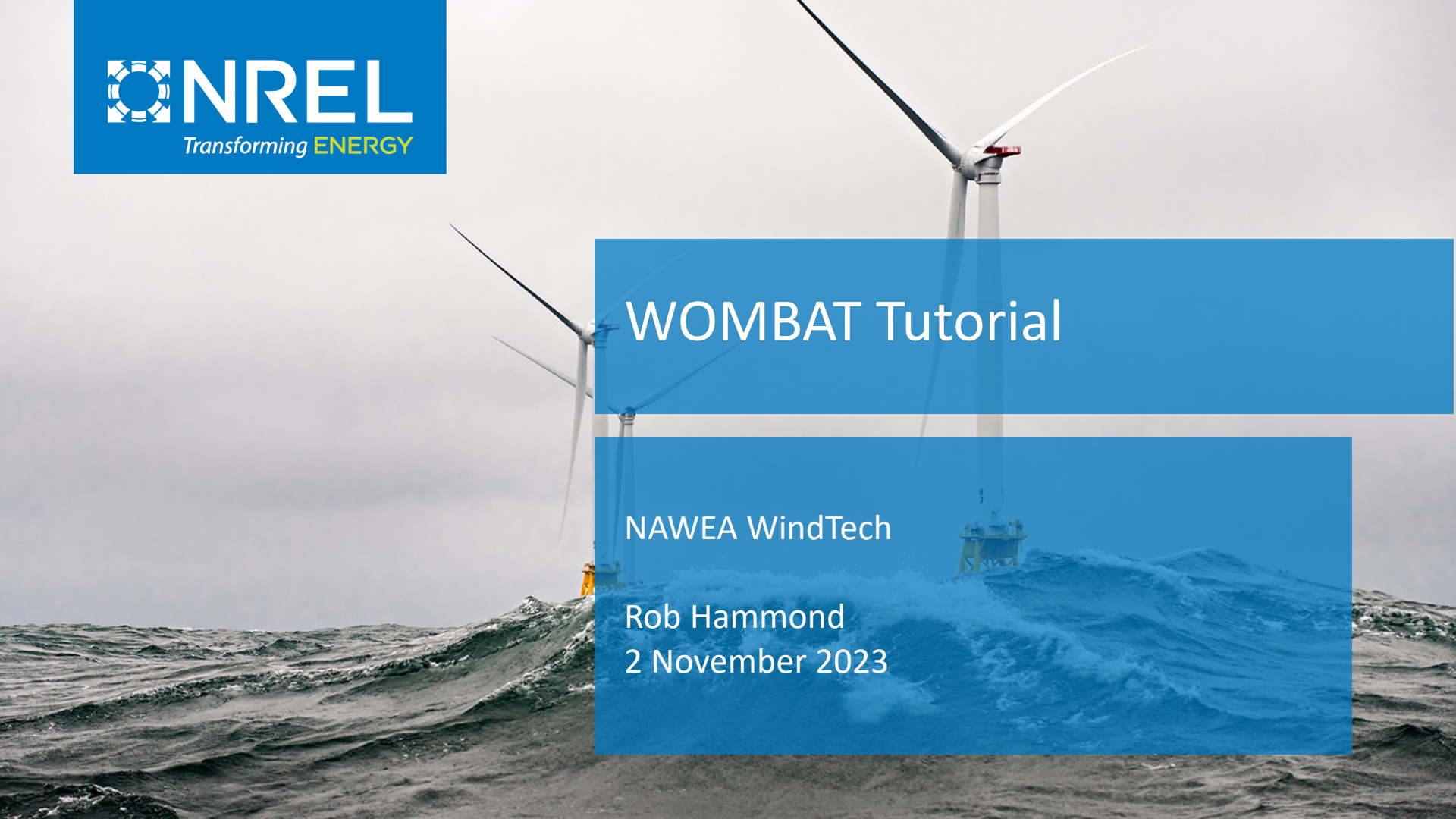


WOMBAT Tutorial

NAWEA WindTech

Rob Hammond
2 November 2023

1 WOMBAT At a Glance

2 Configuring WOMBAT

3 COREWIND Assumptions

4 Setting Up a Simulation

5 Hands-on With WOMBAT

6 Time Permitting: WAVES

7 Time Permitting: Future of WOMBAT

WOMBAT At a Glance

Relevance of O&M Costs

- O&M activities are estimated to comprise between 29% and 34% of total wind plant lifecycle costs (Stehly & Beiter, 2018).
  - $33 – $59/kW/year for land-based wind
  - $65 – $194/kw/year for offshore wind

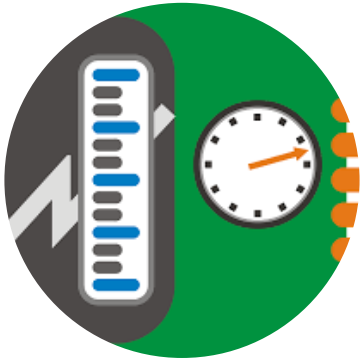
- Innovations in the O&M sector have the potential to drive down the overall cost of wind energy.

- However, quantifying the impact of these innovations on cost is challenging because:
  - Data on wind plant O&M costs are not often publicly available or broken down into detailed categories.
  - Understanding cost impacts and tradeoffs for O&M strategies requires a model with appropriate resolution to capture relatively small changes at the level of individual tasks.

Primary Research Question

How might maintenance strategies, technological innovations, and site conditions influence wind plant OpEx and ultimately LCOE?

Methodology Innovations



Technology Innovations

Site Conditions

Approach

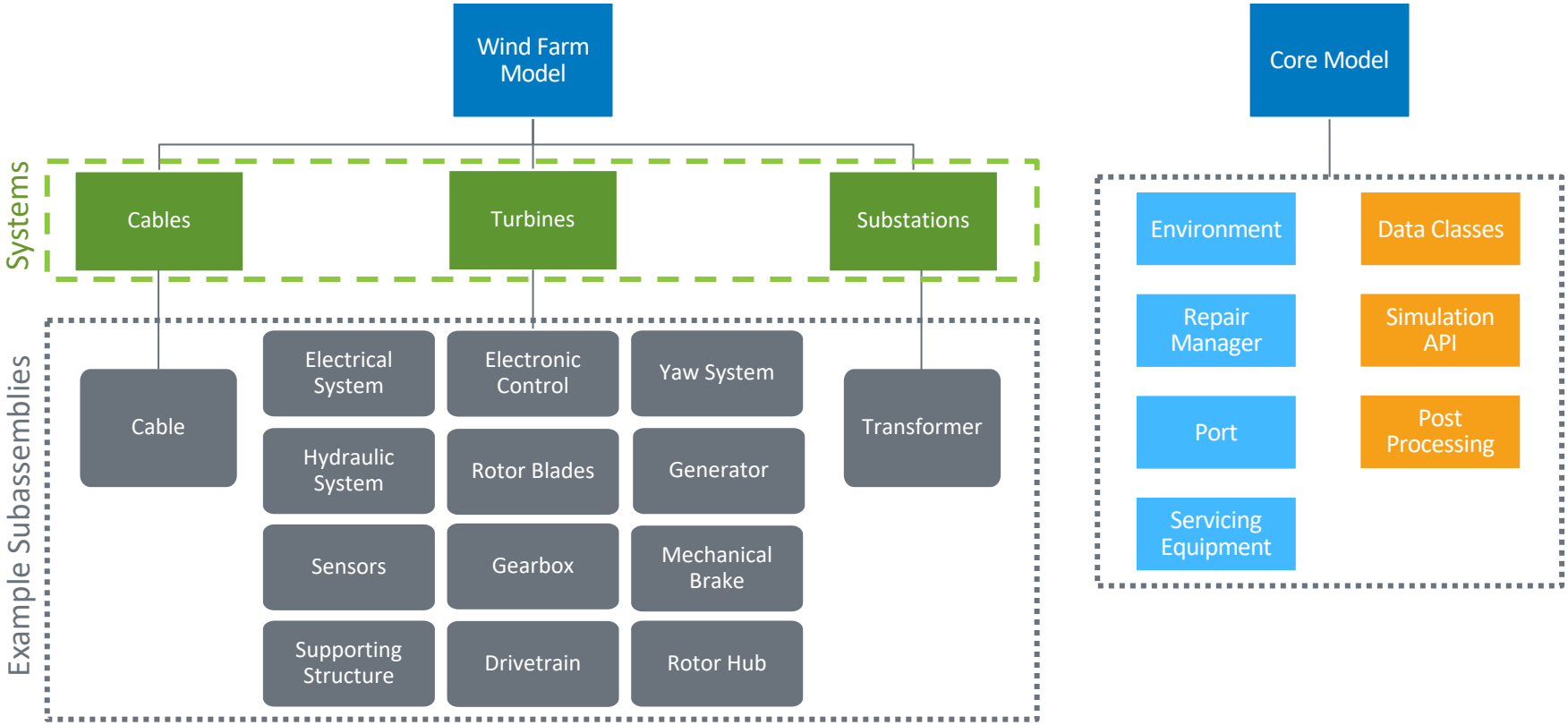
- Minimal code by the user to actual run the model
  - Empower a broad user base by requiring minimal Python skills
  - Configure models through YAML files, not complicated setup code

- Prescriptive modeling via discrete event simulation:
  - Enables weather and site-specific variability
  - Allows a user to define O&M strategies to understand cost and energy impacts
  - Focuses on what-if scenario modeling instead of optimization

- Modular and flexible code base:
  - Allows for new methodologies to be tested with ease
  - Provides a tool to analyze both offshore and land-based windfarm O&M costs
  - Allows users to define their turbine failure models in as much or as little detail as desired

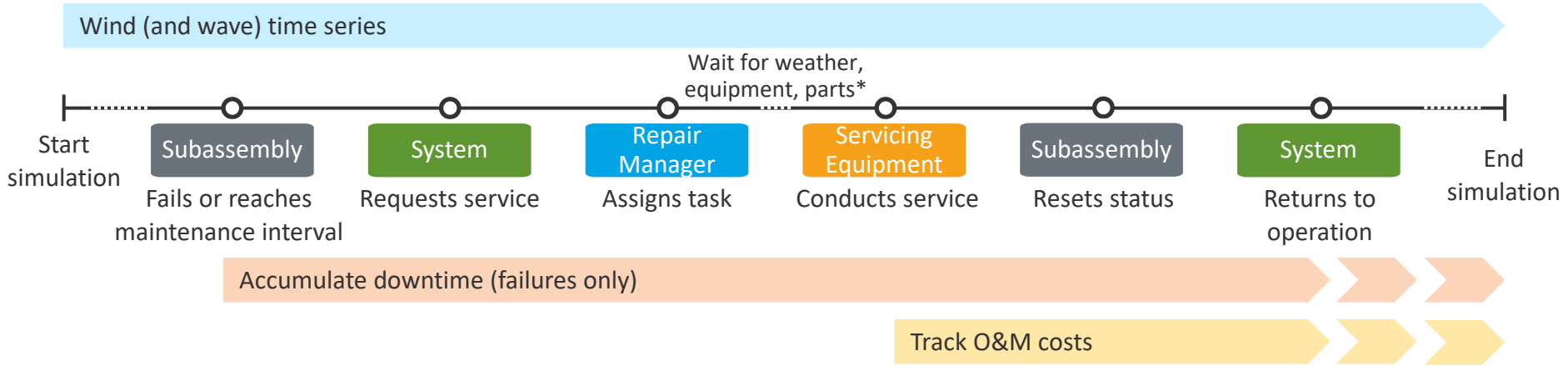
- Documentation driven development:
  - Use of inline comments and docstrings explains what nearly all functionality is supposed to be doing
  - Enables easier modification or improvement for researchers and/or developers

High-Level Software Architecture



High-Level Simulation Architecture

- Model evaluates O&M costs using discrete event simulation (series of events in sequential order where no changes occur between events):
  - Allows for detailed documentation of a system and its processes.
  - Allows for a prescriptive approach for exploring specific impacts compared to an optimization with a “best choice.”



*Parts availability not yet implemented

Configuring WOMBAT

Repairs and Maintenance

- Maintenance (scheduled)
  - Fixed frequency model (number of days) to align with regular servicing intervals
- Failures (unscheduled maintenance)
  - Weibull distribution (exponential if using 1 for the shape parameter)
    - Scale = 1 / (# Failures/system/year) = MTBF (in years)
  - Random sampling for the next time a failure occurs
- Shared parameterizations
  - Materials cost (whole cost or proportional to system CapEx)
  - Repair time
  - Service equipment categories that can perform corrective maintenance
    - CTV, DRN, RMT, SCN, LCN, CAB, DSV, TOW, AHV
  - Operating reduction
  - Severity level (helps prioritize repairs across the farm)

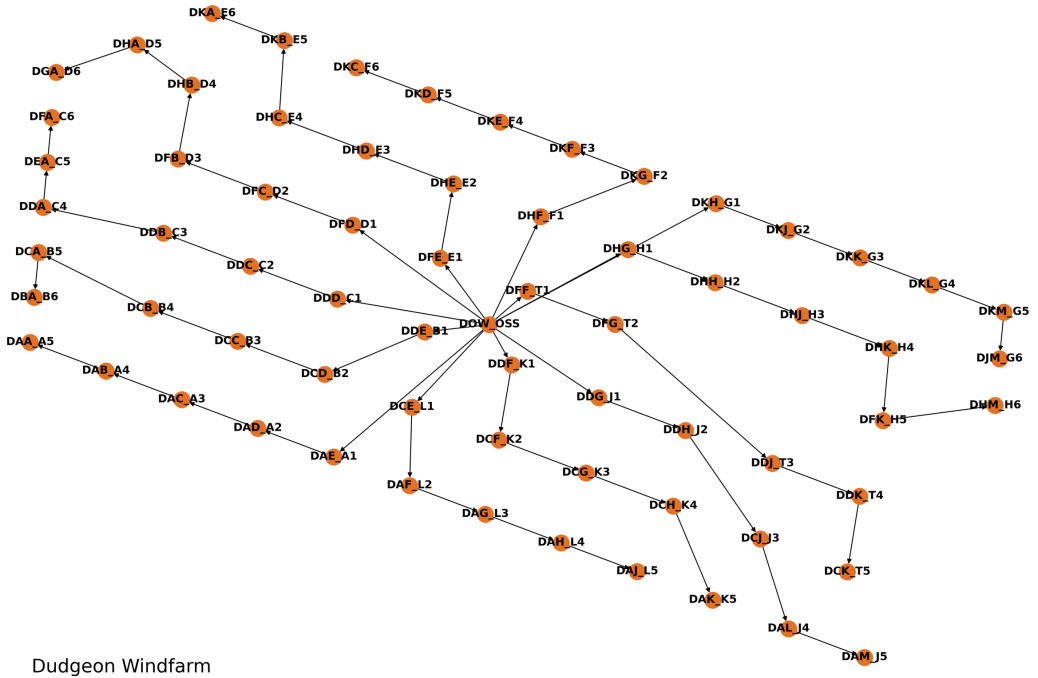
Servicing Equipment

- Strategy
  - Scheduled
  - Unscheduled-downtime
  - Unscheduled-requests
  - Unscheduled-tow (offshore only)
- Capabilities
- Working hours
- Mobilization
- Port distance
- Crew
- Weather constraints
  - Windspeed
  - Waveheight
  - Speed (including during inclement weather)
  - Non-operational periods
  - Slow down periods

The Wind Farm

CSV Layout

- Define each substation and turbine's positions relative to each other and spatially (WGS-84 coordinates)
- Provides pointers to the cable, substation, and turbine definition file
- Gives names and IDs to each of the systems on the farm so that subassembly definitions can be shared between common systems



Dudgeon Windfarm

Baseline Inputs

Subassemblies

- Failure rate(s)
- Maintenance tasks
- Equipment requirements
- Cost and time to complete repairs
- Power curve (if modeling energy production – IEC power curve)

Servicing Equipment

- Maintenance strategy and related definitions
- Port (if required)
- Capabilities
- Labor rates
- Equipment rates
- Operational limits

Miscellaneous

- Weather profile
  - Hourly windspeed and/or wave height
- Windfarm layout
- Site working hours
- Port distance

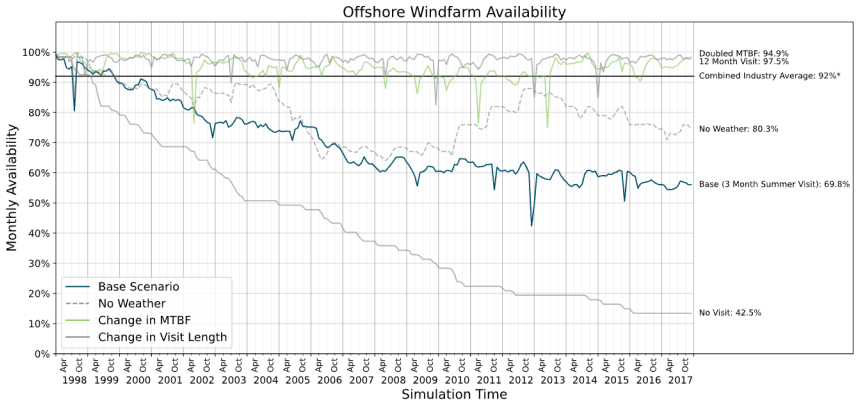
Outputs

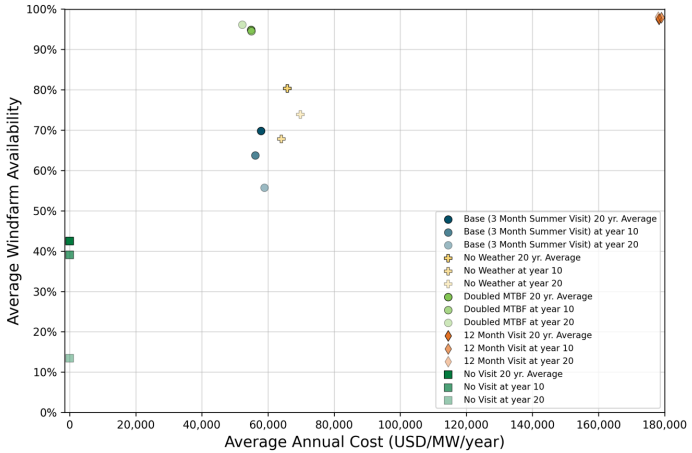
- Time-based availability
- Production-based availability
- Power production
- Fixed costs
- Capacity factor
- Task completion rate
- Service equipment costs
- Service equipment utilization
- Labor costs
- Combined service equipment and labor costs by productivity
- Component costs
- Servicing time breakdown
- OpEx
- More on the way

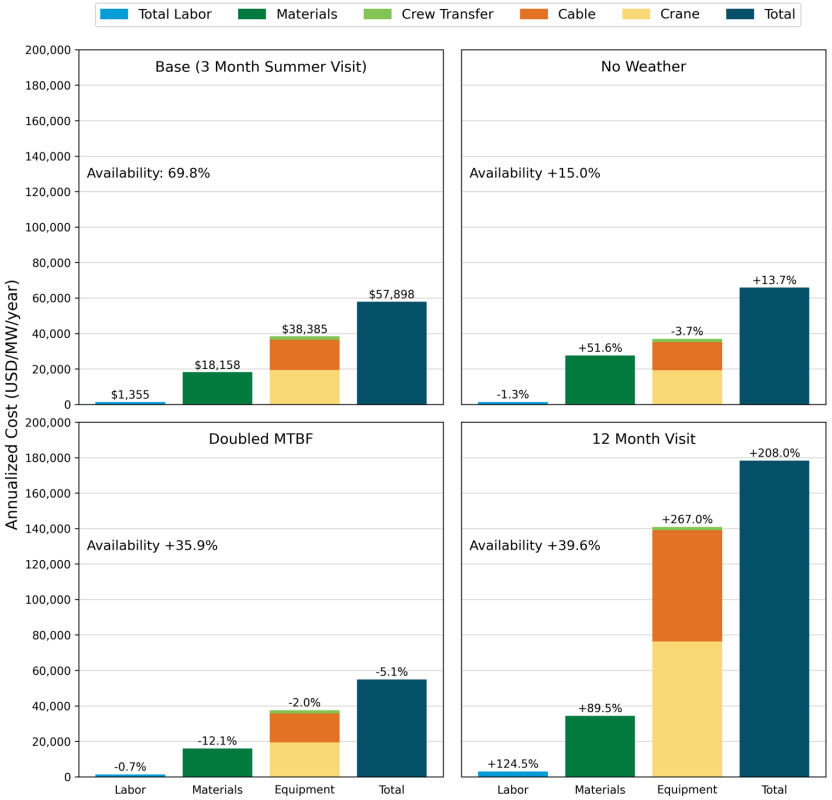
High fidelity log files to compute further metrics

- Event logs
- Operating level logs (hourly)
- Power production logs (hourly)
- Power potential logs (hourly)

Examples


Offshore Windfarm Availability
Monthly Availability
Simulation Time
Doubled MTBF: 94.9%
12 Month Visit: 97.5%
Combined Industry Average: 92%*
No Weather: 80.3%
Base (3 Month Summer Visit): 69.8%
No Visit: 42.5%
Base Scenario
No Weather
Change in MTBF
Change in Visit Length


Average Windfarm Availability
Average Annual Cost (USD/MW/year)
Base (3 Month Summer Visit) 20 yr. Average
Base (3 Month Summer Visit) at year 10
Base (3 Month Summer Visit) at year 20
No Weather 20 yr. Average
No Weather at year 10
No Weather at year 20
Doubled MTBF 20 yr. Average
Doubled MTBF at year 10
Doubled MTBF at year 20
12 Month Visit 20 yr. Average
12 Month Visit at year 10
12 Month Visit at year 20
No Visit 20 yr. Average
No Visit at year 10
No Visit at year 20


Total Labor
Materials
Crew Transfer
Cable
Crane
Total
Annualized Cost (USD/MW/year)
Base (3 Month Summer Visit)
Availability: 69.8%
$1,355
$18,158
$38,385
$57,898
No Weather
Availability +15.0%
-1.3%
+51.6%
-3.7%
+13.7%
Doubled MTBF
Availability +35.9%
-0.7%
-12.1%
-2.0%
-5.1%
12 Month Visit
Availability +39.6%
+124.5%
+89.5%
+267.0%
+208.0%
Labor
Materials
Equipment
Total

COREWIND Assumptions

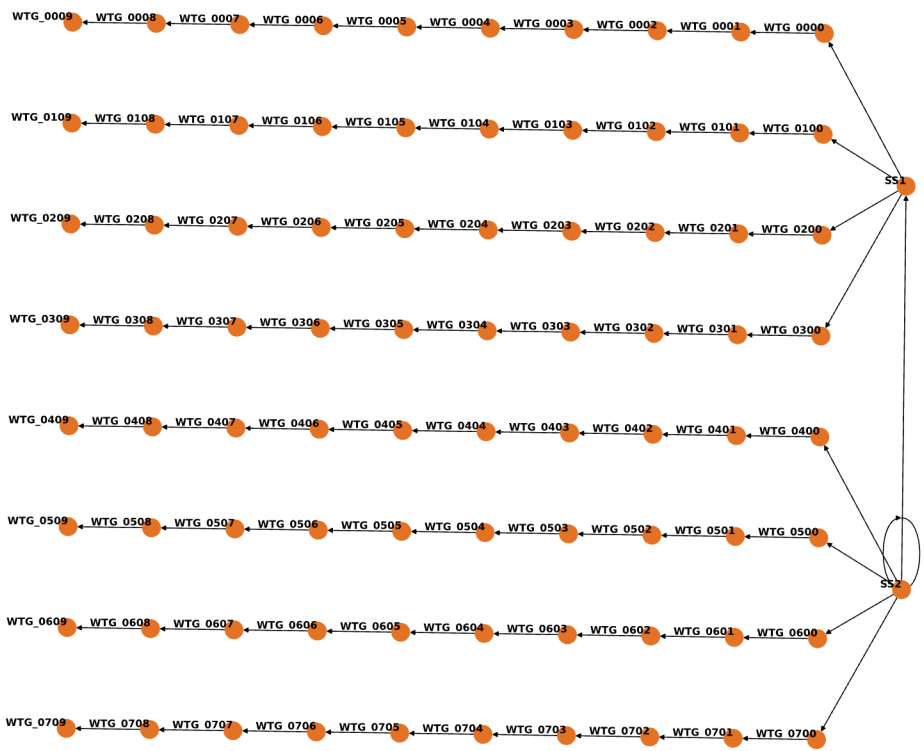
High Level Information

- Plant assumptions
  https://corewind.eu/wp-content/uploads/files/publications/COREWIND-D6.1-General-frame-of-the-analysis-and-description-of-the-new-FOW-assessment-app.pdf
- Assessment assumptions
  https://corewind.eu/wp-content/uploads/files/publications/COREWIND-D4.2-Floating-Wind-O-and-M-Strategies-Assessment.pdf

Summary

- 1.2 GW in Morro Bay, CA with reference coordinate: (35. 367222, -121. 645163)
- 60km distance to shore
- 80 x IEA 15 MW Reference Turbines, but with floating platforms, spaced 2160m apart
- 2 offshore substations
- 7 CTVs, Anchor Handling Vessel, Diving Support Vessel, Cable Lay Vessel, ROV
- Technician salary: €80K

Plant Layout



Failure Data

Component or Maintenance Task	Frequency (scheduled)	Failure Rate (N/turbine/year)	Duration (hours)	Technicians (number)	Cost (Euros)	Vessel
Sub-Sea Inspections of Cables, Mooring Lines and Floater Hull	2 years		12	5	500	SOV with ROV
Direct Drive Generator – Minor Failure		0.546	13	2	1000	CTV
Main Shaft – Replacement		0.009	144	5	232000	CTV

- Data doesn't quite match the expected format in WOMBAT
- Some assumptions don't quite align with expectations
- **So, how would this get translated to WOMBAT given what we know?**

Vessel Data

Vessel	Number	Purpose	Dayrate (Euros)	Mobilization Cost (Euros)	Waveheight Limit (m)
Crew Transfer Vessel (CTV)	7	Access to systems	3,500	0	2 (simplified from matrix)
Anchor Handling Vessel (AHV)	1	Mooring line and anchor repairs	55,000	500,000	2.5
SOV with ROV	2	Subsea inspection and repairs	75,000	225,000	2
Monohull Crane Vessel (HLV)	1	Major component exchange offshore	290,000	325,000	2
Tugboat	2	Tow-in and tow-out	30,000	200,000	2.5

- Data doesn't quite match the expected format in WOMBAT
- Some assumptions don't quite align with expectations
- **So, how would this get translated to WOMBAT given what we know?**

Setting Up a Simulation

Working Through the Simulation Setup

1. We will work through creating a piece of every simulation aspect in this part of the workshop from the layout to servicing equipment
2. Focus on understanding how to get each of the aspects set up and how everything interconnects, not exhaustively recreating the paper's assumptions
3. To run the model, ask new questions of the scenarios, generally tinker, we'll then switch to what is already available on the GitHub repository

Create a New Project

- Note: everything we need is already set up in the repository
- Open a VSCode (or preferred code editor/IDE) window
- In the terminal/Anaconda prompt
  1. “cd <your-path-the-workshop-data>”
  2. “python”
  3. “from wombat import create_library_structure”
  4. “create_library_structure(<your-library-name>)”

```
(wombat_workshop) [~] cd Documents/Projects/nawea_tutorial
(wombat_workshop) [~/Documents/Projects/nawea_tutorial] python
Python 3.10.13 (main, Sep 11 2023, 08:16:02) [Clang 14.0.6 ] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from wombat import create_library_structure
>>> new_library = "library/"
>>> create_library_structure(new_library)
>>>
```

Overarching configuration file

- Create a configuration file in <library>/project/config called “base.yaml”
- Now, we need to create most of the inputs on the right as direct inputs or pointers to other files: https://wisdem.github.io/WOMBAT/API/simulation_api.html#configuration
- To get started, I’ll directly guide this first part

- **name** (*str*) – Name of the simulation. Used for logging files.
- **layout** (*str*) – The windfarm layout file. See `wombat.Windfarm` for more details.
- **service_equipment** (*str | list[str]*) – The equpiment that will be used in the simulation. See `wombat.core.ServiceEquipment` for more details.
- **weather** (*str*) – The weather profile to be used. See `wombat.simulation.WombatEnvironment` for more details.
- **workday_start** (*int*) – Starting hour for a typical work shift. Can be overridden by equipment-specific settings.
- **workday_end** (*int*) – Ending hour for a typical work shift. Can be overridden by equipment-specific settings.
- **inflation_rate** (*float*) – The annual inflation rate to be used for post-processing.
- **fixed_costs** (*str*) – The file name for the fixed costs assumptions.
- **project_capacity** (*int | float*) – The total capacity of the wind plant, in MW.
- **port** (*dict | str | Path*) – The port configuration file or dictionary that will be used to setup a tow-to-port repair strategy, default None.
- **port_distance** (*int | float*) – The simulation-wide daily travel distance for servicing equipment. This should be used as a base setting when multiple or all servicing equipment will be operating out of the same base location, but can be individually modified.
- **start_year** (*int*) – Start year of the simulation. The exact date will be determined by the first valid date of this year in `weather`.
- **end_year** (*int*) – Final year of the simulation. The exact date will be determined by the last valid date of this year in `weather`.
- **non_operational_start** (*str | datetime.datetime | None*) – The starting month and day, e.g., MM/DD, M/D, MM-DD, etc. for an annualized period of prohibited operations. When defined at the environment level, an undefined or later starting date will be overridden for all servicing equipment and any modeled port, by default None.
- **non_operational_end** (*str | datetime.datetime | None*) – The ending month and day, e.g., MM/DD, M/D, MM-DD, etc. for an annualized period of prohibited operations. When defined at the environment level, an undefined or earlier ending date will be overridden for all servicing equipment and any modeled port, by default None.
- **reduced_speed_start** (*str | datetime.datetime | None*) – The starting month and day, e.g., MM/DD, M/D, MM-DD, etc. for an annualized period of reduced speed operations. When defined at the environment level, an undefined or later starting date will be overridden for all servicing equipment and any modeled port, by default None.
- **reduced_speed_end** (*str | datetime.datetime | None*) – The ending month and day, e.g., MM/DD, M/D, MM-DD, etc. for an annualized period of reduced speed operations. When defined at the environment level, an undefined or earlier ending date will be overridden for all servicing equipment and any modeled port, by default None.
- **reduced_speed** (*float*) – The maximum operating speed during the annualized reduced speed operations. When defined at the environment level, an undefined or faster value will be overridden for all servicing equipment and any modeled port, by default 0.0.
- **random_seed** (*int | None*) – The random seed to be passed to a universal NumPy `default_rng` object to generate Weibull random generators, by default None.
- **random_generator** (*np.random._generator.Generator | None*) – An optional numpy random generator that can be provided to seed a simulation with the same generator each time, in place of the random seed. If a `random_seed` is also provided, this will override the random seed, by default None.

Overarching configuration file - Answer

library > project > config > morro_bay_in_situ.yaml > ...

```yaml
name: COREWIND Morro Bay In Situ
library: library
weather: central_ca.csv
service_equipment:
  - ctv1.yaml
  - ctv2.yaml
  - ctv3.yaml
  - ctv4.yaml
  - ctv5.yaml
  - ctv6.yaml
  - ctv7.yaml
  - cab.yaml
  - dsv.yaml
  - ahv.yaml
  - hlv1.yaml
layout: morro_bay_9D_layout.csv
port_distance: 60  # 20/100 for West of Barra Island; 10 for Gran Canaria Island
inflation_rate: 0
workday_start: 6
workday_end: 22
start_year: 2002
end_year: 2021
project_capacity: 1200
```

Layout

- Create a layout CSV file in <library>/project/plant/ using the file name you used for “layout” in base.yaml
- Open it in Excel, Numbers, or whatever you prefer for editing tabular data
- Add the columns you see to the right: https://wisdem.github.io/WOMBAT/examples/how_to.html#windfarm-layout
- Create the first string for each substation with the information that we have (fill in optional pieces as you please)—layout on Slide 17
- NOTE: “subassembly” and “upstream_cable” just need the file names

id (required)
Unique identifier for the asset; no spaces allowed.

substation_id (required)
The id field for the substation that the asset connects to; in the case that this is a substation, then this field should be the same as id; no spaces allowed.

name (required)
A descriptive name for the turbine, if desired. This can be the same as id.

type (optional)
One of “turbine” or “substation”. This is required to accurately model a multi-substation wind farm. The base assumption is that a substation connects to itself as a means to model the export cable connecting to the interconnection point, however, this is not always the case, as substations may be connected through their export systems. Using this filed allows for that connection to be modeled accurately.

longitude (optional)
The longitudinal position of the asset, can be in any geospatial reference; optional.

latitude (optional)
The latitude position of the asset, can be in any geospatial reference; optional.

string (required)
The integer, zero-indexed, string number for where the turbine will be positioned.

order (required)
The integer, zero-indexed position on the string for where the turbine will be positioned.

distance (optional)
The distance to the upstream asset; if this is calculated (input = 0), then the straightline distance is calculated using the provided coordinates (WGS-84 assumed).

subassembly (required)
The file that defines the asset’s modeling parameters.

upstream_cable (required)
The file that defines the upstream cable’s modeling parameters.

upstream_cable_name (optional)
The descriptive name to give to the cable that will be used during logging. This enables users to use a single cable definition file while maintaining the naming conventions used for the wind farm being simulated.

Layout - Answer

4 strings per substation, plus a connections between them and the interconnection point (unmodeled)

Zero-indexing is required for string/order

	A	B	C	D	I	J	K	L	M	N	O	P
1	id	substation_id	name	type	longitude	latitude	string	order	distance	subassembly	upstream_cable	upstream_cable_name
2	SS1	SS2	SS1	substation	-121.6445116	35.40616396	9	0		corewind_substation.yaml	corewind_export.yaml	EXP00
3	SS2	SS2	SS2	substation	-121.6458132	35.32827984	9	1		corewind_substation.yaml	corewind_export.yaml	EXP01
4	WTG_0000	SS1	WTG_0000	turbine	-121.6678132	35.4356351	0	0		corewind_15MW.yaml	corewind_array.yaml	ARR00
5	WTG_0001	SS1	WTG_0001	turbine	-121.6916045	35.43589529	0	1		corewind_15MW.yaml	corewind_array.yaml	ARR01
6	WTG_0002	SS1	WTG_0002	turbine	-121.7153961	35.43615079	0	2		corewind_15MW.yaml	corewind_array.yaml	ARR02
7	WTG_0003	SS1	WTG_0003	turbine	-121.739188	35.43640161	0	3		corewind_15MW.yaml	corewind_array.yaml	ARR03
8	WTG_0004	SS1	WTG_0004	turbine	-121.7629802	35.43664773	0	4		corewind_15MW.yaml	corewind_array.yaml	ARR04
9	WTG_0005	SS1	WTG_0005	turbine	-121.7867727	35.43688918	0	5		corewind_15MW.yaml	corewind_array.yaml	ARR05
10	WTG_0006	SS1	WTG_0006	turbine	-121.8105655	35.43712593	0	6		corewind_15MW.yaml	corewind_array.yaml	ARR06
11	WTG_0007	SS1	WTG_0007	turbine	-121.8343586	35.437358	0	7		corewind_15MW.yaml	corewind_array.yaml	ARR07
12	WTG_0008	SS1	WTG_0008	turbine	-121.8581519	35.43758538	0	8		corewind_15MW.yaml	corewind_array.yaml	ARR08
13	WTG_0009	SS1	WTG_0009	turbine	-121.8819455	35.43780807	0	9		corewind_15MW.yaml	corewind_array.yaml	ARR09

	A	B	C	D	I	J	K	L	M	N	O	P
44	WTG_0400	SS2	WTG_0400	turbine	-121.6690938	35.35775062	4	0		corewind_15MW.yaml	corewind_array.yaml	ARR40
45	WTG_0401	SS2	WTG_0401	turbine	-121.6928623	35.35801007	4	1		corewind_15MW.yaml	corewind_array.yaml	ARR41
46	WTG_0402	SS2	WTG_0402	turbine	-121.7166311	35.35826484	4	2		corewind_15MW.yaml	corewind_array.yaml	ARR42
47	WTG_0403	SS2	WTG_0403	turbine	-121.7404001	35.35851493	4	3		corewind_15MW.yaml	corewind_array.yaml	ARR43
48	WTG_0404	SS2	WTG_0404	turbine	-121.7641695	35.35876036	4	4		corewind_15MW.yaml	corewind_array.yaml	ARR44
49	WTG_0405	SS2	WTG_0405	turbine	-121.7879391	35.35900111	4	5		corewind_15MW.yaml	corewind_array.yaml	ARR45
50	WTG_0406	SS2	WTG_0406	turbine	-121.811709	35.35923719	4	6		corewind_15MW.yaml	corewind_array.yaml	ARR46
51	WTG_0407	SS2	WTG_0407	turbine	-121.8354792	35.35946859	4	7		corewind_15MW.yaml	corewind_array.yaml	ARR47
52	WTG_0408	SS2	WTG_0408	turbine	-121.8592497	35.35969532	4	8		corewind_15MW.yaml	corewind_array.yaml	ARR48
53	WTG_0409	SS2	WTG_0409	turbine	-121.8830204	35.35991738	4	9		corewind_15MW.yaml	corewind_array.yaml	ARR49

Servicing Equipment

- 7 CTVs, 1 Anchor Handling Vessels, 2 SOVs with ROV, 1 Heavy Lift Vessel, 2 Tugboats
- Modifications needed: No SOVs (yet) in WOMBAT, so we will model 2 ROVs
- 2 classes of vessels: scheduled and unscheduled, so how would we classify the vessels in the first bullet?
- https://wisdem.github.io/WOMBAT/API/types.html#scheduled-service-equipment
- https://wisdem.github.io/WOMBAT/API/types.html#unscheduled-service-equipment

Scheduled Servicing Equipment

- D4.2, p. 18
- Create a YAML configuration file in "<library>/vessels/", for a single CTV.
- https://wisdem.github.io/WOMBAT/API/types.html#scheduled-service-equipment

- **name** (*str*) – Name of the piece of servicing equipment.
- **equipment_rate** (*float*) – Day rate for the equipment/vessel, in USD.
- **n_crews** (*int*) – Number of crew units for the equipment.

> **Note**
>
> The input to this does not matter yet, as multi-crew functionality is not yet implemented.

- **crew** (*ServiceCrew*) – The crew details, see `ServiceCrew` for more information. Dictionary of labor costs with the following: `n_day_rate`, `day_rate`, `n_hourly_rate`, and `hourly_rate`.
- **start_month** (*int*) – The day to start operations for the rig and crew.
- **start_day** (*int*) – The month to start operations for the rig and crew.
- **start_year** (*int*) – The year to start operations for the rig and crew.
- **end_month** (*int*) – The month to end operations for the rig and crew.
- **end_day** (*int*) – The day to end operations for the rig and crew.
- **end_year** (*int*) – The year to end operations for the rig and crew.

> **Note**
>
> if the rig comes annually, then the enter the year for the last year that the rig and crew will be available.

- **capability** (*str*) – The type of capabilities the equipment contains. Must be one of:
  - RMT: remote (no actual equipment BUT no special implementation)
  - DRN: drone
  - CTV: crew transfer vessel/vehicle
  - SCN: small crane (i.e., field support vessel)
  - LCN: large crane (i.e., heavy lift vessel)
  - CAB: cabling vessel/vehicle
  - DSV: diving support vessel

  Please note that "TOW" is unavailable for scheduled servicing equipment
- **mobilization_cost** (*float*) – Cost to mobilize the rig and crew.
- **mobilization_days** (*int*) – Number of days it takes to mobilize the equipment.
- **speed** (*float*) – Maximum transit speed, km/hr.
- **speed_reduction_factor** (*flaot*) – Reduction factor for traveling in inclement weather, default 0. When 0, travel is stopped when either *max_windspeed_transport* or *max_waveheight_transport* is reached, and when 1, *speed* is used.
- **max_windspeed_transport** (*float*) – Maximum windspeed for safe transport, m/s.
- **max_windspeed_repair** (*float*) – Maximum windspeed for safe operations, m/s.
- **max_waveheight_transport** (*float*) – Maximum waveheight for safe transport, m, default 1000 (land-based).
- **max_waveheight_repair** (*float*) – Maximum waveheight for safe operations, m, default 1000 (land-based).
- **workday_start** (*int*) – The starting hour of a workshift, in 24 hour time.
- **workday_end** (*int*) – The ending hour of a workshift, in 24 hour time.
- **crew_transfer_time** (*float*) – The number of hours it takes to transfer the crew from the equipment to the system, e.g. how long does it take to transfer the crew from the CTV to the turbine, default 0.
- **onsite** (*bool*) – Indicator for if the servicing equipment and crew are based onsite.

> **Note**
>
> If based onsite, be sure that the start and end dates represent the first and last day/month of the year, respectively, and the start and end years represent the fist and last year in the weather file.

- **method** (*str*) – Determines if the equipment will do all maximum severity repairs first or do all the repairs at one turbine before going to the next, by default severity. Must be one of "severity" or "turbine".
- **port_distance** (*int | float*) – The distance, in km, the equipment must travel to go between port and site, by default 0.
- **non_operational_start** (*str | datetime.datetime | None*) – The starting month and day, e.g., MM/DD, M/D, MM-DD, etc. for an annualized period of prohibited operations. When defined at the environment level, an undefined or later starting date will be overridden, by default None.
- **non_operational_end** (*str | datetime.datetime | None*) – The ending month and day, e.g., MM/DD, M/D, MM-DD, etc. for an annualized period of prohibited operations. When defined at the environment level, an undefined or earlier ending date will be overridden, by default None.
- **reduced_speed_start** (*str | datetime.datetime | None*) – The starting month and day, e.g., MM/DD, M/D, MM-DD, etc. for an annualized period of reduced speed operations. When defined at the environment level, an undefined or later starting date will be overridden, by default None.
- **reduced_speed_end** (*str | datetime.datetime | None*) – The ending month and day, e.g., MM/DD, M/D, MM-DD, etc. for an annualized period of reduced speed operations. When defined at the environment level, an undefined or earlier ending date will be overridden, by default None.
- **reduced_speed** (*float*) – The maximum operating speed during the annualized reduced speed operations. When defined at the environment level, an undefined or faster value will be overridden, by default 0.0.

Scheduled Servicing Equipment - Answer

library > vessels > ctv1.yaml > ...

```yaml
name: Crew Transfer Vessel 1
equipment_rate: 3500
capability: CTV
speed: 37.04  # 20 knots
strategy: scheduled
max_windspeed_transport: 99
max_windspeed_repair: 99
max_waveheight_transport: 2
max_waveheight_repair: 2
onsite: True
mobilization_cost: 0
mobilization_days: 0
port_distance: 30
# workday_start: 0
# workday_end: 24
start_month: 1
start_day: 1
end_month: 12
end_day: 31
start_year: 2002
end_year: 2021
crew_transfer_time: 0.25
n_crews: 1
crew:
  day_rate: 0  # crew comes with vessel
  n_day_rate: 0
  hourly_rate: 0
  n_hourly_rate: 0
# capex: 3000000
```

Unscheduled Servicing Equipment

- **name** (*str*) – Name of the piece of servicing equipment.
- **equipment_rate** (*float*) – Day rate for the equipment/vessel, in USD.
- **n_crews** (*int*) – Number of crew units for the equipment.
- **crew** (*ServiceCrew*) – The crew details, see `ServiceCrew` for more information. Dictionary of labor costs with the following: `n_day_rate`, `day_rate`, `n_hourly_rate`, and `hourly_rate`.
- **charter_days** (*int*) – The number of days the servicing equipment can be chartered for.
- **capability** (*str*) – The type of capabilities the equipment contains. Must be one of: - RMT: remote (no actual equipment BUT no special implementation) - DRN: drone - CTV: crew transfer vessel/vehicle - SCN: small crane (i.e., field support vessel) - LCN: large crane (i.e., heavy lift vessel) - CAB: cabling vessel/vehicle - DSV: diving support vessel - TOW: tugboat or towing equipment - AHV: anchor handling vessel (tugboat that doesn't trigger tow-to-port)
- **speed** (*float*) – Maximum transit speed, km/hr.
- **tow_speed** (*float*) – The maximum transit speed when towing, km/hr.

> **Note**
>
> This is only required for when the servicing equipment is tugboat enabled for a tow-to-port scenario (capability = "TOW")

- **speed_reduction_factor** (*flaot*) – Reduction factor for traveling in inclement weather, default 0. When 0, travel is stopped when either *max_windspeed_transport* or *max_waveheight_transport* is reached, and when 1, *speed* is used.
- **max_windspeed_transport** (*float*) – Maximum windspeed for safe transport, m/s.
- **max_windspeed_repair** (*float*) – Maximum windspeed for safe operations, m/s.
- **max_waveheight_transport** (*float*) – Maximum waveheight for safe transport, m, default 1000 (land-based).
- **max_waveheight_repair** (*float*) – Maximum waveheight for safe operations, m, default 1000 (land-based).
- **mobilization_cost** (*float*) – Cost to mobilize the rig and crew, default 0.
- **mobilization_days** (*int*) – Number of days it takes to mobilize the equipment, default 0.
- **strategy** (*str*) – For any unscheduled maintenance servicing equipment, this determines the strategy for dispatching. Should be on of "downtime" or "requests".
- **strategy_threshold** (*str*) – For downtime-based scenarios, this is based on the operating level, and should be in the range (0, 1). For reqest-based scenarios, this is the maximum number of requests that are allowed to build up for any given type of unscheduled servicing equipment, should be an integer >= 1.
- **workday_start** (*int*) – The starting hour of a workshift, in 24 hour time.
- **workday_end** (*int*) – The ending hour of a workshift, in 24 hour time.
- **crew_transfer_time** (*float*) – The number of hours it takes to transfer the crew from the equipment to the system, e.g. how long does it take to transfer the crew from the CTV to the turbine, default 0.
- **onsite** (*bool*) – Indicator for if the rig and crew are based onsite.

> **Note**
>
> if the rig and crew are onsite be sure that the start and end dates represent the first and last day/month of the year, respectively, and the start and end years represent the fist and last year in the weather file.

- **method** (*str*) – Determines if the ship will do all maximum severity repairs first or do all the repairs at one turbine before going to the next, by default severity. Should by one of "severity" or "turbine".
- **unmoor_hours** (*int | float*) – The number of hours required to unmoor a floating offshore wind turbine in order to tow it to port, by default 0.

> **Note**
>
> Required for the tugboat/towing capability, otherwise unused.

- **reconnection_hours** (*int | float*) – The number of hours required to reconnect a floating offshore wind turbine after being towed back to site, by default 0.

> **Note**
>
> Required for the tugboat/towing capability, otherwise unused.

- D4.2, p. 18
- Create a YAML configuration file in "<library>/vessels/", for an HLV and tugboat.

https://wisdem.github.io/WOMBAT/API/types.html#unscheduled-service-equipment

- **port_distance** (*int | float*) – The distance, in km, the equipment must travel to go between port and site, by default 0.
- **non_operational_start** (*str | datetime.datetime | None*) – The starting month and day, e.g., MM/DD, M/D, MM-DD, etc. for an annualized period of prohibited operations. When defined at the environment level or the port level, if a tugboat, an undefined or later starting date will be overridden, by default None.
- **non_operational_end** (*str | datetime.datetime | None*) – The ending month and day, e.g., MM/DD, M/D, MM-DD, etc. for an annualized period of prohibited operations. When defined at the environment level or the port level, if a tugboat, an undefined or earlier ending date will be overridden, by default None.
- **reduced_speed_start** (*str | datetime.datetime | None*) – The starting month and day, e.g., MM/DD, M/D, MM-DD, etc. for an annualized period of reduced speed operations. When defined at the environment level or the port level, if a tugboat, an undefined or later starting date will be overridden, by default None.
- **reduced_speed_end** (*str | datetime.datetime | None*) – The ending month and day, e.g., MM/DD, M/D, MM-DD, etc. for an annualized period of reduced speed operations. When defined at the environment level or the port level, if a tugboat, an undefined or earlier ending date will be overridden, by default None.
- **reduced_speed** (*float*) – The maximum operating speed during the annualized reduced speed operations. When defined at the environment level, an undefined or faster value will be overridden, by default 0.0.

Unscheduled Servicing Equipment - Answer

library > vessels > hlv1.yaml > {} crew

```yaml
name: Heavy Lift Vessel 1
equipment_rate: 290000
capability: LCN
speed: 20.37  # 11 knots
strategy: requests
strategy_threshold: 1
max_windspeed_transport: 10
max_windspeed_repair: 10
max_waveheight_transport: 2
max_waveheight_repair: 2
onsite: false
mobilization_cost: 325000
mobilization_days: 10
workday_start: 0
workday_end: 24
charter_days: 20
crew_transfer_time: 0.25
n_crews: 1
crew:
  day_rate: 0
  n_day_rate: 0  # crew comes with the vessel
  hourly_rate: 0
  n_hourly_rate: 0
```

library > vessels > tugboat1.yaml > {} crew

```yaml
name: Tugboat 1
equipment_rate: 30000
# mobilization_cost: 200000
capability: [TOW]
speed: 37.04  # 20 knots
strategy: requests
max_windspeed_transport: 99
max_windspeed_repair: 99
max_waveheight_transport: 2.5
max_waveheight_repair: 2.5
onsite: false
port_distance: 0  # automatically set from the port configuration
# workday_start: 6
# workday_end: 18
strategy_threshold: 1  # needs to be here, but doesn't do anything
charter_days: 3  # needs to be here, but doesn't do anything
tow_speed: 9.26  # 5 knots
unmoor_hours: 2
reconnection_hours: 4
crew_transfer_time: 0.25
n_crews: 1
crew:
  day_rate: 0  # crew comes with vessel
  n_day_rate: 0
  hourly_rate: 0
  n_hourly_rate: 0
```

Substations

- D4.2, p13-17
- Create a YAML file in <library>/substations/ using the file name you used in “subassembly” for your layout file
- Create the maintenance and failure configurations for the OSS

Substations

The substation model relies on two specific inputs, and one subassembly input (transformer).

capacity_kw

The capacity of all turbines in the windfarm, neglecting any losses. Only needed if a $/kw cost basis is being used.

capex_kw

The $/kw cost of the machine, if not providing absolute costs.

Additional keys can be added to represent subassemblies, such as a transformer, in the same format as the generator example above. Similarly, a user can define as man or as few of the subassemblies as desired with their preferred naming conventions

The following is an example of substation YAML definition with no modeled subasemblies.

```yaml
capacity_kw: 670000
capex_kw: 140
transformer:
  name: transformer
  maintenance:
    -
      description: n/a
      time: 0
      materials: 0
      service_equipment: CTV
      frequency: 0
  failures:
    1:
      scale: 0
      shape: 0
      time: 0
      materials: 0
      service_equipment: [CTV]
      operation_reduction: 0
      level: 1
      description: n/a
```

https://wisdem.github.io/WOMBAT/examples/how_to.html#substations

- **time** (*float*) – Amount of time required to perform maintenance, in hours.
- **materials** (*float*) – Cost of materials required to perform maintenance, in USD.
- **frequency** (*float*) – Optimal number of days between performing maintenance, in days.
- **service_equipment** (*list[str] | str*) – Any combination of th following `Equipment.capability` options.
  - RMT: remote (no actual equipment BUT no special implementation)
  - DRN: drone
  - CTV: crew transfer vessel/vehicle
  - SCN: small crane (i.e., field support vessel)
  - LCN: large crane (i.e., heavy lift vessel)
  - CAB: cabling vessel/vehicle
  - DSV: diving support vessel
  - TOW: tugboat or towing equipment
  - AHV: anchor handling vessel (tugboat that doesn't trigger tow-to-port)
- **system_value** (*Union[int, float]*) – Turbine replacement value. Used if the materials cost is a proportional cost.
- **description** (*str*) – A short text description to be used for logging.
- **operation_reduction** (*float*) – Performance reduction caused by the failure, between (0, 1]. Defaults to 0.

> **Warning**
>
> As of v0.7, availability is very sensitive to the usage of this parameter, and so it should be used carefully.

- **level** (*int, optional*) – Severity level of the maintenance. Defaults to 0.
- **scale** (*float*) – Weibull scale parameter for a failure classification.
- **shape** (*float*) – Weibull shape parameter for a failure classification.
- **time** (*float*) – Amount of time required to complete the repair, in hours.
- **materials** (*float*) – Cost of the materials required to complete the repair, in $USD.
- **operation_reduction** (*float*) – Performance reduction caused by the failure, between (0, 1].

> **Warning**
>
> As of v0.7, availability is very sensitive to the usage of this parameter, and so it should be used carefully.

- **level** (*int, optional*) – Level of severity, will be generated in the `ComponentData.create_severities` method.
- **service_equipment** (*list[str] | str*) – Any combination of the following `Equipment.capability` options:
  - RMT: remote (no actual equipment BUT no special implementation)
  - DRN: drone
  - CTV: crew transfer vessel/vehicle
  - SCN: small crane (i.e., field support vessel)
  - LCN: large crane (i.e., heavy lift vessel)
  - CAB: cabling vessel/vehicle
  - DSV: diving support vessel
  - TOW: tugboat or towing equipment
  - AHV: anchor handling vessel (tugboat that doesn't trigger tow-to-port)
- **system_value** (*Union[int, float]*) – Turbine replacement value. Used if the materials cost is a proportional cost.
- **replacement** (*bool*) – True if triggering the failure requires a subassembly replacement, False, if only a repair is necessary. Defaults to False
- **description** (*str*) – A short text description to be used for logging.
- **rng** (*np.random._generator.Generator*) –

Left: Maintenance
https://wisdem.github.io/WOMBAT/API/types.html#maintenance-tasks

Right: Failure
https://wisdem.github.io/WOMBAT/API/types.html#failures

Substations - Answer

library > substations > corewind_substation.yaml > { } transformer

```yaml
capacity_kw: 600
capex_kw: 120  # 120,000 EUR/MW; https://guidetoanoffshorewindfarm.com/wind-farm-costs
transformer:
  name: transformer
  maintenance:
    -
      description: oss annual inspection
      time: 24
      materials: 500
      service_equipment: CTV
      frequency: 365
      level: 1
    # n_technicians: 4
  failures:
    2:
      scale: 5
      shape: 1
      time: 12
      materials: 2000
      service_equipment: CTV
      operation_reduction: 0
      level: 2
      description: oss minor repair
      # n_technicians: 2
    4:
      scale: 100
      shape: 1
      time: 60
      materials: 100000
      service_equipment: CTV
      operation_reduction: 0
      level: 4
      description: oss major repair
      # n_technicians: 5

```

Turbines

- D4.2, p13-17
- Create a YAML file in <library>/turbines/ using the file name you used in “subassembly” for your layout file
- What subassemblies should we consider for the turbine model?
- Create the maintenance and failure configurations for the hydraulic system (ballast pump and pitch system)

Right:
https://wisdem.github.io/WOMBAT/examples/how_to.html#turbines

Substations

The substation model relies on two specific inputs, and one subassembly input (transformer).

capacity_kw

The capacity of all turbines in the windfarm, neglecting any losses. Only needed if a $/kw cost basis is being used.

capex_kw

The $/kw cost of the machine, if not providing absolute costs.

Additional keys can be added to represent subassemblies, such as a transformer, in the same format as the generator example above. Similarly, a user can define as man or as few of the subassemblies as desired with their preferred naming conventions

The following is an example of substation YAML definition with no modeled subasemblies.

```yaml
capacity_kw: 670000
capex_kw: 140
transformer:
  name: transformer
  maintenance:
    -
      description: n/a
      time: 0
      materials: 0
      service_equipment: CTV
      frequency: 0
  failures:
    1:
      scale: 0
      shape: 0
      time: 0
      materials: 0
      service_equipment: [CTV]
      operation_reduction: 0
      level: 1
      description: n/a
```

Turbines - Answer

library > turbines > corewind_15MW.yaml > {} drive_train > {} failures

```yaml
capacity_kw: 15000
capex_kw: 1300
power_curve:
  file: 2020ATB_NREL_Reference_15MW_240.csv
  bin_width: 1
electrical_system: ...
hydraulic_system: ...
yaw_system: ...
rotor_blades: ...
generator: ...
supporting_structure: ...
drive_train: ...
```

```yaml
hydraulic_system:
  name: hydraulic_system
  maintenance:
  - description: n/a
    time: 0
    materials: 0
    service_equipment: CTV
    frequency: 0
  failures:
    1:
      scale: 1.214
      shape: 1
      time: 18
      materials: 500
      service_equipment: CTV
      operation_reduction: 0
      level: 1
      description: minor pitch system repair
      # n_technicians: 2
    2:
      scale: 100
      shape: 1
      time: 8
      materials: 1000
      service_equipment: CTV
      operation_reduction: 0
      level: 2
      description: minor ballast pump repair
      # n_technicians: 2
    3:
      scale: 5.587
      shape: 1
      time: 38
      materials: 1900
      service_equipment: CTV
      operation_reduction: 0.0
      level: 3
      description: major pitch system repair
      # n_technicians: 3
    5:
      scale: 1000
      shape: 1
      time: 75
      materials: 14000
      service_equipment: LCN
      operation_reduction: 0
      replacement: true
      level: 5
      description: major pitch system replacement
      # n_technicians: 4
```

Cables

- D4.2, p13-17
- Create a YAML file in <library>/cables/ using the file name you used in “upstream_cable” for your layout file
- Create the maintenance and failure configurations for the export cables

Substations

The substation model relies on two specific inputs, and one subassembly input (transformer).

capacity_kw

The capacity of all turbines in the windfarm, neglecting any losses. Only needed if a $/kw cost basis is being used.

capex_kw

The $/kw cost of the machine, if not providing absolute costs.

Additional keys can be added to represent subassemblies, such as a transformer, in the same format as the generator example above. Similarly, a user can define as man or as few of the subassemblies as desired with their preferred naming conventions

The following is an example of substation YAML definition with no modeled subasemblies.

```yaml
capacity_kw: 670000
capex_kw: 140
transformer:
  name: transformer
  maintenance:
    -
      description: n/a
      time: 0
      materials: 0
      service_equipment: CTV
      frequency: 0
  failures:
    1:
      scale: 0
      shape: 0
      time: 0
      materials: 0
      service_equipment: [CTV]
      operation_reduction: 0
      level: 1
      description: n/a
```

Right: https://wisdem.github.io/WOMBAT/examples/how_to.html#cables

Cables - Answer

library > cables > corewind_export.yaml > {} failures

```yaml
name: export cable
maintenance:
  -
    description: export cable subsea inspection
    time: 12
    materials: 500
    service_equipment: DSV
    frequency: 730
    level: 1
    # n_technicians: 5
failures:
  6:
    scale: 50
    shape: 1
    time: 60
    materials: 30000
    operation_reduction: 0
    service_equipment: [CAB]
    replacement: false
    level: 4
    description: export cable major repair
    # n_technicians: 5
```

Hands-on With WOMBAT

Running a Simulation

- Final questions on setup before moving forward?
- From the terminal, launch a Jupyter Lab session: “jupyter lab”
- I’ll switch over the example notebook that’s setup to finish out the workshop

WAVES

Combining ORBIT, WOMBAT, and FLORIS
in a single model

WAVES

- Release coming soon at: https://github.com/NREL/WAVES
- Combine CapEx (ORBIT), OpEx (WOMBAT), and AEP (FLORIS) into a single model API that creates the required input and output connections to properly consider project financials and real-world energy production (availability and wakes)

Future of WOMBAT

What's on the horizon for WOMBAT over the next few years?

Potential Future Work

Model Development

- Model tow-to-port simulations at a higher resolution.
- Improved repair timing.
- Date-based maintenance.
- Single file/dictionary configuration similar to ORBIT and FLORIS.
- Multi-crew handoff.
- Other ideas? Add them to the Issues board: https://github.com/WISDEM/WOMBAT/issues

Validation and Review

- Floating Wind model comparison publication in 2024 to update the state of O&M modeling.
- Develop a baseline land-based, fixed offshore, and floating offshore wind data set for easier modeling.

Thank you

www.nrel.gov

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 the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Wind Energy Technologies Office. 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.