Rapid control prototyping of model predictive wind turbine control toward field testing

S Dickler¹, T Kallen¹, J Zierath² and D Abel¹

¹ Institute of Automatic Control, RWTH Aachen University, Campus-Boulevard 30, 52074 Aachen, Germany
² W2E Wind to Energy GmbH, Grubenstraße 44, 18055 Rostock, Germany
E-mail: sebastian.dickler@rwth-aachen.de

Abstract. Wind turbine technology has a great demand for advanced control strategies that deal with multiple objectives simultaneously. Model predictive control (MPC) is a strong candidate for modern wind turbine control (WTC). While the design of model predictive wind turbine controllers in simulations has been extensively investigated in academic studies, the application of these controllers to real wind turbines reveals open research challenges. In this work, we address the systematic development of MPC systems for wind turbines. Based on the rapid control prototyping (RCP) methodology, we propose a comprehensive development environment for designing and testing such control systems in system simulations, software-in-the-loop (SiL) and hardware-in-the-loop (HiL) tests. We use a continuous tool chain from the MATLAB/Simulink environment to the wind turbine’s programmable logic controller (PLC) hardware. This study provides an example of how to prepare an MPC algorithm for tests in a real 3 MW wind turbine by demonstrating control operation over the entire operation range in system simulations and SiL tests as well as verifying the MPC’s real-time feasibility in the wind turbine’s PLC in HiL tests.

1. Introduction
Efficient wind turbine operation takes an essential role in wind energy technology as part of the energy transition. Applying advanced control strategies to real wind turbines is challenging but at the same time vital for the success of this technology. Model predictive control (MPC) is a promising approach for modern wind turbine control (WTC): It benefits from its abilities to handle multi-variable control problems as well as to consider process constraints explicitly for calculating optimal values for the manipulated variables.

In its research agenda [1], the European Academy of Wind Energy (eawe) addresses current and future challenges in wind energy and points out that »although several [...] wind plants are currently operating in Europe, most of the academic results in the field of wind plant controls have not yet been validated on a fullscale wind plant«. With this particular challenge, the eawe addresses applied research in the field of WTC and demands comprehensive experimental testing of the WTC systems designed in academia.

The application of MPC algorithms in real wind turbines requires extensive, standardized testing throughout the entire development process. To this end, this study proposes a holistic development environment to design and test such control systems in system simulations, software- and hardware-in-the-loop tests.
1.1. Theory and research

Over the past 15 years, many academic studies investigated the subject of model predictive wind turbine control (MPWTC) [2]. The publications mostly focus on the design and simulative testing of linear and nonlinear MPWTC algorithms. Although the implementation and testing in industrial hardware are vital prior to the evaluation of a new control system in field tests, only very few studies consider these development steps.

However, recent publications, such as [3], explicitly propose real-time capable implementations of (nonlinear) MPWTC algorithms. For example, [4] addresses a nonlinear state observer for WTC and its implementation in a programmable logic controller (PLC). Moreover, [5] focuses on the computational performance of a nonlinear MPC and a moving horizon estimator for airborne wind energy systems using automatic code generation for the experimental validation in an industrial PC. Among these studies, we have already investigated the hardware-in-the-loop testing of MPWTC systems in [6] and [7].

In the field of WTC, a systematic control design has been addressed in [8] and [9]. In general, the methodology of Rapid Control Prototyping (RCP) provides a generic framework for a systematic control development [10]. To our knowledge, the systematic development of MPWTC systems – comprising all required steps from control design to field testing – has not been addressed in the literature yet.

1.2. Objectives and research question

This study aims to derive a systematic development process for MPWTC systems from the RCP methodology and thus addresses the following research question: »How is the general RCP development process best adapted to systematically develop MPWTC systems for field testing?«

To this end, we adapt the existing RCP framework [10] to the specific problem of designing and testing an MPWTC system in a holistic development process for evaluations in a real wind turbine. We consider a structure of the wind turbine’s automation system as shown in Figure 1 and address the control system under test. In this publication, we focus on the development steps of the control testing and implementation – trying to bridge the gap between the control design and field testing, paving the way toward field testing.

The paper is organized as follows. Section 2 explains the RCP methodology with its testing stages and requirements for an integrated development environment with regard to MPWTC systems. Section 3 introduces an RCP system, which is suitable for MPWTC development, and
presents its model structure, submodels and tool chains; this section also points out certain use cases for this development environment. Section 4 uses this RCP system for an exemplary validation of an MPWTC system in system simulations, software- and hardware-in-the-loop tests. Section 5 discusses the usability of the presented RCP process for a systematic MPWTC development. Finally, Section 6 summarizes the overall findings and identifies avenues for future research.

2. Rapid control prototyping methodology

The RCP methodology in [10] provides a generic development framework for control systems that is applicable to any type of technological application and control algorithm.

![V-model and RCP procedure](image)

Figure 2: Graphical representations of the RCP development process comprising the testing stages of the system simulation, SiL and HiL test [10].

The V-model in Figure 2a represents a control development process comprising its iteratively processed development steps. The RCP development process distinguishes between the following testing stages\(^1\) (Figure 2b):

- **System simulation**: The control algorithm and process model are simulated in the development platform. System simulations aim to investigate the process dynamics and to derive initial parameter sets which meet the specified requirements. For MPWTC algorithms, an accurate representation of the identified wind turbine dynamics in the prediction model is vital for the subsequent development steps.

- **Software-in-the-Loop (SiL)**\(^2\): The control algorithm is compiled as executable code and is employed in the development platform with the process model. We use SiL tests to evaluate the code compilation process and the inclusion of external code into the control algorithm.

- **Hardware-in-the-Loop (HiL)**: The control algorithm is executed in the target hardware and controls the real-time simulated process. HiL tests aim to validate complete functionality, robustness and safety of the control system under test.

\(^1\) In the literature, there is no consistent terminology for such testing stages. However, similar test setups are used across the existing publications. In this publication, we rely on the definitions of [10].

\(^2\) According to [10], in ‘software-in-the-loop’ tests, the control algorithm would be executed in a real-time prototyping platform and would control the real process. We modify this testing stage from [10] due to the safety risk of controlling the real wind turbine with an control algorithm that is still under test.
Figure 3: Graphical user interface of the WTCF: (a) main model includes subsystems plant, control system, disturbance and analysis; (b) project’s folder structure corresponds to structure of models and submodels; (c) ‘control system’ interface includes subsystems observer and controller.

An RCP system requires an integrated development environment and a continuous tool chain to process the following steps:

(I) Modeling and simulation of the process in a graphical programming environment
(II) Design of the control system based on the process model
(III) Graphical implementation of the control system
(IV) SiL testing of the control system
(V) Code generation and optimization of the control system for the target hardware
(VI) HiL testing of the control system using the target hardware

In particular, the continuous tool chain provides automatic code generation of program code customized for the target hardware. The integrated development environment enables performing horizontal iteration loops in the V-model – which is vital for an agile control testing throughout the entire development process.

3. RCP system for MPWTC development

In this section, we introduce an RCP system for a systematic development of MPWTC systems. We describe the software and hardware setup and the implemented tool chains.

3.1. Wind Turbine Control Framework

To our knowledge, there is no software package available from stock that innately meets the requirements for an integrated development environment (I)–(VI) for the systematic design and test of MPWTC systems. Therefore, we implemented the so-called Wind Turbine Control Framework (WTCF).

We use MATLAB/Simulink R2018b throughout the development process. For the WTCF, we use Simulink’s graphical programming environment and its native features ‘Simulink Project’,
‘Simulink Buses’, ‘Model References’ and ‘Configuration References’ to set up an agile simulation environment. The entire project (with its models, functions, parameters, etc.) is stored in a Git repository for version control.

Figure 3a and Figure 3c show an example of the hierarchical model structure in the WTCF; its project’s folder structure is depicted in Figure 3b. The Simulink model has three different hierarchical layers: main, interface and submodel layer. The main layer (Figure 3a) corresponds to a standard control loop structure and comprises the interfaces for the underlying submodels (here: plant, control system, analysis, disturbance). The interface layer links the submodels with the main layer (Figure 3c: here, the control system integrates the observer and controller subsystems into the main model). While the main and interface layers establish the structure of the closed loop model, the underlying models in the submodel layer contain the actual program code and define the process dynamics (see Section 3.2). We use Simulink’s Model References and Simulink Buses to provide interchangeable submodels with standardized interfaces.

### 3.2. Submodels in the WTCF

In the following, we briefly describe the submodels used in the WTCF that define the wind turbine process dynamics (compare Figure 3).

**Plant: wind turbine** We distinguish between different simulation model types (representing the employed simulation tool) and different wind turbine types (specifying the considered wind turbine). We use alaska/Wind [11], FAST\(^3\) [12], FLEX5 and a nonlinear reduced order model in MATLAB/Simulink to simulate the wind turbine plant dynamics. The reduced order wind turbine model is described in detail in [6] and [13] and comprises simplified mechanical and aerodynamic submodels (linear dynamics for the drive train, tower and blades as well as static maps for the aerodynamic coefficients). More detailed simulations of the wind turbine dynamics can be conducted with FAST, FLEX5 (aero-elastic simulation) and alaska/Wind (multi-body simulation). By parametrizing the previously described simulation tools, we are able to simulate various wind turbine types. In this study, we consider the ‘W2E-120/3.0fc’ wind turbine designed by W2E Wind to Energy ([7], [14]).

**Control system: MPWTC** The MPWTC system comprises an MPC algorithm and an extended Kalman filter (EKF). The control system with its cost function, prediction model, constraints and state observer has already been described in detail in previous publications ([6], [13], [15]). The MPC algorithm solves the online optimization problem

\[
\Delta u_{opt.,|k} = \min_{\Delta u_{|k}} J(\Delta u_{|k}),
\]

subject to constraints in order to maintain limits for the manipulated, state and controlled variables. The MPC recursively calculates the optimal trajectory \(\Delta u_{opt.,|k}\) by minimizing the quadratic cost function \(J\) with a sampling time of \(\Delta T_{MPC} = 100\, \text{ms}\) and prediction and control horizons of \(H_p = 5\, \text{s}\) and \(H_u = 1\, \text{s}\), respectively. We use qpOASES [16] to solve the quadratic programming problem.

**Disturbance: wind** According to the design load cases and load sets in IEC 61400-1, we consider extreme operating gusts (EOGs) and normal wind conditions in control region 2 and region 3. We examine mean wind speeds from \(\bar{V} = 4\) to \(25\, m/s\) and four random seeds for the wind field generation with TurbSim; we apply the Kaimal turbulence model with the IEC turbulence class A or a turbulence intensity of \(I = 10\%\).

\(^3\) We use FAST v7 and FAST v8; the most recent version OpenFAST has not yet been included in the WTCF.
In addition, we define a ‘benchmark load case’ with turbulent wind segments in the partial- and full-load range (with \( \bar{V} = 5, 11, \) and \( 20 \) m/s using turbulence class A and \( I = 10\% \), respectively) as well as three EOGs (1-year and 50-year EOGs at \( V = 10 \) m/s and a 50-year EOG at \( V = 15 \) m/s) in order to evaluate the control system over the entire operating range. The benchmark load case is used in Section 4 and depicted in Figure 5.

3.3. Use cases of the WTCF

We use the WTCF as central RCP system throughout the entire MPWTC development process and particularly benefit from the reusability of the submodels from Section 3.2.

Several plant models can be used for the wind turbine dynamics by replacing the plant interface in Figure 3a. Thus, we are able to flexibly combine, simulate and test any combination of employed simulation tool and specified wind turbine as well as to extend the WTCF with further wind turbine plant dynamics. By modeling the wind turbine with different simulation tools, the controller can be tested against two different plant models, each of which has deviations from the real plant. Thus, model-based uncertainties in the controller design can be reduced.

Diverse control systems can be evaluated either by replacing the entire control system (Figure 3a) or by adding, exchanging or removing certain subsystems (Figure 3c: controller or observer). With this setup, we are able to test various MPWTC systems against different wind turbine plant dynamics and to evaluate the impact of differing controller and observer instances.

4. RCP of an MPWTC system

In this section, we use the previously described WTCF and present our control testing procedure with system simulations, SiL and HiL tests (see Figure 4) exemplary for the ‘W2E-120/3.0fc’ wind turbine. The development steps of the control specification and the control design are vital for the complete control development, these steps have already been addressed in several publications (see Section 1.1). Thus, in the following, we focus on the control testing and implementation in order to prepare the MPWTC algorithm for field testing.

4.1. System simulation

We use system simulations to analyze and validate the prediction model and to derive MPWTC parameter sets which are applicable in the subsequent testing stages. In the following, we focus on the testing procedure itself instead of investigating the control results – for detailed discussions of the MPWTC’s control behavior in this simulation setup, we refer to [6], [13] and [15].

The system simulation setup is depicted in Figure 4. We use the WTCF to conduct simulations with a closed loop model as shown in Figure 3a and an MPWTC system according
to Figure 3c (with separated, exchangeable controller and observer subsystems). We employ different simulation tools (alaska/Wind, FAST and MATLAB/Simulink) for the wind turbine plant dynamics. Thus, we can simulate different levels of detail of the plant dynamics.

In a first testing step, we use a simulation setup with the nonlinear reduced order plant model and a control system that only comprises the MPC algorithm (without the EKF; compare Figure 3c).\textsuperscript{4} Hence, the MPC should offer ideal reference tracking (within the limits of the system dynamics). With this setup, we validate the general functioning of the MPWTC implementation and derive an initial parametrization for the MPC algorithm.

In a second step, we use the reduced order plant model again and extend the MPWTC system with the EKF (that estimates the state variables and processes their vector to the MPWTC). Since the observer’s estimation has a strong impact on the MPWTC performance, this step validates the observer–controller co-operation in the MPWTC system and aims at deriving applicable parameter sets for the EKF and the MPC (based on the parameters initially identified). Exemplary control results for this simulation setup are shown in Figure 5; they are compared to SiL results in Section 4.2. In particular, these control results indicate the functioning of the MPWTC system in the partial- as well as in the full-load range – which is vital for

\textsuperscript{4} This setup has two implications: a) By using the reduced order plant model, the state variables are measurable and therefore no state estimation is required, b) the prediction model within the MPC perfectly corresponds to the controlled wind turbine plant model.
application in real wind turbines.

In subsequent testing steps, we use – depending on the focus of analysis – the simulation tools FAST or alaska/Wind. We evaluate the accuracy of the prediction model with predicted trajectories of the controlled and manipulated variables over the prediction and control horizon. Deviations among the plant models emulate modeling uncertainties compared to the real wind turbine dynamics. Thus, we evaluate the previously parametrized MPWTC system against more realistic plant dynamics and increase the control system’s robustness by adjusting its parametrization.

4.2. Software-in-the-loop test

After successfully validating the MPWTC algorithm in system simulations, we apply SiL tests in order to a) verify the proper functioning of the MPWTC’s compiled code, b) evaluate the MPWTC’s robustness and c) test the co-operation between MPWTC and automation system in a simulated environment. Thus, in SiL tests, we evaluate the functional integration of the MPWTC system into the existing automation system.

The SiL setup is shown in Figure 4. For the plant model, we employ FLEX5 as a certified, established simulation tool in wind industry (used for load calculations in the wind turbine’s design process). Since FLEX5 does not provide a Simulink interface, we employ the tool as executable program (*.exe) in Windows. We generate a dynamic link library (*.dll) of the MPWTC system out of the Matlab/Simulink environment in order to couple the FLEX5 model, the automation system and the MPWTC system for closed loop simulations. The ‘bypass interface’ in Figure 1 enables the communication between MPWTC and automation system and routes the command, measurement and reference signals between both systems.

Figure 5 shows exemplary control results of the MPWTC system in the SiL test and the system simulation for the benchmark load case. In general, the results match in both testing stages. Slight differences can be identified in the pitch angle signals within 400–550 s: While the MPWTC system is permanently active in the system simulation, in this time interval of the SiL test, the control system of the existing automation system (compare Figure 1) is active and controls the process; consequently, the pitch angle commands diverge. Further deviations can be found in the generator torque signals at approximately 3900 s: These differences arise simultaneously with the third EOG at $\bar{V} = 15 \, m/s$ and result from deviations between the reduced order plant model and FLEX5.

Generally speaking, by using the compiled MPWTC system (*.dll), we particularly verify the proper integration of the external C code for the QP solver. Moreover, by using the FLEX5 tool, we validate the robustness of the MPWTC parametrization due to plant model deviations. Thus, the SiL stage allows us to validate the MPWTC system which co-operates with the existing automation system with the well-established, certified FLEX5 simulation tool prior to evaluations of the MPWTC system in HiL tests.

4.3. Hardware-in-the-loop test

After successful validations in previous testing stages, we conduct HiL tests in order to a) test the generated MPWTC code in the industrial PLC, b) evaluate the implemented communication path and c) validate the co-operation between MPWTC and automation system in their target platforms. Thus, in HiL tests, we validate the structural as well as the functional integration of the MPWTC system into the existing automation system of the real wind turbine.

Figure 4 and Figure 6 illustrate the HiL test’s software and hardware setup, respectively. We propose two HiL stages A and B, whereas stage B extends A by the wind turbine’s automation system, which is executed in a separate PLC. For the plant model, we apply the same simulation tools as in system simulations (see Figure 4). As for SiL testing, based on the Matlab/Simulink source code, we generate the code of the MPWTC system that is executable in the industrial
The HiL hardware setup comprises the real-time simulation platform, the target PLC (employing the MPWTC system) and – for HiL stage B – an additional PLC (employing the automation system). The PLCs correspond to the hardware used in the real wind turbine.

We apply HiL stage A to test the MPWTC code and its computational time in the industrial PLC. Stage A applies a standard PC to simulate the plant model and a ‘Bachmann MH230’ PLC to employ the MPWTC system. The platforms communicate via Ethernet. Figure 7 shows the computing time distribution of the MPWTC system ($\Delta T_{\text{MPC}} = 100 \text{ ms}$) in the MH230 PLC for the entire benchmark load case, it indicates mean and maximum computing times of 16.9 ms and 20.8 ms, respectively, and a standard deviation (SD) of 0.3 ms. While stage A considers the MPWTC system as independent from the automation system, in stage B, both systems co-operate with each other.

We use HiL stage B to evaluate the MPWTC behavior with respect to delay times in the real communication path and to validate the co-operation between MPWTC and automation system in their target PLCs. In this setup, we additionally apply a ‘Bachmann MC210’ PLC that employs the code of the automation system. The platforms (PC and both PLCs) again communicate via Ethernet. HiL stage B is the object of current work in order to prepare MPWTC systems for field testing.

5. Discussion

Above all, it seems pertinent to remember that the general RCP methodology can be applied to systematically develop any type of control system – independently of the considered process application or control algorithm. Likewise, both the introduced WTCF and the proposed testing stages (system simulation, SiL and HiL test) can be used to design and test any type of WTC algorithm – they are not restricted to MPWTC systems. However, in this study, we focused on the development of MPC algorithms for WTC.

In the following, with regard to the question of how to adapt the RCP methodology to prepare an MPWTC system for field testing, we discuss the introduced RCP system and the applied RCP testing stages and evaluate their usability for a systematic MPWTC development. In addition, for this particular development process, we point out several challenges identified in the course of the study.

**RCP system: WTCF as central development environment**

The WTCF, introduced in Section 3, serves as central RCP system throughout the entire development process and enables processing the RCP development steps (I)–(VI). For a systematic control development, we benefit from the reusability and modular structure of the submodels. In particular, the flexible replacing and central versioning of MPWTC submodels facilitate the systematical design and testing of such control systems. Moreover, due to different applicable wind turbine plant dynamics, the WTCF enables validating the MPWTC’s robustness.

**RCP testing stages: systematic control testing**

In Section 4, we proposed system simulations, SiL and HiL tests for systematic control testing throughout the development process – these testing stages constitute a vital part of the systematic development process.

As shown in Section 4.1, System simulation and SiL test indicate very similar control results. Arising differences in the simulation results support the identification of open issues in the design of the considered MPWTC system.

As stated in Section 2, we amended the ‘software-in-the-loop’ definition from [10]. Applying ‘software-in-the-loop’ tests according to [10] for MPWTC development would emerge considerable safety risks by involving tests in the real wind turbine with the control algorithm that is still
under test and executed in a real-time prototyping platform (which is not the target PLC). Furthermore, usually, there is no simulation model of the supervisory control available to be employed with the control system in the same prototyping platform. Thus, ‘software-in-the-loop’ testing according to [10] is not viable for testing MPWTC systems, and comprehensive HiL tests become more decisive in order to prepare for the commissioning of MPWTC systems.

For the SiL tests in Section 4.2, we use FLEX5 as certified tool for evaluating the controlled wind turbine dynamics in a well-established simulation environment for load calculations during the wind turbine design. Thus, we are able to compare the MPWTC behavior to the results of validated, established baseline control systems.

In Section 4.3, we used HiL stage A particularly to validate the execution of the generated MPWTC code in the real industrial PLC. The MPWTC’s computational time \((\text{mean} = 16.9 \text{ ms}, \max = 20.8 \text{ ms})\) with respect to the PLC’s sampling time \((\Delta T_{\text{MPC}} = 100 \text{ ms})\) indicated real-time feasibility of the MPWTC code in the Bachmann MH230 PLC as well as a reasonable PLC’s CPU load. Moreover, we proposed HiL testing according to stage B in order to test the MPWTC system with exactly the same software and hardware communication interfaces and realistic communication delay times as in the real setup in the wind turbine.

What’s more, SiL and HiL tests (in stage B) enable validating the co-operation between MPWTC and automation system in a simulation environment as well as in the target PLCs. Thus, we use these testing stages for both the functional and structural integration of the MPWTC system into the existing automation system of the real wind turbine.

**Challenges in MPWTC development**

The development of MPC algorithms for wind turbines evokes several challenges, which have to be taken into account throughout the development process.

The application of an MPC algorithm causes the following challenges:

- **Prediction model:** The prediction accuracy of future system dynamics immediately affects the control performance, since the prediction trajectories decisively determine the cost function.
- **Real-time capability:** Solving the optimization problem online leads to special requirements for the real-time capability of the control algorithm – special attention must be given to this point during the HiL tests.
- **Feasibility:** The online optimization problem has to be feasible in every operating point – special attention must be given to this point during the system simulations.
- **State observer:** The feedback of the state vector to the MPC requires the parallel use of an observer – in particular, the observability has to be verified.
- **Stability:** The proof of the stability is still an open issue in MPWTC – this issue has to be addressed for a comprehensive control validation.

The development of an WTC system poses the following challenges:

- **Existing automation system:** For field testing, the MPWTC system has to be integrated into the existing automation system. We distinguish between structural and functional integration:
  - **Structural integration:** The control system has to be integrated into the software and hardware. The software integration is mainly realized by specifying the in- and output signals of the ‘bypass interface’ module and its implementation (Figure 1). The hardware integration is realized in HiL stage B by using an additional PLC that employs the MPWTC system and communicates via Ethernet with the PLC that executes the existing automation system.
– Functional integration: The MPWTC system delivers command signals which have to be processed by the bypass interface module. Internally, the logic of this interface must decide whether to use the internal signals from the existing control system or those from the MPWTC system.

• Supervisory control: Comprehensive simulation studies of the controller performance over the full range of operating conditions demand specifications e.g. of the references or process constraints; thus, the supervisory control is essential to be considered in the control testing.

• Standardized load cases: Wind turbine design incorporates evaluating the design load cases and load sets according to IEC 61400-1; control testing with these load cases is essential when preparing field testing.

6. Summary and conclusion
The considered RCP development process with its testing stages offers an appropriate approach and a potential guideline for the systematic and flexible design and testing of MPWTC systems. The introduced ‘WTCF’ serves as central RCP system and provides a suitable environment for a comprehensive MPWTC development (see Section 3). With this framework, we particularly ensure conducting system simulations, SiL and HiL tests in an integrated environment using the same model instances with a continuous tool chain throughout the entire development process (see Section 4). Thus, this work advances the field of WTC by proposing a systematic development procedure for MPWTC systems.

The study demonstrates that the general RCP methodology of [10] only has to be extended slightly to prepare MPWTC algorithms for its integration into the wind turbine automation system. To this end, on the one hand, we modified the ‘software-in-the-loop’ definition from [10] (see Section 2 and 4.2), on the other hand, we introduced the development steps ‘functional integration’ and ‘structural integration’ (see Section 4.2 and 4.3). Thus, we are able to systematically develop MPWTC algorithms co-operating with the existing automation system and to validate the execution of the generated MPWTC code in the wind turbine’s PLC.

Due to emerging safety risks by testing a control algorithm that is executed in a real-time prototyping platform (which is not the target PLC) in the real wind turbine, comprehensive HiL tests become vital in order to prepare for the commissioning of an MPWTC system. To this end, we propose ‘HiL stage B⃯’, which extends the standard HiL structure of ‘stage A⃯’, to conduct comprehensive evaluations of the functional and structural integration of the MPWTC algorithm (compare Section 4.3).

Throughout the MPWTC development process, several challenges have to be taken into account that are caused whether by applying the MPC algorithm or by facing the WTC problem (see listing in Section 5). The presented list does not intend to be complete but reflects the current state of work identified in the course of this study – it can certainly be extended with further aspects identified by future discussions and ongoing research.

The results of the system simulation (Section 4.1) and SiL test (Section 4.2) demonstrate the MPWTC’s capability of controlling the considered ‘W2E-120/3.0fc’ wind turbine in the entire operation range. The HiL test (Section 4.3) verify the real-time feasibility of the considered MPWTC system in the real wind turbine’s PLC. Additional validations of the MPWTC code in HiL stage B⃯ will be investigated in future studies in order to prepare the MPWTC system for field tests in the real wind turbine.

References
[1] van Kuik G A M et al. 2016 *Wind Energ. Sci.* 1 1–39 URL https://doi.org/10.5194/wes-1-1-2016
[2] Lio W H, Rossiter J A and Jones B L 2014 2014 UKACC International Conference on Control (CONTROL) (Loughborough, UK: IEEE) pp 673–678 URL https://doi.org/10.1109/CONTROL.2014.6915220
[3] Gros S and Schild A 2017 *International Journal of Control* **90** 2799–2812 URL https://doi.org/10.1080/00207179.2016.1266514

[4] Ritter B, Schild A and Konigorski U 2016 Making nonlinear state estimation techniques ready for use in industrial wind turbine control systems poster presentation at WindEurope Summit 2016 (Hamburg, Germany)

[5] Vukov M, Gros S, Horn G, Frison G, Geebelen K, Jørgensen J B, Swevers J and Diehl M 2015 *Control Engineering Practice* **45** 64–78 URL https://doi.org/10.1016/j.conengprac.2015.08.012

[6] Jassmann U, Dickler S, Zierath J, Hakenberg M and Abel D 2016 *J. Phys.: Conf. Ser.* **753** 052021 URL https://doi.org/10.1088/1742-6596/753/5/052021

[7] Zierath J, Jassmann U, Dickler S and Abel D 2016 *4th Joint International Conference on Multibody System Dynamics* (Montreal, Canada: IMSD)

[8] Hand M M and Balas M J 2000 *Wind Engineering* **24** 169–187 URL https://doi.org/10.1260/0309524001495549

[9] Stol K A and Fingersh L J 2004 Wind turbine field testing of state-space control designs: August 25, 2003–November 30, 2003 (United States) URL https://doi.org/10.2172/15009600

[10] Abel D and Bollig A 2006 *Rapid Control Prototyping* (Berlin and Heidelberg: Springer) URL https://doi.org/10.1007/3-540-29525-9

[11] Taubert M, Clauß S, Freudenberg H, Keil A, März M, Moser W and Wulf H O 2011 Wind Turbine Design Codes: Eine Validierung von alaska/Wind mit BLADED, FAST und FLEX5 [in German] (Chemnitz, Germany)

[12] Jonkman J M and Buhl, Jr, M L 2005 Fast user’s guide - updated August 2005 (United States) URL https://doi.org/10.2172/15020796

[13] Jassmann U, Zierath J, Dickler S and Abel D 2016 *IEEE Conference on Control Applications (CCA)* (Buenos Aires, Argentina: IEEE) pp 1368–1373 URL https://doi.org/10.1109/CCA.2016.7587997

[14] Zierath J, Rachholz R and Woernle C 2016 *Wind Energy* **19** 1201–1222 URL https://doi.org/10.1002/we.1892

[15] Dickler S, Wiens M, Thönnissen F, Jassmann U and Abel D 2019 *18th European Control Conference (ECC)* (Naples, Italy: IEEE) pp 3346–3352 URL https://doi.org/10.23919/ECC.2019.8795826

[16] Ferreau H J, Kirches C, Potschka A, Bock H G and Diehl M 2014 *Math. Prog. Comp.* **6** 327–363 URL https://doi.org/10.1007/s12532-014-0071-1