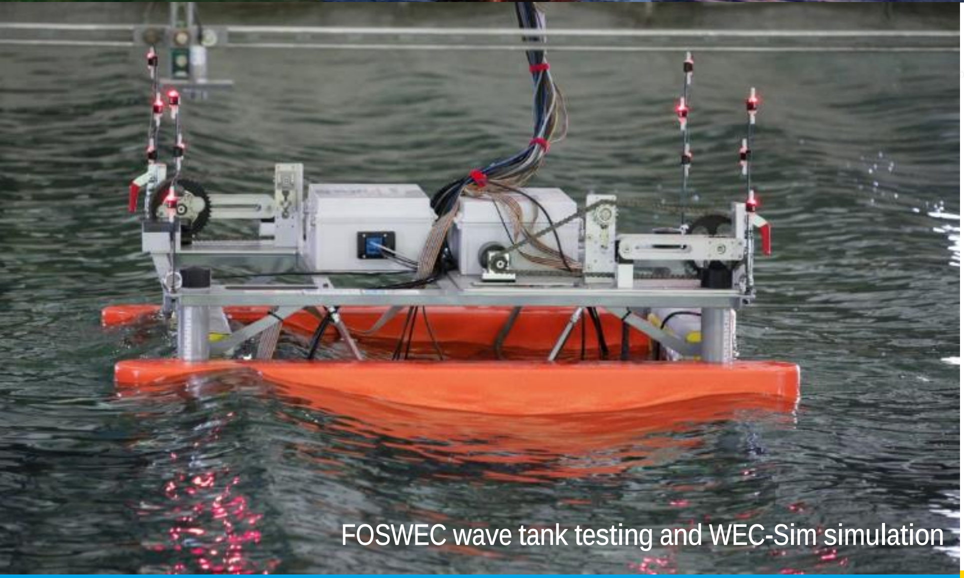
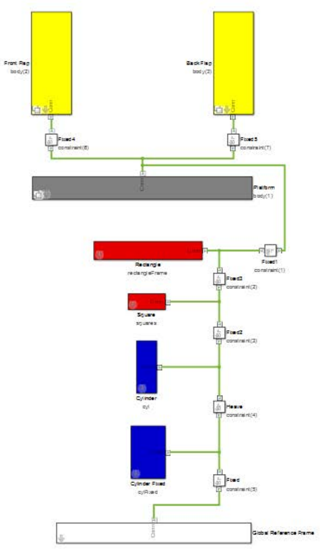
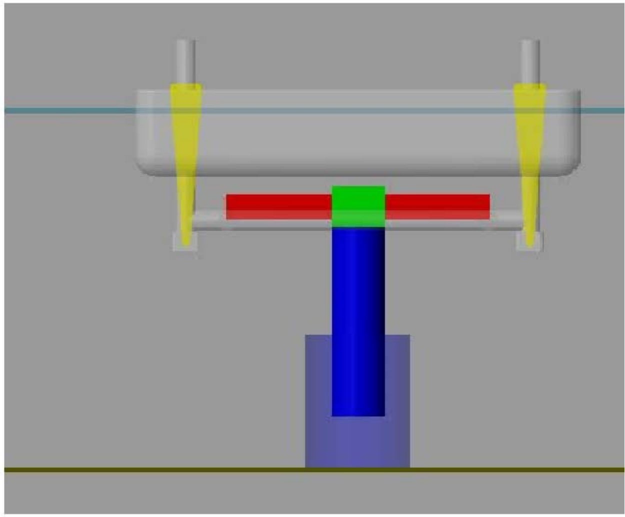
U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy







FOSWEC wave tank testing and WEC-Sim simulation

WEC-Sim Webinar #3

PTO and Control Applications

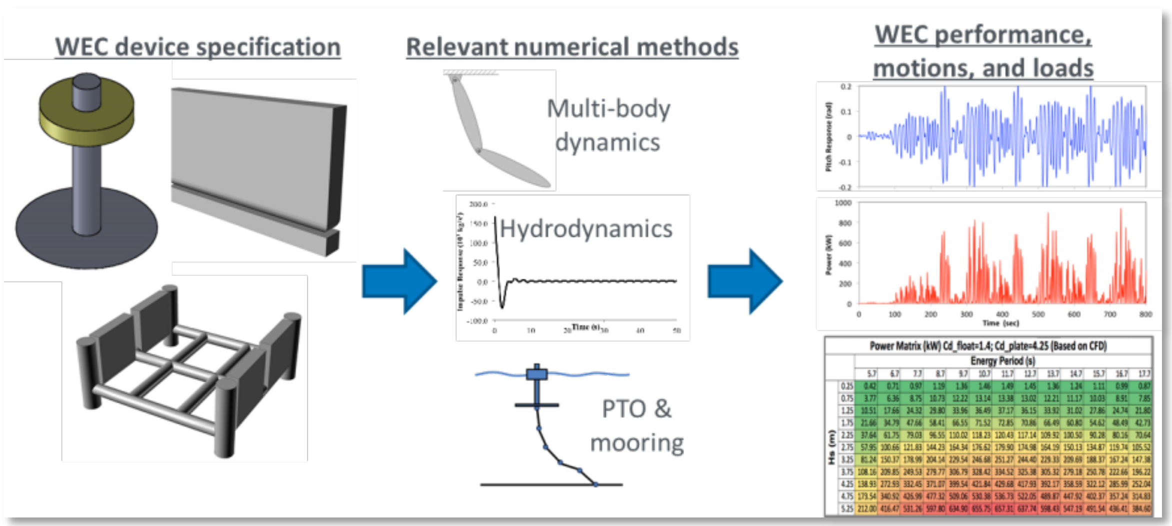
June 13, 2017

Yi-Hsiang Yu (NREL)
Kelley Ruehl (Sandia)

Introduction

WEC-Sim Team

- Kelley Ruehl (Sandia)
- Yi-Hsiang Yu (NREL)
- Jennifer van Rij (NREL)



WEC-Sim Updates

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

Reduced the size of the repo

- Remove publications from the repo since all the publications are available from the website
  http://wec-sim.github.io/WEC-Sim/publications.html
- Working on removing the large data file (e.g., *.h5 and *.mat) from repo history

Created a WEC-Sim_Application submodule in WEC-Sim

- This can be pulled into WEC-Sim currently if desired
  https://github.com/WEC-Sim/WEC-Sim_Applications
- Cleaned up PTO-Sim application cases

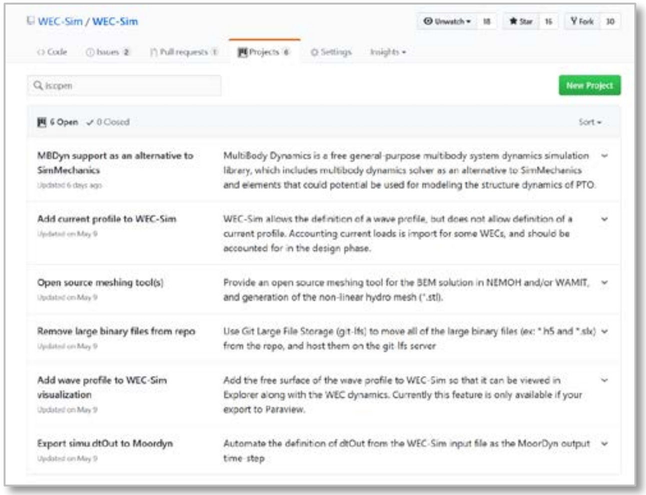
WEC-Sim Updates

Created a separate moorDyn library repo

- Due to different licenses, moorDyn is now saved in another repository
- To use MoorDyn in WEC-Sim, please download moorDyn from repo https://github.com/WEC-Sim/moorDyn
- Place all the files and folders under WEC-Sim/source/functions/moorDyn folder

Added a WEC-Sim Projects Page

- Can be used to track requested feature additions, and their status
- https://github.com/WEC-Sim/WEC-Sim/projects



Schedule

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

Advanced Feature Webinars *1hr each*

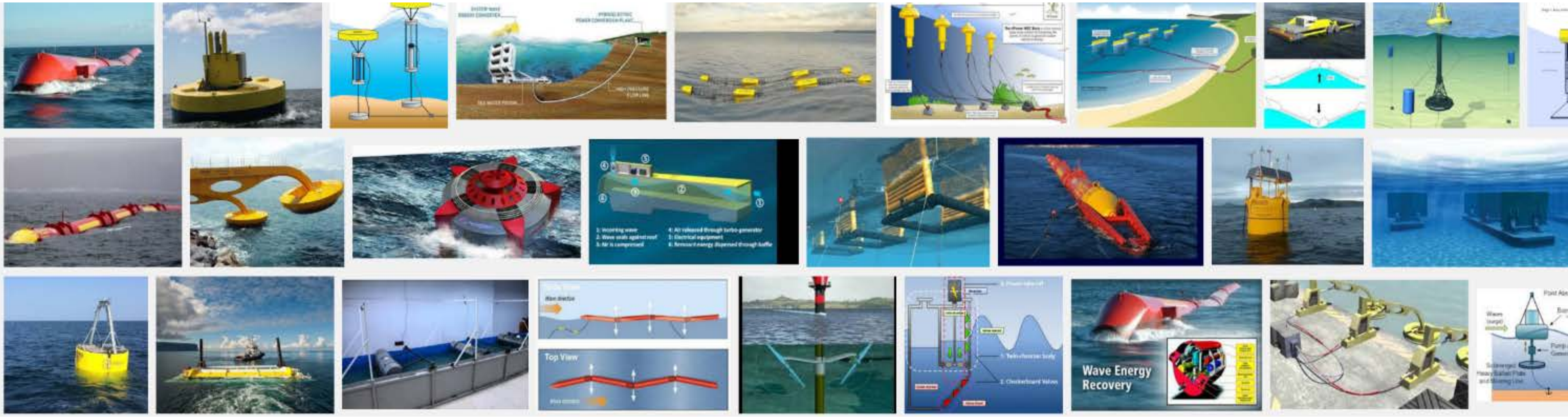
- **April 18:** bemio and mcr, application for power matrix
- **May 24:** nl-hydro, b2b, non-hydro ~~and drag~~
- **June 13:** pto and control, application for desalination
- **July 18:** mooring and visualization
- ***Available Online***: http://wec-sim.github.io/WEC-Sim/webinars.html

Training Courses

- **May 1:** *1hr* WEC-Sim workshop at METS, for new users
- **August TBD:** *half-day* WEC-Sim code structure course, for advanced users/developers

WEC-Sim Webinar #3

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

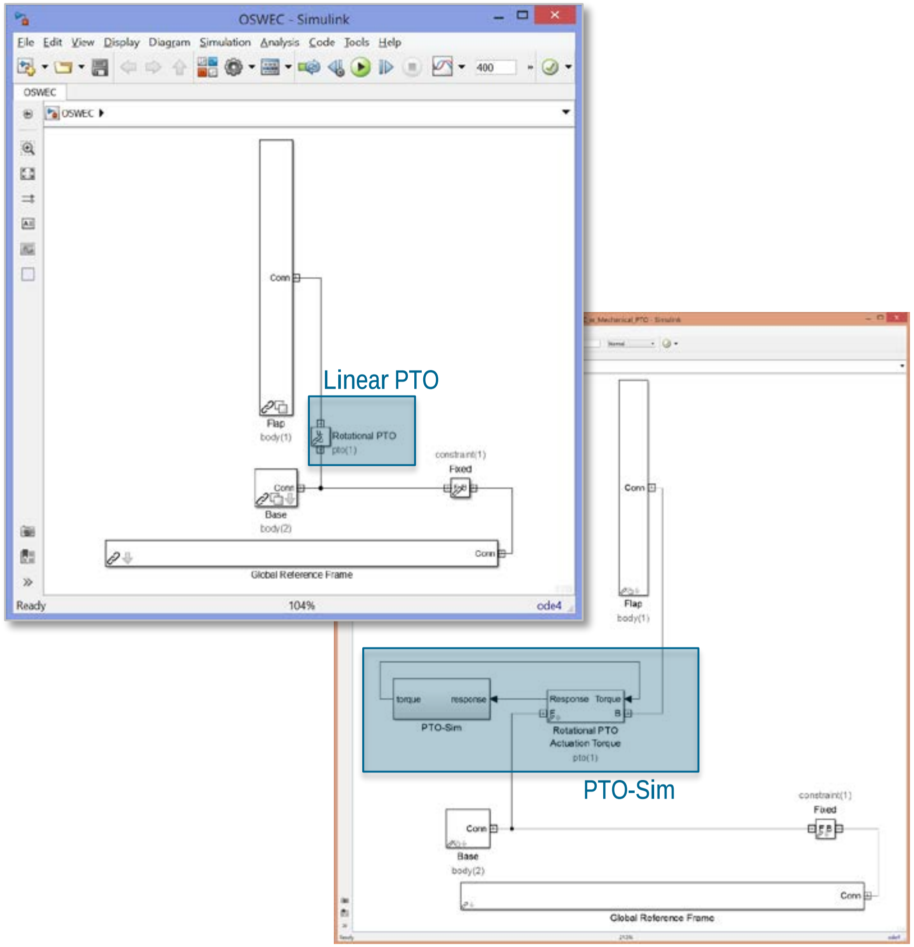


PTO and Control

Kelley Ruehl (Sandia)

WEC-Sim can model Power Take-Off (PTO) and Control with different approaches and level of fidelity

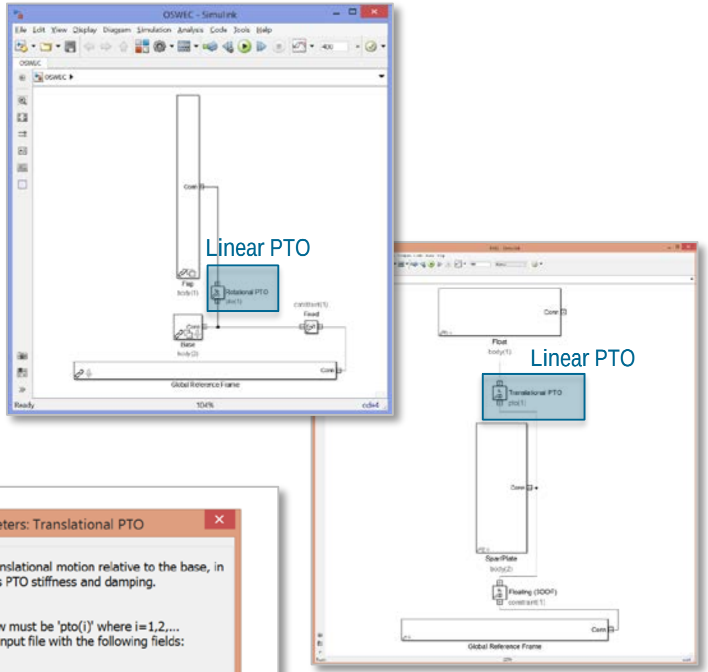
- Linear Stiffness/Damping
- Coupled with PTO-Sim
  - PM Generator
    - Components
    - Look-up table
  - Hydraulic
    - Compressible
    - Non-compressible
- User-Defined Force/Position

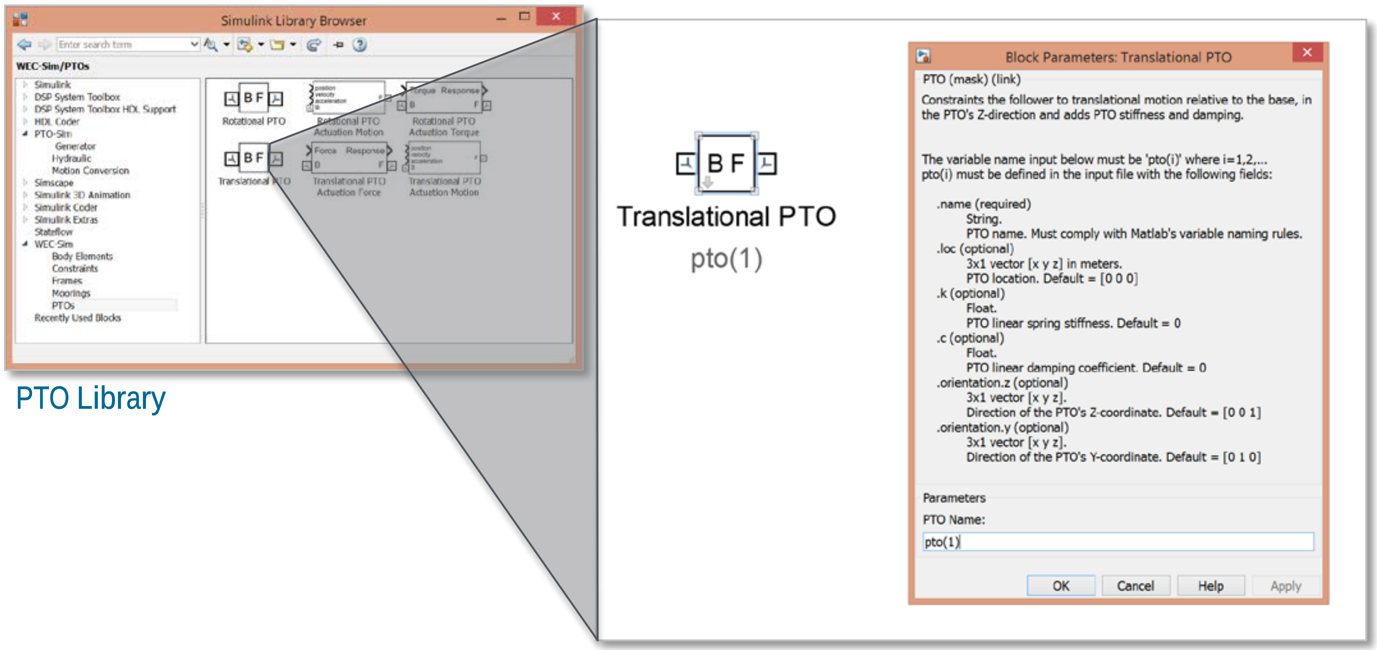


Linear Stiffness/Damping

WEC-Sim can model PTOs with linear stiffness and damping

```
pto(1) = ptoClass('PTO1'); %Initialize PTO Class
pto(1).k=0;                %PTO Stiffness [N/m]
pto(1).c=1200000;          %PTO Damping [N/(m/s)]
pto(1).loc = [0 0 0];      %PTO Location [m]
```





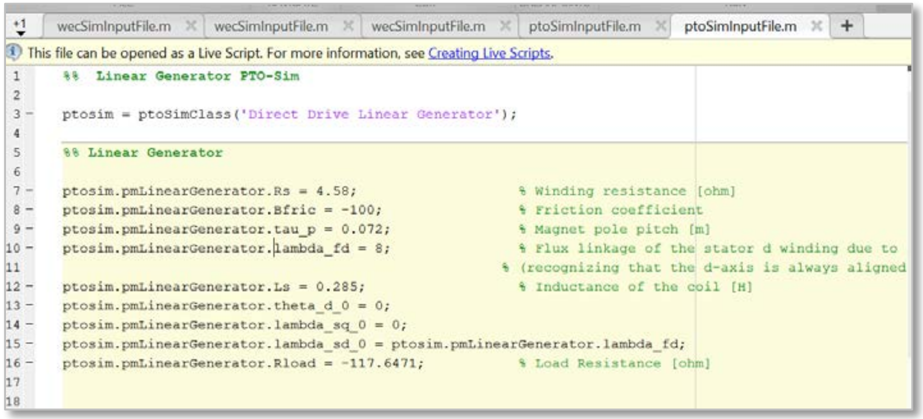
PTO Library

Coupled with PTO-Sim

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

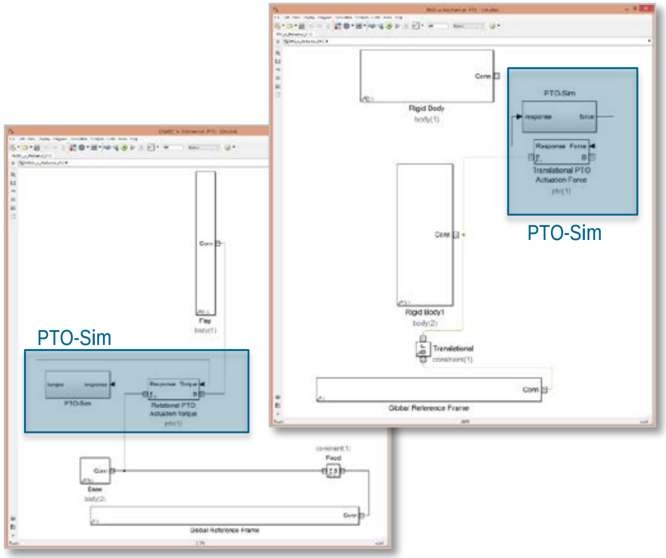
WEC-Sim can model PTOs using PTO-Sim and the actuated PTOs

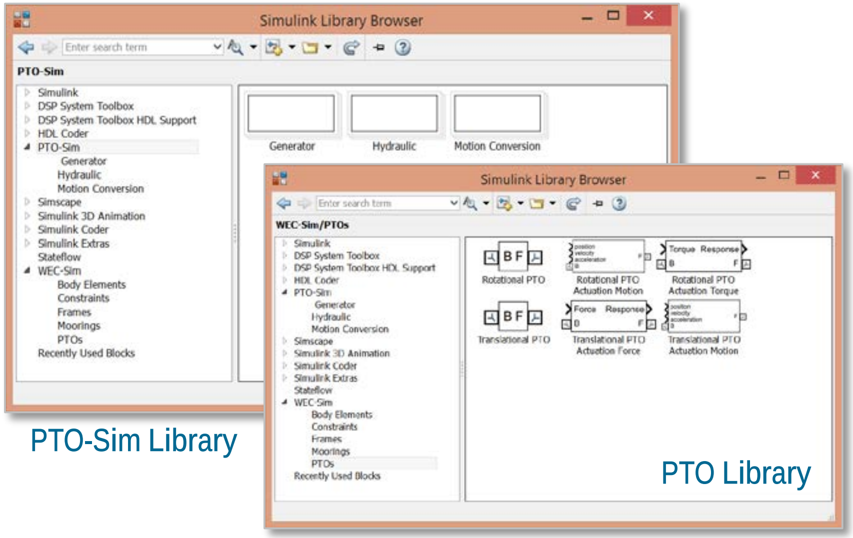
- Initialize PTO-Sim and define parameters in `ptoSimInputFile.m`



```
%% Linear Generator PTO-Sim

ptosim = ptoSimClass('Direct Drive Linear Generator');

%% Linear Generator

ptosim.pmLinearGenerator.Rs = 4.58;                    % Winding resistance [ohm]
ptosim.pmLinearGenerator.Bfric = -100;                 % Friction coefficient
ptosim.pmLinearGenerator.tau_p = 0.072;                % Magnet pole pitch [m]
ptosim.pmLinearGenerator.lambda_fd = 8;                % Flux linkage of the stator d winding due to
                                                      % (recognizing that the d-axis is always aligned
ptosim.pmLinearGenerator.Ls = 0.285;                   % Inductance of the coil [H]
ptosim.pmLinearGenerator.theta_d_0 = 0;
ptosim.pmLinearGenerator.lambda_sq_0 = 0;
ptosim.pmLinearGenerator.lambda_sd_0 = ptosim.pmLinearGenerator.lambda_fd;
ptosim.pmLinearGenerator.Rload = -117.6471;            % Load Resistance [ohm]
```

- The PTO-Sim input file will vary depending on which PTO-Sim blocks are used
- Example PTO-Sim applications are available: https://github.com/WEC-Sim/WEC-Sim_Applications

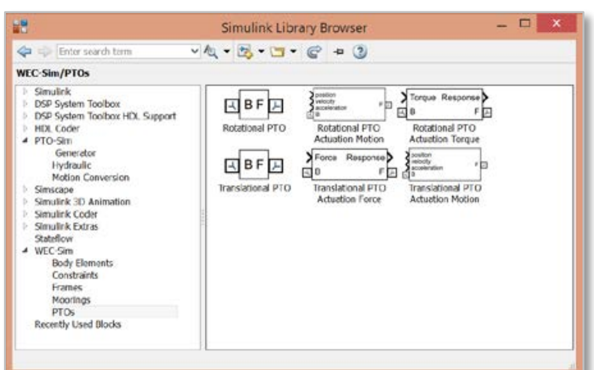




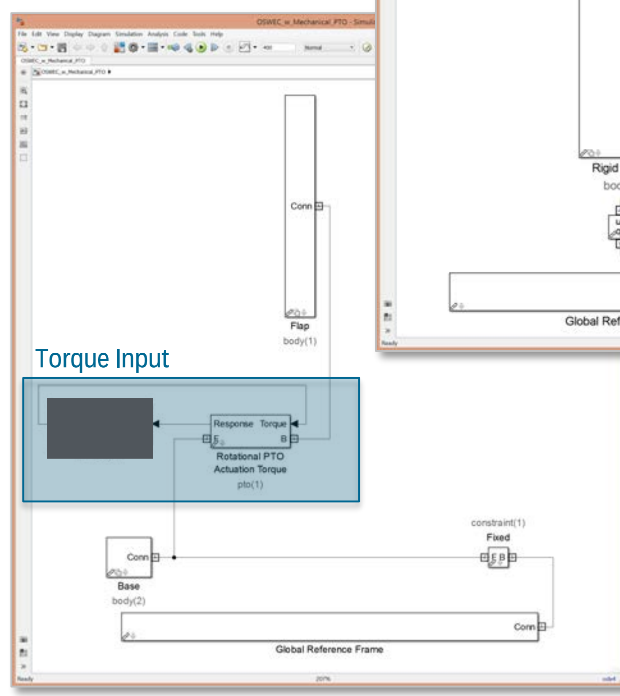
User-Defined Force/Position

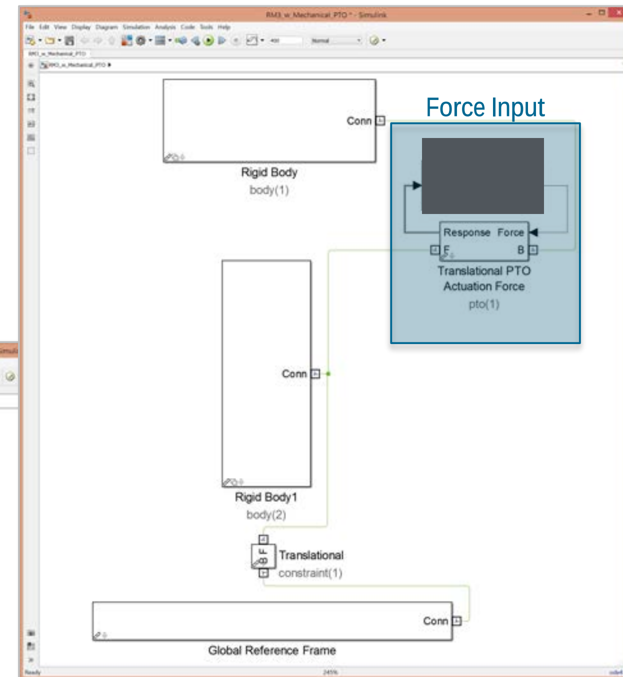
Using the WEC-Sim library PTO blocks, users cam define, input force or position in translation or rotation

- Users can build their own Simulink blocks, call on a *.m file, define a look-up table, implement control, etc.
- Use Project page for requests
- Desalination application next

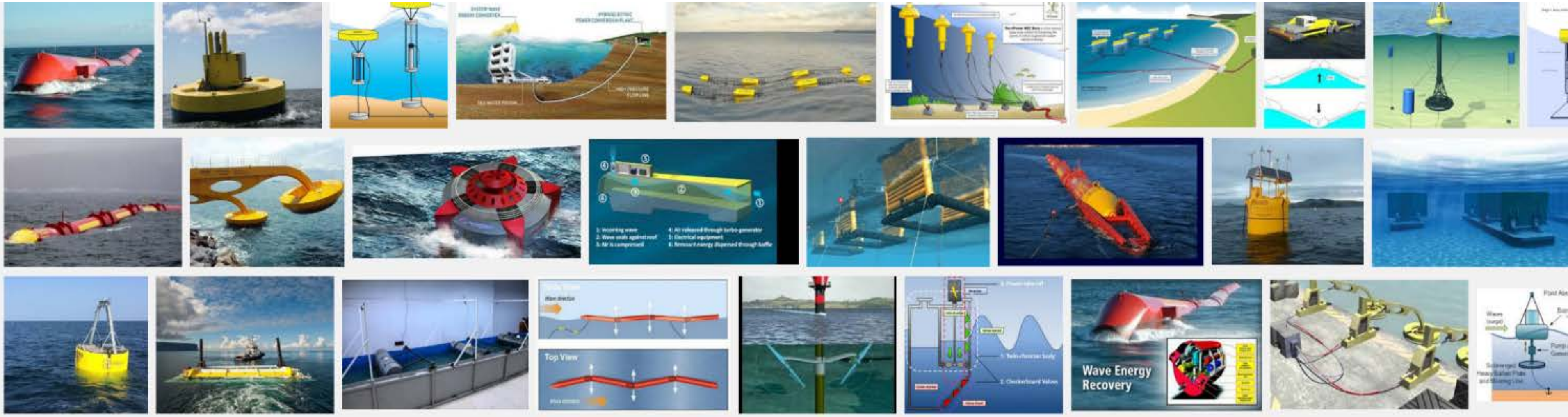


PTO Library





U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy



Desalination Applications Using WEC-Sim

Yi-Hsiang (NREL)

Desalination Applications

WEC-Sim can be further modified for other energy related production analysis by developing your own blocks.

Proceedings of the 36th International Conference on Ocean, Offshore and Arctic Engineering
OMAE 2017
June 25-30, 2017, Trondheim, Norway

OMAE2017-62136

ANALYSIS OF A WAVE-POWERED, REVERSE-OSMOSIS SYSTEM AND ITS ECONOMIC AVAILABILITY IN THE UNITED STATES

Yi-Hsiang Yu*
National Renewable Energy Laboratory
Golden, CO, USA
Email: yi-hsiang.yu@nrel.gov

Dale Jenne
National Renewable Energy Laboratory
Golden, CO, USA
Email: dale.jenne@nrel.gov

ABSTRACT

A wave energy converter (WEC) system has the potential to convert the wave energy resource directly into the high-pressure flow that is needed by the desalination system to permeate saltwater through the reverse-osmosis membrane to generate clean water. In this study, a wave-to-water numerical model was developed to investigate the potential use of a wave-powered desalination system (WPDS) for water production in the United States. The model was developed by coupling a time-domain radiation-and-diffraction-method-based numerical tool (WEC-Sim) for predicting the hydrodynamic performance of WECs with a solution-diffusion model that was used to simulate the reverse-osmosis process. To evaluate the feasibility of the WPDS, the wave-to-water numerical model was applied to simulate a desalination system that used an oscillating surge WEC device to pump seawater through the system. The annual water production was estimated based on the wave resource at a reference site on the coast of northern California to investigate the potential cost of water in that area, where the cost of water and electricity is high compared to other regions. In the scenario evaluated, for a 100-unit utility-scale array, the estimated levelized cost of energy for these WECs is about 3-6 times the U.S.'s current, unsubsidized electricity rates. However, with clean water as an end product and by directly producing pressurized water with WECs, rather than electricity as an intermediary, it is presently only 12% greater than typical water cost in California. This study suggests that a WEC array that produces water may be a viable, near-term solution to the nation's water supply, and the niche application of the WPDS may also provide developers with new opportunities to further develop technologies that benefit both the electric and drinking water markets.

KEYWORDS
Wave energy; desalination; reverse osmosis; time-domain numerical model; cost of energy; cost of water

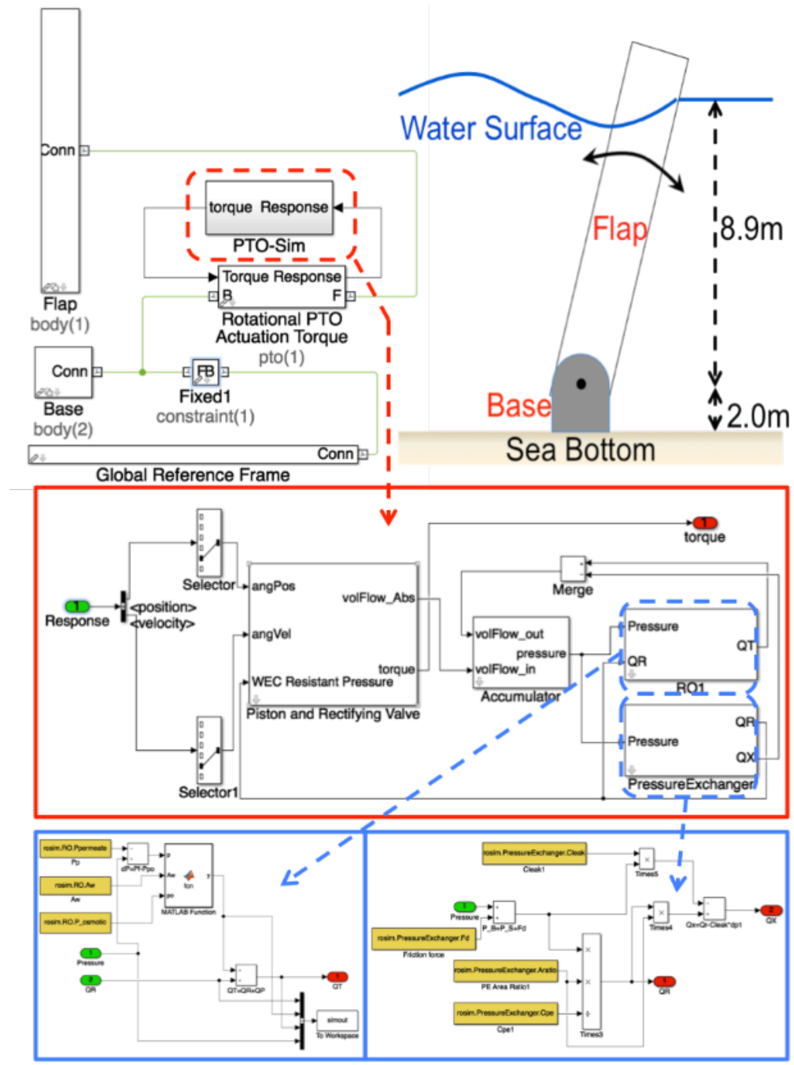
INTRODUCTION

Developing alternative water resources through the use of desalination is important to human activities. But desalination technologies are inherently energy-intensive, with the majority of processes requiring high levels of electricity consumption. Further, in many areas where water is scarce, electricity prices are also high, for example, in California, or areas with limited electrical grid connection (e.g., small islands and isolated coastal communities). Thus, reverse osmosis has been the most commonly used desalination process because of the lower energy consumption than traditional thermal processes, which, among other reasons, makes reverse osmosis one of the most promising desalination approaches. However, it still requires a great amount of energy to create the pressure needed to pump the saltwater through the reverse-osmosis membranes. A wave energy converter (WEC) system has the potential to convert wave energy directly into the high-pressure flow that is needed by the reverse-osmosis system and eliminate the electricity production process to potentially reduce the cost of water. In addition, the niche application of wave-powered desalination will also provide a great opportunity to further advance wave energy technologies for both water and electricity generation.

Several designs have been proposed to develop wave-powered desalination plants in the past 30 years, e.g., a linear-pump-based buoy system (Delbuoy) in the 1980s [1], an oscillating water column type of WEC in India in 2004 [2], and more recently a fixed-bottom flapper design from Resolute Marine [3].

*Address all correspondence to this author.

1

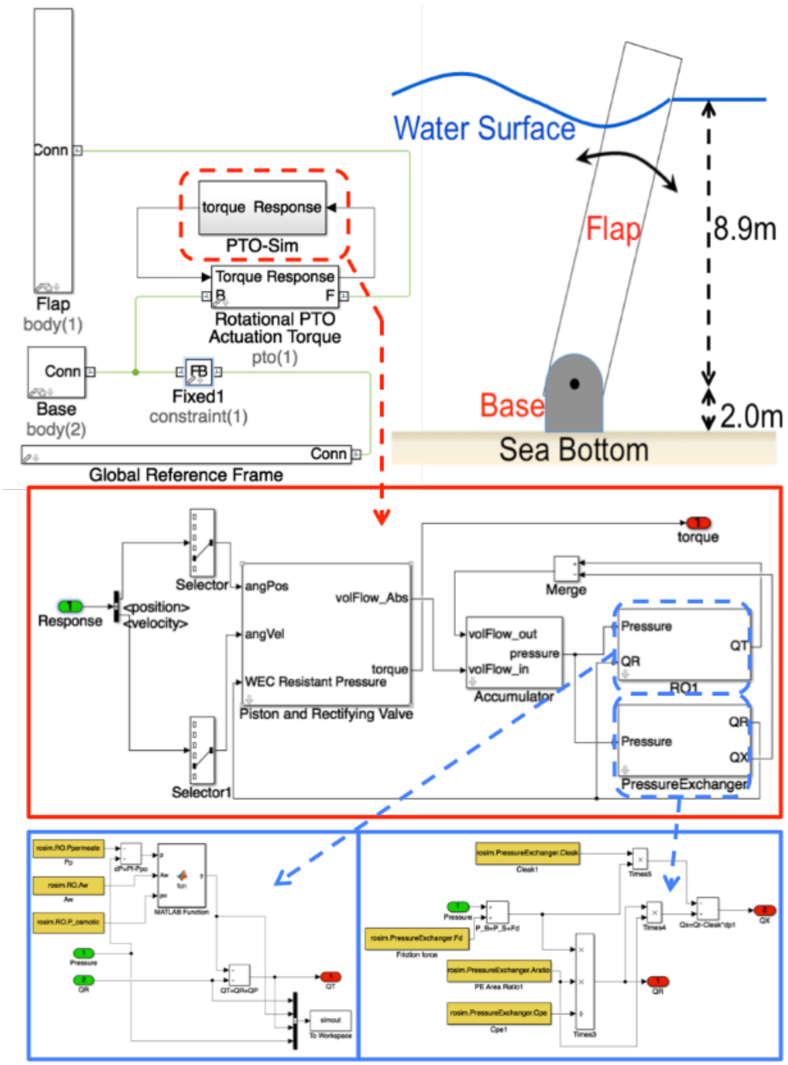


Desalination Applications

WEC-Sim can be further modified for other energy related production analysis by developing your own blocks.

- This webinar will focus on how this can be done from an existing WEC-Sim case.

- The results of the **Reverse Osmosis** (RO) desalination analysis will be presented at OMAE 2017, Trondheim, Norway, June 25-30.



WEC-Sim Theory

- Dynamics simulated by solving time-domain equation of motion (Cummins, 1962)

$$m\ddot{x}(t) = \boxed{f_{hs}(t)} + \boxed{f_{ex}(t)} + \boxed{f_{rad}(t)} + \boxed{f_{v}(t)} + \boxed{f_{pto}(t)} + \boxed{f_{m}(t)}$$

Hydrostatic restoring force

Wave excitation & diffraction force (from BEM simulations)

Radiation force: added mass and radiation damping (from BEM simulations)

Viscous force

Power take-off force

Mooring force

- Use radiation and diffraction method and calculate the hydrodynamic forces from frequency-domain Boundary Element Method (BEM)

$$f_{rad}(t) = -A_{\infty}\ddot{X} - \int_0^t K(t-\tau)\dot{X}(\tau)d\tau$$

BEM BEM

$$f_{ex}(t) = \Re\left[R_f F_X(\omega_r)e^{i(\omega_r t+\phi)}\int_0^{\infty}\sqrt{2S(\omega_r)d\omega_r}\right]$$

BEM

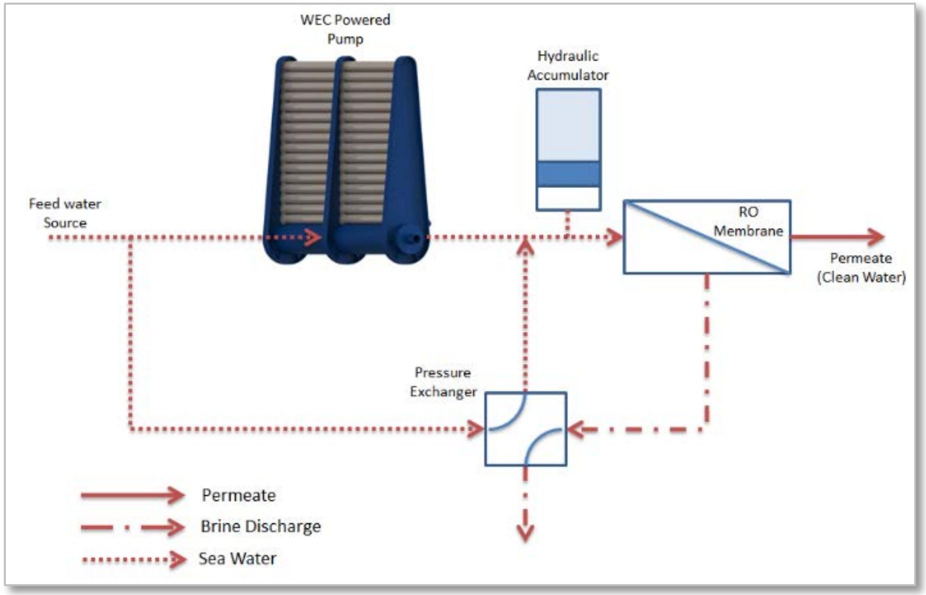
$$= \int_{-\infty}^{\infty}\eta(\tau)f_e(t-\tau)d\tau$$

BEM

Wave-Powered Desalination System

- For WEC, the RO membrane can be modeled as a “PTO”, providing forces back to the WEC system.

- For the RO membrane, the WEC can be modeled as an hydraulic pump with oscillating flow velocity.



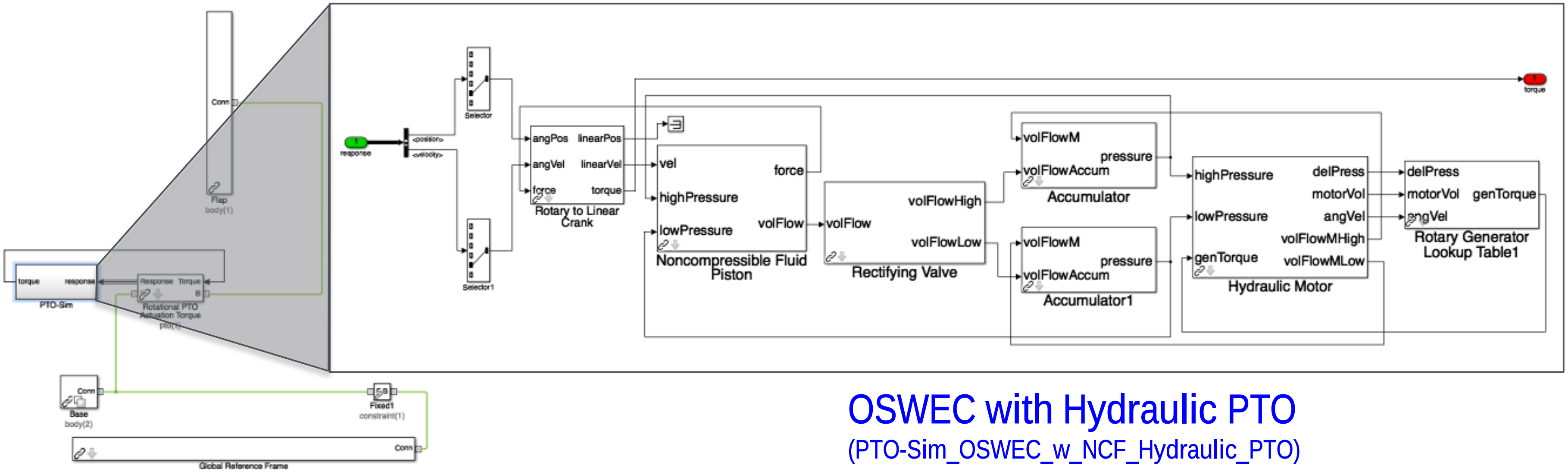
Solution-Diffusion Model:

$$Q_P = A_\omega(\Delta p - \Delta\pi)$$

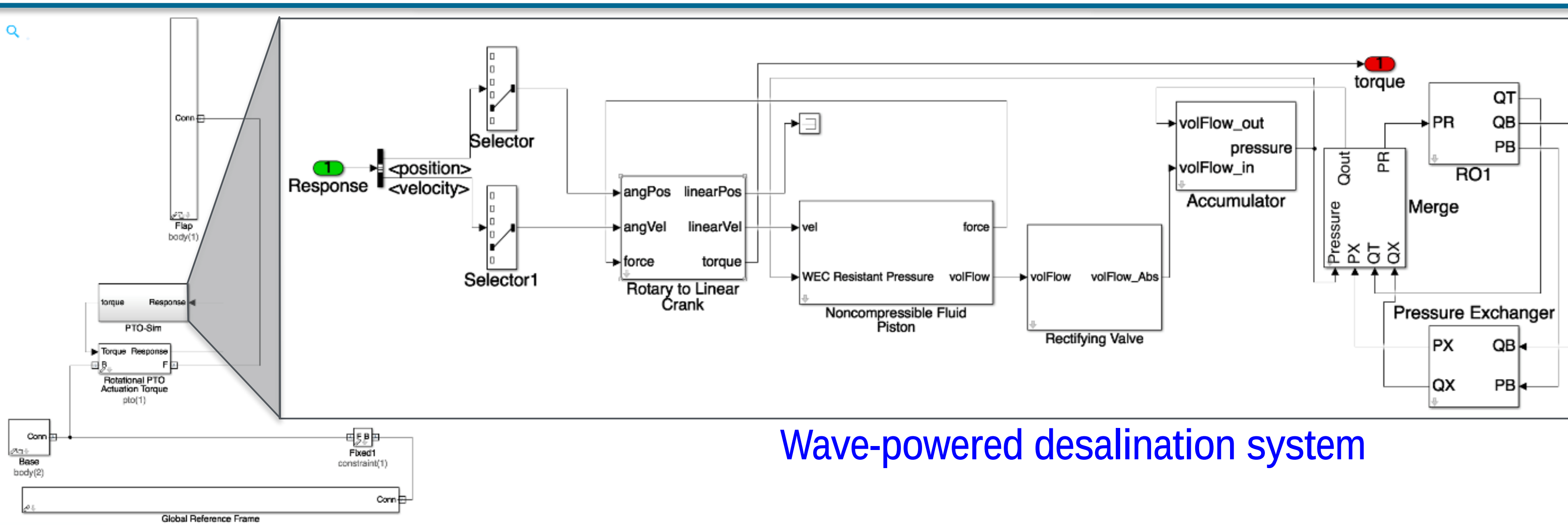
where A_ω is the permeability coefficient, depending on the membrane permeability, membrane surface area, temperature, and fouling factor.

WEC-Sim/PTO-Sim Model

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy



OSWEC with Hydraulic PTO
(PTO-Sim_OSWEC_w_NCF_Hydraulic_PTO)



Wave-powered desalination system

WEC-Sim/PTO-Sim Model

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

- Same hydrodynamic and motion conversion models
  - Hydrodynamics model
  - WEC-Sim input file
  - “Rotary to LinearCrank”, “Noncompressible Fluid Piston” and “Rectifying Valve” blocks under PTO-Sim

- The hydraulic motors and generator were replaced by a RO membrane block and a pressure exchanger block.

- The primary difference between the two models:
  - OSWEC with non-compressible fluid hydraulic system is a closed loop system circulated with high pressure fluid.
  - Wave-powered desalination system is open to atmosphere

WEC-Sim Input File

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

OSWEC with Hydraulic PTO

(PTO-Sim_OSWEC_w_NCF_Hydraulic_PTO)

wecSimInputFile.m | startup.m | wecSimInputFile.m

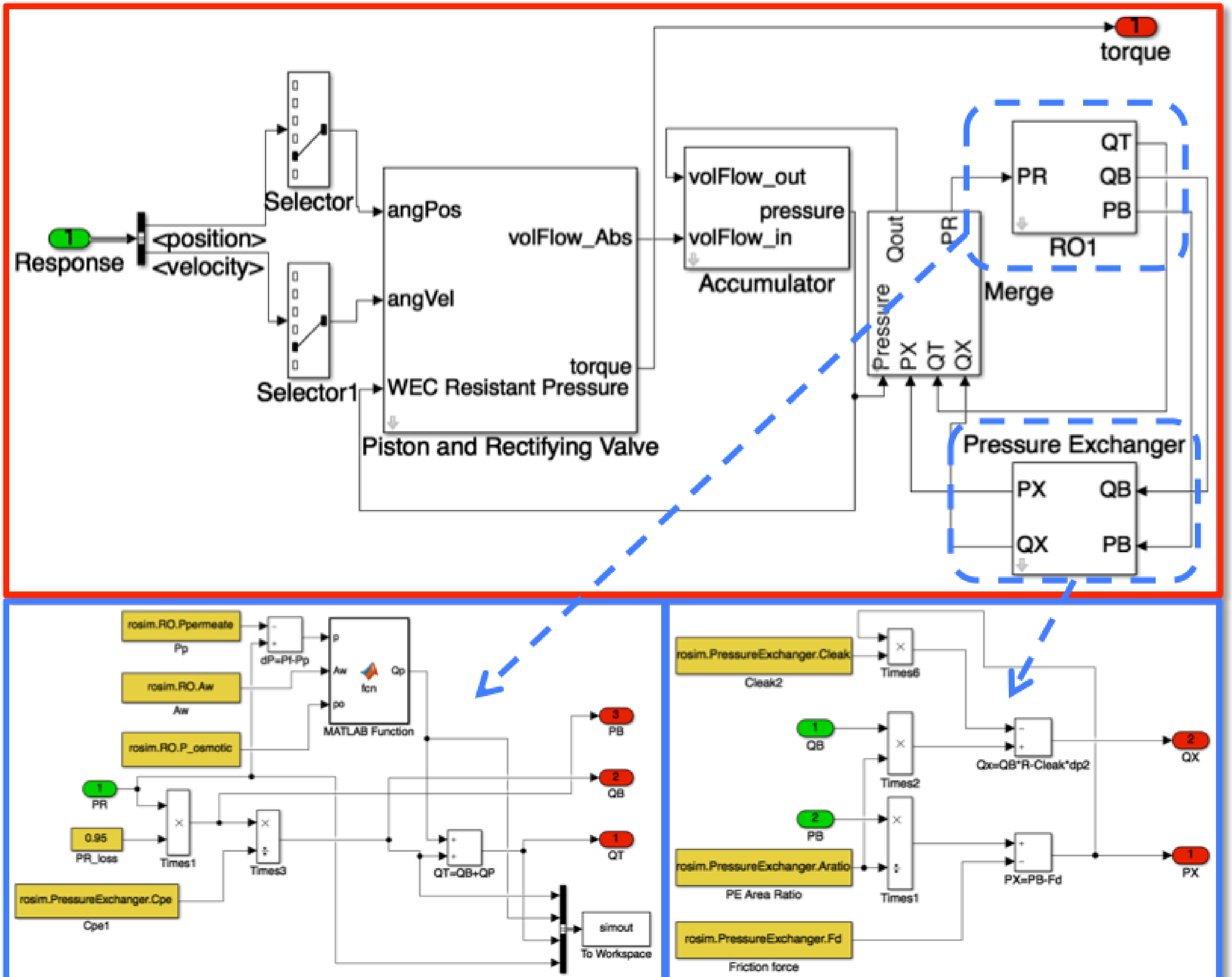
```matlab
%% Simulation Data
simu = simulationClass();                      % Initialize Wave Class
simu.simMechanicsFile = 'OSWEC_w_NCFHydraulic.slx';     % Specify Simulink Model File
simu.mode = 'normal';                          % Specify Simulation Mode ('normal','accelerator','
simu.explorer='on';                            % Turn SimMechanics Explorer (on/off)
simu.startTime = 0;                            % Simulation Start Time [s]
simu.endTime=500;                              % Simulation End Time [s]
simu.dt = 0.01;                                % Simulation Time-Step [s]
simu.rampT = 100;                              % Wave Ramp Time Length [s]
simu.CITime = 30;                              % Specify CI Time

%% Wave Information
%Irregular Waves using PM Spectrum
waves = waveClass('irregular');                % Initializa Wave Class and Specify Type
waves.H = 2.5;                                 % Significant Wave Height [m]
waves.T = 8;                                   % Peak Period [s]
waves.spectrumType = 'PM';                     % Specify Wave Spectrum
waves.randPreDefined=1;                        % Seed Random Phase

%% Body Data
body(1) = bodyClass('../../tutorials/OSWEC/hydroData/oswec.h5');   % Initialize bodyClass f
body(1).geometryFile = '../../tutorials/OSWEC/geometry/flap.stl';    % Geometry File
body(1).mass = 127000;                                % User-Defined mass [kg]
body(1).momOfInertia = [1.85e6 1.85e6 1.85e6];  % Moment of Inertia [kg-m^2]
body(1).linearDamping = [0, 0, 0, 0, 1*10^7, 0];

body(2) = bodyClass('../../tutorials/OSWEC/hydroData/oswec.h5');   % Initialize bodyClass f
body(2).geometryFile = '../../tutorials/OSWEC/geometry/base.stl';    % Geometry File
body(2).mass = 'fixed';                               % Creates Fixed Body

%% PTO and Constraint Parameters
constraint(1)= constraintClass('Constraint1');  % Initialize ConstraintClass for Constraint
constraint(1).loc = [0 0 -10];                  % Constraint Location [m]

%% Rotational PTO
pto(1) = ptoClass('PTO1');                      % Initialize ptoClass for PTO1
pto(1).k = 0;                                   % PTO Stiffness Coeff [Nm/rad]
pto(1).c =0;                                    % PTO Damping Coeff [Nsm/rad]
pto(1).loc = [0 0 -8.9];                        % PTO Global Location [m]
```

Wave-powered desalination system

wecSimInputFile.m | wecSimInputFile.m

```matlab
%% Simulation Data
simu = simulationClass();
simu.simMechanicsFile = 'OSWEC_WM.slx';          % Specify Simulink Model File
simu.mode = 'rapid-accelerator';                        % Specify Simulation Mode ('nc
simu.explorer='off';                           % Turn SimMechanics Explorer (on/off)
% simu.startTime = 0;                           % Simulation Start Time [s]
simu.endTime=2000;
simu.solver = 'ode4';                           %simu.solver = 'ode4' for fixed step &
simu.dt = 0.05;                                     %Simulation time-step [s] for a cor
simu.rampT = 250;
simu.CITime = 30;
simu.morrisonElement = 1;

%% Wave Information
%% noWaveCIC, no waves with radiation CIC
% waves = waveClass('noWaveCIC');       %Create the Wave Variable and Specify Type

%% Regular Waves
% waves = waveClass('regular');
% waves.H = 2.5;
% waves.T = 8;

%% Irregular Waves using PM Spectrum with Convolution Integral Calculation
waves = waveClass('irregular');
waves.H = 2.25;
waves.T = 8.7;
waves.spectrumType = 'BS';
waves.randPreDefined = 1;

%% Irregular Waves using User-Defined Spectrum
% waves = waveClass('irregularImport');
% waves.spectrumDataFile = 'ndbcBuoyData.txt';

%% Body Data
%% Flap
body(1) = bodyClass('./hydroData/oswec.h5');   % Initialize bodyClass for Flap
body(1).bemioFlag = 0;  % If using the new MATLAB based BEMIO
body(1).mass = 127000;                          % User-Defined mass [kg]
body(1).momOfInertia = [1.85e6 1.85e6 1.85e6]; % Moment of Inertia [kg-m^2]
body(1).geometryFile = './geometry/flap.stl';    % Geometry File
body(1).morrisonElement.cd = ones (5,3);
body(1).morrisonElement.ca = zeros(5,3);
body(1).morrisonElement.characteristicArea = zeros(5,3);
body(1).morrisonElement.characteristicArea(:,1) = 18*1.8;
body(1).morrisonElement.characteristicArea(:,3) = 18*1.8;
body(1).morrisonElement.VME  = zeros(5,1);
body(1).morrisonElement.rgME = [0 0 -3; 0 0 -1.2; 0 0 0.6; 0 0 2.4; 0 0 4.2];

%% Base
body(2) = bodyClass('./hydroData/oswec.h5');   % Initialize bodyClass for Base
body(2).bemioFlag = 0;  % If using the new MATLAB based BEMIO
body(2).geometryFile = './geometry/base.stl';    % Geometry File
body(2).mass = 'fixed';                         % Creates Fixed Body

%% PTO and Constraint Parameters
constraint(1)= constraintClass('Constraint1'); % Initialize ConstraintClass for Const
constraint(1).loc = [0 0 -10];

pto(1) = ptoClass('PTO1');                      % Initialize ptoClass for PTO1
pto(1).k = 0;                                   % PTO Stiffness Coeff [Nm/rad]
pto(1).c = 0;                                   % PTO Damping Coeff [Nsm/rad]
pto(1).loc = [0 0 -8.9];                        % PTO Global Location [m]
```

RO-Sim Blocks



RO-Sim Class (roSimInputFile.m)

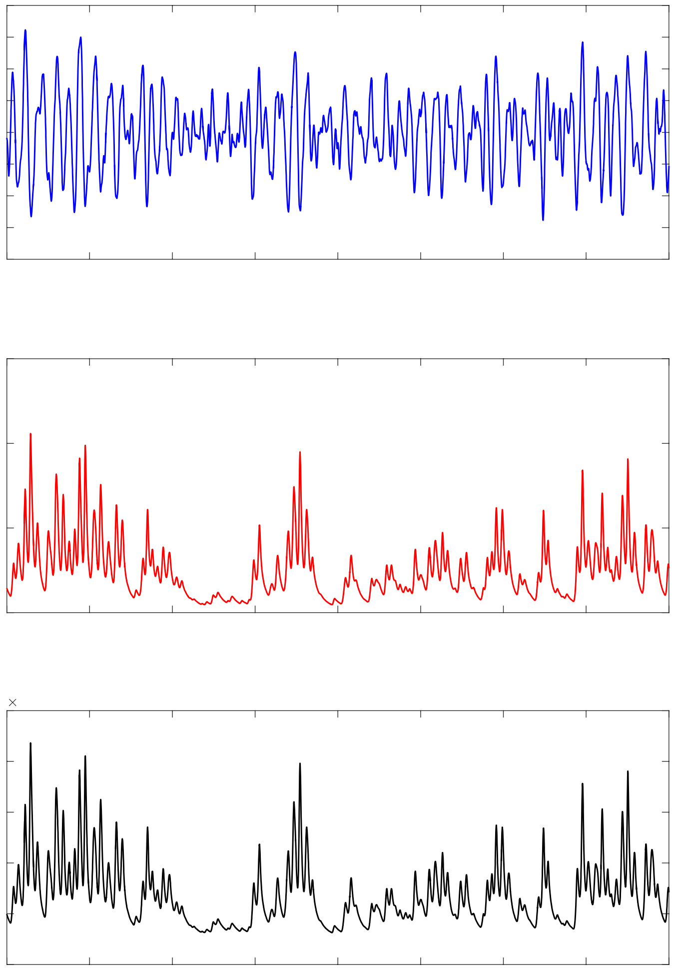
U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

wecSimInputFile.m | wecSimInputFile.m | roSimInputFile.m

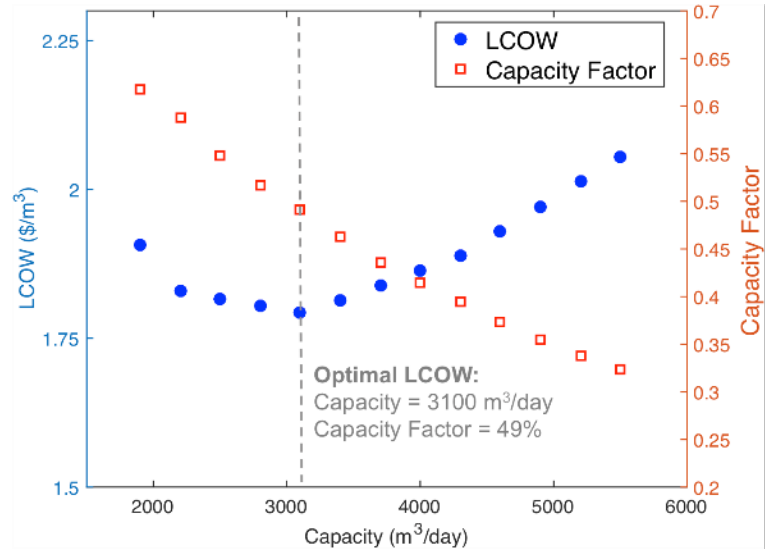
```
rosim = roSimClass('RO-Sim');

% %% Piston
rosim.pistonNCF.topA = 0.180188206642432;
rosim.pistonNCF.botA = 0.180188206642432;

% %% Rotary to Linear Crank
rosim.motionMechanism.crank = 3;
rosim.motionMechanism.offset = 1.3;
rosim.motionMechanism.rodLength = 5;

% %% Env
rosim.inflow.Pinlet   = 0;

% %% RO
rosim.RO(1).Aw         = 184798535.242295; %0.4E7/rosim.pistonNCF.topA^2;
rosim.RO(1).Ppermeate = 0;
rosim.RO(1).P_osmotic = 330*6.89476*1000;

% %% High Pressure Accumulator
rosim.accumulator(1).VI0 = 6;
rosim.accumulator(1).pIprecharge = 0.3789e5;
rosim.accumulator(1).VIeq= rosim.accumulator(1).VI0/2;

% %% Pressure Exchanger
rosim.PressureExchanger.Aratio = 0.95;
rosim.PressureExchanger.Cpe     = 184798535.242295/2; %0.4E7/rosim.pistonNCF.topA^2;
rosim.PressureExchanger.Cleak  = 1e-11;
rosim.PressureExchanger.Fd     = 100/((0.15/2)^2*pi);
```

Results (Yu & Jenne, OMAE 2017)

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy



Significant Wave Height Hs (m) \ Energy Period, Te [sec]	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5
0.25				8	8									
0.75	134	174	199	217	229	238	238	260	273					
1.25	381	504	593	640	668	673	692	731	759	767	752	723		
1.75		1012	1163	1256	1316	1354	1338	1402	1433	1427	1383	1314		
2.25			1945	2111	2144	2198	2147	2244	2266	2234	2146	2027	1895	
2.75			2874	3100	3100	3100	3100	3100	3100	3100	3018	2839	2641	
3.25				3100	3100	3100	3100	3100	3100	3100	3100	3100	3100	3100
3.75				3100	3100	3100	3100	3100	3100	3100	3100	3100	3100	
4.25					3100	3100	3100	3100	3100	3100	3100	3100	3100	
4.75					3100	3100	3100	3100	3100	3100	3100	3100	3100	
5.25						3100	3100	3100	3100	3100	3100	3100		
5.75								3100	3100	3100	3100	3100		
Peak Period, Tp [sec]	5.2	6.4	7.5	8.7	9.9	11.0	12.2	13.3	14.5	15.7	16.8	18.0	19.1	20.3



Work To Be Done

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

- Work in the process
  - Merge back to the latest WEC-Sim release and add the desalination modeling capability to WEC-Sim.
  - Add the documentation for desalination application to WEC-Sim Github website.
  - Upload to the example run case to WEC-Sim Application repo.

Thank you!

U.S. DEPARTMENT OF ENERGY | Energy Efficiency & Renewable Energy

Upcoming scheduled webinars and training courses…

Advanced Feature Webinars *1hr each*

- **April 18:** bemio and mcr, application for power matrix
- **May 24:** nl-hydro, b2b, non-hydro ~~and drag~~
- **June 13:** pto and control, application for desalination
- **July 18:** mooring and visualization
- ***Available Online***: http://wec-sim.github.io/WEC-Sim/webinars.html

Training Courses

- **May 1:** *1hr* WEC-Sim workshop at METS, for new users
- **August TBD:** *half-day* WEC-Sim code structure course, for advanced users/developers