Floating wind turbine control optimization

Daniel Zalkind[1], Nikhar J. Abbas[1,2], John Jasa[1], Alan Wright[1] and Paul Fleming[1]

[1] National Renewable Energy Laboratory, Golden, CO 80401, USA
[2] University of Colorado Boulder, Boulder, CO 80309, USA

E-mail: daniel.zalkind@nrel.gov

Abstract. We present a framework for optimizing the control parameters of floating offshore wind turbines (FOWTs). The framework combines aeroelastic simulations with a systems engineering model and control software. In an example of the optimization framework, we minimize tower damage equivalent loading with generator speed constraints. We also study the effect of thrust-limiting control and quantify the trade off between fatigue loading and energy capture using a set of optimal controller designs. Finally, we optimize the controller of four different FOWT models and compare their dynamic responses. Additional details and other use cases for the framework are presented, which can optimize different control problems and evaluate FOWT designs.

1. Introduction

Using a multifidelity wind turbine optimization framework, we optimize the parameters of the Reference Open-Source Controller (ROSCO [1]) for four different floating offshore wind turbine (FOWT) platforms and the International Energy Agency (IEA) 15-MW reference turbine [2] in different control configurations. The framework uses aeroelastic simulations in OpenFAST [3] coupled with ROSCO [4] to compute merit figures and constraints while iterating toward an optimal solution.

We demonstrate this optimization framework on several control configurations and FOWT models to showcase the repeatability and versatility of a process that can compare FOWT models. By optimizing several control parameters at once, we can account for their design coupling. For example, changing the thrust limit or floating-specific control parameters affects the closed-loop generator speed control performance. By optimizing the control parameters of different FOWT models, we can account for differences in FOWT platform dynamics and ensure that we use the best controller possible, given the optimization problem, for a fair comparison between models.

Because of the computational effort required to evaluate controller performance, previous work in controller tuning was limited to one or a few parameter sweep studies [5, 6]. Recent optimization studies focused on one or a few control design variables and found that non-smoothness in the constraints and merit figures makes traditional, derivative-based optimization procedures difficult to use [7]. To understand the full performance of a controller, hundreds of International

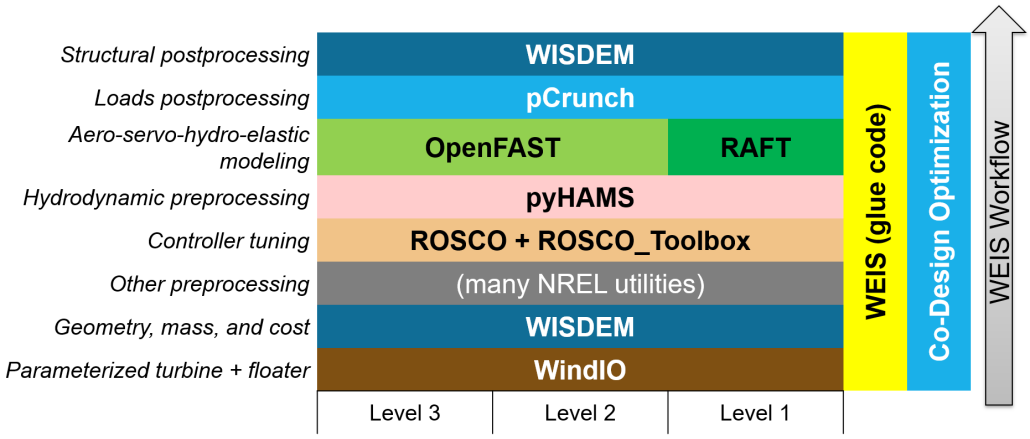

Figure 1. Visualization of the WEIS framework highlighting the three levels of fidelity and multiple tools integrating within the framework.

Electrotechnical Commission (IEC) design load case (DLC) simulations [8] should be run, and sometimes the design-driving case can change depending on the design variable. The best way to ameliorate this issue is to simulate as many DLCs as possible.

Early FOWT optimization with aeroelastic models used simplified models. For example, a linear FOWT model and linear controller was used to optimize a spar platform [9], which provided gradients and an efficient optimization. Another approach used a low fidelity nonlinear hydrodynamic model and controller [10]. A more detailed and higher fidelity optimization framework uses a fully nonlinear model and the Dymola framework to optimize a spar platform [11].

The Wind Energy with Integrated Servo-control (WEIS) software, which we describe in Section 2, uses open-source nonlinear simulation and control models, exactly like those used to compute loads and power predictions. These models enable more design flexibility in both the floating platform and controller, which we describe in Section 3. We demonstrate the optimization capability on the control parameters of the IEA 15-MW reference turbine (Section 4) and on a set of different floating platform models (Section 5).

2. WEIS overview

The WEIS framework [12] is a new design tool being developed at the National Renewable Energy Laboratory (NREL) through the Advanced Research Projects Agency-Energy Aerodynamic Turbines Lighter and Afloat with Nautical Technologies and Integrated Servo-control (ARPA-E ATLANTIS) program. WEIS combines and extends the capabilities of multiple existing NREL tools, including the systems engineering framework Wind Plant Integrated System Design and Engineering Model (WISDEM®) [13], the aero-servo-elastic solver OpenFAST [3], ROSCO [4], the wind solver TurbSim [14], and several pre- and postprocessing routines, including pCrunch. WEIS is designed to provide a common framework for turbine and controller codesign of FOWTs at multiple levels of fidelity. It uses the systems engineering ontology for wind turbine representation developed by IEA Task 37 [15], so the resulting designs can be easily imported to other NREL tools and to external tools.

In this work, all studies are implemented and solved using WEIS. Specifically, we use the software stack corresponding to "Level 3" in figure 1, which entails running fully nonlinear OpenFAST simulations. In WEIS, the turbine class, turbulence level, and sea state specifications can be

configured to simulate IEC design load cases 1.2, 1.3, 1.4, 1.5, 1.6, 6.1, and 6.3 [8]. The outputs of these simulations can be used as merit figures (table 1) and constraints (table 2) in optimization problems. Using parallel computing or high-performance clusters, WEIS can simultaneously simulate as many DLCs as are needed, so one optimization iteration can be computed in the time it takes to run one OpenFAST simulation, which is about 3–5 minutes, on average, in the models studied in this article. WEIS produces multiple forms of output data, including the full optimization histories, allowing users to postprocess their design cases and examine what impacts the design variables have on performance metrics.

Table 1. Merit figures available in WEIS for controller optimizations.

`DEL_RootMyb`	Blade root damage equivalent loading in the flapwise direction, computed using DLC 1.2 simulations
`DEL_TwrBsMyt`	Tower base damage equivalent loading
`rotor_overspeed`	The maximum generator speed relative to the rated generator speed
`Std_PtfmPitch`	The average standard deviation of the platform pitch angle over all DLCs
`Max_PtfmPitch`	The maximum platform pitch angle over all DLCs
`LCOE`	Levelized cost of energy, computed using WISDEM
`AEP`	Annual energy production, computed using DLC 1.2 simulations

Table 2. Constraints available in WEIS for controller optimizations.

`rotor_overspeed`	The maximum generator speed relative to the rated generator speed
`Max_PtfmPitch`	The maximum platform pitch angle over all DLCs
`Std_PtfmPitch`	The maximum standard deviation of the platform pitch angle over all DLCs
`nacelle_acceleration`	The maximum absolute nacelle acceleration over all DLCs
`damage`	The damage accumulated at the tower base in DLC 1.2 simulation of the turbine's lifetime, where 1 indicates a failure
`avg_pitch_travel`	The average blade pitch travel over DLC 1.2
`pitch_duty_cycle`	The number of direction changes in the blade pitch per minute in DLC 1.2
`Max_Offset`	The maximum surge and sway distance the platform moves compared to the equilibrium point

2.1. Discussion of optimization solvers

For all the results shown in this article, we use the `LN_COBYLA` solver from NLOpt [16], but several other methods are also available through WEIS. During this work, we considered different optimization methods, including local, global, derivative-based, and derivative-free methods, which are described in table 3.

In general, gradient-based methods are able to find local optima faster than gradient-free methods [23]. However, WEIS does not provide efficient derivatives for the functions of interest

Table 3. Optimization solver overview.

Method	Type	Description
`LN_COBYLA` [17]	Local, Gradient-Free	Constrained Optimization by Linear Approximation: constructs linear approximations of objective functions and constraints and optimizes these approximations in a trust region
`LD_MMA` [18]	Local, Gradient-Based	Method of Moving Asymptotes: Forms a local approximation of the gradient and constraint functions
`LD_SLSQP` [19]	Local, Gradient-Based	Sequential Least-Squares Quadratic Programming: optimizes successive second-order approximations of the objective function and linear constraints
`GN_ISRES` [20]	Global, Gradient-Free	Improved Stochastic Ranking Evolution Strategy
`GN_DIRECT` [21]	Global, Gradient-Free	DIviding RECTangles: deterministic search algorithm into smaller hyperrectangles
`NSGA2` [22]	Global, Gradient-Free	Simplistic genetic algorithm

with respect to design variables but instead relies on finite differencing to obtain derivative approximations. Derivatives obtained with the finite differencing of simulation results, which can be non-smooth, present difficulty in selecting appropriate step size parameters for these methods. Due to the relatively high computational cost of OpenFAST, we are not able to effectively use global optimization methods, as they would require a large number of iterations to converge.

In this work, we performed controls optimization of a floating turbine using each of the methods listed in table 3. Most methods did not converge well within 100 iterations. The gradient-based methods show promise in some applications, but require more simulations and fine-tuning of parameters for accurate finite difference approximations and sometimes struggle to satisfy the design constraints. `GN_ISRES`, `NSGA2`, and `GN_DIRECT` could not satisfy the constraints at their termination points.

`LN_COBYLA` results in the most reliable performance without significant fine-tuning of the optimization parameters. Sometimes `LN_COBYLA` does not fully satisfy constraints; however, in these cases, using conservative values for constraints yields better results. Because `LN_COBYLA` results in the most reliable performance and has reasonable convergence, all optimization results presented in this paper were achieved using `LN_COBYLA`.

3. ROSCO controller description and design variables

In this article, we optimize the reference open-source controller (ROSCO [4, 1]), which has been designed to reproduce most of the functionality of a modern industry controller and is flexible enough to be tuned for a wide range of turbine and platform models. Several design variables (summarized in table 4) can change the controller behavior.

ROSCO uses a proportional-integral pitch controller, which is parameterized by the natural frequency ω and damping ratio ζ of the closed-loop generator speed regulation mode [4]. We have updated ROSCO so that $\omega(u)$ and $\zeta(u)$ can be functions of the wind speed u, which adds

Table 4. ROSCO controller design variables available in WEIS.

`omega_pc`	$\omega(u)$	Natural frequency of the pitch controller, which can be scheduled on the wind speed, u
`zeta_pc`	$\zeta(u)$	Damping ratio of the pitch controller, which can be scheduled on the wind speed, u
`Kp_float`	k_{float}	Floating feedback gain from tower-top motion to blade pitch angle
`ptfm_freq`	ω_{ptfm}	Floating feedback filter cutoff frequency
`omega_vs`	ω_{tq}	Natural frequency of the torque (variable speed) controller
`zeta_vs`	ζ_{tq}	Damping ratio of the torque (variable speed) controller
`pc_percent`	-	Peak shaving percent: the allowed maximum thrust, compared to the uncontrolled maximum thrust

flexibility to the tuning. We parameterize these functions using breakpoints (as shown in Fig. 3) and a sigma interpolation method [24]. The sigma interpolation provides a smooth step change between parameters. The proportional-integral gains are a function of $\omega(u)$ and $\zeta(u)$ and the rotor's sensitivity of power to pitch angle, which increases with wind speed, thus the gains decrease with wind speed [4].

A minimum pitch limit can optimize power at low wind speeds and reduce thrust near rated wind speeds; we call this the peak shaving percent, or `ps_percent`. A floating feedback control loop filters the tower-top velocity using a low-pass filter with a bandwidth of ω_{ptfm}, applies a gain (k_{float}), and adds the resulting value to the pitch control, which adds damping to the platform pitch dynamics.

Optimizing the near- and above-rated blade pitch control of ROSCO is the primary focus of this article. Increasing ω improves generator speed regulation, but also increases platform pitch motion and tower loads [25]. By optimizing the scheduling of this design variable, we can balance the performance across wind speeds. Floating feedback control also affects both generator speed regulation and platform pitch motion; by including its optimization in the same problem, we can account for design coupling between these variables and find an optimal solution.

In below-rated operation, ROSCO uses a wind speed estimate to determine the optimal generator speed set point for tracking the optimal tip speed ratio. While not the focus of this article, an optimization problem using WEIS could balance power capture with power variation when optimizing the torque control parameters. ROSCO is equipped with a constant power control mode, where the maximum allowable torque is varied to maintain close-to-rated power output in above-rated wind speeds. Set-point smoothing logic handles the transitions between above- and below-rated pitch and torque controllers and ensures that only one control mode is primarily active. Due to the flexible nature of ROSCO, any of its necessary or optional tuning parameters can be defined as design variables using WEIS.

4. IEA 15-MW controller optimization

Using WEIS, we update a number of ROSCO control parameters for the IEA 15-MW reference turbine [2]; this direct-drive, three-bladed, upwind rotor has a hub height of 150 m, a rotor diameter of 240 m, and a rated rotor speed of 7.56 rpm. We minimize the tower base bending damage equivalent load (DEL), or $m_{ty,\text{DEL}}$, and we constrain the maximum generator speed $\omega_g^{\max}$ such that it does not exceed 125% of the rated generator speed ω_g^{rated}. In other words, we

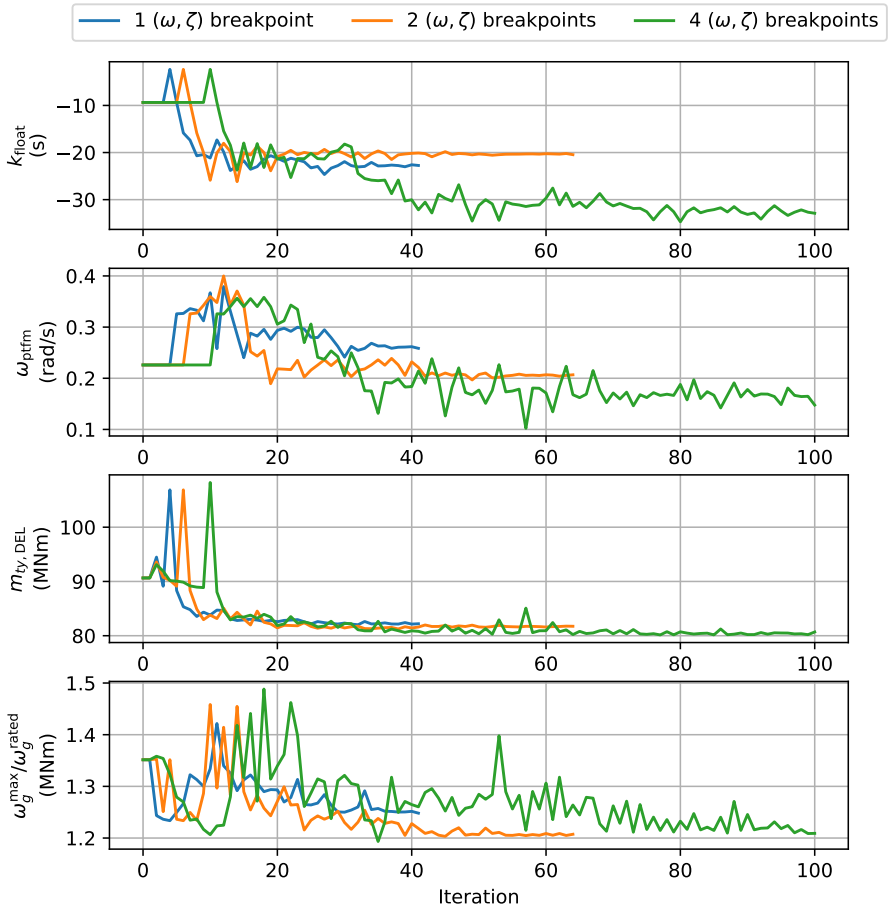

Figure 2. The convergence of design variables, constraints, and merit figures for the IEA 15-MW ROSCO optimization with 1, 2, and 4 equally spaced breakpoints to define $\omega(u)$ and $\zeta(u)$.

solve

$$\min_{\omega(u),\zeta(u),k_{\text{float}},\omega_{\text{ptfm}}} \quad m_{ty,\text{DEL}} \qquad \text{s.t.} \quad \frac{\omega_g^{\max}}{\omega_g^{\text{rated}}} < 1.25 \tag{1}$$

using the `LN_COBYLA` solver described in table 3 and the design variables described in table 4. Both $m_{ty,\text{DEL}}$ and $\omega_g^{\max}$ are determined using the full set of DLC 1.2 and 1.3 simulations [8], which represent regular turbine operation in normal and extreme turbulence, respectively. DLC 1.2 is used to determine the damage equivalent loading and annual energy production (AEP); near-rated simulations in DLC 1.3 usually cause rotor overspeed events.

Because the blade pitch control has the greatest effect on floating platform motion, tower loads, and generator speed regulation, we optimize $\omega(u)$, $\zeta(u)$, k_{float}, and ω_{ptfm}. The pitch control regulator mode, characterized by $(\omega(u), \zeta(u))$, can be scheduled based on the wind speed u with one or more breakpoints. In figure 2, we use a different number of breakpoints to parameterize the functions that define $(\omega(u), \zeta(u))$.

Compared to the baseline ROSCO parameters, the optimized parameters reduce $m_{ty,\text{DEL}}$ by

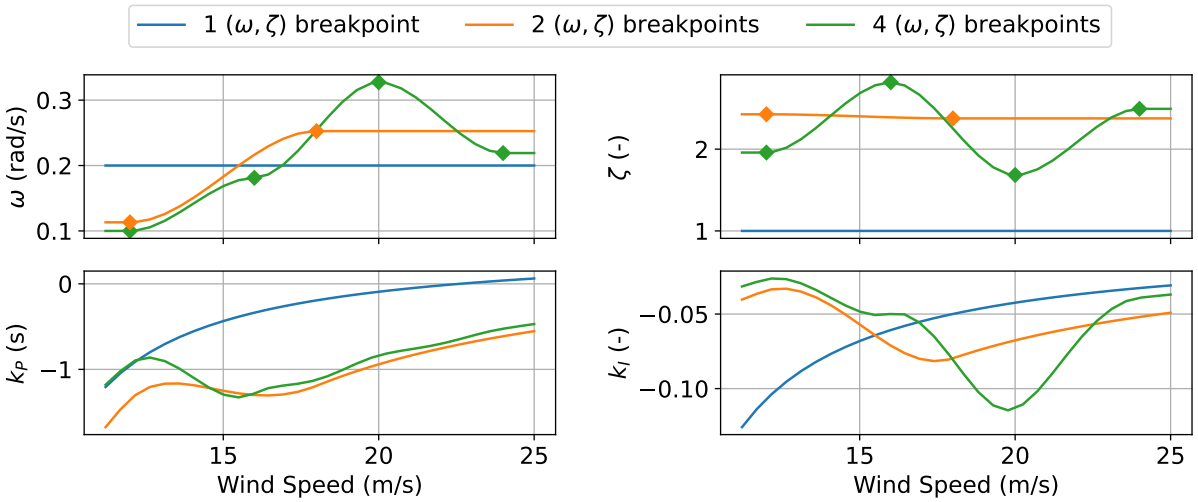

Figure 3. Proportional (k_P) and integral (k_I) pitch control gains, which are determined based on the natural frequency $\omega(u)$ and damping ratio $\zeta(u)$, where u is the wind speed. In ROSCO, k_P and k_I are scheduled using a low-pass-filtered blade pitch angle.

9.59%, 9.82%, and 11.0%, depending on whether 1, 2, or 4 breakpoints are used to define the functions $\omega(u)$ and $\zeta(u)$. Thus, most of the benefit can be realized by optimizing a constant ω and ζ, which converges in 40 iterations, compared to 60 and 100 iterations when 2 and 4 breakpoints are used, respectively.

The proportional-integral gain schedule (figure 3) shows that a lower bandwidth at near-rated wind speeds improves performance, but the proportional gain k_P is nearly constant when a greater amount of flexibility is allowed in the $(\omega(u), \zeta(u))$ functions. At higher wind speeds, the gains resemble those chosen for the fixed-bottom version of the IEA 15-MW turbine.

4.1. Effect of peak shaving

Peak shaving limits the maximum thrust on the rotor, which affects turbine loads and floating platform and rotor motion, which affects generator speed dynamics. Thus, the peak shaving parameter is coupled with the other ROSCO parameters, and to study the true effect of peak shaving on power and loads, we solve the optimization problem in equation (1) at six different thrust limits.

Each problem was solved to determine the optimal set of parameters for a range of thrust limits from 75% to 100%, compared to the maximum rated thrust on the IEA 15-MW turbine. As expected, increasing the thrust limit (or `ps_percent`) increased both the tower DELs and the AEP, as shown in figure 4. There is generally a linear trade-off between tower DELs and AEP, where a 1% increase in AEP yields about a 5% increase in tower DELs; most of the benefit occurs between 0.75 and 0.85, with diminishing returns as `ps_percent` approaches 1.00.

5. Comparing FOWT platforms with optimized controllers

To compare four different floating platform designs, we solved the optimization problem in equation (1) for each model. Because each platform has different dynamics, the relative wind speed on the rotor, which affects generator speed regulation, should result in different optimal control gains. If we consistently tune each of the controllers using the optimization problem in equation (1), we can ensure a more reliable comparison between designs.

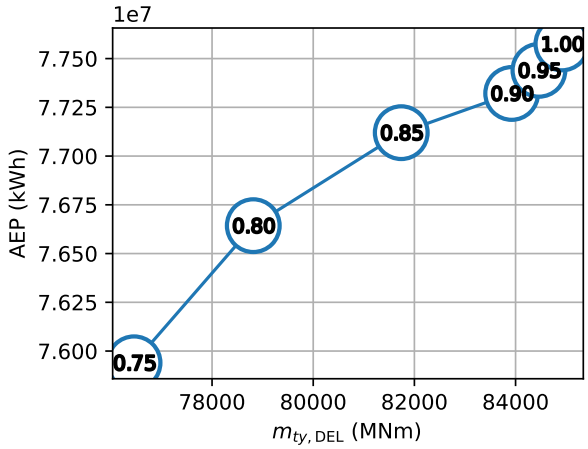

Figure 4. The annual energy production (AEP) versus the tower base damage equivalent loading ($m_{ty,\mathrm{DEL}}$) for six different thrust limits from 0.75 to 1.00, as indicated in the figure.

Table 5. Summary of floating platform models.

	Spar	Semi	Barge	TLP
Draft (m)	100	20	8	35
COG, COB[a] (m)	-53.2, -67.7	-0.36, -13.03	-4, 13	-26.0, 30.6
Length, breadth, height (m)	19.2, 19.2, 115	83.3, 93.8, 35	58,58,23	70.6, 81.4, 50.0
Substructure mass (t)	24,925	16517	12419	4662
Surge, pitch eigenperiods (s)	137.1, 31.3	129.8, 25.5	58.0, 22.1	31.2, 3.6

[a]COG: center of gravity, COB: center of buoyancy

A spar, semisubmersible, barge, and tension-leg platform (TLP) model (summarized in table 5) was developed to provide reference models for projects aimed at investigating standards, yield estimates, and failure rates, among other things. Each floating substructure is designed assuming steel as the material and uses the IEA 15-MW reference turbine rotor [2]. A stiff-stiff tower, with a first bending mode greater than the rotor's thrice-per-revolution (3P) frequency, was designed for each of the floating platform models, except the TLP, which uses a soft-stiff tower (with a mode between 1P and 3P).

The optimal control parameters, merit figure, and constraint for the various platforms are shown in table 6. In general, low values of $\omega(u)$ work well near rated wind speeds but can be increased as wind speeds increase. There is enough variation in the parameters to warrant the optimization effort. More platform feedback (a greater magnitude in k_{float}) appears beneficial, compared to the values previously used. All of the optimizations converge near or below the generator speed constraint of 1.25.

Because we know the controllers have been tuned to minimize tower damage, we can fairly evaluate the platforms' effect on the tower design. It appears that the spar platform results in the greatest tower loading, followed by the semisubmersible, barge, and TLP. The reduced tower loading must be offset by the additional mooring complexity of the TLP and barge, and the substructure design complexity of the semisubmersible.

6. Conclusions and future directions

In this article, we demonstrate the WEIS optimization framework and its ability to optimize the parameters of the ROSCO for FOWTs. Optimizations converge to control solutions that

Table 6. Summary of optimal design variables and merit figures for the various FOWT platforms. The tower base DELs are relative to the original IEA 15-MW control parameters.

	Spar	Semi	Barge	TLP
$\omega(12)$, $\omega(18)$	0.125, 0.170	0.205, 0.220	0.104, 0.246	0.210, 0.290
$\zeta(12)$, $\zeta(18)$	1.62, 1.60	1.72, 1.69	1.51, 1.35	1.16, 1.06
k_{float}, ω_{ptfm}	-22.7, 0.312	-19.9, 0.303	-22.6, 0.330	-13.3, 0.286
$m_{ty,\text{DEL}}$	90.5 (-21.7%)	82.0 (-16.6%)	73.0 (-17.9%)	66.6 (-4.08%)
$\omega_g^{\max}/\omega_g^{\text{rated}}$	1.23	1.21	1.2	1.19

balance tower loading with generator speed regulation, an inherent trade off due to the negative damping that blade pitch control contributes to floating platform dynamics. This optimization procedure is used to update the ROSCO parameters of the IEA 15-MW reference turbine, which reduces the tower base DELs by about 10% as compared to the original controller parameters, and enforces an overspeed constraint where the generator speed cannot exceed 125% of the rated value. We observed diminishing returns if additional flexibility (i.e., more breakpoints in the pitch control parameterization) is allowed in the pitch control gain scheduling. Next, we studied the effect of peak shaving. Because the peak shaving percentage is coupled with the other pitch control parameters, we solved the same optimization problem for a range of peak shaving parameters and found that a 1% increase in AEP results in about a 5% increase in tower base DELs. Using optimal control solutions for a set of four different platform models, we found that a spar platform results in the greatest tower DELs, followed by semisubmersible, barge, and TLP. By tuning the controllers of these models using the same optimization problem, we make fair comparisons between the models.

In addition to more control design variables, the WEIS optimization framework also allows us to investigate platform, tower, and blade design of FOWTs in aeroelastic simulations. In future work, we will investigate the platform design, which is the primary capital cost of FOWTs, has a large effect on its motion, and has design coupling with the controller.

Acknowledgments

This work was authored in part by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by U.S. Department of Energy Advanced Research Projects Agency-Energy. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

[1] Abbas N J, Zalkind D S, Pao L and Wright A 2022 *Wind Energy Science* **7**
[2] Gaertner E *et al.* 2020 Definition of the IEA 15-megawatt offshore reference wind turbine Tech. Rep. NREL/TP-5000-75698 National Renewable Energy Laboratory Golden, CO
[3] National Renewable Energy Laboratory `https://github.com/openfast` 2021 *OpenFAST v2.3.0*
[4] NREL 2021 ROSCO. Version 2.3.0 URL `https://github.com/NREL/rosco`
[5] Hand M M and Balas M J 2000 *Wind Engineering* **24** 169–187
[6] Tibaldi C, Hansen M H and Henriksen L C 2014 *Journal of Physics: Conference Series* **555**
[7] Zalkind D S, Dall'Anese E and Pao L Y 2020 *Wind Energy Science* **5** 1579–1600
[8] International Electrotechnical Commission Wind turbines - part 1: design requirements
[9] Hegseth J M, Bachynski E E and Martins J R 2020 *Marine Structures* **72** 102771 ISSN 0951-8339
[10] Sandner F, Schlipf D, Matha D and Cheng P W 2014 **9B**
[11] Leimeister M, Collu M and Kolios A 2022 *Wind Energy Science* **7** 259–281

[12] NREL `https://github.com/WISDEM/WEIS` 2021
[13] NREL 2021 WISDEM, v3.2.0 `https://github.com/WISDEM/WISDEM`
[14] Jonkman B 2009 TurbSim User's Guide: Version 1.50 Tech. Rep. NREL/TP-500-46198 National Renewable Energy Laboratory
[15] Dykes K L, Zahle F, Merz K, McWilliam M and Bortolotti P 2017 IEA wind task 37: Systems modeling framework and ontology for wind turbines and plants Tech. rep. National Renewable Energy Lab.(NREL), Golden, CO (United States)
[16] Johnson S G The NLopt nonlinear-optimization package `https://github.com/stevengj/nlopt`
[17] Powell M J 1994 A direct search optimization method that models the objective and constraint functions by linear interpolation *Advances in optimization and numerical analysis* (Springer) pp 51–67
[18] Svanberg K 2002 *SIAM Journal on Optimization* **12** 555–573
[19] Kraft D *et al.* 1988
[20] Runarsson T P and Yao X 2005 *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **35** 233–243
[21] Jones D R, Perttunen C D and Stuckman B E 1993 *Journal of Optimization Theory and Applications* **79** 157–181
[22] Gray J S, Hwang J T, Martins J R, Moore K T and Naylor B A 2019 *Structural and Multidisciplinary Optimization* **59** 1075–1104
[23] Martins J R and Ning A 2021 *Engineering Design Optimization* (Cambridge University Press)
[24] Hansen M and Henriksen L 2013 *Basic DTU Wind Energy controller* (*DTU Wind Energy E* no 0028) (Denmark: DTU Wind Energy)
[25] Lemmer (né Sandner) F, Yu W, Schlipf D and Cheng P W 2020 *Wind Energy* **23** 17–30