energies

Article

Implementation of Variable Blade Inertia in OpenFAST to Integrate a Flywheel System in the Rotor of a Wind Turbine

Laurence Alhrshy

Wind Energy Technology Institute, Flensburg University of Applied Sciences, Kanzleistraße 91-93, 24943 Flensburg, Germany; laurence.alhrshy@hs-flensburg.de; Tel.: +49-461-48161-407

Abstract: In this paper, the integration of the dynamic behavior of the flywheel system into the load simulation tool OpenFAST is presented. The flywheel system enables a wind turbine to vary the inertia of its rotor blades to control the power production and, most importantly, to affect the vibratory behavior of wind turbine components. Consequently, in order to simulate the behavior of a wind turbine with a flywheel system in its rotor, the variable blade characteristics need to be considered in the load simulation tool. Currently, computer-aided engineering tools for simulating the mechanical loads of wind turbines are not designed to simulate variable blade inertia. Hence, the goal of this paper is to explain how variable inertias of rotor blades are implanted in such load simulation tools as OpenFAST. OpenFAST is used because of it is free, publicly available, and well documentation. Moreover, OpenFAST is open source, which allows modifications in its source code. This add-on in the load simulation is applied to correct rotor mass imbalance. It can also be applied in many cases related to the change in the inertia of wind turbine rotor blades during its operation as, for example, atmospheric ice accretion on the blades, smart blades, etc.

Keywords: flywheel; flywheel torque; load simulation; mechanical loads; OpenFAST; rotor imbalance; wind turbine

check for updates

Citation: Alhrshy, L. Implementation of Variable Blade Inertia in OpenFAST to Integrate a Flywheel System in the Rotor of a Wind Turbine. *Energies* **2021**, *14*, 2783. https://doi.org/10.3390/en14102783

Academic Editor: Andrzej Bielecki

Received: 2 March 2021
Accepted: 10 May 2021
Published: 12 May 2021

1. Introduction

The general idea of the flywheel (FW) system is based on the inertia variation of wind turbine (WT) rotor blades. This is done by shifting a fluid mass between two-piston accumulators in the rotor blade of a WT. The physical description of such an energy storage system is presented in previous works of Jauch et al. and this covers the kinetic [1] and hydraulic [2] aspects.

The initial application of the FW system was developed to allow a certain variation of the electrical power of the WT, independent of the prevailing wind conditions [3]. Meanwhile, the impact of the FW system on the vibratory behavior of the WT and on the natural frequencies of the rotor blades has become one of the most important applications of this system, because of its influence on the operation life of the WT [4].

In order to analyze the impact of the FW system on the vibratory behavior of the WT components, the dynamic behavior of the FW system should be first implemented in the load simulation tool. Due to the fact that all state-of-the-art load simulation tools for WTs are unable to represent variable rotor blade inertias as they happen in the FW system, the dynamic behavior of the FW system cannot be represented without changing the source code of these load simulation tools. Therefore, all the previous mechanical load simulations, either for the conventional steel FW accumulators design [5] or for the flexible carbon fiber-reinforced plastics (CFRP) FW accumulators design [4], could only be done under two conditions: the FW statically fully charged and the FW statically fully discharged.

The previous load simulations were done for several FW configurations. These configurations were derived via a methodology, which was used to determine the positions and dimensions of the FW accumulators into the rotor blades [6]. However, since the goal of this paper is not to analyze the load impact of all the CFRP FW configurations published in previous work by the author [4], the load analysis is limited in this paper to only one relevant FW configuration. This configuration is the same one used by the first eigenmodes simulation tool of a WT [7]. This tool is developed for control algorithm design at the Wind Energy Technology Institute (WETI) of Flensburg University of Applied Sciences, Germany. In the following, the considered FW configurations are named "types" and are identified with Roman numerals, as introduced in Hippel et al. [6]. Hence, the properties of the CFRP position accumulators of the FW type VI, mentioned in Table 1, are used to implement the variable blade inertia.

Table 1. Properties of the CFRP piston accumulators of FW type VI.

No. of Tip Accums	Rotor Radius m	Total Length of Tip Accums mm	Stationary Weight of Tip Accums kg	Total Length of Root Accum mm	Stationary Weight of Root Accum kg	Fluid Weigh kg
2	Each at 46.6	5047 and 4923	402.47	4437	276.86	925.46 [1]

[1] The fluid mass of 925.46 kg is shifted between the root and tip accumulators.

The stationary masses in Table 1 are added to the distributed mass densities of the rotor blade regarding the total length of the root and tip accumulators. For the Sandia 61.5 m rotor blade design (Sandia National Lab., Albuquerque, USA) [8] that is used in this paper, where the mechanical properties of the rotor blades are distributed over 17 elements, the stationary masses of the root and tip accumulators are added to the first two elements and to the 11th, 12th, and 13th elements, respectively. Whereas the stationary masses of the tip and root accumulators are considered as constant masses, the fluid mass is the mass that is varying along the root and tip elements. Varying the fluid mass is determined by so-called charge-index-factors (K). These factors can be steplessly varied between the limit values of 0 and 1 for the discharged and charged states, respectively. If K in one blade equals 0, the FW is fully discharged and the total fluid mass is added to the root elements of this blade and, if K equals 1, the FW is fully charged and the total fluid mass is added to the tip elements of the same blade.

Due to the public availability of the 5 megawatt (MW) reference WT, and the extensive use of its data in studies by the wind energy research community as a system that represents the current and future state-of-the-art offshore systems, the FW system was adapted to this WT. The Sandia 61.5 m rotor blade design is also applied in the original design of the 5 MW reference WT developed by the National Renewable Energy Laboratory (NREL), USA [9].

First of all, before discussing the code adjustment of a load simulation tool to implement the dynamic behavior of the FW system, it must be decided in which load simulation tool this adjustment should be performed. FLEX5 [10], FAST [11], and Bladed [12] are all recognized WT load simulation tools which are used by the WT industry. However, in this paper, OpenFAST is selected because of it is free, publicly available, and well documented. OpenFAST is a computer-aided engineering tool, which is developed by NREL for simulating the coupled dynamic response of WTs [13]. This software is also used by thousands of WT designers, manufactures, consultants, certifiers, researchers, students, and educators all over the world [11]. Moreover, OpenFAST has the advantages of a self-sustaining community-developed software by working with OpenFAST on a GitHub repository. This feature offers a good opportunity to communicate with the NREL OpenFAST team and other developers throughout the OpenFAST community [14]. All of these advantages make it obvious that OpenFAST is a suitable load simulation tool to implement the dynamic behavior of the FW system in WT. However, the high complexity of the structure of OpenFAST requires a good understanding of the architecture of this program before any modifications to its source code. Hence, the architecture of OpenFAST is briefly introduced in the next section. More details about the program are described in a previous project report of the author [15].

This paper begins with the introduction about the program structure, which helps to understand where and how the variable blade inertias are implemented in the source code of OpenFAST. Subsequently, the computation and integration of the resulting loads from the change in rotor blade inertia are discussed in two following sections. Consequently, the modified code is tested using a simple simulation scenario of a WT without aerodynamic loads and controllers. This test aims to verify the implementation correctness of the modified code and to illustrate the impact of the variable blade inertia on the WT rotor speed behavior. Finally, the applicability of the FW system to correct a rotor mass imbalance is tested during the WT operation with enabled aerodynamic and controller modules.

2. Architecture of OpenFAST

OpenFAST is a multi-physics, multi-fidelity tool for simulating the coupled dynamic response of wind turbines. The theory of OpenFAST is based on the coupled interactions of several computational modules which are responsible for structural dynamics, aerodynamics, control and electrical system dynamics, and hydrodynamics for offshore structures to enable coupled nonlinear aero-hydro-servo-elastic simulation in the time domain [16].

Basically, the architecture of OpenFAST consists of a driver program (glue code), which couples several individual submodules together, see Figure 1. These submodules (AeroDyn, SevoDyn, ElastoDyn, BeamDyn, etc.) correspond to different physical domains of the coupled aero-hydro-servo-elastic solution [16].

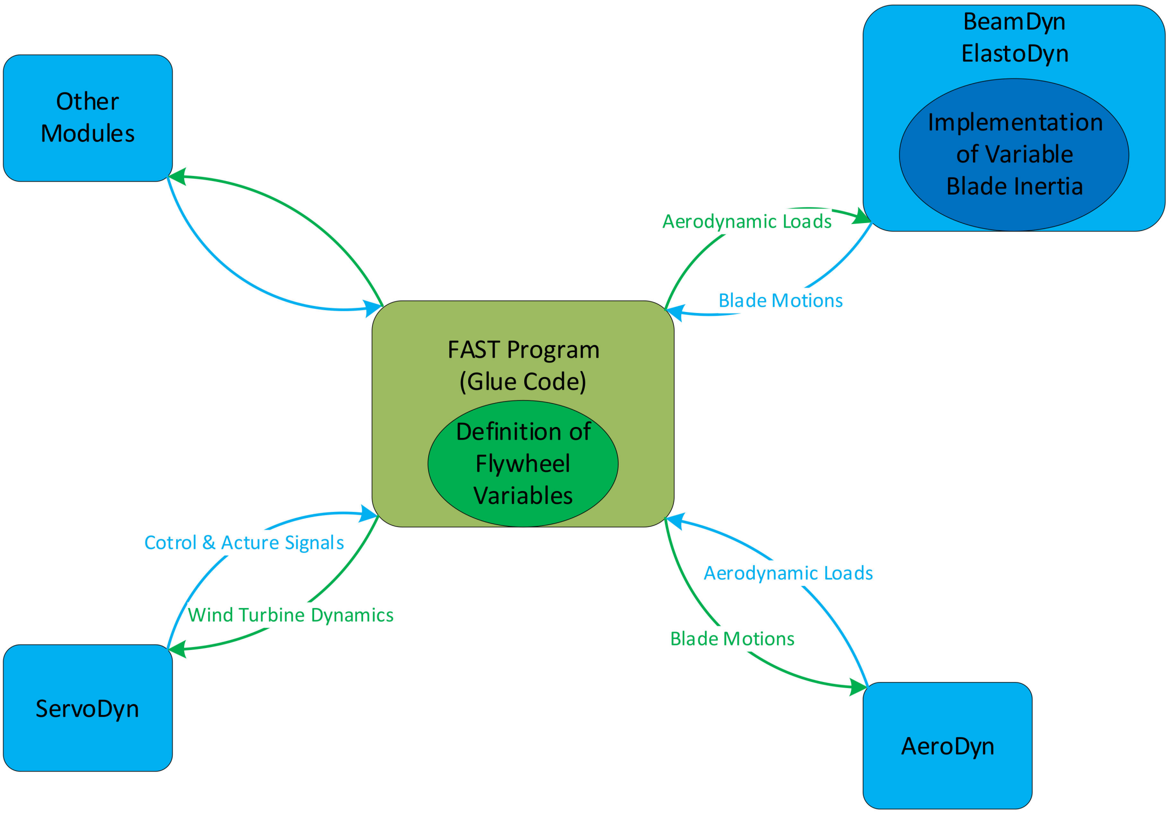


Figure 1. Coupled interaction between FAST and submodules.

ElastoDyn and BeamDyn in Figure 1 are those submodules that are responsible for the blade structural dynamics. The major difference between these two modules lies in the implementation of beam theory. The model underlining BeamDyn is based on the geometrically exact representation of the beam theory (GEBT), whereas ElastoDyn uses Bernoulli-Euler beam theory under bending [16].

ElastoDyn uses Kane's method [17] to set up equations of motion, which can be put into Newton's Second Law (see Equation (1)).

$$F = m{\cdot}a \tag{1}$$

This method simplifies the equation of motion, which makes the equations easier to solve than the method of GEBT.

The GEBT used by BeamDyn makes no approximation on the geometries in formulating the equation of motions (see Equation (2))

$$\begin{gathered}\dot{\underline{h}} - \underline{F}' = \underline{f} \\ \dot{g} + \dot{\tilde{u}}\underline{h} - \underline{M}' + (\tilde{x}_0' + \tilde{u}')^T \underline{F} = \underline{m}\end{gathered} \tag{2}$$

where $\underline{m}$ and $\underline{f}$ are the distributed moment and force applied to the beam structure, respectively; $\underline{g}$ and $\underline{h}$ are the angular and linear moment resolved in the inertial coordinate system, respectively; $\underline{M}$ and $\underline{F}$ are the beam's sectional moment and force, respectively; $\underline{u}$ and x_0 are the one-dimensional displacement and the position vector of a point along the beam's reference line, respectively; the notation $(\bullet)'$ and $(\dot{\bullet})$ indicates a derivative with respect to beam axis x_1 and to time, respectively; the tilde operator $(\tilde{\bullet})$ defines a skew-symmetric tensor corresponding to the given vector [16].

Since the goal of this paper is not to support a full geometric nonlinearity solution or to simulate a large deflection of an anisotropic composite material of a rotor blade with a pre-bended reference axis, it is obvious to choose ElastoDyn to implement the dynamic behavior of the FW system in OpenFAST. Although ElastoDyn uses a simpler mathematical theory than BeamDyn, the exact solutions of ElastoDyn meet the needs in this paper. A detailed comparison between ElastoDyn and BeamDyn is introduced in a previous project report of the author's [15]. Figure 1 shows that the implementation of variable blade inertia takes place within the module of ElastoDyn, whereas the additional variables of the FW system are defined into the glue code of OpenFAST. This enables the FW variables to be passed through the local and external modules of OpenFAST, as discussed later in the next sections.

The executable files, as well as the configuration libraries of the program, are available online on the homepage of OpenFAST [18]. Since variable rotor blade inertias are not considered in the original source code of OpenFAST, the currently available executable cannot be used for the proposed tasks in this paper. Hence compiling from the source code must be done in order to obtain an executable, which includes the add-on of variable blade inertias in its code.

Different methods to obtain the OpenFAST executable from the source code are presented on the homepage of OpenFAST [18]. One of these methods uses Visual Studio (VS) to build OpenFAST on Windows. Building OpenFAST on Windows with VS gives Windows users better options for developing and debugging the source code because of its streamlined manner in solving and maintaining programing errors.

3. Implementation of Variable Blade Elements Masses

Since the main idea of the FW system is based on the variation of rotor blade inertias during the WT operation, this variation is translated in the source code of ElastoDyn by varying the blade element mass within a given simulation. The masses of the blade elements are stored in ElastoDyn blade input files as a part of the blade's mechanical properties. The main challenge facing the implementation of variable blade elements' masses is that the blade's mechanical properties are used as time-invariant input parameters, which are called only once at the module initialization and then they are erased directly after that. Moreover, the data type definition of the blade elements' masses in the programming language of OpenFAST does not allow any change in its value during the whole simulation time. These difficulties and their iterative solutions are discussed in detail in [15]. Nevertheless, the iterative solution is briefly depicted in Figure 2.

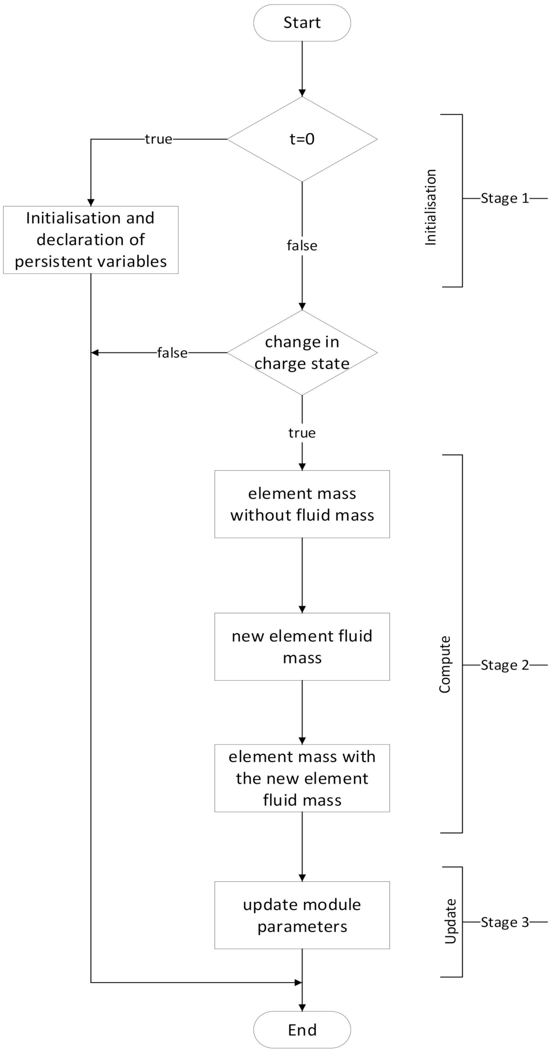


Figure 2. Iterative process for initializing, computing, and updating the change in a blade element mass, and their related parameters.

The flow-chart diagram in Figure 2 divides the iterative solution of the above-mentioned problems in three stages:

- Stage 1: As soon as the simulation starts in OpenFAST at simulation time zero, the variables that declare the initial conditions of the FW system are defined. The detention of these variables is based on the blade input parameters of ElastoDyn, before they are erased after the initialization. This procedure allows access to these variables at any time during the simulation.
- Stage 2: This stage is responsible for the change in the masses of the blade elements regarding the charge-index-factor (*K*). At first, the masses of the blade elements are calculated without the additional fluid mass. Consequently, the new fluid mass of each blade element is calculated depending on the current value of *K*. Finally, the newly calculated fluid mass is added to the corresponded blade element mass. The entrance in this stage is conditional on the change in *K*, i.e., the *K_t-1* from the previous simulation step must not be equal to the *K_t* from the current simulation step in order to enter this stage. Such a condition helps to speed up the simulation when no change in the charge state of the FW in any blade is happening.
- Stage 3: After changing the masses of the blade elements in stage 2 all the parameters that are related to the blade elements' masses must be updated, in order to follow the new values of the changed masses of the blade elements.

These stages are translated in the programming language of OpenFAST as additional subroutines, which are called in the main dynamic calculation of ElastoDyn.

4. Integration of Flywheel System in OpenFAST Simulink Interface

As Simulink is a popular simulation tool for the design and development of control algorithms, OpenFAST has also been implemented as a library that can be called from a Simulink S-Function block, using predefined inputs from Simulink. Therefore, modules such as Generator torque control, nacelle yaw control, pitch control, high-speed shaft brake, and other external modules can be designed in the Simulink environment and, meanwhile, they can use the complete nonlinear aeroelastic wind turbine equations of motion available in OpenFAST [19].

The wind turbine sample model, as shown in Figure 3, contains the S-Function block with the OpenFAST equations of motion. It also contains the control signals that are implanted in the above-mentioned control models. These models are deactivated in the Flywheel_System.mdl model in Figure 4, as the focus is only on the implementation of the FW system. Figure 4 shows the FW system block, which includes the equations of the additional loads and the charge-index-factors for each rotor blade. The next section will discuss in detail which loads are generated by the FW system and not considered in the OpenFAST equations of motion. The function block *f(u)* shown in Figure 4 is used to select certain data from OpenFAST outputs, which are required for the equations in the FW system model.

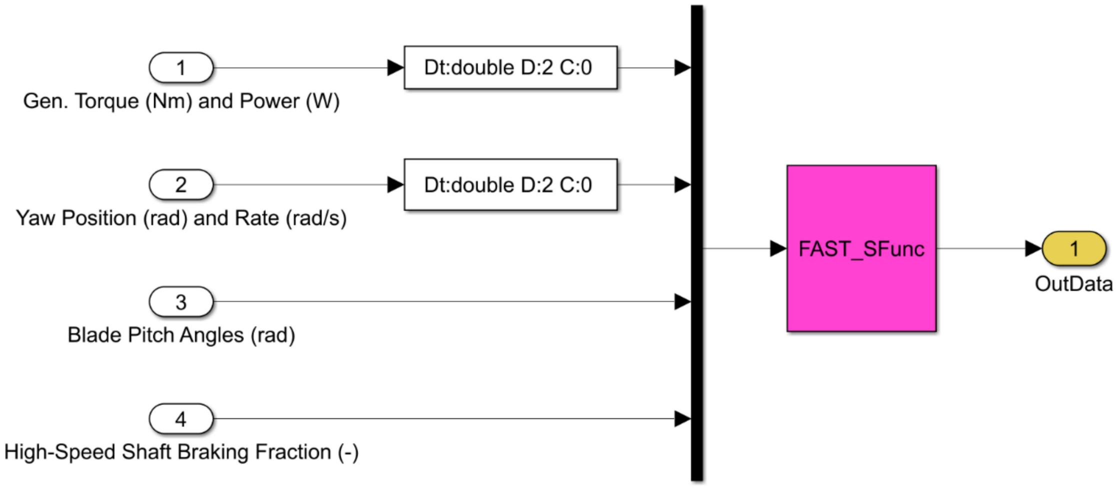

1
Gen. Torque (Nm) and Power (W)
Dt:double D:2 C:0
2
Yaw Position (rad) and Rate (rad/s)
Dt:double D:2 C:0
3
Blade Pitch Angles (rad)
4
High-Speed Shaft Braking Fraction (-)
FAST_SFunc
1
OutData

Figure 3. OpenFAST wind turbine sample model in Simulink.

However, since the OpenFAST Simulink interface is currently set up to receive only the control signals as inputs, as shown in Figure 3, both the OpenFAST glue code and Simulink Interface must be modified. This modification is needed, in order to pass the outputs of the FW system model through the Simulink interface, and then to use them in OpenFAST glue code. Moreover, it must be noted that the largest array of inputs that can be accepted from Simulink interface and passed to OpenFAST glue code must not be more than 11 elements, otherwise, a segmentation fault is transmitted. Both of these issues are discussed in detail on OpenFAST GitHub [20]. Moreover, all source code modifications of OpenFAST that are performed in this paper are available on ResearchGate [21]. However, compiling from the source code is needed in order to obtain an OpenFAST executable with the add-on of variable blade inertia. For compiling from the source code, the NREL OpenFAST team has developed an approach that uses CMake to generate build files for all platforms [18].

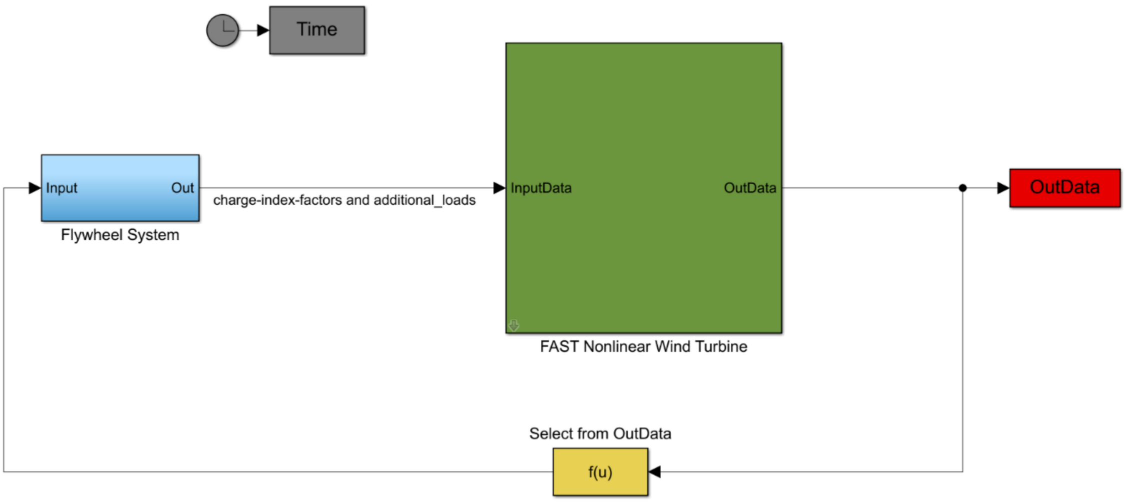


Figure 4. Flywheel_System.mdl model in OpenFAST Simulink interface.

5. Impact of the Variable Blade Elements' Masses on Wind Turbine Loads

In general, the theory basics of ElastoDyn blade structural dynamics does account for the blade loads in two main categories: generalized inertia forces, F_r; and generalized active forces, F_r^*; including the external forces acting on the body, such as blade aerodynamic forces, blade and tower elastic bending forces, gravity, and drive train forces. These are described in Equations (3) and (4) [16].

$$F_r + F_r^* = 0 \tag{3}$$

$$F_r^*|_{Total} = F_r^*|_{Tower} + F_r^*|_{Nacelle} + F_r^*|_{Hub} + F_r^*|_{Blades} \tag{4}$$

Consequently, the resulting velocities and accelerations due to the blade and tower deflections are expressed as partials in the main kinematics and kinetics routines in ElastoDyn. Hence, every force that is based on the acceleration of the blade body is intrinsic to the kinematics and kinetics. This means that the additional loads resulting from the FW system due to the stationary change of the blade elements' masses are intrinsically included in the kinematics and kinetics of ElastoDyn. However, the load that is resulting from the portable mass between two blade elements in a certain time is not considered in the structural dynamics of ElastoDyn. This is because variations in the inertia of the rotor do not happen in state-of-the-art WTs.

This load is already explained in the previous work of Jauch [1] as an additional torque, which is resulting from the change in angular momentum. The law of the conservation of the angular momentum presented in Equations (5)–(7) describes the fact that a torque, T, results if the angular momentum, L, changes, in order to keep the kinetic energy, E_{kin}, constant.

$$E_{kin} = \frac{1}{2} \cdot J \cdot \omega^2 \tag{5}$$

$$L = J \cdot \omega \tag{6}$$

$$T = \frac{\mathrm{d}L}{\mathrm{d}t} \tag{7}$$

where J is the mass moment of inertia, and ω is the rotational speed.

Equation (8) expresses the FW torque, T_{FW_bl}, in one blade in terms of change in the FW inertia in a rotor blade, $\frac{d}{dt}(J_{FW\_bl})$.

$$T_{FW\_bl} = -\frac{d}{dt}(J_{FW\_bl}) \cdot \omega_{rot} \tag{8}$$

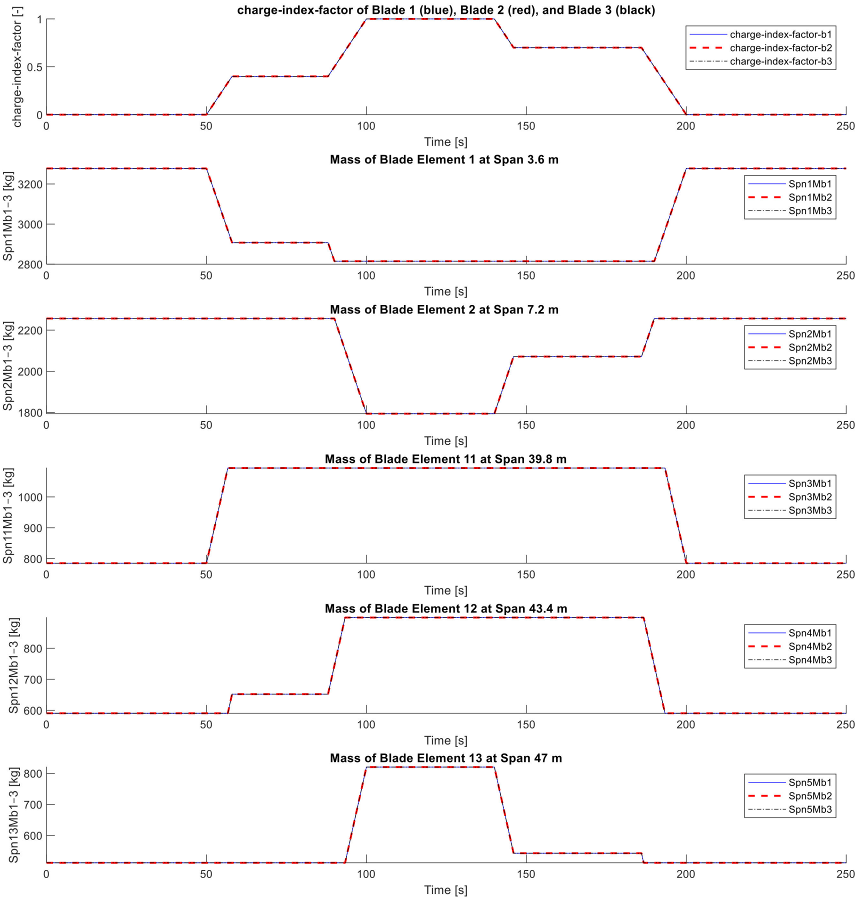
The inertia of the FW in one blade, J_{FW_bl}, changes if the centre of gravity, R_{var}, of the FW fluid, m_{fluid}, changes.

$$J_{FW\_bl} = R_{var}^{2} \cdot m_{fluid} \tag{9}$$

The value of the rotational speed, ω_{rot}, in Equation (8) is derived from the outputs of OpenFAST using the select function *f(u)*, as shown in Figure 4. However, before any outputs are generated in OpenFAST, the FW torque must be implanted correctly in the dynamic calculation of OpenFAST. It must be ensured that the added load does not double count terms that are already added by the change in the masses of the blade elements in the source code of ElastoDyn. This can be accomplished by implementing the FW torque as an external additional torque at the hub since ElastoDyn has a model-level input point mesh for external loads applied at the hub. The FW torque is already calculated in the FW system model in Simulink (see Figure 4) by using Equations (8) and (9), therefore, the value of this additional torque can be used to set the value of the external loads at the hub. Normally, this load is set to zero, in case no external loads have been applied at the hub.

In order to verify the implementation correctness of the FW system in the structural dynamics of the load simulation tool, load simulations are performed for a WT with a deactivated control system and without the impact of the aerodynamic loads. This allows clarification of the impact of variable blade inertia on the behavior of the WT rotor speed independent of any influences from aerodynamic loads and controllers. Therefore, in the following simulation scenario, the aerodynamic, as well as the control modules of OpenFAST are disabled. However, in the next section, these modules are enabled again to cover the impact of the FW system on the behavior of the WT during its power production.

As mentioned in the introduction, blade spans, those that are close to the location of the blade root and tip accumulators of the FW type VI, are selected according to the Sandia 61.5 m rotor blade design. Therefore, blade elements 1 and 2 as well as blade elements 11, 12, and 13 are selected to represent the positions of the root and tip accumulators of the FW type VI, respectively. Figure 5 shows the span locations and the mass of these elements. The definition of the mechanical properties of the blade elements in the source code of ElastoDyn is based on the distributed sectional properties of the Sandia 61.5 m blade design. This design follows the characteristics of the NREL 5 MW reference turbine blade. The NREL 5 MW reference WT has three blades. The structural properties of each blade are distributed in sectional properties along the blade pitch-axis [9]. The distributed blade section mass is based on the geometry and the material selection for each blade section. Regarding the Sandia 61.5 m blade design, the dimensions of the blade root sections are much bigger than the geometry of the blade body and the blade tip sections. Therefore, most of the blade mass is concentrate near to the blade root. This difference in the mass distribution along the rotor blade can also be seen in the difference between the mass of the blade elements in Figure 5, where blade element 1 at a span location of 3.6 m is much heavier than blade element 13 at a span location of 47 m. Figure 5 also shows a fluid mass of 925.46 kg, which is corresponding to the total fluid mass of the FW type VI (see Table 1), which is shifted between the root and tip accumulator. The charge-index-factors determine when and where how much fluid mass must be shifted and which is baled. For the simulation scenario in Figure 5, the FW is charged and discharged simultaneously in the rotor, therefore, the charge-index-factors of the three blades are identical. Unsynchronized charging/discharging is discussed in the next section.


charge-index-factor of Blade 1 (blue), Blade 2 (red), and Blade 3 (black)

Mass of Blade Element 1 at Span 3.6 m

Mass of Blade Element 2 at Span 7.2 m

Mass of Blade Element 11 at Span 39.8 m

Mass of Blade Element 12 at Span 43.4 m

Mass of Blade Element 13 at Span 47 m

Figure 5. Mass change of blade element 1, 2, 11, 12 and 13 depending on the charge-index-factor of each rotor blade.

The FW starts to charge at 50 s and it is fully charged at 100 s, whereas discharging starts at 140 s and ends at 200 s. Meanwhile, the charge/discharge process is interrupted twice: at 58 s as the FW is 40% charged, and at 146 s as the FW is 30% discharged, see subplot 1 in Figure 5. As a response to the changes in the charge-index-factors subplots, 2–6 in Figure 5 show the change in the masses of the above-selected blade elements.

The resulting total FW torque from the three blades, T_{FW_bl}, and its relation to the change in the FW inertia, $\mathrm{d}/\mathrm{d}t(J\_FW\_bl)$ are illustrated in Figure 6. Figure 6 also shows the change in the value of the variable center of gravity, R_var, of the FW type VI in relation to the values of the charge-index-factors in Figure 5.

The impact of variable blade inertia on the rotor speed of the WT due to the application of the FW system is illustrated in Figure 7. Where it is obvious that the increase of the blade inertia by charging the FW system leads to slowing down the rotor speed on the one hand and, on the other hand, the decrease of the blade inertia by discharging the FW system leads to speed up the rotor speed. As mentioned above, since the control system and the aerodynamic loads of the WT are deactivated, the decrease of the rotor speed is not recovered by the aerodynamics or by pitching the rotor blades. This can be obviously seen in Figure 7 as the pitch angle is still unchanged during the whole simulation time, even when the rotor speed changes due to the change in the blade inertia.

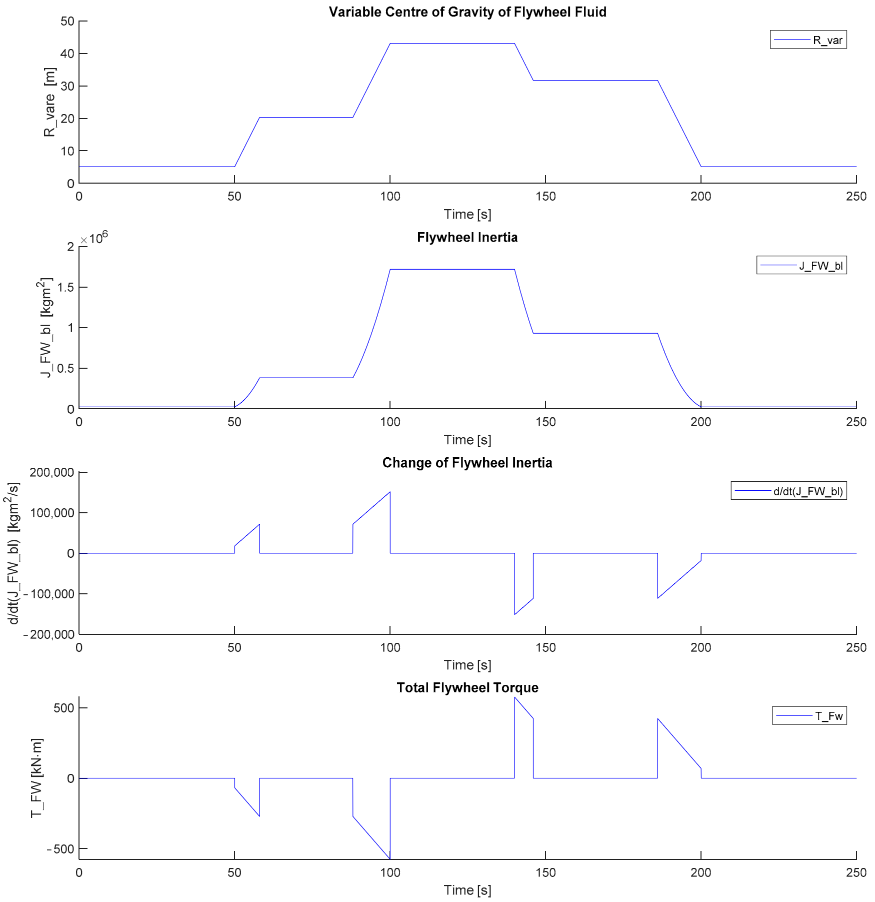


Figure 6. Resulting FW torque of FW type VI.

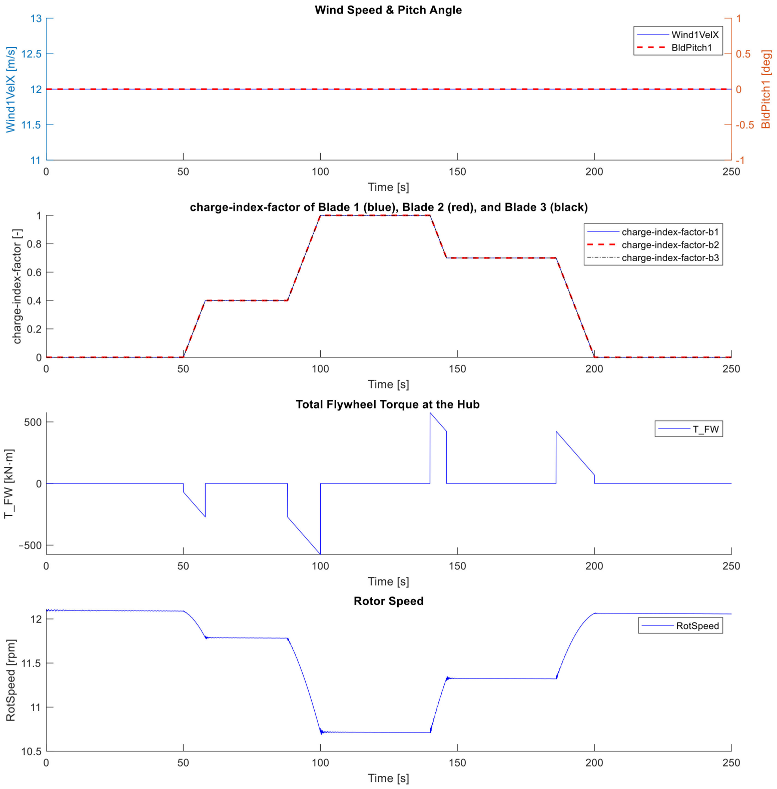


Figure 7. The effect of the FW torque on the rotor speed due to the change in blade elements' masses.

As the response of the rotor speed to the change in the rotor blade inertia shown in Figure 7 meets the physical basics of the law of the conservation of the angular momentum [22], this verifies that the implementation of the FW system in the structural dynamics of OpenFAST is correct. Hence, the modified code can be applied on a WT model during its power production.

6. Analysis of Rotor Imbalance of Wind Turbine with Flywheel System

A rotating system is imbalanced when its center of mass is out of alignment with the center of rotation [23]. In WT, rotor imbalance may be the result of aerodynamic imbalances which could be caused by pitch misalignment of one blade with respect to the other blades, or it may be the result of mass imbalance due to a number of causes, for example, ice accretion, blade damage, poor blade manufacturing tolerances, etc. [24]. Vibrations resulting from rotor imbalances significantly affect WT fatigue, which may reduce the efficiency and the operation life of a WT.

Therefore, there is a need to correct rotor imbalances depending on their causes as soon as possible. Different literature sources contain detailed descriptions on the determination and correction of aerodynamic [25] as well as mass imbalances [24,26]. The rotor imbalance correction methods are generally based on the adjustment of the pitch angle of WT rotor blades, which leads to a reduction in the power output of the WT. Since the FW system has the ability to operate variable blade inertias, this advantage can be applied to correct rotor mass imbalances without the need to adjust the pitch angle. This ability is illustrated in the next simulation scenario.

At first, if the applicability of the FW system to correct mass imbalances to be simulated, the rotor imbalance needs to be represented in a simulated WT model. Therefore, the 5 MW reference WT model presented in the following simulation scenario contains one rotor blade with an additional mass at a certain span location. Since The Sandia 61.5 m blade design is used to implement the FW type VI, this design is also used to represent the rotor imbalance.

Figure 8 shows the rotor imbalance in terms of additional mass of circa 191 kg in blade 1 element 11 at a span location of 39.8 m (see subplot 4 in Figure 8 Spn3Mb1 in blue). The additional mass weight and location are set to those values in order to generate a mass imbalance for this simulation scenario, and they can be varied for other simulation scenarios. This mass difference in blade 1 from the other two blade masses leads to rotor imbalance. In the following simulation scenario, the WT model starts with fully charged FW tip accumulators and with a rotor imbalance simulation time of 150 s. From that time, the FW system is set to discharge not simultaneously in all blades until the three blades have the same inertia. This is done by stopping the FW to discharge in blades 2 and 3 at a simulation time of 163 s, whereas the discharging process is continuing in blade 1 until the mass of blade element 11 in all blades is equal, which is achieved at a simulation time of 167 s. At this point, the total additional mass of blade 1 is shifted from the blade body at a span of 39.8 m to the blade root at a span of 3.6 m. This is how the rotor imbalance is corrected using the FW system, which allows the minimization of the lever arm of the additional mass in one or more blades, and thus, reduces the impact of the mass imbalance on WT behavior. However, since the rotor blades are balanced, the tip FW accumulators in blades 2 and 3 are not allowed to be fully discharged, in order to keep the balance in blade element 11 of all blades.

For the above simulation scenario of rotor imbalance, the WT runs under rated steady wind conditions with a constant horizontal wind speed of 12 m/s. The FW system is initially fully charged in the rotor and is discharged not simultaneously in the three blades, see Figure 9. Such initial rotor imbalance is a 1p excitation, which excites the tower to a later vibration. Figure 9 shows this vibration in terms of side-to-side deflection of the tower top, where the vibration amplitudes of the tower top side-to-side deflections are obviously much greater within the period of the rotor imbalance than the period after the imbalance correction at a simulation time of 167 s. The reason for this is essentially that there is a grater tower-top yaw bearing roll moment before rather than after the correction of rotor imbalance.

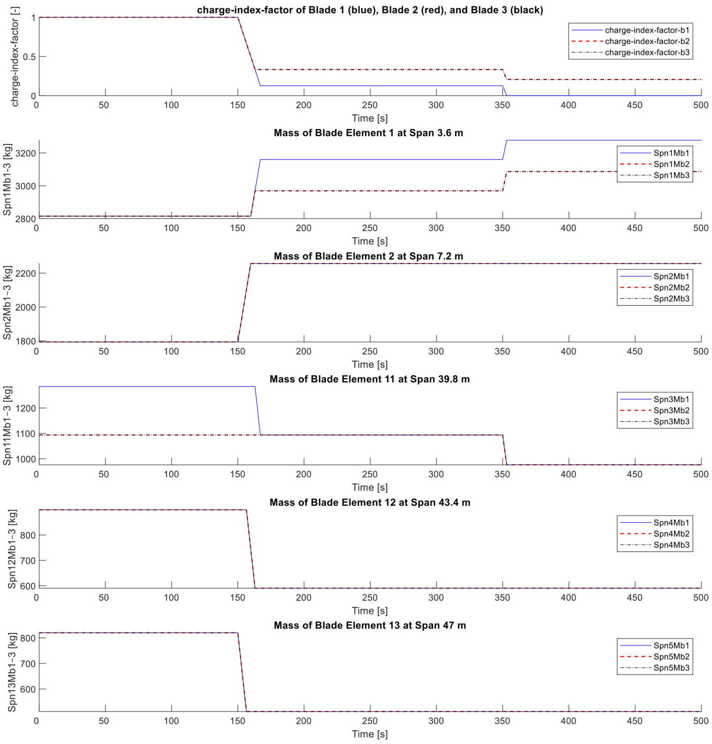


Figure 8. Rotor imbalance due to the additional mass in blade 1, element 11, and elimination of the imbalance by partly charging the flywheel system.

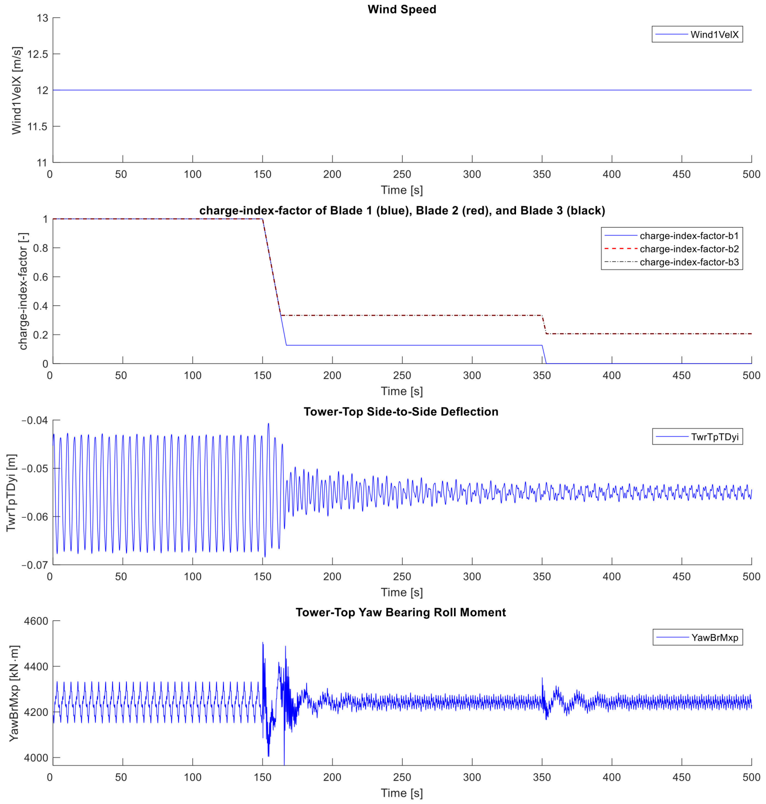


Figure 9. Impact of the flywheel system on a tower-top side-to-side deflection by imbalance in the rotor blade.

In Section 5 it is shown how the WT behaves with the FW system when its aerodynamic loads and control system are disabled. By contrast with Figure 7, Figure 10 shows the impact of the FW system on the WT behavior with enabled aerodynamics and controller modules. Moreover, the presented WT in this section gets excited with a 1p excitation, due to the previously mentioned rotor imbalance. This excitation obviously affects the behavior of rotor speed, pitch angle, and generator power in the first 150 s, where continuous oscillations in the signals are seen in this period. After that, the rotor imbalance correction procedure starts by discharging the FW system asymmetrically in the three blades. The FW torque, T_{FW}, generated by discharging the FW speeds up the rotor over the rated value of rotor speed, and thus, the power production increases over the rated value of 5 MW. Consequently, the pitch controller raises the pitch angle in order to slow down the rotor to the rated rotor speed again. The original pitch controller of the NREL 5 MW reference WT is designed to maintain the balance between power generation and rotor speed at rated conditions. Therefore, such changes in the rotor speed due to the change in blade inertia are immediately maintained by the pitch controller, and thus, the released kinetic energy by discharging the FW system is wasted through the pitch control.

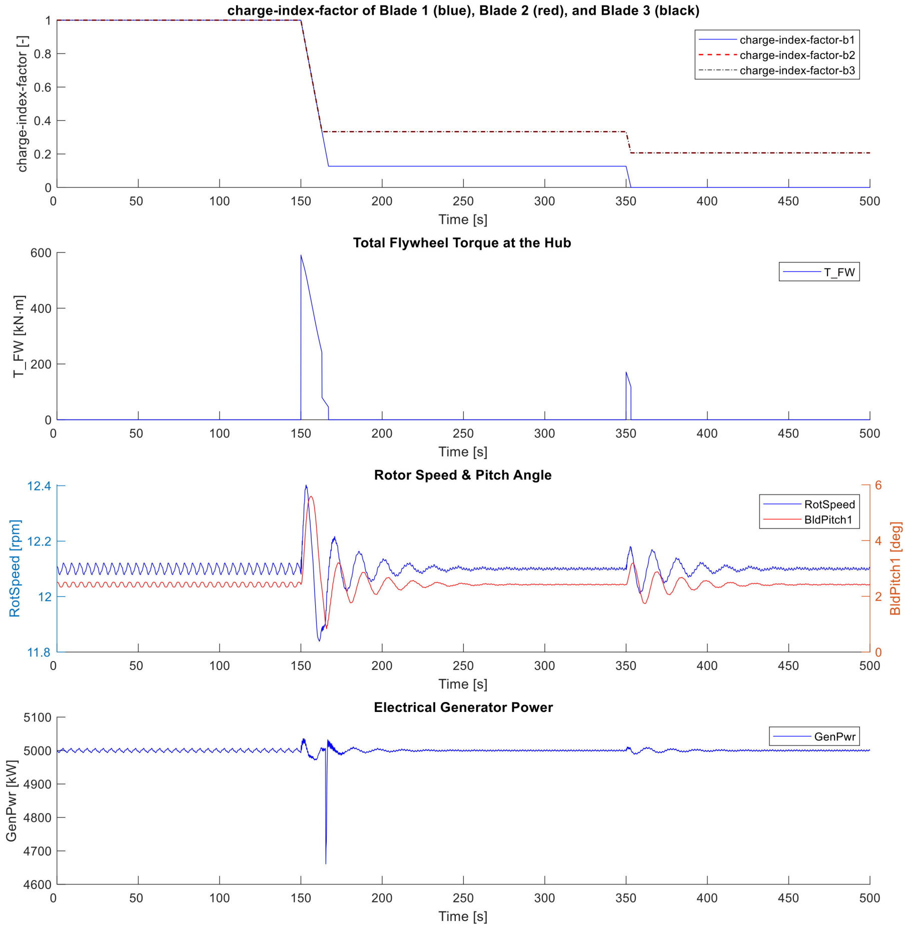


Figure 10. Impact of the flywheel torque on rotor speed and generator power.

Hence, in future, a controller has to be designed that uses the stored kinetic energy, for example, to support grid frequency. The developed controller should also consider the power consumption of the FW system since its tip accumulators are charged with electric pumps, which are supplied from the power production of the WT.

7. Conclusions

In the present paper, variable blade inertia was implemented in the source code of the load simulation tool OpenFAST, in order to integrate an FW system in WT rotor blades. The

FW configurations used to represent the change in blade inertia are those corresponding to the FW type VI. The stationary masses of the root and tip accumulators are added at the respective location of the FW type VI within the blade mechanical properties of the Sandia 61.5 m rotor blade design, whereas the fluid mass of the FW type VI is implanted in the source code of the blade's structural dynamic module of OpenFAST to be shifted in- and outboard.

ElastoDyn and BeamDyn are those two modules that are represented by OpenFAST to implement the structural dynamics of rotor blades. Since the goals of this paper are not to support a full geometric nonlinearity solution, or to simulate a large deflection of a rotor blade, which is supported by BeamDyn, ElastoDyn is used for implanting the variable blade inertia. Moreover, the exact solutions of ElastoDyn fulfill the requirements of the simulated WT scenarios.

However, such implementation needs a good understanding of the source code of the load simulation program. Therefore, an algorithm is prepared to illustrate the major modifications of the source code. The code that the algorithm is based on is described in detail in a previous research project of the author [15], and it is available on ResearchGate [21].

Since OpenFAST is also implemented as a dynamic library that can be called from MATLAB/Simulink, this feature is used to implement the mathematical operations of the FW system as a Simulink model. However, passing outputs from the FW model through the Simulink interface of OpenFAST needs to modify both OpenFAST glue code and Simulink Interface.

The outputs of the FW Simulink models includes inter alia, the FW torque, T_{FW}, This torque represents the additional rotor torque that results from the change in angular momentum. Although this additional load is not considered in the structural dynamics of ElastoDyn, ElastoDyn offers a model-level input point mesh for external loads to be applied at the hub. This model is used to apply the FW toque to the hub.

As the source code, modifications are done. It is, therefore, necessary to verify the implementation correctness of the FW system in the structural dynamics of OpenFAST. This verification is done by simulating a WT with the integrated FW system and without the interaction of the aerodynamic loads and controllers. This simulation scenario aims to illustrate that the implemented variable blade inertia can lead to speeding up the rotor when the FW discharges, and to slowing down the rotor when the FW charges, as provided by the law of the conservation of the angular momentum.

Finally, as one of the several FW system advantages, the change of the blade inertia is applied to correct a rotor mass imbalance. The simulated mass imbalance is produced by adding an additional mass to one blade at a certain span location. The impact of the 1p excitation of rotor imbalance on the vibratory behavior of the WT is shown before and after the imbalance correction by using the FW system.

The impact of the FW system on the WT during its power production is also illustrated, where it was clear that future works should focus on developing control designs which manage the energies that are released and consumed by the FW system. Moreover, it is advantageous to validate the implementation of variable blade inertia in BeamDyn or in different load simulation tools, which will be the focus of future works.

Funding: This work was supported by the Gesellschaft für Energie und Klimaschutz Schleswig-Holstein GmbH (EKSH); Project Nr. 14/12-23.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All source code modifications of OpenFAST (FAST_ Library.f90, FAST_Registry.txt, ElastoDyn.f90, ElastoDyn Registry.f90, FAST_Solver.f90) that are performed in this paper are available on ResearchGate "https://www.researchgate.net/profile/Laurence-Alhrshy-2".

Acknowledgments: The author would like to thank Bonnie Jonkman and Jason Jonkman for their technical supports on OpenFAST GitHub. The author would also like to thank Clemens Jauch and Henning Thiesen for their review.

Conflicts of Interest: The author declare no conflict of interest.

References

1. Jauch, C. A Flywheel in a Wind Turbine Rotor for Inertia Control. *Wind Energy* **2015**, *18*, 1645–1656. [CrossRef]
2. Jauch, C.; Hippel, S. Hydraulic–Pneumatic Flywheel System in a Wind Turbine Rotor for Inertia Control. *IET Renew. Power Gener.* **2016**, *10*, 33–41. [CrossRef]
3. Jauch, C. Controls of a Flywheel in a Wind Turbine Rotor. *Wind Eng.* **2016**, *40*, 173–185. [CrossRef]
4. Alhrshy, L.; Jauch, C.; Kloft, P. Development of a Flexible Lightweight Hydraulic-Pneumatic Flywheel System for Wind Turbine Rotors. *Fluids* **2020**, *5*, 162. [CrossRef]
5. Hippel, S.; Jauch, C. Load Analysis of Hydraulic-Pneumatic Flywheel Configurations Integrated in a Wind Turbine Rotor. *Wind Energy* **2019**, *22*, 1190–1202. [CrossRef]
6. Hippel, S.; Jauch, C.; Ritschel, U. Hydraulic-Pneumatic Flywheel Configurations for Controlling the Inertia of a Wind Turbine Rotor. *Wind Eng.* **2019**, *43*, 114–132. [CrossRef]
7. Jauch, C. First Eigenmodes Simulation Model of a Wind Turbine—For Control Algorithm Design. 2020. [CrossRef]
8. Resor, B. *Definition of a 5MW/61.5m Wind Turbine Blade Reference Model*; Technical Report No. SAND2013–2569; Sandia National Lab.: Albuquerque, NM, USA, 2013; p. 947422.
9. Jonkman, J.; Butterfield, S.; Musial, W.; Scott, G. *Definition of a 5-MW Reference Wind Turbine for Offshore System Development*; Technical Report No. NREL/TP-500-38060; National Renewable Energy Laboratory: Golden, CO, USA, 2009; p. 947422.
10. Jørgensen, M.F. *Aerodynamic and Mechanical System Modelling*; Department of Mechanical Engineering, Technical University of Denmark: Lyngby, Denmark, 2013; ISBN 978-87-7475-370-4.
11. FAST. Available online: https://www.nrel.gov/wind/nwtc/fast.html (accessed on 13 January 2021).
12. Wind Turbine Design Software | Bladed. Available online: https://www.dnvgl.com/services/wind-turbine-design-software-bladed-3775 (accessed on 13 January 2021).
13. FAST v8 and the Transition to OpenFAST—OpenFAST v2.3.0 Documentation. Available online: https://openfast.readthedocs.io/en/master/source/user/fast_to_openfast.html (accessed on 13 January 2021).
14. OpenFAST/Openfast. Available online: https://github.com/OpenFAST/openfast (accessed on 13 January 2021).
15. Alhrshy, L.; Jauch, C.; Bünning, N.; Schaffarczyk, A. Development of a Lightweight Hydraulic-Pneumatic Flywheel System for Wind Turbine Rotors. 2021. [CrossRef]
16. National Renewable Energy Laboratory. OpenFAST Documentation. Available online: https://openfast.readthedocs.io/_/downloads/en/main/pdf/ (accessed on 12 May 2021).
17. Kane, T.R.; Levinson, D.A. Dynamics: Theory and Applications. Available online: File:///C:/Users/Alhrshy/AppData/Local/Temp/Dynamics-Theory_opt-1.pdf (accessed on 25 March 2021).
18. Installing OpenFAST—OpenFAST v2.3.0 Documentation. Available online: https://openfast.readthedocs.io/en/master/source/install/index.html#compile-from-source (accessed on 14 January 2021).
19. Jonkman, J.M.; Buhl, M.L., Jr. *FAST User's Guide—Updated August 2005*; Technical Report No. NREL/TP-500-38230; National Renewable Energy Laboratory: Golden, CO, USA, 2005; p. 15020796.
20. Issue When I Run Simulink Model Which Depends on Openfast S-Function Mex Issue #548 OpenFAST/Openfast. Available online: https://github.com/OpenFAST/openfast/issues/548 (accessed on 12 October 2020).
21. Alhrshy, L. OpenFAST Modified Codes. 2021. [CrossRef]
22. Conservation of Angular Momentum Review (Article). Available online: https://www.khanacademy.org/science/high-school-physics/torque-and-angular-momentum/conservation-of-angular-momentum/a/conservation-of-angular-momentum-ap-physics-1 (accessed on 23 February 2021).
23. Kelm, R.; Kelm, W. Rotor Balancing Tutorial. Available online: https://rotorlab.tamu.edu/me459/Rotor%20Balancing/2016%20TPS%20Tut%20ROTOR%20BALANCING%20Pavelek.pdf (accessed on 12 May 2021).
24. Cacciola, S.; Agud, I.M.; Bottasso, C.L. Detection of Rotor Imbalance, Including Root Cause, Severity and Location. *J. Phys. Conf. Ser.* **2016**, *753*, 072003. [CrossRef]
25. Bertelè, M.; Bottasso, C.L.; Cacciola, S. Automatic Detection and Correction of Pitch Misalignment in Wind Turbine Rotors. *Wind Energy Sci.* **2018**, *3*, 791–803. [CrossRef]
26. Jeffrey, M.; Melsheimer, M.; Liersch, J. Method and System for Determining an Imbalance of a Wind Turbine Rotor. Available online: https://patents.google.com/patent/US8261599B2/en (accessed on 30 January 2021).