm{eq},c}) \tag{1}$$

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

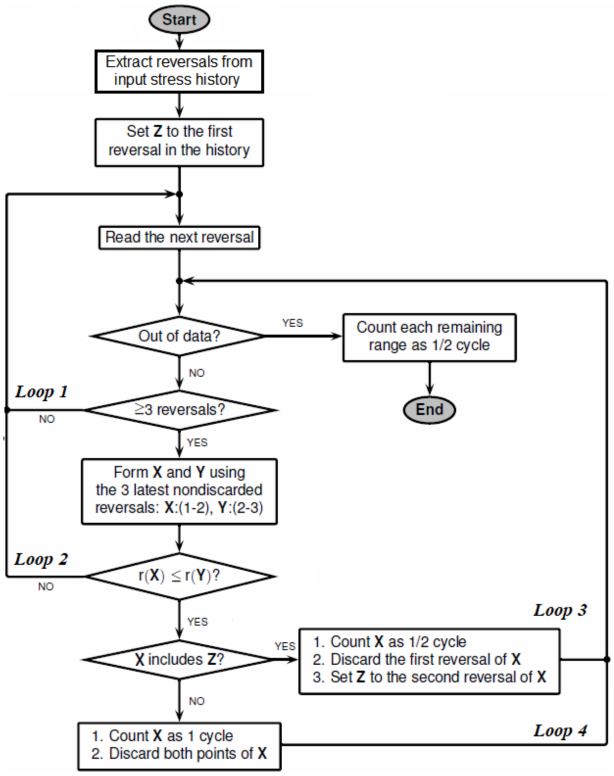

Start
Extract reversals from input stress history
Set Z to the first reversal in the history
Read the next reversal
Out of data?
YES
Count each remaining range as 1/2 cycle
End
NO
Loop 1
NO
≥3 reversals?
YES
Form X and Y using the 3 latest nondiscarded reversals: X:(1-2), Y:(2-3)
Loop 2
NO
r(X) ≤ r(Y)?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

Figure 1. Flowchart of the MATLAB-implementation `rainflow()` of the *Three-point algorithm* (simplified from The MathWorks Inc. (2018)). 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.

via the inverse S-N or *"Woehler"* curve, which typically has a piecewise definition over the stress-axis. Fatigue damage of a given stress cycle

$$D_{\text{fatigue,c}} = 1/N_c \tag{2}$$

is obtained by the reciprocal of the number of cycles to failure. Total damage of the given stress trajectory is obtained by linear accumulation $D_{\text{fatigue}} = \sum_c D_{\text{fatigue,c}}$ of damages of individual stress cycles according to the *Miner-Palmgren-Rule* (Miner, 1945).

2.2 Cycle identification via the Rainflow algorithm

Cycle identification is straightforward if, e.g., a simple sinusoid is analysed. There, amplitudes, mean values and number of cycles are obvious. However, realistic stress trajectories usually are highly complex and contain stress cycles that can be nested *("nested cycles")*. Additionally, half and full cycles can be present, as stated above. The most widely accepted algorithm for cycle identification from complex trajectories is the Rainflow(-counting) algorithm (RFC) (ASTM, 1985). A flowchart of the Rainflow algorithm is displayed in Fig. 1.

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

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 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 Inc. (2018).

As shown above, the Rainflow algorithm contains algorithmic branches and loops. Thus, a crucial property of the Rainflow algorithm is its discontinuous output behavior. Furthermore, the number N_c of identified cycles is not known before execution, but bounded by the number of extrema.

The characteristics of the identified cycles that are output by RFC for each cycle c are: stress range $\sigma_{\mathrm{r},c}$ [Pa], stress mean $\sigma_{\mathrm{m},c}$ [Pa], sample index of cycle start $k_{\mathrm{start},c}$ [-], sample index of cycle end $k_{\mathrm{end},c}$ [-], and cycle weight $w_c[-]$. In the present work, these characteristics will be used in a converted form of 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 $k_{\mathrm{min},c}$ [-], and cycle weight $w_c[-]$.

2.3 Batchwise cycle identification and residue

As shown in Loew et al. (2020b), wind turbine stress trajectories can contain long-term cycles. Thus, the Rainflow analysis has to be carried out over the entire length of an available stress trajectory. For offline purposes, this mode is perfectly adequate. However, for online monitoring and control, a complete Rainflow analysis for each newly measured stress sample is computationally infeasible. As a solution, Heinrich et al. (2019) showed that Rainflow analysis can be performed batchwise if a so-called *residue* is used for carrying along the half-cycle stress samples. Residue, therefore, denotes a set of stress samples that occured in the past and have not formed full cycles as yet.

Depending on the stress signal, a high number of samples can be accumulated in the residue. The maximum possible length of the residue vector results from diverging and converging stress time series, because they generate a large number of half cycles (Köhler et al., 2012). However, long-term diverging series are unrealistic, because unstable machine behavior typically is counteracted by the controller or an emergency shutdown. Long-term converging series are irrelevant, since very low-amplitude cycles can be discarded without significant errors in fatigue estimation. To conclude, the length of the residue vector is finite and remained well below 100 in practical tests (Loew and Obradovic, 2020).

3 Fatigue in Model Predictive Control of wind turbines

Wind turbine fatigue is usually implemented in MPC within the cost function. Common cost types in MPC are *Stage cost* and *Terminal cost*. Stage costs comprise a summation of state samples or a time integral of state trajectories over the prediction horizon, and are preferred for the present application. Terminal costs are defined as a function of the sole state samples at the end of the prediction horizon (Grüne and Pannek, 2017).

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

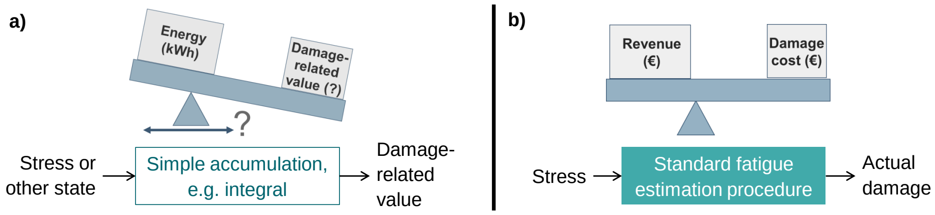


Figure 2. Indirect fatigue metric (a). Direct fatigue metric (b).

3.1 Indirect fatigue metrics in MPC

Several approaches reported in the literature involve indirect fatigue metrics (Barradas-Berglind and Wisniewski, 2016; Gros and Schild, 2017; Evans et al., 2015). However, indirect fatigue metrics have two main disadvantages:

- Instead of actual damage, only a damage-related value is obtained and optimized, as illustrated in Fig. 2a.
- Indirect fatigue terms have different units from harvested energy. Thus, weighting both terms in the cost function is not straightforward.

When considering tower fatigue, the most common approach involves the quadratic penalization of tower tip deflection rate $\dot{d}_\mathrm{T}$. This also can be interpreted as a penalization of kinetic energy of the lumped tower mass m_T, averaged over the prediction horizon T_horiz. In the present work, therefore, the stage cost

$$J_\mathrm{fatigue,TTVP} = \int_{t_0}^{t_\mathrm{end}} \frac{1}{2T_\mathrm{horiz}P_\mathrm{g,max}} m_\mathrm{T}\dot{d}_\mathrm{T}^2 \mathrm{d}t \tag{3}$$

is used for comparison and referred to as *Tower Tip Velocity Penalization (TTVP)*. An additional division by rated power $P_\mathrm{g,max}$ is used for scaling the cost, which is beneficial for optimization.

3.2 Direct fatigue metrics in MPC

In contrast to indirect fatigue metrics, direct fatigue metrics return actual damage, which can be readily converted into monetary fatigue cost, as visualized in Fig. 2b. This conversion is achieved for instance by multiplication with the Initial Capital Cost of the respective component or the entire turbine. Since harvested energy also can be converted into revenue by electricity price, the optimization algorithm can directly maximize profit.

As shown in Sect. 2, direct fatigue estimation involves the Rainflow algorithm. Implementation of RFC within a gradient-based optimization seemed impossible until now due to the following obstacles:

- RFC is a function of all stress samples. Therefore, neither the concept of stage nor of terminal cost applies.
- RFC contains branches. Therefore, it exhibits discontinuous outputs and is not continuously differentiable.

- RFC contains "while" loops, which lead to a changing function execution structure depending on the stress input.

Thus, in all known references, the Rainflow algorithm is approximated to some extent. In Sanchez et al. (2015), a version of *Simple Range Counting* is applied, which is standardized in ASTM (1985). In Barradas-Berglind et al. (2015), hysteresis operators are used to adapt parameters of a cost function in MPC. This cost function penalizes deflection rates, comparable to TTVP. In Luna et al. (2020), damage estimation including standard RFC is performed on a large number of stress time series, which are used to train a surrogate Artificial Neural Network (ANN). The latter seems to be very promising in terms of correct damage estimation. However, the approach involves a high a priori effort in setting up the ANN, as well as a significantly increased computational load in the MPC (Luna et al., 2020).

Stress history is not included in any of these approaches. In Barradas-Berglind et al. (2015), the hysteresis operators only have memory of damage evolution. Similarly, in Luna et al. (2020), only the previous fatigue rate output of the ANN is memorized until the next evaluation.

3.3 Parametric Online Rainflow Counting - Concept

The above mentioned obstacles for a direct implementation of RFC in MPC are overcome by the method of *Parametric Online Rainflow Counting (PORFC)*. In PORFC, all discontinuous parts of the fatigue estimation procedure are carried out before each execution of the MPC algorithm, as shown in Fig. 3. Additionally, the stress history is incorporated via a residue, which is inspired by the batchwise cycle identification in Sect. 2.3. The algorithmic workflow is as follows:

- **Simulation:** The reduced wind turbine model is simulated over the prediction horizon using the current measured states as initial values to produce a stress prediction, as visualized in Fig. 4b.

- **Merge:** The residue (see Fig. 4a) is merged with the stress prediction.

- **Rainflow:** The Rainflow algorithm is used to identify stress cycles over this merged trajectory. Consequently, it is assumed that the structure of identified cycles does not change within the next optimization. The term "structure" denotes here positions ($k_{\mathrm{min,c}}$, $k_{\mathrm{max,c}}$) and weights (w_{c}) of cycles. As shown in Fig. 4b, this assumption implies that the controllable extrema in the prediction horizon only can be shifted vertically (i.e., in the values but not in their positions) by the optimization.

- **Residue update:** Stress cycles can be composed by stress samples only from residue or prediction, or by a combination of both ("mixed cycle"). However, only the samples within the prediction horizon can be controlled by the optimization. Particularly the measured initial value at prediction step 0 cannot be controlled and, therefore, is added to the residue. If a full cycle is detected entirely within the *residue*, both contributing values are discarded from the residue. The reason for this is that also in the future they will never anymore form a cycle with a sample from the prediction and, therefore, are irrelevant for the MPC.

- **Time-varying parameters:** Information from cycle identification is used to fill vectors of time-varying parameters, which are forwarded to the cost function of the MPC. Details on this step are provided in Sect. 3.4.

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

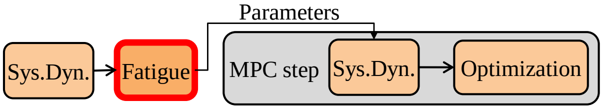


Figure 3. Externalization of fatigue estimation (Rainflow algorithm) from the MPC.

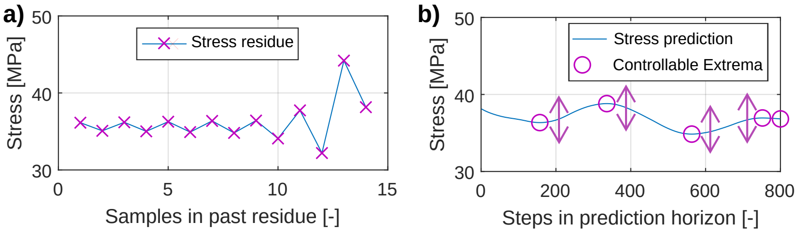


Figure 4. Stress residue from the past (a). Stress prediction into the future (b).

– **Optimization/MPC:** In the cost function of the MPC, the parameters are used to time-continuously calculate fatigue cost over the horizon and accumulate it via integration. Finally, the optimization problem is solved and the resulting control variables are applied to the wind turbine plant.

155 3.4 Parametric Online Rainflow Counting - Time-varying parameters and cost function

3.4.1 Distribution of damage over time

Since information from cycle identification is forwarded to the MPC via parameters, which are varying over the prediction horizon, the total fatigue damage has to be distributed over the prediction horizon, as visualized in Fig. 5b. Therefore, the damage of each stress cycle is split into two halves, which are allocated to the two contributing stress samples. For example, cycle 4 is formed by samples $k = 4$ and $k = 8$. Their fatigue cost terms therefore are allocated to these samples, as shown by the blocks in Fig. 5b. This example also shows an important property of the Rainflow algorithm, which identifies cycle 4 even though it is interrupted by the nested cycle 2, as shown in Fig. 5a. If, for a given stress sample, the complementary stress sample is not controllable (i.e. lies in the residue), all damage is allocated to the given sample. Here, this is the case for cycles 1 and 3, where all damage is allocated to sample $k = 2$ and $k = 4$, respectively.

165 3.4.2 Setup of the time-varying parameters

Figure 5a visualizes the generation of the time-varying parameters. Since each stress extremum belongs to one or two stress cycles (Shi et al., 2018), one or two stress references are set per extremum. These stress references are considered as optimization- or tracking-references for the current MPC-step. If both stress samples of a cycle lie in the prediction, mean stresses (M2, M4) become the stress references. If the complementary stress sample of a cycle lies in the uncontrollable residue ("mixed cycle"),

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

this complementary stress value (C1, C3) becomes the stress reference for the considered sample in the controllable prediction. However, in many cases, a mixed cycle is crossing the level of the initial stress $\sigma(t_0)$. In this case, the best possible tracking-reference is this initial stress value itself (R1, R3), since zero oscillation in the prediction corresponds to zero fatigue cost. A more detailed derivation and explanation can be found in Loew et al. (2020a).

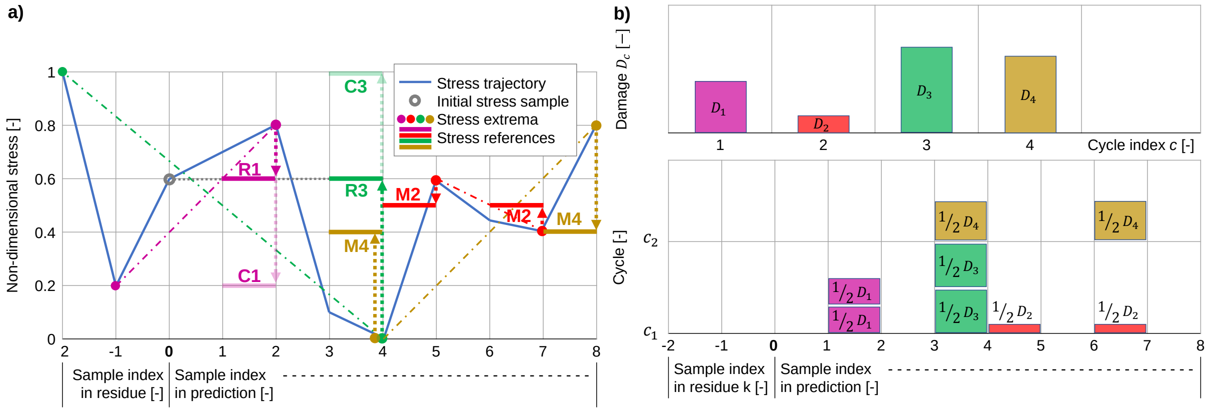


Figure 5. Stress trajectory (blue), its initial value at t_0 (grey circle), its extrema (colored dots), sequence of samples that form a cycle (dash-dotted), generated time-varying reference stresses (solid purple, red, green, yellow), and optimization goals (dotted arrows) for PORFC (a). Corresponding distribution of damage over the prediction horizon (b). Both figures are modified from Anand (2020).

3.4.3 Cost function

The fatigue cost function is defined by an integral over two cost terms, each one representing one potential cycle contribution of a stress sample, i.e.:

$$J_{\text{fatigue,PORFC}}(\sigma,\bar{\boldsymbol{p}}) = \frac{1}{T_{\text{cntrl}}}\int_{t_0}^{t_{\text{end}}} \left(J_{\text{fatigue},c}(\sigma(t),\hat{\bar{\sigma}}_{\text{ref},c1}(t),\hat{\bar{w}}_{c1}(t)) + J_{\text{fatigue},c}(\sigma(t),\hat{\bar{\sigma}}_{\text{ref},c2}(t),\hat{\bar{w}}_{c2}(t))\right)\,\mathrm{d}t \quad [€]. \tag{4}$$

The notation $\hat{(\cdot)}$ means fixed for one MPC-step, while $\bar{(\cdot)}$ means sampled on the control intervals of the prediction horizon. The cost terms are "switched on" by nonzero cycle weights $\hat{\bar{w}}_{c1/2}(t)$. Reference stresses $\hat{\bar{\sigma}}_{\text{ref},c1/2}(t)$ and cycle weights $\hat{\bar{w}}_{c1/2}(t)$ are collected in the parameter vector

$$\bar{\boldsymbol{p}} = \left(\hat{\bar{\sigma}}_{\text{ref},c1}, \hat{\bar{\sigma}}_{\text{ref},c2}, \hat{\bar{w}}_{c1}, \hat{\bar{w}}_{c2}\right), \tag{5}$$

which is defined as piecewise constant over the control intervals of the prediction horizon. The cost of individual cycles is defined by

$$J_{\text{fatigue},c} = \hat{\bar{w}}_{c1/2}(t)\; a_{\text{m}}\; |\sigma(t) - \hat{\bar{\sigma}}_{\text{ref},c1/2}(t)|^{m}, \tag{6}$$

where the fatigue coefficient a_{m} and the fatigue exponent m are derived from the damage curve of Eq. (2).

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

3.5 Optimization problem for TTVP and PORFC

The Economic MPC of a wind turbine is defined by the following optimization problem

$$\min_{\bar{\boldsymbol{u}},\bar{\boldsymbol{s}}} \left(-\alpha_{\text{revenue}} J_{\text{revenue}} + \alpha_{\text{fatigue}} J_{\text{fatigue}} + \int_{t_0}^{t_{\text{end}}} \left(10^{-1} \dot{\beta}_{\text{b}}^2 + 10^{-2} \dot{T}_{\text{g}}^2 + 10^7\, \bar{s}_{\omega}^2 + 10^7\, \bar{s}_P^2 \right) \mathrm{d}t \right), \tag{7}$$

which maximizes the revenue J_{revenue} and minimizes the fatigue J_{fatigue}, which is represented by Eq. (3) for TTVP or Eq. (4) for PORFC. The constants α_{revenue} and α_{fatigue} are weighting factors. Instead of generated electrical energy, harvested aerodynamic energy J_{revenue} is maximized to avoid a greedy extraction of rotor kinetic energy by the MPC (*turnpike effect*), as suggested by Gros and Schild (2017). Furthermore, pitch rate $\dot{\beta}_{\text{b}}$, torque rate $\dot{T}_{\text{g}}$, and slack variables for rotational speed $\bar{s}_{\omega}$ and generator power $\bar{s}_P$ are penalized (see their use in the constraints below). The optimization variables are the demanded pitch angle and torque rate $\bar{\boldsymbol{u}} = (\bar{\boldsymbol{\beta}}_{\text{b,d}}, \dot{\bar{\boldsymbol{T}}}_{\text{g,d}})$, and the slack variables $\bar{\boldsymbol{s}} = (\bar{\boldsymbol{s}}_{\omega}, \bar{\boldsymbol{s}}_P)$.

For both TTVP and PORFC, revenue is weighted by the current electricity price $\alpha_{\text{revenue}} = p_{\text{elec}}$ [€W^{-1} s^{-1}] to match the monetary nature of Eq. (4). The fatigue weight α_{fatigue} remains free and will be determined later in this work.

The optimization problem is subject to:

- The system dynamics of a reduced turbine model $\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t))$, whose six states

  $$\boldsymbol{x}(t) = (\omega_{\text{r}}(t), d_{\text{T}}(t), \dot{d}_{\text{T}}(t), \beta_{\text{b}}(t), \dot{\beta}_{\text{b}}(t), T_{\text{g}}(t))^T \tag{8}$$

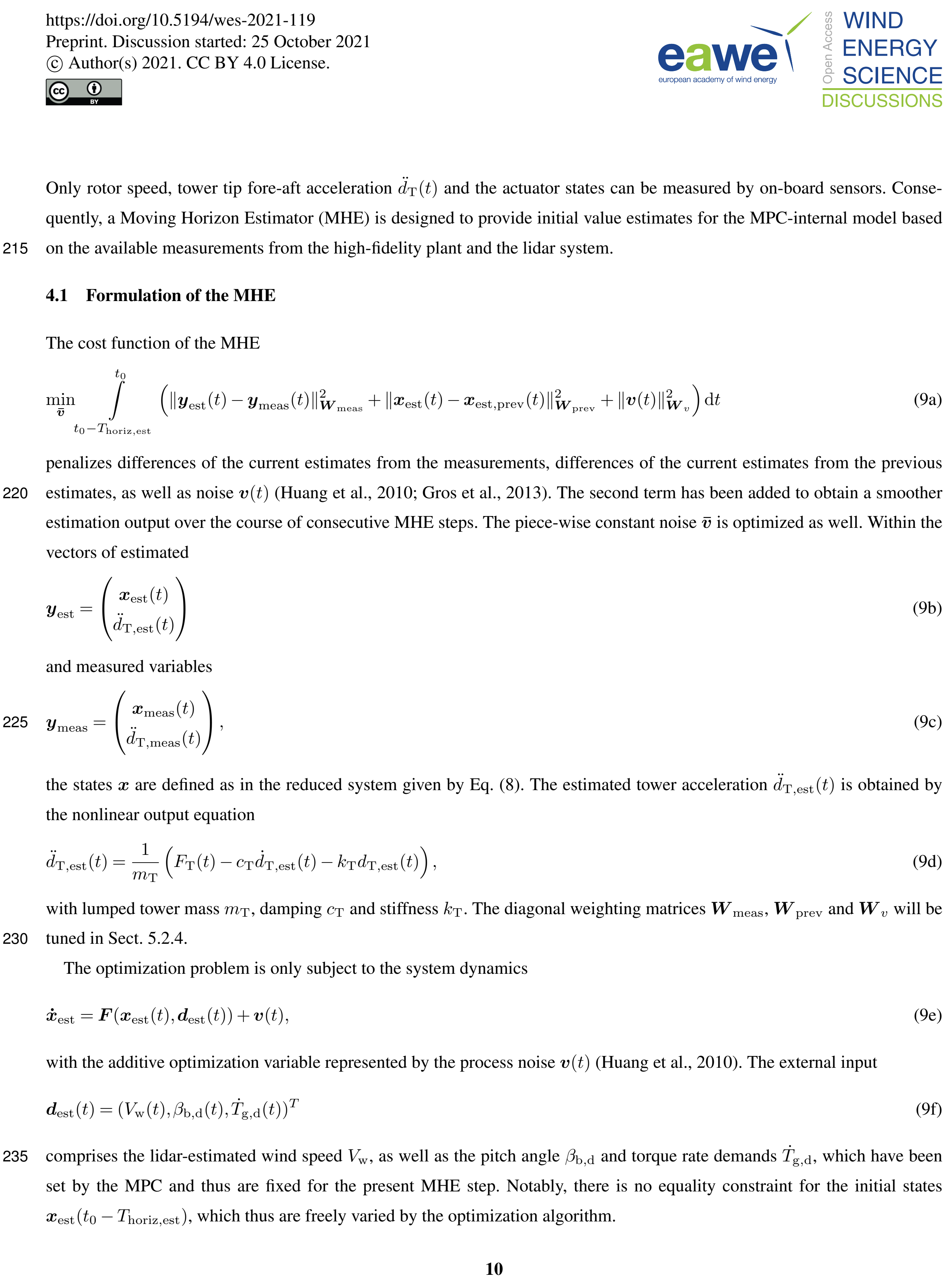
  are rotational speed of the rotor noted ω_{r}, tower tip deflection d_{T}, tower tip velocity $\dot{d}_{\text{T}}$, pitch angle β_{b}, pitch rate $\dot{\beta}_{\text{b}}$, and generator torque T_{g}. More details about the model are given in Löw and Obradovic (2018).

- Inequality constraints over the horizon, to keep rotational speed, tower deflection (yield strength), pitch angle, pitch rate, generator torque, and generator power within their limits. In order to maintain feasibility of the optimization despite model uncertainties and temporary constraint violations, the constraints on rotational speed and generator power are augmented by slack variables, as suggested by Gros (2013).

- Box constraints on control and slack variables.

4 Moving Horizon Estimator

The MPC-internal system model only comprises the 6 states defined by Eq. (8), while the plant model in `OpenFAST` (including the actuators, but excluding the yaw mechanism) comprises 33 states (8 tower states, 6 states for each of the 3 blades, 2 states for drive-shaft torsion, 2 states for rotor rotation, 2 states for the collective blade pitch actuation, 1 state for the generator torque actuation) (Jonkman et al., 2009). Thus, the MPC-internal model is only a reduced representation of the plant model. Furthermore, both the tower deflection and velocity of the MPC-internal model cannot be measured directly on a real turbine.



https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

Only rotor speed, tower tip fore-aft acceleration $\ddot{d}_{\mathrm{T}}(t)$ and the actuator states can be measured by on-board sensors. Consequently, a Moving Horizon Estimator (MHE) is designed to provide initial value estimates for the MPC-internal model based on the available measurements from the high-fidelity plant and the lidar system.

4.1 Formulation of the MHE

The cost function of the MHE

$$\min_{\bar{\boldsymbol{v}}} \int\limits_{t_0-T_{\mathrm{horiz,est}}}^{t_0} \left(\|\boldsymbol{y}_{\mathrm{est}}(t)-\boldsymbol{y}_{\mathrm{meas}}(t)\|^2_{\boldsymbol{W}_{\mathrm{meas}}} + \|\boldsymbol{x}_{\mathrm{est}}(t)-\boldsymbol{x}_{\mathrm{est,prev}}(t)\|^2_{\boldsymbol{W}_{\mathrm{prev}}} + \|\boldsymbol{v}(t)\|^2_{\boldsymbol{W}_v}\right)\mathrm{d}t \tag{9a}$$

penalizes differences of the current estimates from the measurements, differences of the current estimates from the previous estimates, as well as noise $\boldsymbol{v}(t)$ (Huang et al., 2010; Gros et al., 2013). The second term has been added to obtain a smoother estimation output over the course of consecutive MHE steps. The piece-wise constant noise $\bar{\boldsymbol{v}}$ is optimized as well. Within the vectors of estimated

$$\boldsymbol{y}_{\mathrm{est}} = \begin{pmatrix} \boldsymbol{x}_{\mathrm{est}}(t) \\ \ddot{d}_{\mathrm{T,est}}(t) \end{pmatrix} \tag{9b}$$

and measured variables

$$\boldsymbol{y}_{\mathrm{meas}} = \begin{pmatrix} \boldsymbol{x}_{\mathrm{meas}}(t) \\ \ddot{d}_{\mathrm{T,meas}}(t) \end{pmatrix}, \tag{9c}$$

the states $\boldsymbol{x}$ are defined as in the reduced system given by Eq. (8). The estimated tower acceleration $\ddot{d}_{\mathrm{T,est}}(t)$ is obtained by the nonlinear output equation

$$\ddot{d}_{\mathrm{T,est}}(t) = \frac{1}{m_{\mathrm{T}}}\left(F_{\mathrm{T}}(t) - c_{\mathrm{T}}\dot{d}_{\mathrm{T,est}}(t) - k_{\mathrm{T}}d_{\mathrm{T,est}}(t)\right), \tag{9d}$$

with lumped tower mass m_{T}, damping c_{T} and stiffness k_{T}. The diagonal weighting matrices $\boldsymbol{W}_{\mathrm{meas}}$, $\boldsymbol{W}_{\mathrm{prev}}$ and $\boldsymbol{W}_v$ will be tuned in Sect. 5.2.4.

The optimization problem is only subject to the system dynamics

$$\dot{\boldsymbol{x}}_{\mathrm{est}} = \boldsymbol{F}(\boldsymbol{x}_{\mathrm{est}}(t), \boldsymbol{d}_{\mathrm{est}}(t)) + \boldsymbol{v}(t), \tag{9e}$$

with the additive optimization variable represented by the process noise $\boldsymbol{v}(t)$ (Huang et al., 2010). The external input

$$\boldsymbol{d}_{\mathrm{est}}(t) = (V_{\mathrm{w}}(t), \beta_{\mathrm{b,d}}(t), \dot{T}_{\mathrm{g,d}}(t))^T \tag{9f}$$

comprises the lidar-estimated wind speed V_{w}, as well as the pitch angle $\beta_{\mathrm{b,d}}$ and torque rate demands $\dot{T}_{\mathrm{g,d}}$, which have been set by the MPC and thus are fixed for the present MHE step. Notably, there is no equality constraint for the initial states $\boldsymbol{x}_{\mathrm{est}}(t_0 - T_{\mathrm{horiz,est}})$, which thus are freely varied by the optimization algorithm.

After the execution of the MHE, the terminal states at the end of the MHE estimation horizon become the initial states at the beginning of the MPC prediction horizon:

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_{\mathrm{est}}(t_0). \tag{10}$$

In the present controller setup, the optimized noise $\boldsymbol{v}(t)$ of the MHE is not utilized in the MPC, and its role is limited to the improvement of the quality of the estimates by taking into account process noise (which, in this context, also includes model errors).

4.2 Initialization of the MHE

The MHE requires information about the measurements over its entire estimation horizon. Therefore, the past measurements $\boldsymbol{y}_{\mathrm{meas}}$ are buffered.

As mentioned above, the tower deflection and velocity are not measured. However, the MHE optimization benefits from meaningful measurement values as tracking reference. Therefore, a static wind-to-tower-deflection mapping is interpolated over the lidar wind estimate in order to generate a proxy tower deflection trajectory $d_{\mathrm{T,meas}}$. Tower velocity $\dot{d}_{\mathrm{T,meas}}$ is obtained by the numerical time-derivative of the deflection trajectory. These quantities are termed "lidar-based references" in the remainder of this work.

5 Simulation setup, tuning and results

In the following, the simulation setup is presented, each element of the controller toolchain is tuned, and the simulation results are discussed.

5.1 Simulation setup

5.1.1 Plant model

The designed MPC formulations are tested with the NREL 5MW onshore reference turbine (Jonkman et al., 2009) in the aeroelastic simulator `OpenFAST`. This turbine has a hub height of 110m and a rotor diameter $D = 126$m. All mechanical degrees of freedom (DOF) are activated.

5.1.2 Wind model

All turbulent results in this work are mean values of 12 simulations (each with a different seed) of 600s length in DLC 1.2 with category A turbulence. For the chosen turbine with a hub height of 110m and coastal onshore setting, a mean annual wind speed of $V_{\mathrm{w,mean}} = 7\mathrm{ms}^{-1}$ can be assumed (Hau, 2017). Thus, in the turbulent simulations the probability of wind speed is assumed to follow the Rayleigh distribution

$$p(V_{\mathrm{w}}) = \frac{\pi}{2}\left(\frac{V_{\mathrm{w}}}{V_{\mathrm{w,mean}}^2}\right)\exp\left[-\frac{\pi}{4}\left(\frac{V_{\mathrm{w}}}{V_{\mathrm{w,mean}}}\right)^2\right], \tag{11}$$

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

Open Access WIND ENERGY SCIENCE DISCUSSIONS

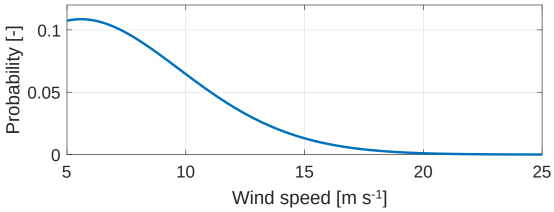


Figure 6. Rayleigh probability density function for a mean wind speed of $V_{\mathrm{w,mean}} = 7\mathrm{ms}^{-1}$.

as shown in Fig. 6. The Rayleigh distribution is a variant of the Weibull distribution with the simplification of having only a single parameter (Manwell et al., 2008).

5.1.3 Lidar simulator

The model of a pulsed lidar with four beams is employed. The model is implemented in the lidar simulator from `sowento GmbH`, which generates lidar wind estimates offline, and thus independently from the wind turbine simulation suite (Raach and Schlipf, 2018). Considered physical effects are: limitation to line-of-sight wind speeds, spatial averaging via a Gaussian range weighting function, discrete scanning, and "unfrozen" wind evolution. Particularly the wind evolution can be parameterized by an exponential decay constant; here, a higher value results in higher variation of the wind during its convection towards the rotor. Finally, the spatially distributed measurements are converted to rotor-effective wind speed by wind field reconstruction.

5.1.4 MHE/MPC framework

The MHE and MPC are implemented in the state-of-the-art `acados` framework (Verschueren et al., 2021), using the interior-point solver `HPIPM` for the underlying Quadratic Programs (QP).

5.1.5 Controller variants

In the following, the performance of five MPC formulations and the baseline conventional controller (CC) from NREL (Jonkman et al., 2009) are compared. The MPCs involve the conventional formulation of TTVP (see Sect. 3.1) and the novel formulation of PORFC (see Sect. 3.3). For PORFC, a fatigue exponent of $m = 2$ (see Eq. (6)) is utilized, which, resulting in quadratic cost functions, is very suitable for Quadratic Programming. This case is assessed in combination with (PORFC-2R) and without (PORFC-2) the use of residue. Additionally, a fatigue exponent of $m = 5$ is tested (PORFC-5), which is present at low stress amplitudes in the actual S-N-curve of the tower material. This parameterization in combination with residue has not led to satisfactory results, and thus is not considered further.

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

5.1.6 Performance indicators

Considered performance indicators are revenue (analog to energy), fatigue cost (based on a realistic piecewise S-N-curve, profit (revenue subtracted by fatigue cost), pitch travel, and torque travel.

5.2 Tuning

Each element of the present controller toolchain (lidar simulator, lidar processing, Moving Horizon Estimator, Model Predictive Controller) comprises a set of tunable parameters, which all impact the control performance. A comprehensive overview of these parameters is provided in Schlipf et al. (2018). Instead of tuning all parameters at once (monolithic approach), the sequential approach of Schlipf et al. (2018) is pursued. Here, the elements are tuned sequentially according to their individual performance criteria.

5.2.1 Tuning of lidar simulator

In Schlipf et al. (2018), the same wind turbine plant model (NREL 5MW onshore) and lidar simulator are used as in the present work. There, the parameters of the lidar simulator are tuned in order to maximize the measurement coherence bandwidth for the rotor-effective wind speed. In other words, the smallest detectable eddy size is maximized, reaching a value of $D_{\mathrm{eddy,min}} = 1.58D = 199$m. Since control of fatigue - but not lidar tuning - is the focus of the present work, the parameters of the lidar simulator in Table 1 are adopted from Schlipf et al. (2018). This "Default lidar" scenario with a low decay constant of 0.1 is accompanied by a "High decay lidar" scenario, where the exponential decay constant is increased to 0.4 (Schlipf, 2016).

Table 1. Parameters of lidar simulator for the "Default lidar" and "High decay lidar" scenarios.

Parameter	Unit	"Default lidar"	"High decay lidar"
Type of lidar	-	Pulsed	Pulsed
Opening angle	[deg]	11.3	11.3
Distance to closest scanning plane $d_{\mathrm{scan,close}}$	[m]	40	40
Distance to farthest scanning plane	[m]	280	280
Number of scanning planes	[-]	10	10
Scanning rate f_{scan}	[Hz]	4	4
Gaussian - Full width at half maximum	[m]	20	20
Gaussian - Evaluation points	[m]	-10,0,10	-10,0,10
Exponential decay constant of turbulent wind	[-]	0.1	**0.4**

5.2.2 Tuning of lidar processing

The raw rotor-effective wind speed from the lidar simulator has to be buffered in order to compensate time delays, and filtered to remove uncorrelated high-frequency information. The buffer and filter parameters have to be tuned. In Schlipf et al. (2018),

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

Open Access WIND ENERGY SCIENCE DISCUSSIONS

the tuning of the lidar processing is performed using the reduced wind turbine model as plant model for performance reasons. However, simulations in Loew et al. (2019) have shown that unrealistically high fatigue reduction is possible if the MPC-internal and the plant model are matching. Thus, in the present work, the high-fidelity OpenFAST model is used for tuning of the lidar processing, in order to benefit from its realistic fatigue behavior.

Buffering: Inspired by Schlipf (2016), the raw rotor-effective wind speed is buffered by an adaptive buffer time span of

$$T_{\text{buffer}}(t) = T_{\text{travel}}(t) - 0.5T_{\text{scan}} - T_{\text{filter}}. \tag{12}$$

Here, the traveling time $T_{\text{travel}}(t) = \frac{d_{\text{travel}}(t)}{V_{\text{mean}}(t)}$ from the closest scanning plane to the rotor is obtained by the traveling distance d_{travel} and the current mean wind speed V_{mean}. The traveling distance $d_{\text{travel}}(t) = d_{\text{scan,close}} + d_{\text{T}}(t)$ is the nominal distance corrected by the current tower tip deflection. The total scan time $T_{\text{scan}} = \frac{1}{f_{\text{scan}}}$ is obtained from the scanning rate. The filter delay T_{filter} is zero here, since a zero-phase filter is utilized.

Filtering: Since the lidar correlation varies with wind speed (Schlipf, 2016), the uncorrelated high-frequency information has to be processed by a low-pass filter, which is adaptive as well. In order to avoid the above mentioned filter-delay compensation, only zero-phase algorithms have been considered. Particularly, a zero-phase forward-backward IIR-filter (function `filtfilt` in `MATLAB`) has been compared to a central moving mean filter (function `movmean` in `MATLAB`). For this purpose, the lidar simulator parameterization from Sect. 5.2.1 and a reasonable initial parameterization of the MHE and MPC have been utilized in turbulent simulations. Despite its simplicity, with different MPC formulations and horizon lengths, the moving mean filter has exhibited superior economic profit, and thus has been chosen for the present application. Beneficially, the moving mean filter only requires tuning of its window length. The empirical formula

$$T_{\text{movmean}}(t) = \frac{D_{\text{eddy,min}}}{V_{\text{mean}}(t)}, \tag{13}$$

which is based on the smallest detectable eddy size and the current mean wind speed, has led to a very good adaptive tuning for the present setup.

Due to its nature, the central moving mean filter requires sufficient information from the past and future. Except for the beginning of a simulation, the amount of past information typically is sufficient and even growing in the course of simulation time. In contrast, sufficient future information beyond the prediction horizon is only ensured if the inequality

$$T_{\text{pred}}(t) - 0.5T_{\text{movmean}}(t) \geq T_{\text{horiz}} \tag{14}$$

holds, where the lidar-predicted time T_{pred} subtracted by half the moving mean filter length T_{movmean} exceeds the MPC horizon length T_{horiz}. The lidar-predicted time $T_{\text{pred}}(t) = d_{\text{scan,far}}/V_{\text{mean}}(t)$ depends on the distance of the farthest scanning plane to the rotor and the current mean wind speed. As shown in Fig. 7, the inequality typically holds and is only slightly violated at $V_{\text{mean}} > 22\text{ms}^{-1}$.

Further scaling of buffer and filter parameters: In order to verify the above adaptation formula, simulations have been executed where these buffer and filter window lengths are increased or decreased. These results are generated for the "High decay lidar" scenario, where lidar data quality is lower and thus correct filtering is more important. The simulations reveal that

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

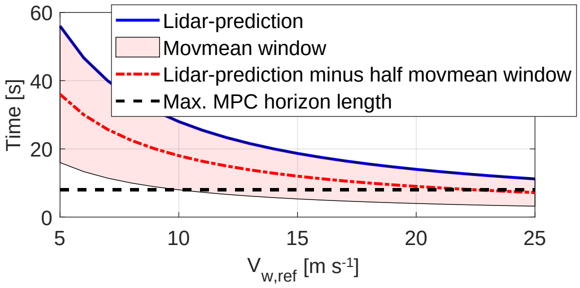


Figure 7. Comparison of the effectively available filtered prediction information (red dash-dotted) to the MPC horizon length (black dashed).

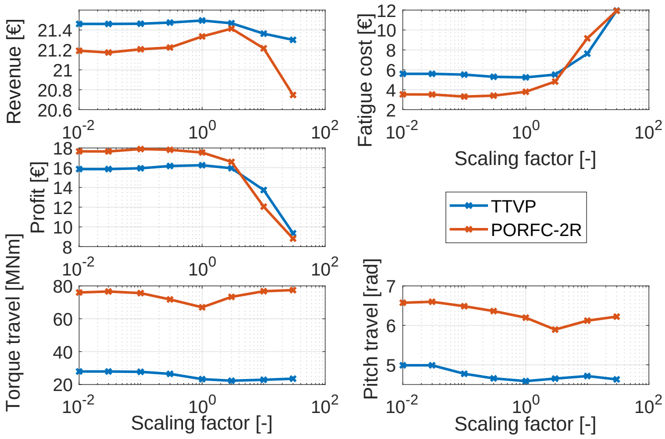


Figure 8. Variation of KPIs depending on the scaling of moving mean filter window length.

the above adaptation laws lead already to optimal or near-optimal profit results. As shown exemplary for the filter window length in Fig. 8, maximum profit is achieved with a scaling factor of 1 for TTVP, and 0.1 for PORFC. Lower scaling factors (shorter filter windows) have a tendency towards higher actuator usage, while higher scaling factors dramatically increase fatigue cost. Consequently, scaling of the buffer and filter parameters is not applied in the present work. 340

5.2.3 Tuning of the MHE algorithm

The MHE is set up with an estimation horizon length of 8s and a sample time of 0.1s, which are fixed for all MPC configurations.

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

WIND ENERGY SCIENCE DISCUSSIONS

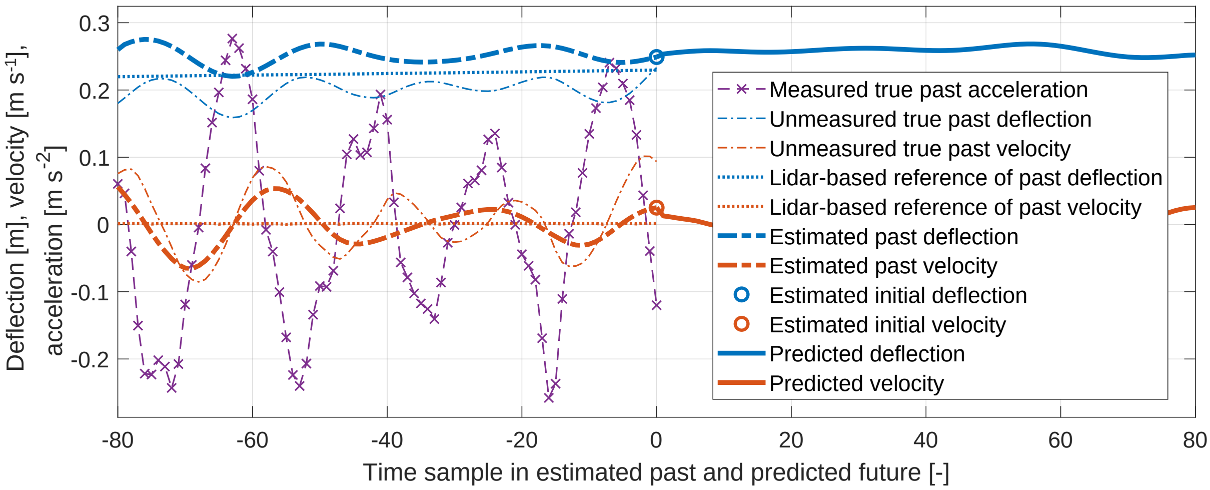


Figure 9. Wind turbine tower quantities ordered by occurrence in the estimation and control process: measured, unmeasured, lidar-based reference (see Sect. 4.2), estimated, and predicted. Negative time samples = estimation horizon of the MHE; positive time samples = prediction horizon of the MPC.

5.2.4 Tuning of the MHE cost weights

Practical experience has shown that the goal for the MHE should not be an accurate reconstruction of the unmeasured true plant states of tower tip deflection and velocity, as shown in Fig. 9. In fact, an accurate reconstruction would contain high frequencies from the plant higher-order degrees of freedom, which would be attributed to the lower-order degrees of freedom in the MPC-internal model.

Instead, for the low-frequency MPC-internal model also low-frequency initial states should be provided by the MHE ("estimated initial" states). Consequently, the MHE is tuned to estimate state trajectories that best fit the behavior of the *reduced model*. This is achieved by setting low weights for the tower variables in the weighting matrix $\boldsymbol{W}_{\mathrm{meas}}$, as shown in Table 2. These low weights allow for significant deviations from the measured tower acceleration and the lidar-based references, and thus for a greater focus on the reduced model dynamics. For tower deflection and velocity, very low values of 10^{-4} are chosen, since these quantities are not measured. For tower acceleration, an intermediate value of 10^{-2} is chosen, since it is measured but its trajectory does not need to be tracked carefully.

By the intermediate weights in the weighting matrix $\boldsymbol{W}_{\mathrm{prev}}$, the current state trajectories are only loosely tied to the previous ones.

In the weighting matrix $\boldsymbol{W}_{v}$, significant noise is only permitted for the pitch angle and torque rate, in order to enable a close match of pitch and torque estimations with their already accurate measurements.

Table 2. Diagonal elements of the weighting matrices $\boldsymbol{W}$ penalizing the corresponding entries of the estimation $\boldsymbol{y}_{\text{est}}$ and noise vector $\boldsymbol{v}$.

	$\omega_{\text{r}}(t)$	$d_{\text{T}}(t)$	$\dot{d}_{\text{T}}(t)$	$\beta_{\text{b}}(t)$	$\dot{\beta}_{\text{b}}(t)$	$T_{\text{g}}(t)$	$\ddot{d}_{\text{T}}(t)$
$\boldsymbol{W}_{\text{meas}}$	1	10^{-4}	10^{-4}	1	1	1	10^{-2}
$\boldsymbol{W}_{\text{prev}}$	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}	-
$\boldsymbol{W}_{v}$	1	1	1	1	10^{-3}	10^{-3}	-

5.2.5 Tuning of the MPC algorithm

The controller sample time is set to 0.1s like in Bottasso et al. (2014) and Gros and Schild (2017). The maximum horizon length of $T_{\text{horiz,max}} = 8$s is chosen based on the findings of Loew et al. (2020b), which indicate that a considerable portion of the plant stress cycles will be contained in this prediction horizon. However, since horizon length has a substantial impact on performance and - moreover - the longest horizon is not always the best, shorter horizons of $T_{\text{horiz}} = \{4; 2; 1\}$s will be tested throughout all turbulent studies.

One QP is solved per MPC step. The Hessian matrix is automatically convexified to account for possible numerical issues due to the highly non-standard cost formulation of PORFC. Practical experience has shown that performance is improved if the Newton step length of the QP is reduced from 1 to 0.1. This can be explained by the frequently changing optimization problems, especially for PORFC. In this case the initialization of the QP might not be sufficiently close to the optimum, and full Newton steps could leave the region of validity of the quadratic approximation of the Nonlinear Program (Diehl and Gros, 2020).

5.2.6 Tuning of the MPC cost weights

As shown in Sect. 3.5, all weights in the cost function except for the fatigue weight α_{fatigue} are pre-defined for simplicity. Thus, only the fatigue weight has to be tuned in the following. Tuning is executed at a single reference wind speed of $V_{\text{w,ref}} = 9\text{ms}^{-1}$. This wind speed is chosen since for the conventional controller, the highest profit contribution occurs there, as shown in terms of *profit density* in Fig. 10. Profit density represents the incremental contribution to total cumulative profit at a certain wind speed. Another meaningful criterion for a suitable tuning wind speed would be the wind speed where half of total cumulative profit is reached.

The variation of important KPIs with fatigue weights is shown in Fig. 11. For brevity, only the results for the maximum MPC horizon length of 8s are shown. All variants of PORFC exhibit higher profit than TTVP. Especially for PORFC-2R there is a profit plateau that spans over a wide range of fatigue weights. Thus, the tuning of PORFC-2R can be considered as less critical than that of TTVP. Since in most cases torque travel and pitch travel increase with fatigue weight, low fatigue weights should be preferred if profit is not harmed significantly. Following this strategy, the fatigue weights are determined for all controller formulations and horizon lengths, as shown in Table 3.

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

Open Access WIND ENERGY SCIENCE DISCUSSIONS

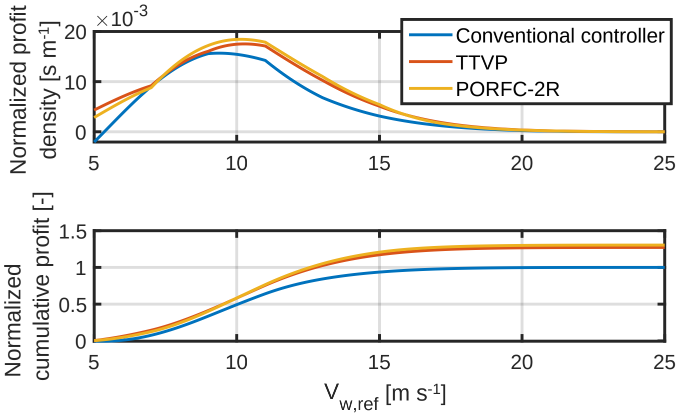


Figure 10. Normalized profit density and cumulative profit. The latter is normalized with respect to total profit of the conventional controller.

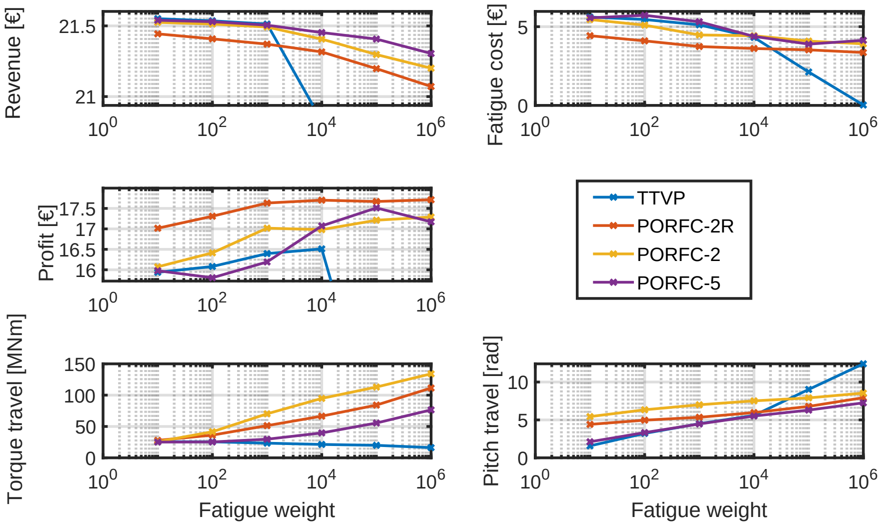


Figure 11. Variation of KPIs for different fatigue weights and a prediction horizon length of $T_{\mathrm{horiz}} = 8\mathrm{s}$.

Table 3. Optimum fatigue cost weights $\alpha_{\mathrm{fatigue}}$ for different controller formulations and prediction horizon lengths.

Controller formulation	$T_{\mathrm{horiz}} = 8\mathrm{s}$	$T_{\mathrm{horiz}} = 4\mathrm{s}$	$T_{\mathrm{horiz}} = 2\mathrm{s}$	$T_{\mathrm{horiz}} = 1\mathrm{s}$
TTVP	10^4	10^3	10^4	10^5
PORFC-2R	10^4	10^3	10^2	10^3
PORFC-2	10^6	10^3	10^4	10^4
PORFC-5	10^5	10^5	10^5	10^4

5.3 Results of turbulent simulations

5.3.1 Comparison of controller formulations in the "Default lidar" scenario

As a next step, the optimal tuning weights from Sect. 5.2.6 are fixed and simulations at different reference wind speeds $V_{\mathrm{w,ref}} = \{5; 7; ...; 25\}\mathrm{ms}^{-1}$ are performed for each controller formulation and prediction horizon length. The simulations result in the Weibull-weighted cumulative KPIs shown in Fig. 12.

Even for the best-performing MPCs, revenue remains below the one of CC. However, since fatigue of the MPCs is much lower, all MPCs with a prediction horizon of $T_{\mathrm{horiz}} \geq 2\mathrm{s}$ exhibit higher profit than CC. The highest profit gain is achieved by PORFC-2R at maximum prediction horizon, which surpasses CC by 30% and the best TTVP by 2.5%. At least for the present setting, very short horizons of 1s cannot be recommended, since they significantly decrease revenue and extremely increase fatigue.

Over different horizon lengths, PORFC-2R exhibits a very stable profit level. In contrast, for the PORFC formulations without residue (PORFC-2, PORFC-5), a shorter horizon of $T_{\mathrm{horiz}} = 4\mathrm{s}$ exhibits higher profit than $T_{\mathrm{horiz}} = 8\mathrm{s}$. This phenomenon may be explained by a higher influence of the prediction errors: due to model errors and wind evolution, the predicted states at the end $4\mathrm{s} < t \leq 8\mathrm{s}$ of a long horizon may be affected by large errors. Since PORFC-2/5 have to rely solely on the predictions, their performance may suffer from long horizons.

For all MPCs, the profit benefit with respect to CC via fatigue reduction comes at the price of a higher pitch travel. Here, TTVP exhibits around 6 times the pitch travel of CC. Since in the literature more moderate increases of e.g. factor 2 are reported (Luna et al., 2020), further studies on pitch penalization are planned. For PORFC, pitch travel is even slightly higher, while a reduction of horizon length also leads to a moderate reduction of pitch travel. Torque travel exhibits a different behavior, where several MPC formulations have a lower torque travel than CC.

A look at the profit density and cumulative profit in Fig. 10 shows that both MPC formulations (TTVP, PORFC) "earn money" very similarly over wind speed. Compared to CC, the MPC fatigue reduction strategies lead to impressive profit benefits at very low and at intermediate wind speeds. With the present tuning, TTVP has a slight extra advantage at very low wind speeds, while PORFC-2R is superior at a broad range of intermediate wind speeds.

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

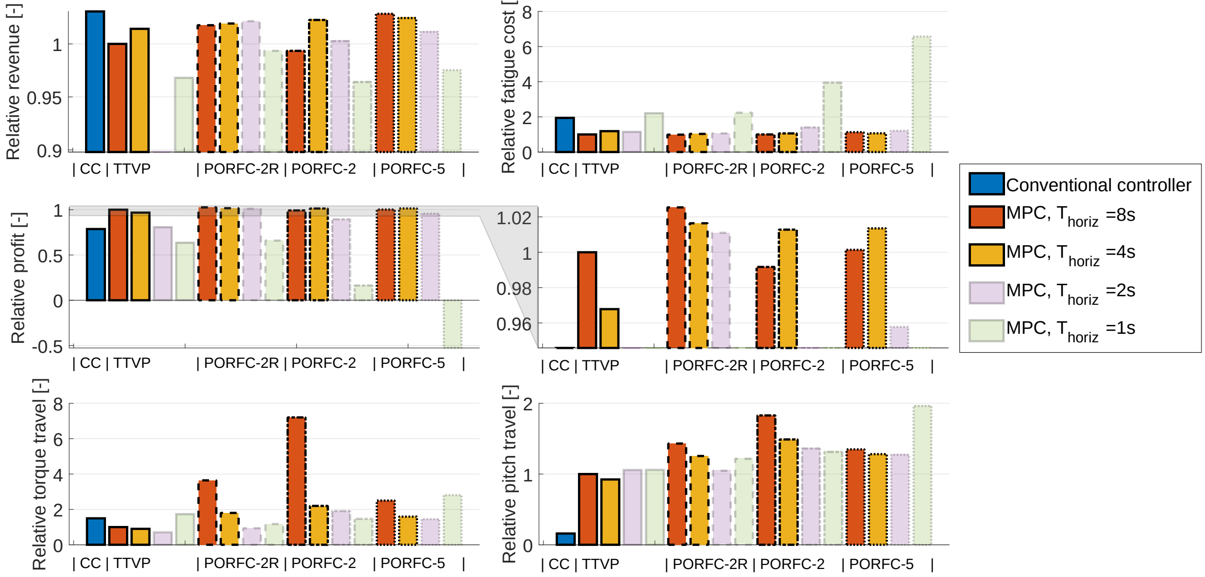


Figure 12. "Default lidar" scenario. Weibull-weighted KPIs for different controller formulations (indicated by edge style) and MPC prediction horizon lengths (indicated by color). The results for the shorter horizons are transparent in order to focus the attention on the more important longer horizons. Results are normalized with respect to the best TTVP configuration with a prediction horizon length of $T_{\mathrm{horiz}} = 8\mathrm{s}$. Middle row, right: Zoomed version of profit plot.

5.3.2 Performance in the "High decay lidar" scenario

The "Default lidar" scenario of the previous sections can be considered as very favorable for lidar-assisted control, since the wind does not change very much between the lidar measurement planes and the rotor. Thus, the lidar provides a fairly good estimate of the true incoming wind. In order to challenge the MPCs even more, a further assessment is performed for the "High decay lidar" scenario (see Table 1).

Despite the significant reduction of lidar signal quality, the profit benefit of the MPCs over CC decreases only slightly, as shown in Fig. 13. Particularly, the best-performing PORFC-2R ($T_{\mathrm{horiz}} = 4\mathrm{s}$) still surpasses CC by 26% for the "High decay lidar" scenario, in comparison to the above mentioned 30% for the "Default lidar" scenario.

The very strong profit benefit of 5.1% of the best PORFC-2R ($T_{\mathrm{horiz}} = 4\mathrm{s}$) over the best TTVP ($T_{\mathrm{horiz}} = 8\mathrm{s}$) shows that PORFC-2R is particularly strong in handling situations of low lidar data quality. In a direct comparison using the same horizon length ($T_{\mathrm{horiz}} = 4\mathrm{s}$), PORFC-2R even benefits by almost 9%. Just like in the "Default lidar" scenario, PORFC-2R exhibits better profit stability over the horizon lengths $T_{\mathrm{horiz}} = \{8; 4; 2\}\mathrm{s}$, and fails at $T_{\mathrm{horiz}} = 1\mathrm{s}$.

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

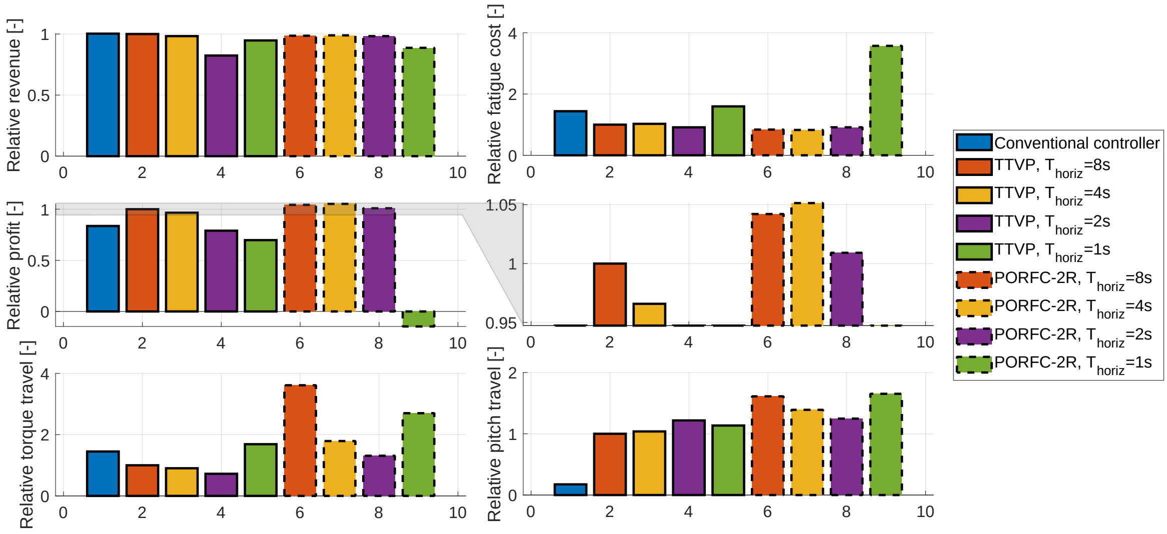


Figure 13. "High decay lidar" scenario. Weibull-weighted KPIs for different controller formulations and prediction horizon lengths. Results are normalized with respect to the best TTVP configuration with a prediction horizon length of $T_{\mathrm{horiz}} = 8$s. Middle row, right: Zoomed version of profit plot.

5.3.3 Performance in the "Perfect prediction" scenario

Increasing benefit of PORFC-2R with respect to TTVP with lower lidar data quality, conversely suggests decreasing benefits with very high lidar data quality. This hypothesis is actually partially confirmed by the extreme scenario of a perfect wind prediction (without lidar errors). As shown in Fig. 14, as expected, the performance of TTVP and PORFC-2R is significantly surpassing the performance from the "Default lidar" scenario of Sect. 5.3.1. However, relative to TTVP, the advantage of PORFC-2R decreases significantly. For the maximum horizon length, TTVP even slightly surpasses PORFC-2R by 0.4%. On the other hand, as the profit of PORFC-2R is more stable for shorter horizons, it retains a significant advantage there.

5.3.4 Benefit of "Perfect prediction" vs. "Perfect persistence"

All previous scenarios assumed a wind preview. However, to date, lidar systems can still account for a significant portion of the capital and operational expenditures of wind turbines (Canet et al., 2020). In order to avoid lidar-related costs or effort, some studies are directed towards predictive control without explicit preview measurement (Evans et al., 2015; Jassmann et al., 2016). In this case, the wind prediction over the MPC horizon can for instance be generated via constant extrapolation of the instantaneous wind estimate at the rotor ("persistence"). This motivates an analysis of how the novel PORFC MPC actually benefits from a predictive preview compared to a persistent preview.

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

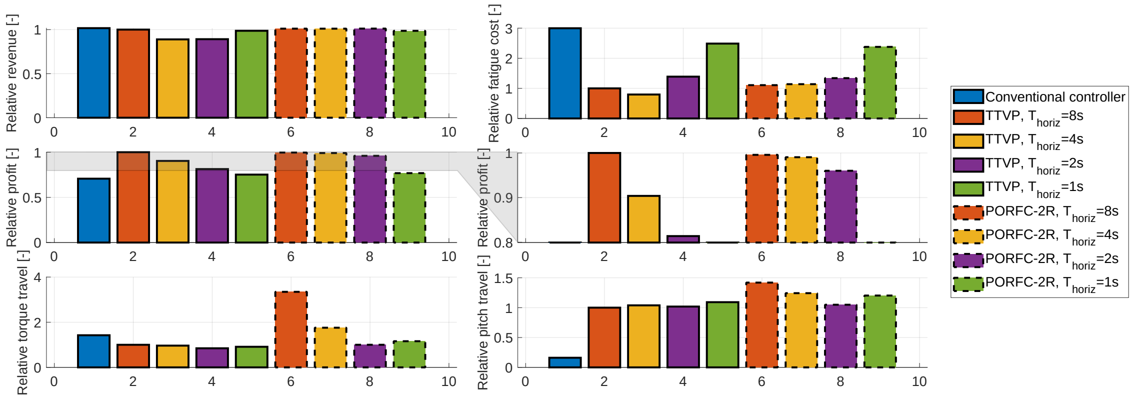


Figure 14. "Perfect prediction" scenario. Weibull-weighted KPIs for different controller formulations and prediction horizon lengths. Results are normalized with respect to the best TTVP configuration with a prediction horizon length of $T_{\mathrm{horiz}} = 8\mathrm{s}$. Middle row, right: Zoomed version of profit plot.

Since the design of a rotor-effective wind speed estimator is out of scope of the present work, a "Perfect persistence" scenario is employed and compared to the above "Perfect prediction" scenario.

As shown in Fig. 15, all MPC configurations - but especially PORFC-2R - significantly benefit from prediction (instead of persistence) by 5% to 25% of profit. For all configurations, this is primarily achieved by high fatigue reduction. At the same time, actuator usage is moderately decreased or even increased at some horizon lengths for TTVP, but is significantly decreased for PORFC-2R if $T_{\mathrm{horiz}} \geq 2\mathrm{s}$.

These results further indicate the technical benefit of lidar-assisted control, and motivate further studies comparing the realistic lidar wind preview with a sophisticated wind speed estimator.

5.4 Insights into PORFC

In order to gain deeper insight into the behavior of PORFC, short time periods within a turbulent simulation in the "Default lidar" scenario are analyzed.

5.4.1 Evolution of residue

Figure 16 shows a situation where the stress prediction at the initial value $\sigma(t_0)$ turns from a mildly rising slope (MPC step 1) to a mildly falling slope (MPC step 2). Consequently, a new stress maximum is formed and added to the "right-hand-side" of the residue set at MPC step 2, as shown in Fig. 16a. In the following steps 3 to 5, the size of the residue set remains constant; only the right-hand-side value is updated by the current initial stress value.

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

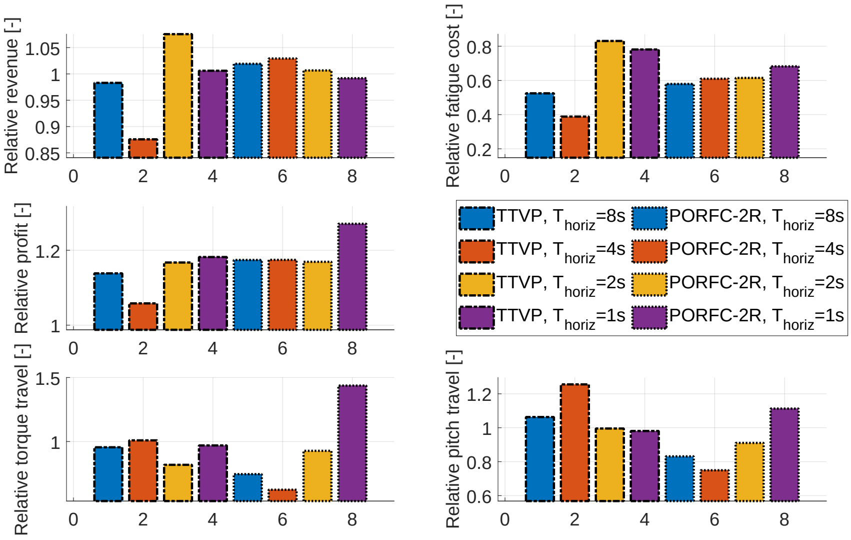


Figure 15. Weibull-weighted KPIs of the "Perfect prediction" scenario normalized with respect to the individually corresponding KPIs of the "Perfect persistence" scenario with same MPC formulation and horizon length.

5.4.2 Evolution of PORFC parameters

As shown in Fig. 16b, the change of extrema in the stress prediction over the course of MPC steps leads to frequent changes in the PORFC stress references (see Sect. 3.4.2) or, more in general, in the PORFC parameter structure (see bars in Fig. 16b). However, since many of the emerging or vanishing stress cycles are small in amplitude, also their corresponding stress reference values are close to the stress prediction trajectory (compare bars to the solid blue line), and thus have low impact on the overall optimization problem.

Since by nature of MPC the stress trajectory is shifted to the left-hand-side with each simulation step, also the PORFC parameter samples are shifted. This becomes even more clear in Fig. 17, where stress reference 1 is plotted over the prediction horizon and over MPC steps (simulation time). Here, over the course of MPC steps, the stress reference pattern is evolving smoothly towards the beginning of the prediction horizon. While some references emerge within the prediction horizon, many references originate at the end and do not vanish before reaching the beginning of the prediction horizon.

5.5 Results of deterministic gust simulations

According to the current standards (IEC, 2005), a central qualification criterion for controllers is their reaction to deterministic gusts. Previous literature already has shown that deterministic gusts are an easy but unrealistic task for predictive controllers like MPCs (Schlipf and Raach, 2016), resulting in too optimistic conclusions regarding extreme load reduction. Besides, extreme

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

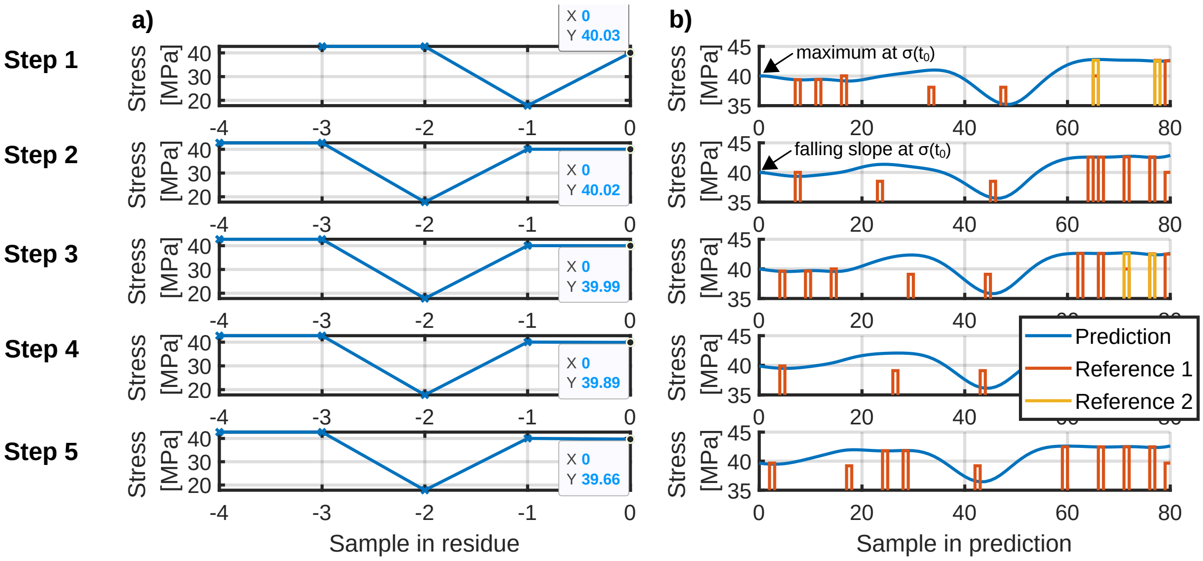


Figure 16. Stress trajectories for PORFC-2R in turbulent wind 10s after the start of the simulation. Top-down: Five consecutive MPC steps. Residue set of variable size, where the values of the last stress samples are labeled (a). Stress prediction of the MPC, and stress references as part of the PORFC parameter set (b).

loads are not even always design-driving for some wind turbine components (Canet et al., 2020). Nonetheless, the study of gust scenarios sheds additional light on the controller dynamic behavior.

Thus, in the following, the conventional controller is compared to the MPC formulations of TTVP and PORFC-2R in an "Extreme operating gust" scenario (IEC, 2005), with a duration of 10.5s and an initial wind speed that is 2ms^{-1} below rated wind speed ($V_{\text{rated}} = 11.4\text{ms}^{-1}$). In order to test the MPCs with partial knowledge of the gust, a prediction horizon of 4s is chosen. Besides this limited horizon, a perfect wind prediction without lidar errors is assumed.

As shown in Fig. 18, even during the gust, for all controllers the rotor speed remains below the rated speed of 1.267 rad s^{-1}. As a result, the conventional controller remains at the minimum pitch angle, and the tower deflection freely follows the gust wind speed, which leads to a high positive excursion. After the gust, the tower oscillation quickly vanishes due to aerodynamic damping.

In contrast, the MPCs anticipate the incoming gust, and react to a significant extent by pitching the blades. Interestingly, despite their different fatigue cost formulations, the MPCs exhibit very similar pitching behavior. As expected, the TTVP and the PORFC-2R MPC attenuate very effectively the tower excursion, and dampen the oscillation immediately. However, PORFC-2R puts less priority on the attenuation of the tower excursion. Since the tower deflection has been flat prior to the gust, the stress residue of PORFC-2R contains only stress values around the steady state. Consequently, PORFC-2R assumes only a small stress cycle with low damage potential during the gust. This behavior is changed, if the stress residue is initialized

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

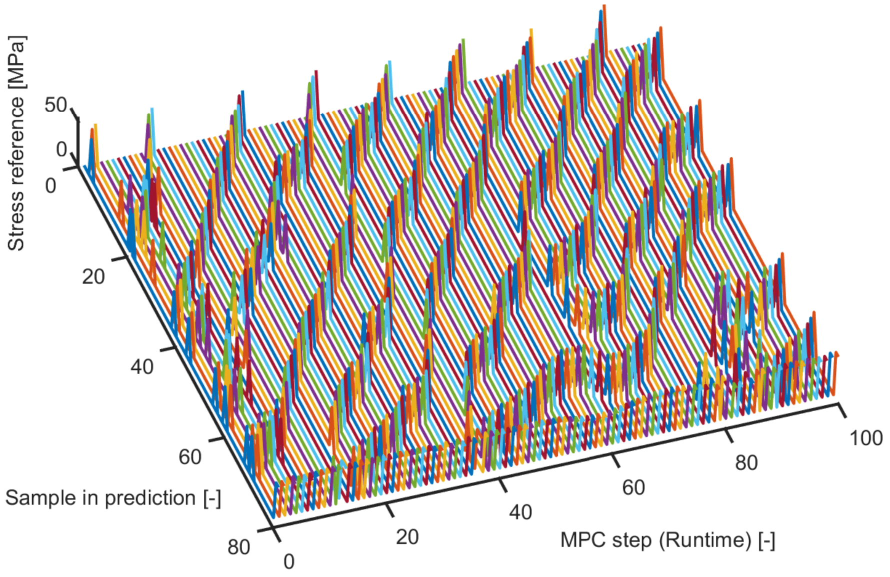


Figure 17. PORFC stress reference over the prediction horizon of 80 samples for 100 consecutive MPC steps.

with 0 MPa, which corresponds to an undeflected tower prior to operation. Due to this stress memory, PORFC-2R identifies a large half cycle, and consequently tries to further limit the maximum tower excursion by peak-shaving, as shown by the purple trajectory in Fig. 18.

Clear differences of PORFC-2R with respect to TTVP can be seen in the rotor speed and generator power dynamics. At the beginning of the gust, the generator power is reduced in order to achieve a high rotor speed during the gust. This behavior can be attributed to an attempt at harvesting the energy of the gust, and also has been observed for an MPC where 5 QPs (instead of 1 QP) have been solved per MPC step for better convergence. For the PORFC-2R MPC with the 0 MPa residue, the rotor speed remains at a high level for a longer time frame, which is an unusual behavior and requires more investigation. Finally, it can be noted that the steady state rotor speed is slightly higher for the MPCs than for the conventional controller, as seen before the gust. Assuming perfect tracking of optimal rotor speed by the conventional controller, this difference can be attributed to the MPC plant-model mismatch. However, as shown in Fig. 18, the rotor speed difference does not result in significant suboptimality of the steady state power capture.

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

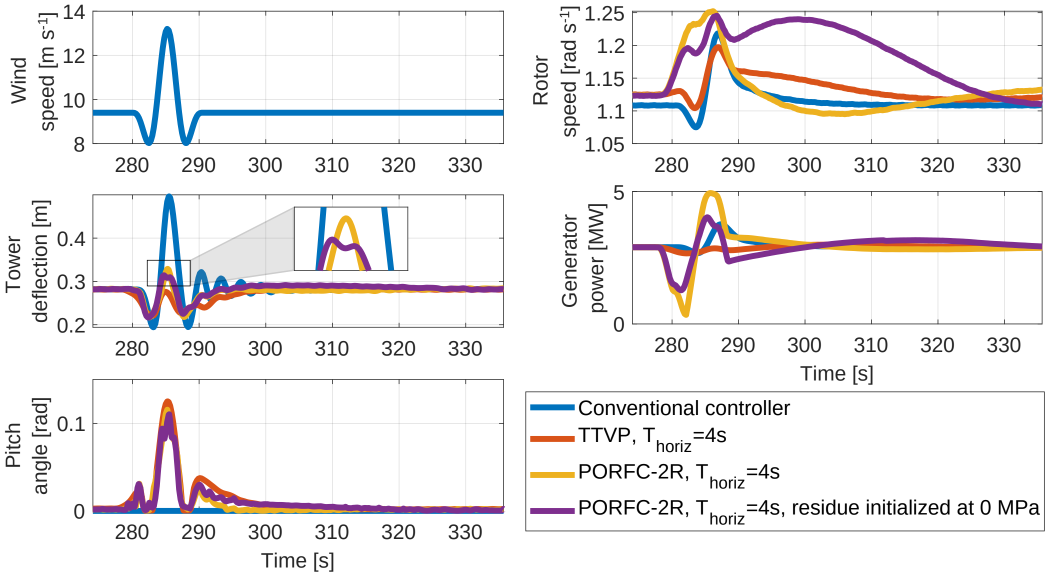


Figure 18. Extreme operating gust at $V = V_{\mathrm{rated}} - 2\mathrm{ms}^{-1}$.

6 Conclusions and outlook

6.1 Conclusions

The present work represents a significant step in assessing the benefits of the MPC formulation of PORFC. For this purpose, the simulation setup of Loew et al. (2020b) has been extended by a realistic lidar simulator, lidar processing, and a Moving Horizon Estimator (MHE).

First, the PORFC formulation has been presented in an application-focused way. It has been highlighted how PORFC directly incorporates mechanical fatigue in predictive wind turbine control. Since fatigue requires long observation windows, stress history has been considered in a consistent manner by carrying along a *residue* (PORFC-2R).

Second, the formulation of the MHE has been explained, where the lidar wind estimate has been used to generate an initialization for the unmeasured tower states.

Third, a sequential tuning approach has been employed for the lidar simulator, lidar processing, MHE, and MPC:

- For the lidar simulator, parameters from the literature have been utilized, which maximize the measurement coherence bandwidth.
- For the lidar buffering and filtering, simple adaptive tuning laws have been employed. Simulations have revealed that they result already in good performance, and that no further tuning seems to be required.

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

Open Access WIND ENERGY SCIENCE DISCUSSIONS

- For the MHE, instead of accurately reconstructing the plant states, the cost weights have been tuned to estimate only the low-frequency state information that can be handled by the MPC-internal model.
- For the MPC, four different prediction horizon lengths have been employed throughout the study, since no single horizon length has led to the best performance in all scenarios. In the MPC cost function, the fatigue weight has been tuned systematically for each controller formulation and horizon length.

Finally, extensive economic and dynamic simulation results have been presented for turbulent and gust wind settings:

- In the "Default lidar" scenario, PORFC-2R has obtained a 2.5% higher profit than a conventional MPC, and a 30% higher profit than a conventional PID controller. For shorter horizons, especially the PORFC formulation with residue has shown a more robust profit than the conventional MPC.
- In the "High decay lidar" scenario with a lower lidar prediction quality, PORFC-2R had an even higher profit benefit over the conventional MPC equal to 5.1%, while surpassing the conventional PID controller by 26%. This suggests that PORFC-2R is a recommended solution especially for lower lidar prediction quality.
- In the "Perfect prediction" scenario, both MPCs have exhibited similar results. A comparison to a wind persistence setting has shown that PORFC-2R benefits more from the availability of this high quality prediction than the conventional MPC.
- In all considered turbulent scenarios, MPCs with a very short prediction horizon of 1s have obtained only modest results.
- An excerpt of a turbulent simulation with PORFC-2R has demonstrated how the residue is updated, and that the parametric stress references evolve smoothly following an expected pattern.
- In an Extreme Operating Gust setting, both MPCs have shown a similar pitching behavior and the effective attenuation of tower excursion. During the gust, PORFC-2R has shown a higher variability in the rotor speed.

6.2 Outlook

The MPC formulation of PORFC still has several aspects worth investigating:

- For the MHE tuning, an automated but still computationally tractable approach should be developed.
- In the MPC cost function, economic terms for the actuator, blade and drive-train damage should be included.
- The MHE- and MPC-internal system model has a significant error with respect to the plant system. Thus, online model adaptation promises further benefits.
- In certain business cases, the goal may not be to minimize fatigue but simply to keep the fatigue rate on average below certain thresholds, or to keep the cumulative damage below a threshold by the end of service. To assist these goals, the PORFC cost function could be modified and fatigue could be added as a parametric constraint in the MPC. Alternatively, an outer control loop based on structural health monitoring could be added (Do and Söffker, 2020), which adapts the MPC cost function weights.
- The novel PORFC MPC has been extensively simulated and is ready for application on real systems. Consequently, just like for conventional MPCs in Sinner et al. (2021) and Dickler et al. (2021), the novel PORFC MPC should be assessed on scaled and full-scale wind turbines.

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

eawe european academy of wind energy
Open Access WIND ENERGY SCIENCE DISCUSSIONS

Nomenclature

Quantity	Unit	Explanation
$\bar{a}$	-	Quantity sampled on the control intervals of the prediction horizon
$\hat{a}$	-	Quantity fixed for one MPC step
$\tilde{a}$	-	Quantity estimated from measurements

Abbreviation		Explanation
CC	-	Conventional PID controller
lidar	-	Light Detection And Ranging
MHE	-	Moving Horizon Estimator
MPC	-	Model Predictive Controller
PORFC	-	Parametric Online Rainflow Counting
QP	-	Quadratic Programming
RFC	-	Rainflow counting algorithm
TTVP	-	Tower Tip Velocity Penalization

Author contributions. SL developed the controller formulation, implemented all elements of the toolchain, and performed the simulation studies. CLB supervised the work. Both authors provided important input to this research work through discussions, feedback and by writing the paper.

Competing interests. The authors declare that they have no conflicts of interest.

Acknowledgements. The authors would like to thank the company `sowento GmbH` for kindly providing a free license of their lidar simulator. Furthermore, the close and active support of Martin Koch, David Schlipf and Steffen Raach from `sowento GmbH` is gratefully acknowledged.

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

References

Anand, A.: Optimal Control of Battery Energy Storage System for Grid Integration of Wind Turbines, Master thesis, TU Munich, Munich, 2020.

ASTM: Standard practices for cycle counting in fatigue analysis, 1985.

Barradas-Berglind, J. d. J., Wisniewski, R., and Soltani, M.: Fatigue damage estimation and data-based control for wind turbines, IET Control Theory & Applications, 9, 1042–1050, https://doi.org/10.1049/iet-cta.2014.0730, 2015.

Barradas-Berglind, J. J. and Wisniewski, R.: Representation of fatigue for wind turbine control, Wind Energy, 19, 2189–2203, https://doi.org/10.1002/we.1975, 2016.

Bottasso, C. L., Pizzinelli, P., Riboldi, C., and Tasca, L.: LiDAR-enabled model predictive control of wind turbines with real-time capabilities, Renewable Energy, 71, 442–452, https://doi.org/10.1016/j.renene.2014.05.041, 2014.

Canet, H., Löw, S., and Bottasso, C. L.: Lidar-assisted control in wind turbine design: Where are the potential benefits?, Journal of Physics: Conference Series, 1618, 042 020, https://doi.org/10.1088/1742-6596/1618/4/042020, 2020.

Dickler, S., Wintermeyer-Kallen, T., Zierath, J., Bockhahn, R., Machost, D., Konrad, T., and Abel, D.: Full-scale field test of a model predictive control system for a 3 MW wind turbine, Forschung im Ingenieurwesen, 85, 313–323, https://doi.org/10.1007/s10010-021-00467-w, 2021.

Diehl, M. and Gros, S.: Numerical Optimal Control: Lecture manuscript, University of Freiburg, Freiburg and Gothenburg, https://www.syscop.de/files/2020ss/NOC/book-NOCSE.pdf, 2020.

Do, H. and Söffker, D.: Wind Turbine Lifetime Control Using Structural Health Monitoring and Prognosis, IFAC-PapersOnLine, 53, 12 669–12 674, https://doi.org/10.1016/j.ifacol.2020.12.1847, 2020.

Evans, M. A., Cannon, M., and Kouvaritakis, B.: Robust MPC Tower Damping for Variable Speed Wind Turbines, IEEE Transactions on Control Systems Technology, 23, 290–296, https://doi.org/10.1109/TCST.2014.2310513, 2015.

Gros, S.: An Economic NMPC Formulation for Wind Turbine Control, in: 52nd IEEE Conference on Decision and Control, pp. 1001–1006, IEEE, 2013.

Gros, S. and Schild, A.: Real-time economic nonlinear model predictive control for wind turbine control: International Journal of Control, International Journal of Control, pp. 1–14, https://doi.org/10.1080/00207179.2016.1266514, 2017.

Gros, S., Vukov, M., and Diehl, M.: A real-time MHE and NMPC scheme for wind turbine control, in: 2013 IEEE 52nd Annual Conference on Decision and Control (CDC), pp. 1007–1012, IEEE, Piscataway, NJ, https://doi.org/10.1109/CDC.2013.6760014, 2013.

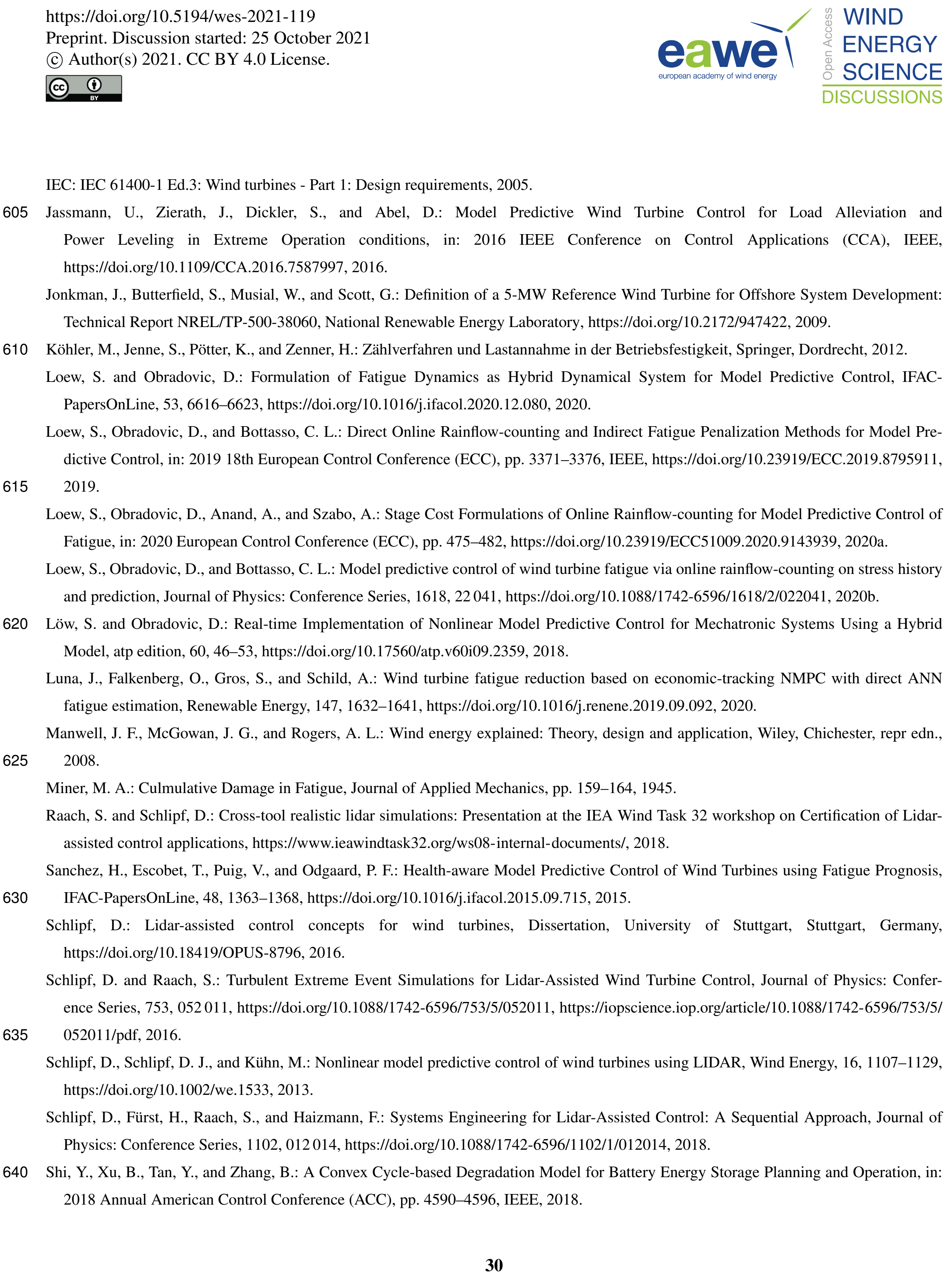
Grüne, L. and Pannek, J.: Nonlinear Model Predictive Control: Theory and Algorithms, Communications and Control Engineering, Springer International Publishing, 2nd ed. 2017 edn., 2017.

Haibach, E.: Betriebsfestigkeit: Verfahren und Daten zur Bauteilberechnung, VDI-Buch, Springer, Berlin, 3., korrigierte und erg. aufl. edn., 2006.

Hau, E.: Windkraftanlagen: Grundlagen. Technik. Einsatz. Wirtschaftlichkeit, Springer Berlin Heidelberg, Berlin, Heidelberg, 6th ed. edn., http://ebookcentral.proquest.com/lib/gbv/detail.action?docID=4818780, 2017.

Heinrich, C., Khalil, M., Martynov, K., and WEVER, U.: Online remaining lifetime estimation for structures, Mechanical Systems and Signal Processing, 119, 312–327, https://doi.org/10.1016/j.ymssp.2018.09.028, 2019.

Huang, R., Biegler, L. T., and Patwardhan, S. C.: Fast Offset-Free Nonlinear Model Predictive Control Based on Moving Horizon Estimation, Industrial & Engineering Chemistry Research, 49, 7882–7890, 2010.



https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

IEC: IEC 61400-1 Ed.3: Wind turbines - Part 1: Design requirements, 2005.

Jassmann, U., Zierath, J., Dickler, S., and Abel, D.: Model Predictive Wind Turbine Control for Load Alleviation and Power Leveling in Extreme Operation conditions, in: 2016 IEEE Conference on Control Applications (CCA), IEEE, https://doi.org/10.1109/CCA.2016.7587997, 2016.

Jonkman, J., Butterfield, S., Musial, W., and Scott, G.: Definition of a 5-MW Reference Wind Turbine for Offshore System Development: Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, https://doi.org/10.2172/947422, 2009.

Köhler, M., Jenne, S., Pötter, K., and Zenner, H.: Zählverfahren und Lastannahme in der Betriebsfestigkeit, Springer, Dordrecht, 2012.

Loew, S. and Obradovic, D.: Formulation of Fatigue Dynamics as Hybrid Dynamical System for Model Predictive Control, IFAC-PapersOnLine, 53, 6616–6623, https://doi.org/10.1016/j.ifacol.2020.12.080, 2020.

Loew, S., Obradovic, D., and Bottasso, C. L.: Direct Online Rainflow-counting and Indirect Fatigue Penalization Methods for Model Predictive Control, in: 2019 18th European Control Conference (ECC), pp. 3371–3376, IEEE, https://doi.org/10.23919/ECC.2019.8795911, 2019.

Loew, S., Obradovic, D., Anand, A., and Szabo, A.: Stage Cost Formulations of Online Rainflow-counting for Model Predictive Control of Fatigue, in: 2020 European Control Conference (ECC), pp. 475–482, https://doi.org/10.23919/ECC51009.2020.9143939, 2020a.

Loew, S., Obradovic, D., and Bottasso, C. L.: Model predictive control of wind turbine fatigue via online rainflow-counting on stress history and prediction, Journal of Physics: Conference Series, 1618, 22 041, https://doi.org/10.1088/1742-6596/1618/2/022041, 2020b.

Löw, S. and Obradovic, D.: Real-time Implementation of Nonlinear Model Predictive Control for Mechatronic Systems Using a Hybrid Model, atp edition, 60, 46–53, https://doi.org/10.17560/atp.v60i09.2359, 2018.

Luna, J., Falkenberg, O., Gros, S., and Schild, A.: Wind turbine fatigue reduction based on economic-tracking NMPC with direct ANN fatigue estimation, Renewable Energy, 147, 1632–1641, https://doi.org/10.1016/j.renene.2019.09.092, 2020.

Manwell, J. F., McGowan, J. G., and Rogers, A. L.: Wind energy explained: Theory, design and application, Wiley, Chichester, repr edn., 2008.

Miner, M. A.: Culmulative Damage in Fatigue, Journal of Applied Mechanics, pp. 159–164, 1945.

Raach, S. and Schlipf, D.: Cross-tool realistic lidar simulations: Presentation at the IEA Wind Task 32 workshop on Certification of Lidar-assisted control applications, https://www.ieawindtask32.org/ws08-internal-documents/, 2018.

Sanchez, H., Escobet, T., Puig, V., and Odgaard, P. F.: Health-aware Model Predictive Control of Wind Turbines using Fatigue Prognosis, IFAC-PapersOnLine, 48, 1363–1368, https://doi.org/10.1016/j.ifacol.2015.09.715, 2015.

Schlipf, D.: Lidar-assisted control concepts for wind turbines, Dissertation, University of Stuttgart, Stuttgart, Germany, https://doi.org/10.18419/OPUS-8796, 2016.

Schlipf, D. and Raach, S.: Turbulent Extreme Event Simulations for Lidar-Assisted Wind Turbine Control, Journal of Physics: Conference Series, 753, 052 011, https://doi.org/10.1088/1742-6596/753/5/052011, https://iopscience.iop.org/article/10.1088/1742-6596/753/5/052011/pdf, 2016.

Schlipf, D., Schlipf, D. J., and Kühn, M.: Nonlinear model predictive control of wind turbines using LIDAR, Wind Energy, 16, 1107–1129, https://doi.org/10.1002/we.1533, 2013.

Schlipf, D., Fürst, H., Raach, S., and Haizmann, F.: Systems Engineering for Lidar-Assisted Control: A Sequential Approach, Journal of Physics: Conference Series, 1102, 012 014, https://doi.org/10.1088/1742-6596/1102/1/012014, 2018.

Shi, Y., Xu, B., Tan, Y., and Zhang, B.: A Convex Cycle-based Degradation Model for Battery Energy Storage Planning and Operation, in: 2018 Annual American Control Conference (ACC), pp. 4590–4596, IEEE, 2018.

https://doi.org/10.5194/wes-2021-119
Preprint. Discussion started: 25 October 2021

Sinner, M., Petrovic, V., Langidis, A., Neuhaus, L., Holling, M., Kuhn, M., and Pao, L. Y.: Experimental Testing of a Preview-Enabled Model Predictive Controller for Blade Pitch Control of Wind Turbines, IEEE Transactions on Control Systems Technology, pp. 1–15, https://doi.org/10.1109/TCST.2021.3070342, 2021.

The MathWorks Inc.: Rainflow counts for fatigue analysis, 2018.

Verschueren, R., Frison, G., Kouzoupis, D., Frey, J., van Duijkeren, N., Zanelli, A., Novoselnik, B., Albin, T., Quirynen, R., and Diehl, M.: acados—a modular open-source framework for fast embedded optimal control, Mathematical Programming Computation, https://doi.org/10.1007/s12532-021-00208-8, 2021.

4.3 Paper 8: Battery energy storage system in frequency regulation market

4.3.1 Summary

Stationary and transient operation of LiIon battery cells cause calendric and cyclic aging, respectively. Cyclic aging is found to be influenced significantly by the Depth of Discharge of individual State of Charge (SOC) cycles, which is very similar to mechanical fatigue. In this sense, SOC in the battery domain acts like stress in the mechanical domain.

This similarity is exploited to improve existing simulation models and MPCs for battery systems. For cyclic aging simulation, the one-step value-continuous damage estimation from **Paper 5** is adapted. For calendric aging simulation, a new time-variant term is proposed which improves consistency.

For the MPC, the wind turbine formulation of PORFC from **Paper 6** is adapted to batteries, and applied to the PJM frequency regulation market. In this market, the battery can provide less than the demanded reference power, e.g. in order to limit its own aging. However, each mismatch results in monetary penalization. Even though the underlying formulation of PORFC with residue is similar to **Paper 6**, several modifications of the optimization problem are required:

- Additional time-variant cost function for calendric aging;
- Power tracking cost instead of the economic revenue cost function;
- Highly dynamic power reference and mismatch penalty instead of constant weights;
- Additional time-variant power tracking constraint.

The PORFC MPC is compared to a simple rule-based controller and to a state-of-the-art MPC from the literature with simplified SOC cycle identification by a switched linear system. An open loop test demonstrates that this simplified approach leads to an underestimation of cyclic aging. In contrary, a validation of the PORFC cyclic aging cost function shows that correct aging estimation is maintained despite the distribution of aging over time and the moving-horizon mode.

In a simplified market setting, the PORFC MPC incurs 13% less total penalty than the state-of-the-art MPC and 23% less total penalty than a conventional rule-based controller. Total penalty here is the sum of power mismatch penalty and aging cost. In a full market setting with additional constraints, the PORFC MPC exhibits its ability to adapt even to unusual economic situations, and to outperform manually chosen control policies by up to 10% in terms of total penalty. In an artificial scenario with a ramp power reference, the behaviors of the closed-loop PORFC MPC and of an open-loop OCP match well. Low Karush-Kuhn-Tucker (KKT) values further indicate sufficient optimality.

4.3.2 Contribution

Abhinav Anand (AA) and the author of this dissertation contributed equally to this publication. The author of this dissertation developed the initial model architecture and controller formulation. AA and the author of this dissertation jointly developed the model and the controllers, analyzed the simulations, and wrote the paper. AA improved the formulations, performed the literature survey, implemented the controllers, and performed the simulations. AS initiated and closely supervised the work. All authors provided important input to this research work through discussions, feedback and by improving the manuscript.

4.3.3 Reference

S. Loew, A. Anand, and A. Szabo, “Economic model predictive control of li–ion battery cyclic aging via online rainflow–analysis,” *Energy Storage*, 2021. doi:10.1002/est2.228

Received: 2 November 2020 | Revised: 8 January 2021 | Accepted: 9 January 2021

DOI: 10.1002/est2.228

RESEARCH ARTICLE

ENERGY STORAGE WILEY

Economic model predictive control of Li-ion battery cyclic aging via online rainflow-analysis

Stefan Loew[1,2] | Abhinav Anand[1,2] | Andrei Szabo[1]

[1]Corporate Technology, Siemens AG, Munich, Germany

[2]Wind Energy Institute, Technical University of Munich, Garching bei Muenchen, Germany

Correspondence
Stefan Loew, Corporate Technology, Siemens AG, Otto-Hahn-Ring 6, Munich, Germany.
Email: loew.stefan@siemens.com

Abstract

Cycle identification via the rainflow-algorithm is implemented online in a model predictive controller (MPC) for Li-ion batteries. This is achieved by externalization of the cycle identification from the optimization problem. The limitation for cyclic aging estimation due to short prediction horizons is overcome by updating and utilizing a State of Charge memory. Furthermore, a comprehensive plant model for Li-ion batteries is presented with novel submodels for calendric and cyclic aging. The novel MPC is implemented in the ACADO Toolkit and tested with the aforementioned plant model. Simulation results indicate that—even without tuning—the novel MPC clearly outperforms a rule-based controller and an extensively tuned MPC from literature.

KEYWORDS

cyclic aging, Li-ion battery, model predictive control, rainflow-analysis

1 | INTRODUCTION

1.1 | Motivation and previous work

Battery energy storage systems are very well suited to absorb and release electrical energy with intermediate storage periods which can last over different time scales. This stationary and transient operation causes calendric and cyclic aging, respectively. Aging manifests in the decrease of charge capacity and the increase of internal resistance.[1] When a defined aging level is reached, the battery reaches its end-of-life and has to be replaced. Consequently, an important task of modern battery operation strategies is the economic balancing of the revenue from energy storage and the cost of aging. This is a classical optimal control problem (OCP). Model predictive control (MPC) enables the solution of this OCP on a moving horizon while respecting constraints on system states and inputs.[2]

A crucial and difficult task in the setup of the MPC is the correct modeling of abovementioned aging mechanisms. Specifically, cyclic aging estimation requires cycle identification from the state of charge (SOC) trajectory. The absolute range of SOC excursion of a cycle is called Depth of Discharge (DOD), and has a strong influence on the amount of aging.[3] The most accepted method for cycle identification for aging estimation is the rainflow-counting (RFC) algorithm.[4] However, RFC is a discontinuous algorithm and does not have an analytical mathematical form.[5,6] Thus, until recently, RFC could not be implemented in online controllers of batteries, and could only be used for post-processing of measured and simulated data. Instead, typically only surrogates for the cycling have been utilized in the setup of controllers, as described in the following.

A fairly simple surrogate is the penalization of charge throughput which is implemented as a time integral of discharge current in Wang et al.[7] This method completely lacks the consideration of cycles. Thus, the disproportionate aging of big cycles is missed.

Some touch of cyclic behavior is captured by the simplistic definition of DOD as the difference of SOC between the beginning and the end of a control step in Yallamilli et al.[8] However, this method still lacks the identification of extrema and actual cycles. Thus, on the one hand, cycles which may be located within the considered control step are missed entirely. On the other hand, the aging effect of cycles which span over multiple control steps is highly underestimated.

This limitation to fixed evaluation samples is not present in He et al,[9] where the location of extrema is identified on a historic SOC trajectory and used in the control problem. However, here DOD is simply defined as the SOC difference between two adjacent extrema. This definition corresponds to *Simple Range Counting*, which is standardized in ASTM.[4] As a consequence of this definition, no nesting of small cycles within big cycles can be considered. Instead, a big cycle is split into two cycles by a nested small cycle, and thus will be underestimated again in terms of aging.

The same simplistic definition of DOD is used in Koller et al.[10] Unlike the preceding formulation, here identification of extrema is based on the predicted SOC trajectory, and is implemented by a switched linear dynamic subsystem within the MPC. Even though the above drawback of aging underestimation persists, this formulation can be classified as sufficiently advanced, and thus is utilized as comparison object in the present work.

Proper cycle identification by the rainflow algorithm in offline optimal control of a battery is presented in Shi et al[11]. Further, this work provides an excellent proof of convexity of the rainflow algorithm which guarantees reliable convergence of the considered OCP. However, computational times will be high due to the slowly-converging gradient-descent method.

In order to enable online implementation, a simple control policy with upper and lower SOC thresholds is derived from above offline method in Shi et al [5] and Xu et al.[12] The distance of these SOC thresholds represents a DOD which is cost-optimal between mismatch penalty and aging cost. While this cost-optimal DOD can be calculated explicitly a priori, the actual threshold levels arise online. Benefits of this policy are its simplicity and a quantified optimality gap to the offline method. Furthermore, these thresholds enable carrying along reduced information about SOC history, whose benefit will be explained later. As drawbacks, the policy does not seem to support more realistic battery dynamics where SOC is not a simple integral of power, or other cost terms than mismatch penalty and cyclic aging. Thus, this policy is not suitable if also calendric aging should be considered, or if a hybrid energy system controller is pursued.

In Loew et al,[13] an MPC formulation has been presented which allows for RFC evaluation externally from the MPC algorithm, and the inclusion of its results into the MPC via time-varying parameters. Therefore, this formulation is referred to as *Parametric Online Rainflow-counting (PORFC)*. In PORFC, cyclic aging is calculated based on SOC information from the prediction horizon of the MPC. However, cyclic aging is a long-term effect where SOC cycles are usually defined on much longer time spans. Therefore, in Loew et al,[14] PORFC has been combined with the concept of residue to systematically incorporate SOC history. In the same work, this concept has been applied successfully for MPC of a fatigue-affected wind turbine.

1.2 | Contribution

In the present work, these controller formulations for wind turbines[14] are extended, and adapted to the requirements and use cases of battery energy storage systems. To the best of the authors' knowledge, this introduces the first model predictive controller (MPC) that considers correctly and without approximation the cyclic aging of batteries.

As a by-product for battery modeling, novel submodels for calendric and cyclic aging are presented which improve consistency and accuracy.

The novel MPC formulation is compared to a conventional rule-based controller and the above mentioned state-of-the-art MPC from Koller et al[10] in a realistic market scenario, showing significantly better economic performance. Furthermore, the accuracy of the cyclic aging cost function, the degree of optimality, and the computational performance of the MPC are assessed in detail.

1.3 | Outline

In section 2, a battery plant model with specific focus on aging is presented. This model is the basis for the MPC formulation in section 3 whose key novelty is the direct implementation of cycle identification. In section 4, this MPC formulation is compared with state of the art controllers in a realistic market setting. Furthermore, key features of the MPC are analyzed and validated. In section 5, the work is concluded and an outlook is provided.

2 | BATTERY PLANT MODEL

The goal of this section is to present the approach used to model Lithium-ion batteries with particular focus on degradation. A grey-box modeling approach is used which involves first-principle models, data-fitted static maps and algorithmic parts. A comprehensive battery storage

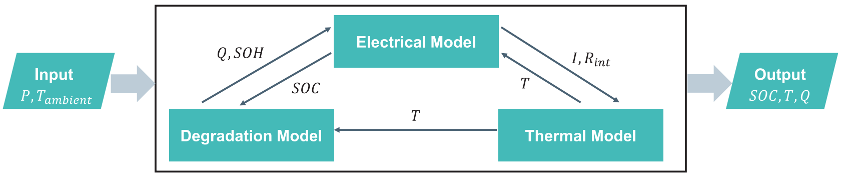


FIGURE 1 Process flow of the battery model

system model includes several individual sub-models, capturing the electrical, thermal, and aging behavior as shown in Figure 1. These sub-models will be presented in the following.

2.1 | Electrical model

A simple electrical equivalent circuit as shown in Figure 2 is chosen.[15] Important variables are charge/discharge power P, open circuit voltage U_{ocv}, internal resistance R_{int}, current I and terminal voltage U_{terminal}. The states considered are SOC, SOH, and cell temperature T. State of Charge $SOC = \frac{Q_{\text{cont}}}{Q}\ [-]$ is defined as the ratio of instantaneous charge *content* Q_{cont} with respect to the instantaneous charge *capacity* Q of the battery. Thus, the rate of SOC is directly proportional to the current, and can be expressed as

$$\frac{dSOC}{dt} = -\frac{I}{Q}\ \left[\frac{1}{s}\right]. \tag{1}$$

By convention, current is considered positive for the discharging case and negative for the charging case. The calculation of open circuit voltage, internal resistance, battery current and terminal voltage is detailed in section A.1.

State of Health $SOH = \frac{Q}{Q_{nominal}}\ [-]$ is defined as the ratio of instantaneous charge capacity Q with respect to the nominal charge capacity of the battery $Q_{nominal}$ at the time of production. Charge capacity and thereby State of Health are decreased by aging which will be detailed in section 2.3.

Note: Within the present work, the term "capacity" refers to the charge capacity of the battery (in [As]) whereas "energy capacity" refers to the energy capacity of the battery (in [Ws]).

2.2 | Thermal model

The thermal model captures the dynamic behavior of cell temperature T (in [K]) due to internal heating and external cooling. The most commonly used thermal model

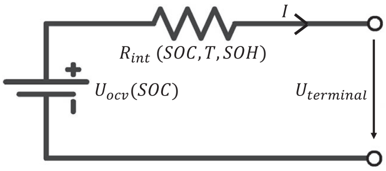


FIGURE 2 Battery electrical equivalent circuit model

$$\frac{dT}{dt} = \frac{1}{C_{\text{H}}}\left(I^2 R_{\text{int}} - C_{\text{R}}(T - T_{\text{ambient}})\right)\ \left[\frac{K}{s}\right] \tag{2}$$

is based on a lumped heat capacitance with the parameters heat capacity C_{H} (in [J/K]) and cooling rate C_{R} (in [J/(Ks)]). Ambient temperature T_{ambient} is an external input.[16]

2.3 | Degradation model

The degradation model captures the loss in charge capacity and the increase of internal resistance over time and usage. Typically, only the degradation of capacity is modeled, and used to express the degradation of resistance via (A5).

2.3.1 | Charge capacity

Typically, the charge capacity decreases over the lifetime of the battery. By convention, the amount of loss of charge capacity is assigned to two quantities: calendric capacity loss Q_{cal} which occurs intrinsically at any time, and cyclic capacity loss Q_{cyc} which occurs during usage of the battery. Consequently, the instantaneous charge capacity

$$Q = Q_{nominal} - \left(Q_{cal} + Q_{cyc}\right)$$

can be expressed by aforementioned quantities. The battery is considered to be at the end of its service life when its capacity Q has decreased to a predefined end-of-life capacity Q_{eol}.

2.3.2 | Calendric aging

Calendric aging refers to the permanent loss in battery capacity, independently of usage. The corresponding rate of calendric capacity loss

$$\frac{\mathrm{d}Q_{\mathrm{cal}}}{\mathrm{d}t}(t,SOC,T)=A_{\mathrm{cal}}f_{Q\mathrm{cal}}(t)f_{Q\mathrm{cal}}(SOC)f_{Q\mathrm{cal}}(T)\quad\left[\frac{As}{s}\right]\tag{3}$$

depends on time since production t, and on the additional stress factors of SOC and cell temperature T.

Experiments have shown that calendric capacity loss exhibits a square-root behavior $Q_{\mathrm{cal}}\sim\sqrt{t}$ over time for constant SOC and temperature.[1] In the literature, this square-root dynamical behavior often is modeled in a way which does not suit continuous simulation.[1,17] Therefore, in the present work, the time-derivative of the square-root

$$f_{Q\mathrm{cal}}(t)=\frac{1}{2\sqrt{t}}\approx\frac{1}{2\sqrt{t+t_{\mathrm{shift}}}}\quad\left[\frac{1}{\sqrt{s}}\right]\tag{4}$$

is proposed for seamless integration into the dynamic model. This term corresponds to a time-variance of the dynamical system. The shift time t_{shift} is added to avoid numerical errors at $t=0$. Its value can be very small. Alternatively, the shift time can also be used to account for the previous operation time duration of an already aged battery. The constant A_{cal} $\left[\frac{As}{\sqrt{s}}\right]$ is derived from experimental data. Its unit has no physical meaning and simply results from the chosen mathematical model.

The temperature effect typically is represented by the Arrhenius law

$$f_{Q\mathrm{cal}}(T)=A_{\mathrm{arrhenius}}\exp\left(\frac{-E_{\mathrm{activation}}}{R_{\mathrm{gas}}T}\right)\quad[-]\tag{5}$$

where $A_{\mathrm{arrhenius}}$ is a model parameter obtained by experiments, $E_{\mathrm{activation}}$ is the activation energy of the cell, and R_{gas} is the Universal gas constant. The SOC effect

$$f_{Q\mathrm{cal}}(SOC)=K_1+K_2SOC\quad[-]\tag{6}$$

is represented by a linear fit of experimental data via the parameters K_1 and K_2.

2.3.3 | Cyclic aging

Concept

Cyclic aging refers to the loss of battery capacity during dynamic operation by repeated charging and discharging.

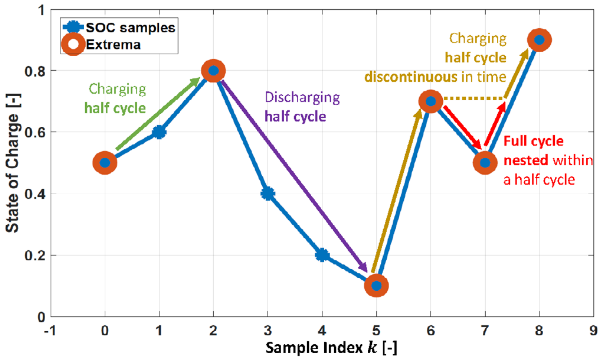


FIGURE 3 Exemplary result of a rainflow analysis. Sampled SOC trajectory (blue), extrema (red circles). There are three half cycles (green, purple, and yellow) and one nested full cycle (red)

The major stress factor for cyclic aging is the absolute range of magnitude of the individual SOC cycles, also referenced as "depth of discharge" *DOD*.[3,11] Other than that, current, temperature, and terminal voltage also affect cyclic damage, but are neglected in the present work.[1,18] The most accepted algorithm for cycle identification from SOC trajectories is the well-known *Rainflow* algorithm.[4] Identified cycles can be "full" or "half cycles," as visualized in Figure 3. A full cycle refers to a sequence of battery charge and discharge with identical initial and final SOC values. A half cycle refers to only one charging or discharging sequence. Furthermore, cycles can be nested in other cycles. The output of the rainflow algorithm is summarized in Table 1.

Damage and capacity loss

Lifetime of the battery is usually specified as the number of full cycles of a given cycle depth which the battery can provide before reaching end-of-life capacity Q_{eol}. Non-dimensional damage of an individual cycle of a specific depth is obtained by the reciprocal of the number full of cycles. For an increased cycle depth, the damage of the battery typically increases over-proportionately.[3] Based on experimental data at discrete DODs, this dependency is represented by a piece-wise linear interpolation

$$D_{\mathrm{cyc,c}}=\phi\left(DOD_c^{\mathrm{RFC}},w_c^{\mathrm{RFC}}\right)=\text{piecewise linear}\quad[-].\tag{7}$$

Consequently, capacity loss

$$Q_{\mathrm{cyc,c}}=A_{\mathrm{cyc}}D_{\mathrm{cyc,c}}\quad[As]\tag{8}$$

of a specific cycle is obtained by multiplying the constant $A_{\mathrm{cyc}}=Q_{\mathrm{nominal}}-Q_{\mathrm{eol}}$.

Cycle identification requires an entire SOC trajectory as input and therefore originally only can be performed

TABLE 1 Output of rainflow algorithm in MATLAB for each cycle c[19]

Quantity	Variable	Description	Unit
Cycle weight	w_c^{RFC}	1 for a full cycle, 0.5 for a half cycle	[−]
Cycle depth	DOD_c^{RFC}	Absolute difference of both SOC magnitudes forming a cycle	[−]
Cycle mean	$SOC_{m,c}^{\mathrm{RFC}}$	Mean value of both SOC magnitudes forming a cycle	[−]
Sample index of cycle maximum	$k_{max,c}^{\mathrm{RFC}}$	–	[−]
Sample index of cycle minimum	$k_{min,c}^{\mathrm{RFC}}$	–	[−]

batch-wise. The total batch-wise cyclic capacity loss of an SOC trajectory is obtained via linear accumulation

$$Q_{\mathrm{cyc}}^{\mathrm{Batch}} = \sum_{c=1}^{N_c} Q_{\mathrm{cyc},c} \quad [As] \tag{9}$$

of damage of all N_c identified cycles.[20]

Due to batch-wise execution, unlike calendric aging, the cyclic aging model originally cannot be expressed as a time-continuous dynamic system. Instead, for the present battery model, a time-discrete cyclic aging model will be developed. As a basis for this, in the following, the concept of "residue" will be presented first.

Residue

The SOC values which are extrema, but have not contributed to a full cycle so far, form half cycles. These SOC values can later form a new half or full cycle with the newly acquired SOC samples of subsequent simulation time-steps. This new cycle will have a higher DOD, and thus higher damage, than the original half cycle.[21] Thus, it is important to preserve these half-cycle extrema and merge them with the new SOC samples before performing the next cycle identification. This set of extrema is referred in current work as "residue" SOC_{residue} and plays a central role in accurate identification of cyclic damage.[21]

One-step time-discrete model

Based on the rainflow algorithm and the concept of residue, a one-step time-discrete model of cyclic aging has been proposed in Loew and Obradovic.[22] This means that by introducing residue as an additional "memory" state, with each new SOC sample an update of cyclic capacity loss can be computed. The procedure is presented in Algorithm 1.

Algorithm 1

One-step time-discrete cyclic aging estimation[22]

```
Input: Existing SOC string $\boldsymbol{SOC}_{\mathrm{exist}}$, Periodic update of a scalar new SOC sample $SOC(k+1)$ at each step $k$.
Output: Periodic update of cyclic capacity loss $Q_{\mathrm{cyc}}^{\mathrm{Step}}$ at each step $k$.
Initialization: Zero total cyclic capacity loss $Q_{\mathrm{cyc}}^{\mathrm{Step}} = 0$ and cyclic capacity loss $Q_{\mathrm{cyc,FC}} = 0$ of full cycles, SOC residue $\boldsymbol{SOC}_{\mathrm{residue}} = \boldsymbol{SOC}_{\mathrm{exist}}$.
while true do.
    1: Merge residue $\boldsymbol{SOC}_{\mathrm{residue}}(k)$ and new SOC sample $SOC(k+1)$
    2: Extract full and half cycles from $\{\boldsymbol{SOC}_{\mathrm{residue}}(k), SOC(k+1)\}$ using Rainflow algorithm, Store residue in $\boldsymbol{SOC}_{\mathrm{residue}}(k+1)$
    3: Calculate cyclic capacity loss based on full cycles, Add result to $Q_{\mathrm{cyc,FC}}(k)$ to obtain $Q_{\mathrm{cyc,FC}}(k+1)$
    4: Calculate cyclic capacity loss based on full and half cycles, Add result to $Q_{\mathrm{cyc,FC}}(k)$ to obtain $Q_{\mathrm{cyc}}^{\mathrm{Step}}(k+1)$
end
```

In a mathematical way, time-discrete cyclic aging estimation is expressed by the update of the residue set

$$\boldsymbol{SOC}_{\mathrm{residue}}(k+1) = G_{\mathrm{cyc,residue}}(\boldsymbol{SOC}_{\mathrm{residue}}(k), SOC(k+1)), \tag{10}$$

by the update of cyclic capacity loss of full cycles

$$Q_{\mathrm{cyc,FC}}(k+1) = G_{\mathrm{cyc,FC}}\left(\boldsymbol{SOC}_{\mathrm{residue}}(k),\ SOC(k+1), Q_{\mathrm{cyc,FC}}(k)\right) \tag{11}$$

and by the update of total cyclic capacity loss

$$Q_{\mathrm{cyc}}^{\mathrm{Step}}(k+1) = G_{\mathrm{cyc,tot}}\left(\boldsymbol{SOC}_{\mathrm{residue}}(k),\ SOC(k+1), Q_{\mathrm{cyc,FC}}(k)\right). \tag{12}$$

Thus, the time-discrete equations $G_{\mathrm{cyc},\cdot}$ represent the algorithmic steps which have been described in Algorithm 1.

2.4 | Time-discrete expression of continuous dynamics

Since the cyclic aging estimation is performed in time-discrete fashion, the other submodels of the battery model as well are expressed as time-discrete state updates:

$$SOC(k+1) = G_{\mathrm{int}}(I(k), Q(k), (1)) + SOC(k) \qquad (13a)$$

$$T(k+1) = G_{\mathrm{int}}(I(k), R_{\mathrm{int}}(k), T(k), T_{\mathrm{ambient}}(k), (1)(2)) + T(k) \qquad (13b)$$

$$Q_{\mathrm{cal}}(k+1) = G_{\mathrm{int}}(t_k, SOC(k), T(k), (1)(2)(3)) + Q_{\mathrm{cal}}(k). \qquad (13c)$$

This is obtained by application of integrators G_{int} to the respective ODEs. The time t_k [s] denotes the absolute time of the current sampling instance k since production of the battery.

3 | CONTROLLER DESIGN USING PARAMETRIC ONLINE RAINFLOW-COUNTING

The battery model of section 2 consists of mainly continuous submodels; namely the electrical, thermal and calendric aging models. For those submodels, the calculation of sensitivities w.r.t. the control inputs, and the integration of the submodels into gradient-based optimal control is straightforward.

In contrast, the cyclic aging submodel contains the Rainflow algorithm which contains algorithmic branches and loops. Thus, a crucial property of the Rainflow algorithm is its discontinuous output-behavior. Furthermore, the number N_c of identified cycles is not known before execution, but bounded by the number of extrema. Therefore, until recently, integration of the cyclic aging submodel into gradient-based Model Predictive Control has not been possible.

3.1 | Concept

In Loew et al,[13] the above mentioned obstacles for a direct implementation of RFC in MPC are overcome by the method of *Parametric Online RainFlow-Counting* (PORFC). In PORFC, all discontinuous parts of the cyclic aging estimation procedure are carried out in a preprocessing step before each execution of the MPC algorithm, as shown in in Figure 4. In order to base this externalized cyclic aging estimation on the same SOC trajectory like the MPC, the preprocessing step has to start with a predictive forward simulation using the same model, sampling and horizon length as the internal simulation of the MPC. Additionally to this prediction, the SOC history is incorporated in the preprocessing via a periodically updated residue SOC_{residue} (see section 2.3.3).

The algorithmic workflow within the controller is as follows:

• *Simulation*: The battery model is simulated over the prediction horizon using the currently measured states $\tilde{\boldsymbol{x}}$ as initial states $\boldsymbol{x}_0$, and the current guess of the optimal

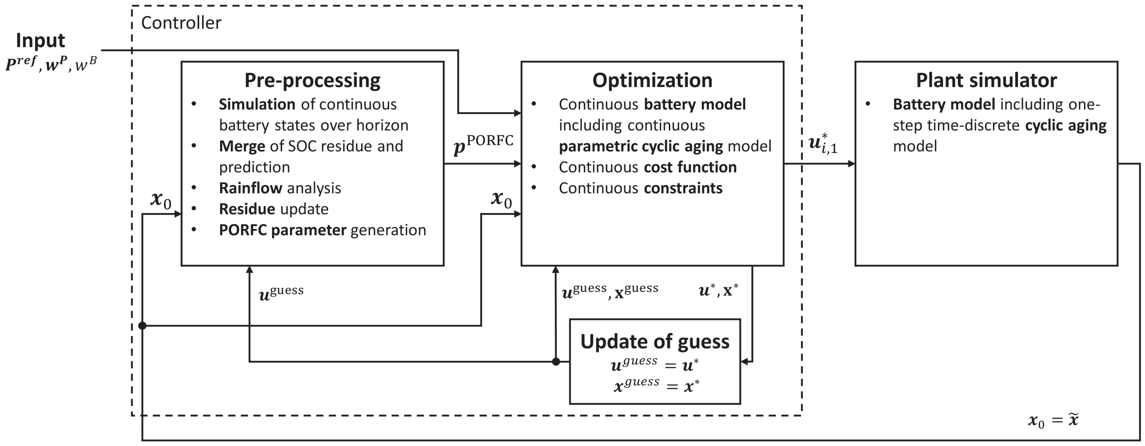

Controller
Input
p^{ref}, w^P, w^B
Pre-processing
Simulation of continuous battery states over horizon
Merge of SOC residue and prediction
Rainflow analysis
Residue update
PORFC parameter generation
p^{PORFC}
Optimization
Continuous battery model including continuous parametric cyclic aging model
Continuous cost function
Continuous constraints
$u^*_{i,1}$
Plant simulator
Battery model including one-step time-discrete cyclic aging model
x_0
x_0
u^{guess}
$u^{\mathrm{guess}}, x^{\mathrm{guess}}$
u^*, x^*
Update of guess
$u^{guess} = u^*$
$x^{guess} = x^*$
$x_0 = \tilde{x}$

FIGURE 4 Control loop for battery plant simulator and MPC using PORFC

control trajectory $\mathbf{u}^{\text{guess}}$. Relevant result is an SOC prediction over the entire horizon.

- *Merge*: The residue is merged with the SOC prediction.
- *Rainflow*: The Rainflow algorithm is used to identify SOC cycles over this merged trajectory. Consequently, it is assumed that the structure of identified cycles does not change within the upcoming optimization run. The term "structure" denotes here positions ($k^{\text{RFC}}_{\text{max},c}$, $k^{\text{RFC}}_{\text{min},c}$) and weights (w^{RFC}_{c}) of cycles. This assumption implies that the controllable extrema in the prediction horizon only can be shifted vertically by the optimization.
- *Residue update*: SOC cycles can be composed by SOC samples only from residue or prediction, or by a mixture of both. However, only controllable samples within the prediction horizon can be altered by the optimization. Especially the measured initial value at the beginning t_0 of the horizon cannot be altered and, therefore, is added to the residue. If a full cycle is detected entirely within the *residue*, both contributing values are discarded from the residue. The reason for this is that also in the future they will never anymore form a cycle with a sample from the prediction and, therefore, are irrelevant for the MPC.
- *Time-varying parameters*: Information from cycle identification is used to fill vectors of time-varying parameters, which are forwarded to the cost function of the MPC. Details on this step are provided in section 3.2.
- *Optimization/MPC*: In the cost function of the MPC, the parameters are used to time-continuously calculate the cyclic capacity loss over the horizon and accumulate it via integration. Finally, the optimization problem is solved and the resulting control variables are applied to the battery plant.

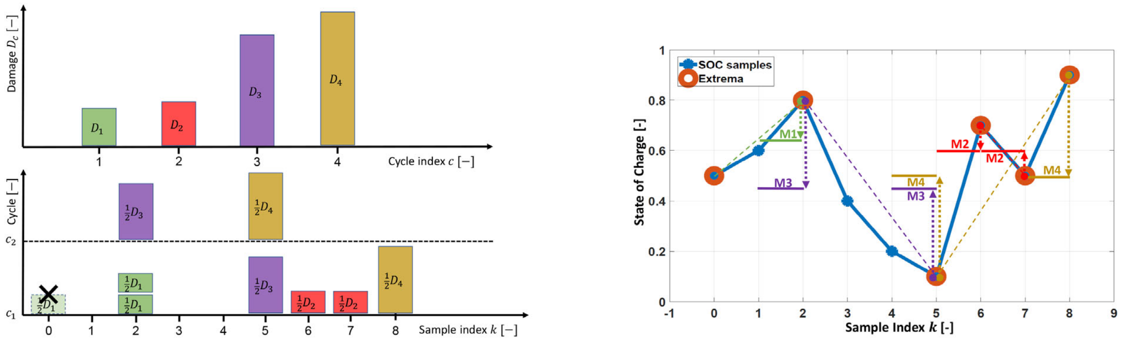
The simulation flow of the control loop in Figure 4 is as such: Measurement of the plant states and controller execution happen at a rather coarse sample time T_{cntrl}. Within the preprocessing step of the controller, the forward simulation is performed with a fixed step size of $T_{\text{sim,preproc}} < T_{\text{cntrl}}$. The PORFC parameters are sampled on the coarse control grid T_{cntrl} since parameters typically only can be inserted into the MPC problem on this granularity for current MPC frameworks. Within the MPC-step of the controller, Multiple-shooting is used where the shooting nodes are defined on the control grid T_{cntrl}. Within each shooting interval, the battery model is simulated on the fine grid $T_{\text{sim,MPC}} = T_{\text{sim,preproc}}$. The optimized control variables and shooting states are output from the controller again at a sample time of T_{cntrl}. The first entries of each control variable are applied to the plant.

3.2 | Time-varying parameters

Distribution of cyclic capacity loss over time: Since information from cycle identification is supposed to be forwarded to the MPC via parameters, which are time-varying over the prediction horizon, the total cyclic capacity loss has to be distributed over the prediction horizon, as visualized in Figure 5A.

Any particular half or full cycle always corresponds to exactly two "complementary" SOC samples. Consequently, the assumption is used that the capacity loss of a cycle can be evenly split over the corresponding two complementary sample instances.

Therefore, the capacity loss of each SOC cycle is split into two halves, which are allocated to the two contributing SOC samples $k^{\text{RFC}}_{\text{max},c}$ and $k^{\text{RFC}}_{\text{min},c}$. For example, cycle 4 is formed by samples $k = 5$ and $k = 8$. Their capacity loss terms therefore are allocated to these samples, as shown by the yellow blocks in Figure 5A. This example also



(A) Distribution of damage per cycle to damage per sample

(B) Piecewise constant mean SOCs as tracking goals

FIGURE 5 PORFC approach for distribution of cyclic damage over time, and setup of SOC tracking goals. A, Distribution of damage per cycle to damage per sample. B, Piecewise constant mean SOCs as tracking goals

shows an important property of the rainflow algorithm, which identifies cycle 4 even though it is interrupted by the nested cycle 2.

Although a SOC sample is not allocated uniquely to one identified cycle, it can at maximum be part of two cycles.[11,23] Thus, in a general case, it can be assumed that a particular SOC sample may belong to either none (when it is not an extremum), or one (when it is part of only one-half or full cycle), or two cycles (when it sits at a cycle junction point).

Setup of the parameters: Figure 5B visualizes the generation of the time-varying parameters. Since each SOC extremum belongs to one or two SOC cycles,[11] the Rainflow algorithm provides one or two mean SOC values per extremum. These mean SOC values (M1-M4) are considered as optimization- or tracking-goals for the current MPC-step. For a specific *cycle*, capacity loss appears twice, once at each of the corresponding sample indices. Thus, the equivalent capacity loss at each sample point has to be halved if both samples lie in the controllable prediction horizon. If, for a given controllable SOC sample, its complementary SOC sample is not controllable (residue or initial value), all damage is allocated to the controllable sample. Here, this is the case for cycle one, where SOC sample $k = 0$ is not controllable. Therefore, the weight is defined to

$$w_c^{\text{PORFC}} = \begin{cases} \frac{1}{2} w_c^{\text{RFC}} & \text{if the complementary SOC sample is controllable} \\ w_c^{\text{RFC}} & \text{if the complementary SOC sample is } \textit{not} \text{ controllable} \end{cases} \tag{14}$$

with an additional dependency on the location of the complementary SOC sample.

Cycle mean values $SOC_{\text{m,c}}^{\text{PORFC}}$ and cycle weights $w_{\text{c}}^{\text{PORFC}}$ are collected in the parameter vector

$$\boldsymbol{p}^{\text{PORFC}} = \left(SOC_{\text{m,1}}^{\text{PORFC}}, SOC_{\text{m,2}}^{\text{PORFC}}, w_1^{\text{PORFC}}, w_2^{\text{PORFC}}\right) \tag{15}$$

which is defined piecewise constant over the control intervals of the prediction horizon.

A more detailed derivation and explanation can be found in Loew et al.[13]

3.3 | Cyclic aging rate

Consequently, the cyclic capacity loss term of PORFC is defined. The capacity loss function is defined by a time-integral over two cost terms, which each represent one potential cycle-contribution of a SOC sample. In a time-differential form, the rate of cyclic capacity loss is expressed by:

$$\dot{Q}_{\text{cyc}}^{\text{PORFC}}(t) = \frac{\mathrm{d}\Delta Q_{\text{cyc}}^{\text{PORFC}}(SOC(t), \boldsymbol{p}^{\text{PORFC}}(t))}{\mathrm{d}t} = \frac{1}{T_{\text{cntrl}}} \sum_{c=1}^{2} Q_{\text{cyc},c}^{\text{PORFC}}\left(SOC(t), SOC_{\text{m},c}^{\text{PORFC}}(t), w_c^{\text{PORFC}}(t)\right) \quad [\text{As/s}]. \tag{16}$$

The cost terms are "switched on" by nonzero cycle weights $w_{c1/2}^{\text{PORFC}}$. Here, the capacity loss

$$Q_{\text{cyc},c}^{\text{PORFC}}(t) = w_c^{\text{PORFC}}(t) K_1 \left(DOD_c^{\text{PORFC}}(t)\right)^{K_2} \quad [As] \tag{17}$$

by individual cycles c is defined based on (8) and on a power-function fitting of (7). The fitting by an analytic function with its parameters K_1 and K_2 is necessary to achieve a continuous optimization problem. Furthermore, DOD

$$DOD_c^{\text{PORFC}}(t) = 2\,|\,SOC_{\text{m},c}^{\text{PORFC}}(t) - SOC(t)\,| \quad [-] \tag{18}$$

here is obtained based on the parametric mean SOC values $SOC_{\text{m},c}^{\text{PORFC}}$.

As an overview, Figure 6 shows the process flows of the novel online PORFC cyclic aging estimation and the conventional offline batch-wise procedure. As depicted, the main differences of PORFC w.r.t. the conventional procedure are the cycle evaluation per time sample, the usage of a rate of cyclic capacity loss and the accumulation via an integral.

3.4 | Optimization problem

The task of the present battery controller is to operate the battery at an economically optimal point between aging cost and power mismatch penalty. Thus the Economic Nonlinear MPC optimization problem is formulated as

$$
\begin{aligned}
&PORFC(t,\boldsymbol{x}(t_0),\boldsymbol{p}^{\mathrm{PORFC}}(t),\quad P^{\mathrm{ref}}(t),w_{\mathrm{P}}(t),w_{\mathrm{B}},d(t)):\\
&\min_{\bar{\boldsymbol{P}}}\; w_{\mathrm{B}}^2\Delta Q_{\mathrm{cal}}(t_{\mathrm{end}})^2+T_{\mathrm{horiz}}w_{\mathrm{B}}^2\int_{t_0}^{t_{\mathrm{end}}}\dot{Q}_{\mathrm{cyc}}^{\mathrm{PORFC}}(t)^2\mathrm{d}t+T_{\mathrm{horiz}}w_{\mathrm{P}}^2(t)\int_{t_0}^{t_{\mathrm{end}}}\left(P(t)-P^{\mathrm{ref}}(t)\right)^2\mathrm{d}t\quad[\$^2]\\
&\qquad\text{such that}\quad \dot{\boldsymbol{x}}(t)=f(t,\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{d}(t),\boldsymbol{x}(t_0))\\
&\qquad\qquad\qquad\quad \boldsymbol{x}(t_0)=\left(SOC_0,T_0,\Delta Q_{\mathrm{cal},0}=0\right)\\
&\qquad\qquad\qquad\quad SOC^{\min}\le SOC(t)\le SOC^{\max}\\
&\qquad\qquad\qquad\quad T^{\min}\le T(t)\le T^{\max}\\
&\qquad\qquad\qquad\quad P^{\min}(t)\le P(t)\le P^{\max}(t)
\end{aligned}
\tag{19}
$$

with the power reference P^{ref} which is variable over the prediction horizon T_{horiz}, the power mismatch penalty w_{P} which can be variable over the horizon, and the aging penalty w_{B} which is constant during the entire simulation. The aging penalty is calculated as

$$w_{\mathrm{B}}=\frac{C_{\mathrm{purch}}Q_{\mathrm{nominal}}U_{\mathrm{ocv,nominal}}}{Q_{\mathrm{nominal}}-Q_{\mathrm{eol}}}\quad[\$/As] \tag{20}$$

with the battery purchase price C_{purch} and the nominal Open Circuit Voltage $U_{\mathrm{ocv,nominal}}$. Each quantity is adjusted to SI-units. In order to avoid a linear cost term of power mismatch, all cost terms in (19) are squared individually. In practical tests, this significantly improved the convergence behavior.

The cyclic aging rate $\dot{Q}_{\mathrm{cyc}}^{\mathrm{PORFC}}(t)$ is obtained by the novel parametric ODE (16). The states of the controller-internal model $f(\cdot)$ are defined as $\boldsymbol{x}=(SOC,T,\Delta Q_{\mathrm{cal}})^T$, and are obtained by the ODEs (1), (2), and (3). The state ΔQ_{cal} is initialized to zero since it represents the *added* capacity loss within the horizon.

The control input $u(t)=P(t)$ is defined by the piecewise-constant charge−/discharge power. The external input $d(t)=T_{\mathrm{ambient}}(t)$ is the ambient temperature.

The constraints on the system states SOC and T, and on the input P are denoted as box constraints with respective upper and lower limits. For the power, the limits

$$P^{\min}(t)=\begin{cases}0 \text{ if } P^{\mathrm{ref}}(t)\ge 0\\ P^{\mathrm{ref}}(t) \text{ if } P^{\mathrm{ref}}(t)<0\end{cases}\text{and } P^{\max}(t)=\begin{cases}P^{\mathrm{ref}}(t) \text{ if } P^{\mathrm{ref}}(t)\ge 0\\ 0 \text{ if } P^{\mathrm{ref}}(t)<0\end{cases} \tag{21}$$

are time-varying, and depend on the power reference to enforce a meaningful power. For instance, in the case of a discharging reference ($P^{\mathrm{ref}}(t)\ge 0$) only a positive power of maximum the power reference power can be chosen by the optimization. Outside of these limits following meaningless situations would occur:

- A negative power would lead to more cyclic capacity loss than zero power, and additionally cause higher power mismatch penalty.

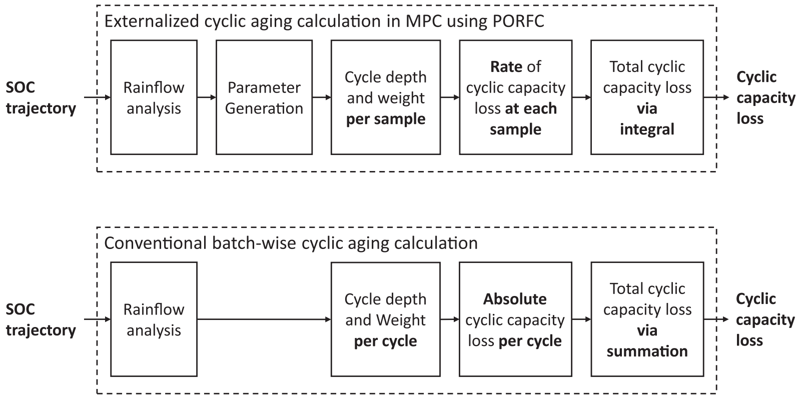

Externalized cyclic aging calculation in MPC using PORFC
SOC trajectory
Rainflow analysis
Parameter Generation
Cycle depth and weight **per sample**
Rate of cyclic capacity loss at each **sample**
Total cyclic capacity loss **via integral**
Cyclic capacity loss
Conventional batch-wise cyclic aging calculation
SOC trajectory
Rainflow analysis
Cycle depth and Weight **per cycle**
Absolute cyclic capacity loss **per cycle**
Total cyclic capacity loss **via summation**
Cyclic capacity loss

FIGURE 6 Process flow of calculation of cyclic capacity loss using PORFC and using the conventional batch-wise procedure (9). The differences are highlighted in bold

- A greater power than reference power would lead to more cyclic capacity loss than reference power, and would additionally cause power mismatch penalty.

4 | RESULTS, ANALYSIS, AND VALIDATION

In this section, the closed-loop behavior of the MPC of section 3 with the battery plant model of section 2 is presented and compared to other controllers in a realistic simulation case. Furthermore, the controller is assessed in terms of optimality, cyclic aging estimation, and computational time. A validation of the plant model with a special emphasis on the calendric and cyclic aging submodels can be found in section A.2.

4.1 | Comparison to other controllers in PJM market setting

This section presents a comparison of controllers for the use case of the US-American real-time ancillary service market of the transmission operator "PJM Interconnection LLC." After an overview on the PJM market setting, the controller configurations and the simulation results are provided.

4.1.1 | PJM market setting

In the PJM frequency regulation market, the system operator sends an *Automatic Generation Control* (AGC) based secondary frequency control signal every 2 seconds to the participating generating units.[24,25] The control signal has two components: the RegA signal is the low-pass and the RegD signal is the high-pass filtered component of the area control error (ACE). RegD exhibits very high ramp rates but is designed to have a zero mean over a certain duration.[24,25] Thus, the RegD signal is ideal for storage units and has been considered as the reference regulation signal $r(t)$ for the present battery system. The economics for providing the frequency regulation service depend on two components[26]:

- The power-based option fee which is paid *to* the battery operator based on the offered regulation power P_{offered}.
- The energy-based mismatch penalty w_P [\$/Ws] which is paid *by* the battery operator depending on the mismatch between energy provided and energy requested by the system operator via the RegD signal.

These prices are provided to the generating entities every 5 minutes for the next 5 minutes interval.[24,25]

The dynamic performance of the participating entity is further constrained by the system operator using a performance score $\delta \in [0, 1]$, where a value of 1 represents complete tracking of the RegD signal.[24,25,27] The participating entity must maintain a minimum performance score to be qualified for participation in the regulation market. Originally, this score is calculated based on tracking error, response delay, and correlation.[25] In the present work, response delay and correlation are neglected for simplicity. This leads to the simplified performance score $\delta(t) = \frac{P(t)}{P^{\text{ref}}(t)}$ $[-]$.

The key simulation parameters and input signals for the battery plant model and the frequency regulation market are summarized in Table 2.

4.1.2 | Customized PORFC MPC

In the following, the optimization problem (19) is customized to the PJM market setting. The reference power signal $P^{\text{ref}}(t) = P_{\text{offered}} r(t)$ results from the offered regulation power and the unitless RegD signal. An additional constraint $\delta^{min} \leq \delta(t) \leq 1$ for the performance score is added to prevent disconnection from the grid (see section 4.1.1). The MPC is implemented in the MPC framework ACADO Toolkit.[28]

4.1.3 | Constraint-based controller

The reference controller is designed to follow the power reference and only deviate from it when the SOC-limits are reached. This controller therefore also can be considered as "rule-based."

4.1.4 | MPC with piecewise-affine cycle identification ("PWA MPC")

As a more sophisticated comparison object, the time-discrete MPC formulation of Koller et al[10] is implemented which focuses explicitly on optimal control of battery aging. Here, cycle identification is approximated via a piecewise-affine dynamical system. Therefore, this formulation is referenced as "PWA MPC" in the present work.

In the original work, a Mixed-integer Quadratic Programming (MIQP) formulation is used. In the present work, only the continuous decision on power is required. Therefore, a QP formulation is sufficient. This MPC is implemented in MATLAB using the fmincon optimization solver with two QP steps per MPC execution.

The cycle identification method of this formulation is designed according to the *Simple Range Counting* of

TABLE 2 Key parameters and input signals of the present Li-ion battery model

Quantity	Variable	Value	Unit
Battery peak power	P_{limit}	20	[kW]
Battery nominal capacity	Q_{nominal}	200	[Ah]
Defined EOL capacity	Q_{eol}	160	[Ah]
Number of cycles until EOL	-	3000 (at $DOD = 0.8$)	[–]
Plant cyclic aging curve	Piecewise linear	–	[–]
MPC cyclic aging parameters in (19)	K_1, K_2	$3.8 \cdot 10^{-4}$, 2	[–]
Battery purchase price	C_{purch}	300	[$/kWh]
Maximum state of charge	SOC^{max}	0.9	[–]
Minimum state of charge	SOC^{min}	0.1	[–]
Maximum battery temperature	T^{max}	65	[°C]
Minimum battery temperature	T^{min}	−10	[°C]
Ambient temperature	T_{ambient}	25	[°C]
Regulation power offered	P_{offered}	18.2	[kW]
Minimum performance score[a]	δ^{min}	0.4	[–]
Power mismatch penalty	w_{P}	Time-series	[$/Ws]
Normalized RegD control signal	$r(t)$	Time-series	[–]

[a] As per PJM regulation,[25] a minimum performance score of 0.4 is necessary to avoid disconnection from the grid.

ASTM[4] which is simplistic in comparison to Rainflow counting. Here, this is implemented by adding two extra states $\boldsymbol{DOD} = (DOD(k)_{\text{charge}}, DOD(k)_{\text{discharge}})^T$ to the controller-internal system model, which represent instantaneous charging and discharging DODs, respectively. The evolution of these DOD states is governed by a piecewise affine (PWA) dynamical system

$$\boldsymbol{DOD}(k+1) = \boldsymbol{A}(k)\boldsymbol{DOD}(k) + \boldsymbol{b}(k)|SOC(k+1) - SOC(k)|. \tag{22}$$

The input to the PWA system is the change in SOC over the current simulation time step. The dynamics of DOD is defined by the system matrix

$$\boldsymbol{A}(k) = \begin{cases} \text{diag}(1,0) \text{ if } P(k) > 0 \wedge P(k-1) > 0 \\ \text{diag}(0,1) \text{ if } P(k) < 0 \wedge P(k-1) < 0 \\ \text{diag}(0,0) \text{ if } (P(k) > 0 \wedge P(k-1) < 0) \vee (P(k) < 0 \wedge P(k-1) > 0) \\ \text{diag}(1,1) \text{ if } P(k) = 0 \end{cases} \tag{23}$$

and the input vector

$$\boldsymbol{b}(k) = \begin{cases} (1,0)^T \text{ if } P(k) > 0 \\ (0,1)^T \text{ if } P(k) < 0 \\ (0,0)^T \text{ if } P(k) = 0 \end{cases} \tag{24}$$

which both are switched by the net battery power $P(k)$. The first two cases for the system matrix lead to growth of the instantaneous (dis)charging DOD. The third case diag(0, 0) results in a reset of the DOD states at each sign change of power. The fourth case holds the DOD constant.

Figure 7 shows the evolution of the DOD states for a defined SOC trajectory. Here, following disadvantages of this PWA cycle identification method become apparent:

- Cycle growth is terminated even by small nested cycles, for instance once at t = 1400 seconds and several times at t > 4000 seconds.
- The detected maximum DOD of up to 0.12 is much lower than the actual SOC excursion which by visual inspection is greater than 0.2 and defined by the SOC extrema at t = 1900 seconds and t = 5000 seconds.

For a comparison, a second plot in Figure 7 shows DODs which are obtained from the cycle identification method of the battery plant model of section 2 using the standard RFC algorithm. At each sample, that DOD is plotted which is related to the instantaneous SOC sample. Following advantageous behavior can be seen:

- Continuation of cycle growth after completion of a small nested cycle, for instance at t = 1600 seconds.
- During discharge at t = 3050 seconds, the discharging DOD exhibits a jump in magnitude since the nested full cycle is completed and cycle identification switches to a former larger cycle.

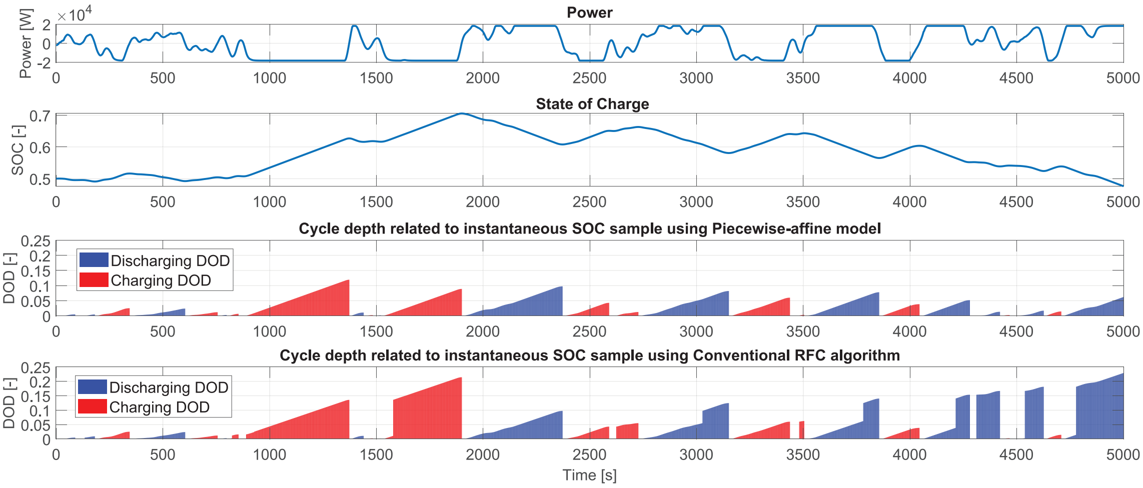


FIGURE 7 Evolution of DOD states for the PWA and the RFC cycle identification

- As expected, the final charging and discharging DODs related to the same nested cycle are equal, like for instance at t = 2700 seconds and t = 3050 seconds.
- The visible maximum SOC excursion of DOD = 0.2 is reached for a charging half cycle at t = 1900 seconds and for a discharging half cycle at t = 5000.

To conclude, the PWA cycle identification systematically detects too low cycle depths. Consequently, also cyclic capacity loss is underestimated. Furthermore, the present comparison highlights that for correct cycle identification it is necessary to memorize past SOC extrema, as it is done in the novel PORFC formulation.

The optimization problem for the PWA MPC is defined as

$$PWA\left(\boldsymbol{P}^{\mathrm{ref}}(t),\mathbf{d}(t),w_{\mathrm{DOD}}^{\mathrm{PWA}},w_{\mathrm{SOC}}^{\mathrm{PWA}},w_{\mathrm{P,wear}}^{\mathrm{PWA}},w_{\mathrm{P}}^{\mathrm{PWA}}\right):$$

$$\min_{\bar{\boldsymbol{P}}_{\mathbf{charge}},\bar{\boldsymbol{P}}_{\mathbf{discharge}}} \sum_{k=1}^{N_{\mathrm{cntrl}}} w_{\mathrm{DOD}}^{\mathrm{PWA}}\left(DOD(k)_{\mathrm{charge}}^{2}+DOD(k)_{\mathrm{discharge}}^{2}\right)+w_{\mathrm{SOC}}^{\mathrm{PWA}}\left(SOC(k)-SOC_{\mathrm{ref}}^{\mathrm{PWA}}\right)^{2} \\ +w_{\mathrm{P,wear}}^{\mathrm{PWA}}\left(P_{\mathrm{charge}}(k)^{2}+P_{\mathrm{discharge}}(k)^{2}\right)+w_{\mathrm{P}}^{\mathrm{PWA}}\left(P(k)-P^{\mathrm{ref}}(k)\right)^{2} \tag{25a}$$

$$\text{subject to}\quad P(k)=P_{\mathrm{charge}}(k)+P_{\mathrm{discharge}}(k) \tag{25b}$$

$$(22) \tag{25c}$$

$$SOC(k+1)=SOC(k)+\frac{T_{\mathrm{cntrl}}}{Q\ \ U_{\mathrm{ocv,nominal}}} \times\left(-\eta_{\mathrm{charge}}P_{\mathrm{charge}}-\frac{P_{\mathrm{discharge}}}{\eta_{\mathrm{discharge}}}\right) \tag{25d}$$

$$\mathbf{x}_0=\left(SOC_0,\mathbf{DOD}_0=(0,0)^{T}\right) \tag{25e}$$

$$SOC^{\min}\le SOC(k)\le SOC^{\max} \tag{25f}$$

$$\min(0,P_{\mathrm{ref}}(k))\le P_{\mathrm{charge}}(k)\le 0 \tag{25g}$$

$$0\le P_{\mathrm{discharge}}(k)\le \max(0,P_{\mathrm{ref}}(k)) \tag{25h}$$

$$P_{\mathrm{charge}}\ \ P_{\mathrm{discharge}}=0. \tag{25i}$$

The decision variables for the OCP are charging P_{charge} and discharging $P_{\mathrm{discharge}}$ power which are applied zero-order-hold over the N_{cntrl} steps k of the prediction horizon. Their sum (25b) is the net power of the battery. Both powers are also the inputs to the SOC difference Equation (25d), where the charge/discharge efficiencies are set to $\eta_{\text{charge}} = 1$ and $\eta_{\text{discharge}} = 1$ in the present work. In (25e), the DODs are initialized to zero at the beginning of the prediction horizon for each new MPC step. Constraint (25f) enforces a SOC within its limits, and (25g, 25h) enforce meaningful (dis)charging powers analogously to (21). A complementary constraint (25i) is imposed on the system to prevent simultaneous charging and discharging control actions. The cost function (25a) contains three terms for aging costs and one term for power mismatch penalty. The aging cost terms quadratically penalize the present approximate DOD, the deviation of SOC from a SOC reference $SOC_{\text{ref}}^{\text{PWA}} = 0.5$ and the (dis)charge powers.

4.1.5 | Comparison of simplified PORFC MPC to constraint-based controller and PWA MPC

Figure 8 shows the closed-loop simulation results for the constraint-based controller, the PWA MPC and the new PORFC MPC. For PORFC, the cases with ("PORFC-R MPC") and without ("PORFC MPC") utilization of the residue are shown. In order to achieve a fair comparison to the PWA MPC with its reduced capabilities, in the PORFC MPC the performance score constraint and the calendric aging cost function are turned off for the present simulation. Additionally, a fixed power deviation penalty of $w_{\text{P}} = 9.37 \cdot 10^{-9}\$/Ws$ is used.

The PORFC(−R) MPCs are parameterized using the physical and economical parameters in Table 2. In contrast, for the PWA MPC, profit-optimal aging weights ($w_{\text{DOD}}^{\text{PWA}} = 10^{-4}, w_{\text{SOC}}^{\text{PWA}} = 10^{-9}, w_{\text{P,wear}}^{\text{PWA}} = 10^{-20}$) are obtained by an extensive parameter study of 126 simulations of 10 minutes each using exactly the power profile of the present test. Interestingly, in essence, these weights switch off the SOC and power cost terms. This indicates that the remaining DOD cost has highest correlation to the true cyclic aging cost; despite abovementioned underestimation of DODs. The mismatch penalty is set to $w_{\text{P}}^{\text{PWA}} = w_{\text{P}}^2 = 8.78 \cdot 10^{-17}$.

In order to cause significant cyclic aging within the simulation, it is assumed that the evolution of SOC started from $SOC = 0$ before the current simulation time. Thus, the battery is operating within a big charging half-cycle already at the start of the simulation. This information is used to initialize the residue SOC set $SOC_{\text{residue}} = 0$ of the PORFC-R MPC.

During the simulation, the SOC constraints are not reached. Thus, the power of the constraint-based controller is equal to the power reference. All MPCs deviate from the reference power to a certain extent. The PWA MPC frequently even chooses zero power. In contrast, the PORFC MPC stays very close to the power reference. The PORFC-R MPC follows the discharging reference to a certain extent but does not perform any further charging to avoid further increase of the high charging DOD from the past. This is caused by its awareness of the big charging half-cycle via the residue. Consequently, the PORFC-R MPC stays below the $SOC = 0.5$ level during the present simulation. However, it is not obvious why it does not follow the charging power at least up to this

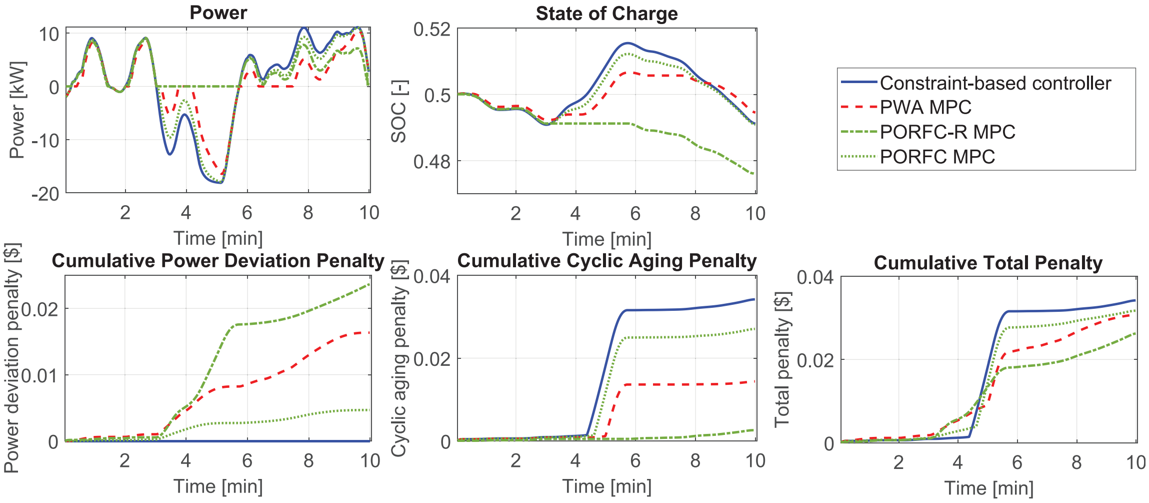


FIGURE 8 Comparison of PORFC MPC to piecewise-affine MPC[10] and a rule based controller. For the MPCs: Prediction horizon 120 seconds, controller sampling time 1 second

level without incurring high cyclic aging. This effect will be investigated in the future.

The PORFC-R MPC incurs highest cumulative power deviation penalty but also lowest cyclic aging penalty. The PWA MPC reaches intermediate values for both penalties. In terms of total penalty, the PWA MPC improves w.r.t. the constraint-based controller by 10%. The PORFC MPC reaches similar total penalty. The PORFC-R MPC clearly exhibits lowest total penalty which is 23% below the one of the constraint-based controller.

It should be noted that the PORFC(−R) MPCs achieve this performance after straightforward parameterization with given physical and economical parameters. In contrast, the PWA MPC has been extensively tuned specifically for the present case.

To put the penalties of the order $O(0.01\$/10\ \text{minutes})$ into perspective, it should be noted that the revenue (power-based option fee, see section 4.1.1) reaches up to an order $O(10\$/10\ \text{minutes})$. Profit is the revenue subtracted by the penalties. As a consequence, the variations in profit for different controllers remain fairly limited for the present PJM market setting.

4.1.6 | Comparison of full PORFC MPC to rule-based controllers

The above test involves several simplifications in the PORFC MPC in order to be comparable to the PWA MPC. Therefore, in the following, the PORFC-R MPC (with residue) is tested in a full configuration with these additional features:

- Calendric aging cost function active.
- Performance score constraint active.
- Power deviation penalty factor variable over time.

Four controllers are compared: the "baseline PORFC-R MPC" with a performance score constraint of $\delta_{\text{min}} = 0.4$, an "unconstrained PORFC-R MPC" without performance score constraint $\delta_{\text{min}} = 0$, the constraint-based controller for power following, and a policy of applying zero battery power.

The present test is extreme in the sense that the variable power deviation penalty is at a very low level. Therefore, pure power following is far from being optimal, and the MPCs are expected to converge to strategies of low or even zero power.

As shown in Figure 9, during large time spans the PORFC-R MPCs do not apply the full reference power, and the baseline PORFC-R MPC hits but does not violate its lower power limit. The unconstrained PORFC-R MPC in general applies low power values to limit the DODs due to cyclic aging, and to remain close to $SOC = 0.5$ due to calendric aging. Thereby, its power deviation penalty almost reaches the one for zero-power policy. The baseline PORFC-R MPC exhibits an intermediate power deviation penalty. On the other hand, the MPCs are able to drastically reduce battery aging. Since the zero-power policy does not cause SOC cycles, its aging trajectory is at

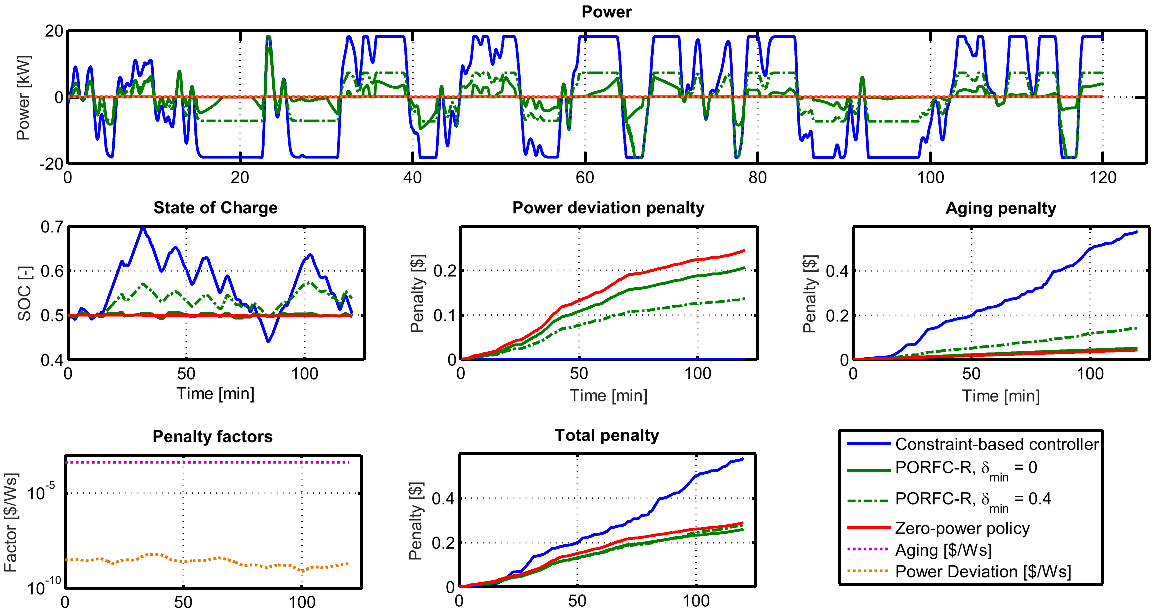


FIGURE 9 Comparison of PORFC MPC including calendric aging, performance score constraint and variable power deviation penalty factor to two rule-based controllers. For the MPCs: Prediction horizon 300 seconds, controller sampling time 1 second

a very low level and only reflects the unavoidable calendric aging.

The behavior of total penalty exhibits that little power following is advantageous, and that the intermediate strategies of the MPCs are superior to the rule-based controllers. More specific, the zero-power policy reduces total penalty by 50% w.r.t. the constraint based controller. The baseline PORFC-R MPC additionally reduces total penalty by 3%, and the unconstrained PORFC-R MPC by 10% w.r.t. the zero-power policy.

Furthermore, this test proves that the present PORFC-R MPC can handle time-varying penalty factors very well. It may be worth mentioning that the penalty factors are interpolated over time, in order to achieve smooth transitions, and to facilitate convergence of the optimization algorithm.

4.2 | Analysis and validation of the MPC

4.2.1 | Degree of optimality

In order to validate the operational performance of the PORFC-R MPC, a comparison to a "one-shot" OCP solution is presented in the following. The latter algorithm has full information about the reference power and penalty costs over the entire simulation duration. Its optimization horizon equals the simulation duration. The MPC problem is an approximation of this OCP, and is based on the formulation in section 4.1.2. Different horizon lengths are tested for the MPC. A standard power reference signal (constant and ramp) over a simulation time of 1 hour is used, as shown in Figure 10.

The OCP solution exhibits a reduced and constant charging power level already at the beginning of the simulation. Exactly in parallel to the reference ramp, the charging power goes down to zero and remains there while the reference power crosses zero. After this crossing the discharging power again increases in parallel to the reference until a new constant power level is reached. In absolute terms, this discharging power level is lower than the earlier charging power level. This difference can be explained by the longer time span of discharging in comparison to charging reference power. By decreasing the applied discharging power the corresponding cycle DOD remains limited.

The MPCs start charging from reference power and gradually converge to the OCP solution as the cycle DOD grows. This difference in initial behavior may be caused by the difference in cost function design, where the OCP penalizes absolute power mismatch and the PORFC-R MPC penalizes a square of power mismatch. Especially the point of reaching zero power is matched very well, although some spikes are visible whose cause is unclear. Due to their limited horizon, the MPCs initially again rather follow the discharging reference power. However, later on, their power drops even below the OCP solution in order to compensate for the already high DOD.

During the discharging phase, the MPCs with longer horizon approximate better the OCP solution. In terms of total economic penalty (mismatch + aging), OCP exhibits best performance, followed by the MPCs in descending order of horizon length. Worst performance results from following the reference power.

Note that the OCP does not converge toward zero power at the end, only because of the fact that its horizon ends at the end of the simulation time t = 60 minutes. In other words, such a constant non-negative power is an optimum control result only for limited time spans.

On a lower level, the degree of optimality of the optimization problems at separate MPC steps is quantified using the Karush Kuhn Tucker (KKT) values.[2] KKT values indicate how close the solution of the constrained optimization problem is to its closest local optimum. Figure 11 shows the KKT values over the duration of the PJM simulation of section 4.1. During most of the simulation time, the KKT values remain below 10^{-3}. Therefore, the optimization problems can be considered as fairly converged.

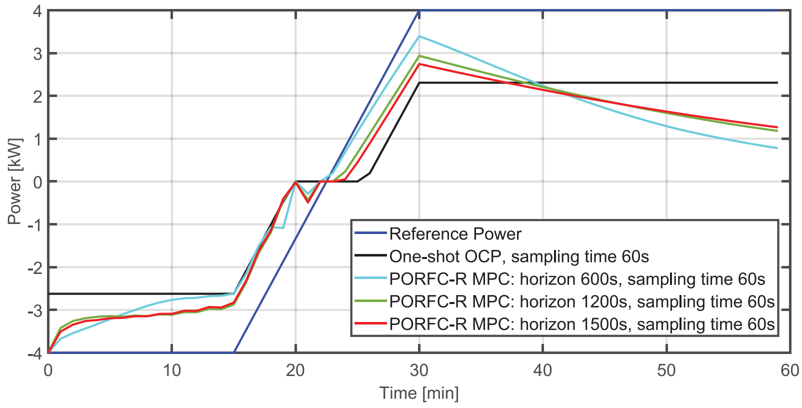


FIGURE 10 Validation of the PORFC-R MPC against a one-shot optimal control solution

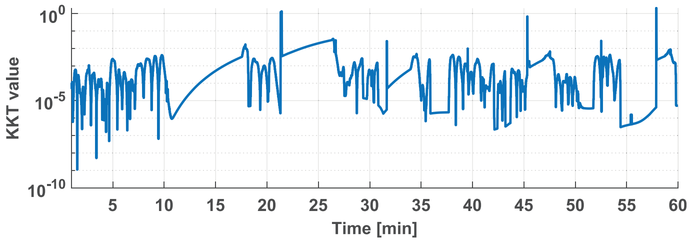


FIGURE 11 KKT values for the PJM setting

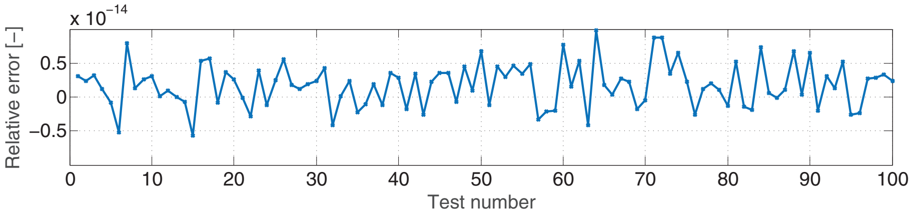


FIGURE 12 Relative error between PORFC cost function in one-shot fashion and conventional cyclic aging calculation

4.2.2 | Comparison of PORFC in one-shot fashion to conventional cyclic aging calculation

In section 3.2, it has been explained how cyclic capacity loss is distributed and decoupled over time. In order to check, if this approximation introduces an error, it is compared to the conventional cyclic aging calculation (9). For PORFC, cyclic aging is calculated by $\int_0^{t_{k_{end}}} \dot{Q}_{cyc}^{PORFC}(t)dt$ using (16). The process flows of the two approaches have been shown in Figure 6.

The present comparison is based on random SOC sets of $k_{\text{set}} = 10{,}000$ samples each. Consequently, both cyclic aging functions are applied in one-shot to the full SOC sets $k = [0, k_{\text{set}}]$ which results in two cyclic capacity loss values for each SOC set. The relative errors of 100 of those tests are depicted in Figure 12. For all the tests, it turns out that both approaches result in the same total cyclic capacity loss up to machine precision.

Therefore, it can be concluded that the preprocessing step and the cyclic capacity loss dynamics of PORFC do not introduce errors in terms of absolute capacity loss.

4.2.3 | Comparison of PORFC in moving-horizon fashion to conventional cyclic aging calculation

In section 3.1, the PORFC concept is explained of utilizing a residue to correctly estimate cyclic aging in a moving-horizon fashion. In order to validate this approach, in the present assessment, the dynamic evolution of cyclic aging over simulation time for PORFC is compared with the evolution for conventional cyclic aging calculation.

Here, PORFC is evaluated at each time step k on a moving horizon $[k, k + N]$ of $N = 100$ samples, while the residue is carried along from previous time steps. In contrast, the conventional calculation is applied to all samples $[0, k]$ up to the current one k. Figure 13 shows that in principle both approaches match w.r.t. the dynamics and the end values. The only difference lies in the leading time shift of 100 samples by PORFC. This, however, is expected since PORFC is evaluated on the prediction horizon and therefore outputs a future capacity loss.

4.2.4 | Length of SOC residue

In order for the formulated controller to be implemented on an embedded hardware, the arrays used should have a finite length such that a finite memory space can be allocated on the hardware. Specifically the length of the SOC residue vector, which is used in the preprocessing algorithm, is not clear a priori.

However, numerous practical tests have shown that its length typically remains well below 100 entries. As an example, in Figure 14 the length of the SOC residue even remains at around 10 during a closed-loop simulation of 18 hours.

4.2.5 | Computational performance

Table 3 shows the average computational time taken by the preprocessing and MPC step to generate the optimal control input for the plant at every simulation step (compare with flowchart of Figure 4). The averaging is done over 600 execution steps each. Results are shown for different combinations of controller sample times and lengths of the prediction horizon.

FIGURE 13 Evolution of cyclic aging from PORFC cost function (using residue and a moving horizon of [k, $k + N$]) in comparison to the conventional approach over [0, k]

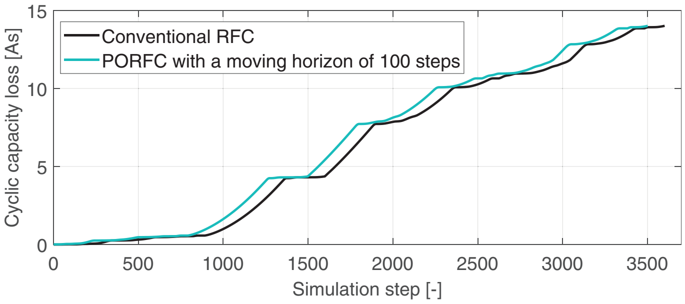


FIGURE 14 Number of entries in the SOC residue set plotted over the controller execution steps

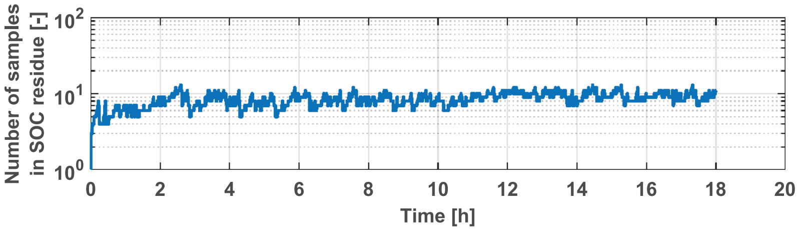
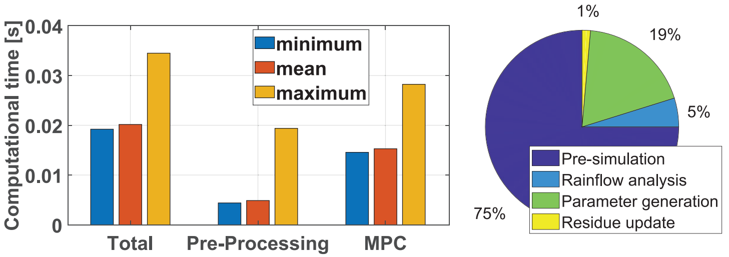


TABLE 3 Average computational time of PORFC MPC at different sample times and prediction horizon lengths

Sample time	Discretization steps	Horizon length	Computational time of controller Preprocessing	MPC	Total
100 ms	50	5 seconds	1.7 ms	3.6 ms	5.3 ms
100 ms	100	10 seconds	2.8 ms	7.4 ms	10.2 ms
100 ms	200	20 seconds	5.0 ms	16.2 ms	21.1 ms
100 ms	300	30 seconds	6.7 ms	27.4 ms	34.1 ms
1000 ms	20	20 seconds	1.1 ms	1.5 ms	2.6 ms
1000 ms	200	200 seconds	5.0 ms	15.8 ms	20.8 ms

FIGURE 15 Computational times of the PORFC MPC for a setting with sample time of 100 ms and horizon length of 20 second. A, Minima, maxima and mean values of computational times. B, Breakdown of computational cost within preprocessing step



(A) Minima, maxima and mean values of computational times

(B) Breakdown of computational cost within pre-processing step

As expected, the preprocessing causes less effort than the MPC execution, and an increase in prediction time leads to increases in both modules. The increase is conveniently sublinear for the preprocessing and only slightly superlinear for the MPC execution. In any case the average total computational time is significantly smaller than the sample time, assuring real-time implementability of the present controller.

Figure 15A shows that in rare cases the computational times can be significantly higher than their mean values. This is especially true for the preprocessing step. In the present test, the maxima of preprocessing and MPC execution time do not occur at the same time since the maximum of total time is lower than the sum of the individual maxima. However, even if the maxima of preprocessing and MPC execution time occurred at the same time, the total computational time would remain below 50 ms.

Figure 15B provides a further breakdown of computational cost within the preprocessing step. Obviously, the presimulation of the battery dynamics consumes most time. However, this part already has high numerical efficiency since it is based on a code-generated integrator from `ACADO Toolkit`.[28] Concluding, there remains only minor improvement potential in the array-handling of the parameter generation.

5 | CONCLUSION AND OUTLOOK

5.1 | Conclusion

A comprehensive Li-ion battery model including electrical, thermal, and aging submodels has been presented. For calendric aging, a novel time-variant term has been proposed for time-continuous simulation of the square-root aging dynamics. For cyclic aging, cycle identification and damage accumulation have been cast into a novel time-discrete dynamical model.

Consequently, a MPC comprising an exact cyclic aging cost function has been formulated for Li-ion batteries. This cyclic aging cost formulation is enabled by

- Performing a Rainflow-analysis prior to each MPC step based on a pre-simulation of the system dynamics.
- Distributing and decoupling of cyclic capacity loss over time.
- Memorizing a condensed set of SOC samples from past half cycles.

In a simplified market setting, the novel MPC incurs 13% less total penalty than a state-of-the-art MPC from literature and 23% less total penalty than a conventional rule-based controller. Total penalty here is the sum of power mismatch penalty and aging cost.

In a full market setting with additional constraints, the novel MPC exhibits its ability to adapt even to unusual economic situations, and to outperform manually chosen control policies.

A validation of the cyclic aging cost function shows that correct aging estimation in the MPC is maintained despite the distribution of capacity loss over time and the moving-horizon mode.

5.2 | Outlook

The promising results of the present study motivate the following future work:

- Integration of the plant model and controller in an in-house hybrid power plant simulation suite.
- Testing with a real-life battery energy storage system.
- Application of the controller formulation to further domains where fatigue or aging play a crucial role for profitable operation.

NOMENCLATURE

Variable/ Abbreviation	Unit	Explanation
$k_{\max,c}$	[−]	Sample index of cycle maximum
$k_{\min,c}$	[−]	Sample index of cycle minimum
$\boldsymbol{u}$		Control variables vector
w_c	[−]	Weight of one cycle
$\boldsymbol{x}$		State vector
DOD	[−]	Depth of discharge
P	[W]	Charge/discharge power of battery
Q	[As]	Instantaneous charge capacity of battery
Q_{cal}	[As]	Calendric capacity loss
Q_{cyc}	[As]	Cyclic capacity loss
Q_{eol}	[As]	Charge capacity at end of life
Q_{nominal}	[As]	Charge capacity at the time of production
SOC	[−]	State of charge
T	[K]	Cell temperature of battery
T_{ambient}	[K]	Ambient temperature
T_{cntrl}	[s]	Controller sample-time
T_{horiz}	[s]	Length of prediction horizon
MPC		Model predictive control
OCP		Optimal control problem
ODE		Ordinary differential equation
PORFC		Parametric online Rainflow-counting
RFC		Rainflow-counting

ACKNOWLEDGMENT

Open access funding enabled and organized by Projekt DEAL.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

Data available on request from the authors

ORCID

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

REFERENCES

1. Schmalstieg J, Käbitz S, Ecker M, Sauer DU. A holistic aging model for li(nimnco)o2 based 18650 lithium-ion batteries. *J Power Sources*. 2014;257:325-334.
2. Rawlings JB, Mayne DQ, Diehl MM. *Model Predictive Control: Theory, Computation, and Design*. 2nd ed. Madison, Wisconsin: Nob Hill Publishing; 2017.
3. Laresgoiti I, Käbitz S, Ecker M, Sauer DU. Modeling mechanical degradation in lithium ion batteries during cycling: solid electrolyte interphase fracture. *J Power Sources*. 2015;300:112-122.
4. ASTM, Standard Practices for Cycle Counting in Fatigue Analysis, 1985.
5. Shi Y, Xu B, Tan Y, Kirschen D, Zhang B. Optimal battery control under cycle aging mechanisms in pay for performance settings. *IEEE Trans Autom Control*. 2019;64:2324-2339.
6. 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;33:2248-2259.
7. Wang C, Du Z, Ni Y, Li C, Zhang G. Coordinated predictive control for wind farm with bess considering power dispatching and equipment ageing. *IET Gener Transm Distrib*. 2018;12:2406-2414.
8. R. S. Yallamilli, L. K. Vedulla, M. K. Mishra 2018, Cost savings oriented microgrid control strategy considering battery degradation. Paper presented at: 2018 7th International Conference on Renewable Energy Research and Applications (ICRERA), pp. 661–666.
9. He G, Chen Q, Kang C, Pinson P, Xia Q. Optimal bidding strategy of battery storage in power markets considering performance-based regulation and battery cycle life. *IEEE Transact Smart Grid*. 2016;7:2359-2367.
10. M. Koller, T. Borsche, A. Ulbig, G. Andersson, Defining a degradation cost function for optimal control of a battery energy storage system. Paper presented at: 2013 IEEE Grenoble Conference, 2013, pp. 1–6.
11. Y. Shi, B. Xu, Y. Tan, B. Zhang, A convex cycle-based degradation model for battery energy storage planning and operation, Paper presented at: 2018 Annual American Control Conference (ACC), IEEE, 2018, pp. 4590–4596.
12. Xu B, Shi Y, Kirschen DS, Zhang B. Optimal battery participation in frequency regulation markets. *IEEE Trans Power Syst*. 2018;33:6715-6725.
13. S. Loew, D. Obradovic, A. Anand, A. Szabo, Stage cost formulations of online rainflow-counting for model predictive control of fatigue, Paper presented at: 2020 European Control Conference (ECC), 2020, pp. 475–482.
14. 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;1618:022041.
15. Mousavi SM, Nikdel GM. Various battery models for various simulation studies and applications. *Renew Sust Energ Rev*. 2014;32:477-485.
16. Gao Z, Chin C, Woo W, Jia J. Integrated equivalent circuit and thermal model for simulation of temperature-dependent lifepo4 battery in actual embedded application. *Energies*. 2017;10:85.
17. Maheshwari A. *Modelling, Aging and Optimal Operation of Lithium-ion Batteries*. Eindhoven: Technische Universiteit Eindhoven; 2018.
18. Wang J, Liu P, Hicks-Garner J, et al. Cycle-life model for graphite-lifepo4 cells. *J Power Sources*. 2011;196:3942-3948.
19. The MathWorks Inc Rainflow counts for fatigue analysis, 2018.
20. Miner MA. Cumulative damage in fatigue. *Journal of Applied Mechanics*. 1945;3:159-164.
21. Marsh G, Wignall C, Thies PR, et al. Review and application of rainflow residue processing techniques for accurate fatigue damage estimation. *Int J Fatigue*. 2016;82:757-765.
22. S. Loew, D. Obradovic, Formulation of fatigue dynamics as hybrid dynamical system for model predictive control, Paper presented at: Accepted for IFAC World Congress 2020, 2020.
23. Shi Y, Xu B, Wang D, Zhang B. Using battery storage for peak shaving and frequency regulation: joint optimization for superlinear gains. *IEEE Trans Power Syst*. 2018;33:2882-2894.
24. PJM, Ancillary service lmps, 2019.
25. PJM, Pjm manual: Energy and ancillary service market operations, 2019.
26. PJM, Pjm regulation market, 2017.
27. B. Xu, Y. Dvorkin, D. S. Kirschen, C. A. Silva-Monroy, J.-P. Watson, A comparison of policies on the participation of storage in u.s. frequency regulation markets, Paper presented at: 2016 IEEE Power and Energy Society General Meeting (PESGM), 2016, pp. 1–5.
28. Houska B, Ferreau HJ, Diehl M. An auto-generated real-time iteration algorithm for nonlinear mpc in the microsecond range. *Automatica*. 2011;47:2279-2285.
29. Fotouhi A, Auger DJ, Propp K, Longo S, Wild M. A review on electric vehicle battery modelling: from lithium-ion toward lithium–sulphur. *Renew Sust Energ Rev*. 2016;56:1008-1021.
30. Wang J, Purewal J, Liu P, et al. Degradation of lithium ion batteries employing graphite negatives and nickel–cobalt–manganese oxide + spinel manganese oxide positives: part 1, aging mechanisms and life estimation. *J Power Sources*. 2014;269:937-948.
31. Birkl CR, Roberts MR, McTurk E, Bruce PG, Howey DA. Degradation diagnostics for lithium ion cells. *J Power Sources*. 2017;341:373-386.

How to cite this article: Loew S, Anand A, Szabo A. Economic model predictive control of Li-ion battery cyclic aging via online rainflow-analysis. *Energy Storage*. 2021;e228. https://doi.org/10.1002/est2.228

APPENDIX

Battery modeling details

Open circuit voltage

The open circuit voltage of the battery (U_{OCV}) is defined as the voltage that can be derived from the battery when it is under no-load. This voltage is assumed to depend only on the SOC. It has been modeled as a linear dependency

$$U_{\mathrm{ocv}}(SOC) = U_{\mathrm{ocv,nominal}} + U_{\mathrm{ocv,slope}}(SOC - SOC_{\mathrm{nominal}}) \quad [V] \tag{A1}$$

with a fixed predefined slope $U_{\mathrm{ocv,slope}}$, centered around a nominal state of charge SOC_{nominal} and with an offset nominal open circuit voltage $U_{\mathrm{ocv,nominal.}}$[15,29]

Internal resistance

The internal resistance of a battery cell is function of *SOC*, cell temperature *T* and State of Health *SOH*. Here, the internal resistance

$$R_{\mathrm{int}}(SOC, T, SOH) = f_R(SOC) f_R(T) f_R(SOH) \quad [\Omega] \tag{A2}$$

is modeled as a product of the individual dependency functions. The dependency on temperature is modeled using a decreasing exponential function

$$f_R(T) = A_r \exp(-B_r T) + C_r \quad [\Omega] \tag{A3}$$

with the constants A_r, B_r, and $C_{r.}$[15,29] The dependency on SOC is modeled using a piece-wise linear function

$$f_R(SOC) = \text{piecewise linear} \quad [-] \tag{A4}$$

based on available battery data.[30] Battery degradation leads not only to reduction in capacity but also to additional increase in the internal resistance.[31] The *in*crease in internal resistance is observed to be approximately linearly proportional to the *de*crease in battery capacity.[1] Consequently, the dependency of internal resistance on State of Health is modeled as

$$f_R(SOH) = 1 + \beta_{\mathrm{degradation}}(1 - SOH) \quad [-] \tag{A5}$$

with the proportionality constant $\beta_{\mathrm{degradation}}$ which is obtained from experiments.

Battery current and terminal voltage

By applying Kirchoff's voltage law in the battery electrical equivalent circuit shown in Figure 2 for a given power *P*, battery current *I* can be obtained by the solution

$$I = \frac{U_{\mathrm{ocv}} - \sqrt{{U_{\mathrm{ocv}}}^2\text{-}4R_{\mathrm{int}}P}}{2R_{\mathrm{int}}} \quad [A]. \tag{A6}$$

of the underlying quadratic equation. The terminal voltage $U_{terminal}$ can be subsequently obtained by

$$U_{\mathrm{terminal}} = U_{\mathrm{ocv}} - IR_{\mathrm{int}} \quad [V]. \tag{A7}$$

Verification and validation of the battery plant model

Exemplary simulation output

In order to qualitatively verify the behavior of the submodels, the battery plant model is simulated in open-loop and subject to power steps and ramps. The sample time of this time-discrete model here is set to 1 minute, and the total simulation time to 12 hours. The results of the simulation are shown in Figure A1.

Each of the output plots can be divided into six sections (each of duration 2 hours) based on the power input type:

In the first section, a discharging power is applied for 2 hours. As a result, the battery SOC decreases and the temperature increases due to resistive heating.

In the second section, an equal magnitude of charging power is applied to the battery for 2 hours which brings the SOC back to the initial value.

In the third section, the battery discharges at a higher power than that of the first section. Therefore, the rates of SOC decrease and temperature increase are higher than that in first section.

In the fourth section, a zero power allows the battery to cool down. Hence, the temperature asymptotically converges toward the ambient temperature. As expected, there is no change in SOC during this time.

In the fifth section, a charging power ramp is input to the battery which shows a gradual increase in SOC, and a smaller initial rate of temperature increase compared with that of section three, where a power step is applied.

Throughout the simulation, the charge capacity of the battery and therefore the State of Health is decreasing. This decrease can be understood better by looking at the individual behaviors of calendric and cyclic capacity loss:

Calendric aging, The increase in calendric capacity loss follows a square root behavior with respect to simulation time. According to (3), the slope of this square root function is scaled depending on the temperature and SOC levels.

Cyclic aging, During section one, the battery only discharges, and thus the Rainflow algorithm only

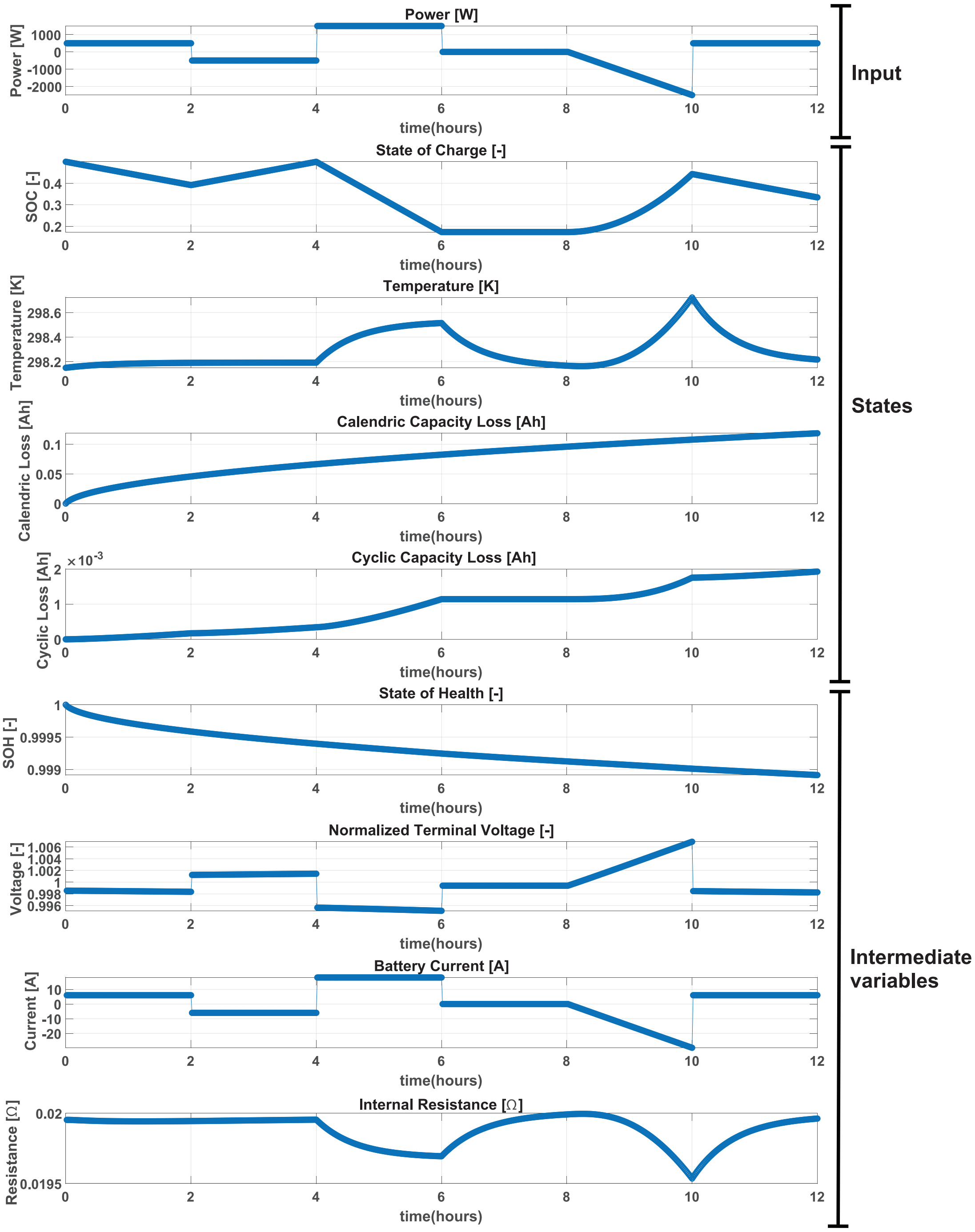


FIGURE A1 Open-loop simulation results for the battery storage system model. The signals are grouped into input, states, and intermediate variables

calculates one half cycle with linearly increasing cycle depth, and over-proportionally increasing cycle damage. As the SOC decreases stronger in section three compared to section one, the corresponding increase in cyclic capacity loss is also higher. For a constant SOC trajectory (as in section four), the cyclic capacity loss as well is constant because the battery does not exhibit any SOC cycle.

Electrical submodel, During the entire simulation, the battery current follows the behavior of input power. The terminal voltage (A7) mainly is governed by the current, and secondarily by the SOC via the Open Circuit Voltage. Here, it is normalized by the nominal Open Circuit Voltage $U_{\text{ocv,nominal}}$. The cell internal resistance (A2) has complex dependencies on temperature, SOC, and state of health as can be seen in the lowest subplot.

Calendric aging

The calendric aging submodel is validated by simulating the battery plant at defined combinations of SOC and ambient temperature. For each combination, the simulation duration is the corresponding expected lifetime. Thus, at the end of each simulation the end-of-life amount of capacity loss should be reached. These simulations are done at zero power such that there is no cyclic aging and thus the resultant total capacity loss is only due to calendric degradation. Furthermore, the SOC and temperature are fixed during the entire simulation.

The results for variations of SOC are summarized in Table A1, and for variations of ambient temperature in Table A2. All results show that exactly the expected capacity loss is reached.

As discussed in section 2, the rate of calendric capacity loss is scaled depending on the instantaneous temperature and SOC. Higher temperature or SOC increase the aging rate. The simulation output in Figure A2 verifies this behavior.

Cyclic aging

The cyclic aging submodel is validated by simulating the battery plant at defined alternating power magnitudes such that the SOC is cycling at fixed DODs. In order to obtain a fixed DOD for every cycle throughout the

TABLE A1 Validation of calendric aging at an ambient temperature of 21°C

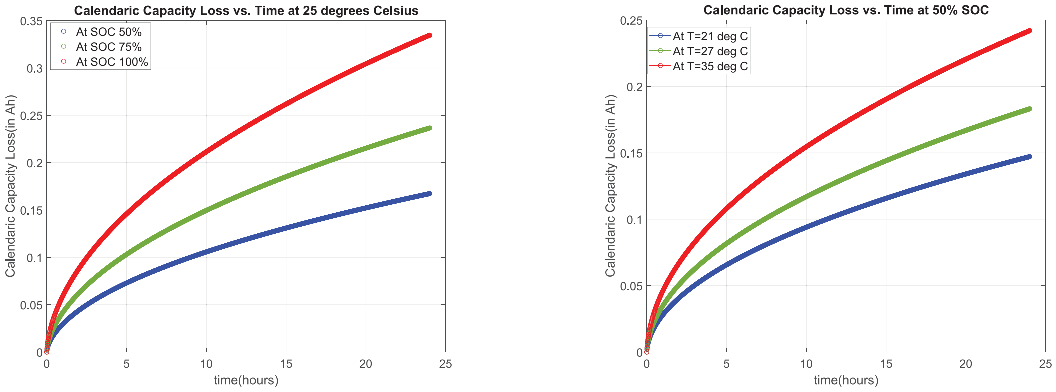
SOC	Lifetime	Lifetime Loss (Exp.)	Lifetime Loss (Model)	Error
1	4 years	20 Ah	20 Ah	0%
0.75	7.5 years	20 Ah	20 Ah	0%
0.5	10 years	20 Ah	20 Ah	0%
0.1	15 years	20 Ah	20 Ah	0%

Note: The errors are rounded to integer values.

TABLE A2 Validation of calendric aging at a state of charge of 0.5

Temperature	Lifetime	Lifetime loss (exp.)	Lifetime loss (model)	Error
21°C	12 years	20 Ah	20 Ah	0%
41°C	4 years	20 Ah	20 Ah	0%

Note: The errors are rounded to integer values.



(A) Calendric capacity loss at different SOCs for a given temperature

(B) Calendric capacity loss at different temperatures for a given SOC

FIGURE A2 Validation of calendric capacity loss at different SOCs and temperature for zero power. A, Calendric capacity loss at different SOCs for a given temperature. B, Calendric capacity loss at different temperatures for a given SOC

TABLE A3 Results of cyclic degradation validation

Cycle depth	Lifetime (exp.)	Lifetime loss (exp.)	Lifetime loss (model)	Error
1	2300 full-cycles	40 Ah	40 Ah	0%
0.8	4500 full-cycles	40 Ah	40 Ah	0%
0.5	7700 full-cycles	40 Ah	40 Ah	0%
0.2	28 000 full-cycles	40 Ah	40 Ah	0%

Note: The errors are rounded to integer values.

simulation, the dynamics of calendric aging and temperature are deactivated. For a specific DOD, the simulation is performed until the expected number of cycles to end-of-life is reached. Thus, at the end of each simulation the end-of-life amount of capacity loss should be reached. The results are summarized in Table A3. For all simulations, the expected capacity loss is obtained up to machine precision.

4.4 Paper 9: Hybrid energy system tracking a scheduled power

4.4.1 Summary

In this publication, a grid-connected hybrid energy system comprising one wind turbine (5 MW rated power) and one battery (1 MW rated power, 1 MWh energy capacity) is controlled by a single Economic Nonlinear Model Predictive Controller (ENMPC). This novel ENMPC simultaneously tries to maximize energy production of the wind turbine, and to minimize wind turbine fatigue and battery cyclic aging, while a net power demand for the total hybrid energy system is met.

In contrary, in the literature only a combination of static economic optimization with a dynamic Tracking MPC has been observed. Furthermore, those control schemes typically do not consider wind turbine fatigue, and only employ heavy approximations of battery cyclic aging.

The basis for the novel ENMPC is a combination of the PORFC MPCs for wind turbines and batteries from **Papers 6** and **8**, respectively. The novel ENMPC exhibits the following extra features:

- Simultaneous consideration of wind turbine fatigue and battery aging;
- Thus, there are two cost functions of PORFC with residue;
- Additional soft equality constraint between net power generation and grid demand.

A comparison to a sole maximization of wind turbine energy production shows that the PORFC MPC leads to higher total profit (revenue minus fatigue and aging cost) for the present application. This is achieved by a significant reduction of wind turbine fatigue and a slight reduction of battery aging, while retaining the net power mismatch at a very low level and accepting a slight reduction of wind turbine revenue.

A comparison to a PORFC MPC without residues for past mechanical stress and SOC shows that this "blinded" controller underestimates fatigue and aging, and thus focuses on the maximization of wind turbine revenue. In total, this leads to lower profit at similar power mismatch levels.

4.4.2 Contribution

The author of this dissertation assisted in conceptualizing the research work and in analyzing the simulation results. AA led the research work, developed the formulations, performed the implementation, executed and analyzed the simulations, and wrote the paper. CLB supervised the work. All authors provided important input to this research work through discussions, feedback and by improving the manuscript.

4.4.3 Reference

A. Anand, S. Loew, and C. L. Bottasso, “Economic control of hybrid energy systems composed of wind turbine and battery,” in *2021 European Control Conference (ECC)*, 2021, pp. 2565–2572. doi:10.23919/ECC54610.2021.9654911 © 2021 IEEE. Reprinted with permission.

2021 European Control Conference (ECC)
June 29 - July 2, 2021. Rotterdam, Netherlands

Economic control of hybrid energy systems composed of wind turbine and battery

Abhinav Anand[1,*], Stefan Loew[1], and Carlo L. Bottasso[1]

***Abstract*— An Economic Nonlinear Model Predictive Controller (ENMPC) is designed for a wind turbine and battery based hybrid energy system. An explicit consideration of cyclic damages within the controller is implemented via externalization of Rainflow based cycle counting (RFC) algorithm from the Model Predictive Controller (MPC). This is achieved using Parametric Online Rainflow counting (PORFC) approach. Additionally, impact of stress history is considered directly inside the optimization problem by employing a stress residue which also helps overcome the limitation of using shorter horizon for cyclic damage estimation. The designed MPC controller is implemented using the state-of-the-art *ACADOS* framework. The performance of the controller is assessed in closed loop with a hybrid plant model consisting of a NREL 5MW onshore wind turbine and a 1MWh/1MW Li-ion battery. Simulation output indicates that the formulated controller results in profit gain with respect to a realistic base-case controller. Moreover, the formulated controller is found to conveniently handle model complexities, non-linearities, and system constraints resulting in suitable dynamic performance. An economically optimal closed-loop operation of the grid-connected hybrid plant is achieved, where the controller, using PORFC algorithm, optimizes a realistic monetary objective while explicitly considering the requirements from the electricity grid.**

I. INTRODUCTION

Large scale grid integration of wind power is a continuously evolving research field. One emerging approach involves hybridizing the wind energy systems with other systems such as batteries [1], [2], [3]. Operating such a hybrid plant in closed loop with an economically optimal controller can allow profit optimal hybrid plant operation while abiding by the grid requirements at the same time. Moreover, such formulation can enable participation of wind based generation units in modern grid ancillary services, generating additional revenue streams for the plant owner [4].

Cyclic damage corresponds to the loss in the capacity or lifetime of a material due to cyclic application of the stress factors [5], [6]. For wind turbines (WT), the stress factors are the material stress cycles leading to structural fatigue damage of various turbine components such as tower or blades. For batteries, the stress factors are the charging-discharging cycles. These damages result in significant economic cost to the plant owner.

The standard economic control formulations for wind based hybrid energy systems do not perform dynamic economic optimization [7], [8]. A two-level approach is rather utilized which only includes tracking pre-computed economically optimal steady state set-points via a standard model predictive controller (MPC) [9]. Also, these formulations are subjected to simplified hybrid plant model [8], [10], [11], where the wind turbine cyclic fatigue minimization is never considered. In some cases, the battery cyclic damage minimization is considered, however, the damage evaluation approaches are heavily approximated [11], [12], [13], [14], [15].

In general, explicit and accurate consideration of cyclic damages in the optimization objective is seldom witnessed [16], [17], [18]. This is possibly because the standard cyclic fatigue evaluation approaches, such as Rainflow algorithm, do not have an analytical formulation and contain algorithmic branches and loops resulting in a discontinuous output-behaviour [19], [20]. This does not allow calculation of sensitivities required for standard gradient-based optimization techniques. Furthermore, the total number of identified cycles N_c for a given stress trajectory σ is not known before the MPC execution. Consequently, cyclic damage minimization objective is approximated or even not considered [15], [16]. Therefore, until recently, integration of the cyclic damage evaluation using RFC into gradient-based MPC has not been possible.

In the present work, an economic nonlinear model predictive controller (ENMPC) is formulated for a grid-connected wind turbine and battery based hybrid energy system. To the best of our knowledge, a direct economic optimization by simultaneous consideration of wind turbine and battery damage against energy generation in an MPC framework has not been previously reported. The current work utilizes the concept of Parametric Online Rainflow Counting (PORFC) for cyclic damage minimization, proposed first in our previous contribution [21] and illustrated more in details in [20] and [19]. Unlike standard nonlinear MPC where the optimal control variables are generated with the objective of tracking pre-specified set-points, an ENMPC generates optimal control variables by directly minimizing a realistic/high-level monetary cost function. The formulated controller aims to maximize the overall profit of the plant by directly balancing between revenue and cost objectives enabled by an accurate and explicit handling of cyclic damage, while providing the power demanded by the electricity grid.

In comparison to our previous contributions, the present work proposes and utilizes a more realistic and complex optimization objective including the interplay of two cyclic damage cost functions. Also, the inclusion of the hard equality constraint between net power generation and demand over

[1] Technical University of Munich, Wind Energy Institute, Boltzmannstrasse 15, 85748 Garching b. Muenchen, Munich, Germany

*All correspondence to: abhinav.anand@tum.de

978-94-6384-236-5 ©2021 EUCA

the MPC prediction horizon renders the overall online optimal control problem difficult. In addition to these, the closed loop performance of the formulated controller is assessed in conjunction with a comparatively more complex plant. Moreover, there are small underlying modelling differences between the controller internal model and the employed plant model such as in the nonlinear battery damage function, challenging the robustness of the controller.

This paper is divided into five sections. We begin setting the context in Sec. II by providing some fundamentals about cyclic damage evaluation and background about the corresponding control problem. Section III focuses on controller design: first a description of using PORFC in MPC and second formulating the optimal control problem. Section IV shows a comparison of the formulated economic controller with a realistic baseline power-maximizing controller. Furthermore, the benefit of accurate damage estimation considering stress history inside the MPC controller is illustrated via simulation output. Section V concludes this work and presents future goals.

II. Background & Fundamentals

A. General Plant model and Control problem

The present work is applicable to a plant model represented as a continuous non-linear system

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)), \tag{1}$$

where $\mathbf{x}(t)$, $\mathbf{u}(t)$, and $\mathbf{d}(t)$ represent continuous system states, control variables, and external disturbance respectively. Function $\mathbf{F}(.)$ represents the mapping of ordinary differential equations governing the continuous dynamics of the system. The stress factor $\sigma(t)$ for cyclic damage could be an output or a state of this model.

The standard Nonlinear Model Predictive Control (NMPC) problem is solved by formulating a Nonlinear Programming (NLP) problem over the prediction horizon $T_{horizon}$. The continuous nature of NLP is approximated by discretization of the time continuous control variables $u_i(t)$ into piecewise constant optimization variables $\bar{u}_{i,j}$. Here, i denotes the individual control input to the plant system $i \in N_{in}$, determined by the total number of control inputs N_{in}, and j denotes the respective control interval within the prediction horizon after discretization $j \in N_u$, determined by the total number of control intervals N_u. The duration of one control interval can thus be denoted as $T_{ctrl} = \frac{T_{horizon}}{N_u}$.

B. Cyclic Damage evaluation

1) Conventional Approach: The conventional cyclic damage evaluation approach utilizes cycle counting algorithms, where damage cycles and their characteristics due to a particular stress trajectory are identified. Rainflow based cycle-counting (RFC) algorithm is widely accepted for its accuracy and involves identifying points of reversal (extrema) in the trajectory of the stress factor $\sigma(t)$ discretized into $\sigma(k)$, resulting in total cyclic damage $D(k)$ [22]. Fig. 1 shows an exemplary stress trajectory and the corresponding points of reversal leading to damage cycles. The identified cycles are classified either as full-cycle or half-cycle, depending on whether the initial and final stress magnitude are the same or different.

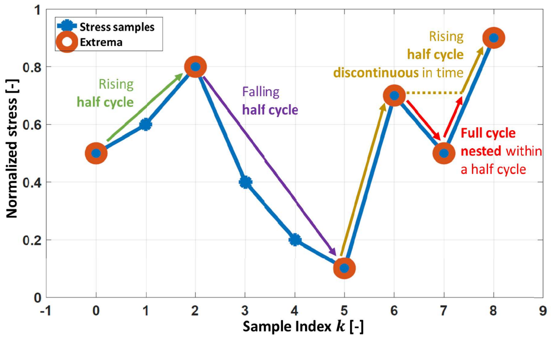


Fig. 1: Exemplary stress profile and identified cycles using RFC algorithm

Table I summarizes the outputs of the RFC algorithm. Here, cycle weight $\sigma_{w,c}^{\mathrm{RFC}}$ is either 1 or 0.5 depending on whether the identified cycle c is a full-cycle (closed cycle) or a half-cycle (open cycle). The units of identified cycle range and cycle mean are equal to those of the corresponding stress trajectory σ. The remaining three outputs $\sigma_{w,c}^{\mathrm{RFC}}$, $k_{max,c}^{\mathrm{RFC}}$, and $k_{min,c}^{\mathrm{RFC}}$ are unit-less. The cyclic damage D is a nonlinear function of the Rainflow outputs.

TABLE I: Output of Rainflow algorithm in MATLAB for each cycle c [23]

Quantity	Variable
Cycle weight	$\sigma_{w,c}^{\mathrm{RFC}}$
Cycle range	$\sigma_{r,c}^{\mathrm{RFC}}$
Cycle mean	$\sigma_{m,c}^{\mathrm{RFC}}$
Index of sample resulting in cycle maximum	$k_{max,c}^{\mathrm{RFC}}$
Index of sample resulting in cycle minimum	$k_{min,c}^{\mathrm{RFC}}$

2) Residue: The stress sample values which are extrema, but have not contributed to a full cycle so far, form half cycles. These stress values can later form a new half or full cycle with the newly acquired stress samples of subsequent simulation time-steps. This new cycle will have a higher cycle range, and thus higher damage, than the original half cycle [24]. Thus, it is important to preserve these half-cycle extrema and merge them with the new stress samples before performing the next cycle identification. Thus, accurate cyclic damage evaluation is a time-cumulative process. This set of extrema is referred in present work as "residue" $\sigma_{residue}$ and plays a central role in accurate identification of cyclic damage [24]. The conventional cyclic damage evaluation extracts the damage due to residue samples at current simulation step and discards the samples for next simulation step [24]. This leads to under-evaluation of cyclic damage [19].

3) One-step time discrete approach: Based on the Rainflow algorithm and the concept of residue, a one-step time-discrete model for cyclic damage evaluation has been proposed in [21]. This means that by introducing residue as an additional "memory" state, with each new stress sample an update of cyclic damage can be computed. The procedure is presented in Alg. 1.

Algorithm 1 One-step time discrete cyclic aging evaluation [21]

```
Input: Existing stress factor string $\boldsymbol{\sigma}_{\text{exist}}$, Periodic update of a scalar new stress sample $\sigma(k+1)$ at each step $k$
Output: Periodic update of cyclic damage $D_{\text{cyc}}^{\text{Step}}$ at each step $k$
Initialization: Zero total cyclic damage $D_{\text{cyc}}^{\text{Step}} = 0$ and cyclic damage $D_{\text{cyc,FC}} = 0$ of full cycles, stress residue $\boldsymbol{\sigma}_{\text{residue}} = \boldsymbol{\sigma}_{\text{exist}}$
while true do
    1) Merge residue $\boldsymbol{\sigma}_{\text{residue}}(k)$ and new stress sample $\sigma(k+1)$
    2) Extract full and half cycles from $\{\boldsymbol{\sigma}_{\text{residue}}(k), \sigma(k+1)\}$ using Rainflow algorithm, Store residue in $\boldsymbol{\sigma}_{\text{residue}}(k+1)$
    3) Calculate cyclic damage based on full cycles, Add result to $D_{\text{cyc,FC}}(k)$ to obtain $D_{\text{cyc,FC}}(k+1)$
    4) Calculate cyclic damage based on full and half cycles, Add result to $D_{\text{cyc,FC}}(k)$ to obtain $D_{\text{cyc}}^{\text{Step}}(k+1)$
end
```

The time-discrete cyclic damage estimation is expressed by the update of the residue set

$$\boldsymbol{\sigma}_{\text{residue}}(k+1) = G_{\text{cyc,residue}}\left(\boldsymbol{\sigma}_{\text{residue}}(k), \sigma(k+1)\right), \tag{2}$$

by the update of cyclic capacity loss of full cycles

$$D_{\text{cyc,FC}}(k+1) = G_{\text{cyc,FC}}\left(\boldsymbol{\sigma}_{\text{residue}}(k), \sigma(k+1), D_{\text{cyc,FC}}(k)\right) \tag{3}$$

and by the update of total cyclic damage

$$D_{\text{cyc}}^{\text{Step}}(k+1) = G_{\text{cyc,tot}}\left(\boldsymbol{\sigma}_{\text{residue}}(k), \sigma(k+1), D_{\text{cyc,FC}}(k)\right). \tag{4}$$

Thus, the time-discrete equations $G_{\text{cyc},\cdot}$ represent the algorithmic steps as described in Alg. 1.

C. Modified plant model

Since the cyclic damage evaluation can only be performed in time-discrete fashion, the present work is hence applicable to a modified plant model where the continuous system dynamics shown in (1) in Sec. II-A are expressed in time-discrete fashion as

$$\mathbf{x}(k+1) = \mathbf{G}_{int}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{d}(k)). \tag{5}$$

This is obtained by the application of integrator $\mathbf{G}_{int}$ to the system of ODEs shown in Eq. 1.

III. Economic Controller formulation for Hybrid system using Parametric Online RFC

A. PORFC concept[1]

The challenges for a direct implementation of RFC in MPC (refer to Sec. I) are overcome by the method of PORFC [21], where all discontinuous parts of the cyclic damage estimation procedure are performed in a pre-processing step before each execution of the MPC algorithm, as shown in Fig. 2. In order to base this externalized cyclic damage estimation on the same stress trajectory like the MPC, the pre-processing step has to start with a predictive forward simulation using the same model, sampling and horizon length as the internal simulation of the MPC. In addition to this prediction, the stress history is incorporated in the pre-processing via a periodically updated residue $\boldsymbol{\sigma}_{\text{residue}}$ (see Sec. II-B.2).

The workflow within the PORFC approach can be explained as:

- **Simulation:** The plant internal model is simulated over the prediction horizon using the currently measured states $\widetilde{\mathbf{x}}$ as initial states $\mathbf{x}_0$, and the current guess of the optimal control trajectory $\mathbf{u}^{guess}$. Relevant result is the stress prediction over the entire horizon.
- **Merge:** The residue is merged with the stress prediction.
- **Rainflow:** The Rainflow algorithm is used to identify stress cycles over this merged trajectory. Consequently, it is assumed that the structure of identified cycles does not change within the upcoming optimization run. Here, the term "structure" denotes positions ($k_{max,c}^{\text{RFC}}$, $k_{min,c}^{\text{RFC}}$) and weights ($\sigma_{w,c}^{\text{RFC}}$) of cycles. This assumption implies that only the values and not the positions of the controllable extrema can be shifted in the prediction horizon by the optimization.
- **Residue update:** Stress cycles can be composed by stress samples only from residue or prediction, or by a mixture of both. However, only controllable samples within the prediction horizon can be altered by the optimization. If a full cycle is detected entirely within the *residue*, both contributing values are discarded from the residue. The reason for this is that also in the future they will never anymore form a cycle with a sample from the prediction and, therefore, are irrelevant for the MPC.
- **Time-varying parameters:** Information from cycle identification is used to fill vectors of time-varying parameters $\mathbf{p}^{parameters}$, which are forwarded to the objective function of the MPC. Since, any particular half or full cycle always corresponds to exactly two "complementary" stress samples $\sigma(k_{min,c}^{\text{RFC}})$ and $\sigma(k_{max,c}^{\text{RFC}})$, it is assumed that the damage of a cycle c can be evenly split over the corresponding two complementary sample instances $k_{max,c}^{\text{RFC}}$ and $k_{min,c}^{\text{RFC}}$.
  Although a stress sample is not allocated uniquely to one identified cycle, it can at maximum be part of two cycles [25], [15]. Thus, in a general case, it can be assumed that a particular stress sample may belong to either none (when it is not an extrema), or one (when it is part of only one half or full cycle), or two cycles (when it sits at a cycle junction point). Since each stress

[1] The content of Sec. III-A explains briefly the detailed formulation presented in [19]

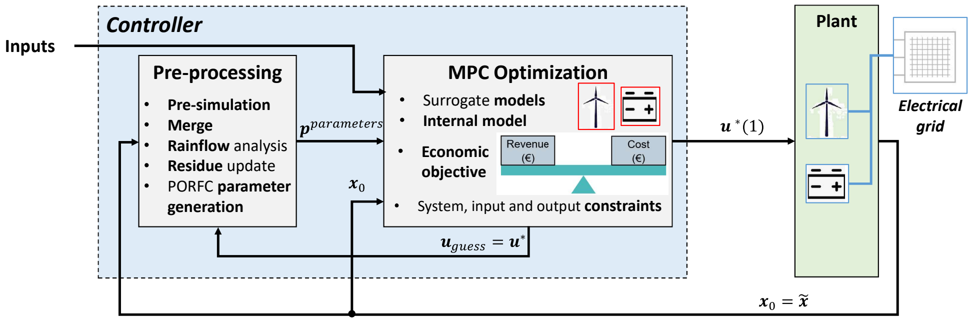


Fig. 2: Process-flow using the proposed methodology

extremum belongs to either one or two stress cycles [25], [15], the Rainflow algorithm provides one or two mean stress values per extremum. These mean stress values are considered as optimization- or tracking-goals for the current MPC-step.

Cycle mean values $\sigma^{\text{PORFC}}_{m,c1/2}$ and cycle weights $\sigma^{\text{PORFC}}_{w,c1/2}$ are collected in the parameter vector

$$\mathbf{p}^{PORFC} = \left(\sigma^{\text{PORFC}}_{\text{m,c1}}, \sigma^{\text{PORFC}}_{\text{m,c2}}, \sigma^{\text{PORFC}}_{\text{w,c1}}, \sigma^{\text{PORFC}}_{\text{w,c2}}\right) \quad (6)$$

which is defined piece-wise constant over the control intervals of the prediction horizon.

A more detailed derivation and explanation can be found in [19].

In the cost function of the MPC, the parameters are used to time-continuously calculate the cyclic damage over the horizon and accumulate it via integration. Consequently, the cyclic damage term of PORFC is defined by a time-integral over two cost terms, where each represents one potential cycle-contribution of a stress sample [21]. In a time-differential form, the rate of cyclic damage is expressed by:

$$\dot{D}^{PORFC}_{cyc}(t) = \frac{\mathrm{d}\Delta D^{PORFC}_{cyc}\left(\sigma(t), \mathbf{p}^{PORFC}(t)\right)}{\mathrm{d}t} =$$

$$\frac{1}{T_{ctrl}} \sum_{c=1}^{2} D^{\text{PORFC}}_{cyc,c}\left(\sigma(t), \sigma^{\text{PORFC}}_{m,c}(t), \sigma^{\text{PORFC}}_{w,c}(t)\right) \quad . \quad (7)$$

This expression can thus be used to formulate cyclic damage minimization objectives for the optimal control problem.

B. Optimal control problem

The proposed methodology, as shown in an implemented process flow in Fig. 2, describes an MPC controller operating in closed loop with the hybrid plant. Such a controller utilizes an internal model of the plant to calculate economically optimal control variables by optimizing a chosen realistic and operationally meaningful economic objective (maximizing profit by balancing between cost and revenue). Measurement of the plant states $\mathbf{x}_0$ and controller execution happen at a rather coarse sample time T_{ctrl}. Within the pre-processing step of the controller, the forward simulation is performed with a fixed step size of $T_{\text{sim,preproc}} < T_{\text{ctrl}}$. The PORFC parameters are sampled on the coarse control grid T_{ctrl} since parameters typically only can be inserted into the MPC problem on this granularity for current MPC frameworks. For the MPC-step of the controller, the inputs, which might contain external parameters such as references and weights, are inserted as parameters on the control grid T_{ctrl}. Within the MPC-step of the controller, multiple-shooting is used where the shooting nodes are defined on the control grid T_{ctrl}. Within each shooting interval, the plant model is simulated on the fine grid $T_{\text{sim,MPC}} = T_{\text{sim,preproc}}$. The optimized control variables and shooting states are output from the controller again at a sample time of T_{ctrl}. The first entries of each control variable $\mathbf{u}^*(1)$ are applied to the plant.

1) Hybrid system model:

a) Wind turbine model: A reduced order model of NREL 5 MW wind turbine has been considered in this work [6]. This model serves as the plant model as well as the internal controller model. This section summarizes the modelling approach which has been presented in detail in [26].

The turbine is assumed to receive perfect foresight of the incoming rotor equivalent wind velocity from a Light Detection and Ranging (LiDAR) sensor. The incident wind $V_w(t)$ induces aerodynamic torque T_Q in the rotor shaft and thrust force F_T on the rotor. The aerodynamic torque directly couples with the drive-train dynamics

$$J_r\dot{\omega} = T_Q - T_g, \quad (8)$$

where J_r, ω, and T_g represent rotor moment of inertia, rotor angular velocity, and generator torque respectively. The thrust force F_T excites fore-aft oscillations in the tower quantified using tower tip deflection d_T having dynamics

$$\ddot{d}_T = \frac{1}{a_1}(F_T - a_2\dot{d}_T - a_3 d_T) \quad (9)$$

resulting in a severe cyclic stress $\sigma(t)$ at the tower root causing cyclic damage. The turbine model has two control variables: rate of change of generator torque $\dot{T}_g$ and blade pitch actuator set-point β_c with blade dynamics

$$\ddot{\beta}_b = -a_4\dot{\beta}_b - a_5(\beta_b - \beta_c). \quad (10)$$

Here, β_b represents the collective blade pitch angle and the parameters a_1 through a_5 are fixed model parameters representing turbine properties [6]. Non-linearity in the model originates from $T_Q(\omega,\beta_b,(V_w-\dot{d}_T))$ and $F_T(\omega,\beta_b,(V_w-\dot{d}_T))$, and are calculated at stationary operating points using high-fidelity $C_p-Lambda$ simulator [26].

b) Battery model: An electrical equivalent circuit model of a 1MW/1MWh Li-ion battery has been considered in this work. This section summarizes the modelling approach which is presented in detail in [19]. The considered model serves as the plant model and with some approximations as an internal controller model.

The battery model consists of three sub-models: electrical, thermal, and degradation. For a given control variable P_B, denoting battery power, the electrical sub-model captures the battery current $I(t)$ dynamics and state of charge $SOC(t)$ dynamics

$$S\dot{O}C=-\frac{I}{Q}. \tag{11}$$

Here, Q represents the maximum charge capacity of the battery at a given time. Q always decreases over time and usage because of the permanent loss in the capacity of the battery over time (calendric damage Q_{cal}), calculated using

$$\dot{Q}_{cal}=f_1(SOC)f_2(T)\sqrt{t}, \tag{12}$$

and over charging and discharging operations (cyclic damage Q_{cyc}), calculated using one step time-discrete cyclic ageing evaluation approach shown in Alg. 1 and in (4). The functions $f_1(SOC)$ and $f_2(T)$ denote the dependency of state of charge SOC and battery temperature T on the calendric capacity loss. The battery thermal model captures the evolution of battery temperature $T(t)$ dynamics

$$\dot{T}=\frac{1}{C_H}(I^2R_{int}-C_R(T-T_{ambient})) \tag{13}$$

based on a lumped heat capacitance model [27]; where C_R and C_H denote cooling rate and heat capacity respectively. Here, $T_{ambient}$ denote ambient temperature considered to be fixed at 298.15 K.

The wind turbine and battery based hybrid plant model having system dynamics, as shown in (1) and (5), consists of ten system states $\mathbf{x}=(\omega,d_T,\dot{d}_T,\beta_b,\dot{\beta}_b,T_g,T,SOC,Q_{cal},Q_{cyc})$ and three control input variables $\mathbf{u}=(\beta_c,\dot{T}_g,P_B)$.

2) Optimization problem: The formulated economic optimal control problem

$$\min_{\mathbf{u},\mathbf{s}}-(J^{WT}_{generation})^2+(J^{WT}_{towerfatigue})^2+(J^{B}_{cyclicloss})^2 \\ +\int_{t_0}^{t_{end}}((10^3s_1^2+10^6s_2^2)dt) \tag{14a}$$

subject to

$$\dot{\mathbf{x}}=h(\mathbf{x},\mathbf{u}) \tag{14b}$$

$$\underline{\mathbf{x}_{aug}}\leq\mathbf{x}_{aug}\leq\overline{\mathbf{x}_{aug}} \tag{14c}$$

$$\underline{\mathbf{u}}\leq\mathbf{u}\leq\bar{\mathbf{u}} \tag{14d}$$

$$\underline{\mathbf{s}}\leq\mathbf{s}\leq\bar{\mathbf{s}} \tag{14e}$$

$$\omega T_g+P_B+s_2=P^{grid}_{demand} \tag{14f}$$

is solved in an online MPC fashion for the hybrid system model described in III-B.1. Here the interval $t\in[t_0,t_{end}]$ denotes the prediction horizon $T_{horizon}$ of MPC. The optimization variables are the control variables $\mathbf{u}=(\beta_b,\dot{T}_g,P_B)$ and the slack variables $\mathbf{s}=(s_1,s_2)$.

The purpose of introducing slack variables is to achieve recursive feasibility of the MPC optimization problem in presence of model uncertainties and system perturbations [28]. In the present formulation, the state variable ω and the wind turbine electrical power output ωT_g are augmented using the bounded slack variables s_1 and s_2 respectively, as suggested in [28]. The augmented system states $\mathbf{x}_{aug}$ can be represented as $\mathbf{x}_{aug}=(\omega+s_1,d_T,\dot{d}_T,\beta_b,\dot{\beta}_b,T_g,SOC)$.

The optimization objective includes

1) **Maximizing wind turbine generation** by maximizing the aerodynamic power capture

$$J^{WT}_{generation}=w_P\int_{t_0}^{t_{end}}(\omega(t)T_Q(\omega,\beta_b,(V_w-d\dot{T})))dt \tag{15}$$

at the given input condition and revenue rate of providing electricity w_P. It should be noted that instead of the electrical power ωT_g, the aerodynamic power ωT_Q is maximized in order to avoid the greedy extraction of rotor kinetic energy by MPC (*turnpike effect*), as suggested by [28].

2) **Minimizing tower cyclic damage** due to stress at its root σ. A continuous estimation of fatigue cost rate due to a particular stress cycle $J^{WT}_{towerfatigue}$ is given as (refer Eq. 7 and for details check [20])

$$\begin{aligned} J^{WT}_{towerfatigue}&=\int_{t_0}^{t_{end}}(J^{PORFC}_{cyc,\sigma})dt,\\ J^{PORFC}_{cyc,\sigma}&=\frac{1}{T_{ctrl}}\sum_{c=1}^{2}J^{\mathrm{PORFC}}_{cyc,c,\sigma}(\sigma(t),\sigma^{\mathrm{PORFC}}_{m,c}(t),\sigma^{\mathrm{PORFC}}_{w,c}(t)),\\ J^{\mathrm{PORFC}}_{cyc,c1/2,\sigma}&=\sigma^{\mathrm{PORFC}}_{w,c1/2}(t)a_m\sigma^{m}_{eq,c1/2}(t),\\ \sigma_{eq,c1/2}(t)&=|\sigma(t)-\sigma^{\mathrm{PORFC}}_{m,c1/2}(t)|\frac{R_m}{R_m-\sigma^{\mathrm{PORFC}}_{m,c1/2}(t)}. \end{aligned} \tag{16}$$

Here a_m is determined from the initial capital cost of the machine (see also [6]), R_m denotes the ultimate tensile strength of the material, and m represent the positive exponent derived from the material S-N characteristic. In present work, $m=2$.

3) **Minimizing battery cyclic damage** due to charging and discharging operation represented by $SOC(t)$ cycles. A continuous estimation of cyclic loss rate due to a particular for a given SOC cycle $J^{B}_{cyclicloss}$ is given

as (refer Eq. 7 and for details check [19])

$$J^B_{cyclicloss} = w_B \int_{t_0}^{t_{end}} (Q^{PORFC}_{cyc,SOC})dt,$$

$$Q^{PORFC}_{cyc,SOC} =$$

$$\frac{1}{T_{ctrl}} \sum_{c=1}^{2} Q^{\text{PORFC}}_{cyc,c,SOC}\left(SOC(t), SOC^{\text{PORFC}}_{m,c}(t), SOC^{\text{PORFC}}_{w,c}(t)\right),$$

$$Q^{PORFC}_{cyc,c1/2,SOC} =$$

$$\sigma^{\text{PORFC}}_{w,c1/2}(t) A_{cyc} a_w |SOC(t) - SOC^{\text{PORFC}}_{m,c1/2}(t)|^w. \quad (17)$$

Here, A_{cyc}, a_w, and w represent the total cyclic capacity loss till defined battery end of life [19], Woehler constant and Woehler exponent obtained from curve fitting the battery manufacturer data. It should be noted that the plant model of battery has a Woehler constant of $w = 1.4$ whereas the controller internal model uses a Woehler constant of $w = 2$. The weight factor w_B represent the unit replacement cost of the battery [19].

The optimization problem is subjected to

- system dynamics of hybrid plant (More details on the model in [26], [19]): as shown in (14b)
- inequality constraints on augmented states: as shown in (14c)
- box constraints on control and slack variables: as shown in (14d) and (14e)
- power balance as equality constraint: as shown in (14f)

IV. Results

A. ENMPC setup

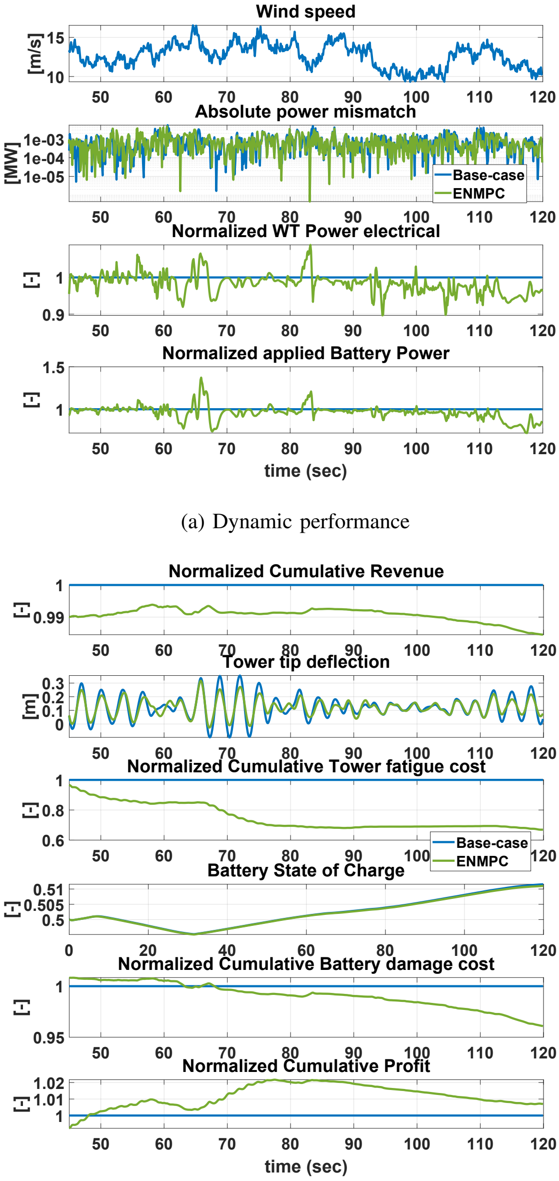
The formulated ENMPC problem is solved via state-of-the-art *ACADOS* framework [29], [30], using the interior-point solver *HPIPM* for the underlying Quadratic Program (QP) in an NLP. A single QP is solved per MPC step except for the first MPC step, where twenty QPs are solved. Multiple shooting approach is used and the Hessian matrix is automatically convexified to address possible numerical issues due to highly non-standard formulation using PORFC [20]. The horizon length $T_{horizon}$ is $2s$ with total 20 control steps such that T_{ctrl} is $100ms$. The plant model is also sampled at $100ms$.

B. Output

1) Comparison to a realistic base-case scenario: The optimal control problem, formulated in (14), is applied to the hybrid model described in Sec. III-B.1. The output is compared w.r.t a base case scenario similar to (14) but with the only difference of the optimization objective

$$\min_{\mathbf{u},\mathbf{s}} -J^{WT}_{generation} + \int_{t_0}^{t_{end}} (10^3 s_1^2 + 10^7 s_2^2)dt \quad (18)$$

which only focuses on maximizing power capture of the wind turbine instead of maximizing the profit (as in (14a)). The net profit is calculated as the difference of revenue from wind power generation and costs due to tower fatigue damage and battery capacity loss.



(a) Dynamic performance

(b) Economic performance

Fig. 3: Simulation output for a realistic power reference ramped up/down periodically around turbine rated power. The green curve shows output of formulated ENMPC controller and the blue curve shows the base-case controller.

Fig. 3 shows the simulation comparison results of the formulated ENMPC (green curve), as described in (14), which performs economically better than the base-case scenario (blue curve), as described in (18). The enhanced economic performance is because the formulated ENMPC controller manages to find an optimal spot for hybrid plant operation (refer to the green curve in subplot "Normalized WT power

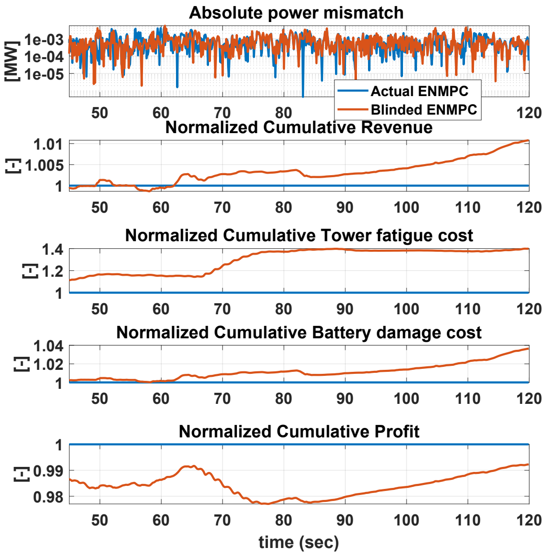


Fig. 4: Simulation results showing impact of residue on economic performance of plant in closed-loop operation. The red curve shows ”blinded” ENMPC controller and the blue curve shows actual ENMPC controller.

electrical” and ”Normalized applied battery power” in Fig. 3a). This optimal operating point balances the wind turbine tower fatigue cost and revenue from power generation at the same time exerting lesser power demand on the battery.

The controller reduces the wind turbine tower root fatigue cost by reducing the tower tip oscillation amplitudes (refer subplots ”Tower tip deflection” and ”Normalized cumulative tower fatigue cost” in Fig. 3b) that lead to damage cycles at the tower root. Moreover, the battery cyclic capacity loss and the corresponding cost of battery damage is reduced by optimally reducing the SOC cycle depths (refer subplots ”Battery State of Charge” and ”Normalized cumulative battery damage cost” in Fig. 3b), which effectively depend on power applied/demanded to/from the battery. As a consequence, the cumulative profit (refer subplot ”Normalized Cumulative Profit” in Fig. 3b) of the formulated economic controller (green curve) is higher than the base case (blue curve). This is because of a comparatively higher reduction in overall costs than reduction in generated revenue (refer subplot ”Normalized Cumulative Revenue” in Fig. 3b) for the hybrid plant.

Although, the equality constraint of power balancing is numerically violated for both the controllers (see subplot ”Absolute power mismatch” in Fig. 3a), the mismatch values are in orders of magnitudes smaller than the reference power.

The average computational time for the controller over the simulation duration of 2 min sampled every 100 ms is found to be 160 ms. This also includes the computational time of 2 ms required for PORFC setup in the pre-processing step (refer Fig. 2). The closed-loop problem was simulated in MATLAB on a desktop with Intel i7 processor having 64-bit operating system and 8 GB RAM. Although, the controller is not yet real-time feasible, further tuning of the numerical setup of the optimal control problem such as reducing the prediction horizon length would improve the controller computational performance.

2) Impact of stress history on economic performance in closed loop: The impact of residue stress samples on economic performance of controller is evident from the simulation output shown in Fig. 4 where the red curve represents the normalized plant performance when the controller is blinded to the history of mechanical stress σ for the wind turbine and State of Charge SOC for the battery. This means that during the pre-processing stage of the controller (refer Sec. III-A), at any given MPC simulation step, the residue stress samples from previous step are not merged with the stress samples from the prediction. This leads to under-evaluation of cyclic damages leading to an economically sub-optimal solution of the optimization problem. The normalization has been done against the output for the formulated optimization problem in (14) shown as blue curve in Fig. 4 and as green curve in Fig. 3. As the costs are under-evaluated, the blinded controller (red curve) sees lesser equivalent damage and thus tries to capture more power from the wind turbine to increase the revenue (see ”Normalized Cumulative revenue” subplot in Fig. 4). This in turn results in comparatively higher cost (see ”Normalized Cumulative tower fatigue” and ”Normalized Cumulative battery damage” subplots in Fig. 4) than the case when the controller is not blinded (blue curve). This comparative analysis clearly shows that accurate consideration of stress history leads to economically better performance without much difference in the absolute power mismatch (see subplot ”Absolute power mismatch” in Fig. 4) between generation and demand.

V. Conclusion

In present work, an ENMPC has been formulated to operate in a closed loop with a wind turbine and battery based hybrid energy system providing power to the electrical grid. The economic controller directly balances between the revenue obtained from power generation and costs due to mechanical fatigue damage of wind turbine tower and cyclic capacity loss of the considered Li-ion battery while respecting the system, input, and grid constraints. The controller utilizes PORFC formulation to explicitly and accurately formulate associated cyclic damages as an optimization objective. This is achieved by performing Rainflow analysis prior to each MPC step, distributing and de-coupling cyclic damage over time, and by memorizing a condensed set of stress samples from past open-cycles.

The initial results for the formulated economic controller shows profit gain against a realistic base-case scenario with suitable dynamic performance. Moreover, the explicit consideration of stress cycles from past in the controller results in enhanced economic performance. This motivates extending the approach towards more complex dynamical systems such as wind farm and solar plants based hybrid systems participating in modern grid ancillary services for obtaining profit optimal dynamic operation. In parallel to this, the controller formulation can be further refined to improve the plant dynamic performance by analysing the impact of prediction horizon length, numerical setup of the QP and NLP solvers, and possibly by formulating a more holistic economic objective function.

References

[1] H. Zhao, Q. Wu, S. Hu, H. Xu, and C. N. Rasmussen, "Review of energy storage system for wind power integration support," *Applied Energy*, vol. 137, pp. 545–553, 2015.

[2] F. Díaz-González, A. Sumper, O. Gomis-Bellmunt, and R. Villafáfila-Robles, "A review of energy storage technologies for wind power applications," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 4, pp. 2154–2171, 2012.

[3] H. Ibrahim, M. Ghandour, M. Dimitrova, A. Ilinca, and J. Perron, "Integration of wind energy into electricity systems: Technical challenges and actual solutions," *Energy Procedia*, vol. 6, pp. 815–824, 2011.

[4] Farbod Jahanbakhsh, Masoud Davoudi, Vahraz Zamani, Nima Yousefpoor, and Douglas Proudfoot, *BESS Performance Assessment for Integration of Wind Generation*. Piscataway, NJ: IEEE, 2017. [Online]. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=8093780

[5] A. Barré, B. Deguilhem, S. Grolleau, M. Gérard, F. Suard, and D. Riu, "A review on lithium-ion battery ageing mechanisms and estimations for automotive applications," *Journal of Power Sources*, vol. 241, pp. 680–689, 2013.

[6] S. Loew, D. Obradovic, and C. L. Bottasso, "Direct online rainflow-counting and indirect fatigue penalization methods for model predictive control," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3371–3376.

[7] H. Borhan, M. A. Rotea, and D. Viassolo, "Optimization-based power management of a wind farm with battery storage," *Wind Energy*, vol. 24, no. 3, 2012.

[8] P. Beltran, *An economic MPC approach for a micro grid energy management system: 10 - 13 Sept. 2014, Medellín, Colombia*. Piscataway, NJ: IEEE, 2014. [Online]. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=6943281

[9] T. Tran, K.-V. Linga, and J. M. Maciejowski, "Economic model predictive control - a review," in *Proceedings of the 31st International Symposium on Automation and Robotics in Construction and Mining (ISARC)*, ser. Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC), Q. Ha, X. Shen, and A. Akbarnezhad, Eds. International Association for Automation and Robotics in Construction (IAARC), 2014.

[10] N. T. Mbungu, R. Naidoo, R. C. Bansal, and M. Bipath, "Optimisation of grid connected hybrid photovoltaic–wind–battery system using model predictive control design," *IET Renewable Power Generation*, vol. 11, no. 14, pp. 1760–1768, 2017.

[11] R. H. L. Rodriguez, I. Vechiu, S. Jupin, S. Bacha, Q. Tabart, and E. Pouresmaeil, "A new energy management strategy for a grid connected wind turbine-battery storage power plant," in *2018 IEEE International Conference on Industrial Technology (ICIT)*, 2018, pp. 873–879.

[12] H. H. Abdeltawab and Y. A.-R. I. Mohamed, "Market-oriented energy management of a hybrid wind-battery energy storage system via model predictive control with constraint optimizer," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 11, pp. 6658–6670, 2015.

[13] L. Johnston, F. Díaz-González, O. Gomis-Bellmunt, C. Corchero-García, and M. Cruz-Zambrano, "Methodology for the economic optimisation of energy storage systems for frequency support in wind power plants," *Applied Energy*, vol. 137, pp. 660–669, 2015.

[14] M. Koller, T. Borsche, A. Ulbig, and G. Andersson, "Defining a degradation cost function for optimal control of a battery energy storage system," in *2013 IEEE Grenoble Conference*, 2013, pp. 1–6.

[15] Y. Shi, B. Xu, Y. Tan, D. Kirschen, and B. Zhang, "Optimal battery control under cycle aging mechanisms in pay for performance settings," *IEEE Transactions on Automatic Control*, vol. 64, no. 6, pp. 2324–2339, 2019.

[16] C. Wang, Z. Du, Y. Ni, C. Li, and G. Zhang, "Coordinated predictive control for wind farm with bess considering power dispatching and equipment ageing," *IET Generation, Transmission & Distribution*, vol. 12, no. 10, pp. 2406–2414, 2018.

[17] H. E. Sanchez, T. Escobet, V. Puig, and P. F. Odgaard, "Health-aware model predictive control of wind turbines using fatigue prognosis," *International Journal of Adaptive Control and Signal Processing*, vol. 32, no. 4, pp. 614–627, 2018.

[18] J. Luna, O. Falkenberg, S. Gros, and A. Schild, "Wind turbine fatigue reduction based on economic-tracking nmpc with direct ann fatigue estimation," *Renewable Energy*, vol. 147, pp. 1632–1641, 2020.

[19] S. Loew, A. Anand, and A. Szabo, "Economic model predictive control of li–ion battery cyclic aging via online rainflow–analysis," *Energy Storage*, 2021.

[20] S. Loew, D. Obradovic, and C. L. Bottasso, "Model predictive control of wind turbine fatigue via online rainflow-counting on stress history and prediction," in *Accepted to TORQUE2020*, 2020.

[21] S. Loew, D. Obradovic, A. Anand, and A. Szabo, "Stage cost formulations of online rainflow-counting for model predictive control of fatigue," in *2020 European Control Conference (ECC)*, 2020, pp. 475–482.

[22] ASTM, "Standard practices for cycle counting in fatigue analysis," West Conshohocken, PA, 1985.

[23] The MathWorks Inc., "Rainflow counts for fatigue analysis," 2018. [Online]. Available: https://de.mathworks.com/help/signal/ref/rainflow.html

[24] G. Marsh, C. Wignall, P. R. Thies, N. Barltrop, A. Incecik, V. Venugopal, and L. Johanning, "Review and application of rainflow residue processing techniques for accurate fatigue damage estimation," *International Journal of Fatigue*, vol. 82, pp. 757–765, 2016.

[25] Y. Shi, B. Xu, Y. Tan, and B. Zhang, "A convex cycle-based degradation model for battery energy storage planning and operation," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 4590–4596.

[26] S. Loew and D. Obradovic, "Real-time implementation of nonlinear model predictive control for mechatronic systems using a hybrid model," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, 2018, pp. 164–167.

[27] Z. Gao, C. Chin, W. Woo, and J. Jia, "Integrated equivalent circuit and thermal model for simulation of temperature-dependent lifepo4 battery in actual embedded application," *Energies*, vol. 10, no. 1, p. 85, 2017.

[28] S. Gros, "An economic nmpc formulation for wind turbine control," in *52nd IEEE Conference on Decision and Control*, 2013, pp. 1001–1006.

[29] R. Verschueren, G. Frison, D. Kouzoupis, N. van Duijkeren, A. Zanelli, R. Quirynen, and M. Diehl, "Towards a modular software package for embedded optimization," in *Proceedings of the IFAC Conference on Nonlinear Model Predictive Control (NMPC)*, 2018.

[30] R. Verschueren, G. Frison, D. Kouzoupis, N. van Duijkeren, A. Zanelli, B. Novoselnik, J. Frey, T. Albin, R. Quirynen, and M. Diehl, "acados: a modular open-source framework for fast embedded optimal control," *arXiv preprint*, 2019. [Online]. Available: https://arxiv.org/abs/1910.13753

CHAPTER 5

Integrated design

5.1 Paper 10: Exploration of mechanical benefits from Lidar-assisted control

5.1.1 Summary

In this work, a design method is presented to rapidly assess the potential benefits of Lidar-assisted control (LAC) already at the first stages of wind turbine design. By revealing the potential of fatigue reduction, this work also supports the motivation for **Papers 1** - **7** and puts them into a new perspective. The method is applied to one offshore and two onshore reference machines, and comprises two steps:

1) An analysis of potential load reduction margins is performed, based on simulations of a full set of Design Load Cases (DLCs). A distinction is made between DLCs which are *modifiable* by LAC , or not modifiable and thus are *blocking* further improvement. A key load quantity (e.g. bending moment at tower bottom) can be improved if its maximum value is caused by a *modifiable* DLC. In this case, the potential load reduction margin is the difference between the leading *modifiable* and the first *blocking* DLC. For the reference machines, the margins are for blade extreme load at up to 30%, for blade tip deflection at up to 21%, and for tower extreme load close to zero. Noteworthy, for fatigue there are no blocking DLCs, and fatigue reductions from LAC fully translate into relaxed design constraints.

2) A numerical optimization for structural redesign is performed, where the design constraints are relaxed by factors obtained from the literature. This "load reduction model" replaces simulations with LAC in the loop. Additional optimistic and pessimistic scenarios are generated by a correction factor. Best results are obtained for the large offshore machine with tower mass reductions of up to 17%, blade mass reductions of up to 7%, and Levelized Cost of Energy (LCOE) reductions of up to 2%. A general finding is that expensive machines with fatigue-driven towers profit the most from LAC. Finally, further performance improvement of LAC is recommended since it decreases LCOE more than a cost reduction for Lidar hardware, and since remaining load reduction margins indicate high extra benefits.

5.1.2 Contribution

The author of this dissertation prepared the lidar load reduction model, and assisted in its application in the design framework. HC led the research work, developed the design optimization, conducted the design studies, and performed the analysis. CLB formulated the methodology and supervised the work. All authors provided important input to this research work through discussions, feedback and by writing the paper.

5.1.3 Reference

H. Canet, S. Löw, and C. L. Bottasso, “Lidar-assisted control in wind turbine design: Where are the potential benefits?” *Journal of Physics: Conference Series*, vol. 1618, p. 042020, 2020. doi:10.1088/1742-6596/1618/4/042020

The Science of Making Torque from Wind (TORQUE 2020) IOP Publishing
Journal of Physics: Conference Series **1618** (2020) 042020 doi:10.1088/1742-6596/1618/4/042020

Lidar-assisted control in wind turbine design: Where are the potential benefits?

H Canet[1], S Löw[1], CL Bottasso[1]

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

E-mail: carlo.bottasso@tum.de

Abstract. This study explores the potential benefits of considering Lidar-assisted control (LAC) at the first stages of wind turbine design. The proposed methodology starts with a load analysis of several reference wind turbines to understand which design constraints can be influenced by the use of LAC. The blade and tower of each analyzed model are redesigned considering LAC-induced reductions in key driving quantities. Preliminary results suggest modest reductions in LCOE with potentially significant benefits limited to the tower. The study also discusses the requirements on LAC system purchase and O&M costs, for both onshore and offshore machines, to achieve a reduction in LCOE.

1. Introduction

Turbine-mounted *Light detection and ranging* (Lidar) sensors are able to measure various properties of the incoming wind up to several hundred meters ahead of the wind turbine rotor plane. This preview information has been successfully used to augment conventional feedback controllers with feedforward loops [7], or to replace conventional controllers by advanced predictive ones [13]. These strategies are generically termed *Lidar-assisted control* (LAC).

Multiple studies have concluded that LAC can be used to improve the tracking of C_p, which can lead to an increase of AEP and can reduce fatigue damage and extreme loads in various structural components [7]. These benefits have so far been used to extend the lifetime of existing wind turbines, originally designed to operate with conventional controllers [14].

Even though significant research efforts are currently being devoted to the development of LAC, the benefits of considering LAC within the design of wind turbines are still not fully understood. Within the present work, LAC is considered already at the initial stages of turbine design to fully exploit its potential and reduce Levelized Cost of Energy (LCOE).

This study focuses on two main research questions: first, the paper analyses which key loads can be reduced by a basic LAC implementation and which LCOE reduction can be expected from a LAC-based redesigned turbine. Second, the paper explores the requirements on both the performance and cost of LAC for this device to be economically feasible.

This paper is organized as follows. Sections 2 and 3 are respectively devoted to the description of the approach and the models implemented to answer the two research questions. Section 4 describes the resulting effects of applying LAC at the first stages of design for three different turbines and three different scenarios. Furthermore, the required costs of Lidar are discussed for rendering LAC economically feasible.

2. Approach

The study starts with a load analysis of three reference wind turbines with the goal of understanding the potential reduction margin of their design constraints by the use of LAC. Each model is simulated under a variety of Design Load Cases (DLCs) [12], including power production with normal turbulence (DLC 1.1, DLC 1.2), extreme turbulence (DLC 1.3), loss of electrical network (DLC 2.1) and during extreme operating gusts (DLC 2.3). Also situations where the machine is parked are considered under multiple conditions, such as yaw misalignment (DLC 6.1), grid loss (DLC 6.2) and extreme yaw misalignment (DLC 6.3).

These DLCs are classified into two groups: *modifiable* and *blocking*, according to the influence of the controller on the load envelope. *Modifiable* DLCs are those in which the controller can modify the load envelope. *Blocking* DLCs represent conditions in which the controller performance does not effect the loads, as for example in parked conditions. Table 1 includes a detailed description of the DLCs considered in this study.

The potential load reduction is defined through rankings, where the values of each quantity are ranked in descending order, noting the originating DLC. The value of a key quantity can only be reduced by LAC if the ranking is led by a *modifiable* DLC. Its reduction potential is defined as the difference between its absolute maximum value and the value of the highest ranked *blocking* DLC.

As a second step, a baseline LAC load-reduction model is applied to the resulting loads of all *modifiable* DLCs. This model replaces the simulation of LAC and consists in the application of load-reduction coefficients to the load envelope resulting from aeroelastic simulations with a *non-LAC* controller. Differently performing LAC systems are considered by introducing an optimistic and a pessimistic scenario, defined by a correction factor. This factor multiplies the load-reduction coefficients and is defined as 1.5 for the optimistic scenario, 1 for the baseline scenario and 0.5 for the pessimistic one. The presented method intentionally does not commit to a specific Lidar hardware or controller types, and thus enables a fast preliminary generic assessment for design purposes. Finally, for all three scenarios, the structural redesign of the blades and tower is performed. The resulting changes in the structure are evaluated from an economic point of view through corresponding cost models, according to the wind turbine characteristics.

Table 1: Classification of Design Load Cases (DLC) according to the influence of the controller on the wind turbine load envelope. NTM = Normal Turbulence Model; ETM = Extreme Turbulence Model; EOG = Extreme Operating Gust; EWM = Extreme Wind speed Model

Classification	DLC	Seeds	Design situation	Wind speed	Wind profile	Other condition
Modifiable	1.1	3	Power production	V_{in}:V_{out}	NTM	
	1.2	3	Power production	V_{in}:V_{out}	NTM	
	1.3	3	Power production	V_{in}:V_{out}	ETM	
	2.1	3	Power production	V_{in}:V_{out}	NTM	Grid loss
	2.3 V_o	1	Power production	V_{out}	EOG	Grid loss
	2.3 V_r	1	Power production	$V_{rated} \pm 2m/s$	EOG	Grid loss
Blocking	6.1	3	Parked	V_{ref}	EWM 50 year	Yaw mis. ±8°
	6.2	3	Parked	V_{ref}	EWM 50 year	Grid loss
	6.3	3	Parked	V_{ref}	EWM 1 year	Ext. yaw mis. ±20°

3. Methodology and models

3.1. Aeroelastic simulation and design procedure

Aeroelastic calculations are performed with the Blade Element Momentum (BEM) based aeroelastic simulator `Cp-Lambda` (Code for Performance, Loads, Aeroelasticity by Multi Body Dynamic Analysis) [6], coupled with a conventional non-LAC controller [13]. This aeroelastic simulator is also the core of the wind turbine design suite, `Cp-Max` [1]. This code can perform the combined preliminary optimization of a wind turbine, including both blade and tower sizing. The optimization of the blade aeroelastic characteristics can be divided into two smaller subloops, which size the external aerodynamic shape and the structural components separately. In this work, the aerodynamic shape is kept frozen, and the turbine is redesigned only from the structural point of view. The structural optimization algorithm aims at minimizing blade cost, while guaranteeing its structural integrity and other requirements by enforcing a set of constaints. The optimization variables include the thickness of the structural elements for given blade layout and materials. The inertial and structural characteristics of each blade section are computed with the 2D finite element cross-sectional analysis code ANBA [10].

The tower structural sizing aims at minimizing tower cost, while satisfying a number of constraints to ensure the safety of the machine and other design requirements. The optimization variables include the diameter and thickness of the different tower segments for given material characteristics. The formal description of these algorithms can be found in [4, 1]. Both the blade and tower procedures employ a Sequential Quadratic Programming (SQP) optimization algorithm, in which gradients are computed by means of forward finite differences.

3.2. Baseline LAC load-reduction model

Multiple studies in the literature report the effects of LAC on loads for different controller formulations, such as feedforward, Linear Quadratic Regulator (LQR) or Model Predictive Control (MPC). Within these references, one specific study [7] is chosen to define the load-reduction model employed here. Reference [7] used a simple feedforward Lidar-assisted controller in combination with a conventional feedback controller on a 5 MW turbine.

The work reports reductions on an extensive set of loads for multiple components such as blade, tower and main bearing. Large reduction of fatigue loads resulting from DLC 1.2 for blade, main bearing, tower top and tower base are observed. Extreme loads resulting from DLC 2.3 also significantly benefit from the implementation of LAC. Table 2 reports the considered reduction coefficients for each component and *modifiable* DLC. For simplicity, this model does not include Lidar faults and assumes a Lidar availability of 100%.

DLC 2.1 deserves a specific discussion. Even though it is in principle a *modifiable* DLC, the precise estimation of LAC-induced reductions of extreme loads is difficult in this case. Extreme loads usually result from the wind turbine shut-down manoeuvre after grid disconnection. Since the wind excitation and the time at which grid loss occurs are both random, the state of the turbine at the time the shut-down manoeuvre is initatied is also random. This clearly makes it difficult to reliably estimate the load reduction, unless as specific dedicated simulation is conducted. Since these LAC-induced load reductions are still not fully described in the literature, they are here considered to be negligible.

In terms of Annual Energy Production (AEP), benefits are assumed to be 0.2% below rated speed and nonexistent above rated speed [7].

3.3. Cost models

The economic assessment is performed with different cost models. The SANDIA Blade Cost Model [11] is used to compute the blade cost for both onshore and offshore models. The 2015 NREL Cost Model [16], an updated version of the 2006 model [9], is applied for onshore machines, while the INNWIND Cost Model [8] is used for the study on offshore machines. The outputs of both models are expressed in 2020€, correspondingly inflated with the consumer price index and exchange rate. In order to ensure its comparability, LCOE is computed as

$$LCOE = \frac{FCR \cdot ICC}{AEP} + AOE, \tag{1}$$

where FCR [−] is the Fixed Change Rate, assumed to be 7%, ICC [€] is the Initial Capital Cost, AEP [MWh] is the Annual Energy Production and AOE [€/MWh] are the Annual Operating Expenses.
In addition to the standard turbine costs, the costs for the Lidar system have to be considered. Very conservative cost values for purchase and O&M of the Lidar hardware have been considered, based on the information from two major Lidar manufacturers. It has been assumed that two Lidar scanners have to be purchased over a turbine lifetime of 20 years. This results in additional $100,000$€ of ICC and $2,500$€/year of AOE. Only hardware-related costs have been regarded. Due to lack of information, the costs of development or licensing of Lidar-assisted turbine control software, related commissioning and software maintenance have been neglected.

Table 2: Load reduction coefficients considered in the baseline LAC load-reduction model

				BLADE			
	Description	Fx	Fy	Fz	Mx	My	Mz
DLC 1.1 & 1.3	Extreme loads					-2.0%	
	Tip deflection					-2.0%	
DLC 1.2	DEL	-3.8%	-0.1%	-0.25%	-0.4%	-3.8%	-3.5%
DLC 2.3	Extreme loads					-2.9%	
	Tip deflection					-2.9%	
				MAIN BEARING			
	Description	Fx	Fy	Fz	Mx	My	Mz
DLC 1.1 & 1.3	Extreme loads						
DLC 1.2	DEL	-10.0%			-1.2%	-0.4%	-1.0%
				TOWER TOP (YAW BEARING)			
	Description	Fx	Fy	Fz	Mx	My	Mz
DLC 1.1 & 1.3	Extreme loads						
DLC 1.2	DEL	-12.0%	-0.1%	-2.1%	-2.0%	-1.8%	-0.2%
				TOWER BASE			
	Description	Fx	Fy	Fz	Mx	My	Mz
DLC 1.1 & 1.3	Extreme loads					-5.0%	
DLC 1.2	DEL	-3.0%	0.2%	-2.2%	-0.1%	-12.0%	-0.2%
DLC 2.3	Extreme loads					-40.0%	

4. LAC-based effect on LCOE

4.1. Analyzed reference machines

The study is performed on three reference machines of different wind classes: an offshore 10 MW turbine (1A) [6] and two onshore machines: a 2.2 MW (2A) [1] and a 3.4 MW (3A) [2] turbine. The general characteristics of these turbines, including blade length and tower height, are described in Table 3. A detailed characterization of these machines can be found in the corresponding references.

Table 3: Main characteristics of the reference models included in the study

Turbine	1 [6]	2 [1]	3 [2]
IEC Class & Category	1A	2A	3A
Rated electric power [MW]	10	2.2	3.4
Rotor diameter [m]	178.3	92.4	130.0
Specific power [W/m^2]	400.5	298.3	252.4
Hub height [m]	119.0	80.0	110.0
Blade mass [t]	42.5	8.6	16.4
Tower mass [t]	628	125	553

These machines are representative of currently installed wind turbines. The costs of these three machines are compared to the costs of reference projects in the US in terms of capital (CAPEX), operational expenses (OPEX), AEP and LCOE. The first three figures are normalized by rated power. Table 4 shows a good match between the costs of the onshore 2A machine and a generic 2.32 MW turbine of a reference onshore project in the US in 2017 [15]. The costs of the 3A turbine, even if slightly higher for some figures, also follow well the values of the reference turbine. In the second column, the costs of a bottom-fixed offshore 5 MW machine are compared with the 1A machine used in this study. Large differences are found here, for instance in OPEX costs, due to the very different rating of the turbines. In general, the cost distribution presents a similar pattern to the considered reference. The cost breakdown comparison is expressed in 2017 United States Dollars (USD) and CAPEX does not include financial costs. LCOE has been recomputed in all cases as indicated in Eq. (1).

Table 4: Comparison of cost breakdown of the different reference models in 2017 USD

	Onshore			Offshore	
	Stehly et al. [15]	2A	3A	Stehly et al. [15]	1A
Rating [MW]	2.32	2.2	3.4	5	10
CAPEX [USD/kW]	1454	1297	1759	3846	4379
OPEX [USD/kW]	43.6	48.1	51.4	144	225
AEP [MWh/MW]	3633	3520	3866	3741	4500
FCR [%]	7.9	7.9	7.9	7.0	7.0
LCOE [USD/MWh]	43.6	42.9	49.2	110.5	118.1

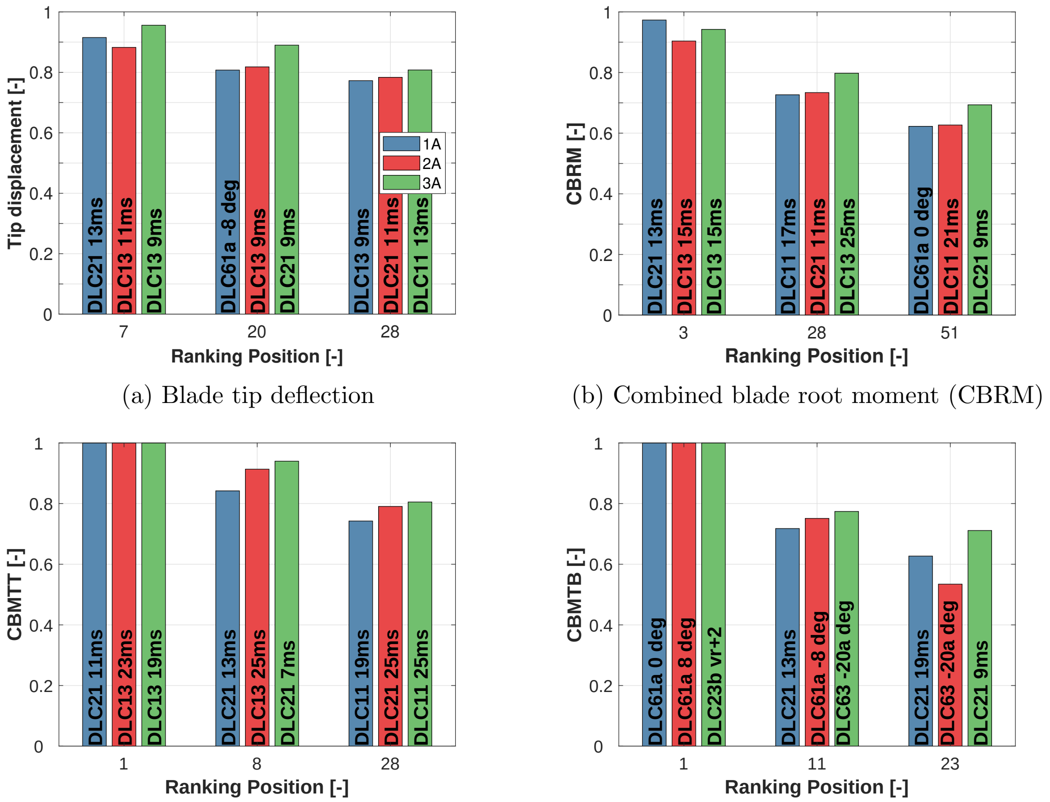
4.2. Load analysis: Potential reduction margins

The analysis of the load rankings highlights important potential reduction margins that could be exploited by LAC. Figure 1 provides an overview of the ranking position of the first *blocking* DLC for each machine.

Blade tip deflection: As shown in Fig. 1a, this ranking is led by *modifiable* DLCs and reductions are blocked by DLC 2.1 for all machines. More specifically, the reduction margin for turbine 1A is blocked at ranking position 7, for turbine 3A at ranking position 20, and for turbine 2A at ranking position 28. The analysis unveils potential reduction margins of tip displacement between 8% (1A) and 21% (2A). These margins could be exploited in the design of the blade, since maximum tip deflection is typically an important active design driver for the spar caps.

Extreme loads: The analysis of the combined blade root moment ranking (Fig. 1b) leads to similar conclusions. Indeed, DLC 2.1 is here also the first *blocking* DLC to appear for the three turbines, with large potential reduction margins in machines 2A and 3A.
Combined bending moment at tower top (Fig. 1c) shows no potential margin for machine 1A, and reduced margins for turbines 2A and 3A. This potential reduction could relax the buckling constraint. No potential reduction margin is found at tower bottom (Fig. 1d), a load clearly driven by *blocking* cases.

Fatigue: In the case of fatigue, only DLC 1.2 has to be considered according to the standards. Thus, there are no *blocking* DLCs and fatigue reductions from LAC fully translate into relaxed design constraints.



(a) Blade tip deflection

(b) Combined blade root moment (CBRM)

(c) Combined bending moment at tower top (CBMTT)

(d) Combined bending moment at tower bottom (CBMTB)

Figure 1: Ranking position of the first appearing *blocking* case for different key quantities and machines. Results are normalized with the maximum load of the respective turbine.

4.3. LAC-induced reductions in blade and tower mass

For the LAC-based redesign of the turbines, the load-reduction coefficients of Table 2 are applied to the loads resulting from initial *non-LAC* runs of all DLCs. Consequently, the design parameters are varied until convergence of the SQP algorithm. This sequence of simulation, application of reduction margins and optimization is repeated until convergence.

The LAC-based redesign leads to large reductions in tower mass and more modest savings in blade mass, as reported in Fig. 2.

Tower: Both the 1A and 3A towers enjoy significant benefits from the large reductions in fatigue and achieve a mass reduction between approximately 17% for the optimistic scenario and approximately 5% for the pessimistic one.

The tower of the 2A machine presents a smaller improvement due to different active design drivers. Indeed, this model presents an increased importance of the buckling constraints when compared to the other turbines. Even though a potential reduction margin of 10% was found in the load analysis for combined bending moment at tower top (Fig. 1c) —a load that may drive the tower buckling behavior— this is not reduced by LAC according to the applied load-reduction model (Table 2), and therefore cannot be exploited.

Blades: Modest reductions are achieved at the blades of all models and for all scenarios, due to the moderate influence of LAC in design-driving constraints. Even though a large potential margin was found for tip deflection, the applied LAC load-reduction model in Table 2 only allows for a reduction of 2%.

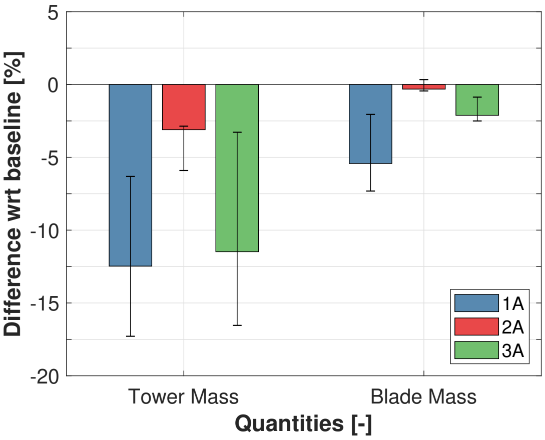


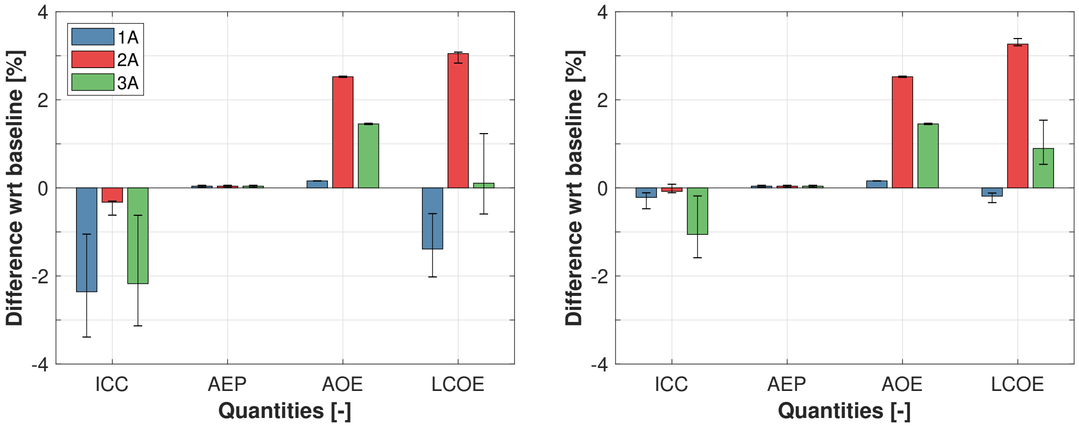
Figure 2: LAC-induced reductions in tower mass and blade mass. The confidence intervals show the values for the optimistic (lower end) and pessimistic scenarios (upper end), introduced by a correction factor.

4.4. LAC-induced reductions in LCOE

LAC-generated improvements not always translate into noticeable LCOE changes. For the 2A turbine, low tower mass reductions in combination with the significant Lidar costs clearly lead to an increase of LCOE. More interesting conclusions are obtained when analyzing the 1A and 3A machines.

While both machines present significant reductions in ICC, a different effect is observed in annual operating expenses (AOE). Indeed, the additional expenses created by maintenance of the Lidar system do not significantly increase the overall AOE for offshore machines, due to the already high O&M expenses. For onshore machines, these costs play a larger role and increase AOE by approximately 2%.the Additionally, AEP is slightly increased for both onshore and offshore machines. While turbine 1A achieves reductions between 0.5% and 2% in LCOE, for machine 3A a slight decrease of 0.5% is achieved with the best performing scenario.

The reduction in ICC reached by the blade redesign is generally not high enough to compensate for the increase of AOE. Therefore LCOE increases for all onshore machines and slightly decreases for the offshore machine.



(a) LAC-induced redesigned tower (b) LAC-induced redesigned blade

Figure 3: LAC-induced reductions in Initial Capital Costs (ICC), Annual Energy Production (AEP), Annual Operating Expenses (AOE) and Levelized Cost Of Energy (LCOE) for tower (left) and blade redesigns (right). The confidence intervals show the values for the optimistic (lower end) and pessimistic scenarios (upper end), introduced by the correction factor.

4.5. Cost sensitivity analysis

Finally, a cost sensitivity analysis is performed to understand how the purchase and maintenance cost of a Lidar system can influence the reduction in LCOE. The analysis is performed on the scenario with baseline LAC load-reduction coefficients.

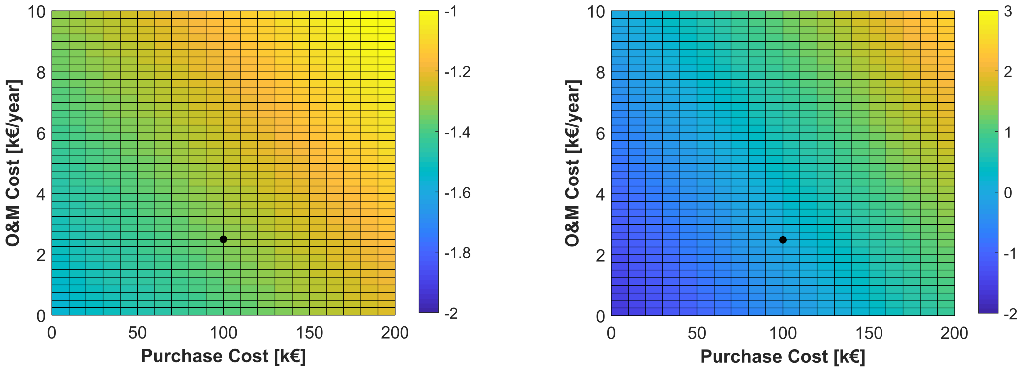
In the case of the 1A machine (Fig. 4a), LCOE shows little variability when both purchase and maintenance costs are modified. Larger effects are observed for the 3A machine (Fig. 4b). Here an increase or redution of LCOE can be obtained by modifying the LAC system costs. Both machines show modest reductions in LCOE, even with very low LAC costs, implying that a real effect on LCOE can only be achieved by an improved LAC performance in some key loads.

5. Conclusions

This paper has presented a preliminary analysis on the potential benefits of considering LAC at the first stages of wind turbine design. A first load analysis highlights an interesting potential for the blade as well as for fatigue-driven tower designs. For instance, potential reduction margins of up to 20% are observed in tip deflection, as well as in the combined bending moment at tower top. The current LAC systems are only partially exploiting the reduction potential of these loads. Indeed, according to the load-reduction model considered here, blade tip deflection is reduced up to 2% and combined bending moment at tower top is not reduced, leading to negligible benefits for the rotor and buckling-driven towers. Fatigue-driven towers enjoy a more significant effect, since fatigue is greatly reduced by LAC.

In terms of LCOE, the offshore machine shows the largest reductions for all considered systems. This turbine significally benefits from the reduced influence of the additional LAC-based costs, given the already high O&M and ICC figures. In this case, the purchase and O&M costs of

The Science of Making Torque from Wind (TORQUE 2020) IOP Publishing
Journal of Physics: Conference Series **1618** (2020) 042020 doi:10.1088/1742-6596/1618/4/042020



(a) Machine 1A

(b) Machine 3A

Figure 4: Sensitivity analysis of the effect of the purchase and O&M cost of LAC systems on LCOE reduction for the offshore (1A) and onshore (3A) machines. The baseline costs are indicated with a black circle.

Lidar do not play a large role in the overall achieved LCOE reduction. For the onshore machine, the effect on LCOE shows a higher sensitivity to the performance and costs of a LAC system.

This study only partially explores the benefits of introducing LAC in the early stages of wind turbine design. Indeed, even though this study gave some insight into how design differences in the tower can influence the LAC-induced benefits, a deeper analysis of the design drivers should be performed to get a more complete picture. Additionally, the list of analyzed DLCs should be expanded to include other design conditions that occasionally result in design drivers, such as DLC 1.4 (power production with extreme coherent gust with direction change) or DLC 1.5 (power production with extreme wind shear). Further work should also analyze the effects of a reduced Lidar availability, since the current load-reduction model assumes an availability of 100%. Additionally, other applications of LAC should be explored. For instance, LAC-induced lower loading could be exploited to increase hub height and gain power capture. Finally, it should be remarked that the use of a generic load model implies some significant approximations, and more precise conclusions could be obtained with the use of a Lidar simulator and a specific controller implementation.

6. Acknowledgements

The authors would like to thank the participants of the IEA Task 32+37 workshop “Optimizing Wind Turbines with Lidar-Assisted Control using Systems Engineering” for the valuable discussions.

7. References

[1] Bortolotti P and Bottasso CL and Croce A 2016 Combined preliminary-detailed design of wind turbines *Wind Energ. Sci.* **1** 71-88 10.5194/wes-1-71-2016

[2] Bortolotti P, Canet H, Dykes K, Merz K, Sethuraman L, Verelst D and Zahle F 2019 IEA Wind TCP Task 37: Systems engineering in wind energy - WP2.1 Reference wind turbines *National Renewable Energy Laboratory technical report* Golden, CO 10.2172/1529216

[3] Bottasso CL, Croce A, Savini B, Sirchi W and Trainelli L 2006 Aero-servo-elastic modeling and control of wind turbines using finite-element multibody procedures. *Multibody Syst. Dyn.* **16** 291-308 10.1007/s11044-006-9027-1

[4] Bottasso CL, Campagnolo F and Croce A 2012 Multi-disciplinary constrained optimization of wind turbines *Multibody Syst. Dyn.* **27** 21-53 10.1007/s11044-011-9271-x

[5] Bottasso CL, Croce A, Nam Y and Riboldi CED 2012 Power curve tracking in the presence of a tip speed constraint *Ren. En.* **40** 1-12 10.1016/j.renene.2011.07.045

[6] Bottasso CL, Bortolotti P, Croce A and Gualdoni F 2016 Integrated aero-structural optimization of wind turbines *Multibody Syst. Dyn.* **38** 317-344 10.1007/s11044-015-9488-1

[7] Bossanyi E, Kumar A and Hugues-Salas O 2014 Wind turbine control applications of turbine-mounted Lidar *J. Phys.: Conf. Ser.* **555** 012011 10.1088/1742-6596/555/1/012011

[8] Chaviaropoulos P, Karga I, Harkness C and Hendriks B 2014 Deliverable 1.23 PI-Based assessment of innovative concepts *INNWIND.EU technical report*

[9] Fingersh L, Hand M and Laxson A 2006 Wind turbine design cost and scaling model *National Renewable Energy Laboratory technical report* Golden, CO NREL/TP-500-40566

[10] Giavotto V, Borri M, Mantegazza P and Ghiringhelli G 1983 Anisotropic beam theory and applications *Comput. Struct.* **16** 403-13 10.1016/0045-7949(83)90179-7

[11] Griffith DT and Johans W 2013 Large blade manufacturing cost studies using the sandia blade manufacturing cost tool and Sandia 100-meter blades *Sandia National Laboratories technical report* Albuquerque, NM SAND2013-2734

[12] International Electrotechnical Commission 2005 IEC 61400-1 Ed.3: Wind turbines - Part 1: Design requirements

[13] Riboldi CED 2012 Advanced control laws for variable-speed wind turbines and supporting enabling technologies *Politecnico di Milano Doctoral dissertation* Milan, Italy

[14] Schlipf D, Fürst H, Raach S and Haizmann F 2018 Systems engineering for Lidar-assisted control: A sequential approach. *J. Phys.: Conf. Ser.* **1102** 012014 10.1088/1742-6596/1102/1/012014

[15] Stehly T, Beiter P, Heimiller D and Scott G 2017 Cost of wind energy review *National Renewable Energy Laboratory technical report* Golden, CO NREL/TP-6A20-72167

[16] WISDEM Repository 2020 github.com/WISDEM

5.2 Paper 11: Redesign of key properties utilizing Lidar-assisted control

5.2.1 Summary

In **Paper 10**, the structural elements - and thus the mass of tower and blades - have been re-optimized for Lidar-assisted control (LAC), while the macroscopic dimensions and lifetime have remained unchanged. The present publication extends this work by three additional re-optimization scenarios:

1. re-optimization of the tower structural elements with manually increased tower height while keeping turbine lifetime unchanged;
2. re-optimization of the rotor structural elements with manually increased rotor diameter while keeping turbine lifetime unchanged;
3. re-optimization of the tower structural elements with manually extended tower lifetime while keeping the macroscopic dimensions unchanged.

In Scenario 1) and 2), the procedure is to increase the respective macroscopic dimension, optimize the turbine with the default controller, and finally optimize the turbine with the LAC load reduction model. For the large offshore turbine, a redesign with 15% higher tower leads to 2% higher annual energy production (AEP), since the rotor is exposed to higher wind speeds at higher altitudes. Due to the load reduction of LAC, this redesign is possible without significantly increasing the tower mass. For small onshore turbines, the tower increase did not lead to a benefit in LCOE. Rotor increases neither led to LCOE benefits since lower limits for material thicknesses are hit, and thus fatigue reduction benefits cannot be exploited.

In Scenario 3), the lifetime constraint is manually increased, and consequently the tower is optimized with the LAC load reduction model. For the large offshore turbine, again, the most promising result was obtained: for a doubled turbine lifetime of 40 years with LAC, the tower mass still can be reduced by 10%. Similar results also could be achieved with a smaller onshore machine.

General findings beyond the ones of **Paper 10** are:

- Particularly for the tower, there is very high improvement potential via the reduction of fatigue, but less via the reduction of ultimate loads.
- Among all redesign options with LAC, the design for higher lifetime is most promising.

5.2.2 Contribution

The author of this dissertation prepared the lidar load reduction model, and assisted in its application in the design framework. HC led the research work, developed the design optimization, conducted the design studies, and performed the analysis. CLB formulated the methodology and supervised the work. HC and CLB wrote the paper. All authors provided important input to this research work through discussions, feedback and by improving the manuscript.

5.2.3 Reference

H. Canet, S. Loew, and C. L. Bottasso, “What are the benefits of lidar-assisted control in the design of a wind turbine?” *Wind Energy Science*, vol. 6, no. 5, pp. 1325–1340, 2021. [Online]. Available: https://wes.copernicus.org/articles/6/1325/2021/. doi:10.5194/wes-6-1325-2021

Wind Energ. Sci., 6, 1325–1340, 2021
https://doi.org/10.5194/wes-6-1325-2021

eawe european academy of wind energy
Open Access
WIND ENERGY SCIENCE

What are the benefits of lidar-assisted control in the design of a wind turbine?

Helena Canet, Stefan Loew, and Carlo L. Bottasso

Wind Energy Institute, Technical University of Munich, 85748 Garching bei München, Germany

Correspondence: Carlo L. Bottasso (carlo.bottasso@tum.de)

Received: 2 December 2020 – Discussion started: 29 December 2020
Revised: 5 July 2021 – Accepted: 19 August 2021 – Published: 8 October 2021

Abstract. This paper explores the potential benefits brought by the integration of lidar-assisted control (LAC) in the design of a wind turbine. The study identifies which design drivers can be relaxed by LAC, as well as by how much these drivers could be reduced before other conditions become the drivers. A generic LAC load-reduction model is defined and used to redesign the rotor and tower of three representative turbines, differing in terms of wind class, size, and power rating. The load reductions enabled by LAC are used to save mass, increase hub height, or extend lifetime. For the first two strategies, results suggest only modest reductions in the levelized cost of energy, with potential benefits essentially limited to the tower of a large offshore machine. On the other hand, lifetime extension appears to be the most effective way of exploiting the effects of LAC.

1 Introduction

Wind turbines are highly dynamical systems, excited by stochastic and deterministic disturbances from wind. Among their various goals, wind turbine control systems aim at limiting structural loads. In fact, lower ultimate and fatigue loading can be exploited to reduce mass and cost or to design larger and taller turbines that can generate more energy; in turn, all these effects may lead to a reduction of the cost of energy.

Traditional wind turbine controllers rely on feedback measurements to drive blade pitch, generator torque, and yaw. Since they operate based on the response of the system as expressed by live measurements, these controllers are only capable of reacting to wind disturbances that have already impacted the wind turbine. This is an intrinsic limitation of all feedback-based mechanisms: since control actions are based on past measurements, the controller is always “late”, in the sense that it reacts to events that are already taking place. To improve on this situation, control systems can be augmented with preview information, which informs the controller on the wind that will affect the turbine in the immediate future.

Wind preview can be obtained from turbine-mounted light detection and ranging (lidar) sensors, which are capable of measuring various properties of the incoming flow field up to several hundred meters in front of the rotor. Lidar-augmented control strategies are generically termed lidar-assisted control (LAC).

Several LAC formulations have already been investigated, and their performance in terms of power capture and load mitigation are reported in the literature. Bossanyi et al. (2014) describe a standard feedback controller enhanced by a feedforward blade pitch branch enabled by lidar wind preview. Results indicate promising reductions in blade flap and tower fore–aft fatigue damage, without any appreciable loss in power production. Similar benefits are described by other sources such as, for example, Dunne et al. (2011, 2012). Benefits have also been confirmed in the field (Schlipf et al., 2013c), albeit to the present date only on a small research wind turbine. Feedforward torque control strategies have also been investigated; results indicate marginal increments in mean power capture at the expense of high power and torque variations (Bossanyi et al., 2014; Wang et al., 2013; Schlipf et al., 2013). More advanced formulations, such as nonlinear model-predictive controllers (Schlipf et al., 2013b) or flatness-based controllers (Schlipf et al., 2014), have also been enhanced with lidar wind preview information. Promising results were reported in terms of load reductions and power increase, at the expense of a much higher computational cost, which makes real-time execution more challenging to achieve and test in the field (Scholbrock et al., 2016).

Even though the potential of LAC is widely recognized, the system-level benefits that LAC may possibly bring to the levelized cost of energy (LCOE) are still not fully understood. In general two strategies have been suggested for reducing LCOE by LAC (Schlipf et al., 2018). The first is the *retrofit strategy*, which consists in using lidars to extend the lifetime of a wind turbine that has already been designed and installed. For example, Schlipf et al. (2018) reported the extension of the lifetime of a tower by 15 years. A second strategy is the *integrated approach*, in which LAC is considered as part of the system from its very inception. The idea in this second case is that, by considering LAC within the design process, its full potential can be realized by translating the benefits of load reductions directly into an improved turbine. Indeed, the adoption of a holistic system-level design approach was identified as an opportunity to assess the cost-benefit tradeoffs among turbine, lidar and control system by two IEA Wind Tasks: Task 32 on wind energy lidar systems, and Task 37 on systems engineering for wind energy (Simley et al., 2018, 2020).

This work aims at taking a first step in this direction, providing an initial rough assessment of the potential benefits of considering LAC in the sizing of the two primary components of a wind turbine, namely the rotor and tower. The present work refines and expands the study described in Canet et al. (2020). In a nutshell, this study tries to give a preliminary general answer to the following main research questions:

- To which extent can design-driving constraints be relaxed by LAC?
- What is the best way of reaping the benefits brought by LAC in the design of rotor and tower?
- To make LAC beneficial at the system level, is it necessary to improve its performance or reduce its cost?

The present investigation intentionally does not commit to a specific lidar hardware or control formulation. In fact, the effects of LAC are considered here through a load-reduction model, defined according to the average performance of LAC systems reported in the literature. To understand trends, rather than focusing on a specific case, this baseline average literature-sourced model is expanded to cover an optimistic and a pessimistic scenario, thereby providing a range of possible behaviors. The study is performed on three wind turbines, which differ for wind class, size and power rating. These three reference machines are reasonable representations of current wind turbines available on the market. Clearly, the application of a literature-sourced range of load-reductions to three very different machines cannot give final and precise answers, which would require dedicated turbine and control-specific analyses conducted with coupled LAC-turbine simulations. However, the present approach offers a way of obtaining an initial preliminary assessment of the potential benefits of adopting LAC, and it helps pinpoint the most promising applications that should be further analyzed.

The paper is organized as follows. Section 2 describes the approach and the models used in the study, while Sect. 3 analyzes the potential benefits of integrating LAC in the design of the tower and rotor of three different reference wind turbines. The study considers mass (and hence cost) reductions of these two components, but also investigates the design of towers that are taller or have a longer lifetime, including the effects of the purchase and maintenance costs of the onboard lidar system. Section 4 closes the paper by reporting and discussing the main conclusions of the study.

2 Approach

Figure 1 presents a graphical depiction of the approach used in the present work. In a first phase, each turbine model is analyzed using a baseline non-LAC controller. This analysis highlights the benefits of reducing some design-driving quantity, and indicates by how much that quantity could be improved before another effect starts driving the design. Based on this information, a second phase of the analysis initially considers each turbine equipped with a LAC controller, and then exploits the obtained load-reduction benefits to perform a structural redesign. Finally, the improved design is subjected to an economic analysis, whose goal is to establish tradeoffs between weight savings made possible by LAC and the additional expenses due to the purchase and operation and maintenance (O&M) costs of the lidar. More details on these analysis processes are provided in the next sections.

2.1 Assessment of potentially exploitable margins

Design-driving quantities are those key indicators that define active constraints, thereby affecting the design solution. Design-driving quantities can be modified by LAC – or, more in general, by any control or technological solution – only to some extent, past which some other effect beyond the reach of LAC becomes the driver, preventing further improvements. The extent by which a design-driving quantity can be affected before another one becomes the driver is called here a *potentially exploitable margin* (PEM). It is an exploitable margin because, if it can be achieved, the design-driving constraints can be relaxed and, therefore, the design can be improved. It is, however, only a potential margin because it represents an upper bound: in fact, a smaller improvement might be actually obtainable by LAC than this maximum limit.

A PEM is clearly a very valuable piece of information: there is no point in using LAC to reduce a certain quantity past the value where the driver switches to some condition that is not controllable by LAC. In fact, any further reduction would be futile, as it would not affect the design-driving constraints and therefore the final design.

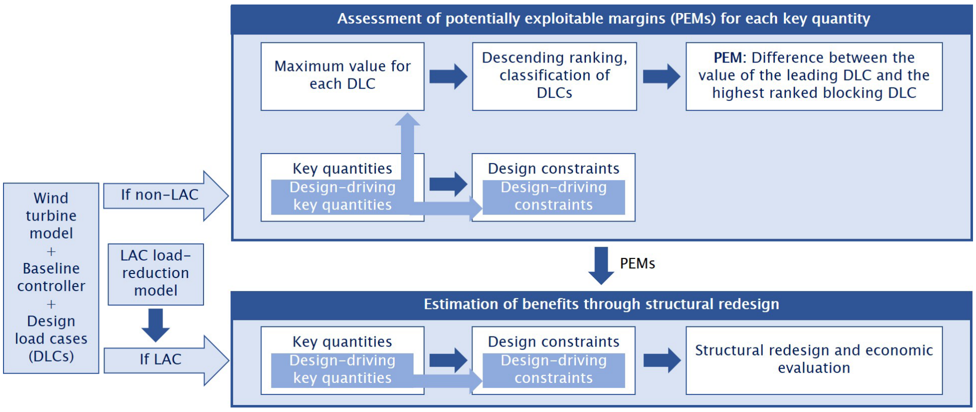

Assessment of potentially exploitable margins (PEMs) for each key quantity
Maximum value for each DLC
Descending ranking, classification of DLCs
PEM: Difference between the value of the leading DLC and the highest ranked blocking DLC
Key quantities
Design-driving key quantities
Design constraints
Design-driving constraints
Wind turbine model + Baseline controller + Design load cases (DLCs)
If non-LAC
LAC load-reduction model
If LAC
PEMs
Estimation of benefits through structural redesign
Key quantities
Design-driving key quantities
Design constraints
Design-driving constraints
Structural redesign and economic evaluation

Figure 1. Approach overview.

These considerations clearly do not apply exclusively to LAC but more in general hold for any technology that has the potential to relax the design constraints of a system. Therefore, the analysis of PEMs is an extremely useful exercise, because

- it is able to highlight the possible design benefits brought by the introduction of a new technology and
- it gives a target maximum margin of improvement that that technology should bring.

In the context of the current analysis, the assessment of PEMs is based on key quantities such as ultimate and fatigue loads, and elastic deflections, which result from the aeroservoelastic simulation of a comprehensive set of design load cases (DLCs) run with a non-LAC controller. DLCs represent the different operating conditions that a wind turbine encounters throughout its lifetime, as defined by certification standards (IEC, 2005).

For the purposes of this work, DLCs are classified in two distinct groups: *modifiable* and *blocking*. In modifiable DLCs, the maximum value of each key quantity depends on the controller. For example, this is the case of loads obtained in power production conditions (DLC 1.X). In fact, by modifying the pitch–torque controller of the turbine, the response of the machine changes, and consequently the loads that are produced also change. On the contrary, in blocking DLCs the key quantities are not affected by the controller. For instance, this is the case of loads generated in parked conditions (DLC 6.X). In fact, as the pitch–torque controller is not active when the turbine is parked, it clearly cannot influence the loads that are generated in that condition. Table 1 presents a classification of a selection of DLCs, including a description of the corresponding operating condition.

PEMs are obtained via a two-step procedure.

First, the (active) design constraints that determine the sizing of a given wind turbine component are identified; these are termed *design drivers* or *design-driving constraints*. Design constraints are introduced in the structural design process of a wind turbine component to guarantee structural safety during its lifetime, ensuring that admissible values for stress, strain, and fatigue damage are never exceeded. Additional constraints are enforced to guarantee a safe clearance and to avoid collisions between the blade and tower, to prevent buckling, and to ensure all other desired characteristics from the resulting design (Bottasso and Bortolotti, 2019). These constraints are functions of the key quantities resulting from the various DLCs, augmented by safety factors as prescribed by the norms. Other constraints, such as those enforced to avoid resonant conditions, are not dependent on DLCs.

Second, the maximum value of a key quantity is extracted from each considered DLC. The values are then sorted in descending order and labeled with the indication of the originating DLC. Each DLC is classified as modifiable or blocking. Clearly, the maximum value of a key quantity can only be reduced by LAC if its ranking is led by a modifiable DLC. The PEM is computed for each design-driving key quantity, and it is obtained as the difference between the quantity maximum value and the value of the highest-ranked blocking DLC.

2.2 Estimation of benefits through structural redesign

PEMs can be exploited to improve the structural design of the wind turbine components that are driven by modifiable DLCs. To this end, DLCs should be run again, this time using LAC to yield new reduced values of the key quantities. However, as argued earlier on, instead of focusing on a particular case, it is more interesting to perform an analysis that is less specific and more general in character. To this end, a LAC load-reduction model was used here instead of re-running all DLCs with a given LAC in the loop. The load-reduction model is simply represented by a set of multiplicative co-

Table 1. Classification of a selection of the design load cases into *modifiable* and *blocking* (see text for a definition). NTM: normal turbulence model; ETM: extreme turbulence model; ECD: extreme coherent gust with direction change; EWS: extreme wind shear; EOG: extreme operating gust; EWM: extreme wind speed model.

Classification	DLC	Design situation	Wind speed	Wind profile	Other condition
Modifiable	1.1	Power production	$V_{\mathrm{in}} : V_{\mathrm{out}}$	NTM	
	1.2	Power production	$V_{\mathrm{in}} : V_{\mathrm{out}}$	NTM	
	1.3	Power production	$V_{\mathrm{in}} : V_{\mathrm{out}}$	ETM	
	1.4	Power production	$V_{\mathrm{rated}} \pm 2\,\mathrm{ms}^{-1}$	ECD	
	1.5	Power production	$V_{\mathrm{in}} : V_{\mathrm{out}}$	EWS	
	2.1	Power production	$V_{\mathrm{in}} : V_{\mathrm{out}}$	NTM	Grid loss
	2.3	Power production	$V_{\mathrm{out}}, V_{\mathrm{rated}} \pm 2\,\mathrm{ms}^{-1}$	EOG	Grid loss
Blocking	6.1	Parked	V_{ref}	EWM 50 year	Yaw mis. ±8°
	6.2	Parked	V_{ref}	EWM 50 year	Grid loss
	6.3	Parked	V_{ref}	EWM 1 year	Ext. yaw mis. ±20°

efficients, which are defined for each key quantity associated with a modifiable DLC. Each coefficient expresses how LAC affects a key quantity with respect to a non-LAC controller; therefore, load reductions correspond to coefficients smaller than one in the model. Clearly, such coefficients depend on a multiplicity of factors, such as the specific control formulation, the tuning of its gains, or the performance of the lidar system. While a specific analysis is crucial when actually designing a wind turbine and its control system, a specific analysis also clearly hinders somehow the generality of the results and conclusions that can be drawn from it. In this spirit, a range of possible performances – in contrast to a case-specific performance – is considered here by defining different load-reduction scenarios. The load-reduction model and additional scenarios are based on results sourced from the literature, as more precisely discussed in Sect. 2.3.

The application of a LAC load-reduction model lowers some of the key quantities, in turn deactivating the associated design-driving constraints. To exploit the slack generated by LAC in the formerly active constraints, a redesign is performed to determine the structure that minimizes a desired figure of merit while guaranteeing structural integrity, in turn reactivating the constraints. After the redesign, an economic evaluation reveals the potential gains in LCOE, as discussed in Sect. 2.4.

2.3 LAC load-reduction model

The load-reduction model is based on a literature survey. The study reported in Bossanyi et al. (2014) was chosen as reference, because it presents a comprehensive list of the effects of LAC for several key quantities of various components. Additionally, that work was based on a rather standard controller, which might be representative of an initial conservative deployment on production machines. The implementation used a simple feedforward collective pitch LAC combined with a conventional feedback controller, applied to a 5 MW turbine. The paper reports a significant reduction of damage equivalent loads (DELs) resulting from DLC 1.2 for the blades, main bearing, tower top, and tower bottom. Extreme loads resulting from extreme operating gust conditions also experience significant benefits. On the other hand, power capture – and hence annual energy production (AEP) – is largely unaffected by this LAC implementation.

The load-reduction model derived from Bossanyi et al. (2014) is reported in Table 2 for each component and modifiable DLC, in terms of percent changes with respect to a non-LAC controller. In the table, F and M respectively indicate force and moment components, expressed in the (x, y, z) righthanded triad, where x points downstream, y is in the crossflow direction, and z points vertically upwards. Components not reported in the table experience either null or negligible reductions.

The load-reduction model reported in Table 2 prompts a few important remarks.

First, the model only includes DLC 1.1, 1.2, and 1.3, which represent power production cases. In reality, these are not the only DLCs that are modifiable – in the sense that they can be affected by a change in the controller. In fact, additional modifiable DLCs are represented by DLC 1.4 (power production with extreme wind direction), 1.5 (power production with extreme wind shear), 2.1 (power production with control system fault or grid disconnection under normal turbulence conditions), and 2.3 (power production with control system fault or grid disconnection under extreme operating gusts). The first two of these DLCs are not considered in the LAC load-reduction model because they do not typically generate design-driving loads, as further explained in Sect. 3.1. The case of DLC 2.1 and 2.3 is, however, different: here, maximum loads are typically generated during a shutdown, triggered by an extreme ambient condition change, a fault, or a grid disconnection. When this happens, the entity of the generated loads will be largely dictated by the behavior of the shutdown procedure, which here is as-

Table 2. Load-reduction coefficients based on Bossanyi et al. (2014), expressed as percentages with respect to a non-LAC controller.

Blade							
	Key quantity	Fx	Fy	Fz	Mx	My	Mz
DLC 1.2	DEL	−3.8 %	−0.1%	−0.25 %	−0.4 %	−3.8 %	−3.5 %
DLC 1.X	Extreme loads					−2.0 %	
	Tip deflection					−2.0 %	
Main bearing							
	Key quantity	Fx	Fy	Fz	Mx	My	Mz
DLC 1.2	DEL	−10.0 %			−1.2 %	−0.4 %	−1.0 %
DLC 1.X	Extreme loads						
Tower top (yaw bearing)							
	Key quantity	Fx	Fy	Fz	Mx	My	Mz
DLC 1.2	DEL	−12.0 %	−0.1 %	−2.1 %	−2.0 %	−1.8 %	−0.2 %
DLC 1.X	Extreme loads						
Tower bottom							
	Key quantity	Fx	Fy	Fz	Mx	My	Mz
DLC 1.2	DEL	−3.0 %	0.2 %	−2.2 %	−0.1 %	−12.0 %	−0.2 %
DLC 1.X	Extreme loads					−5.0 %	

sumed not to be assisted by a lidar for safety reasons. On the other hand, loads generated during a shutdown might also depend to some extent on the state of the turbine at the time the shutdown was triggered, which does depend on the behavior of the LAC controller. A precise quantification of the effects of LAC on these DLCs would therefore require simulations with LAC in the loop, which are outside of the scope of the present preliminary work.

This point, however, leads to a second, more general, observation: the model in fact includes both DELs and extreme loads, neglecting lidar faults and assuming a lidar availability of 100 %. While faults and availability (as long as it is not excessively low) will not impact DELs significantly, the situation is much more complicated for extreme loads. In fact, the malfunctioning of a lidar might in principle generate increases in ultimate loads, compared to a non-LAC case. A precise analysis of the possible faults and their consequences is clearly not only complex, but also highly case-specific. A mitigation of negative effects caused by faults could be achieved, for example, through triple modular redundancy (Koren and Krishna, 2020), which would, however, clearly affect costs. A comprehensive analysis of these effects is outside of the scope of the present simplified study, and fault-induced increases of ultimate loads are therefore neglected here. Although this is an apparently strong assumption, in the end it does not affect the results of this study. In fact, as shown later, the benefits of the present LAC model on the turbines considered here are confined to fatigue mitigation, and hence only fatigue-driven components do benefit from LAC in this study. At a more general level, one could wonder whether system-level benefits could be obtained by using LAC also for components driven by ultimate loads. While this remains an open question for now – as the present work is not able to provide definitive answers – it is clear that such an approach drastically raises the bar in terms of the complexity of the analysis and of the implementation, because of its obvious safety-related implications.

Third, differences in the formulation and tuning of a LAC controller will generally imply different reductions of key quantities. To estimate these effects, the results obtained from various authors were compared. The most complete set of results was found for DLC 1.2 in terms of DELs for fore–aft tower bending at tower bottom (FABMTB), flapwise blade root moment (FBRM), and shaft torsional moment (STM), as reported by Schlipf et al. (2014, 2015), Bottasso et al. (2014), Haizmann et al. (2015), Schlipf (2016), and Sinner et al. (2018). Table 3 reports the outcome of this analysis. There is a significant scatter in the results, especially for DEL FBRM and DEL STM, because of the variety of controller formulations and target wind turbine models. For instance, for DEL STM Schlipf et al. (2014) report a load reduction of 30 % using a flatness-based feedforward controller, while Schlipf (2016) reports an improvement of 6 % when using a feedforward–feedback controller. The lower values reported in Bossanyi et al. (2014) are most likely caused by the utilization of a fairly simple controller.

The scatter shown in Table 3 motivates the definition of two additional sets of coefficients that represent optimistic

Table 3. LAC-enabled load reductions from Bossanyi et al. (2014) compared to other references.

	Bossanyi et al. (2014)	Additional literature
DEL FABMTB	−12%	−16.4% ± 9.1%
DEL FBRM	−3.8%	−13.4% ± 6.6%
DEL STM	−1.2%	−11.8% ± 9.3%

and pessimistic scenarios and provide a more general view of the benefits of LAC. The optimistic scenario is obtained by multiplying the baseline coefficients by a factor of 1.5, whereas the pessimistic one is obtained by using a factor of 0.5. Here again, it is worth remembering that the present study does not target one specific LAC controller but aims at understanding basic trends.

A distinction must be made between the application of load-reduction coefficients to ultimate loads and deflections, which is straightforward (with the caveat of the effects of faults, as previously discussed), and to fatigue loads. The former simply consists in the correction of the key quantities obtained by a non-LAC controller with the corresponding coefficients of the load-reduction model. Combined loads – for example, at tower base or at the main and blade pitch bearings – are computed from the corrected individual load components.

For fatigue damage, the following procedure is used. Site-weighted DELs are computed as

$$\mathrm{DEL} = \sum_{v=V_\mathrm{in}}^{v=V_\mathrm{out}} f(v) L_\mathrm{eq}(v), \tag{1}$$

where $f(v)$ is the Weibull probability density function at a wind speed v, while the damage equivalent load at that same wind speed is expressed as

$$L_\mathrm{eq} = \left(\frac{\sum_{i=1}^{n} S_{\mathrm{r},\,i}^{m}}{N_\mathrm{eq}} \right)^{1/m}, \tag{2}$$

where m is the Wöhler coefficient, $S_{\mathrm{r},\,i}$ is the load range of a cycle i, n is the total number of cycles, and N_eq is the equivalent number of cycles (Hendriks and Bulder, 1995).

To compute LAC-reduced DELs, it is assumed that load reductions are independent of wind speed and load range. This way, the Weibull-weighted DEL reductions reported in the literature can be applied directly to the load time histories obtained here with a non-LAC controller by aeroelastic simulations. Clearly this is an approximation, as LAC-enabled reductions generally depend on the wind speed, as reported by several studies (Bottasso et al., 2014; Schlipf et al., 2018, 2013). However, it was verified by aeroelastic analyses that this assumption does not significantly affect the results when the reduction coefficients are small, as those reported in Tables 2 and 3. For example, with reference to Table 3, considering the DEL FBRM reduction of −3.8 %, the difference in fatigue margin at the blade root between wind-speed-dependent and independent reductions was found to be less than 2 %; for the DEL FABMTB reduction of −12 %, the fatigue margin difference at tower base was found to be approximatively equal to 5%. Given the character of this study, these differences were deemed to be acceptable and well within the margin of uncertainty of the analysis.

To complete the calculation of LAC-reduced DELs, transient combined loads are computed from the relevant components (for example, combining fore–aft and side–side components at tower base and similarly combining the associated components at the main and pitch bearings) and then processed by rainflow counting to obtain DELs, finally searching for the point in the cross section of interest with the maximum damage. The computation of fatigue margin constraints for the steel tower is performed following the European regulations (EN 1993-1-9, 2006).

2.4 Economic evaluation

During the redesign phase, the components are evaluated from an economic point of view through suitable cost models, based on the characteristics of the wind turbine. The 2015 NREL cost model (NREL, 2020), which is an updated version of the 2006 model (Fingersh et al., 2006), is used for onshore machines, whereas the INNWIND cost model (Chaviaropoulos et al., 2014) is used for offshore turbines. The blade cost for both onshore and offshore models is computed based on the SANDIA cost model (Griffith and Johans, 2013). All cost model estimates are expressed in 2020 Euros (EUR), inflated by the consumer price index and exchange rate. The comparison of the various designs is based on LCOE, which is computed as

$$\mathrm{LCOE} = \frac{\mathrm{FCR} \cdot \mathrm{ICC}}{\mathrm{AEP}} + \mathrm{AOE}, \tag{3}$$

where FCR [–] is the fixed change rate, ICC [EUR] the initial capital cost, AEP [MWh] the annual energy production, and AOE [EUR/MWh] the annual operating expenses.

2.5 Design and simulation environment

Aeroelastic analyses are performed with the blade element momentum (BEM)-based aeroelastic simulator Cp-Lambda (Bottasso et al., 2016), coupled with a conventional non-LAC controller (Riboldi et al., 2012). The aeroelastic simulator Cp-Lambda is also the core of the wind turbine design suite Cp-Max (Bottasso and Bortolotti, 2019; Bortolotti et al., 2016). This code can perform the combined preliminary optimization of a wind turbine, including both rotor and tower sizing.

The optimization of the blade aeroelastic characteristics can be divided into two coupled sub-loops, which size the external aerodynamic shape and the structural components. In this work, the aerodynamic shape of the blade is kept frozen, and the rotor is redesigned only from the structural point of view. The blade structural optimization algorithm aims at minimizing cost while guaranteeing structural integrity and other requirements by enforcing a set of constraints that include, among others, extreme conditions, fatigue damage, buckling, tower clearance, frequency placement, manufacturability, and transportation. The optimization variables include the thickness of the structural elements (skin, spar caps, shear webs) for given blade layout and materials. The inertial and structural characteristics of each blade section are computed with the 2D finite-element cross-sectional analysis code ANBA (Giavotto et al., 1983).

The structural sizing of the tower aims at minimizing its cost while satisfying constraints from extreme loads, buckling, and fatigue damage, as well as geometric constraints for manufacturing and transportation. The optimization variables include the diameter and thickness of the different tower segments for given material characteristics.

The formal description of the design algorithms can be found in Bottasso et al. (2012) and Bortolotti et al. (2016). Optimization is based on sequential quadratic programming (SQP), where gradients are computed by means of forward finite differences.

3 Results

The potential benefits of adopting LAC in the early stages of the design of the rotor and tower of different wind turbines are analyzed next, following the approach described in Sect. 2.

3.1 Reference machines

Three reference wind turbines are considered: WT1, an offshore class 1A developed in Bottasso et al. (2016) as an evolution of the original DTU 10 MW reference wind turbine (Bak et al., 2013); WT2, an onshore class 2A (Bortolotti et al., 2016); and WT3, an onshore class 3A (Bortolotti et al., 2019). The principal characteristics of these machines are reported in Table 4, while additional details can be found in the corresponding references. These turbines are reasonable representatives of current products available on the market. The three machines have blades made of a glass-reinforced polymer and towers made of thin-walled tubular tapered steel sections.

Table 5 compares the three machines in terms of capital cost (CAPEX), operational expenses (OPEX), AEP, and LCOE with some actual installations in the United States according to Stehly et al. (2017). The cost breakdown is expressed in 2017 United States Dollars (USD), and CAPEX does not include financial costs. The comparison shows a good match between the costs of the onshore 2.2 MW WT2 turbine and the 2017 US land-based 2.32 MW machine. The costs of the 3.4 MW WT3 turbine, even if slightly higher for some figures, are also in reasonable agreement with the US reference. For the offshore case, a bottom-fixed 5 MW machine is compared to the 10 MW used in the present study. Larger differences are found here, for instance in the OPEX costs, due to the very different rating of the two turbines, although the LCOEs are relatively similar.

Table 4. Principal characteristics of the three reference turbines.

Turbine	WT1	WT2	WT3
IEC class and category	1A	2A	3A
Rated electrical power [MW]	10	2.2	3.4
Type	Offshore	Onshore	Onshore
Rotor diameter [m]	178.3	92.4	130.0
Specific power [W/m^2]	400.5	298.3	252.4
Hub height [m]	119.0	80.0	110.0
Blade mass [t]	42.5	8.6	16.4
Tower mass [t]	628	125	553

3.2 Assessment of potentially exploitable design margins

The present study considers a reduced set of DLCs (IEC, 2005), which are responsible for generating the design drivers of these machines (Bottasso et al., 2016; Bortolotti et al., 2016, 2019). The set includes power production with normal turbulence (DLC 1.1 and DLC 1.2), extreme turbulence (DLC 1.3), loss of electrical network in normal turbulence (DLC 2.1), and with extreme operating gusts (DLC 2.3). Additionally, parked conditions are also considered in yaw misalignment (DLC 6.1), with grid loss (DLC 6.2) and with extreme yaw misalignment (DLC 6.3).

3.2.1 Tower

A first analysis of the design-driving key quantities and constraints of the three towers unveils a significant potential that could be exploited by LAC.

For the design constraint analysis, several cross sections are considered along the tower height, where three local conditions are evaluated: buckling, ultimate strength based on von Mises stresses, and fatigue damage. Additionally, the placement of the first fore–aft and side–side frequencies is constrained to avoid crossing the one per rev at rated rotor speed.

For simplicity of discussion, only results at the tower top and bottom cross sections are shown in Fig. 2, where the constraint margins are displayed. These are formulated as the relative difference between the local conditions and their admissible values. A null value therefore indicates an active constraint, whereas a positive value indicates a slack condition, i.e., a constraint that is satisfied but inactive.

Considering first the tower top section, Fig. 2a shows that at this location the towers of WT1 and WT3 are driven by

Table 5. Cost breakdown of the different reference models expressed in 2017 USD.

Cost [USD/kW]	Onshore			Offshore	
	Stehly et al. (2017)	WT2	WT3	Stehly et al. (2017)	WT1
Rating [MW]	2.32	2.2	3.4	5	10
CAPEX [USD/kW]	1454	1297	1759	3846	4379
OPEX [USD/kW]	43.6	48.1	51.4	144	225
AEP [MWh/MW]	3633	3520	3866	3741	4500
FCR [%]	7.9	7.9	7.9	7.0	7.0
LCOE [USD/MWh]	43.6	42.9	49.2	110.5	118.1

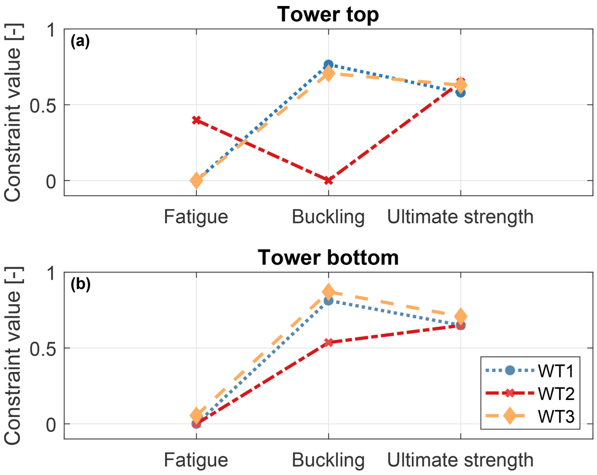


Figure 2. Design constraints at tower top **(a)** and tower bottom **(b)**.

fatigue, whereas buckling and strength are well below their maximum allowed values. The design of this section can therefore benefit from reductions in fatigue damage, which is mostly produced by the modifiable DLC 1.2 (power production in normal turbulence). On the other hand, the upper section of the WT2 tower is driven by buckling, whereas fatigue damage and ultimate strength are inactive. The PEM at this position along the tower is related to the combined bending moment at tower top (CBMTT). The rankings of this key quantity for the three turbines are shown in Fig. 3a. All values are normalized with respect to the leader, and, for clarity, only the leading and first blocking DLCs are shown. The ranking for WT2 is led by DLC 1.3, a modifiable DLC. The first blocking DLC is 2.1, which appears at position 28 in the ranking, leading to a PEM of about 20 %.

Considering the tower bottom cross section, Fig. 2b indicates that all three towers are driven by fatigue. Load rankings for combined bending moment at tower bottom (CBMTB) are reported in Fig. 3b. Results show no potential reduction for the extreme-load constraints, since the load rankings of the WT1 and WT2 towers are led by blocking DLCs. A PEM of about 21 % is present for the WT3 tower, which, however, cannot be exploited since extreme loads do not drive the design at this section.

3.2.2 Rotor

Rotor design constraints include limits on the placement of the lowest natural frequencies to avoid resonant conditions, as well as a safe clearance with respect to the tower. Additionally, several cross sections are considered along the length of the blade, where upper limits for strains, stresses, and fatigue damage are prescribed on the spar caps, shell, and shear webs. An excerpt from this extensive set of constraints is shown in Fig. 4; the shell, spar cap, and shear web constraints are shown only at the midspan section of the blade, for simplicity of illustration.

The spar caps are the components that play the largest role in dictating the overall blade mass, as they mainly provide the blade flapwise bending stiffness. The design of these elements is driven by the blade-tower clearance constraint, which limits the maximum blade tip displacement (Fig. 4a). On the other hand, stress, strain, and fatigue constraints are all inactive (Fig. 4b). The tip displacement rankings, shown in Fig. 5a, indicate a significant reduction potential for all turbines, since they are all led by modifiable DLCs. This key quantity for all three turbines is first blocked by DLC 2.1, leading to PEMs between 8 % (WT1, ranking position 7) and 21 % (WT2, ranking position 28).

The sizing of the shell is mainly driven by the fatigue damage constraint (Fig. 4c). This is also the main driver in the design of the shear webs, which are elements made of sandwich panels that carry shear. Fatigue damage is driven by the modifiable DLC 1.2. However, here the reduction potential is limited by technological constraints that bound from below the thickness of these elements. The load ranking of the combined blade root moment (CBRM) is shown in Fig. 5b, highlighting potential reductions. Indeed, all turbines are again first blocked by DLC 2.1, with large PEMs for WT2 (25%, ranking position 2) and WT3 (30 %, ranking position 3).

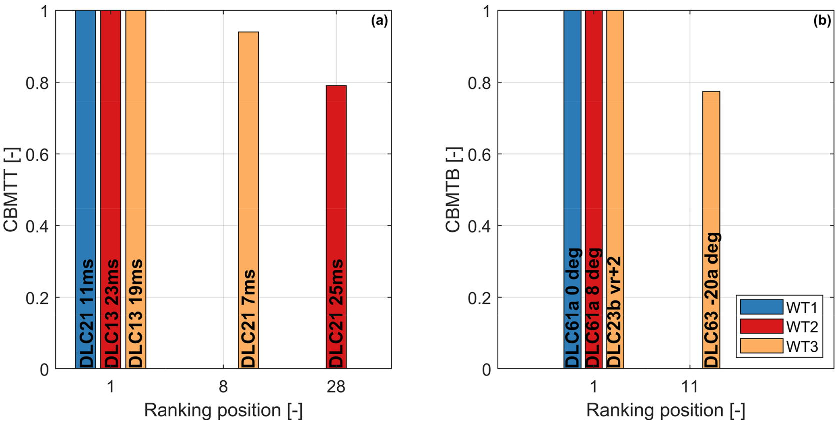


Figure 3. Ranking of normalized combined bending moment at tower top (CBMTT) **(a)** and tower bottom (CBMTB) **(b)**, for the three turbines. Only the leading and first blocking DLCs are shown.

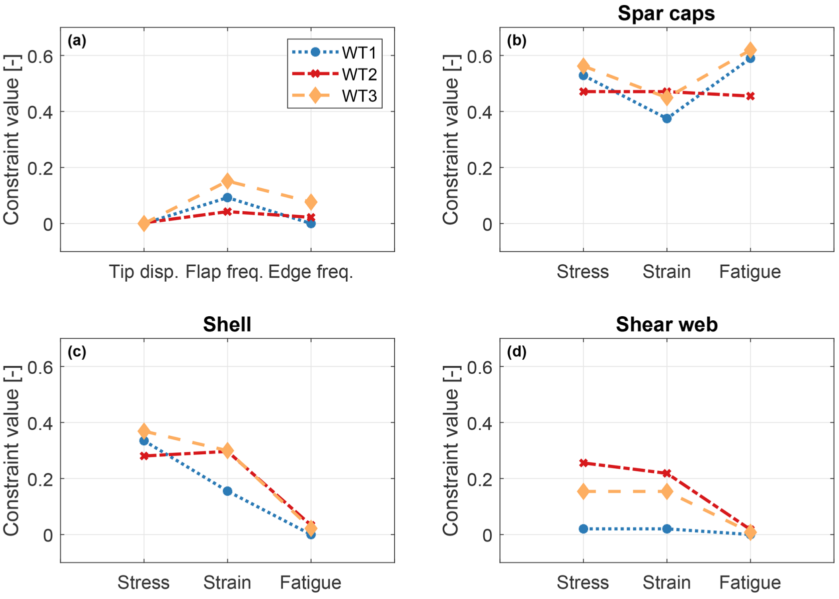


Figure 4. Rotor design constraints for tip displacement and frequency placement **(a)**. For a midspan section of the blade, design constraints at the spar caps **(b)**, shell **(c)**, and shear webs **(d)**.

3.3 Estimated benefits through structural redesign with LAC

This section aims at quantifying the benefits of integrating LAC within the design of the blade and tower of the three reference wind turbines. To this end, the rotor and tower of each turbine are reoptimized, considering loads and elastic deflections as reduced by the coefficients of the load-reduction model (Table 2) and the additional optimistic (values incremented by 50 %) and pessimistic (values reduced by 50 %) scenarios. The economic evaluation is performed as indicated in Sect. 2.4, considering a fixed change rate (FCR) of 7 %. It is further assumed that two lidar scanners have to be purchased over a turbine lifetime of 20 years. This results in an additional EUR 100 000 of ICC. Furthermore, the AOE includes an additional EUR 2500 per year of lidar O&M cost. These costs have been estimated based on input from two major lidar manufacturers and only include hardware-related costs. Due to a lack of information, the costs of development or licensing of LAC control software, related commissioning, and software maintenance have been neglected.

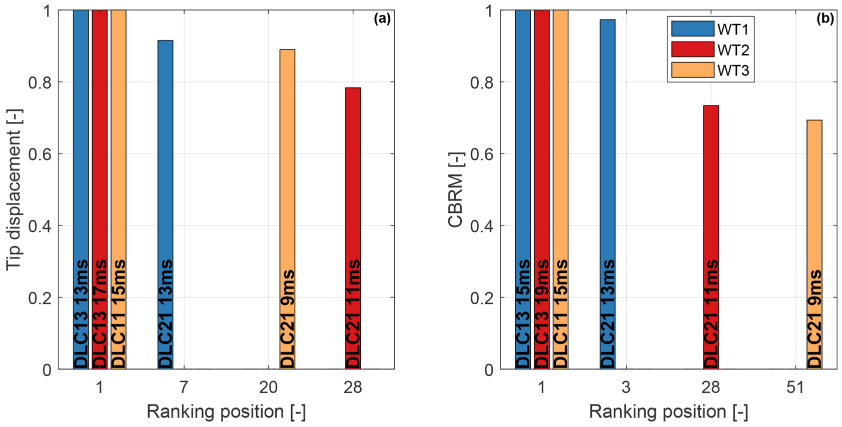


Figure 5. Ranking of normalized blade tip displacement **(a)** and combined blade root moment (CBRM) **(b)** for the three turbines.

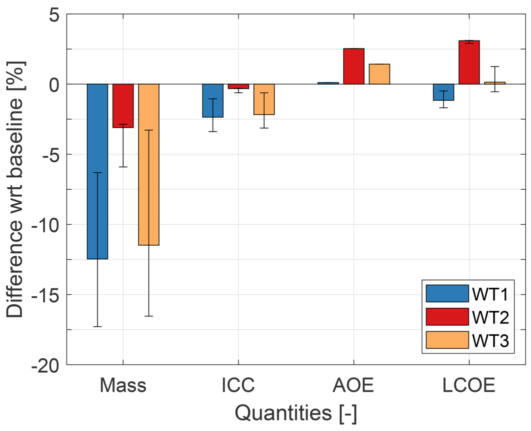


Figure 6. Effects of LAC on the redesign of the tower with respect to the initial baselines. Solid bars: load-reduction model of Table 2; whiskers: range of the pessimistic and optimistic scenarios.

3.3.1 Tower redesign

Figure 6 reports changes in the LAC-based redesigned towers with respect to the initial baselines, when the tower height is held fixed. The solid color bars correspond to the nominal load-reduction model, while whiskers indicate the effects of considering the pessimistic and optimistic scenarios. To ensure direct comparability with the baselines, the redesigned towers are considered to be made of several thin-walled tubular tapered steel sections. Additional geometric constraints to ensure realistic tower shapes are also considered.

Both towers of WT1 and WT3 enjoy significant benefits from large reductions in fatigue damage, which decrease mass between 5 % for the pessimistic scenario and 17 % for the optimistic one. In turn, the lighter weight induces significant reductions in the ICC of both turbines. On the other hand, annual operating expenses (AOE) show a different behavior. Indeed, the additional expenses generated by the maintenance of a lidar system do not significantly add to the already high O&M costs of the offshore turbine WT1. For the onshore machines WT2 and WT3, where these costs play a larger role, AOE increases by approximately 2 %. For all turbines, AEP is essentially unaffected. In the end, the combination of these various effects produces a reduction in LCOE of about 1.2 % for WT1 and a very slight increase of 0.1 % for this same figure of merit for WT3 (Fig. 6).

The WT2 tower presents a different trend. Indeed, the upper segment of this tower is driven by buckling, and CBMTT presents a significant PEM of about 20 % (see Fig. 3a). However, this PEM cannot be exploited, since the LAC load-reduction model (Table 2) does not affect extreme loads at tower top. As a consequence, the redesign is only capable of a limited mass reduction that, in combination with the significant lidar costs, leads to an increase in LCOE.

3.3.2 Taller tower redesign

Instead of reducing tower mass (and hence cost), LAC-enabled improvements in fatigue damage and ultimate loads can be exploited to design taller towers. In fact, by reaching higher above the ground, the rotor is exposed to faster wind speeds, thus increasing AEP; thanks to LAC, this can be achieved without significantly increasing the cost of the tower. To explore the effects of this concept, towers of increasing heights were designed. The study assumes that LAC performance does not depend on tower height. To ensure direct comparability, the redesigned towers are also considered to be made of several thin-walled tubular tapered steel sections. The corresponding geometrical constraints are therefore also included in the redesign problem.

The study is here performed in two steps. First, the tower structure is sized with a non-LAC controller for a given height. The design objective is minimum mass, constrained to guarantee structural integrity. Next, the resulting tower design is reoptimized considering the different scenarios of the LAC load-reduction model, exploiting the slack that it generates in some design-driving constraints. The procedure is repeated for increasing tower heights until no further improvements are possible or an upper limit of 15% height increase with respect to the baseline is reached.

The effects on mass, ICC, AEP, AOE, and LCOE for the three reference machines are reported in Fig. 7.

Different trends are observed for the three turbines. The tower of the offshore machine shows a large potential: for each of the analyzed heights, mass reductions with respect to the non-LAC configuration always translate into decreases in ICC. At the same time AEP increases, whereas AOE remains mostly constant due to the already high O&M costs. LCOE decreases gradually as tower height is increased. However, most of the gains are already achieved for a height increase of 5 %, which is associated with an LCOE decrease of about 1.5 % (Fig. 7e).

An opposite trend is obtained with the tower of WT2: because of its different design drivers, this machine does not benefit from a taller tower, as already noted in Sect. 3.2.1. However, the trend indicates that some LCOE improvements might be possible for very tall towers, which were, however, deemed unrealistic past the upper bound of a 15 % height increase.

Similarly, a taller tower appears not to be very promising even for the onshore fatigue-driven WT3 turbine, although for different reasons. Here, although a 5 % height increase lowers tower mass and ICC and improves AEP by about 2 %, these benefits are offset by an increase in AOE, resulting in marginal – if not completely negligible – benefits in LCOE.

3.3.3 Tower redesign for longer lifetime

Instead of aiming for less expensive or taller towers, as done so far, yet another way to try and exploit the load benefits brought by LAC is to extend the tower lifetime. In this case, the baseline towers are first designed for a 20-year lifetime based on the key quantities resulting from a non-LAC controller. Here again, the towers are redesigned for increasing lifetime in two steps. First, the tower structure is sized with a non-LAC controller for a given lifetime. Next, the resulting tower is reoptimized based on key quantities modified by the LAC load-reduction model (Table 2). WT2 is excluded from this analysis, because of the very limited relevance of fatigue in the sizing of its tower, as shown earlier. To ensure direct comparability with the baseline, the redesigned towers are considered to be made of several thin-walled tubular tapered steel sections, and the corresponding geometrical constraints are included in the sizing.

The tower mass of both WT1 and WT3 increases substantially when sizing for a longer lifetime without using LAC. This negative effect is very nicely counteracted by the use of LAC. Figure 8 reports mass changes generated by LAC for increasing lifetime; all results are computed with respect to initial non-LAC 20-year baselines. At a lifetime of 40 years, which is double the conventional life duration, the tower mass of WT1 is still 10 % lower than for the non-LAC 20-year case. The effect is similar, although a bit less pronounced, even for WT3: for a lifetime of 40 years with LAC, this tower has in fact nearly the same mass of the 20-year non-LAC design.

It should be remarked that these trends are obtained under the assumption of a 100 % lidar availability; additionally, because of the approximations implicit in the assumed load-reduction model, these results can only be regarded as preliminary rough trends. However, the use of LAC to design towers with longer lifetimes seems to be much more promising than the alternative strategies of aiming for reduced costs or improved AEP by taller towers. Indeed, the trends shown here are in line with the results reported in Schlipf et al. (2018), which estimated a 15-year extended lifetime for a tower without redesign. Additionally, since the tower cost plays a large role in ICC, reductions in LCOE could be expected by the installation of towers with a longer lifetime. Alternatively, the towers could be reused to support more modern rotor-nacelle assemblies, playing the role of long-term support structures that do not necessarily have to be upgraded at the same pace of the rest of the turbine.

3.3.4 Rotor redesign

Only rather modest mass reductions are achieved for the blades of all models and for all scenarios, due to the moderate influence of LAC in design-driving constraints. The situation is more precisely illustrated by Fig. 9, which shows the largest improvements for WT1 and essentially no effect for WT2.

Indeed, the LAC load-reduction model reported in Table 2 shows a larger effect of LAC in fatigue damage mitigation than in the reduction of ultimate loads and deflections. Although shell and shear webs are both driven by fatigue, they are already thin structures with limited reduction potential before technological constraints on their thickness become active. In turn, this leads to the fatigue PEMs not being fully exploited. The design of the spar caps is also not significantly affected by LAC. In principle, a significant PEM is present for tip deflection, but unfortunately here again the LAC load-reduction model has only modest 2 % improvements for this key quantity. Additionally, as previously noted, the exploitation of the reduction of an ultimate condition by LAC raises important issues related to safety and might imply drastically increased costs to ensure redundancy.

For all three turbines, the reduction in ICC generated by the use of LAC in the redesigned rotors is not significant enough to compensate for the increase in AOE. Therefore, LCOE increases for all onshore machines and decreases in a negligible way for the offshore turbine.

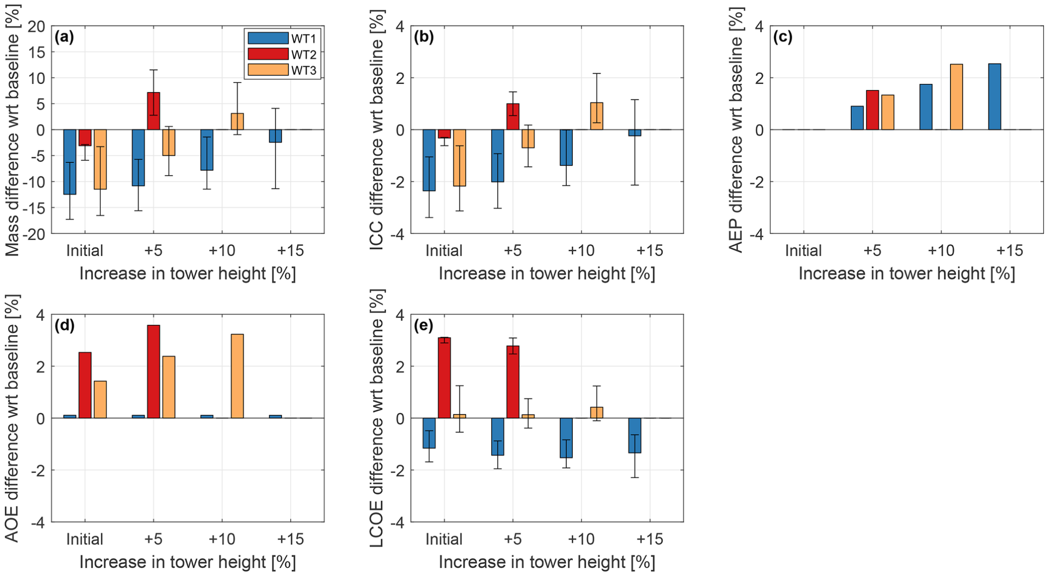


Figure 7. Effects of LAC on the redesign of towers of increasing height with respect to the initial non-LAC baselines. Solid bars: load-reduction model of Table 2; whiskers: range of the pessimistic and optimistic scenarios. The study considers increments of +5 %, 10 %, and 15 % in tower height for WT1; an increment of 5 % in tower height for WT2; and increments of 5 % and 10 % in tower height for WT3.

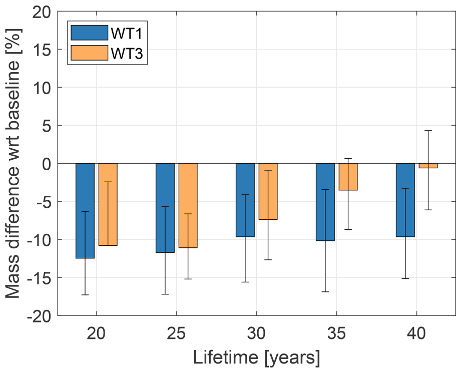


Figure 8. Effects of LAC on the redesign of towers of increasing lifetime with respect to 20-year non-LAC baselines. Solid bars: load-reduction model of Table 2; whiskers: range of the pessimistic and optimistic scenarios.

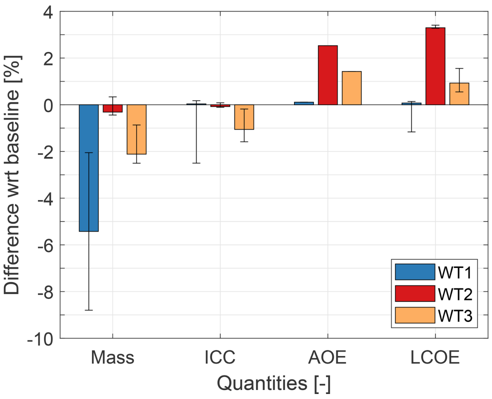


Figure 9. Effects of LAC on the redesign of the rotor with respect to the initial baselines. Solid bars: load-reduction model of Table 2; whiskers: range of the pessimistic and optimistic scenarios.

3.4 Cost sensitivity analysis

Finally, a sensitivity analysis is performed to understand to what extent the purchase and maintenance costs of a lidar system can influence the reduction in LCOE. Baseline values of EUR 100 000 and EUR 2500 per year, respectively for purchase and maintenance, are gradually modified until reaching the limit of ±100 % variations. It is assumed that lidar-related yearly maintenance costs are constant throughout the wind turbine lifetime and are therefore not affected by external factors, such as the replacement of the lidar system. Purchase price includes both the cost and the number of lidar systems required throughout the wind turbine lifetime. The analysis considers the nominal LAC load-reduction model of Table 2 applied only to WT1 and WT3, as WT2 did not seem to have any real potential for improvement. Clearly, redundancy to ensure safety would significantly increase all of these costs.

It should be noticed that purchase and maintenance costs are treated here as two independent variables. In reality, purchase price could be correlated with performance, and therefore it might affect load reductions. Additionally, purchase price could be correlated with maintenance: a higher cost of the lidar could imply a more sophisticated device, which might be more costly to maintain, but it could also be correlated with build quality, which then might be inversely related to maintenance cost. Such considerations would require a sophisticated cost model of the lidar, which was, however, unfortunately not available for this research. The present analysis, being based on the simple change of the two independent quantities, purchase and maintenance costs, could then be interpreted as a price positioning study, where the lidar manufacturer tries to understand the correct price range for the device to make it appealing to customers.

Figure 10a shows that only a modest effect in LCOE can be achieved for WT1 when purchase and maintenance costs are modified. On the other hand, an-order-of-magnitude-larger effect is observed for WT3 (Fig. 10b), where the incidence of the lidar-associated costs is more prominent given the smaller size and rating of this turbine.

Break-even is indicated in both figures as a dotted line, located in the white area that separates reductions (blue) from increments (red) in LCOE. The break-even line is almost perpendicular to the purchase cost axis, implying a large sensitivity of LCOE to this quantity. The figure shows that reductions in purchase costs appear more effective than reductions in O&M costs. This seems to indicate that lidar manufacturers should try to keep the cost of the device as low as possible. The fact that maintenance costs are less relevant might indicate that simple and cheap lidars – although possibly a bit more expensive to maintain – would be more appealing than sophisticated but expensive ones. Cheap single units, as long as availability remains sufficiently high, might also be very interesting from the point of view of redundancy, which might open up the possibility of exploiting ultimate load reductions. However, as noticed earlier, more sophisticated models – capable of capturing the couplings among purchase price, performance (including availability), lifetime, and maintenance – would be necessary to identify economically optimal development strategies for lidar systems.

Overall, results indicate that only modest reductions in LCOE are possible, even with very low LAC-based costs.

4 Conclusions

This paper has presented a preliminary general analysis on the potential benefits of integrating LAC within the design of the rotor and tower of a wind turbine. The design was performed as a constrained optimization based on aeroelastic simulations, conducted in close accordance with international design standards.

The benefits generated by the use of a lidar for controlling a turbine were quantified through a load-reduction model sourced from the literature, considering an average performance of the lidar-assisted controller and additional pessimistic and optimistic scenarios. This approach, in contrast to the use of an actual lidar-assisted controller in the loop, was chosen in order to draw conclusions on general trends rather than on the effects of a specific LAC implementation. Realizing that any such redesign exercise is typically highly problem-specific, the study was conducted considering three representative turbines of different class, size, and rating.

Based on the results of this study, the following conclusions can be drawn.

First, a significant improvement potential was observed when the design is driven by fatigue. Indeed, fatigue damage is primarily generated in power production in turbulent wind conditions. Here, the lidar-generated preview of the wind that will shortly affect the rotor is clearly beneficial: as the controller “sees” what will happen, it can anticipate its action. This is in contrast to the case of a pure feedback controller that, since it can only operate in response to a phenomenon that has already taken place, is by definition late in its reaction. In turn, the lidar preview information leads to a general reduction of load fluctuations and hence of fatigue damage.

On the contrary, the improvement potential is only very limited for components driven by ultimate conditions (such as maximum stresses, strains, or blade tip deflection). Indeed, these ultimate conditions cannot always be modified by LAC. In addition, even when LAC plays a role, other factors may have an even larger effect; for example, this is the case of shutdowns, where the pitch-to-feather policy may have a dominant role in dictating the peak response. But even when LAC does improve design-driving ultimate conditions, an even more general question still remains: shall one design a component based on an ultimate condition that was reduced by LAC? If so, what are the extra precautions that should be taken in order to hedge against faults, inaccuracies, misses, or unavailability of the lidar? These issues were not considered here, which is a limitation of the present study. However, it is possible that – at least in some of the cases analyzed in this work – the improvements to ultimate conditions brought by LAC would have to be completely discarded when these additional aspects are considered or where extra costs have to be added, for example, to ensure redundancy by the use of multiple lidars.

It was also found that, for fatigue-driven towers, significant benefits in mass (on average equal to about 12 %, for the cases considered here) can be obtained by the use of a LAC controller. However, these benefits are largely diluted by looking at the more general metric LCOE. In fact, only a large offshore machine showed improvements for this figure of merit: since O&M costs are already high for an offshore turbine, the extra costs due to the lidar play a lesser role. For smaller turbines the situation is different, and the benefits in mass do not repay for the costs of the lidar.

Instead of simply reducing mass, LAC can be used to either increase hub height (which increases power capture in sheared inflow) or to extend lifetime. Both approaches were considered here. The most interesting results were again ob-

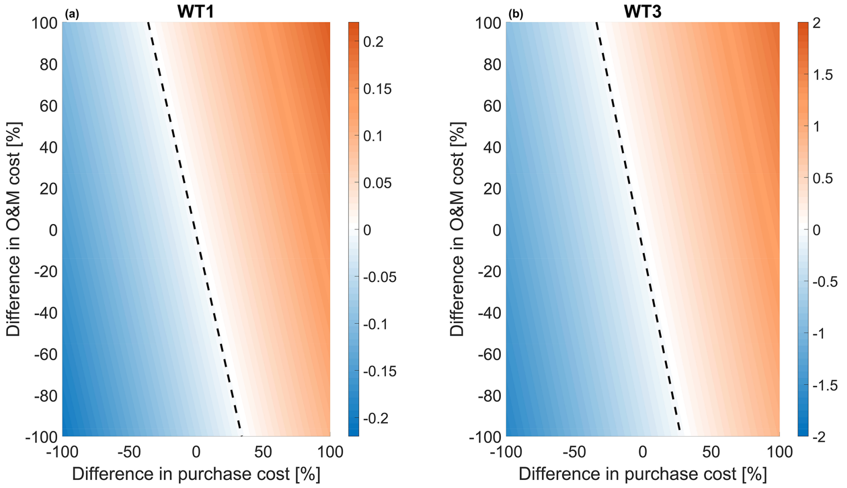


Figure 10. Percent variation of LCOE (shown in the color bars) as a function of purchase and O&M costs of LAC systems for the offshore machine WT1 **(a)** and the onshore machine WT3 **(b)**.

tained for fatigue-driven offshore towers. Indeed, a 15 % taller tower was found to present approximately the same mass of the baseline, but with a 2% higher AEP. Even more interestingly, a LAC-enabled tower was designed with double the lifetime and 10 % less mass than the baseline.

The situation for the rotor is less promising. In principle, spar caps – which are the main contributors to blade mass – could greatly benefit from LAC when tip deflection is the main driver. Here again lidar preview can clearly help when maximum deflections are triggered by strong wind gusts. On the other hand, stiffness requirements caused by the placement of the flap frequency can substantially reduce this margin of improvement, as this is a blocking effect. Additionally, one would have again to guarantee that the safety-critical tip clearance constraint is always satisfied during operation, which might require redundancy of the lidar or other safety measures. Shear webs and shell are often driven by fatigue, a condition that could in principle be exploited by LAC. However, the improvement potential is limited due to the already limited thickness of these components. In summary, the integration of LAC into the design of the rotor does not seem to lead to significant benefits in terms of LCOE.

Finally, a simple parametric study on the purchase and O&M costs of a lidar system was performed. As previously observed, the study shows that LCOE is largely independent from the LAC purchase and O&M costs in the offshore case. Although a larger effect is visible in the onshore case, improvements in LCOE caused by reductions in the lidar costs are still quite modest. This might indicate that, instead of targeting price reductions, lidar research and development should focus on performance. On the other hand, significant price reductions might allow for redundancy, which in turn would enable the targeting of drivers based on ultimate conditions.

The present work is based on a number of assumptions, and further work should be performed before more definitive conclusions can be drawn. First, only three turbines were considered; although these machines are reasonable approximations of contemporary products, it is clear that design drivers are typically turbine specific, and a more ample range of cases should be investigated. Additionally, only the conventional configuration of thin-walled steel towers with circular tubular tapered sections was considered. This configuration presents important geometric constraints that impact the benefits of LAC. Second, there was no attempt here to consider lidar availability, faults, and possible redundancy; an analysis of these aspects would help in clarifying whether LAC-enabled reductions in ultimate conditions can indeed be exploited in the structural redesign of the blade and the tower or not. Finally, it should be remarked that the use of a generic load model implies some significant approximations. Although this was done here on purpose with the goal of making the study more general, it is also clear that the performance of different LAC systems can be very different, depending on the lidar characteristics and on the controller formulation and tuning. Therefore, here again, more specific studies based on fully coupled simulations should be performed to further explore the trends reported here and find additional niches of applicability of LAC missed by the present general analysis.

Notwithstanding the limitations of this study, in the end it appears that the answer to the question of whether LAC is beneficial or not might not be so clear-cut, and in reality the situation is much more complex and varied (and also interesting). In hindsight, this is also a useful reminder that apparently obvious improvements do not always necessarily translate into real system-level benefits. For example, reducing some loads might be irrelevant if the design is driven by other factors or might not pay off if the cost of that reduction neutralizes its benefits. This also stresses once more the central importance of systems engineering and design for the understanding of the true potential of a technology.

Code and data availability. The data used for the present analysis can be obtained by contacting the authors.

Author contributions. HC performed the analysis on potentially exploitable margins and conducted the design studies; SL prepared the lidar load-reduction model and assisted in the application of the model in the design framework; CLB formulated the analysis methodology based on the new concept of potentially exploitable margins, proposed the use of a generic lidar load-reduction model, and supervised the work; HC and CLB wrote the paper. All authors provided important input to this research work through discussions, through feedback, and by improving the manuscript.

Competing interests. The authors declare that they have no conflict of interest.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Acknowledgements. The authors acknowledge the participants of the IEA Task 32 and 37 workshop "Optimizing Wind Turbines with Lidar-Assisted Control using Systems Engineering" for the valuable discussions.

Review statement. This paper was edited by Sandrine Aubrun and reviewed by two anonymous referees.

References

Bak, C., Zahle, F., Bitsche, R., Kim, T., Yde, A., Henriksen, L. C., Natarajan, A., and Hansen, M. H.: INNWIND. EU Deliverable D1.21: Reference Wind Turbine Report, available at: http://www.innwind.eu/publications/deliverable-reports (last access: 18 December 2019), 2013.

Bortolotti, P., Bottasso, C. L., and Croce, A.: Combined preliminary–detailed design of wind turbines, Wind Energ. Sci., 1, 71–88, https://doi.org/10.5194/wes-1-71-2016, 2016.

Bortolotti, P., Canet, H., Dykes, K., Merz, K., Sethuraman, L., Verelst, D., and Zahle, F.: IEA Wind TCP Task 37: Systems Engineering in Wind Energy – WP2.1 Reference Wind Turbines Technical Report, International Energy Agency, Golden (CO), https://doi.org/10.2172/1529216, 2019.

Bossanyi, E., Kumar, A., and Hugues-Salas, O.: Wind turbine control applications of turbine-mounted lidar, J. Phys. Conf. Ser., 555, 012011, https://doi.org/10.1088/1742-6596/555/1/012011, 2014.

Bottasso, C. L., Campagnolo, F., and Croce, A.: Multi-disciplinary constrained optimization of wind turbines, Multibody Syst. Dyn., 27, 21–53, https://doi.org/10.1007/s11044-011-9271-x, 2012.

Bottasso, C. L., Pizzinelli, P., Riboldi, C. E. D., and Tasca, L.: LiDAR-enabled model predictive control of wind turbines with real-time capabilities, Ren. En., 71, 442–452, https://doi.org/10.1016/j.renene.2014.05.041, 2014.

Bottasso, C. L., Bortolotti, P., Croce, A., and Gualdoni, F.: Integrated aero-structural optimization of wind turbines, Multibody Syst. Dyn., 38, 317–344, https://doi.org/10.1007/s11044-015-9488-1, 2016.

Bottasso, C. L. and Bortolotti, P.: Rotor Design and Analysis, Wind Energy Modeling and Simulation: Turbine and System, edited by: Veers, P., ISBN 978-1785615238, The Institution of Engineering and Technology (IET), London, UK, 2019.

Canet, H., Loew, S., and Bottasso, C. L.: Lidar-assisted control in wind turbine design: Where are the potential benefits?, J. Phys. Conf. Ser., 1618, 042020, https://doi.org/10.1088/1742-6596/1618/4/042020, 2020.

Chaviaropoulos, P., Karga, I., Harkness, C., and Hendriks, B.: Deliverable 1.23 PI-Based assessment of innovative concepts INNWIND.EU technical report, available at: http://www.innwind.eu/ (last access: 6 October 2021), 2014.

Dunne, F., Pao, L. Y., Wright, A. D., Jonkman, B., and Kelley, N.: Adding feedforward blade pitch control to standard feedback controllers for load mitigation in wind turbines, Mechatronics, 21, 682–690, https://doi.org/10.1016/j.mechatronics.2011.02.011, 2011.

Dunne, F., Schlipf, D., Pao, L. Y., Wright, A. D., Jonkman, B., Kelley, N., and Simley, E.: Comparison of two independent lidar-based pitch control designs, Proc. AIAA Aerospace Sciences Meeting, Nashville, Tennessee, 9–12 January 2012, https://doi.org/10.2514/6.2012-1151, 2012.

EN 1993-1-9: Eurocode 3: Design of steel structures – Part 1–9: Fatigue, CEN, Brussels, 2006.

Fingersh, L., Hand, M., and Laxson, A.: Wind Turbine Design Cost and Scaling Model, National Renewable Energy Laboratory technical report, Golden, CO, available at: https://www.nrel.gov/docs/fy07osti/40566.pdf (last access: 20 February 2020), 2006.

Giavotto, V., Borri, M., Mantegazza, P., and Ghiringhelli, G.: Anisotropic beam theory and applications, Comput. Struct., 16, 403–413, https://doi.org/10.1016/0045-7949(83)90179-7, 1983.

Griffith, D. T. and Johans, W.: Large blade manufacturing cost studies using the sandia blade manufacturing cost tool and Sandia 100 m blades, Sandia National Laboratories technical report, Albuquerque, NM, available at: https://energy.sandia.gov/wp-content/gallery/uploads/dlm_uploads/SAND_SNLLargeBladeManufacturingCostTrendsAnalysis_SAND2013-2734.pdf (last access: 17 February 2020), 2013.

Haizmann, F., Schlipf, D., Raach, S., Scholbrock, A., Wright, A., Slinger, C., Medley, J., Harris, M., Bossanyi, E., and Cheng, P. W.: Optimization of a feed-forward controller using a CW-lidar system on the CART3, Proceedings of the American Control Conference, Chicago, USA, 1–3 July 2015, https://doi.org/10.18419/opus-3975, 2015.

Hendriks, H. B. and Bulder, B. H.: Fatigue Equivalent Load Cycle Method: A general method to compare the fatigue loading of different load spectrums, Energy Research Centre of the Netherlands technical report, Netherlands, available at: https://publicaties.ecn.nl/PdfFetch.aspx?nr=ECN-C--95-074 (last access: 24 September 2020), 1995.

International Electrotechnical Commission: International Electrotechnical Commission, IEC 61400-1 Ed. 3: Wind turbines – Part 1: Design requirements, IEC, Geneva, Switzerland, 2005.

Koren, I. and Krishna, C. M.: Fault-Tolerant Systems, 2nd Edition, ISBN 978-0-12-818105-8, https://doi.org/10.1016/C2018-0-02160-X, Elsevier, Amsterdam, the Netherlands, 2020.

NREL: WISDEM, v2.1.5, GitHub [code], available at: https://github.com/WISDEM/WISDEM, last access: 8 July 2020.

Riboldi, C. E. D.: Advanced Control Laws for Variable-Speed Wind Turbines and Supporting Enabling Technologies, Ph.D. thesis, Politecnico di Milano, Milan, Italy, 2012.

Schlipf, D., Fleming, P., Kapp, S., Scholbrock, A., Haizmann, F., Belen, F., Wright, A., and Cheng, P. W.: Direct speed control using lidar and turbine data, American Control Conference 2013, Washington, DC, USA, 17–19 June 2013, https://doi.org/10.1109/ACC.2013.6580163, 2013a.

Schlipf, D., Schlipf, D. J., and Kühn, M.: Nonlinear model predictive control of wind turbines using lidar, Wind Energ., 16 1107–1129, 2013b.

Schlipf, D., Fleming, P., Haizmann, F., Scholbrock, A., Hofsäß, M., Wright, A., and Cheng, P. W.: Field Testing of Feedforward Collective Pitch Control on the CART2 Using a Nacelle-Based Lidar Scanner, J. Physics, 555, 012090, https://doi.org/10.1088/1742-6596/555/1/012090, 2013c.

Schlipf, D. and Cheng, P. W.: Flatness-based Feedforward Control of Wind Turbines Using Lidar, IFAC Proceedings Volumes, 47, 5820–5825, https://doi.org/10.3182/20140824-6-ZA-1003.00443, 2014.

Schlipf, D., Simley, E., Lemmer, F., Pao, L., and Cheng, P. W.: Collective pitch feedforward control of floating wind turbines using lidar, Journal of Ocean and Wind Energy, 2, 223–230, https://doi.org/10.17736/jowe.2015.arr04, 2015.

Schlipf, D.: Lidar-Assited Control Concepts for Wind Turbines, Ph.D. Thesis, University of Stuttgart, Dr.Hut editions, Munich, Germany, 2016.

Schlipf, D., Fürst, H., Raach, S., and Haizmann, F.: Systems Engineering for Lidar-Assisted Control: A Sequential Approach, J. Phys. Conf. Ser., 1102, 012014, https://doi.org/10.1088/1742-6596/1102/1/012014, 2018.

Scholbrock, A., Fleming, P., Wright, A., Wang, N., Schlipf, D., and Johnson, K.: Lidar-Enhanced Wind Turbine Control: Past, Present and Future, NREL/CP-5000-65879, NREL, Golden, CO, USA, available at: https://www.nrel.gov/docs/fy16osti/65879.pdf (last access: 6 October 2021), 2016.

Simley, E., Fürst, H., and Schlipf, D.: Optimizing Lidars for Wind Turbine Control Applications – Results from the IEA Wind Task 32 Workshop, Remote Sens., 10, 863, https://doi.org/10.3390/rs10060863, 2018.

Simley, E., Bortolotti, P., Scholbrock, A., Schlipf, D., and Dykes, K.: IEA Wind Task 32 and Task 37: Optimizing Wind Turbines with Lidar-Assisted Control Using Systems Engineering, J. Phys. Conf. Ser., 1618, 042029, https://doi.org/10.1088/1742-6596/1618/4/042029, 2020.

Sinner, M. N. and Pao, L. Y.: A Comparison of individual and collective pitch model predictive controllers for wind turbines, Annual American Control Conference (ACC), Milwaukee, WI, 27–29 June 2018, 1509–1514, https://doi.org/10.23919/ACC.2018.8431598, 2018.

Stehly, T., Beiter, P., Heimiller, D., and Scott, G.: Cost of Wind Energy Review, Tech. Rep. (National Renewable Energy Laboratory), available at: https://www.nrel.gov/docs/fy18osti/72167.pdf (last access: 20 September 2019), 2017.

Wang, N., Johnson, K., and Wright, A.: Comparison of strategies for enhancing energy capture and reducting loads using lidar and feedforward control, IEEE T. Contr. Syst. T., 21, 1129–1142, https://doi.org/10.1109/TCST.2013.2258670, 2013.

CHAPTER 6

Discussion & Outlook

In the following, the highlights of the present dissertation are discussed and further linked. On the basis of that, an outlook towards future research directions is provided.

6.1 Discussion

In the present work, the vision has been pursued that devices are aware of their fatigue behavior, and utilize this information to optimally balance revenue and fatigue. Based on this vision, the online estimation of fatigue has been improved, novel formulations for Model Predictive Control (MPC) of fatigue have been developed, and the integrated design scenario has been investigated. This research has led to the following key achievements:

- For the first time, an online fatigue estimation procedure has been presented, which outputs value-continuous fatigue information, and operates on arbitrarily small time steps.
- Also for the first time, a new family of fatigue cost formulations has been presented, which directly implement the standard Rainflow-based fatigue estimation procedure in MPC, and even consider stress information from the past.
- A glimpse of the broad applicability of the MPC formulations has been provided by successfully controlling a high-fidelity wind turbine model, a complex battery model, and a challenging hybrid energy system comprising both device models.
- Particularly, for the wind turbine, a comprehensive controller toolchain has been set up, comprising a Lidar simulator, Lidar data processing, a Moving Horizon Estimator, and a novel Model Predictive Controller.
- The benefit of fatigue control in an integrated wind turbine design scenario has been demonstrated.

Looking at the existing literature, and combining the results of all papers of the present dissertation, the high-level questions can be discussed that have been raised in Section 1.2.2.

How far can wind turbine fatigue be reduced by control, and how much is needed?

For the wind turbine setup in **Paper 7**, a conventional MPC employing stress rate penalization [31,32] has reduced tower base fatigue by about 50% compared to a conventional PID controller [79]. The novel PORFC MPC from the present dissertation can be tuned to achieve the same revenue and a further fatigue reduction of about 5-10%. Note that in **Paper 7** profit-optimality has been the goal, which can lead to less fatigue reduction for the benefit of revenue increase.

Fatigue reduction can be utilized via a retrofit or an integrated design scenario [27], as explained in Section 1.1.4. In both scenarios, fatigue reduction can be converted to lifetime extension of the respective component, if the market setting allows for that. For instance, if all regions of the tower experience at least 50% of fatigue reduction, inversely, lifetime can be doubled from 20 to 40 years. This considerable temporal order of magnitude is still in line with the literature, where lifetime extensions of up to 15 years have been considered [80].

For the integrated design scenario, in terms of *improvement potential* by control, fatigue has a clear advantage over other design constraints. The reason lies in the cumulative nature of fatigue in contrary to the maximum evaluation of the other design constraints as for instance extreme stresses. As shown in **Papers 10** and **11**, the other design constraints can be governed - or "blocked" - by non-operational Design Load Cases as grid-loss or parked conditions. In these Design Load Cases, typically, the operational turbine controller is not active, and improvements of the controller will not translate to improvement of these design constraints. In contrast, fatigue is mainly driven by the operational Design Load Cases, were the turbine controller has full impact.

In terms of *actual design improvement* for specific wind turbines by control, fatigue also is in a good position. As shown in **Paper 11**, very often fatigue is a main design constraint for the important components tower and blades. For all three turbines of this publication, at the tower bottom, fatigue is the active constraint with significant margin to other constraints of buckling and ultimate strength. Also for blade shell skin and shear web, the fatigue constraint is driving the design. Thus, during the design optimization process, very frequently, fatigue reduction alone leads to significant benefits before the other design constraints become important. And even beyond this point, further fatigue reduction is beneficial in many cases.

To conclude, the novel formulations for Model Predictive Control result in significant further fatigue reduction, which improves lifetime and design of wind turbines to a very large extent.

How can cyclic aging control influence operational profit of batteries, and when is it applicable?

For the battery setup in **Paper 8**, a conventional MPC employing a *piece-wise affine* cycle identification model [35] has decreased total penalty by 10% compared to a conventional rule-based controller. Total penalty is the sum of power mismatch penalty and aging cost, while profit is revenue subtracted by this total penalty. The novel PORFC MPC from the present dissertation has further reduced total penalty by 13%. These benefits of both MPCs originate from accepting some mismatch penalty while significantly reducing cyclic aging.

It should be noted that accepting power mismatch is not possible in every market scenario or energy system setup. For instance in the German frequency regulation market, participating entities are forced to supply exactly the demanded positive or negative power [81]. Consequently, for a single battery there is no flexibility for balancing power tracking with cyclic aging. Instead, in **Paper 8**, the US-American PJM frequency regulation market is targeted, where mismatch is permitted under some constraints [82]. On the other hand, in a market like the German one, cyclic aging control also can be useful if a further flexibility is introduced; for instance in the form of a second power generation unit. This complex setting has been demonstrated in **Paper 9** with a hybrid energy system comprising one wind turbine and one battery energy storage system. Here, a single PORFC MPC has successfully balanced wind turbine tower fatigue and battery cyclic aging, while ensuring a defined power supply to the grid.

How do the simulation setups influence the results?

The wind turbine MPCs have been tested in both ideal and realistic simulation setups.

The core idealizations in **Papers 2**-**5** have been the absence of plant-model mismatch, the measurability of all states, and the perfect wind prediction. On the one hand, this ideal setup has enabled a detailed understanding of the dynamic behavior of the novel MPCs. On the other hand, in the ideal setup, the conventional and novel MPCs have differed only moderately in terms of profit (0.1-1%). The reason is that, with perfect state predictions, all tested MPCs have been able to radically dampen the tower oscillation, while maintaining a similarly high revenue level. Consequently, fatigue cost has been lower than revenue by about two orders of magnitude, and relative differences between the MPCs in terms of fatigue have not translated to large differences in terms of profit.

A very different picture has been revealed for the realistic simulation setups, which have comprised a high-fidelity plant model in **Paper 6**, and, additionally, estimated states and Lidar wind foresight in **Paper 7**. These realistic setups pose significant challenges regarding fatigue reduction for the controllers, leading to revenue and fatigue being in the same order of magnitude. Consequently, their balancing is more important, and the controllers have more opportunities to stand out. Indeed, the novel MPC formulation has been able to achieve impressive profit advantages of 1-5% by smarter control of fatigue.

How and to which level of accuracy do the novel formulations represent fatigue?

The underlying physical mechanisms of mechanical fatigue occur at the microscopic level in the form of micro-fissures, which grow and eventually become macro cracks. Thus, it should be noted that the fatigue estimation procedure, built around Rainflow stress cycle identification, represents only a macroscopic representation of the actual small-scale processes. However, the stress cycles identified by the Rainflow algorithm still correlate to the energy that is injected locally by cyclic loading, and eventually drives the crack growth [56]. Due to this sufficient level of coherence, and due to the computational efficiency, the Rainflow-based procedure is accepted as the de facto standard for fatigue estimation throughout science and industry [56].

The novel one-step fatigue estimation algorithm of **Paper 5** utilizes the Rainflow algorithm and the residue concept without modifications. Thus, no information is lost with respect to the original fatigue estimation procedure, as demonstrated in the same paper.

For the existing and novel MPC fatigue cost formulations of **Papers 2**-**5** and Section 2.2, there is a more diverse picture. As shown in Fig. 6.1, fatigue - or a simplified metric - is integrated at different points in the execution flow of the MPC; either as part of the system dynamics (Hybrid, PWA, TTVP), as part of the MPC algorithm (DORFC, AP), or even externalized from the MPC algorithm (PORFC, TORFC). Consequently, different proxies for cycle identification are employed in the cost functions, leading to different proximity to the original fatigue estimation procedure:

- The MPC formulation employing a hybrid dynamical system (Hybrid) from **Paper 5** is the closest to physical fatigue, since it is based on the above mentioned one-step fatigue estimation algorithm, which enables a continuous fatigue estimation over the prediction horizon.
- The formulation Direct Online Rainflow Counting (DORFC) from **Paper 3** exhibits the same accuracy as the Hybrid formulation, with the only difference of a batch-wise evaluation of the added fatigue over the prediction horizon, instead of a continuous evaluation.
- The formulation Parametric Online Rainflow Counting (PORFC) from **Paper 4** represents a reformulation of DORFC. As stated in the same paper, PORFC exhibits only minor deviation from DORFC in terms of absolute fatigue cost, but loses information in the second derivative (Hessian) due to the decoupling of fatigue in time.
- The formulation Tracking Online Rainflow Counting (TORFC) from **Paper 4** is a reformulation of PORFC, where information is lost by condensing all stress cycle information to a time-varying tracking reference and weight. However, TORFC contains still more fatigue information than the formulation Amplitude Penalization (AP) from **Paper 2**, where a simple constant stress reference is defined by stress mean.
- In contrast, the formulation employing a piecewise-affine cycle identification (PWA) from [35] actually aims at identifying cycles. However, instead of the Rainflow algorithm, a simplified algorithm is utilized, which is less related to actual fatigue.
- Last, the formulation employing stress rate penalization (TTVP) from [31] does not even consider the quantity of stress, but rather stress rate, which has no direct relation to fatigue.

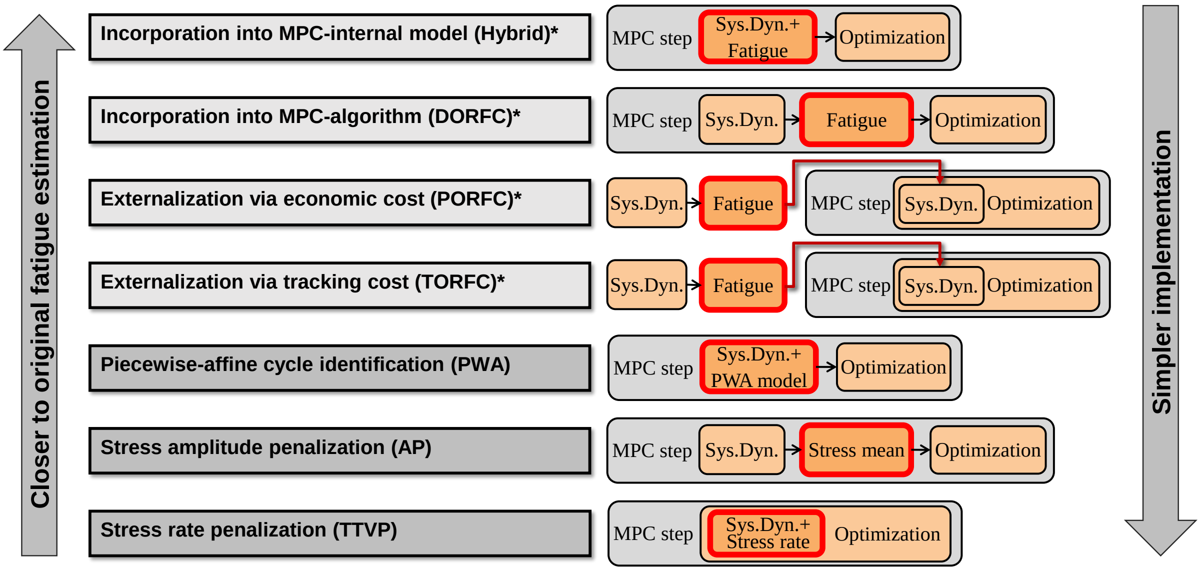

Closer to original fatigue estimation
Incorporation into MPC-internal model (Hybrid)*
MPC step
Sys.Dyn.+ Fatigue
Optimization
Incorporation into MPC-algorithm (DORFC)*
MPC step
Sys.Dyn.
Fatigue
Optimization
Externalization via economic cost (PORFC)*
Sys.Dyn.
Fatigue
MPC step
Sys.Dyn.
Optimization
Externalization via tracking cost (TORFC)*
Sys.Dyn.
Fatigue
MPC step
Sys.Dyn.
Optimization
Piecewise-affine cycle identification (PWA)
MPC step
Sys.Dyn.+ PWA model
Optimization
Stress amplitude penalization (AP)
MPC step
Sys.Dyn.
Stress mean
Optimization
Stress rate penalization (TTVP)
MPC step
Sys.Dyn.+ Stress rate
Optimization
Simpler implementation

Figure 6.1: Comparison of existing and novel (marked by *) fatigue cost formulations, which are closer to the original fatigue estimation procedure towards the top, and simpler to implement towards the bottom. The red box indicates the evaluation of fatigue or a related metric. Sys.Dyn. = solution of the system dynamics. Optimization = solution of the optimization problem.

6.2 Outlook

The novel formulations for estimation and control of fatigue from the present dissertation have closed research gaps, but also have opened up new interesting aspects to be investigated.

Alternatives to minimization of fatigue

The novel MPC formulations perform a minimization of fatigue, prioritized by a fixed weight.

However, so far, the MPC does not evaluate the success of its actions based on actual fatigue at the real plant. This link could be established by employing a high-level controller similar to Do et al [83], which compares the actual fatigue evolution at the plant to a nominal expected evolution, and adapts the fatigue weight of the MPC accordingly. The link to actual plant behavior could also be established by converting the fatigue-minimizing cost function to a fatigue-tracking cost function, where a situation-based acceptable fatigue prognosis is tracked, which is provided by a high-level instance.

Finally, there are market settings, where the device is built already and its lifetime is fixed as well. Thus, the controller simply has to ensure that end-of-life does not occur before end-of-design-life. For this purpose, the fatigue cost function could be reformulated to a nonlinear constraint, continuously keeping fatigue evolution below its nominal trajectory.

Improvements for the controller-internal models

The novel MPC formulations utilize internal system models, which are based on first-principle dynamics and data-driven elements, which have been identified offline. Consequently, the real plant is represented with only limited accuracy. This gap could be closed by utilizing a more sophisticated internal model, or by online model adaptation. A more sophisticated model can be achieved by replacing the first-principle dynamics with reduced-order models, derived from accurate Finite Element

Models [29]. Online model adaptation can be achieved by extending the state estimator by a parameter estimation, based on past measurements from the plant [84]. Alternatively, dedicated error terms can be inserted in the model, and learned online based on measurements from the plant [85].

Additional realism for the assessments

The applications of the present dissertation have demonstrated controller performance in significantly realistic simulation settings. However, clearly, these assessments can be improved further by introducing more uncertainties or objectives in the simulations, or by moving to real test plants.

In the battery context, for instance, uncertain power demand predictions can be introduced. Clearly, the fast battery typically is able to fulfill virtually any realistic power demand immediately. However, the economical balancing of revenue and cyclic aging in the MPC will be challenged by the uncertain demand prediction.

In the wind turbine context, apart from the important tower fore-aft oscillation, the novel fatigue cost formulations can be simultaneously applied to the blades and the drive train. An application to tower side-side oscillation is currently under investigation.

Finally, the novel wind turbine MPCs can be applied on small-scale wind turbine models in the wind tunnel, or to full-scale turbines in the field. In terms of computational load, interestingly, the small-scale wind turbine models are expected to entail higher challenges. Since their eigenfrequencies are significantly higher, the controller sample time has to be reduced, which limits the solution time of the MPC algorithm. A typical counter-action in this case would be the reduction of solution time by decreasing the prediction horizon length. In this regard, an impressive result of the present dissertation is that, in contrast to conventional MPCs, the novel MPCs comprising a stress history retain high fatigue performance even with short prediction horizons.

BIBLIOGRAPHY

[1] IPCC, “Summary for policymakers,” in *Climate change 2014*, O. Edenhofer, Ed. Cambridge: Cambridge University Press, 2014. ISBN 978-1-107-05821-7

[2] IPCC, “Summary for policymakers,” in *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, V. Masson-Delmotte, P. Zhai, A. Pirani, S. L. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M. I. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J. Matthews, T. K. Maycock, T. Waterfield, O. Yelekçi, R. Yu, and B. Zhou, Eds. Cambridge University Press, 2021. [Online]. Available: https://www.ipcc.ch/report/ar6/wg1/

[3] Agora Energiewende und Aurora Energy Research, “The german coal commission: A roadmap for a just transition from coal to renewables,” Berlin, 2019. [Online]. Available: https://www.agora-energiewende.de/en/publications/the-german-coal-commission/

[4] D. C. DiGiulio and R. B. Jackson, “Impact to underground sources of drinking water and domestic wells from production well stimulation and completion practices in the pavillion, wyoming, field,” *Environmental Science & Technology*, vol. 50, no. 8, pp. 4524–4536, 2016. doi:10.1021/acs.est.5b04970

[5] U.S. EPA, “Hydraulic fracturing for oil and gas: Impacts from the hydraulic fracturing water cycle on drinking water resources in the united states,” Washington, DC, 2016. [Online]. Available: https://cfpub.epa.gov/ncea/hfstudy/recordisplay.cfm?deid=332990

[6] World Health Organization, “Ambient (outdoor) air pollution,” 2018. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health

[7] Statistisches Bundesamt, “Beschäftigte, umsatz, produktionswert und wertschöpfung der unternehmen in der energie- und wasserversorgung,” 2021. [Online]. Available: https://www.destatis.de/DE/Themen/Branchen-Unternehmen/Energie/Beschaeftigte-Umsatz-Investitionen/_inhalt.html

[8] Working Group on Renewable Energy-Statistics, “Development of renewable energy sources in germany in the year 2020,” 2021. [Online]. Available: https://www.erneuerbare-energien.de/EE/Navigation/DE/Service/Erneuerbare_Energien_in_Zahlen/Entwicklung/entwicklung-der-erneuerbaren-energien-in-deutschland.html

[9] H. Garrett-Peltier, “Green versus brown: Comparing the employment impacts of energy efficiency, renewable energy, and fossil fuels using an input-output model,” *Economic Modelling*, vol. 61, pp. 439–447, 2017. doi:10.1016/j.econmod.2016.11.012

[10] M. Roser, “Why did renewables become so cheap so fast? and what can we do to use this global opportunity for green growth?” 2020. [Online]. Available: https://ourworldindata.org/cheap-renewables-growth

[11] BloombergNEF, "Annual battery price survey: Battery pack prices cited below $100/kwh for the first time in 2020," 2020. [Online]. Available: https://about.bnef.com/blog/battery-pack-prices-cited-below-100-kwh-for-the-first-time-in-2020-while-market-average-sits-at-137-kwh/

[12] J.-F. Mercure, H. Pollitt, J. E. Viñuales, N. R. Edwards, P. B. Holden, U. Chewpreecha, P. Salas, I. Sognnaes, A. Lam, and F. Knobloch, "Macroeconomic impact of stranded fossil fuel assets," *Nature Climate Change*, vol. 8, no. 7, pp. 588–593, 2018. doi:10.1038/s41558-018-0182-1

[13] S. Mishra, K. Anderson, B. Miller, K. Boyer, and A. Warren, "Microgrid resilience: A holistic approach for assessing threats, identifying vulnerabilities, and designing corresponding mitigation strategies," *Applied Energy*, vol. 264, p. 114726, 2020. doi:10.1016/j.apenergy.2020.114726

[14] Global Commission on the Geopolitics of Energy Transformation, "A new world - the geopolitics of the energy transformation," 2019. [Online]. Available: http://geopoliticsofrenewables.org/Report

[15] J. D. Sachs and A. M. Warner, "The curse of natural resources," *European Economic Review*, vol. 45, no. 4-6, pp. 827–838, 2001. doi:10.1016/S0014-2921(01)00125-8

[16] trend:research GmbH, "Eigentümerstruktur: Erneuerbare energien: Entwicklung der akteursvielfalt, rolle der energieversorger, ausblick bis 2020," 2017.

[17] D. Nelson, M. Huxham, S. Muench, and B. O'Connell, "Policy and investment in german renewable energy," 2016.

[18] D. Scholten and R. Bosman, "The geopolitics of renewables; exploring the political implications of renewable energy systems," *Technological Forecasting and Social Change*, vol. 103, pp. 273–283, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0040162515003091. doi:10.1016/j.techfore.2015.10.014

[19] J. Colgan, "Fueling the fire: Pathways from oil to war," *International Security*, vol. 38, pp. 147–180, 2013. doi:10.1162/ISEC_a_00135

[20] M. Ross, "A closer look at oil, diamonds, and civil war," *Annual Review of Political Science*, vol. 9, no. 1, pp. 265–300, 2006. doi:10.1146/ANNUREV.POLISCI.9.081304.161338

[21] A. Levallois, J.-C. Cousseran, and L. Kerrello, "The financing of the 'islamic state' in iraq and syria (isis)," 2017. [Online]. Available: https://www.europarl.europa.eu/thinktank/en/document.html?reference=EXPO_IDA(2017)603835

[22] V. Bove, K. S. Gleditsch, and P. G. Sekeris, ""oil above water": Economic interdependence and third-party intervention," *Journal of Conflict Resolution*, vol. 60, no. 7, pp. 1251–1277, 2016. doi:10.1177/0022002714567952

[23] J. A. Paul, "Oil companies in iraq," in *2003 Conference on Corporate Accountability*, Global Policy Forum, Ed., Berlin, 2003. [Online]. Available: https://archive.globalpolicy.org/component/content/article/185-general/40586.html

[24] NBC News, "Mccain clarifies remark about oil, iraq war," *NBC News*, vol. 2008, 03.05.2008. [Online]. Available: https://www.nbcnews.com/id/wbna24434071

[25] J. F. Manwell, J. G. McGowan, and A. L. Rogers, *Wind energy explained: Theory, design and application*, repr ed. Chichester: Wiley, 2008. ISBN 0-471-49972-2

[26] A. Maheshwari, *Modelling, Aging and Optimal Operation of Lithium-ion Batteries.* Eindhoven: Technische Universiteit Eindhoven, 2018. ISBN 978-90-386-4607-7. [Online]. Available: https://research.tue.nl/en/publications/modelling-aging-and-optimal-operation-of-lithium-ion-batteries

[27] D. Schlipf, H. Fürst, S. Raach, and F. Haizmann, "Systems engineering for lidar-assisted control: A sequential approach," *Journal of Physics: Conference Series*, vol. 1102, p. 012014, 2018. doi:10.1088/1742-6596/1102/1/012014

[28] ASTM, "Standard practices for cycle counting in fatigue analysis," West Conshohocken, PA, 1985.

[29] C. Heinrich, M. Khalil, K. Martynov, and U. WEVER, "Online remaining lifetime estimation for structures," *Mechanical Systems and Signal Processing*, vol. 119, pp. 312–327, 2019. doi:10.1016/j.ymssp.2018.09.028

[30] M. Musallam and C. M. Johnson, "An efficient implementation of the rainflow counting algorithm for life consumption estimation," *IEEE Transactions on Reliability*, vol. 61, no. 4, pp. 978–986, 2012. doi:10.1109/TR.2012.2221040

[31] S. Gros and A. Schild, "Real-time economic nonlinear model predictive control for wind turbine control: International journal of control," *International Journal of Control*, pp. 1–14, 2017. doi:10.1080/00207179.2016.1266514

[32] M. A. Evans, M. Cannon, and B. Kouvaritakis, "Robust mpc tower damping for variable speed wind turbines," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 290–296, 2015. doi:10.1109/TCST.2014.2310513

[33] J. J. Barradas-Berglind and R. Wisniewski, "Representation of fatigue for wind turbine control," *Wind Energy*, vol. 19, no. 12, pp. 2189–2203, 2016. doi:10.1002/we.1975

[34] S. Loew, D. Obradovic, and C. L. Bottasso, "Model predictive control of wind turbine fatigue via online rainflow-counting on stress history and prediction," *Journal of Physics: Conference Series*, vol. 1618, p. 22041, 2020. doi:10.1088/1742-6596/1618/2/022041

[35] M. Koller, T. Borsche, A. Ulbig, and G. Andersson, "Defining a degradation cost function for optimal control of a battery energy storage system," in *2013 IEEE Grenoble Conference*, 2013, pp. 1–6.

[36] H. Sanchez, T. Escobet, V. Puig, and P. F. Odgaard, "Health-aware model predictive control of wind turbines using fatigue prognosis," *IFAC-PapersOnLine*, vol. 48, no. 21, pp. 1363–1368, 2015. doi:10.1016/j.ifacol.2015.09.715

[37] J. d. J. Barradas-Berglind, R. Wisniewski, and M. Soltani, "Fatigue damage estimation and data-based control for wind turbines," *IET Control Theory & Applications*, vol. 9, no. 7, pp. 1042–1050, 2015. doi:10.1049/iet-cta.2014.0730

[38] D. Collet, M. Alamir, D. Di Domenico, and G. Sabiron, "Non quadratic smooth model of fatigue for optimal fatigue-oriented individual pitch control," *Journal of Physics: Conference Series*, vol. 1618, p. 022004, 2020. doi:10.1088/1742-6596/1618/2/022004

[39] J. Luna, O. Falkenberg, S. Gros, and A. Schild, "Wind turbine fatigue reduction based on economic-tracking nmpc with direct ann fatigue estimation," *Renewable Energy*, vol. 147, pp. 1632–1641, 2020. doi:10.1016/j.renene.2019.09.092

[40] M. Behrendt, "A basic working principle of model predictive control: Cc by-sa 3.0," 2009. [Online]. Available: https://de.wikipedia.org/wiki/Model_Predictive_Control

[41] S. Löw and D. Obradovic, "System control," Patent WO2 020 094 461A1, 2020.

[42] S. Loew and D. Obradovic, "Real-time implementation of nonlinear model predictive control for mechatronic systems using a hybrid model," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, 2018, pp. 164–167.

[43] S. Loew and D. Obradovic, "Real-time nonlinear model predictive control: Predictive control for mechatronic systems using a hybrid model," *atp magazin*, no. 08, pp. 46–52, 2018.

[44] S. Loew, D. Obradovic, and C. L. Bottasso, "Direct online rainflow-counting and indirect fatigue penalization methods for model predictive control," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3371–3376. doi:10.23919/ECC.2019.8795911

[45] S. Loew, D. Obradovic, and C. L. Bottasso, "Economic nonlinear model predictive control of fatigue—formulation and application to wind turbine control," *Optimal Control Applications and Methods*, 2022. doi:10.1002/oca.2870

[46] S. Loew, D. Obradovic, A. Anand, and A. Szabo, "Stage cost formulations of online rainflow-counting for model predictive control of fatigue," in *2020 European Control Conference (ECC)*, 2020, pp. 475–482. doi:10.23919/ECC51009.2020.9143939

[47] S. Loew and D. Obradovic, "Formulation of fatigue dynamics as hybrid dynamical system for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6616–6623, 2020. doi:10.1016/j.ifacol.2020.12.080

[48] S. Loew and C. L. Bottasso, "Lidar-assisted model predictive control of wind turbine fatigue via online rainflow-counting considering stress history," *Wind Energy Science Discussions*, vol. 2021, pp. 1–31, 2021. doi:10.5194/wes-2021-119

[49] S. Loew, A. Anand, and A. Szabo, "Economic model predictive control of li–ion battery cyclic aging via online rainflow–analysis," *Energy Storage*, 2021. doi:10.1002/est2.228

[50] A. Anand, S. Loew, and C. L. Bottasso, "Economic control of hybrid energy systems composed of wind turbine and battery," in *2021 European Control Conference (ECC)*, 2021, pp. 2565–2572. doi:10.23919/ ECC54610.2021.9654911

[51] H. Canet, S. Löw, and C. L. Bottasso, "Lidar-assisted control in wind turbine design: Where are the potential benefits?" *Journal of Physics: Conference Series*, vol. 1618, p. 042020, 2020. doi:10.1088/1742-6596/1618/4/042020

[52] H. Canet, S. Loew, and C. L. Bottasso, "What are the benefits of lidar-assisted control in the design of a wind turbine?" *Wind Energy Science*, vol. 6, no. 5, pp. 1325–1340, 2021. [Online]. Available: https://wes.copernicus.org/articles/6/1325/2021/. doi:10.5194/wes-6-1325-2021

[53] E. Haibach, *Betriebsfestigkeit: Verfahren und Daten zur Bauteilberechnung*, 3rd ed., ser. VDI-Buch. Berlin: Springer, 2006. ISBN 9781280618024

[54] M. A. Miner, "Culmulative damage in fatigue," *Journal of Applied Mechanics*, pp. 159–164, 1945.

[55] G. Marsh, C. Wignall, P. R. Thies, N. Barltrop, A. Incecik, V. Venugopal, and L. Johanning, "Review and application of rainflow residue processing techniques for accurate fatigue damage estimation," *International Journal of Fatigue*, vol. 82, pp. 757–765, 2016. doi:10.1016/j.ijfatigue.2015.10.007

[56] M. Köhler, S. Jenne, K. Pötter, and H. Zenner, *Zählverfahren und Lastannahme in der Betriebsfestigkeit*. Dordrecht: Springer, 2012. ISBN 978-3-642-13163-9

[57] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model predictive control: Theory, computation, and design*, 2nd ed. Madison, Wisconsin: Nob Hill Publishing, 2017. ISBN 9780975937730

[58] K. Hammerum, "A fatigue approach to wind turbine control," Master's thesis, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark, 2006.

[59] J. J. Barradas-Berglind, B. Jayawardhana, and R. Wisniewski, Eds., *Wind turbine control with active damage reduction through energy dissipation: 2016 American Control Conference (ACC)*, 2016. doi:10.1109/ACC.2016.7526594

[60] B. Xu, J. Zhao, T. Zheng, E. Litvinov, and D. S. Kirschen, "Factoring the cycle aging cost of batteries participating in electricity markets," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 2248–2259, 2018. doi:10.1109/TPWRS.2017.2733339

[61] M. A. Müller and F. Allgöwer, "Improving performance in model predictive control: Switching cost functionals under average dwell-time," *Automatica*, vol. 48, no. 2, pp. 402–409, 2012. doi:10.1016/j.automatica.2011.11.005

[62] R. K. Al Seyab, "Nonlinear model predictive control using automatic differentiation," Ph.D. dissertation, Cranfield University, Bedfordshire, UK, 2007.

[63] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms*, 2nd ed., ser. Communications and Control Engineering. Springer International Publishing, 2017. ISBN 978-3-319-46023-9

[64] R. Findeisen and F. Allgöwer, "An introduction to nonlinear model predictive control," in *21st Benelux Meeting on Systems and Control*, 2002, pp. 1–23.

[65] J. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Pub., 2009.

[66] B. Houska, H. J. Ferreau, and M. Diehl, "Acado toolkit—an open–source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/pdf/10.1002/oca.939. doi:10.1002/oca.939

[67] A. Anand, "Optimal control of battery energy storage system for grid integration of wind turbines," Master Thesis, TU Munich, Munich, 2020.

[68] Y. Shi, B. Xu, Y. Tan, and B. Zhang, "A convex cycle-based degradation model for battery energy storage planning and operation," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 4590–4596. doi:10.23919/ACC.2018.8431814

[69] B. J. Jonkman and L. Kilcher, *Turbsim User's Guide: Version 1.06*, 2012.

[70] IEC, "Iec 61400-1 ed.3: wind turbines - part 1: design requirements," 2005.

[71] sowento GmbH, "Lidar simulator v1.4.4: Documentation and user guide," Stuttgart, 2020.

[72] D. Schlipf, "Lidar-assisted control concepts for wind turbines," Dissertation, University of Stuttgart, Stuttgart, Germany, 2016.

[73] C. L. Bottasso, P. Pizzinelli, C. Riboldi, and L. Tasca, “Lidar-enabled model predictive control of wind turbines with real-time capabilities,” *Renewable Energy*, vol. 71, pp. 442–452, 2014. doi:10.1016/j.renene.2014.05.041

[74] D. Schlipf, P. Grau, S. Raach, R. Duraiski, J. Trierweiler, and P. W. Cheng, “Comparison of linear and nonlinear model predictive control of wind turbines using lidar,” in *2014 American Control Conference*. IEEE, 04.06.2014 - 06.06.2014, pp. 3742–3747. doi:10.1109/ACC.2014.6859205

[75] S. M. Mousavi G. and M. Nikdel, “Various battery models for various simulation studies and applications,” *Renewable and Sustainable Energy Reviews*, vol. 32, pp. 477–485, 2014. doi:10.1016/j.rser.2014.01.048

[76] Z. Gao, C. Chin, W. Woo, and J. Jia, “Integrated equivalent circuit and thermal model for simulation of temperature-dependent lifepo4 battery in actual embedded application,” *Energies*, vol. 10, no. 1, p. 85, 2017. doi:10.3390/en10010085

[77] Y. Shi, B. Xu, Y. Tan, and B. Zhang, “A convex cycle-based degradation model for battery energy storage planning and operation,” in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 4590–4596.

[78] I. Laresgoiti, S. Käbitz, M. Ecker, and D. U. Sauer, “Modeling mechanical degradation in lithium ion batteries during cycling: Solid electrolyte interphase fracture,” *Journal of Power Sources*, vol. 300, pp. 112–122, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378775315302949. doi:10.1016/j.jpowsour.2015.09.033

[79] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, *Definition of a 5-MW Reference Wind Turbine for Offshore System Development: Technical Report NREL/TP-500-38060*. National Renewable Energy Laboratory, 2009.

[80] T. Rubert, D. McMillan, and P. Niewczas, “A decision support tool to assist with lifetime extension of wind turbines,” *Renewable Energy*, vol. 120, pp. 423–433, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0960148117312685. doi:10.1016/j.renene.2017.12.064

[81] J. Engels, B. Claessens, and G. Deconinck, “Techno-economic analysis and optimal control of battery storage for frequency control services, applied to the german market,” *Applied Energy*, vol. 242, pp. 1036–1049, 2019. doi:10.1016/j.apenergy.2019.03.128

[82] PJM, “Pjm regulation market,” 2017. [Online]. Available: https://learn.pjm.com/three-priorities/buying-and-selling-energy/ancillary-services-market/regulation-market.aspx

[83] H. Do and D. Söffker, “Wind turbine lifetime control using structural health monitoring and prognosis,” *IFAC-PapersOnLine*, vol. 53, pp. 12 669–12 674, 2020. doi:10.1016/j.ifacol.2020.12.1847

[84] B. Ritter, A. Schild, M. Feldt, and U. Konigorski, “The design of nonlinear observers for wind turbine dynamic state and parameter estimation,” *Journal of Physics: Conference Series*, vol. 753, p. 052029, 2016. doi:10.1088/1742-6596/753/5/052029

[85] C. Bottasso, A. Croce, R. Nicastro, B. Savini, and L. Riviello, “Neural adaptive control of multibody systems,” in *3rd Asian Conference on Multibody Dynamics*, Komaba, Tokyo, Japan, 2006.

[86] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, “From linear to nonlinear mpc: Bridging the gap via the real-time iteration,” *International Journal of Control*, pp. 1–19, 2016. doi:10.1080/00207179.2016.1222553

[87] S. Gros, “An economic nmpc formulation for wind turbine control,” in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 1001–1006.

APPENDIX **A**

Model Predictive Control

In the present chapter, selected existing concepts in the domain of Model Predictive Control are explained and compared. Instead of claiming completeness, a brief and rather application-focused presentation is chosen.

A.1 Linear MPC & Nonlinear MPC

Many controlled plants are of nonlinear nature. However, MPCs for these plants can be set up based either on a simplified linear or on a more realistic nonlinear model of the system dynamics. This corresponds to Linear MPC (LMPC) or Nonlinear MPC (NMPC) formulations, respectively.

In LMPC, the system dynamics and constraints are linearized offline at the desired reference states. During online operation, these linearizations become constraints of a Quadratic Program (QP). At each control step, this QP provides new control variables which are applied to the plant system. An advantage of LMPC is the reliable solution of this QP to its global optimum. However, this solution does not necessarily reflect the best control action for the real nonlinear plant. A further disadvantage of LMPC is the decrease of performance if the plant does not operate in close proximity to the desired reference states. [86]

In NMPC, the nonlinear system dynamics and constraints are explicitly evaluated during online operation, and become constraints of a Nonlinear Program. An advantage of NMPC is that the performance does not systematically depend on the current position in the state space. A disadvantage of NMPC is that the solution time of the Nonlinear Program is variable throughout subsequent MPC steps, which renders real-time implementations being not straightforward. The challenge of variable solution times is targeted in the subsequent section. [86]

A.2 Sequential Quadratic Programming & Real-time Iteration

A common method for solving NLPs is the *Sequential Quadratic Programming* (SQP) approach. Here, the NLP is sequentially approximated by QPs. This procedure is repeated until convergence, which can be very time-consuming. Thus, instead, all MPC implementations of the present dissertation are based on the *Real-Time Iteration* (RTI) [86] approach. Here, instead of an entire SQP, only one QP is solved at each control step. The RTI approach is based on the assumption that the optimization problems vary only slowly over time. Thus, the QPs in subsequent MPC steps approximate an SQP, and each individual QP solution can be assumed to be close to convergence. [86]

A.3 Simultaneous & sequential approach

The solution of the system dynamics and of the optimization problem can be performed simultaneously or sequentially. In the simultaneous approach, the solution of the system dynamics is split over small time intervals in the prediction horizon. The initial states for each time interval are defined as extra optimization variables (see also *Multiple Shooting approach*). Additional constraints are needed in order to ensure smooth trajectories of the system dynamics. This approach is advantageous for highly nonlinear systems and if parallelization is intended. Main disadvantages are the increased optimization effort and the chance of discontinuous state trajectories in case of convergence problems. [64]

Alternatively, in the *Sequential approach* the system dynamics are solved in a forward simulation over the entire prediction horizon; in this case no extra optimization variables or constraints are needed. After solving the simulation, the optimization problem is set up and solved. [64]

A.4 Tracking & economic MPC

The simplest and, by now, most common formulation of MPC cost functions is the quadratic penalization of deviation of the state trajectories from reference trajectories. This formulation represents *Tracking MPC*. In contrast, for *Economic MPC*, arbitrary nonlinear functions are employed as cost functions. Very often, these functions are based on revenue or profit considerations which defines the name *Economic*. [87]

A.5 Stage & terminal cost function

Stage costs comprise a summation of state samples or a time integral of state trajectories over the prediction horizon. Terminal costs are defined as functions of the state samples at the end of the prediction horizon. [63]