Design optimization of a wind turbine blade under non-linear transient loads using analytic gradients

Iori, Jenna

Published in:
Journal of Physics: Conference Series

Link to article, DOI:
10.1088/1742-6596/1618/4/042032

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Design optimization of a wind turbine blade under non-linear transient loads using analytic gradients

To cite this article: Jenna Iori 2020 J. Phys.: Conf. Ser. 1618 042032

View the article online for updates and enhancements.
Design optimization of a wind turbine blade under non-linear transient loads using analytic gradients

Jenna Iori
DTU Wind Energy, Technical University of Denmark, Frederiksborgvej 399, 4000 Roskilde, Denmark
E-mail: jenio@dtu.dk

Abstract. This work presents and compares two formulations for the co-design optimization of a wind turbine blade under non-linear transient loads: the Nested Analysis and Design (NAND) and the Simultaneous Analysis and Design (SAND) approaches. Analytic sensitivities are used in order to ensure the convergence of the optimization within reasonable computational resources. The two formulations are compared on a mass minimization problem with dynamic constraints, solved with the interior-point method in IPOPT, for a gust input and a turbulent input. Results show that the NAND and SAND approaches converge towards the same optimum with similar performances. The SAND approach benefits from a simpler design sensitivity analysis and a sparse Jacobian of the constraints.

1. Introduction

One key challenge of wind turbine blade optimization is to take into account the complexity of loads on the blade and to estimate the corresponding gradients with good accuracy. During its lifetime a blade is subjected to a wide variety of loads due to the diversity of operational conditions. Existing optimization frameworks in the field represent these loads in a way that is easier to manipulate for the optimizer. Several methods are used, such as analysis in the frequency domain, use of load envelopes based on dynamic simulations or reduced load cases. In addition, there are strong couplings between the different disciplines involved in the analysis, especially for modern blades that tend to be longer and more flexible. In the optimization framework HAWTOpt2 [1], the aero-elastic couplings are taken into account by using the aero-elastic tool HAWC2 [2] to run dynamic simulations and estimate the gradient of the optimization constraints. Another well-known optimization framework, Cp-max [3], formulates optimization sub-problems to tackle the uncoupled structural and aeroelastic aspects. The aero-elastic coupling is captured by doing iterations until convergence is reached. When it comes to control design in the optimization, a common practice in the field of wind energy is to update the control law or control parameters at each optimization iteration by solving a nested problem [4] [5]. Deshmukh and Allison [6] compared this co-design approach with a classic sequential design approach on a wind turbine optimization problem. They show that with the co-design approach, the optimizer has more freedom in the design space and consequently gives a better optimum design.

In wind turbine blade optimization, most softwares are using a Nested Analysis and Design (NAND) formulation, where the analysis is nested in the optimization. The analysis is then conducted by an external analysis software. Studies have investigated the possibility of another
formulation, the Simultaneous Analysis and Design (SAND) [7] or all-at-once formulation. In this formulation, the design parameters and the state parameters of the system are optimized simultaneously. Instead of using an analysis software to solve the state equations, they are included as equality constraints in the optimizer. With the SAND approach, it is possible to use fine-grain parallelism and take advantage of the sparsity of the jacobian when considering non-linear problems [8]. Allison et al [9] explain that this approach allows to reformulate and solve efficiently control co-design problems that are challenging to solve in the nested approach.

This work aims at presenting and comparing the NAND and SAND formulations for the co-design optimization of wind turbine blades under transient non-linear loads. The aero-servo-elastic design problem contains a high degree of coupling between all the disciplines, a type of problem that is computationally expensive to solve and for which the SAND approach is potentially better suited than the NAND approach. The results presented in this work extend the findings of a MSc thesis [10] by using a different optimization software and comparing the two approaches on additional test cases. The interested reader may refer to this document for additional information on the method.

2. Analysis and Model
In a fully coupled aero-servo-elastic problem, the aerodynamic properties $x_a$, the structural properties $x_s$, the control properties $x_c$ and the state of the blade $u$ are linked together through the state equation for each discipline (equation of motion, wake model and control law for example) and through the couplings. Here, state variables refers to any quantity describing the behavior of the system. Examples are displacement, stress, aerodynamic torque, etc. It is possible to formulate three sets of equations, one for each discipline involved. Each equation depends on all properties and on the state variables in the general case, assuming full coupling. The three equations can be written in a residual form, where the indexes $s$, $a$ and $c$ correspond to the structural dynamics, the aerodynamics and the controller, and $x = [x_a, x_s, x_c]^T$.

$$ r_s(x, u) = 0 \quad r_a(x, u) = 0 \quad r_c(x, u) = 0 $$

In this work, steady state equations and dynamic equations are considered. The residual equations and the state variables are typically divided into a steady state part and a dynamic part at each time steps, i.e. $r_s = [r_{s, steady}^1, r_{s, steady}^2, ..., r_{s, steady}^{N_t}]^T$, $u = [u_{steady}, u_1, u_2, ..., u_{N_t}]^T$ where $r_{s, steady}$ and $u_n$ represent the residual equation and the state at the time step $n$ and $N_t$ is the total number of time steps. In this study, simple analysis is used in order to reduce the complexity of the analytic design sensitivity. The next paragraphs outline the different aspects of the analysis model. The model is presented with more details in [10].

Structural dynamics The structural dynamics are calculated through 3 parts: the computation of the mass, stiffness and damping matrices of the blade, the computation of the aerodynamic force and solving the structural state equation. The mechanical properties of the structure are modeled as an assembly of two spar caps and an ellipsoidal skin. The structure is analyzed using the Finite-Element Method, with Timoshenko beam elements. From there, the cross section properties of the blade are computed and assembled in the total blade mass, damping and stiffness matrices. The aerodynamic loads are calculated using the information of the blade twist, the blade velocity, the control pitch angle and the aerodynamic induced wind. 2D inviscid aerodynamic coefficient used are adapted from the FFA-W3-xxx airfoils series. Then, the state equation is solved. The Newmark-beta method for numerical integration [11] is used for the time integration.
Aerodynamics The aerodynamics model of the blade represent the impact of the wake on the system. It is represented in this work by an induced wind computed using the classic Blade Element Momentum (BEM) method presented in [12]. The equation for calculating the components of the induced wind depends on the aerodynamic loads, and is non-linear.

Controller The controller implemented is a variable speed, variable pitch PI controller based on the basic DTU Wind Energy controller [13]. The only control region considered is for wind speed above rated wind-speed, for reasons of simplicity. The controller uses the information of the aerodynamic torque and the previous pitch angle and rotational speed for setting the next value of the pitch angle and rotational speed.

Solving the state equations The residual equations $r_a$ and $r_s$ are non-linear and require iterations to be solved. Figure 1 shows the process in a simplified way at a given time step. Let $u_{n+1}$ be the set of state variables corresponding to the time step $n+1$. This variable set can be divided in two: one part for the variables that are non-linear in the structural equation $u_{n+1}^s$, and another part for the variables that are non-linear in the aerodynamic equation $u_{n+1}^a$. The structural and aerodynamic residuals at time step $n + 1$ are dependent on the design variables, and the state at the previous and current time steps.

\[
\begin{align*}
    r_{s}^{n+1}(x, u_{n+1}^s, u_{n+1}^a, u_n) &= 0, \\
    r_{a}^{n+1}(x, u_{n+1}^s, u_{n+1}^a, u_n) &= 0
\end{align*}
\]  

The structural variables are updated using Newton-Raphson iterations [14]. At each structural iterations, the aerodynamic variables are updated in a nested fixed-point iteration loop.

![Figure 1. Simplified representation of the method to solve the structural and aerodynamic state equations for the time step $n + 1$: one fixed-point iteration loop for solving the aerodynamic residual is nested into a Newton-Raphson iteration for solving the structural residual.](image)

3. Optimization

The NAND and SAND approaches are two possible formulations of an optimization problem. The difference between the two lies in how the state equations are solved and how the state variables are considered. In the general case, the objective function and the constraints depend on the design variables and on some state variables, for example the tip displacement or the aerodynamic torque. In the nested approach, each function evaluation would require to solve the state equation in order to know what are the state variables $u(x)$ corresponding to a given set of design variables $x$. In the simultaneous approach, the state variables are considered as optimization variables. Instead of solving the state equations, an equality constraint is added to
the optimization problem in order to ensure the state variables chosen by the optimizer satisfy the state equations. Equations (3) and (4) report the NAND and the SAND formulations respectively for the same optimization problem, with \( r = [r_s, r_a, r_c]^T \) and \( N_x, N_u \) represent the number of entries of \( x \) and \( u \).

\[
\begin{align*}
\text{minimize} & \quad \tilde{f}(x) = f(x, u(x)) \\
\text{subject to} & \quad g(x, u(x)) \leq 0 \\
& \quad h(x, u(x)) = 0 \\
\text{with} & \quad u \in \mathbb{R}^{N_u} \text{ s.t. } r(u, x) = 0
\end{align*}
\]

(3)

\[
\begin{align*}
\text{minimize} & \quad f(x, u) \\
\text{subject to} & \quad g(x, u) \leq 0 \\
& \quad h(x, u) = 0 \\
& \quad r(u, x) = 0
\end{align*}
\]

(4)

In this work, the optimization aims at finding a lighter blade with similar properties, i.e. where the extrema of the operational outputs are the same compared to the initial design. The design variables chosen are the chord along the blade because it influences both the stiffness of the blade and the aerodynamic loads, and the gains of the PI controller (gain reduction \( k_K \), integral gain \( k_I \) and proportional gain \( k_P \)) because they can change the evolution of the operational outputs in time. The relative thickness of the blade is set constant during the optimization and as such, the aerodynamic efficiency is maintained. The objective function chosen is the mass of the blade. Time dependent inequality constraints are set on the power \( P \), the tip displacement \( \delta \) and the pitch angle \( \theta \).

In theory, more design variables and constraints can be included. For the scope of this study, a small and manageable optimization problem is chosen to compare the two formulations.

For the SAND formulation, state variables are added as optimization variables: the position, velocity and acceleration of the blade, the load applied to the blade, the induced velocity, the pitch angle and the integral component of the pitch angle, the rotational speed and the power. The choice of which state variables to include is important: if there are too many, the problem becomes too big. In practice, a state variable should be added as an optimization variable if it is part of a non-linear equation, because it simplifies the sensitivity analysis. Even though the force and power are not directly part of a non-linear equation, they are added as optimization variables because it makes the computation of the gradient easier. Constraints are added to enforce the state equations, and to make sure that the force and the power state variables have correct values.

4. Design Sensitivity Analysis

The design sensitivity analysis is done using analytic gradients. The direct method is chosen as the adjoint method seems to be more complex for non-linear time dependent problems [15]. The same method is used for the design sensitivity analysis with regards to the control parameters and the chord. All gradients are checked thoroughly and compared with central finite-difference. The methods for design sensitivity in the NAND and SAND approaches are detailed below.

4.1. NAND approach

Consider a function depending on the design variables and the state variables, \( \eta(x, u(x)) \). The derivative of the function with regards to \( x_i, i = 1, ..., N_x \) is obtained with the chain rule.

\[
\frac{\text{d} \eta}{\text{d} x_i} = \frac{\partial \eta}{\partial x_i} + \frac{\partial \eta}{\partial u} \cdot \frac{\text{d} u}{\text{d} x_i}
\]

(5)

The partial derivatives of \( \eta \) are straightforward from the analytic expression of the function. The last term can be derived from the state equation linking \( u \) to \( x \), as shown in Equation (6).

\[
r(x, u(x)) = 0 \quad \Rightarrow \quad \frac{\text{d} r}{\text{d} x_i} = 0 \quad \Rightarrow \quad \frac{\partial r}{\partial x_i} + \frac{\partial r}{\partial u} \cdot \frac{\text{d} u}{\text{d} x_i} = 0
\]

(6)
The term \( \frac{du}{dx} \) can then be obtained by solving a linear system. However in practice forming the linear system, i.e. creating the matrices \( \frac{\partial r}{\partial x_i} \) and \( \frac{\partial r}{\partial u} \) is not straightforward, because the system is time dependent and non-linear. Consider the derivative of the state variable \( \frac{du_{n+1}}{dx_j} \). Using the chain rule on Equation (2), the derivative of the state variables can be isolated to form Equation (7).

\[
\begin{bmatrix}
\frac{\partial r_{s}^{n+1}}{\partial u_{n+1}^{s}} & \frac{\partial r_{s}^{n+1}}{\partial u_{n+1}^{s}} \\
\frac{\partial r_{a}^{n+1}}{\partial u_{n+1}^{a}} & \frac{\partial r_{a}^{n+1}}{\partial u_{n+1}^{a}}
\end{bmatrix}
\begin{bmatrix}
\frac{du_{n+1}^{s}}{dx_j} \\
\frac{du_{n+1}^{a}}{dx_j}
\end{bmatrix} = -
\begin{bmatrix}
\frac{\partial r_{s}^{n+1}}{\partial x_j} + \frac{\partial r_{s}^{n+1}}{\partial u_{n}^{s}} \cdot \frac{du_{n}}{dx_j} \\
\frac{\partial r_{a}^{n+1}}{\partial x_j} + \frac{\partial r_{a}^{n+1}}{\partial u_{n}^{a}} \cdot \frac{du_{n}}{dx_j}
\end{bmatrix}
\tag{7}
\]

4.2. SAND approach
In the SAND approach, the differentiation is straightforward. The optimization variables are the design variables and the state variables. There is no need to differentiate the state variables with regards to the design variables. Hence, for \( x_i, \ i = 1, ..., N_x, \ u_j, \ j = 1, ..., N_u \) one has:

\[
\frac{d \eta}{dx_i} = \frac{\partial \eta}{\partial x_i}, \quad \frac{d \eta}{du_j} = \frac{\partial \eta}{\partial u_j}
\tag{8}
\]

Thus, the coupling effects between disciplines are easily accounted for by knowing how the state variables and design variables relate to each other. Also, the jacobian of the constraint is large and sparse (see Figure 2). The optimizer can take advantage of this special structure for a more efficient optimization. Finally, the SAND formulation allows to consider each time step independently from one another. From the optimizer point of view, each optimization variable is considered independently and then the constraints link them together. As a consequence, the jacobian of the constraints can be computed using fine-grain parallelism.

5. Implementation
The implementation of the objective function, constraints, and gradient evaluations is done using MATLAB. The optimization is done using IPOPT 3.12.9 [16] through the OPTI framework [17]. IPOPT is chosen for this work as the software is widely used in the domain of structural optimization, is adapted for large sparse non-linear problems and is updated on a regular basis. All optimizations are run using a workstation with an Intel i7–8665U processor running at 1900 MHz using 16 MB of RAM.

One of the key challenges in the implementation of the SAND approach is the scaling of the state variables as optimization variables. They have various scales: the aerodynamic force is around \( 10^4 \) N/m, the pitch angle is around 0.1 rad and the power is around \( 10^7 \) W. It is important to use a correct scaling, as it can influence greatly the behavior of the optimizer [8].

Figure 3 shows the evolution of the stationarity during optimization for three scaling choices of the state variables: (i) no scaling, (ii) reference scaling and (iii) alternative scaling. The scaling (ii) is the one used in the results presented in this article and scales the aerodynamic force and the power. The alternative scaling is similar but add scaling on the pitch angle and the integral pitch angle. The scaling (i) and (iii) present poor performances, and do not reach convergence within 50 iterations contrary to the scaling (ii). Table 1 reports the differences between the three scales. This suggests that for a successful implementation of an optimization in the SAND formulation, the scaling of the optimization needs to be calibrated with special care. For this study, the reference scaling was decided by trial and error.
In terms of the implementation effort, both formulations have benefits and drawbacks. The SAND approach lacks flexibility: if the constraints are modified and a new state variable needs to be introduced, the entire code needs to be adapted because the format of the optimization variable has changed. However, modifying constraints is straightforward, because it only requires changing the bounds on the state variables. When it comes to the gradient derivation, specialized optimization frameworks exist to reduce the implementation effort required for both formulations.

6. Results and Discussion

The initial blade design considered in this problem is based on the DTU 10MW reference wind turbine [18]. Two types of test cases are considered: a gust input and a turbulent input. The characteristics of the different optimization problems are reported in Table 2. Since the SAND approach considers the state variables as optimization variables, the number of optimization variables and the number of constraints is larger than for the NAND approach. The choice of a low number of time steps is motivated by the limited memory capacity of the implementation of IPOPT. The time increment is $\Delta t = 0.2s$. The desired convergence tolerance of the optimizer is $10^{-7}$. Table 3 reports the bounds on the design variables.
Table 2. Optimization problem statistics

<table>
<thead>
<tr>
<th>Problem</th>
<th>Variables</th>
<th>Constraints</th>
<th>$N_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAND - Gust</td>
<td>39949</td>
<td>40707</td>
<td>52</td>
</tr>
<tr>
<td>NAND - Gust</td>
<td>13</td>
<td>130</td>
<td>52</td>
</tr>
<tr>
<td>SAND - Turb. input</td>
<td>46093</td>
<td>46851</td>
<td>60</td>
</tr>
<tr>
<td>NAND - Turb. input</td>
<td>13</td>
<td>136</td>
<td>60</td>
</tr>
</tbody>
</table>

Table 3. Variable bounds

<table>
<thead>
<tr>
<th>Variables</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chord</td>
<td>$0.8c_{ref} \leq c \leq 1.2c_{ref}$</td>
</tr>
<tr>
<td>$k_K$</td>
<td>$0.1 \leq k_K \leq 0.5$</td>
</tr>
<tr>
<td>$k_I$</td>
<td>$0.1 \leq k_I \leq 0.5$</td>
</tr>
<tr>
<td>$k_P$</td>
<td>$0.1 \leq k_P \leq 1.0$</td>
</tr>
</tbody>
</table>

Table 4. Constraints on the operational parameters

<table>
<thead>
<tr>
<th>Gust constraints</th>
<th>Time window</th>
<th>Turbulent constraints</th>
<th>Time window</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(t) \geq P_{\text{ref}}_{\text{min}}$</td>
<td>$t \in [2s, 10.2s]$</td>
<td>$P_{\text{ref}}<em>{\text{min}} \leq P(t) \leq P</em>{\text{ref}}_{\text{max}}$</td>
<td>$t \in [5.2s, 12s]$</td>
</tr>
<tr>
<td>$\delta(t) \leq \delta_{\text{ref}}_{\text{max}}$</td>
<td>$t \in [2s, 7.2s]$</td>
<td>$\delta(t) \leq \delta_{\text{ref}}_{\text{max}}$</td>
<td>$t \in [5.2s, 12s]$</td>
</tr>
<tr>
<td>$\theta_p(t) \leq \theta_{p,\text{max}}$</td>
<td>$t \in [2s, 10.2s]$</td>
<td>$\theta_p(t) \leq \theta_{p,\text{max}}$</td>
<td>$t \in [5.2s, 12s]$</td>
</tr>
</tbody>
</table>

6.1. Wind input and constraints

The gust input formulation is based on the recommendation of the IEC 61400-1 standard. The time window is $T = 10.4s$, where the gust is 6s long and has a speed increase of $1m.s^{-1}$. Different reference wind speeds $V_o$ are considered: $12m.s^{-1}$, $14m.s^{-1}$, $16m.s^{-1}$ and $18m.s^{-1}$. The initial boundary condition is set such that the steady state equation is satisfied. Constraints are enforced during the gust itself. The turbulent input is based on the MANN model [19] with a turbulence intensity of 7.4% and a reference wind speed $V = 14m.s^{-1}$. The time window is $T = 12s$. The initial condition is fixed to correspond to the steady state of the initial design. Constraints are enforced during the second half of the time window, in order to allow a possible transient adjustment. The constraints are summarized in Table 4, where the reference value correspond to the extremum of the corresponding signal for the initial design.

6.2. Example of optimum design

The input considered here is a gust with a reference wind speed of $12m.s^{-1}$ over 52 time steps corresponding to 10.4s. The two optimization formulations are implemented and compared. The optimal design is 1.89 % lighter than the initial design. The constraints on the tip displacement and on the power are active and the control parameters and the chord reach their lower bounds. The NAND and SAND formulations give the same optimal design, as shown in Figure 4 and Table 5. The relative difference on the design parameters is below 0.006 %. Figure 5 shows the value of the optimum tip displacement and the difference between the value computed through the SAND approach and through the analysis code used in the NAND approach. The state variables obtained with the SAND approach are close to identical to what is obtained with the analysis code. The absolute difference between the two signal is below $2 \cdot 10^{-6}mm$.

The SAND approach converges in 28 iterations corresponding to 29 objective function calls, 29 constraint evaluations and a run-time of 2611 s. The NAND approach converges in 17 iterations corresponding to 18 objective function calls, 18 constraint evaluations and a run-time of 3589 s. It is expected to have more iterations with SAND, because the optimizer is in charge of solving the state equations in this case.
Table 5. Optimal mass and control parameters for a gust with $V_o = 12\text{m.s}^{-1}$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Initial value</th>
<th>Optimum - SAND</th>
<th>Optimum - NAND</th>
<th>Rel. difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass $m$ [T]</td>
<td>33.819</td>
<td>33.179705</td>
<td>33.179717</td>
<td>0.00004 %</td>
</tr>
<tr>
<td>$k_K$</td>
<td>0.244</td>
<td>0.182215</td>
<td>0.182203</td>
<td>0.0060 %</td>
</tr>
<tr>
<td>$k_I$</td>
<td>0.141</td>
<td>0.100000</td>
<td>0.100000</td>
<td>0%</td>
</tr>
<tr>
<td>$k_P$</td>
<td>0.524</td>
<td>1.00000</td>
<td>1.000000</td>
<td>0%</td>
</tr>
</tbody>
</table>

Figure 4. Optimal chord for a gust input at $V = 12\text{m.s}^{-1}$ and relative difference between the NAND and SAND formulations

Figure 5. Optimal tip displacement for a gust input at $V = 12\text{m.s}^{-1}$ and difference between the NAND and SAND formulations

6.3. Performance and computational resources

In order to compare the two formulations, 6 test cases are run: 4 gust cases for wind speeds at $12\text{m.s}^{-1}$, $14\text{m.s}^{-1}$, $16\text{m.s}^{-1}$ and $18\text{m.s}^{-1}$ respectively and 2 turbulent cases with different seeds at $V_o = 14\text{m.s}^{-1}$. Figure 6 shows the evolution of the constraint violation and the stationarity during the optimization process for all test cases. Both figures show that the NAND approach converges towards the optimum with fewer iterations than the SAND approach. All test cases converge in less than 43 iterations for the SAND formulation and less than 25 iterations for the NAND approach. In addition, there is no particular difference between the turbulent cases and the gust cases. Overall, the optimization under the SAND formulation behave in a similar way that with the NAND formulation: the stationarity and the constraint violation decrease steadily until convergence is reached, the optimization does not stall. Thus, the optimizer is able to handle the non-linearities of the problem in the SAND formulation with good performances.

The run time for the two approaches is of the same order of magnitude. Since the implementation of the analysis and design sensitivity code is not optimized for speed, it is not relevant to compare the run-time more precisely. However, the time spent in IPOPT is in average 7% of the total run-time for the NAND formulation and 88% for the SAND formulation. This indicates that the
code in the SAND approach can potentially be optimized by using a specialized implementation of IPOPT or a better linear solver.

![Graph](image-url)

**Figure 6.** Constraint violation and stationarity during the optimization for the NAND and SAND formulations for different test cases.

These results indicate that the SAND formulation can be considered as a serious alternative to the NAND formulation. The SAND formulation simplifies the design sensitivity analysis which can be beneficial in terms of the computational effort.

7. Limitations and future work
There are several important limitations to this study. First, the analysis model and the constraints on the blade are not representative of a real blade and the design requirements in the field of wind energy. The optimum design presented in this article is reported in order to show the difference between the two different formulations but not to present a novel blade design. In addition, the optimization problem is simplified and do not necessarily represent the complexity of a realistic wind turbine blade optimization problem. This should be addressed in future work by implementing more functionalities in the analysis code and by running test cases including several types of wind input. In addition, future work should focus on the possibilities to use parallelism with the SAND approach. It is an important benefit of this approach: the structure of the optimization problem allow to calculate the constraints at each time step independently, contrary to the NAND approach where the time integration require
each time step to be considered one after the other. Finally, it would be interesting to study whether the SAND approach scales well with the length of the simulation. The increase in the number of variables necessarily increases the size of the matrices manipulated. This can bring challenges that do not exist in the NAND approach.

8. Conclusion
This study presents and compares two possible formulations of the same co-design optimization problem for wind turbine blades submitted to dynamic loads. The design sensitivity analysis for each formulation is reported, showing that the SAND approach simplifies the process. Two types of case study are studied: a gust input and a turbulent input. Results show that the two formulations converge to the same optimal design, and the state equations are correctly solved in the SAND approach. In addition, the performances of the two formulations are compared over 6 different test cases, showing similar optimization path with the SAND approach requiring a couple more iterations. These results demonstrate the equivalence of the two formulations for the optimization problem considered. The simplicity of the design sensitivity analysis in the SAND formulation suggests that this approach should be considered for future work using analytic gradients in dynamic optimization problem.

Acknowledgments
This work was funded by the Technical University of Denmark. The author would like to thank Mathias Stolpe and Michael McWilliams from DTU for their valuable feedback on the manuscript. In addition, sincere thanks are addressed to two reviewers for their comments and suggestions that improved the quality of this article.

References