Load Mitigating Feedback Control of WECs

D Forbush

April 6, 2021

1 Dependencies

Please ensure you have an up-to-date clone of WEC-Sim before attempting this applications case, as it relies upon some recent code modifications.
In addition to the standard WEC-Sim toolboxes, this applications case requires:

1. Signal Processing Toolbox
2. DSP Toolbox

Add the sub-folders "CalcImpedance" and "ControlTests" to your path.

2 Overview

This WEC-Sim Applications case contains two main folders. "CalcImpedance" contains a worked example of a linear system identification procedure for a three degree of freedom (DOF) point absorber WEC device, modeled after the WaveBot [6]. The system identification procedure is detailed in [10], and has been applied in [9], [2], [7], [5], [4], [8] and is briefly summarized in subsequent sections.

The second folder, "ControlTests", uses this identified system model to develop an adaptive linear controller that approximates the performance of an optimal complex conjugate controller. This linear controller self-tunes to changing sea-states, with the optimal controller parameters determined by a multi-objective optimization specified by user-defined weights and/or objective thresholds.

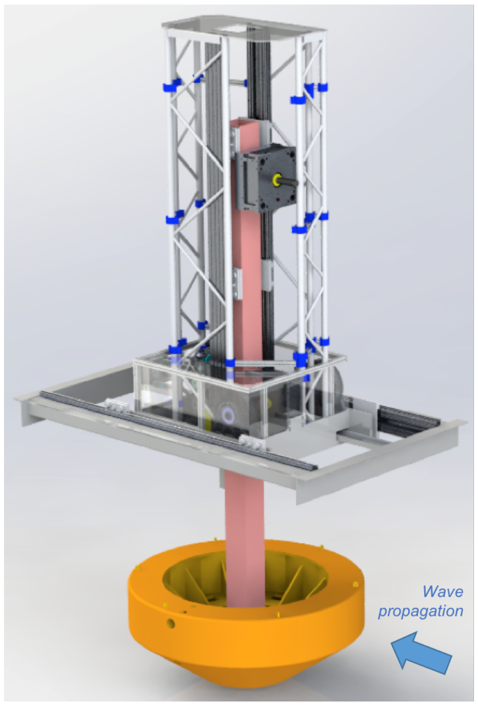

Figure 1: A rendering of the WaveBot device.

The additional folder "hydroData" contains the *.h5 file of boundary element method (BEM) data relevant for this device. Folder "geometry" contains an *.stl model, a simplification of the WaveBot device, used only for visualization.

3 Device Model

The modeled device is based on the WaveBot, a 1:9 scale 3-DOF single-body point absorber free in surge, heave, and pitch (Figure 1). During physical testing, the device was secured from above. This attachment is considered as a mooring matrix in the WEC-Sim model.

The device has three independent power-take-offs (PTOs) on each degree of freedom. The default values specified in the WEC-Sim input files of each main folder are representative of the parameters of the physical device examined in wave tank tests, and will produce a realistic response when tested in the suggested sea states (Table 1).

Table 1: Suggested sea-states

H (m)	T (s)	**Spectrum Type**
0.127	1.58	Pierson-Moskowitz
0.127	3.50	Pierson-Moskowitz
0.254	1.58	Pierson-Moskowitz
0.254	3.50	Pierson-Moskowitz

4 Identifying the Impedance Model

4.1 Background

The system identification procedure is briefly summarized. Intrinsic impedance Z_i defines the input/output relationship in the frequency domain (ω) between device velocity v (input) to actuator force F (output) for each degree of freedom, and is thus for this 3 DOF device a 3 x 3 matrix at each frequency.

$$Z_i(\omega) = F(\omega)/v(\omega), \tag{1}$$

Knowing hydrodynamic coefficients from BEM data and some device parameters, impedance can be estimated directly as

$$Z_{i,BEM}(\omega) = i\omega(M + m(\omega)) + B_v + R(\omega) + \frac{S}{i\omega}. \tag{2}$$

However, it is unlikely that BEM-estimated hydrodynamic coefficients will be perfect estimates of as-built device dynamics, so there is some benefit in identifying an impedance model directly from test data. In this context, the WEC-Sim model acts as a "numerical experiment": the exact same test procedure could be applied to a physical device to develop an impedance model of the system.

Each actuator is driven with a periodic uncorrelated band-limited white-noise signal ("multisine"), which spans the frequency range of interest. This signal is scaled up as needed in each DOF to bring about measurable motion (this scaling precaution is clearly more of a concern for a physical system, wherein static friction or instrument noise floors would confound the identification procedure without it, but is included here to illustrate best practice). Because the resulting matrix is 3 x 3, a minimum of 3 independent tests are needed to solve the system (Equation 1) at each frequency. This is achieved by cycling the uncorrelated signals among degrees of freedom for a minimum of 3 iterations. If more than the minimum number of tests are performed then the system becomes overdefined, and the solution of Equation 1 is achieved via pseudo-inverse, which results in a least-squares determination intrinsic impedance. This can lead to a more robust estimate, and decreases the need for each driving signal to be precisely uncorrelated with each other.

Of course, WEC-Sim remains a simulation, so the user must specify any additional parameters for the "as-built" device for them to be captured by this example of the system identification workflow. However, the procedure does not use knowledge of the specified values: the developed models use only measurements of the actuation force and the device velocity, and so the simulation is in fact a strong analog to a test of a physical device.

4.2 Running the Example

From the ./CalcImpedance directory, type "wecSimMCR" into the command window. This will execute several simulations in sequence, defined by the WaveBot3DOF Simulink model.

As mentioned, a minimum of three independent tests are needed to define the impedance of this three DOF system. Six such tests are carried out in sequence, with a unique combination of multisines (denoted 'multisine3DOFA' through 'multisine3DOFF') loaded at each iteration of wecSim as specified in the 'mcrImpedanceRuns'. At the end of each wecSim run, the velocity and actuator force data are saved to the output log of 'mcrOut' by "userDefinedFunctions". At the end of the sixth run, 'mcrOut' is saved, and "Identify_Impedance" runs.

This script takes the Fourier transform of the time-domain simulation output data and then solves Equation 1 at each frequency for Z_i. For comparison purposes, this script also calculates the BEM estimate of $Z_{i,BEM}$ via Equation 2, and plots it alongside Z_i. Throughout these applications cases, we assume that heave is decoupled from surge and pitch, this is true of the physical device being modeled. However, solving Z_i in this way will likely result in non-zero off-diagonal components implying a coupling between heave and other DOFs: these are neglected.

By default, no additional dynamics are specified in the WEC-Sim input file, and so one expects the identified model to align closely with the BEM estimate. By uncommenting the lines in wecSimInputFile.m in the Body Data section, additional linear damping and quadratic viscous drag is introduced. By uncommenting the lines in the Mooring Data section, additional mooring stiffness and damping is added to the heave and pitch DOFs. It should be noted that by default a surge mooring stiffness is introduced for all cases. This ensures the device maintains a mean-zero surge position, and is not intended to be a parameter specific to the as-built device model. Thus, this additional surge stiffness is also included in the $Z_{i,BEM}$ calculation.

This example is intended to provide a confined sandbox for users to experiment with the effect of different parameters on the identified impedance model. While the example will probably run, it is not likely to deliver meaningful impedance estimations for extreme values.

4.3 Code Summary

1. **wecSimInputFile.m**: The input file for the WEC-Sim model. The user can specify additional terms here, like viscous drag, additional mooring forces, etc. that will be included in the identified system model.

2. **userDefinedFunctions.m**: This executes at the end of each of the (by default) 6 WEC-Sim runs, and logs the velocity and force data by that book. After 6 runs, this function saves the logs, and calls "Identify_Impedance".

3. **userDefinedFunctionsMCR.m**: This is an intentionally blank function. If not included, MATLAB will attempt to find another with the same name along its path, which can cause unexpected behavior or errors, so this was included to avoid potential path issues.

4. **Identify_Impedance.m**: This loads the log files saved by "userDefinedFunctions.m" and calculates and plots the impedance models according to Equations 1 and 2. This uses helper functions "ampSpectraForZcalc" to perform the Fourier transform.

5. **ampSpectraForZcalc.m**: This calculates the Fourier transform of the velocity and force time series along the appropriate dimensions, and outputs the frequency vector (in Hz) and the frequency-domain data.

6. **WaveBot3Dof.slx**: This is the Simulink model used for simulation of the 3-DOF device. Note that the floating 3-DOF constraint block has been modifed from the library reference to accept actuation inputs.

5 Developing the Controller

5.1 Background

Here we offer only a brief background concerning the formulation of the multi-objective cost function. Details and supporting theory on the workings of the full controller can be found [3]. The control law under consideration is a linear feedback controller acting on velocity

$$F(t)_{PTO} = C(t)V(t) \tag{3}$$

where $V(t)$ is the 3 x 1 vector of velocity in surge (m/s), heave (m/s), and pitch (rad/s), F_{PTO} is the force applied to the actuator in surge (N), heave (N), and pitch (N-m), and C is a 3 x 3 control matrix.

The controller explored in this example is a diagonal proportional-integral type controller

$$C = \begin{bmatrix} K_p^s + \dfrac{K_i^s}{s} & 0 & 0 \\ 0 & K_p^h + \dfrac{K_i^h}{s} & 0 \\ 0 & 0 & K_p^p + \dfrac{K_i^p}{s} \end{bmatrix}. \tag{4}$$

where there is a proportional gain (K_p) and an integral gain (K_I) for each DOF, where the exponent letters denote the relevant degree of freedom: s the exponent indicating the surge degree of freedom should not be confused with s in a denominator of the integral gain, indicating the Laplace transform variable. To attain a particular controls objective, there are thus 6 parameters over which we can optimize.

There are three constituents of the cost function under consideration. The first is electrical power capture, **which by sign convention, is negative**:

$$P_{abs} = \mathcal{R}((NK_t)^{-1}C\Omega)^*((K_eN + R(NK_t)^{-1}C)\Omega) \tag{5}$$

summed over all frequencies, where R is the phase-to-neutral resistance of the motor (Ohm), K_t and K_e are the motor torque and back-EMF constants, N is the gear ratio relating DOF motion to that DOF motor angular velocity, and Ω is the closed-loop model of device velocity as calculated from the impedance model Z as

$$\Omega = (Z_i - C)^{-1} F_e$$

.

The second is force applied by the actuator to the power-take-off (Equation 3). The third is the fatigue damage absorbed by the actuator, using the spectral formulation presented by [11].

$$D_f = \frac{v_0^+ T}{K} (2\sqrt{2}\sigma_X)^m \Gamma(\frac{m}{2} + 1) \tag{6}$$

where the total force F_{TOT} is the sum of the wave-exerted excitation loads and the force applied by the actuator. Parameters v_0 and σ_X are the mean zero-upcrossing rate and the standard deviation of the F_{TOT} spectra, while m, K, and T are user-defined material-and-geometry specific parameters describing the relationship between the F_{TOT} spectra and the accumulated fatigue damage, D_f. For this example, m, K, and T were arbitrarily selected.

The controller optimization problem thus becomes a minimization of these three constituent terms, a cost function formulated

$$\eta_{opt} = \arg\min(W_1 P_{abs} + W_2 \sum |F_{PTO}| + W_3 D_f). \tag{7}$$

where W is a 3 x 3 matrix describing the relative importance of the objectives (columns of the matrix) for each degree of freedom (rows of the matrix). Thus, W_1, W_2 and W_3 in Equation 7 are the first, second, and third column of the matrix W, respectively. Note that because D_f, $\sum F_{PTO}$, and P_{abs} have not been normalized, scaling these terms to a common order of magnitude is also performed through the selection of the elements of W. The output, η_{opt} is a vector containing the optimal controller gains.

In an evolving sea, objectives may change over time. For example, mitigating loads is not likely to be a concern in small seas. To accommodate this, W can be adjusted based upon the current $\sum F_{PTO}$ and D_f and user-defined thresholds. Here, this adjustment is performed via

$$W_{j,k} = \{\Phi_{j,k} > p_{j,k}\}\alpha_{j,k}(\Phi_{j,k} - p_{j,k}) + \beta_{j,k} \int_{t_0,j,k}^{t} (\Phi_{j,k} - p_{j,k}) d\tau \tag{8}$$

where t_0 is the first time a threshold is exceeded for a particular objective, $\Phi_{j,k}$ is the $\sum F_{PTO}$ and D_f for each DOF (surge, heave, pitch, indicated by j), $p_{j,k}$ is the user-defined threshold for $\sum F_{PTO}$ ($k = 2$) and D_f ($k = 3$) for each DOF above which the controller tuning will adjust to attempt to reduce the excessive load. The rate of adjustment is controlled by $\alpha_{j,k}$, proportional to the current excess, and $\beta_{j,k}$, proportional to the integral of the excess over time. Because no element of W can be less than zero, the anti-windup scheme suggested by [1] is included on the integrator for saturation events.

5.2 Running the Example

This example is found in ./ControlTests. It needs to load an impedance model Z in order to inform the optimizer, and is intended to be run after the model is identified in the previous example, to demonstrate a controller for that particular combination of user-specified WEC parameters. However, 'Z_included.mat' is an identified model of the default parameters in the wecSimInputFile.m, at the frequencies (Hz) defined in 'f_vec.mat'.

This example can run stand-alone cases with command "wecSim" or can study a range of static weight parameter combinations using "wecSimMCR". The wecSimInputFile is lengthy, as it contains all necessary control parameters. In the Controller section, we begin by specifying the parameters of the frequency estimation that uses Z and measured velocity and PTO actuation force to estimate the wave excitation spectra. This calls helper function "interp_impedance" to interpolate the impedance model to the frequency vector utilized in simulation. The parameters in the Frequency Estimation subsection are used by blocks in WaveBot3Dof/Subsystem/Autotuner/Spectral Estimation.

The Control Parameters subsection defines W (Equation 7). Four control objectives are used described in Table 2. If 'WeightAdapt = 0', the static gains are prescribed based upon the selected control objective. If not using the default impedance model and device parameters, the user may need to adjust these weights to achieve stable and desirable responses.

If WeightAdapt = 1, then the weights associated with load mitigation are initially zero, and they are adapted via Equation 8 according to thresholds and parameters specified in this section. Again, if not using the default impedance model and device parameters, these may need to be adjusted for stable and desirable response. Control parameters are used in the gainOptimizer and weightTransition functions of WaveBot3Dof_Control/Subsystem/Autotuner.

Table 2: Controller Objectives

Obj. Number	**Load Mitigated**
1	None, maximize power capture only
2	F_{PTO}
3	D_f
4	F_{PTO} and D_f

With regard to selection of weights, an example file “mcrWeights.mat” is included. Running “wecSimMCR” will run each of a series of specified static gains, but ensure that ‘objCase = 4’ and ‘WeightAdapt = 0’ for this to run properly.

The PTO Parameters subsection simply specifies the parameters used by Equations 5 and 6.

The remainder of the input file should be identical to the input file used in ./CalcImpedance to ensure that the simulated WEC is the same for which the impedance model was identified.

5.3 Code Summary

1. **wecSimInputFile.m**: The input file for the WEC-Sim model, wherein controller parameters are specified. Additional terms here, like viscous drag, additional mooring forces, etc. are intended to match those specified in the CalcImpedance example, so that the identified model represents the system examined here.

2. **userDefinedFunctions.m**: This executes at the end of each of the (by default) 51 WEC-Sim MCR runs, and logs the F_{PTO}, F_{TOT}, and controller gains. After all runs, this function saves the logs.

3. **userDefinedFunctionsMCR.m**: This is an intentionally blank function. If not included, MATLAB will attempt to find another with the same name along its path, which can cause unexpected behavior or errors, so this was included to avoid potential path issues.

4. **interp_Impedance.m**: This helper function interpolates the identified impedance model to the frequencies used by the spectral estimator in the Simulink model.

5. **WaveBot3Dof_Control.slx**: This is the Simulink model used for the demonstration of the control of the 3-DOF device. Note that the floating 3-DOF constraint block has been modifed from the library reference to accept actuation inputs.

6 Acknowledgements

The development of this example was funded by the Department of Energy Water Power Technologies Office as part of the Seedling Program.

References

[1] K.J. Aström and R.M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers.* Princeton University Press, 2010.

[2] Giorgio Bacelli, Ryan G. Coe, David Patterson, and David Wilson. System identification of a heaving point absorber: Design of experiment and device modeling. *Energies*, 10(10):472, 2017.

[3] Giorgio Bacelli, Victor Nevarez, Ryan G. Coe, and David Wilson. Feedback resonating control for a wave energy converter. *IEEE Industrial Automation and Control*, 2019.

[4] Ryan G. Coe, Giorgio Bacelli, and Dominic Forbush. A practical approach to wave energy modeling and control.

[5] Ryan G. Coe, Giorgio Bacelli, Dominic Forbush, Steven J. Spencer, Kevin Dullea, Bret Bosma, and Pedro Lomonaco. Foswec dynamics and controls test report. Technical Report SAND2020-XXXX, Sandia National Laboratories, 2020.

[6] Ryan G. Coe, Giorgio Bacelli, Steven J. Spencer, and Hancheol Cho. Initial results from wave tank test of closed-loop WEC control, 2018.

[7] Ryan G. Coe, Giorgio Bacelli, Steven J. Spencer, Dominic Forbush, and Kevin Dullea. Advanced WEC dynamics and controls MASK3 test. Technical Report SAND2019-15428, Sandia National Laboratories, 2019.

[8] Dominic Forbush, Giorgio Bacelli, Ryan G. Coe, Steven J. Spencer, Pedro Lomonaco, and Bret Bosma. Design and testing of a free floating dual flap wave energy converter.

[9] Dominic D. Forbush, Giorgio Bacelli, Steven J. Spencer, and Ryan G. Coe. A self-tuning wec controller for changing sea states (in press). In *Proceedings of the 21st IFAC Wolrd Congress*, 2020.

[10] Lennart Ljung. *System identification - Theory for the User*. Prentice-Hall, 1999.

[11] Andrew S Zurkinden, Søren Heide Lambertsen, Lars Damkilde, Zhen Gao, and Torgeir Moan. Fatigue analysis of a wave energy converter taking into account different control strategies. In *ASME 2013 32nd International Conference on Ocean, Offshore and Arctic Engineering*, pages V008T09A057–V008T09A057. Citeseer, 2013.