



# WEC-Sim Training Course



## Using MoorDyn with WEC-Sim

September 2023

*PRESENTED BY*

David Ogden



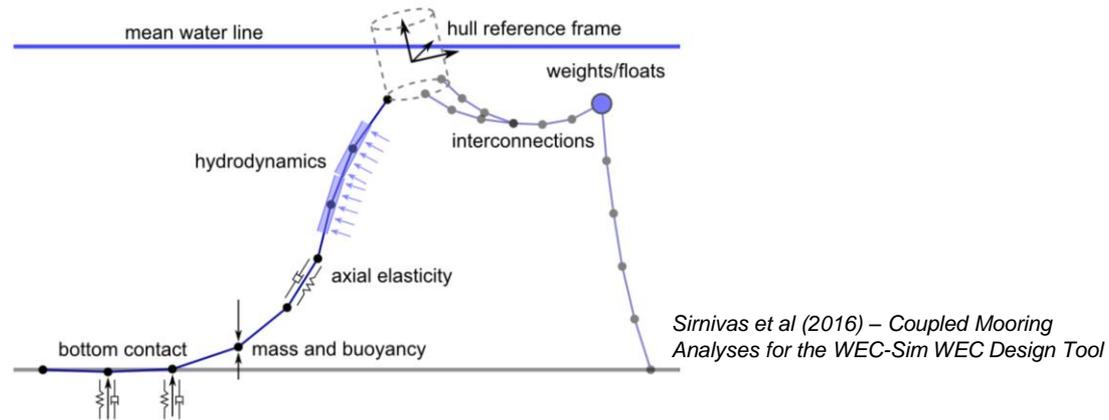
# Using MoorDyn with WEC-Sim

Contents:

- 1. What is MoorDyn?**
- 2. Download the necessary MoorDyn files**
- 3. Run the WEC-Sim RM3 example with MoorDyn**
- 4. Check the results**
- 5. Modify the MoorDyn input file**

# What is MoorDyn?

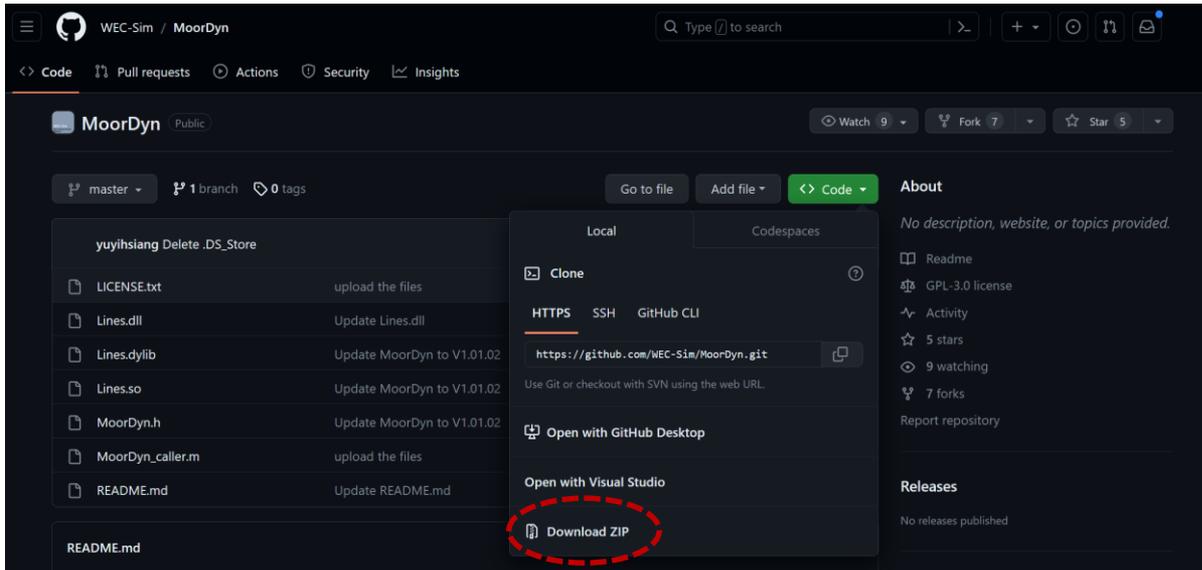
- MoorDyn is a lumped-mass mooring line model for simulating the **dynamics of moorings** connected to floating offshore structures.



- Dynamic mooring models are critical in order to obtain a more accurate estimate of a WEC's response and the mooring line loads.

# Download the necessary MoorDyn files

- Navigate to <https://github.com/WEC-Sim/MoorDyn> and download the repo:

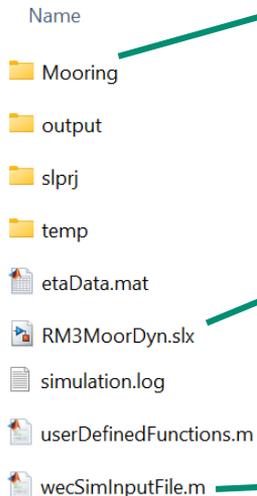


- Save the repo to the directory: `WEC-Sim/source/functions/moorDyn`

# WEC-Sim RM3 MoorDyn Example

- Now head to the MoorDyn example: *WEC-Sim\_Applications/Mooring/MoorDyn*

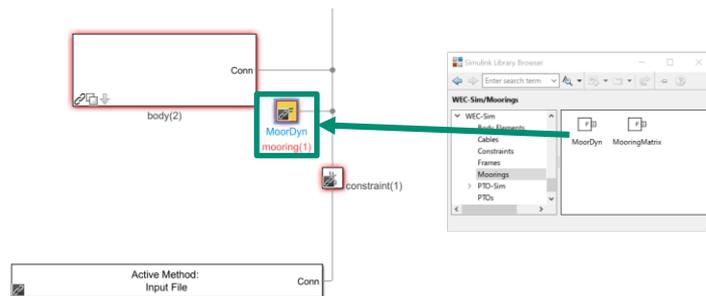
WEC-Sim\_Applications > Mooring > MoorDyn



lines.txt MoorDyn input file

```
1 MoorDyn line data file for MoorDyn in Lines.dll
2 ----- LINE DICTIONARY -----
3 LineType Diam MassDenInAir EA BA/-data Cam Cat Cdn Cdt
4 (-) (m) (kg/m) (N) (Pa/-) (-) (-) (-) (-)
5 chain 0.144 225.0 583.256E+05 -0.5 2.0 0.0 1.5 -0.0
6 ----- NODE PROPERTIES -----
7 Node Type X Y Z H V H V FY FZ CGA CA
8 (-) (-) (m) (m) (m) (kg) (m/s) (m/s) (kN) (kN) (m^2) (-)
9 1 Vessel -1.0 0 -10.00 0 0 0 0 0 0 0 0
10 2 Fix -257.0 0 -70.00 0 0 0 0 0 0 0 0
11 3 Vessel 1.5 2.598 -70.00 0 0 0 0 0 0 0 0
12 4 Fix 133.5 215.0 -70.00 0 0 0 0 0 0 0 0
13 5 Vessel -2.598 -10.00 0 0 0 0 0 0 0 0 0
14 6 Fix 133.5 -231.23 -70.00 0 0 0 0 0 0 0 0
15 7 Connect -40.0 0 -20.00 16755 33.510 0 0 0 12.566 1
16 8 Connect 20.0 34.642 -10.00 16755 33.510 0 0 0 12.566 1
17 9 Connect 20.0 -34.642 -10.00 16755 33.510 0 0 0 12.566 1
18 ----- LINE PROPERTIES -----
19 Line LineType UnitLen NumNodes NodeLen NodeFair Flags/Outputs
20 (-) (-) (m) (m) (m) (m) (m) (m)
21 1 chain 240.0 15 2 7 tp
22 2 chain 240.0 15 4 8 tp
23 3 chain 240.0 15 6 9 tp
24 4 chain 40.0 5 7 1 tp
25 5 chain 40.0 5 8 3 tp
26 6 chain 40.0 5 9 5 tp
27 ----- SOLVER OPTIONS -----
28 0.0005 ddt - time step to use in mooring integration
29 0 Hevelin - use kinematics flag (Hevelin, the only option currently supported)
30 3.0e5 kBot - bottom stiffness
31 3.0e5 cBot - bottom damping
32 70 WtrDepth - water depth
33 5.0 CGScaleIC - factor by which to scale drag coefficients during dynamic relaxation IC gen
34 0.001 DvsScaleIC - threshold for IC con
35 0 WtrUnits - option to ship units line in output files if zero
36 ----- OUTPUTS -----
37 FairLen FairLen FairLen
38 ----- read this line -----
```

Check MoorDyn block is within .slx file



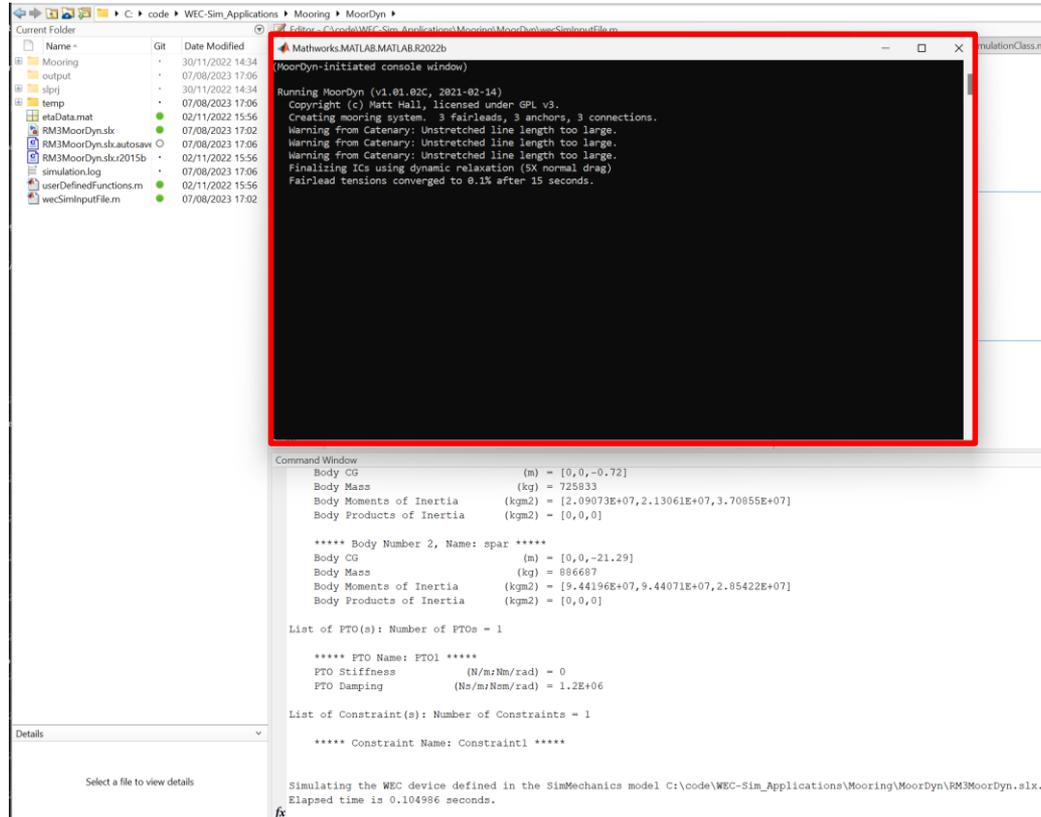
MoorDyn info is specified in wecSimInputFile.m

```
41 % MoorDyn
42 % MoorDyn
43 mooring(1) = mooringClass('mooring'); % Initialize mooringClass
44 mooring(1).moorDynLines = 6; % Specify number of lines
45 mooring(1).moorDynNodes(1:3) = 16; % Specify number of nodes per line
46 mooring(1).moorDynNodes(4:6) = 6; % Specify number of nodes per line
47 mooring(1).initialDisplacement = [0 0 -0.21]; % Initial Displacement
```

**Note**  
You may need to install the MinGW-w64 compiler to run this simulation.

# Running the WEC-Sim RM3 MoorDyn Example

- When you run the example, you should see a MoorDyn-initiated console window pop up:



The screenshot shows a MATLAB console window titled "Mathworks.MATLAB.MATLAB.R2022b" with a sub-window "MoorDyn-initiated console window". The console output displays the following text:

```
Running MoorDyn (v1.01.02c, 2021-02-14)
Copyright (c) Matt Hall, licensed under GPL v3.
Creating mooring system, 3 fairleads, 3 anchors, 3 connections.
Warning from Catenary: Unstretched line length too large.
Warning from Catenary: Unstretched line length too large.
Warning from Catenary: Unstretched line length too large.
Finalising ICS using dynamic relaxation (SV normal drag)
Fairlead tensions converged to 0.1% after 15 seconds.
```

Below the console window, the Command Window displays simulation parameters:

```
Command Window
Body CG (m) = [0,0,-0.72]
Body Mass (kg) = 725833
Body Moments of Inertia (kgm2) = [2.09073E+07,2.13061E+07,3.70855E+07]
Body Products of Inertia (kgm2) = [0,0,0]

***** Body Number 2, Name: spar *****
Body CG (m) = [0,0,-21.29]
Body Mass (kg) = 886697
Body Moments of Inertia (kgm2) = [9.44196E+07,9.44071E+07,2.85422E+07]
Body Products of Inertia (kgm2) = [0,0,0]

List of PTO(s): Number of PTOs = 1

***** PTO Name: PTO1 *****
PTO Stiffness (N/m/Nm/rad) = 0
PTO Damping (Ns/m/Nm/rad) = 1.2E+06

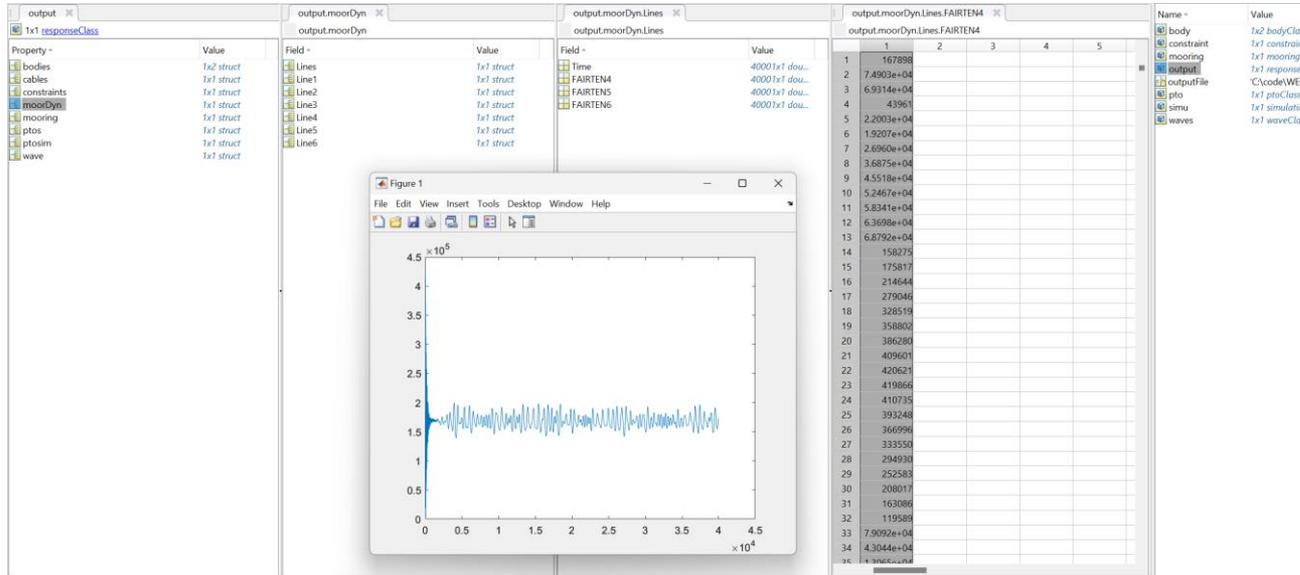
List of Constraint(s): Number of Constraints = 1

***** Constraint Name: Constraint1 *****

Simulating the WEC device defined in the SimMechanics model C:\code\WEC-Sim_Applications\Mooring\MoorDyn\RM3MoorDyn.slx...
Elapsed time is 0.104986 seconds.
```

# Check results of the WEC-Sim RM3 MoorDyn Example

- The MoorDyn data should automatically appear in the workspace, within the 'output' object:



- Modify this part of the MoorDyn input file in order to change which data is saved:

```
36 ----- OUTPUTS -----  
37 FairTen4 FairTen5 FairTen6
```

# Developing your own MoorDyn models

For a comprehensive explanation of the different sections of the MoorDyn input file, please see the MoorDyn documentation:

<https://moordyn.readthedocs.io>

Line types

Nodes

Line properties

Solver options

```
1 Mooring line data file for MoorDyn in Lines.d11
2 ----- LINE DICTIONARY -----
3 LineType Diam MassDenInAir EA BA/-zeta Can Cat Cdn Cdt
4 (-) (m) (kg/m) (N) (Pa-s/-) (-) (-) (-) (-)
5 chain 0.144 126.0 583.375E6 -0.8 1.0 0.0 1.6 0.05
6 ----- NODE PROPERTIES -----
7 Node Type X Y Z R V FX FY FZ CDR CA
8 (-) (-) (m) (m) (m) (kg) (m^3) (kN) (kN) (kN) (m^2) (-)
9 1 Vessel -3.0 0 -10.00 0 0 0 0 0 0 0
10 2 Fix -267.0 0 -70.00 0 0 0 0 0 0 0
11 3 Vessel 1.5 2.598 -10.00 0 0 0 0 0 0 0
12 4 Fix 133.5 231.23 -70.00 0 0 0 0 0 0 0
13 5 Vessel 1.5 -2.598 -10.00 0 0 0 0 0 0 0
14 6 Fix 133.5 -231.23 -70.00 0 0 0 0 0 0 0
15 7 Connect -40.0 0 -10.00 16755 33.510 0 0 0 12.566 1
16 8 Connect 20.0 34.642 -10.00 16755 33.510 0 0 0 12.566 1
17 9 Connect 20.0 -34.642 -10.00 16755 33.510 0 0 0 12.566 1
18 ----- LINE PROPERTIES -----
19 Line LineType UnstLen NumSegs NodeAnch NodeFair Flags/Outputs
20 (-) (-) (m) (-) (-) (-) (-)
21 1 chain 240.0 15 7 7 tp
22 2 chain 240.0 15 4 8 tp
23 3 chain 240.0 15 6 9 tp
24 4 chain 48.0 5 7 1 tp
25 5 chain 48.0 5 8 3 tp
26 6 chain 48.0 5 9 5 tp
27 ----- SOLVER OPTIONS -----
28 0.0005 dtm - time step to use in mooring integration
29 0 WaveKin - wave kinematics flag (@neglect, the only option currently supported)
30 3.0e6 kBot - bottom stiffness
31 3.0e5 cBot - bottom damping
32 70 WtrDpth - water depth
33 5.0 CdScaleIC - factor by which to scale drag coefficients during dynamic relaxation IC gen
34 0.001 threshIC - threshold for IC con
35 0 WriteUnits - option to skip units line in output files if zero
36 ----- OUTPUTS -----
37 FairTen5 FairTen5 FairTen5
38 ----- need this line -----
```

In your WEC-Sim input file, ensure that the number of lines and nodes (segments+1) matches the MoorDyn lines.txt input file:

```
41 %% Mooring
42 % Moordyn
43 mooring(1) = mooringClass('mooring'); % Initialize mooringClass
44 mooring(1).moorDynLines = 6; % Specify number of lines
45 mooring(1).moorDynNodes(1:3) = 16; % Specify number of nodes per line
46 mooring(1).moorDynNodes(4:6) = 6; % Specify number of nodes per line
47 mooring(1).initial.displacement = [0 0 -0.21]; % Initial Displacement
```

# Thank you



For more information please visit the WEC-Sim website:

<http://wec-sim.github.io/WEC-Sim>

If you have questions on this presentation please reach out to any of the WEC-Sim Developers on GitHub:

<https://github.com/WEC-Sim/WEC-Sim>

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

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 the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Water Power 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.

