Google / Makani Energy Kite Modeling

Cooperative Research and Development Final Report

CRADA Number: CRD-18-00569

NREL Technical Contact: Jason Jonkman

**NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Technical Report
NREL/TP-5000-80635
August 2021

Google / Makani Energy Kite Modeling

Cooperative Research and Development Final Report

CRADA Number: CRD-18-00569

NREL Technical Contact: Jason Jonkman

Suggested Citation

Jonkman, Jason. 2021. Google / Makani Energy Kite Modeling: *Cooperative Research and Development Final Report, CRADA Number CRD-18-00569*. Golden, CO: National Renewable Energy Laboratory. NREL/TP-5000-80635.
<https://www.nrel.gov/docs/fy21osti/80635.pdf>.

**NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Technical Report
NREL/TP-5000-80635
August 2021

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

NOTICE

This work was authored 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 Office of Energy Efficiency and Renewable Energy Wind Energy Technologies Office. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government.

This work was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, its contractors or subcontractors.

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via www.OSTI.gov.

Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.

NREL prints on paper that contains recycled content.

Cooperative Research and Development Final Report

Report Date: August 3, 2021

In accordance with requirements set forth in the terms of the CRADA agreement, this document is the CRADA final report, including a list of subject inventions, to be forwarded to the DOE Office of Scientific and Technical Information as part of the commitment to the public to demonstrate results of federally funded research.

Parties to the Agreement: Makani Technologies LLC (Google, Inc; Google LLC,)

CRADA Number: CRD-17-00569

CRADA Title: Google / Makani Energy Kite Modeling

Responsible Technical Contact at Alliance/ National Renewable Energy Laboratory (NREL):

Jason Jonkman jason.jonkman@nrel.gov

Name and Email Address of POC at Company:

Ruth Marsh ruthmarsh@live.com

Sponsoring DOE Program Office(s):

Office of Energy Efficiency and Renewable Energy (EERE), Wind Energy Technologies Office

Joint Work Statement Funding Table showing DOE commitment:

No NREL Shared Resources

Estimated Costs	NREL Shared Resources a/k/a Government In-Kind
Year 1	\$0.00
TOTALS	\$0.00

Executive Summary of CRADA Work:

In support of Makani's energy kite development program, Makani engaged with NREL to develop, verify, and document a multiphysics engineering model of a megawatt-scale tethered energy kite (named KiteFAST). After the original development of KiteFAST was completed, Makani further engaged NREL to develop a software enhancement of KiteFAST to enable the dynamics modeling of an energy kite tethered to a floating offshore platform (named KiteFAST-OS). Before this project, the capability to model the aero-hydro-servo-elastic dynamics of an airborne wind energy (AWE) system with electricity generation on the flying device (fly gen) and crosswind flight operation did not exist. Such physics-based modeling capability is needed

for loads analysis, structural design, and certification of energy kites. Throughout the project, an exclusive license kept KiteFAST and KiteFAST-OS available only to Makani, NREL, and its counterparts. With the ending of Makani, the KiteFAST and KiteFAST-OS software were merged into a single code base (named KiteFAST), the exclusive license was terminated, and the source code, documentation, and example models were released publicly.

Summary of Research Results:

KiteFAST has the potential to become the leading engineering tool for detailed design (loads analysis) of fly-gen-type AWEs. The following list summarizes the key outcomes in terms of software records, license, publication, and the repository containing the source code, documentation, and sample models:

- KiteFAST software model NREL Software Record # SWR-18-56
- KiteFAST-OS software model NREL Software Record # SWR-19-58
- Time-limited exclusive Software License Agreement (#LIC-19-349), now terminated
- One conference publication at AIAA SciTech 2019 [1]
- Software repository (including source code, documentation, and example models):
<https://github.com/OpenFAST/KiteFAST>
- Online user documentation:
<https://kitefast.readthedocs.io/en/kitefast/source/user/kitefast/index.html>
- Online developer documentation:
<https://kitefast.readthedocs.io/en/kitefast/source/dev/index.html#kitefast-development>
- Online examples and test cases:
<https://kitefast.readthedocs.io/en/kitefast/source/examples/kitefast/index.html>.

The guiding principles used to develop KiteFAST included:

- Capture important physical phenomena and system couplings to enable loads analysis, including excitation from wind/aerodynamics and waves/current/hydrodynamics, full-system dynamics (fuselage, wings, stabilizers, pylons, nacelles, rotors, tether, ground station/buoy, control surfaces, controller), aero-hydro-servo-elastic coupling, focus on operational mode (crosswind flight), and prioritize which physics are most important
- Use existing or modified FAST/OpenFAST modules where possible
- Use a commercial multibody dynamics software for primary structural dynamics
- Develop new physics models where necessary (aerodynamics)
- Verify to ensure correctness
- Document theory and usage.

At the project conclusion, KiteFAST generally functioned as expected, but there were problems with:

- MBDyn documentation / technical support
- Interfacing of Makani's internal controller
- Robustness of the new aerodynamic solver (vortex step method)
- Slower than desired simulation times.

These robustness and useability issues, which were not fully resolved by the conclusion of the project, should be addressed in follow-on work. Other important next steps include:

- Validation of KiteFAST against measured data
- Addressing limitations, capturing more physics, and making geometry more generic in KiteFAST
- Modeling modes other than crosswind flight
- Measuring turbulence up to 400 m and developing AWE-specific turbulence models
- Addressing wind-plant effects.

Over the course of the project, NREL also learned a lot about Makani's M600 energy kite technology (see Figure 1), design process, and market focus. Makani focused on offshore utility-scale energy generation and—because of its minimization of materials—expected to be a strong competitor relative to conventional floating offshore wind turbines that consist of a horizontal-axis wind turbine atop a tower and moored substructure. The M600 technology consisted of a ground station for land-based systems, or moored floating buoy for offshore systems, upon which the energy kite would perch when not operating, a tether that could reel out and in that is connected to the energy kite through a bridle, and the energy kite. During power production, the M600 would fly crosswind loops. The M600 had a flight controller that supported the ability to hover, vertically take off and land, and transition in and out of operational mode; see Figure 2. The M600 is a very dynamically active structure. The flight controller is extremely sophisticated and the key to Makani's success.

Figure 1: Overview of Makani's energy kite

Figure 2: Flight control operations

The project had eight tasks, whose results are summarized next.

Task 1: Project Management

This task was used to track and report on the progress of the project, track financials, and host regular (approximately monthly) meetings between NREL and Makani to discuss the project and next steps. NREL delivered monthly financial updates and quarterly reports that summarized the key accomplishments and status of the schedule and financials. Three physical face-to-face meetings were held between NREL and Makani, including the kick-off meeting at Makani, a mid-project review meeting at NREL, and a final project meeting at Makani, where the full project results were presented.

Task 2: Selection of Commercial MBS Software

The capability, customizability, and cost of various commercial multibody dynamics (MBS) software were assessed to form the basis of the main structural dynamics model of KiteFAST. Functional requirements of this MBS software included:

- Allows user to interface to user-routines written in Fortran and C
- Compatibility with Ubuntu and Debian used at Makani
- Permits the import of 6x6 sectional mass/stiffness matrices for flexible bodies
- Supports rigid bodies, rigid and rotation joints, and flexible nonlinear beam elements
- Supports nonlinear time-domain simulation
- Supports a linearization option (preferred, but not mandatory)
- Licensing cost.

The MBS software assessed included Altair HyperWorks, CP-Lambda, DYMORE, FEDEM, HAWC2, MBDyn, Modelica, MSC.ADAMS, SAMCEF, SIM-MECHANICS/SIMSCAPE, and SIMPACK as well as an option to develop a new (custom-made) MBS model. DYMORE and MBDyn were downselected as the top contenders due to their capabilities and open-source availability. [MBDyn](#) was finally selected.

The key attributes of MBDyn include:

- Free, public, open-source, combined finite element and MBS solver
- Released under the GNU GPL 2.1 license
- Developed by Prof. Pierangelo Masarati et al of Politecnico di Milano
- Used in the aircraft, rotorcraft, wind energy, automotive, and mechatronics industries
- PC/Linux both supported
- Written in C/C++
- User elements in C, C++, or Fortran
- FE beams support 6x6 sectional stiffness matrices
- Mass is lumped at nodes
- Supports many types of joints
- Nonlinear time-domain simulation via the backward differentiation formula (BDF) integrator

- Partially supports linearization (work in progress)
- Post-processor graphics through [Blender](#) or [EasyAnim](#).

Task 3: Development of the energy kite model (KiteFAST)

The newly developed engineering model KiteFAST was developed to be [OpenFAST](#)-like, in that it models the important physical phenomena and system couplings, including the wind/aerodynamic excitation and full-system dynamics response (fuselage, tail, wings, pylon(s), nacelle(s), drivetrain(s), rotor(s), tether, ground station, control surfaces, and controller) under both normal (for fatigue) and extreme (for ultimate) loading conditions. The details of the functional requirements and modeling approach were established between NREL and Makani and included the following features:

- Flexible wing, fuselage, tail, and pylons
- Rigid rotor, with flexible coupling to the wing
- Tether structural dynamics, including bridle attachments
- Full-field turbulent wind-inflow
- Rotors modeled as quasi-steady actuator disks, with wakes assumed independent
- Quasi-steady aerodynamic loads on wing, tail, pylons, and tether
- Variable-speed torque (paired between rotors)
- Control surfaces (flaps, ailerons, elevators, and rudder)
- Post-processing of fatigue and extreme loads via use of existing tools.

KiteFAST was developed to meet these functional requirements based on the agreed-upon modeling approach and selected commercial MBS software ([MBDyn](#)), using existing or modified [OpenFAST](#) modules as appropriate (including InflowWind for wind-inflow processing, supporting TurbSim, MoorDyn for tether dynamics, and loads postprocessing via MCrunch, MLife, and MExtremes), and developing new physics models where necessary (aerodynamics). KiteFAST solves the aero-servo-elastic dynamics of the AWE system with electricity generation on the flying device (fly gen) in crosswind flight operation nonlinearly in the time domain. KiteFAST was developed following the requirements of the OpenFAST modularization framework [2]. A submodel hierarchy of KiteFAST is given in Figure 3.

Figure 3: KiteFAST submodel hierarchy

The general configuration supported by KiteFAST (Figure 4) includes:

- One fuselage
- Two wings (starboard, port) attached to fuselage
- One vertical stabilizer attached to fuselage
- Two horizontal stabilizers (starboard, port) attached to vertical stabilizer
- User-specified number of pylons attached to each wing (Makani's controller assumes two)
- Two rotors (top, bottom) per pylon
- Nominal circular motion of the energy kite in the port (left) direction
- Each component of the energy kite has a primary axis (x,y,z), with twist defined about the primary axis and offsets along the transverse axes are allowed, but assumed to be small relative to the primary direction.

Figure 4: KiteFAST coordinate system

The general control surfaces supported by KiteFAST include:

- User-specified number of flaps per wing (Makani controller assumes three)
- Two rudders on the vertical stabilizer
- Two elevators per horizontal stabilizer
- Variable rotor speed
- Variable rotor-collective blade-pitch angles.

A summary of each module of KiteFAST is given next.

KiteAeroDyn, Including the KiteAeroDyn Driver, AirfoilInfo, KiteVSM, and ActuatorDisk

KiteAeroDyn computes quasi-steady aerodynamic loads as part of the coupled aero-servo-elastic solution of KiteFAST. KiteAeroDyn can also be computed uncoupled from KiteFAST by prescribing time histories of the inputs and steady wind via the KiteAeroDyn Driver.

AirfoilInfo is an existing submodel of OpenFAST (from the AeroDyn aerodynamics module). AirfoilInfo computes quasi-steady airfoil aerodynamics based on airfoil polars (lift, drag and pitching moment versus angle of attack), including interpolation between multiple Reynolds numbers or control surface settings.

KiteVSM computes quasi-steady induction from lifting lines using the Vortex Step Method (VSM), which is based on Weissenger's method. Novel developments in this method include the use of airfoil polars to account for large angles of attack and controller surfaces. The vorticity is

distributed through a finite number of horseshoe vortices. The induction is calculated via the Biot-Savart law. The angle of attack, relative wind velocity, and circulation are solved as a set of nonlinear algebraic equations relating the circulation distribution with the aerodynamic applied loads. Theoretical details and preliminary verification are published in [1].

ActuatorDisk models the rotors of the energy kite as quasi-steady actuator disks to calculate three forces, three moments, and power dependent on the rotor speed, rotor-inflow relative wind speed, rotor-inflow skew angle, and rotor-collective blade-pitch angle. The loads are derived from table-look up based on user-specified nondimensional coefficients force, moment, and power coefficients.

InflowWind, Including TurbSim

InflowWind is a module of OpenFAST, unchanged in KiteFAST, that processes the undisturbed (ambient) wind inflow. InflowWind supports steady, uniform winds; uniform, but time-varying winds (from IECWind); full-field turbulence (from TurbSim); or user-defined wind.

TurbSim is a pre-processor for InflowWind that computes full-field stochastic realizations of synthetic wind turbulence. The inputs to TurbSim include the desired wind characteristics—including the mean wind profile, velocity spectrum, spatial coherence, and velocity component correlations (Reynolds stresses). International Electrotechnical Commission (IEC), site-specific, and user-defined turbulence models are available, including the ability to construct a wind field around known time histories at points.

MoorDyn

MoorDyn is also a module of OpenFAST, which models the dynamics of cable elements using a lumped-mass dynamics. MoorDyn is unchanged in KiteFAST and used to model the tether. MoorDyn supports a multi-segmented array of taut or catenary lines, elastic stretching and damping, nonlinear geometric stiffness, still-air aerodynamic drag (and added mass).

Controller

The project developed an interface between Makani's internally developed controller, written in C, and KiteFAST. The interface included a controller wrapper written in C, which was called by another controller wrapper written in Fortran, which itself was called by KiteFAST; see Figure 5. A Fortran-based standalone controller driver was also developed so that the controller could be called uncoupled from KiteFAST (for debugging purposes).

Figure 5: KiteFAST controller interface

MBDyn

As discussed above, MBDyn is a general purpose MBS solver. The nacelles and rotors in KiteFAST are models as rigid bodies. The flexibility of the fuselage, wings, stabilizers, and pylons in KiteFAST are modeled via 3-node, 18-degree of freedom straight-beam finite elements. Features of these finite elements include:

- Stiffness is defined at two Gauss points, where internal loads may also be computed
- Lumped mass/inertia is defined at the three finite-element nodes
- External loads are applied at the three finite-element nodes
- The finite elements can be made rigid by adding constraints (computationally expensive).

The MBDyn input files are very verbose, complex, and in a nonstandard format. This necessitated the need to develop a preprocessor for generating MBDyn input files of energy kites—named the KiteMBDyn Preprocessor in KiteFAST. MBDyn interfaces to the various modules of KiteFAST through a user element, which is named KiteFASTMBD in KiteFAST.

KiteMBDyn Preprocessor

The KiteMBDyn Preprocessor abstracts the model definition and provides a user-friendly interface to MBDyn for energy kites, including a YAML file format that uses energy-kite terminology, modeling guidance with error checking, and basic pre- and post-processing. The inputs to the KiteMBDyn Preprocessor specify:

- Structural definition of the energy kite
- Solver settings
- KiteFAST modules to enable/disable
- Initial conditions
- Outputs requested.

The main components of the energy kite are defined by:

- Key point (origin of each component)
- Finite element end-node positions, twist, additional point masses (internal finite-element nodes are added automatically)
- Cross-sectional stiffness (the upper half of symmetric 6x6) (these are internally interpolated to the Gauss points)
- Cross-section mass and inertia (these are internally integrated and lumped at the finite-element nodes)
- Mapping of other attached components; for example, noting where the wing root node attaches to the fuselage.

The nacelles and rotors are defined by lumped masses, inertias, and center of mass offsets.

KiteFASTMBD

KiteFASTMBD is a custom-written MBDyn user element that connects MBDyn with the other modules of KiteFAST. The basic features of KiteFASTMBD include:

- Follows the MBDyn interface scheme rather than that of OpenFAST's modularization framework
- Applies mesh-to-mesh mapping from OpenFAST between MBDyn, KiteAeroDyn, and MoorDyn
- MoorDyn is called for both the prediction and correction steps of MBDyn
- KiteAeroDyn and InflowWind are called once per KiteAeroDyn time step (uses interpolation / extrapolation)
- Controller is called once per Controller time step (uses zero-order hold)
- Models spinning rotors / drivetrains analytically (assumed axisymmetric)
- Writes output data to a file.

The KiteFASTMBD source code is written in both C++ and Fortran. The KiteFASTMBD C++ code:

- Implements an MBDyn loadable module (user-defined element);
- Sends simulation information (time step, output flags) to Fortran
- Sends structural motions to Fortran and receives aerodynamic, hydrodynamic, tether, mooring loads which are applied to the MBDyn model object; and
- Couples the OpenFAST physics modules to the MBDyn solver and advances time by calling the Fortran interface.

The KiteFASTMBD Fortran code:

- Creates Fortran interfaces corresponding to the various MBDyn user defined element functions (Constructor, Assemble Residual, Output, Deconstructor)
- Communicates with the physics modules using the OpenFAST framework calls (Module_Init, Module_UpdateStates, Module_CalcOutput, etc.)
- Manages inputs, states, and outputs for all time-domain calculations outside of the structural-elastic modeling (ambient wind, aerodynamics, tethers, mooring lines, hydrodynamics)
- Receives structural motions from C++ (directly from MBDyn) and sends aerodynamic, hydrodynamic, tether, mooring loads back to C++
- Handles mapping between MBDyn structural meshes and the various KiteFAST meshes (VSM, Tether, Mooring, Hydrodynamic).

Task 4: Verification and validation of the energy kite model (KiteFAST)

The functional KiteFAST tool was verified to ensure its correctness by comparing the results obtained with KiteFAST against the solutions from two software used internally by Makani:

- [ASWING](#) is a steady-state aero-elastic solver for flexible aircraft from MIT and
- CSim is a six degree of freedom rigid-body model of the energy kite developed within Makani and interfaced to the internal Makani controller.

The verification of KiteFAST was achieved by gathering the configuration properties of various energy kites, assembly a model within KiteFAST, defining load-case simulations, running the simulations in KiteFAST and ASWING or CSim, and comparing the results in a side-by-side fashion. The results from three case studies are summarized next.

Verification of aerodynamics from KiteAeroDyn against ASWING for a simplified energy kite

A simplified aerodynamics-only model of the M600 energy kite was simulated, with symmetric geometry; only the wing, fuselage, vertical tail, and horizontal tail modeled; and no pylons, rotors, tether, or bridles considered. The simplified M600 energy kite was simulated for simple maneuvers in free flight, including forward flight, forward flight with pitch sweeps, forward flight with yaw sweeps, a banked horizontal turn, forward flight with vertical motion, and forward flight with control surface action. Overall, the results compare quite well between KiteAeroDyn and ASWING, with small differences associated with differences in the underlying model assumptions. Exemplary results are given in Figure 6.

Forward Flight

Pitch Angle Sweep

Yaw Angle Sweep (-20° Pitch Angle)

Banked Horizontal Turn

Forward Flight with Vertical 5-m/s Velocity

10° Flap Doublet

Figure 6: Aerodynamic comparison between KiteAeroDyn and ASWING

Verification of natural frequencies from MBDyn against ASWING for the M600

An eigenanalysis was performed in both MBDyn and ASWING for the M600 energy kite. The results for the first eight modes are quantified in Table 1 and visualized in Figure 7. There is relatively good agreement between the solutions. Differences can be explained by the variations in some of the specified stiffness terms that were not corrected.

Table 1: Eigenanalysis comparison between MBDyn and ASWING

Mode	MBDyn (Hz)	ASWING (Hz)	Description
1	2.05	2.38	Flapwise bending, Symmetric
2	4.41	3.61	Flapwise bending, asymmetric
3	4.04	3.82, 3.84	Pylons, asymmetric
4	4.52	3.89	Pylons, uniform
5	4.11	3.94	Pylons, symmetric
6	4.99	4.31	Edgewise
7	5.51	4.58	Wing torsion with pylon engagement
8	5.33	4.71	Fuselage bending, asymmetric flap

Mode 1: Flapwise Bending, Symmetric

Mode 2: Flapwise Bending, Asymmetric

Mode 3: Pylons, Asymmetric

Mode 4: Pylons, Uniform

Mode 5: Pylons, Symmetric

Mode 6: Edgewise

Mode 7: Wing Torsion With Pylon Engagement

Mode 8: Fuselage Bending, Asymmetric Flap

Figure 7: Eigenanalysis Comparison Between MBDyn (Left) and ASWING (Right)

Verification of tether loads for prescribed crosswind loops from KiteFAST against CSim for the M600

The tether and bridle loads in the M600 model were compared between KiteFAST and CSim for prescribed crosswind loops. In this test, the wind speed was 10 m/s without turbulence. The tether loads applied to the kite and the bridle forces are shown in Figure 8. Overall the results are in good agreement, with some high-frequency noise in the KiteFAST solution, which was caused by the way the flight trajectory was prescribed in MBDyn (this was fixed in future simulations, not shown here).

Figure 8: Tether and bridle load comparison between KiteFAST and CSim

Task 5: Documentation of the energy kite model (KiteFAST)

The theory and usage of KiteFAST was well documented and includes:

- KiteFAST user's guide, including an overview, general workflow, and modeling guidance
- KiteMBDyn Preprocessor user's guide, theory manual, and developer's guide
- MBDyn user-defined element guide
- KiteFASTMBD implementation plan
- KiteAeroDyn user's guide and theory manual, KiteAeroDyn implementation plan, KiteVSM implementation plan, and the AIAA SciTech 2019 conference paper [1].

Supplemental documentation that previously existed, but is pertinent to the use of KiteFAST, includes:

- MBDyn documentation
- InflowWind user's guide
- TurbSim user's guide
- MoorDyn user's guide and theory manual.

Task 6: Add floating functionality as KiteFAST-OS (Development of KiteFAST-OS)

KiteFAST-OS add functionality to KiteFAST by enhancing the original development to enable the dynamics modeling of an energy kite tethered to a floating offshore platform. The existing [OpenFAST](#) capability to model floating offshore platforms was used as much as possible, including the use of the HydroDyn hydrodynamics module to model the hydrodynamics of the floating platform and the MoorDyn mooring-dynamics module to model the mooring system. KiteFAST and KiteFAST-OS were separated for the purposes of licensing. But at the project conclusion, when Makani ended and the license was terminated, KiteFAST and KiteFAST-OS were merged into a single code based, named KiteFAST, that can model both floating offshore and land-based energy kites.

A submodel hierarchy of KiteFAST-OS is given in Figure 9 and the physical control volumes are illustrated in Figure 10.

Figure 9: KiteFAST-OS submodel hierarchy

Figure 10: KiteFAST control volumes

The KiteMBDyn Preprocessor was extended to include the buoy as an additional component. The buoy is modeled as a rigid body with several reference points, including the buoy reference point (0,0,0), a buoy inertial measurement unit, a wind measurement point (anemometer location), and ground station reference point.

KiteFASTMBD was extended to connect MBDyn with another instance of MoorDyn for modeling the mooring system and HydroDyn. MoorDyn and HydroDyn are called for both the prediction and correction steps of MBDyn. As part of this development, MoorDyn was updated to allow it to connect lines between two bodies that move relative to each other, i.e., the energy kite and the buoy.

HydroDyn is a module of OpenFAST, unchanged in KiteFAST-OS. HydroDyn models the hydrodynamic loads via a hybrid combination of slender members (dominated by viscous effects) and/or large-volume bodies (where radiation and diffraction are important). Slender members are modeled with an advanced strip-theory solution using extensions to the relative form of the Morison equation. Large-volume bodies are modeled with a potential-flow solution, which is converted to the time domain from the frequency domain. Wave kinematics are based on linear (Airy) plus second-order (Sharma and Dean) wave theory for regular or irregular sea states, including directional spreading and sea currents.

Task 7: Demonstrate the floating functionality of KiteFAST-OS (Verification of KiteFAST-OS)

The functional KiteFAST-OS tool was verified to ensure its correctness by comparing the results obtained with KiteFAST-OS against the solutions from OpenFAST for the moored floating buoy only. The OpenFAST model was first verified against OrcaFlex (not shown here), a commercial mooring dynamics software used by Makani. As in the verification of KiteFAST, the verification of KiteFAST-OS was achieved by comparing the simulation results in a side-by-side fashion. The results from this case study are summarized next, followed by a demonstration of the full system capability of KiteFAST-OS.

Verification of buoy response from KiteFAST-OS against OpenFAST for the moored floating buoy only

The motion response of the mooring floating buoy (without the M600 attached) excited by an irregular sea state was compared between KiteFAST-OS and OpenFAST. In KiteFAST-OS, a portion of the hydrodynamic added mass was moved into the buoy body mass to avoid numerical problems in the implicit fluid-structure interaction solution. The surge, heave, and pitch motions are shown in Figure 11. The results are in excellent agreement.

Figure 11: Buoy surge, heave, and pitch comparison between KiteFAST-OS and OpenFAST

Buoy response from KiteFAST-OS for the full floating offshore configuration of the M600 system

To demonstrate the functionality of KiteFAST-OS, a full-system model of the M600 energy kite tethered to the moored floating buoy was made in KiteFAST-OS, including wind inflow and irregular wave excitation. The kite is given prescribed motion (no active controller) operating in a crosswind loop. The results—shown in Figure 12—show the buoy being tugged from its initial position towards the operating kite. A screenshot from this demonstration is shown in Figure 13.

Figure 12: Buoy motions from the KiteFAST-OS demonstration

Figure 13: Screenshot of the KiteFAST-OS demonstration

Task 8: Document the floating functionality of KiteFAST-OS

The KiteFAST documentation was extended to reflect the enhanced floating offshore functionality. Supplemental documentation that previously existed, but is pertinent to the use of KiteFAST-OS, includes the HydroDyn user's guide and theory manual.

References:

- [1] Damiani, R.; Wendt, F.; Jonkman, J.; and Sicard, J. "A Vortex Step Method for Nonlinear Airfoil Polar Data as Implemented in KiteAeroDyn." *AIAA Science and Technology Forum and Exhibition (SciTech 2019)*, 7–11 January 2019, San Diego, CA [online proceedings]. URL: <https://doi.org/10.2514/6.2019-0804>. AIAA-2019-0804. Reston, VA: American Institute of Aeronautics and Astronautics, January 2019; NREL/CP-5000-72777. Golden, CO: National Renewable Energy Laboratory.
- [2] Jonkman, J. M. "The New Modularization Framework for the FAST Wind Turbine CAE Tool." *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 7–10 January 2013, Grapevine (Dallas/Ft. Worth Region), TX [online proceedings]. URL: <http://arc.aiaa.org/doi/pdf/10.2514/6.2013-202>. AIAA-2013-0202. Reston, VA: American Institute of Aeronautics and Astronautics, January 2013; NREL/CP-5000-57228. Golden, CO: National Renewable Energy Laboratory.

Subject Inventions Listing: None

ROI #: None