



## WECCOMP Numerical Model Description

A WEC-Sim model of the WaveStar device is available on the [WECCOMP GitHub repository](#). This model should be used to develop the WECCOMP control algorithm.

### WEC-Sim code

WEC-Sim (Wave Energy Converter SIMulator) is a publicly available, open-source wave energy converter simulation code jointly developed by the [National Renewable Energy Laboratory \(NREL\)](#) and [Sandia National Laboratories \(Sandia\)](#), funded by the U.S. Department of Energy's Water Power Technologies Office. WEC-Sim can model devices that are comprised of rigid bodies, power-take-off systems, and mooring systems. Simulations are performed in the time-domain by solving the governing WEC equations of motion for each body and system constraints.

The WEC-Sim code is publicly available on the [WEC-Sim GitHub repository](#) and requires the following MATLAB toolboxes: Simulink, Simscape, and Simscape Multibody. For WECCOMP, use WEC-Sim v3.0 (or newer) and MATLAB 2017a (or newer and please use [Option 1 Clone with GitHub](#) for the WEC-Sim download). For more information about WEC-Sim, refer to the [WEC-Sim website](#).

### MathWorks licenses

The WECCOMP organizers have negotiated 25 temporary free licenses for competitors to use in developing and simulating their controllers. Licenses will be distributed upon registration to WECCOMP, on a first-come, first-served basis. A maximum of one license will be available to each group/institutional entry. To download MATLAB and associated toolboxes, each participant will need to register and create a MathWorks account; a link to the registration page can be found [here](#). Once participants have registered, send an email notifying the organizers who in return will provide the MATLAB download link. Please follow the download prompts and if there are any issues with the download notify the organizers.

### Online forums

Two forums are available to field queries and responses. The [WECCOMP forum](#) will deal with issues related to the competition, while the [WEC-Sim forum](#) is for all queries related to WEC-Sim. The WEC-Sim forum is in the public domain to benefit the WEC-Sim user community.

## Running the WECCOMP model

Once the [WEC-Sim code](#) and the [WECCOMP repository](#) have been cloned locally, and WEC-Sim has been added to [the MATLAB path](#), the WECCOMP model is run by navigating to the WECCOMP model's local path and typing `wecSim` into the MATLAB Command Window. For more information on how to set up and run WEC-Sim, refer to the [WEC-Sim website](#).

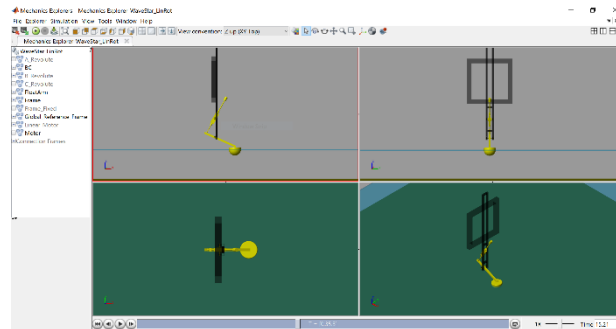


Figure 1. WECCOMP model visualization in WEC-Sim Simscape Mechanics Explorer

## Controller development

The [WECCOMP model](#) consists of a WEC-Sim input file ([wecSimInputFile.m](#)), a Simulink model file ([WaveStar.slx](#), see LHS of Figure 2), a controller initialization script (`controller_init.m`), and a post-processing script (`userDefinedFunctions.m`). The WECCOMP controller can either be developed using inputs of the linear force and displacement of the motor, or the rotary torque and displacement of the float. The controller linear/rotary implementation may be changed by selecting the appropriate variant subsystem, as shown in Figure 2. The WECCOMP controller should be developed in the *WaveStar->Linear/Rotary->WECCOMP Controller* subsystem in the *WaveStar.slx* model.

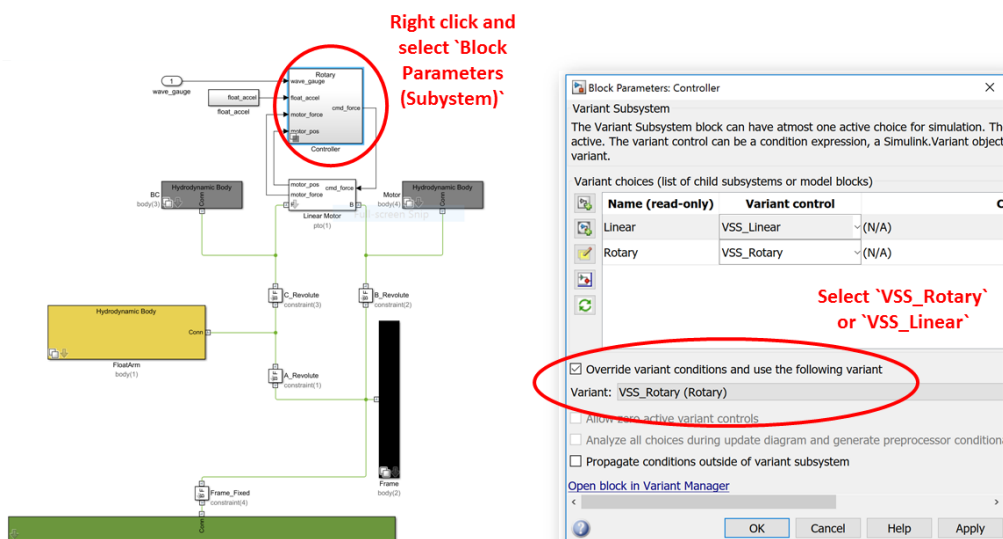


Figure 2. (LHS) WEC-Sim Simulink model (RHS) how to change the 'Controller' variant subsystem from rotary to linear

Additionally, contestants will need to modify the `wecSimInputFile.m` and `controller_init.m` scripts to run different wave cases, and initialize their controller. For more information about the WEC-Sim model files and input file options, refer to the [WEC-Sim website](#).

The example depicted in Figure 3 shows the implementation of a spring/damper/mass control system. The controller generates the control signal “cmd\_torque\_rot” by adding a signal proportional to the position, a signal proportional to the velocity (which is obtained as the low pass filtered derivative of the position), and another signal proportional to the acceleration.

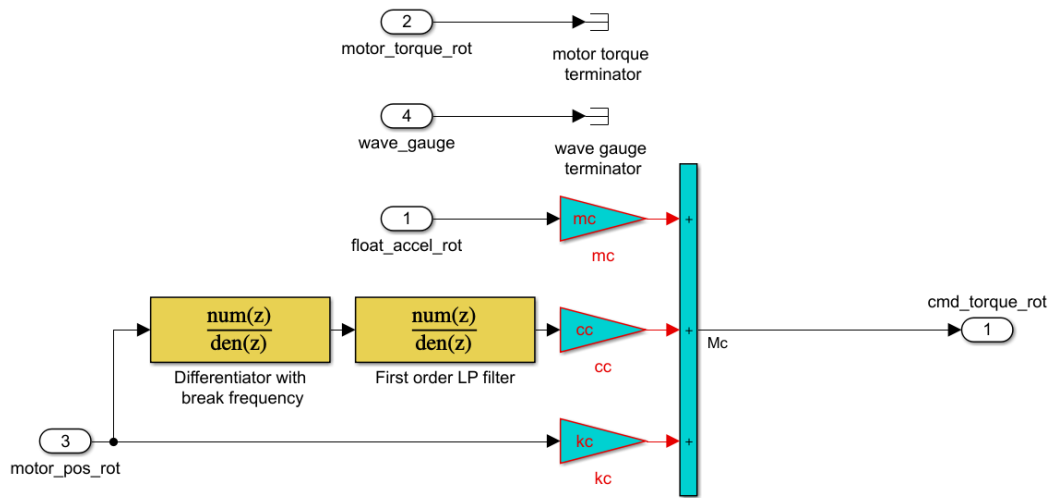


Figure 3: Implementation of a spring / damper control system

## Running WECCOMP sea states

WEC-Sim [multiple condition runs](#) were developed to run the WECCOMP sea states, allowing contestants to perform batch runs by typing `wecSimMCR` into the MATLAB Command Window. The WEC-Sim multiple condition runs are defined in `WECCOMP_ss.mat` and post-processing scripts are in `userDefinedFunctionsMCR.m`. Each of the WEC-Sim multiple condition runs has a ramp time of  $5 \cdot T_p$ , and a simulation end of time  $100 \cdot T_p$ . The first 25 seconds of WEC-Sim simulation will be discarded for calculation of the competition metrics (the evaluation criteria will only be calculated for time  $> 25s$ ).

## Calculating power and performance

For the WECCOMP control competition, the controller’s performance will be calculated according to the performance metrics defined in the **WECCOMP info sheet**. The power performance will be calculated discarding the first 25 seconds of WEC-Sim simulation for calculation of the competition metrics. This has already been implemented in the post-processing script (`userDefinedFunctions.m`).

## Organisers



AALBORG UNIVERSITY  
DENMARK