Data-driven and experimental modeling of an oscillating surge wave energy converter

Brittany Lydon

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2025

Reading Committee:
Brian Polagye, Chair
Steven Brunton, Chair
Michelle DiBenedetto

Program Authorized to Offer Degree:
Mechanical Engineering

University of Washington

Abstract

Data-driven and experimental modeling of an oscillating surge wave energy converter

Brittany Lydon

Co-Chairs of the Supervisory Committee:
Brian Polagye
Mechanical Engineering

Steven Brunton
Mechanical Engineering and Applied Mathematics

Wave energy converters (WECs) are devices that harness power from ocean waves with useful applications including powering remote communities and desalination plants. Oscillating surge WECs (OSWECs) are a promising sub-category of WECs because they can absorb power over a wide range of wave frequencies and rely less on resonance than other WEC archetypes. However, OSWECs can exhibit complex dynamics that can be difficult to accurately model without significant computational burden. Often, this results in either low-fidelity, potentially inaccurate linear models or high-fidelity numerical simulations that are too computationally expensive for operational use. Scaled experimental modeling can address these issues, however there are limitations when the scaled device does not achieve exact similarity to the full-scale system. When this occurs, data from the experiment may not be representative of the full-scale system and could cause errors when predicting device behavior and performance. This work explores data-driven and experimental methods that address these limitations. Specifically, we use data-driven algorithms that balance accuracy and computation speed to build accurate system models of OSWEC dynamics. We address realistic conditions such as noisy signals and nonlinear events, including wave overtopping. Additionally, we demonstrate experimental techniques that mitigate common scale effects

using control, providing an efficient method to investigate and address hardware limitations.

To bridge the gap between model accuracy and computation speed, we propose the use of dynamic mode decomposition (DMD) as a purely data-driven technique that can generate an accurate and computationally efficient model of OSWEC dynamics. Specifically, we model and predict the behavior of an OSWEC in mono- and polychromatic seas without an equation of motion or knowledge of the incident wave field. We generate data with the open-source code WEC-Sim, then evaluate how well DMD can describe past dynamics and predict future behavior. We consider realistic challenges including noisy sensors, weakly nonlinear dynamics, and irregular wave forcing. Specifically, by using an extension of DMD we reduce the effect of noise on our system and significantly increase model accuracy outside the training region. Additionally, by introducing time delays we accurately describe weakly nonlinear dynamics, even though DMD is a linear algorithm. Finally, we use Optimized DMD (optDMD) to model OSWEC behavior in response to irregular waves. While optDMD accurately models training data, future prediction remains inaccurate, demonstrating the limits of modeling efforts without access to information about the incident wave field. These findings provide insight into the use of DMD, and its extensions, on systems with limited time-resolved data and present a framework for applying similar analysis to lab- or field-scale experiments.

Limitations of linear models are magnified in energetic seas where extreme events, such as overtopping and slamming, and large-amplitude motions can introduce significant nonlinearities in the dynamics. To address this, we use the sparse identification of nonlinear dynamics (SINDy) algorithm to build parsimonious reduced-order models of WEC dynamics from nonlinear experimental data. Specifically, we learn interpretable, nonlinear models to describe OSWEC dynamics in response to varying amounts of overtopping. The SINDy models are trained on experimental data from a laboratory-scale device operating in regular waves. These models describe the surge and heave forces acting on the OSWEC flap using

only kinematic time series. While surge force is relatively unaffected by overtopping severity, heave force shows significant nonlinear behavior with increased overtopping. Regardless of complexity, SINDy generates accurate models of both surge and heave forces in tests ranging from no overtopping to severe overtopping, especially when compared to linear techniques. We explore how well these models generalize to other experiments and find that we can use a single model to describe surge force over the full range of overtopping, while heave force requires a continuum of models with varying degrees of nonlinearity to accurately describe the full range of dynamics. Overall, this work supports the use of SINDy to generate accurate models of WEC dynamics, especially when strong nonlinearities are present.

While scaled-model testing is a promising method for collecting high-quality data, scale effects can reduce result accuracy. Here, we identify, quantify, and mitigate scale effects of the same laboratory-scale OSWEC that would not be equally present in full-scale systems, including reduced buoyancy due to sensor weight and driveline friction. Specifically, after we characterize the buoyancy and friction torque profile, we use a servomotor to emulate additional buoyancy and offset friction using real-time position and torque feedback. We find that emulating additional buoyancy substantially improves system performance by adjusting the natural period of the device, essentially implementing phase control to operate closer to resonance. Because of this, sensor weight should be considered in scaled-devices to ensure performance is representative of the full-scale system. However, friction, while appreciable in comparison to the prescribed control torque, has a limited effect on system kinematics and performance. This is because the frictional torque is in phase with velocity and therefore acts as a damper in the same manner as the power takeoff. These examples demonstrate the importance of considering scale effects in experimental testing and introduces novel methods to address and mitigate these effects using feedback control, providing a low-cost and efficient method to investigate and address hardware limitations.

With this work, we aim to further WEC development by introducing data-driven and

experimental methods that expand the existing WEC modeling tools and provide low-cost and efficient methods for accurate scale model testing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Acronyms

ANN	Artificial neural network
BEM	Boundary element method
CFD	Computational fluid dynamics
CoM	Center of mass
CWR	Capture width ratio
DMD	Dynamic mode decomposition
DMD+TD	Dynamic mode decomposition with time delay(s)
eDMD	Extended DMD
FFT	Fast-Fourier transform
FK	Froude-Krylov
LC	Load cell
MPC	Model predictive control
NREL	National Renewable Energy Laboratory
ODE	Ordinary differential equation
optDMD	Optimized DMD
OSWEC	Oscillating surge wave energy converter
OT	Overtopping
PSD	Power spectral distribution
PTO	Power takeoff

RMS	Root mean square
SINDy	Sparse identification of nonlinear dynamics
SNR	Signal-to-noise ratio
SPH	Smoothed particle hydrodynamics
SVD	Singular value decomposition
SWEL	Sea Wave Environmental Laboratory
SWL	Still water level
TLS DMD	Total-least-squares DMD
WEC	Wave energy converter
WG	Wave gauge

Coordinates

x	Surge direction, [m]
y	Sway direction, [m]
z	Heave direction, [m]

DMD Parameters

$\boldsymbol{\mathcal{A}}$	Continuous-time system dynamics
$\mathbf{A}$	Discrete-time system dynamics
b	DMD mode amplitude
$\mathbf{b}$	Vector of DMD mode amplitudes
r	Rank
$\mathbf{U}, \mathbf{V}$	Left and right singular matrices
$\boldsymbol{\Sigma}$	Singular value matrix
$\boldsymbol{x}_k$	Column vector of all state variables at time t_k, $(n \times 1)$
$\boldsymbol{X}$	Matrix of state variables for DMD $(n \times m)$

γ	Continuous-time DMD eigenvalue
$\mathbf{\Gamma}$	Matrix of continuous-time DMD eigenvalues
$\boldsymbol{\phi}$	DMD mode
$\mathbf{\Phi}$	Matrix of DMD modes
ψ	Discrete-time DMD eigenvalue
$\mathbf{\Psi}$	Matrix of discrete-time DMD eigenvalues

Dynamic Parameters

$\mathbf{F}_{\mathrm{b}}$	Buoyancy force/torque vector, [N, Nm]
$\mathbf{F}_{\mathrm{e}}$	Excitation force/torque vector, [N, Nm]
$\mathbf{F}_{\mathrm{md}}$	Mean drift force/torque vector, [N, Nm]
$\mathbf{F}_{\mathrm{rad}}$	Radiation force/torque vector, [N, Nm]
$\mathbf{F}_{\nu}$	Viscous force/torque vector, [N, Nm]
F_{x}	Surge force, [N]
F_{z}	Heave force, [N]
p	Pressure, [Pa]
τ_b	Buoyancy torque, [Nm]
τ_c	Control torque, [Nm]
$\tau_{c,b}$	Buoyancy control torque, [Nm]
$\tau_{c,d}$	Linear damping control torque, [Nm]
$\tau_{c,f}$	Friction control torque, [Nm]
τ_{drive}	Pitch torque measured by drive load cell, [Nm]
τ_{free}	Pitch torque measured by free load cell, [Nm]
τ_h	Hydrodynamic torque, [Nm]
τ_{seal}	Driveline torque applied by seal, [Nm]

Geometric Parameters

a	Flap thickness, [m]
c	Foundation height, [m]
c_f	Total OSWEC height, [m]
h	Flap height, [m]
h_m	Distance between hinge and CoM, [m]
h_{SWL}	Height of flap above SWL, [m]
L	Wave tank length, [m]
L_B	Beach length, [m]
w	Flap width, [m]
w_c	Channel width, [m]
Δx_f	Distance from wavemaker to OSWEC, [m]
Δx_1	Distance from wavemaker to first wave gauge, [m]
$\Delta\text{x}_{1,2}$	Distance from first to second wave gauge, [m]
$\Delta\text{x}_{2,3}$	Distance from second to third wave gauge, [m]

Kinematic Parameters

θ	Angular position [rad or °]
$\dot{\theta}$	Angular velocity [rad/s or °/s]
$\ddot{\theta}$	Angular acceleration [rad/s^2 or °/s^2]

Mass Properties

C	Linear flap spring stiffness, [Nm/rad]
I	Flap moment of inertia, [kg m^2]
m_f	Flap mass, [kg]
$\mathbf{M}$	General WEC mass matrix, [kg, kg m^2]

PTO Parameters

$\mathbf{F}_{\text{PTO}}$	PTO force/torque vector, [N, Nm]
κ	PTO buoyancy coefficient
μ	Empirical friction compensation tuning parameter
ν	Nominal PTO damping coefficient, [Nms]
$\hat{\nu}$	Effective PTO damping coefficient, [Nms]
P_a	Absorbed power, [W]
$\hat{P}_a$	Average absorbed power, [W]
τ_{PTO}	PTO torque, [Nm]

SINDy Parameters

λ	Sparsity-producing hyperparameter
λ_{opt}	Optimal sparsity-producing hyperparameter
$\boldsymbol{\Lambda}$	Nonlinear function library
N	Polynomial order of nonlinear functions in $\boldsymbol{\Lambda}$
$\boldsymbol{Y}$	Matrix of state variables for SINDy $(m \times n)$
$\boldsymbol{Z}$	Matrix of modeled variables $(m \times q)$
ξ	Function weight
$\boldsymbol{\Xi}$	Matrix of function weights
$\boldsymbol{\chi}_j$	Column vector of state variable j for all time, $(m \times 1)$
$\boldsymbol{\zeta}_j$	Column vector of modeled variable j for all time, $(m \times 1)$

Time and Frequency Parameters

f	Frequency, [1/s]
f_s	Sampling frequency, [1/s]
t	Time, [s]

Δt	Timestep duration, [s]
ω	Angular frequency, [rad/s]

Water and Wave Parameters

c_g	Wave group velocity, [m/s]
d	Still water depth, [m
g	Gravitational constant, [m/s^2]
H	Wave height, [m]
H_s	Significant wave height, [m]
η_I	Incident wave elevation, [m]
P_{wave}	Wave power per unit length, [W/m]
ρ	Water density, [kg/m^3]
T	Wave period, [s]
T_p	Peak wave period, [s]

ACKNOWLEDGMENTS

This work would not have been possible without my colleagues and community. First, I thank my advisors, Brian and Steve, for guiding this research and motivating me throughout my time here. Your ideas and encouragement were integral in this work and your support and understanding when things were difficult helped me tremendously. I learned so much from both of you, thank you both for being great teachers and patient advisors. Thank you to my committee, Drs. Michelle DiBenedetto and Jim Thomson, for engaging with this work over the years and providing thoughtful feedback. Your questions and ideas inspires multiple avenues of future work I am excited to explore.

I would also like to thank my colleagues at MREL and APL that helped with the development of the OSWEC. Thank you to Ama Hartman for your amazing work in the early development of the device, including design and characterization, and for helping run experiments at NREL. Your company was always appreciated and ideas always helpful. Thank you to Greg Talpey, for helping advance the OSWEC controller and for debugging with me for hours over the phone when I was at NREL. Thank you to Corey Crisp for designing and building the OSWEC electronics and for helping set up the device at NREL. Thank you to Gemma Calandra for your work designing the base structure for the OSWEC. Thank you to Sadie Kass for your work developing the BEM simulations for the OSWEC. Thank you to Cassie Riel and Harlin Wood for their help designing and building the OSWEC. Finally, thank you to Charles Candon, Rebecca Fao, Mark Murphy, Kyle Swartz, and Victor Castillo at the National Renewable Energy Laboratory for your assistance running the experiments in this work and for making me feel welcome in Boulder.

I also want to thank my friends at MREL for their presence and support during my time

here. Thank you to Lindsey Jones, who never failed to make me laugh, even when times were tough. Thank you to Sarah Palmer, for talking to me about WECs for hours and letting me sing to musicals and Taylor Swift while we worked. Thank you to Abby Snortland, Isabel Scherl, and Jayden Wood for making research more fun and enjoyable.

I would not be where I am today without past advisors. Thank you to Dr. Nicholas Boechler, who introduced me to academic research in undergrad. Working in your lab not only helped me believe in myself as an engineer and researcher, but also inspired me to pursue graduate school, which is something I never thought I could do. Thank you to Drs. Chiara Daraio and Domniki Asimaki, who advised me during my time at Caltech. You both showed me what being a female leader in STEM should look like and inspire me to be a better researcher, and for that I am so thankful.

I have been blessed with a community of friends that not only support me, but inspire me to be a better person. I want to thank my best friend, Alyssa Jennings, for being with me through it all since fifth grade. I'm so incredibly grateful for your loyalty and for always bringing color into my life. Thank you to Lucy Li, for always being there for me, whether it's making me laugh, bringing me snacks, listening to my vent, or just being in the same room together. Your support helped me through many trying times and I am so thankful. Thank you to Renee Engleson, Tyler Dyer, Kayla Hogan, Kyle Owsen, Maddy McKeague, Lindsey Boisvin, and Sarah Oliphant, for being the best, and most supportive group of friends. Some of my best memories are with you all, and you all mean the world to me. Thank you to Sage Berglund, Emily Roach, and Joshua Jaworski, for supporting me since undergrad and for always making me feel like I am capable of anything. And thank you to Aidan Hunt and Stacey Dixon, for your steady friendship and support.

Finally, nothing I've done would be possible without my family. I thank my parents, for providing me everything I need to be able to fully explore academics and for unconditional love. Without you, I never would have made it this far and I am so blessed to have you

with me every step of the way. Thank you to my brother and sister for encouraging me, but also reminding me that the world is bigger than research and academics. Thank you to my grandma and Aunty Badra, for showing me what strength and perseverance looks like, especially in the face of adversity. Thank you to my grandpa for always believing in me. And finally, thank you to my partner Conor, who has seen every high and low through this journey and loved me regardless. Thank you for our life together.

The work presented here was supported by the U.S. Department of Energy through the TEAMER program (grant number DE-EE0008955) and ORISE Graduate MHK Fellowship (grant number DESC0014664), Naval Facilities and Engineering Systems Command (grant number N0002421D6400/N0002421F8712), the Alice C. Tyler Perpetual Trust, and the National Science Foundation through the AI Institute in Dynamic Systems (grant number 2112085) and the Graduate Research Fellowship Program (grant number DGE-2140004).

DEDICATION

To my family and friends.

Chapter 1
INTRODUCTION

As the threat of climate change intensifies and resources become more limited, there is an increasing need for green energy generation that does not rely on fossil fuels. Wave energy converters (WECs) offer an innovative carbon-free solution to the growing energy crisis by harnessing useful power from ocean waves [1]. Oceans have a tremendous amount of exploitable power, and WECs have potential for exciting applications beyond grid power generation, including powering remote coastal communities, desalination plants, and ocean observation technology [2].

A promising WEC archetype is the oscillating surge wave energy converter (OSWEC) because these devices can absorb power over a wide range of wave frequencies and rely less on resonance for efficiency [3]. OSWECs are composed of a buoyant flap that primarily harnesses the surge component of waves to oscillate in pitch about a hinge [4]. Additionally, these devices are well suited for shallow ocean environments where the wave direction is consistent and the wave orbitals elongate due to shoaling, amplifying surge motion [5]. OSWECs can also be used for nearshore desalination, as the mechanical pitching motion can be directly translated to a pressure differential for reverse osmosis without losses from intermediate electricity generation [6].

Despite their benefits, there are significant challenges to OSWEC development, as evidenced by the commercial failure of the Aquamarine Oyster, a grid-scale OSWEC installed in Orkney, Scotland, in 2015 [4]. One significant challenge is building accurate models to describe OSWEC behavior in realistic sea states. Although OSWECs operate in a single degree of freedom, they exhibit complex hydrodynamics that can be difficult to describe

mathematically. This is partly because diffraction is one of the dominant forces that drives OSWEC dynamics, making common assumptions such as small-body approximations invalid for these systems [7]. The dynamics are also influenced by drag and viscous forces, resulting in complex fluid-structure interaction that can lead to errors in modeling and predicting energy absorption [8]. Finally, OSWECs are subjected to highly nonlinear events, such as overtopping and slamming [9], that are not easily described using first principles.

Accurate system models are imperative for WEC development, as they are required for state estimation, control, and optimizing performance [10]. Modeling methods often rely on linear potential flow theory and small displacement approximations to describe WEC dynamics. While these methods are computationally efficient, the resulting models can be inaccurate, particularly in realistic seas where large displacements and nonlinear events, such as slamming and overtopping, can occur [11]. Because of these complexities, common design and modeling practices are often not suitable for OSWECs [12]. Simulations using computational fluid dynamics (CFD), such as mesh-based methods or particle-based smoothed particle hydrodynamics (SPH), are better suited to capture nonlinear dynamics, but at a higher computational cost [11]. There have been a number of studies that apply mesh-based CFD [13–15] and SPH [13, 16–21] to model nonlinear OSWEC behavior, including large amplitude rotations [15], slamming [13, 14], and nonlinear power takeoff (PTO) effects [21]. However, these models are often computationally expensive and specific to one device, wave input, and controller, and therefore are not applicable to real-time state prediction or optimal control [11].

Data-driven modeling techniques are a promising option to bridge the gap between model fidelity and computational intensity [22–25]. Instead of solving governing equations, data-driven algorithms derive reduced-order system models directly from data, leading to accurate models with lower computational burden than CFD. Data-driven methods are well suited for WEC modeling because the dynamics are complex and can be difficult to describe mathematically using first principles. There have been a number of studies that apply data-driven methods to WEC modeling, with many focusing on autoregressive models [7, 26–33] or ar-

tificial neural networks [34–36]. However, there are areas of data-driven modeling for WEC dynamics that are yet to be explored, including describing strong nonlinearity and using data-driven models for physical interpretation.

In addition to analytical, numerical, and data-driven techniques, experimental testing is not limited by linear assumptions and can generate rich data sets over a range of wave and control conditions [37]. Because of this, there is a significant body of work focused on experimental testing of scaled OSWEC devices. Early experimental literature largely involves scale models of the Aquamarine Oyster [3, 38, 39]. However, experiments have been conducted with other OSWEC variants to investigate performance [40, 41] and flow field dynamics [42]. Other studies have provided insight into nonlinear slamming [9, 14, 43–45], geometry variation for power shedding [46–49], modular designs [50, 51], and nonlinear control [52, 53]. However, there are limitations to experimental testing, including the presumption that the scaled model exactly replicates the behavior of the full-scale system, i.e., simultaneously achieving geometric, kinematic, and dynamic similarity [37, 54, 55]. If complete similarity is not achieved, results may not accurately describe the full-scale system [37, 56]. For example, drivetrain friction, sensor weight, and viscous effects can disproportionately affect scale models and are referred to as "scale effects". These differences between model- and full-scale devices can limit the value of experimental testing, as predictions of full-scale dynamics, performance, and loads may be inaccurate [37]. Quantifying scale effects and assessing their importance is critical to collecting experimental data that accurately informs full-scale design.

In this work, we explore data-driven and experimental modeling techniques to accurately describe OSWEC behavior and address gaps in research by considering complications such as data quality, nonlinearity, and experimental scale effects. By demonstrating their effectiveness, we aim to inform future modeling practices for WECs by introducing novel ways to study and describe their behavior. The remainder of this work is laid out as follows: Chapter 2 gives background information on recurring topics, including introducing the linear and data-driven modeling tools used in this work. Chapter 3 describes experimental methods

that are used in Chapters 5 and 6. Chapter 4 explores using dynamic mode decomposition (DMD), a data-driven algorithm, to model OSWEC behavior. Specifically, we assess the accuracy of DMD when introducing common modeling barriers such as noisy data, nonlinearity, and irregular wave forcing, and improve the performance of the algorithm using extensions including total-least-squares DMD, time delays, and Optimized DMD. With these extensions, we demonstrate that DMD is a computationally efficient tool that can accurately model OSWEC behavior. The work presented in this chapter was published in the Journal of Renewable and Sustainable Energy in March 2025 [57]. In Chapter 5, we use a nonlinear data-driven algorithm called sparse identification of nonlinear dynamics (SINDy) to model experimental data from a laboratory-scale OSWEC that contains various degrees of nonlinear behavior caused by overtopping and confinement effects. We explore how these nonlinearities affect system loads as a function of device kinematics and demonstrate the ability for SINDy to create accurate and generalizable system models over the full operational range, regardless of dynamic complexity. The work from this chapter is under review to be published for the proceedings of the 16th European Wave and Tidal Energy Conference (EWTEC) and is being prepared for a journal publication. In Chapter 6, we explore mitigation techniques for experimental scale effects including sensor weight and driveline friction. We implement control strategies to emulate additional flap buoyancy that counteracts an effective buoyancy reduction from sensor weight and offset the frictional torque in the driveline. Additionally, we evaluate the role these scale effects have in device kinematics and performance to inform future experimental design. We demonstrate that scale effects can be investigated and addressed using real-time feedback control, rather than costly changes to hardware and setup. The work from this chapter is also being prepared for a journal publication. Finally, Chapter 7 summarizes the conclusions of this work and describes future avenues of research. With this work, we aim to progress WEC development by expanding the existing WEC modeling tools using data-driven algorithms and demonstrating experimental techniques that provide low-cost and efficient methods for accurate scale model testing.

Chapter 2
BACKGROUND

This section describes information that is referenced in the remainder of this work. Specifically, we discuss the linear modeling tool WEC-Sim [58] and the two data-driven algorithms used in Chapters 4 and 5. In the publications based on these chapters, this background information is included in the individual methods, but consolidated here to avoid repetition.

2.1 WEC-Sim

Although this work focuses on data-driven and experimental WEC modeling, we use the linear modeling tool, WEC-Sim, to generate training data and use as a comparison to nonlinear modeling techniques. WEC-Sim [58] is an open-source program implemented in MATLAB that solves the following six degree-of-freedom equation of motion in the time domain:

$$\boldsymbol{M}\ddot{\boldsymbol{y}}(t) = \boldsymbol{F}_{\mathrm{e}}(t) + \boldsymbol{F}_{\mathrm{rad}}(t) + \boldsymbol{F}_{\mathrm{b}}(t) + \boldsymbol{F}_{\nu}(t) + \boldsymbol{F}_{\mathrm{md}}(t) + \boldsymbol{F}_{\mathrm{PTO}}(t), \tag{2.1}$$

where $\boldsymbol{M}$ is the mass matrix of the system, $\boldsymbol{F}_{\mathrm{e}}$ is the excitation force vector, $\boldsymbol{F}_{\mathrm{rad}}$ is the radiation force vector, $\boldsymbol{F}_{\mathrm{b}}$ is the buoyancy force vector, $\boldsymbol{F}_{\nu}$ is damping force vector, $\boldsymbol{F}_{\mathrm{md}}$ is the mean drift force vector, $\boldsymbol{F}_{\mathrm{PTO}}$ is the control (PTO) force vector, and $\boldsymbol{y}$ is the six degree-of-freedom translation vector. WEC-Sim requires a separate BEM solver based on linear potential flow theory to determine the hydrodynamic parameters, including added mass/inertia, radiation damping, and excitation force/torque coefficients, then solves Equation 2.1 in the time domain using a six degree-of-freedom multi-body solver in Simulink (Simscape Multibody). Along with solving for the kinematics, WEC-Sim provides all hydrodynamic, mooring, control, and constraint forces/torques as outputs. Details of WEC-Sim development and validation, including a study on an OSWEC with the same physical and

mass properties used in Chapter 4, can be found in Lawson *et al.* [59] and the source code is open access [58].

We use WEC-Sim in Chapter 4 to produce the data on a full-scale OSWEC modeled after the Aquamarine Oyster [4] and in Chapter 5 to model a laboratory-scale OSWEC operating in a wave tank. Details on the BEM used in the WEC-Sim models are given in Appendix A.

2.2 Data-driven algorithms

As previously discussed, data-driven algorithms create dynamical models based on data rather than solving governing equations derived from physics [60, 61]. There have been a number of studies that have applied data-driven algorithms to modeling WEC dynamics. Folley [62] and Davidson and Costello [11] provide in-depth reviews of data-driven WEC modeling. Here, we review a few common methods. Multiple authors have used system identification techniques such as training autoregressive exogenous input (ARX) and Kolmogorov-Gabor polynomial (KGP) models using experimental [26–30] or numerical [7, 31–33] wave tank data to relate WEC kinematics to wave elevation. Specifically, Giorgi *et al.* [26] built accurate linear (ARX) and nonlinear (KGP) polynomial models that relate float position to wave elevation for a point absorber WEC, where the model weights, order, and structure, were all determined using experimental wave tank data. In addition, Hoff *et al.* [63] used Volterra theory to build accurate models of OSWEC kinematics where they calculated the first-order Volterra kernel using experimental data. These kernels were then used in a numerical simulation to model and predict OSWEC position and velocity, with goals of expanding this work to include higher-order kernels to model nonlinear behavior. Finally, artificial neural networks (ANN) have also shown to be effective in creating black-box data-driven models of WEC behavior [34–36]. In particular, Katsidoniotaki *et al.* [36] used an ANN to model mooring forces on a heaving point absorber in response to extreme waves. The neural network was trained on CFD data and the resulting data-driven model mapped extreme sea states (including significant wave height and peak period) and WEC power take-off characteristics to mooring forces acting on the WEC. Despite the body of prior work for

data-driven WEC modeling, there remain open areas of research, including addressing signal noise present in real systems, incorporating nonlinearity in the system model, describing extreme events, and obtaining physical interpretation from data-driven models. In addition, most data-driven WEC studies focus on point absorbers or oscillating water columns, which can exhibit vastly different hydrodynamic behavior than OSWECs [12].

Although there are a range of data-driven techniques that can model dynamical systems [61], this work focuses on two algorithms: dynamic mode decomposition (DMD) and sparse identification of nonlinear dynamics (SINDy). Both algorithms are well suited for WEC modeling because they are:

- **Equation-free.** Data-driven algorithms are most attractive for systems whose dynamics cannot be easily described using common analytical modeling techniques or first principles, as is the case with nonlinear OSWEC dynamics [12]. Neither DMD nor SINDy require any knowledge of the governing equations to generate accurate system models.

- **Computationally efficient.** Both DMD and SINDy require significantly less computation time than CFD, making them appropriate to use in applications such as real-time state estimation and control [11].

- **Control-oriented.** Both algorithms are fast and accurate enough to be integrated into optimal control schemes, including model predictive control (MPC) [64]. Each algorithm has extensions to address control including dynamic mode decomposition with control (DMDc) [65] and SINDy with control (SINDyc) [66, 67]. Kaiser *et al.* [64] compared the performance of DMDc and SINDyc in a model predictive controller for multiple example nonlinear systems. They found while both algorithms were fast enough to implement in MPC, SINDy showed slightly better prediction and control performance, especially for systems with nonlinear dynamics and noisy signals. MPC is a particularly attractive option for WECs, as optimal control schemes have been

shown to increase the efficiency of different WEC archetypes compared to passive control methods [68, 69]. Traditional linear models may not be accurate enough to be used for real-time optimal control [52, 70], making data-driven algorithms a promising option. Although we do not explore control implementation in this work, the capability of these models to incorporate to future optimal control schemes is important when considering their use in WEC development.

The following sections describe the two algorithms in detail, including additional benefits and limitations of each method.

2.2.1 Dynamic mode decomposition (DMD)

The first data-driven modeling technique we explore is dynamic mode decomposition (DMD) [71–74]. DMD is an equation-free modal decomposition technique that can model and predict the behavior of complex dynamical systems, including fluid flows [71, 75, 76], as well as create robust and computationally efficient models for optimal control [65, 77, 78]. DMD is an established and powerful method in multiple fields (a summary of applications is given by Kutz *et al.* [74]) and has been effective in modeling experimental systems [71, 75].

In its most general form, DMD decomposes spatiotemporal data into spatially coherent dynamic modes with associated complex eigenvalues that determine the frequency and growth/decay rates of the respective mode behavior in time. These modes and eigenvalues can be used to build reduced-order models of dynamical systems and forecast system behavior, all without knowing the physics of the system. DMD eigenvalues and eigenvectors approximate those of the Koopman operator [72, 76, 79–84], which is an infinite-dimensional linear operator that describes the dynamics of a fully nonlinear system. DMD contains properties of both proper orthogonal decomposition (POD) and the Fourier transform, as it identifies spatial modes (characteristic of POD) whose time behavior is characterized by a corresponding frequency (characteristic of the Fourier transform) [65, 85]. DMD has three major applications: identifying and characterizing system behavior, state estimation and/or

future state prediction, and control [74]. Although often applied to spatiotemporal data, DMD also can be used with time-resolved sensor data with minimal modification.

One of the main benefits of DMD is its simplicity and adaptability. DMD is based in straight-forward linear algebra, resulting in low computational cost and interpretable outputs. The algorithm only requires a few lines of code, which means it is not a "black box" approach and also allows for adaptations of the basic algorithm to expand its use and address common issues in practical applications. Schmid [86] summarizes the expansive adaptations that have been developed for DMD. These adaptations, along with its connection to the Koopman operator, are a unique benefit of DMD and make the algorithm applicable to a wider range of systems. For example, DMD can be sensitive to noise, causing high error in systems with low signal-to-noise ratio (SNR). Both Hemati *et al.* [87] and Dawson *et al.* [88] developed extensions to DMD to significantly reduce the effect noise has on DMD models with a minimal increase in computation efforts (further discussed in Section 4.3.3). Because data from WECs can range in quality and complexity, these extensions can drastically increase the applicability of DMD. Brief derivations of DMD and relevant extensions are provided in Section 4.3.3, with detailed derivations given in Appendix B.

2.2.2 Sparse identification of nonlinear dynamics (SINDy)

The second data-driven algorithm we explore is sparse identification of nonlinear dynamics (SINDy), which utilizes sparse regression techniques to build parsimonious reduced-order models of dynamical systems [89]. Specifically, SINDy identifies dominant dynamics using a sparse combination of terms from a library of nonlinear functions created from time series measurement data. The result is an analytical, nonlinear expression relating the modeled and state variables. SINDy is parsimonious, in that it uses a sparsity-promoting hyperparameter to find the the minimum number of terms necessary to describe the observed data. Because of this, models discovered by SINDy tend to be more interpretable and generalizable, and less prone to overfitting. This, along with incorporating nonlinearity, makes SINDy an attractive option to model complex OSWEC behavior. Section 5.3 provides a description of

the algorithm and a full derivation is provided in Brunton *et al.* [89].

Despite their different approaches, SINDy and DMD are closely related. SINDy reduces to DMD if no sparsity is applied and only linear terms are included in the function library [64]. However, there are some advantages to each method. For example, SINDy models are better suited for modeling highly nonlinear phenomena, such as overtopping and slamming, and are more robust to sensor noise when compared to DMD [64]. Additionally, SINDy is more likely to result in physically interpretable models than can be used to learn about the underlying system. However, DMD does not require hyperparameters and can sometimes be trained on less data when compared to SINDy, making it a promising tool when available data is limited [64]. DMD also has a more comprehensive network of extensions that can target specific data complexities to expand its use to a wider range of systems [86]. Because of these tradeoffs, in this work we explore both algorithms and their ability to model OSWEC dynamics.

Chapter 3
METHODS

In Chapters 5 and 6, we analyze data from an laboratory-scale oscillating surge wave energy converter operating in a wave tank. Here, we describe the experimental setup, data collection, and analysis methods used those chapters.

3.1 Experimental Setup

There are two components of the experimental setup: the wave tank and the OSWEC device, both of which are described in the following sections.

3.1.1 Testing facility

We conducted experiments in the Sea-Wave Environment Laboratory (SWEL) wave tank at the National Renewable Energy Laboratory (NREL) in Boulder, Colorado. Figure 3.1 shows a diagram of the wave tank with the OSWEC installed. The tank is 14 meters long, 2.5 meters wide and operates at a 1.3 meter water depth (Table 3.1). A paddle wavemaker generates waves on one end of the tank and a parabolic beach dissipates the waves at the opposite end to limit reflections. We also installed three laterally-centered wave gauges along the +x direction to measure wave elevation between the wavemaker and the OSWEC.

3.1.2 OSWEC

The OSWEC is composed of a flap, driveline, and base structure (Figure 3.2). OSWEC dimensions are given in Table 3.1. During testing, it was positioned approximately 6.6 meters from the wavemaker to ensure the waves were fully developed before reaching the flap. We visually aligned the base to be parallel to the wavemaker using markings on the

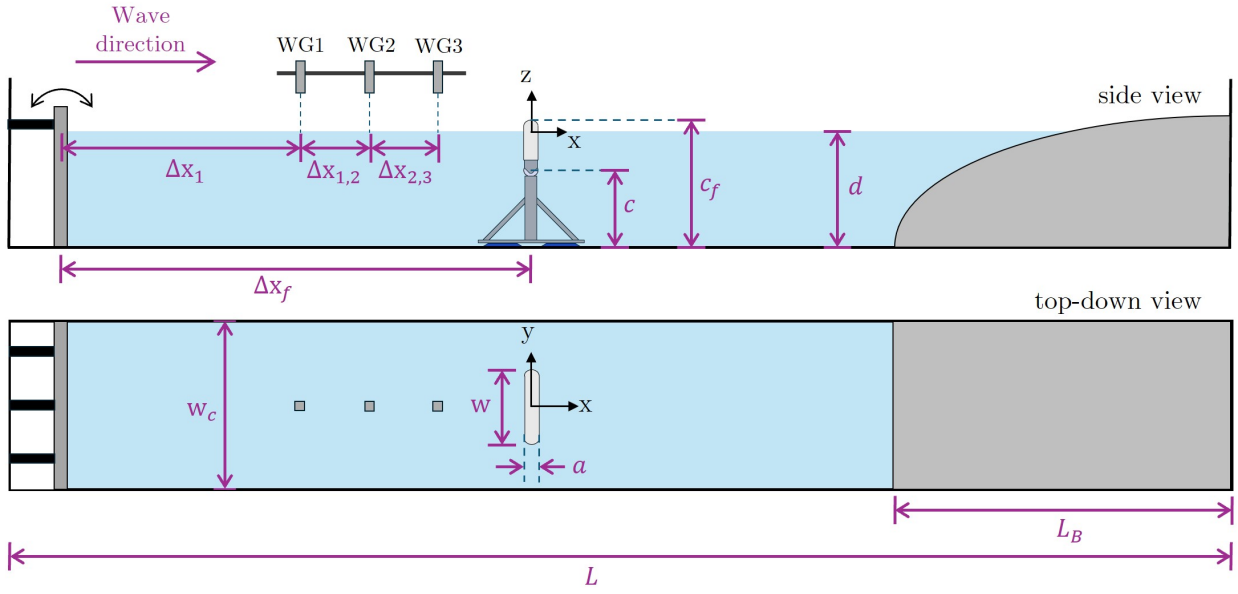

Figure 3.1: Diagram of NREL's SWEL tank (not to scale). Waves propagate from left to right using a paddle wavemaker and are attenuated on the far right side using a parabolic beach. Dimensions are shown in pink and given in Table 3.1.

tank floor. Once the OSWEC was installed, the base structure was secured to the tank floor by a pair of vacuum plates and remained in the same position for all tests presented in each chapter.

The flap (85 cm wide, 48 cm tall, and 14 cm thick) pierces the water surface by 12 cm in the vertical position and occupies 34% of the tank width. The flap is constructed of a low-density polyethylene case with an aluminum frame filled with foam to generate buoyancy. There is an array of fifteen pressure sensors (Keller PR25Y) installed in the flap to measure fluid pressure. While these measurements are not reported in this work, the sensors and their cables displace more than 20% of the interior volume, which decreases the flap buoyancy and affects device behavior and performance (Sections 5.4.1 and 6.3.1).

The driveline (Figure 3.3) is contained in a custom aluminum housing and includes a motor (Parker MPP092) and gearbox (Parker PS90) assembly, which we use to emulate a

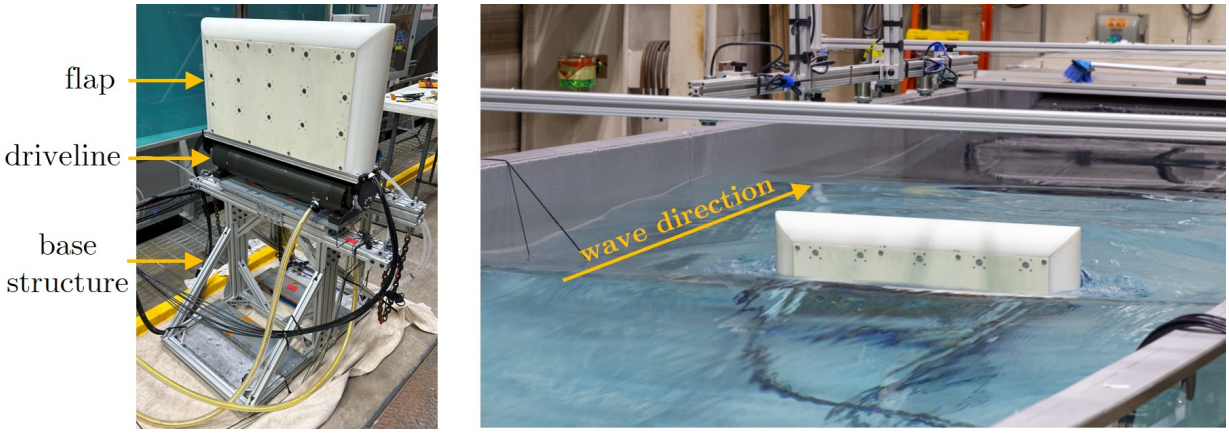

Figure 3.2: [Left] Laboratory-scale OSWEC including flap, driveline, and base structure. The flap is hinged at the driveline, such that it oscillates in response to incident waves or motor torque. [Right] OSWEC operating in SWEL wave tank at the National Renewable Energy Laboratory in Boulder, CO. The flap is surface piercing and waves propagate from bottom left to top right.

power takeoff (PTO) by applying a variable control torque about the hinge via an electrical drive (Compax3 T10). We refer to the end of the driveline connected to the motor as the "drive" side and the end not connected to the motor as the "free" side (left and right side of Figure 3.3, respectively). We measure flap position with an encoder on the gearbox output shaft with a resolution of 65536 counts/revolution. There are waterproof seal and bearing assemblies on both ends of the driveline. The seals apply considerable friction in the driveline (Section 6.3.1), but are necessary to prevent water ingress to the bearings, gearbox, and generator.

The flap and driveline are connected by two 6-axis reaction load cells (ATI Mini58) that measure the torque imposed by the driveline and forces imposed by the foundation. Following the driveline convention (Figure 3.3), we refer to these load cells as the "drive" and "free" load cells. We consider surge and heave force (F_x and F_z, respectively) as the sum of forces from both load cells, such that:

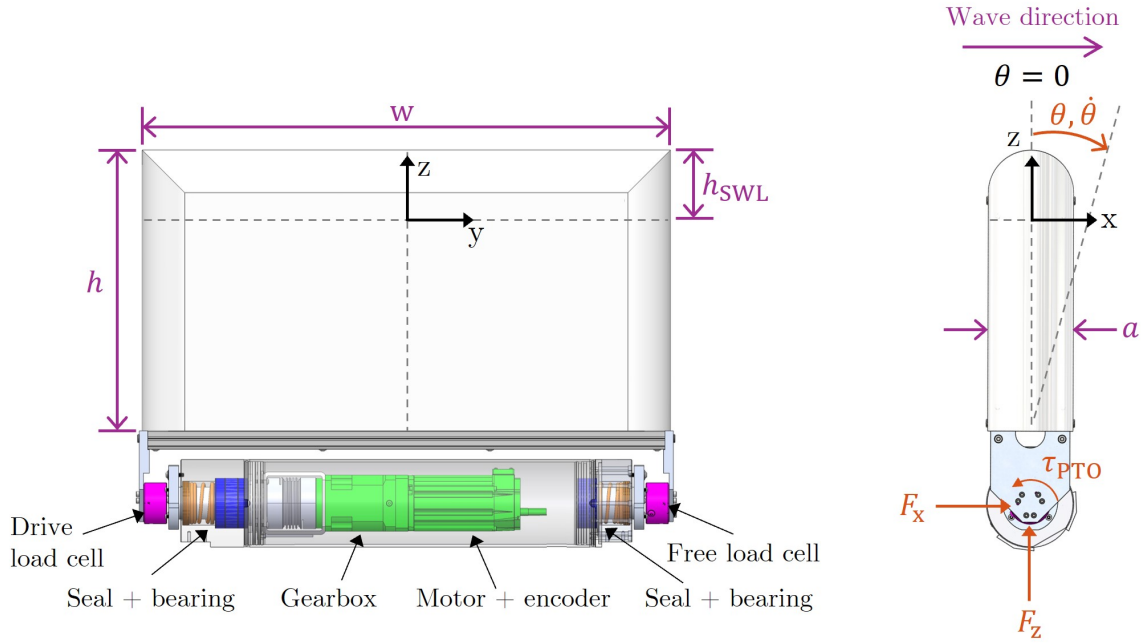

Figure 3.3: [Left] Front and [right] side profile of OSWEC flap and driveline. The driveline consists of two load cells (pink) on either end of the flap to measure control and foundational loads, two sets of seals and bearings (orange and blue, respectively) to waterproof the housing, and a motor/gearbox/encoder assembly (green) to emulate a power take-off and measure flap position. The flap oscillates about the y-axis with the upright position defined as $\theta = 0$. Measured quantities are shown in orange and include flap position (θ), velocity ($\dot{\theta}$), surge force (F_x), heave force (F_z), and pitch torque (τ_{PTO}). Dimensions are shown in pink and documented in Table 3.1.

$$\begin{aligned} F_x &= F_{x,\,\text{drive}} + F_{x,\,\text{free}}, \\ F_z &= F_{z,\,\text{drive}} + F_{z,\,\text{free}}, \end{aligned} \tag{3.1}$$

where $F_{i,\text{drive}}$ and $F_{i,\text{free}}$ refers to the force measured by the drive and free load cell, respectively, in the global reference frame defined in Figure 3.1 (details on load cell coordinate transformations are provided in Appendix C). In addition to surge and heave force, we also

consider the pitch torque about the hinge. Because the flap is free to move in pitch, the load cells only measure torques imposed by the driveline. Specifically, the drive load cell measures the sum of the control torque from the motor/gearbox, τ_c, and torque from the seals and bearings acting on the drive shaft, $\tau_{\text{seal,drive}}$, and the free load cell only measures torques from the seal and bearings on the free end of the driveline. We describe the sum of these load cell measurements as the total "PTO" torque of the system, such that:

$$\tau_{\text{PTO}} = \tau_{\text{drive}} + \tau_{\text{free}}, \tag{3.2}$$

where τ_{drive} and τ_{free} are the pitch torque measurements from the drive and free load cell in the global reference frame, respectively, and τ_{PTO} is the total torque acting on the flap imposed by the driveline.

3.2 Data collection and processing

We used a National Instruments data-acquisition (DAQ) system (PCI-6255) to collect time-synced data of all OSWEC sensors sampled at 1 kHz. After collecting position data from the encoder, we calculate velocity and acceleration using numerical differentiation (details in Appendix C). For load cell measurements, we first identify any tests that exceed the calibration limit of 60 Nm to exclude them from further analysis. Because the torque cell overload limit is 3× the calibrated range, exceeding this limit does not compromise the torque cell functionality. Then, we transform the data to the global reference frame using the position measurements. To limit signal noise, we apply a low-pass filter to all time series. Finally, once all signals are processed, we cut the time series of all relevant measurements to a shorter duration that is used for analysis. We do this to avoid transient behavior due to wavemaker ramp-up. First, we find the time at which the flap moves 0.1°, signaling the initial wave interaction with the flap. For the data presented in Chapter 5, we define the start of the trimmed time series to be 13 seconds after the first identified flap motion and include 12 total periods of oscillations. For Chapter 6, we define the start of the trimmed time series to

Table 3.1: Experimental OSWEC and wave tank parameters

Symbol	Quantity	Value [m]
a	Flap thickness	0.14
c	Driveline axis height	0.86
c_f	Total OSWEC height (including base structure)	1.42
d	Water depth	1.3
h	Flap height	0.48
h_{SWL}	Exposed flap height	0.12
L	Total tank length (including beach)	14
L_b	Beach length	4.73
w	Flap width	0.85
w_c	Channel width	2.5
$\Delta \mathrm{x}_f$	Distance between OSWEC and wavemaker	6.62
$\Delta \mathrm{x}_1$	Distance between wavemaker and first wave gauge	4.41
$\Delta \mathrm{x}_{1,2}$	Distance between first and second wave gauge	0.604
$\Delta \mathrm{x}_{2,3}$	Distance between second and third wave gauge	0.596

be 22 seconds after the first flap motion and include 20 total periods of oscillations. Details on post-processing procedures are given in Appendix C.

3.3 Control

The system was controlled using MATLAB Simulink Desktop Realtime. We used full sensor feedback to integrate real-time position, velocity, and load cell measurements into the control schemes (Section 6.3.1). The control torque is regulated by the drive in response to an analog current output from the DAQ. This current is linearly related to the control torque by the motor's torque constant and the drive's internal logic adjusts the motor winding current to

realize the commanded value. As a consequence of limited deviations from this linearization and drive control tuning, the mechanical torque realized by the motor modestly deviates from the command control torque [90]. Because we measure the actual torque applied to the flap, this deviation does not affect overall results.

For all tests, we employed a linear damping control scheme to emulate wave power absorption by a PTO. The linear damping control torque commanded, $\tau_{c,d}$, takes the form:

$$\tau_{c,d} = -\nu\dot{\theta}, \tag{3.3}$$

where ν is the nominal damping coefficient, and $\dot{\theta}$ is the flap velocity. We varied the nominal damping coefficient, ν, from 0 to 100 Nms and refer to that collection of tests as a "damping sweep".

3.4 Performance

The power absorbed by the OSWEC, P_a, is:

$$P_a = \tau_{\mathrm{PTO}}\dot{\theta}. \tag{3.4}$$

We quantify OSWEC performance by the capture width ratio (CWR):

$$CWR = \frac{P_a}{\mathrm{w}P_{\mathrm{wave}}}, \tag{3.5}$$

where w is the flap width, $P_{\mathrm{wave}} = \frac{1}{8}\rho g H^2 c_g$ is the power of the incident wave per unit width, ρ is the water density, g is the gravitational constant, H is the nominal wave height, and c_g is the group velocity of the wave, which is calculated using the dispersion relation [5].

Chapter 4

DATA-DRIVEN MODELING OF AN OSCILLATING SURGE WAVE ENERGY CONVERTER USING DYNAMIC MODE DECOMPOSITION

This chapter includes work from our paper with the same title published in Journal of Renewable and Sustainable Energy in Spring 2025 [57]. Co-authors on this work include Drs. Brian Polagye and Steve Brunton.

4.1 Motivation

As previously mentioned, dynamic mode decomposition (DMD) is a powerful data-driven algorithm with promising characteristics that make it well suited for modeling WEC dynamics. In this chapter, we explore how well DMD and its extensions describe OSWEC behavior. Our objective is to generate a data-driven model using DMD that can predict OSWEC behavior without incurring an infeasible computational cost or sacrificing accuracy. In particular, we aim for this model to address well known issues in WEC modeling, including noisy signals, nonlinear WEC dynamics, and irregular wave excitation. The remainder of the chapter is laid out as follows: Section 4.2 reviews prior work that uses DMD in ocean engineering problems. Section 4.3 describes the methods we used in this work, as well as the three modeling cases we explore. Section 4.4 presents the results and discussions of DMD performance on OSWEC behavior, with an emphasis on treatment of realistic challenges. Specifically, Section 4.4.1 considers DMD with noisy data, Section 4.4.2 considers DMD when weakly nonlinear dynamics are present, and Section 4.4.3 considers DMD modeling of an OSWEC in an irregular wave field. Finally, Section 4.5 outlines our conclusions and future work.

4.2 Review of prior work

While DMD has been used in a wide range of fields [74], its application in ocean engineering is limited. Serani *et al.* [91] and Diez *et al.* [92] successfully applied DMD to ship maneuvering in both regular and irregular waves. Both studies found that models generated from DMD accurately forecasted ship forces and trajectories for various ship operations in multiple wave periods. In general, the studies found while DMD works well for situations with predominantly monochromatic dynamics, irregular wave inputs and nonlinear dynamics can reduce the accuracy of the DMD forecasts and often benefit from variants of DMD.

In regards to wave energy, Jia *et al.* [93] successfully built a model predictive controller for a nonlinear heaving point absorber in simulation and used a variation of DMD called extended DMD (eDMD) to create the system model. They showed that DMD was capable of building accurate models to predict and optimize the behavior of the WEC, and even outperformed other linear modeling techniques. However, there are some limiting assumptions, including noise-free data, that we explore in this work.

4.3 Methods

An overview of the methods we used for this study are outlined in Figure 4.1. There are four major steps: generate data, preprocess data, run DMD, and evaluate performance.

4.3.1 Generate data

The first step of our workflow is to generate the data that will be used to train the DMD model (Figure 4.1, Block 1). This step includes choosing the data source, which OSWEC parameters to include as system states, and wave conditions. Because there is a limited quantity of benchmark experimental or high-fidelity CFD data for OSWECs in the public domain, we use the mid-fidelity modeling tool, WEC-Sim [58, 59] to generate the training and testing data for this work (Section 2.1). However, this same process can be used with limited variations on data from high-fidelity CFD simulations or experiments. The synthetic data

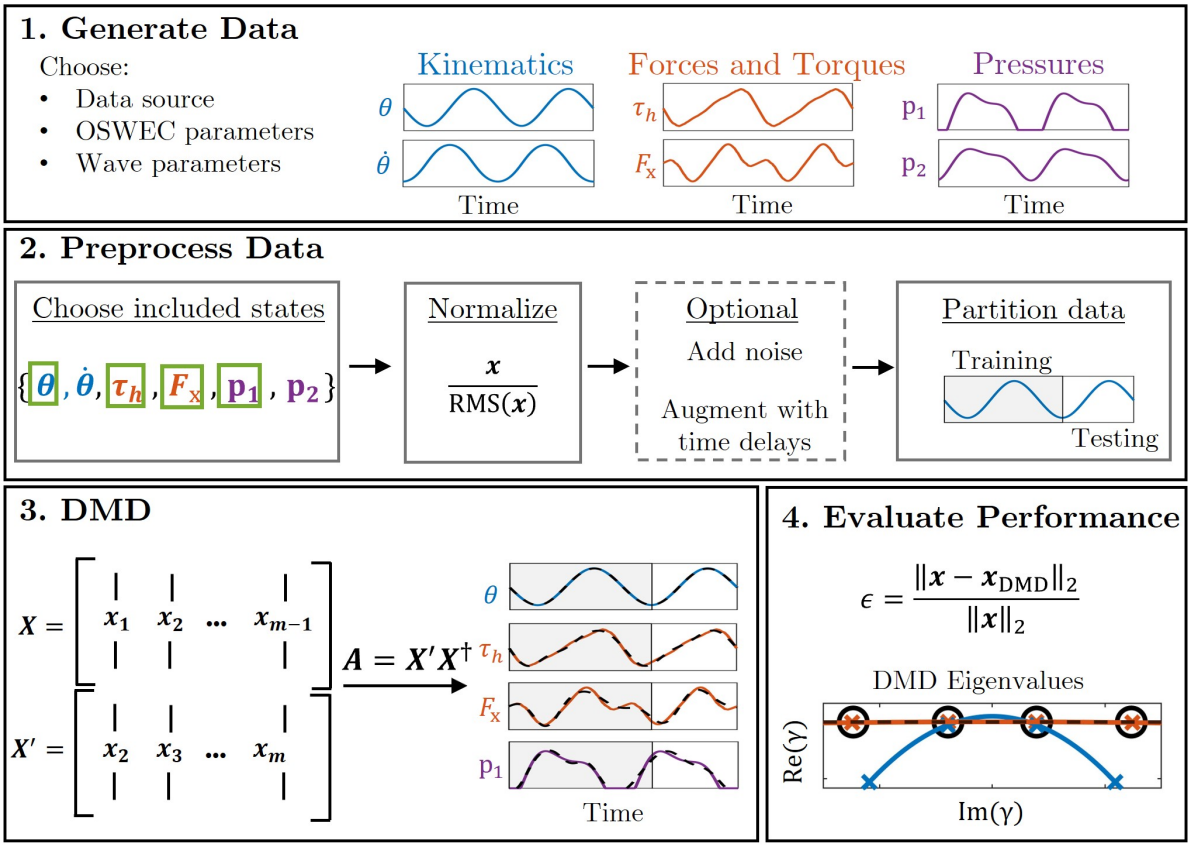

Figure 4.1: Data analysis workflow. The first step is to generate data by choosing the data source (WEC-Sim, CFD, etc.) and wave input. Next, we preprocess data to prepare for DMD algorithm. We then run DMD on training data, and finally we evaluate the performance of the algorithm on both the training and testing data using a nondimensionalized error parameter, ϵ.

approach also allows us to highlight the existing network of DMD extensions in a systematic manner.

OSWEC system

A diagram of the OSWEC system we use in this study is shown in Figure 4.2 and consists of a surface-piercing flap oscillating in the x-z plane with wave input, system states, and

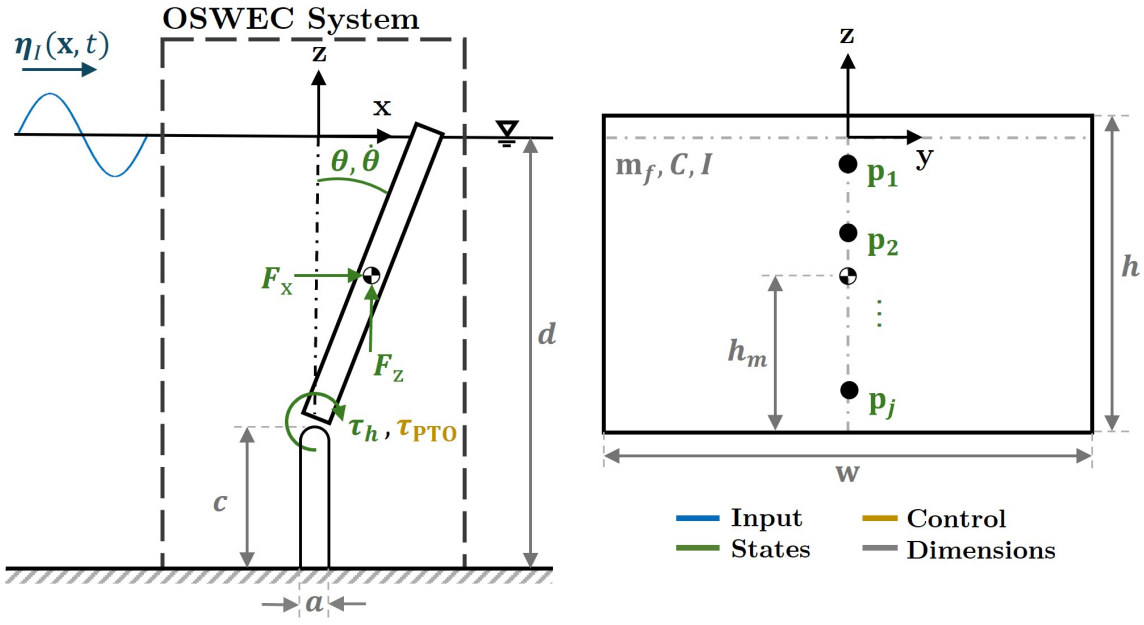

Figure 4.2: Diagram of OSWEC system with incident wave input in blue, system states in green, control parameters in yellow, and dimensions in gray.

control input. For all data in this chapter, we consider a utility-scale OSWEC analogous to the Aquamarine Oyster [4] with a flap width of 18 m operating in a still water depth of 10.9 m. Relevant OSWEC system dimensions and mass properties shown in Figure 4.2 are enumerated in Table 4.1, including moment of inertia about the flap center of mass (I) and hydrostatic stiffness (C) of the flap.

Figure 4.2 also shows the states of this system. These states represent time series measurements that describe system behavior and are known for the purposes of DMD (these form our data matrices in the DMD algorithm). We chose the states of this system to be quantities that can be realistically measured in OSWEC systems and have been measured in previous physical studies [9, 39, 45]. The kinematic states include angular position, $\boldsymbol{\theta}$, and angular velocity, $\dot{\boldsymbol{\theta}}$. Other dynamic states include the total surge and heave force acting on the flap, $\boldsymbol{F}_{\mathrm{x}}$ and $\boldsymbol{F}_{\mathrm{z}}$, respectively. These are quantities calculated from WEC-Sim directly and are

Table 4.1: WEC-Sim OSWEC system properties.

Symbol	Quantity	Value
a	Flap thickness	1.8 m
c	Foundation height	1.5 m
C	Flap hydrodynamic stiffness	6.4 * 10^6 N m
d	Still water depth	10.9 m
h	Flap height	9.4 m
h_m	CoM to hinge distance	5 m
I	Flap moment of inertia	1.85 * 10^6 kg m^2
m_f	Flap mass	127 * 10^3 kg
w	Flap width	18 m

the difference between the inertial force of the flap and the force required to constrain the device in the respective direction. We also consider the hydrodynamic torque acting about the hinge, $\boldsymbol{\tau}_h = I\ddot{\boldsymbol{\theta}} - \boldsymbol{\tau}_{\text{PTO}}$, where $\boldsymbol{\tau}_{\text{PTO}}$ is the control torque applied by the PTO. For this study we consider a linear damping control scheme, where $\boldsymbol{\tau}_{\text{PTO}} = -\nu\dot{\boldsymbol{\theta}}$, and ν is the linear PTO damping coefficient (12000 N m s). Finally we consider pressure measurements, $\mathbf{p}_j$, from an array of sensors aligned with the midline of the flap in the cross-flow direction.

The input to the system is the incident wave field, $\boldsymbol{\eta}_I(\mathbf{x}, t)$. This wave field can be regular or irregular and is a function of both space and time. For the purposes of DMD, the incident wave field is an unknown quantity. This reflects the practical difficulty of obtaining time-resolved wave field measurements, which is a general challenge in WEC modeling and control [94].

WEC-Sim

All data in this chapter is generated using the open-source WEC modeling tool, WEC-Sim [58] (Section 2.1). We use the same OSWEC system that is presented in the Tutorials folder in the WEC-Sim directory, including the BEM files provided in the tutorial. WEC-Sim has a built-in capability to model weakly nonlinear WEC dynamics through the Froude-Krylov (FK) and restoring forces. There are two options for including nonlinearity in these terms. The first option uses the instantaneous body position, rather than the body's neutral position, to calculate the FK force and body restoring force. The second option uses both the instantaneous body position and wave elevation to calculate both forces, increasing the nonlinearity. Lawson *et al.* [95] describes and validates this process. Nonlinearity must be included in the system for WEC-Sim to report pressure measurements. When pressure is reported, WEC-Sim outputs a linear (calculated with no displacement and mean water line) and nonlinear (calculated with the instantaneous displacement and free surface) wave pressure. We utilize the first nonlinearity option with linear wave pressure in Section 4.4.1 and the second nonlinearity option and nonlinear wave pressure in Section 4.4.2. Significant nonlinear events such as overtopping and slamming cannot be modeled with WEC-Sim and are outside the scope of the work presented in this chapter. WEC-Sim also allows for user input for incident wave elevation, which we use in Section 4.4.3 where we input an irregular wave elevation time series collected from field data. Unfortunately, currently WEC-Sim does not report pressure when a time series is used as the wave input.

4.3.2 Preprocess data

The next step is to preprocess the data for the DMD algorithm (Figure 4.1, Block 2), beginning with choosing which measurements to include. For systems with limited sensor measurements, as is the case here, the number of states we include determines the rank of our data matrix (as described in Section 4.3.3), which must be large enough to resolve the system dynamics. Because of this, in cases with more complex dynamics, it may be necessary to

include more of the available states (Figure 4.2) or employ a DMD variant with time delays (as described in Section 4.3.3). The number of states and which states to include may not be possible to know a priori, but is critical to DMD accuracy, so care must be taken to include enough dynamically-rich states to adequately capture the system dynamics.

Next, we normalize each state time series by its root mean square (RMS) value. In general, normalizing data is a common step in DMD studies to promote accuracy and stability [91, 96, 97]. For states with the same units (e.g., $\boldsymbol{F}_{\mathrm{x}}$ and $\boldsymbol{F}_{\mathrm{z}}$), we normalize each time series by the maximum RMS value of all of the states with those units. This normalization step scales the relative importance of states with similar units, and eliminates the significant difference in magnitude between states with different units.

The next steps are only used in specific test cases. The first option is to add artificial noise to the state measurements, which we use in Section 4.4.1 to test robustness. There, we add white, Gaussian noise with a defined signal-to-noise (SNR) ratio to approximate the noise present in physical systems. The second option is augmenting the available states using time delays [81] (Section 4.3.3), which we use for Section 4.4.2 to model nonlinear hydrodynamics.

The final step of preprocessing is partitioning the state time series into training and testing regions. The training data is the input to the DMD algorithm to estimate $\boldsymbol{A}$, and the testing data is reserved to assess how well the reduced order model can predict future state behavior.

4.3.3 Dynamic mode decomposition

After preprocessing, we input the training data into the DMD algorithm (Figure 5.3, Block 3). We introduce DMD and its benefits in Section 2.2.1. Here, we provide brief derivations of DMD and the extensions we use in this work [74, 82, 86], and provide detailed derivations in Appendix B. DMD code is described in Kutz *et al.* [74] and is available at the accompanying website [98].

Exact DMD

Consider a discrete-time dynamical system with state measurements sampled at a time step, Δt. The first step of the DMD algorithm is to arrange the preprocessed training data into data matrices of the form:

$$\boldsymbol{X} = \begin{bmatrix} | & | & & | \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & ... & \boldsymbol{x}_m \\ | & | & & | \end{bmatrix}, \quad \boldsymbol{X}^{'} = \begin{bmatrix} | & | & & | \\ \boldsymbol{x}_2 & \boldsymbol{x}_3 & ... & \boldsymbol{x}_{m+1} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times m}, \tag{4.1}$$

where $\boldsymbol{x}_k$ is a column vector of all the states at time step t_k, and $k = 1, 2, ...m+1$. In these data matrices, each state vector time series corresponds to one *row* of $\boldsymbol{X}$, while each *column* is a snapshot of *all* states at the associated time step, t_k. Note that the two data matrices $\boldsymbol{X}$ and $\boldsymbol{X}^{'}$ are shifted by a single time step. We can then define a best-fit linear operator, $\boldsymbol{A}$, that relates the time-evolution of our two data matrices:

$$\boldsymbol{X}^{'} \approx \boldsymbol{A}\boldsymbol{X}, \tag{4.2}$$

where $\boldsymbol{A}$ relates the states of the OSWEC system from one time step to the next. In other words, $\boldsymbol{A}$ represents the system dynamics, and if we can approximate $\boldsymbol{A}$, we can approximate past and future system behavior. To solve for $\boldsymbol{A}$, we use the pseudoinverse of data matrix $\boldsymbol{X}$, denoted $\boldsymbol{X}^{\dagger}$, such that:

$$\boldsymbol{A} = \boldsymbol{X}^{'}\boldsymbol{X}^{\dagger}. \tag{4.3}$$

Because data matrices are often large, the singular value decomposition (SVD) is commonly used to calculate $\boldsymbol{X}^{\dagger}$, which allows for better scaling and numerical conditioning [74], and also allows for the choice of rank (i.e. how many singular values/SVD modes to include) when calculating $\boldsymbol{A}$ (Appendix B).

The eigenvectors of $\boldsymbol{A}$ are the DMD modes, $\boldsymbol{\phi}_j$, with corresponding discrete-time eigenvalues λ_j that represent their growth/decay rates and frequencies. Using the DMD eigenvectors and eigenvalues, we can describe the system dynamics:

$$\boldsymbol{x}(t) \approx \sum_{j=1}^{r} \boldsymbol{\phi_j} \exp(\gamma_j t) b_j = \boldsymbol{\Phi} \exp(\boldsymbol{\Gamma} t)\boldsymbol{b}, \tag{4.4}$$

where r is the SVD rank, the columns of $\boldsymbol{\Phi}$ contain the DMD modes $\boldsymbol{\phi}_j$, $\gamma_j = \ln(\lambda_j)/\Delta t$ are the complex continuous-time DMD eigenvalues, and $\boldsymbol{b} = \boldsymbol{\Phi}^\dagger \boldsymbol{x}_1$ are the DMD mode amplitudes with $\boldsymbol{x}_1$ being the states at the first time step, t_1. Note that $\boldsymbol{x}(t)$ contains the time series for all state variables. With Equation 4.4, we can now describe and predict system state dynamics using only snapshot data of our system. This approach is referred to as "exact" DMD and, while suitable for some dynamical systems, has limitations. For example, noise, nonlinearity, and sampling errors all can reduce the accuracy of DMD [99]. Even for a linear system with clean data, if the system is only partially measured (i.e., additional states are required to resolve the system dynamics), or there is a limited amount of training data, DMD may lead to inaccurate models. These limitations motivate a number of extensions to the exact DMD algorithm that can improve model accuracy.

Total-least-squares DMD

One of exact DMD's limitations is its sensitivity to noise [100, 101]. In particular, noise biases eigenvalues in a way that can cause an artificial exponential decay in DMD dynamics [88, 101]. These effects cannot be counteracted by increasing the duration of training data or sampling frequency [87]. To remedy this, Hemati *et al.* [87] developed a modification to the DMD algorithm called total-least-squares DMD (TLS DMD) that uses a total-least-squares approach to solve the regression problem laid out in Equation 4.2, rather than the traditional least-squares method [87]. By doing this, TLS DMD solves for $\boldsymbol{A}$ assuming noise is present in both data matrices, rather than just $\boldsymbol{X}$, as is the case with exact DMD [87]. The mathematical derivation of the TLS DMD algorithm is described in Appendix B.2 and we use this variation in Section 4.4.1. Code to implement this algorithm is described in Kutz *et al.* [74, 98].

Time delays

Another variation of DMD involves the use of "time delays" (i.e., time delay coordinates or Hankel matrices), which augments the data matrices to include time-shifted copies of state

measurements. There is a rich history of using time delays and Hankel matrices in system identification techniques, including the eigensystem realization algorithm (ERA) [102]. There are two ways that time delays can improve DMD performance. The first, which was discovered by Tu [73], is that time-shifted matrices provide additional phase information that is critical to describing oscillatory dynamics. For example, DMD cannot model a standing wave without the addition of time-shifted data [73]. Second, including time delays increases the number of rows (and therefore the effective number of states) in $\boldsymbol{X}$, therefore increasing the number of SVD modes one can include when calculating $\boldsymbol{A}$. For systems with limited measurements, as is the case for this work, adding time delays increases the rank of data matrix $\boldsymbol{X}$, which can be necessary to resolve and recreate complicated dynamics. While the addition of time delays increases the size of the data matrices, because this system (and likely other WEC systems) has a relatively small number of sensor measurements, the additional cost of calculating linear operator $\boldsymbol{A}$ using the SVD is insignificant. Details on using time delays are provided in Appendix B.3 and we use this variation in Section 4.4.2.

Optimized DMD

The last variation of DMD we employ is optimized DMD (optDMD), as described in Askham and Kutz [103]. Instead of solving a regression problem for linear operator $\boldsymbol{A}$ using two time-shifted data matrices as shown in Equation 4.2 (i.e., exact DMD), optDMD aims to fit the data matrix $\boldsymbol{X}$ directly to a series of exponential functions, $\exp(\gamma_j t)$, where γ are the continuous-time DMD eigenvalues. The algorithm then uses variable projection techniques to efficiently solve the resulting nonlinear least squares problem and determine DMD eigenvalues, γ_j, eigenvectors, $\boldsymbol{\phi}_j$, and weights, b_j, simultaneously. One main difference between optDMD and exact DMD is optDMD simultaneously optimizes DMD modes and amplitudes using the full data set, $\boldsymbol{X}$, rather than finding the DMD modes, and then solving for DMD mode amplitudes using only the first data snapshot like exact DMD (Equation 4.4). As a result, optDMD provides a model that can handle non-ideal data and has less bias for noisy inputs. This algorithm also allows for the addition of constraints on DMD eigenvalues that

restrict the real part of the continuous-time eigenvalues to be approximately zero, which limits the artificial temporal decay and promotes stability in the DMD reconstruction. We use optDMD in Section 4.4.3 to model OSWEC behavior in response to irregular waves and restrict the real part of the eigenvalue to be between -0.001 and 0, to prevent spurious decay or growth in the DMD modes. We provide a brief derivation of optDMD in Appendix B.4, and refer the reader to Askham and Kutz [103] for a full derivation, as well as open-source code.

4.3.4 Evaluate performance

The last step in the workflow (Figure 4.1, Block 4) is to evaluate the performance of the DMD model. For each case, the DMD model is initialized using the first time step of the training data, then propagated through both the training and testing region using Equation 4.4. We evaluate the accuracy with which the DMD model describes the training data (hindcasting) and predicts the testing data (forecasting) using a normalized error parameter ϵ, defined as:

$$\epsilon = \frac{||\boldsymbol{\chi} - \boldsymbol{\chi}_{\mathrm{DMD}}||_2}{||\boldsymbol{\chi}||_2}, \tag{4.5}$$

where $\boldsymbol{\chi}$ is an arbitrary state variable and $\boldsymbol{\chi}_{\mathrm{DMD}}$ is the corresponding DMD fit. This error parameter can be applied to the training or testing region.

4.3.5 Test cases

We consider three cases to evaluate how well DMD can model realistic OSWEC behavior. Analogous considerations apply to other WEC archetypes. The relevant parameters used in each case are outlined in Table 4.2.

The first case evaluates how well DMD can model OSWEC dynamics when noise is present in the training data. Because any real WEC system will have some level of sensor noise, it is critical that our modeling method can account for that. For this case, we utilize the first nonlinearity option in WEC-Sim (weakly nonlinear restoring and Froude-Krylov force based on flap position) to generate the training data and a polychromatic wave input composed

of two wave components. The first wave component emulates a "swell" wave, with a longer period and larger wave height ($T_{\text{swell}} = 8$ s, $H_{\text{swell}} = 0.75$ m) while the second emulates a "wind" wave with a shorter period and smaller wave height ($T_{\text{wind}} = 2.6$ s, $H_{\text{wind}} = 0.3$ m), as outlined in Table 4.2. Under these conditions, the maximum flap oscillation amplitude is about 6 degrees, resulting in effectively linear dynamics. We add Gaussian, white noise to the data after normalization (Figure 4.1, Block 2) at different signal-to-noise ratios (SNRs) and evaluate the accuracy of exact DMD and total-least-squares DMD, given this input. We use 32 seconds of training data ($4T_{\text{swell}}$), and consider a testing region of the same length. Finally, we include states $\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$, $\boldsymbol{\tau}_h$, and three linear pressure sensors (Table 4.2).

The second case evaluates how well DMD, a linear algorithm, can model weakly nonlinear dynamics. We use the second nonlinearity option in WEC-Sim (restoring and Froude-Krylov force based on flap position and water surface elevation) to generate the training data and evaluate the accuracy of exact DMD with and without time delays. For this case, we consider a significantly larger wave height than Case 1 and the same wave period (Table 4.2), which results in a maximum flap oscillation amplitude of about 23 degrees. We use 10 seconds of training data and consider a 30 second testing region. We include states $\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$, $\boldsymbol{\tau}_h$, $\boldsymbol{F}_{\text{x}}$, and three nonlinear pressure sensors. We include surge force as a state variable because it exhibits interesting nonlinear behavior that could be challenging to model.

For the third case, we evaluate how well DMD can model OSWEC behavior in response to an irregular wave forcing. We use a wave elevation time series collected from field data as input into WEC-Sim and evaluate the linear response of the OSWEC. Unlike the other cases, we calculate the power spectra of our state variables over time and use DMD to model the temporal behavior of the power spectra, rather than the time evolution of the state variables. We expect the resulting dynamics to be more complex and high-dimensional than the previous two cases, which can result in inaccurate results from exact DMD, especially with a limited number of system states (Section 4.3.3). Because of this, we compare the performance of exact DMD to optDMD. With optDMD, we restrict the real part of the eigenvalue to be between -0.001 and 0 to prevent spurious decay or growth in the DMD modes.

Table 4.2: Summary of data parameters and algorithms used for the three cases. DMD refers to the exact DMD algorithm and serves to contrast with variants including total-least-squares DMD (TLS DMD), DMD with time delay (DMD+TD), and optimized DMD (optDMD).

	Case 1	**Case 2**	**Case 3**
Condition	Measurement noise	Nonlinearity	Irregular waves
Wave type	Polychromatic	Monochromatic	Irregular
Wave parameters	$H_{\text{swell}} = 0.75$ m, $H_{\text{wind}} = 0.3$ m $T_{\text{swell}} = 8$ s, $T_{\text{wind}} = 2.55$ s	H = 2 m T = 8 s	$H_s = 0.9$ m $T_p = 5.8$ s
Algorithms	exact DMD TLS DMD	exact DMD DMD+TD	exact DMD optDMD
Training time	32 s ($4T_{\text{swell}}$)	10 s	276 s ($35T_p$)
Testing time	32 s	30 s	102 s
States	$\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$, $\boldsymbol{\tau}_h$, $\mathbf{p}_1 - \mathbf{p}_3$	$\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$, $\boldsymbol{\tau}_h$, $\boldsymbol{F}_{\text{x}}$, $\mathbf{p}_1 - \mathbf{p}_3$	$\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$, $\boldsymbol{\tau}_h$, $\boldsymbol{F}_{\text{x}}$

In addition, because of the increased complexity we use a significantly longer training time of 276 seconds ($35T_p$), and reserve the remainder of the time series (just over 100 seconds) as testing data (Table 4.2). Finally, we include only four states: $\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$, $\boldsymbol{\tau}_h$, and $\boldsymbol{F}_{\text{x}}$. Because WEC-Sim does not currently report pressure outputs for tests with a user-defined wave elevation, pressure measurements are not included as state variables.

In summary, we chose these three cases as examples not only because they specifically address common conditions for WEC modeling and control, but also because these are cases where we expect the exact DMD algorithm to perform poorly. We then demonstrate that the different extensions of DMD can overcome each of these complications (i.e., noise, nonlinearity, irregular wave forcing). By highlighting these cases, we show that DMD can be an effective tool at modeling realistic WEC dynamics.

4.4 Results

4.4.1 Case 1: Noisy measurements

Figure 4.3 summarizes exact and TLS DMD performance on OSWEC data augmented with white, Gaussian noise (e.g., broadband electromagnetic interference). The top right panel shows normalized error, ϵ, for hydrodynamic torque, $\boldsymbol{\tau}_h$, as a function of SNR. We only report error for $\boldsymbol{\tau}_h$, however we observe these trends for all state variables. Although error rises monotonically with increasing noise for both training and testing data, TLS DMD consistently outperforms exact DMD. For SNR < 40, the testing error for TLS DMD approaches the error for exact DMD, however the source of the errors are different. Time series of hydrodynamic torque, $\boldsymbol{\tau}_h$ with an SNR of 40 and 30 show that exact DMD predictions artificially decay through the test region (Figure 4.3, bottom panels). This is common behavior of DMD models trained on noisy data [101]. However, TLS DMD is captures higher frequency oscillations from wind waves without significant decay. The TLS prediction accuracy does decrease over time, which is most notable in the SNR $= 30$ case, but this is caused by an increasing phase shift rather than an exponential decay like the exact DMD model.

For further insight into the different performance of exact and TLS DMD, we consider the continuous-time DMD eigenvalues, $\gamma_j \in \mathbb{C}$, in the complex plane (Figure 4.4). Because the system is oscillatory and approximately linear, the "true" eigenvalues are purely imaginary conjugate pairs whose components correspond to the two frequencies included in the training data. As a result, we chose to include only four SVD modes in the DMD reconstruction, which results in four eigenvalues for exact and TLS DMD (Figure 4.4.) Bagheri [101] showed that adding noise to a system decreases the real component of the DMD eigenvalues and this effect is amplified for higher harmonics of the eigenvalue frequency. This results in a parabolic form of DMD eigenvalues in the complex plane, with higher curvature associated with higher noise levels. In addition, a negative real part of the DMD eigenvalue results in an exponential decay in the time dynamics, which can be anticipated from Equation 4.4. Figure 4.4 visualizes this pattern for SNR=40 and SNR=30. As expected, we see a

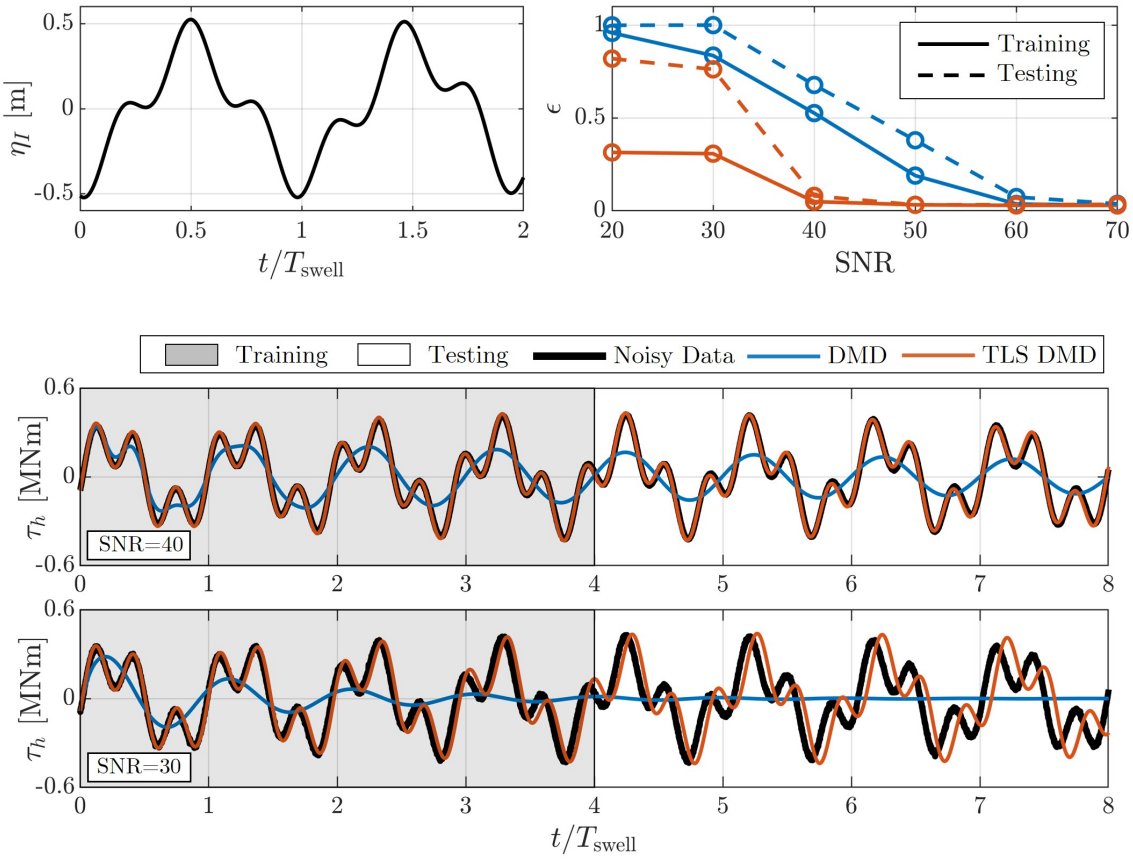

Figure 4.3: [Top left] Undisturbed incident wave field at the flap over time normalized by the wave period of the swell wave component, T_{swell}. [Top right] Normalized error of DMD output, ϵ, for state variable $\boldsymbol{\tau}_h$ as a function of SNR. Lower values of SNR correspond to a noisier signal. Dashed lines represent errors in the training region (hindcasting) and solid lines represent error in the testing region (forecasting). As SNR decreases, TLS DMD (orange) consistently outperforms exact DMD (blue) in both the testing and training region. [Bottom panels] Hydrodynamic torque from WEC-Sim (black), exact DMD (blue), and TLS DMD (orange) for SNR value of 40 (top) and SNR of 30 (bottom). The time series show the artificial decay in the DMD model, but much better accuracy for TLS DMD, consistent with the normalized error values.

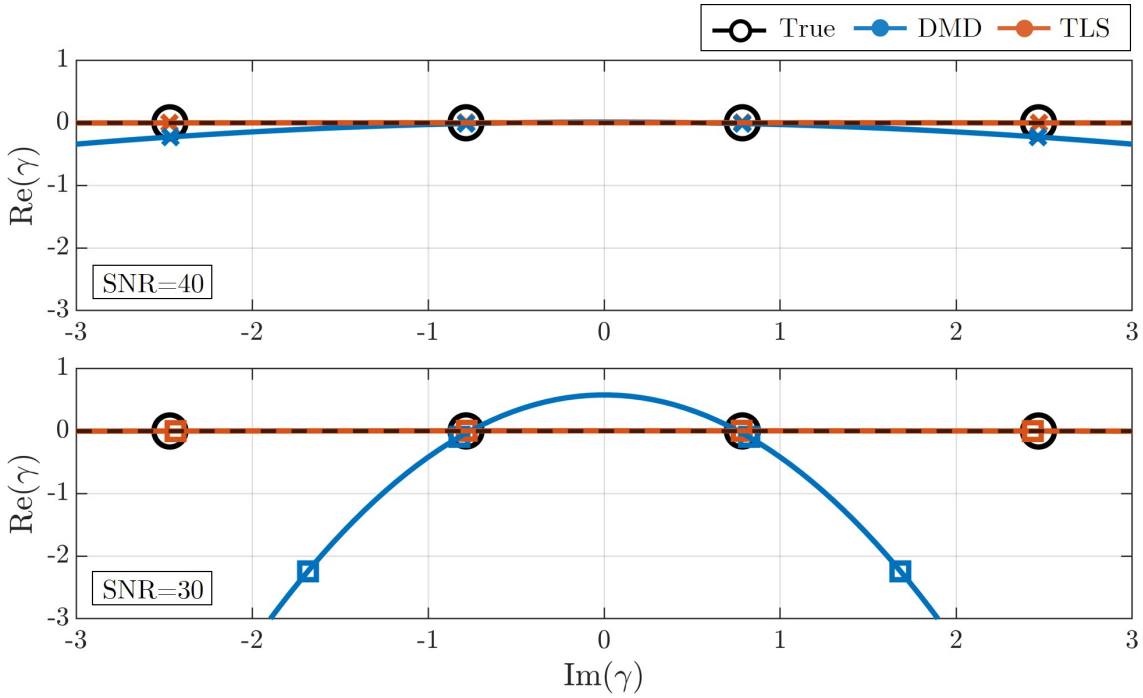

Figure 4.4: Continuous-time DMD eigenvalues, γ, on the complex plane with best-fit parabolic curves. High curvature of the eigenvalues represent a decrease in the real part of the eigenvalues, which in turn results in higher exponential decay in the time series forecasts [101]. Black open circles are the true DMD eigenvalues of the system, the blue symbols represent exact DMD, and the orange symbols represent TLS DMD. The $\times$ markers represent the SNR=40 case (top panel) and the square markers represent the SNR=30 case (bottom panel). As SNR increases, the curvature of the best-fit lines for exact DMD increases significantly, which corresponds to the exponential decay seen in the two time series in Figure 4.3. In contrast, the best-fit curve for the TLS DMD eigenvalues (orange) show almost no curvature, which corresponds to significantly less error in the time series predictions.

significant increase in curvature with noise amplitude (i.e., SNR decrease), but we also see the dramatic reduction in curvature of the TLS DMD eigenvalues for both noise amplitudes. This means TLS DMD has significantly less spurious decay in time and models noisy data better

than exact DMD, which is evident by the time series models in Figure 4.3. There is slight discrepancy in the higher frequency component for the "true" and TLS DMD eigenvalue for the SNR=30 case, which manifests as the TLS DMD model oscillating at a slightly different frequency than the true dynamics, causing a growing phase error in the time domain (Figure 4.3). While error introduced by noise cannot be remedied with increasing amounts of data [87], this discrepancy can be improved by increasing the length of training data. For example, doubling the training time reduces the testing error for TLS DMD at SNR = 30 to from 0.76 to 0.17, while the testing error for exact DMD remains at 1.

These results show that while exact DMD may not be suitable for systems with even low levels of noise, TLS DMD is an effective tool for modeling noisy data and can substantially improve the accuracy of the models in both the training and testing regions. This is promising because it extends the use of DMD to include training on non-ideal experimental or field-scale measurements, with only a slight modification to the original DMD algorithm. For substantially higher noise amplitudes, a different DMD variant, such as optDMD [103], may be more accurate, albeit with a slightly higher computational cost.

4.4.2 Case 2: Nonlinear behavior

Next we explore the potential for DMD, a linear algorithm, to model weakly nonlinear dynamics. Although DMD uses linear least-squares regression, its relation to the Koopman operator means it can model nonlinear systems with an appropriate choice of state variables [72, 76, 82–84]. Here, we summarize the ability of exact and time-delayed DMD to model weakly nonlinear OSWEC dynamics in response to a regular wave.

Figure 4.5a shows the normalized error for the testing period, ϵ_{test}, as a function of system states for DMD with and without a time delay. Some states have near zero error for both DMD models (such as $\mathbf{p}_2$ and $\mathbf{p}_3$), but in every case with nonzero error, DMD with time delay outperforms exact DMD. To contextualize these errors, we consider time series for three states with varying error: angular position ($\boldsymbol{\theta}$), surge force ($\boldsymbol{F}_{\text{x}}$), and the top pressure sensor ($\mathbf{p}_1$). For $\boldsymbol{\theta}$ (Figure 4.5b), both DMD with and without a time delay model the time

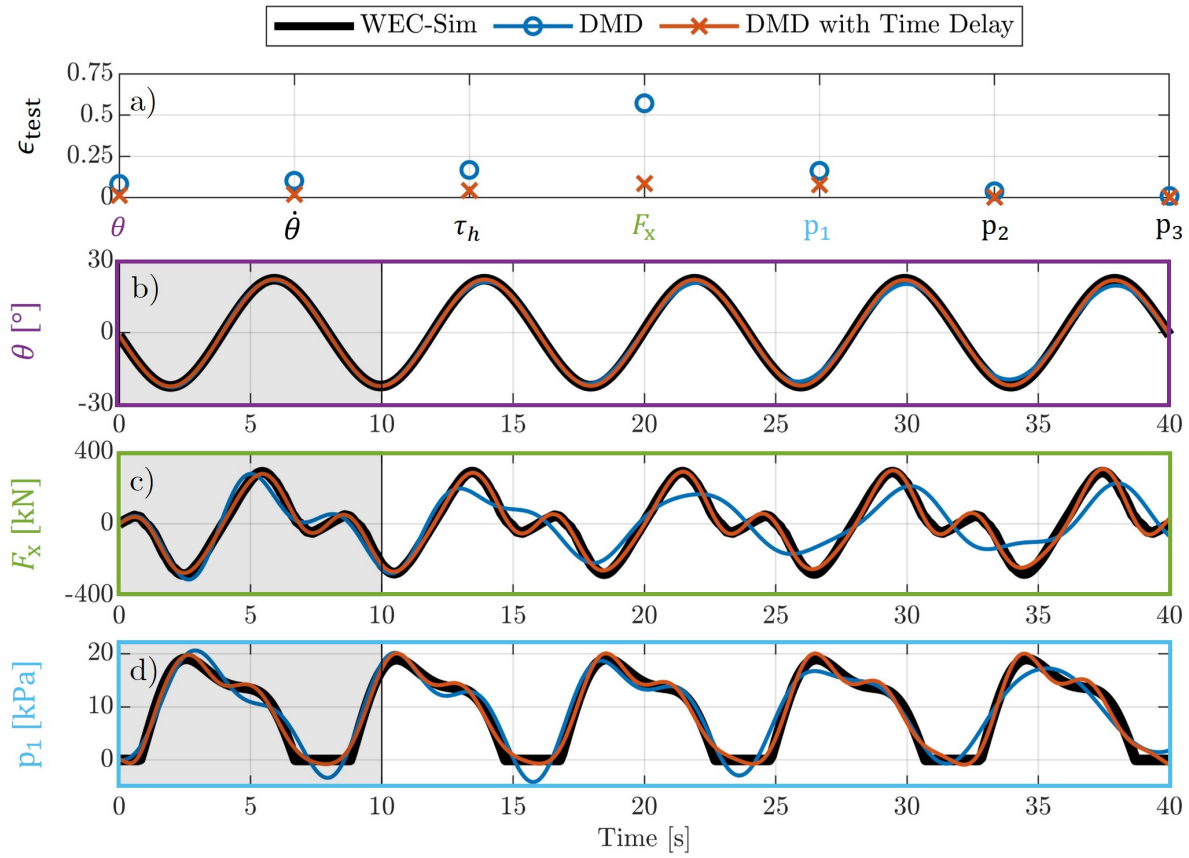

Figure 4.5: (a) Normalized testing error, ϵ_{test}, as a function of system states, showing that DMD with time delay (orange crosses) outperforms exact DMD (blue circles) for all states with nonzero error. Time series of (b) angular rotation, $\boldsymbol{\theta}$, (c) surge force, $\boldsymbol{F}_{\text{x}}$, and (d) the top pressure sensor, $\mathbf{p}_1$. WEC-Sim output is in black, the exact DMD fit is in blue, the TLS DMD output is in orange, and the training region is highlighted in gray. By adding a time delay, the model captures higher-order oscillations with better accuracy for $\boldsymbol{F}_{\text{x}}$ and $\mathbf{p}_1$, and performs better over time in the testing region for all three states shown.

series well, but even in this case where the dynamics appear linear and sinusoidal, there is a slight decay in the exact DMD result without a time delay, meaning we are losing accuracy as we make predictions further into the future. However, adding a time delay eliminates

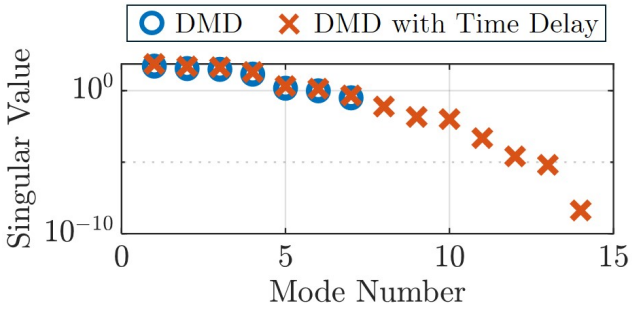

Figure 4.6: Singular values of data matrix $\boldsymbol{X}$ from exact DMD (blue circles) and DMD with one time delay (orange crosses). The singular values for the time delay case show that the true system is a higher rank (contains important information on dynamics in higher modes) than can be captured with exact DMD.

that decay, resulting in near-zero testing error (Figure 4.5a). Surge force, $\boldsymbol{F}_{\mathrm{x}}$, contains more complicated dynamics (Figure 4.5c) and is most improved with a time delay. Pressure, $\mathbf{p}_1$, shows a similar trend (Figure 4.5d). These measurements have no negative values, leading to a piece-wise behavior that can be difficult to model. Both DMD with and without a time delay are able to model the general dynamics, the time delay improves prediction accuracy, particularly later in the testing period.

As previously mentioned in Section 4.3.3, there are two ways time delays improve performance relative to exact DMD. The first is by providing necessary phase information for oscillatory systems [73], and the second is increasing the available rank of the data matrix to better resolve complex dynamics. To illustrate the latter, we consider the singular values of data matrix $\boldsymbol{X}$ with and without a time delay (Figure 4.6). Because we used a full rank SVD, the number of singular values for each model corresponds to the number of rows in the data matrix $\boldsymbol{X}$, which is doubled when adding a time delay. For a system with enough states to fully resolve the dynamics (i.e., the rank of data matrix $\boldsymbol{X}$ is larger than the rank of the underlying dynamics), there is often a steep drop in singular values indicating the number of SVD modes required for accurate modeling. However, the singular values without a time

delay decay slowly and do not have a clear cutoff. Because the additional singular values available with a time delay have a similar magnitude, they provide additional, important information about the dynamics that are otherwise missed. This allows for inclusion of all dynamically-significant SVD modes when calculating $\boldsymbol{A}$ and explains why adding a time delay increases the accuracy of the DMD model for states with meaningful nonlinearity (e.g., $\boldsymbol{F}_{\mathrm{x}}$, $\mathbf{p}_1$).

These results demonstrate that adding a time delay can significantly increase the performance of DMD for weakly nonlinear systems. Even in cases where DMD provides a reasonably good description of the training data, adding a time delay can increase accuracy of higher-order nonlinear patterns well into the testing region. We note that similar improvement to exact DMD may be possible by either increasing the duration of training data or including additional states. For example, when we increase the training duration from 10 to 40 seconds and consider the same amount of testing data, ϵ_{test} for $\boldsymbol{F}_{\mathrm{x}}$ drops from 0.57 to 0.21. However, this the value is still more than double the error from DMD with a time delay trained on only 10 seconds of data ($\epsilon_{\mathrm{test}} = 0.09$). Therefore, time delays offer a computationally efficient way to improve accuracy that can be more effective than increasing the duration of training data.

4.4.3 Case 3: Irregular wave input

For this final case, we investigate how well DMD can model OSWEC behavior in response to real wave elevation data from field measurements. Modeling WEC behavior in irregular waves raises a range of potential complications, as the process can be nonlinear, nonstationary, and stochastic. This means that with limited sensor measurements and a broad frequency range of incident waves, DMD may not be able to consistently model time-domain dynamics of the state variables with a reasonable number of sensors and training time, even using time delays, due to the rank of the dynamics being significantly larger than the rank of state measurements.

Because of this, rather than using DMD to model and predict magnitudes of state vari-

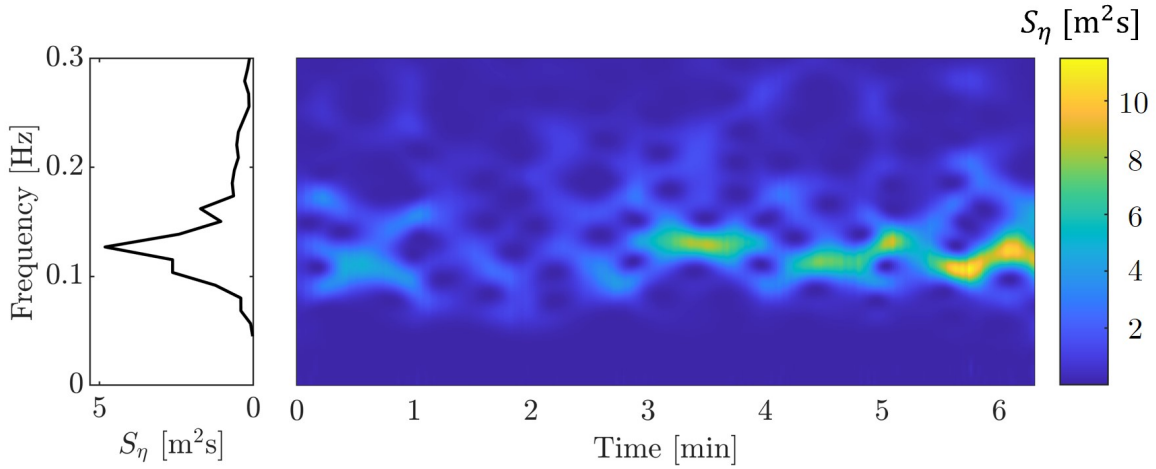

Figure 4.7: [Left] Periodogram and [right] spectrogram of the incident wave elevation time series used for Case 3. The time series was measured from field data [104] and used as input to WEC-Sim. To generate the spectrogram, we use a sliding window short-time Fourier transform on detrended and zero-padded time series data using a Hann filter (window width of 60 seconds and a window time step of 1 second). We note that parameters such as filter type, window width, and window time step are tunable and may affect the accuracy of the DMD model.

ables, as in Cases 1 and 2, we instead use DMD to model and predict the time behavior of the power spectral distributions (PSDs) of the state variables. The power spectrum of arbitrary state $\boldsymbol{\chi}$ over a specified time window with m time steps is described as:

$$\boldsymbol{S}_{\boldsymbol{\chi}} = \frac{2\,|\mathrm{FFT}(\boldsymbol{\chi}^*)|^2}{m\,\mathrm{f}_s}, \tag{4.6}$$

where $\boldsymbol{\chi}^*$ is the windowed and detrended time signal of state $\boldsymbol{\chi}$, and f_s is the sampling frequency. We apply a 60-second sliding window Hann filter to the state variable time series, such that $\boldsymbol{S}_{\boldsymbol{\chi}}$ represents how the frequency content of the state varies in time.

For this case, WEC-Sim is driven by wave elevation time series measured by SWIFT drifters [104] off the east coast of O'ahu. The frequency distribution is relatively narrow,

with a peak period of 7.9 s and a significant wave height of 2.1 m (Figure 4.7, left). The frequency content evolves over time (Figure 4.7, right) which we expect will translate to complex OSWEC dynamics. Our goal is to recreate the spectrograms of state variables in response to this wave field to model and predict their spectro-temporal behavior. Similar to Case 1 and 2, we partition the spectrogram for each state variable into training and testing periods (Table 4.2).

Since DMD was developed for spatio-temporal data, this dimensional expansion to spectro-temporal data is readily accommodated, but requires a minor modification to the data matrices $\boldsymbol{X}$ and $\boldsymbol{X}^{'}$. Specifically, we stack the frequency data of each state in a single column at the corresponding time step. For example, if the state variables are $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$, the data matrix $\boldsymbol{X}$ is arranged as:

$$\boldsymbol{X} = \begin{bmatrix} | & | & & | \\ \boldsymbol{S}_{\boldsymbol{\theta},1} & \boldsymbol{S}_{\boldsymbol{\theta},2} & \dots & \boldsymbol{S}_{\boldsymbol{\theta},m} \\ | & | & & | \\ & & & \\ | & | & & | \\ \boldsymbol{S}_{\dot{\boldsymbol{\theta}},1} & \boldsymbol{S}_{\dot{\boldsymbol{\theta}},2} & \dots & \boldsymbol{S}_{\dot{\boldsymbol{\theta}},m} \\ | & | & & | \end{bmatrix}, \tag{4.7}$$

where $\boldsymbol{S}_{\boldsymbol{\theta},1}$ would correspond to the frequencies of state variable $\boldsymbol{\theta}$ at t_1, $\boldsymbol{S}_{\boldsymbol{\theta},2}$ at t_2, and so on. The rest of the DMD process is unchanged. Here, we contrast the performance of exact DMD with optDMD (Section 4.3.3) and use a reconstruction rank of 17 modes to model the state variable spectrograms. We compare their performance with a normalized error metric, $\bar{\boldsymbol{\epsilon}}$, defined as:

$$\bar{\boldsymbol{\epsilon}} = \frac{\boldsymbol{S}_{\boldsymbol{\chi}} - \boldsymbol{S}_{\boldsymbol{\chi},\mathrm{DMD}}}{\max(\boldsymbol{S}_{\boldsymbol{\chi}})}, \tag{4.8}$$

where $\max(\boldsymbol{S}_{\boldsymbol{\chi}})$ is the maximum value of the true spectrogram of that state. Unlike the error parameter used in the previous two cases, $\bar{\boldsymbol{\epsilon}}$ is a matrix of values the same size as the state spectrograms, not a scalar value.

Figure 4.8 shows the DMD and optDMD models of the spectrograms for state variables

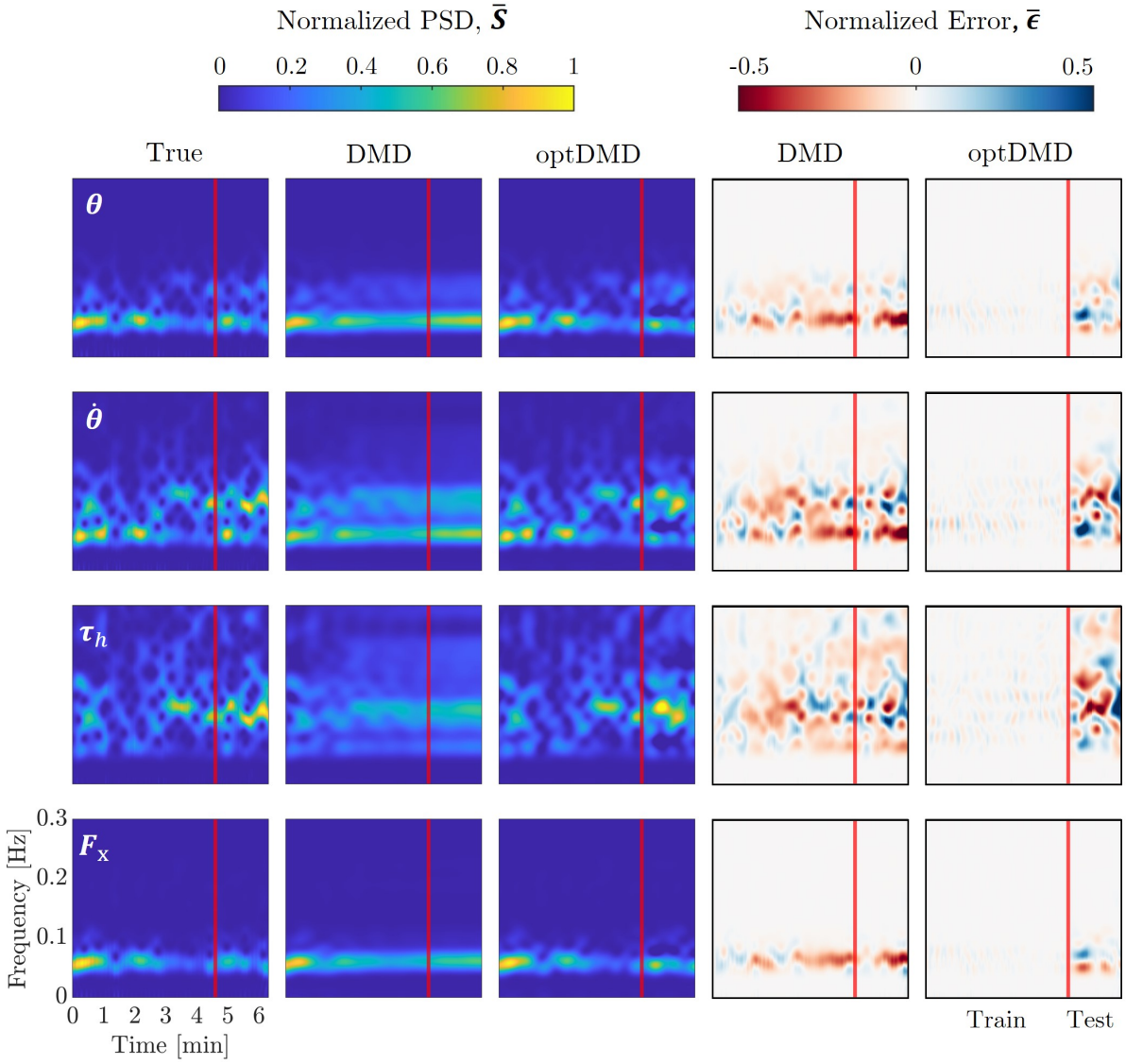

Figure 4.8: Spectrograms of WEC-Sim output ("True") (first column), exact DMD (second column), and optDMD (third column) for state variables $\boldsymbol{\theta}$ (top row), $\dot{\boldsymbol{\theta}}$ (second row), $\boldsymbol{\tau}_h$ (third row), and $\boldsymbol{F}_{\mathrm{x}}$ (bottom row). All spectrograms are normalized by the maximum value of the true spectrogram for the corresponding state variable. The last two columns represent the normalized error for exact DMD (fourth column) and optDMD (fifth column). optDMD outperforms exact DMD in the training region for all state variables, which ends at the red vertical line, while both models have significant error in the testing region.

$\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$, $\boldsymbol{\tau}_h$, and $\boldsymbol{F}_{\mathrm{x}}$. Each spectrogram is normalized by its maximum value. Overall, opt-DMD outperforms exact DMD in the training region. Exact DMD models the beginning of the training data with reasonable accuracy, but quickly degrades after about one minute. In contrast, optDMD models the entire training region accurately, capturing the changing spectral behavior with low error. While neither model can accurately predict dynamics in the testing region, the structure of the two models vary considerably. Exact DMD only shows low-resolution spectral characteristics and does not capture any underlying structure present in the state variables. However, optDMD models some of the general trends and structure present in the testing region. For example, optDMD predicts the spectral behavior well for both angular position, $\boldsymbol{\theta}$, and the surge force, $\boldsymbol{F}_{\mathrm{x}}$. For velocity and hydrodynamic torque, where the spectral behavior is more complex, optDMD identifies energetic peaks at similar frequencies and times to the true dynamics, even with no knowledge of the incident wave field.

One reason optDMD outperforms exact DMD is optDMD solves for the DMD mode amplitudes, b_j, using the full time series data contained in $\boldsymbol{X}$, whereas exact DMD uses only the first snapshot, $\boldsymbol{x}_1$ (Equation 4.4). This results in the exact DMD fit to only match the initial training data, whereas optDMD can accurately model the entire training region. We also enforced the optDMD eigenvalues to have nearly no real component to eliminate any growth or decay in time. This means that optDMD is more robust to non-ideal data (e.g., limited training data, small nonlinearities, or numerical errors) all of which can bias DMD eigenvalues and lead to non-physical growth or decay. This also contributes to the decay of spectral resolution of the exact DMD model in time that is not present with optDMD.

To further evaluate the performance of exact DMD and optDMD, we consider how well these algorithms can model the average power the OSWEC generates over time, or absorbed power. Since the device is operating in irregular seas, the power absorbed can vary significantly in time, making electricity generation and storage difficult. Average power is important to know and predict because it can inform when the device is generating more power than necessary and needs to shed power to accommodate energy storage constraints,

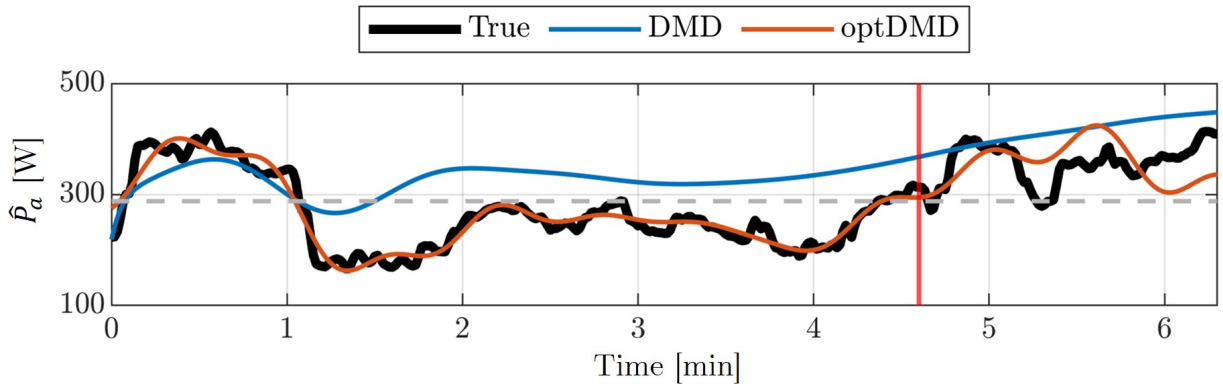

Figure 4.9: Time series plot of average absorbed power, $\hat{P}_a(t)$, with true value in black, the DMD fit in blue, and optDMD in orange. Red line separates training region (left) and testing region (right), and the gray dashed line represents the true average absorbed power over the full time series.

or when energy generation is low and the device can operate at full efficiency. Because we cannot model the actual velocity time series to calculate instantaneous power absorption, we instead look at the average power absorbed by the OSWEC over all frequencies, $\hat{\boldsymbol{P}}_a(t)$. Because the OSWEC model uses a linear PTO, the average power absorbed over a time period t_p to t_q is $\hat{\boldsymbol{P}}_a(t) = (\nu/(t_q - t_p)) \int_{t_p}^{t_q} \left|\dot{\boldsymbol{\theta}}(\tau)\right|^2 d\tau$. By Parseval's theorem, we can approximate $\hat{\boldsymbol{P}}_a(t)$ by taking each "slice" of the $\dot{\boldsymbol{\theta}}$ spectrogram in time and integrating the power spectral density over all frequencies, as:

$$\hat{\boldsymbol{P}}_a(t) = \nu \int_{\mathrm{f}} \boldsymbol{S}_{\dot{\boldsymbol{\theta}}}(\mathrm{f}, t) d\mathrm{f}. \tag{4.9}$$

Figure 4.9 shows $\hat{\boldsymbol{P}}_a(t)$ calculated from the true spectrograms generated from WEC-Sim data as well as both exact DMD and optDMD spectrograms from Figure 4.8. Similar to the previous results, optDMD outperforms exact DMD in modeling the average absorbed power, especially in the training region where optDMD models $\hat{\boldsymbol{P}}_a(t)$ with near perfect accuracy. The exact DMD estimate of $\hat{\boldsymbol{P}}_a$ follows a similar trend as the true value, but does not accurately model its magnitude beyond the first minute of training data. This

suggests that DMD can only identify the low rank dynamics of $\boldsymbol{S}_{\dot{\theta}}$, which is also evident in Figure 4.8. While optDMD matches the general trend of the true values in the testing region, neither algorithm accurately forecasts the time evolution of absorbed power consistently in the testing region.

Even though optDMD shows improvement over exact DMD in modeling and predicting OSWEC behavior in irregular waves, there are limitations to these algorithms that could make them unfit in practice. OptDMD could be helpful for state identification in irregular seas, but is less helpful for state forecasting. We note that there are additional adjustments we can make to optDMD hyperparameters, such as adjusting the initial guess of the eigenvalues or optimization parameters, that could improve its accuracy. However it is possible that this algorithm may not be suitable for real-time state prediction. Extended DMD [105], which was used by Jia *et al.* [93] to build a model predictive controller for a WEC in irregular seas and incorporated wave elevation data directly, could be better fit for this problem and should be explored in future work.

4.5 Conclusions

The work presented in this chapter demonstrates the accuracy with which dynamic mode decomposition can model and predict OSWEC behavior using only physically attainable measurement data and without knowledge of the past or future incident wave field. Specifically, we tested the ability of DMD to model OSWEC behavior when considering three common WEC modeling considerations: noisy signals, nonlinear behavior, and irregular wave input. We showed that for noisy data, exact DMD breaks down with relatively low levels of noise (SNR < 50), but total-least-squares DMD can significantly improve the system model in both the training and testing region with minimal modifications to the original algorithm. Second, we showed that although DMD is a linear algorithm, it can model weakly nonlinear dynamics with reasonable accuracy. When nonlinear dynamics are present, adding even a single time delay can significantly improve DMD accuracy with the same amount of training time by increasing the available rank of the data matrices and providing additional phase

information. Finally, we showed we can describe OSWEC behavior in response to irregular waves by transforming the state variables to the frequency domain and modeling the spectro-temporal behavior of the system states. Optimized DMD can significantly improve accuracy in recreating the spectro-temporal structure of system states, particularly in the training region. Analysis of the irregular wave case also highlights the limitations of a predictive model without knowledge of the future, stochastic wave field.

Overall, we see potential for data-driven algorithms, like DMD, to bridge the gap between the wave energy and data science fields and solve problems encountered in WEC modeling and control. In this study, we highlight the capabilities of DMD to accurately model and predict WEC dynamics in non-ideal conditions, but acknowledge that data from experiments and field deployments will include combinations of the cases presented in this paper along with other complexities not considered. Consequently, future work should apply these methods to experimental or field systems to advance the ultimate objective of using data-driven models to inform model predictive control schemes and optimize power absorption. With these applications in mind, DMD is a promising tool that has significant potential to support the development of the wave energy field.

Chapter 5

DATA-DRIVEN MODELING OF NONLINEAR EVENTS FOR AN OSCILLATING SURGE WAVE ENERGY CONVERTER

The work in this chapter is co-authored by Ama Hartman, Greg Talpey, Corey Crisp, Gemma Calandra, Charles Candon, Rebecca Fao, and Drs. Steve Brunton and Brian Polagye.

5.1 Motivation

In the previous chapter, we demonstrated that DMD and its extensions can be effective in modeling OSWEC behavior. However, all the training data was generated using primarily linear theory and did not include nonlinear phenomena such as large-amplitude motion or overtopping. In this chapter, we explore using data-driven methods to describe a more realistic dataset. To generate this data, we conduct a series of experiments at the National Renewable Energy Laboratory (Section 3.1). Data from these experiments contain a range of nonlinear behavior, including confinement effects and overtopping, that would likely be too complex to describe using DMD with the available sensor measurements. Instead, we use sparse identification of nonlinear dynamics (SINDy [89]) to build parsimonious, reduced-order models of surge and heave forces acting on the foundation of the OSWEC (Figure 3.2) as a function of its kinematics. Our goal is to build accurate models over an entire operation range, including various degrees of overtopping severity. We also aim to identify the role nonlinearity plays in model accuracy, gain physical intuition of the SINDy models, and explore how the reduced-order models generalize. The remainder of the chapter is laid out as follows: Section 5.2 summarizes relevant past works. Section 5.3 provides a derivation for the SINDy algorithm. Section 5.4 describes the methods used in this chapter, including

experimental methods (Section 5.4.1) and workflow (Section 5.4.2). Section 5.5 shows the results of the SINDy algorithm in modeling surge (Section 5.5.1) and heave (Section 5.5.2) forces. Section 5.6 discusses the comparison between the SINDy results and linear models (Section 5.6.1), the physical interpretation of the SINDy models (Section 5.6.2), and the ability for the models to generalize beyond the data they were trained on (Section 5.6.3). Finally, Section 5.7 gives conclusions and future work.

5.2 Review of prior work

SINDy has been used in various scientific applications [106], including a small number of studies from the wave energy field. In previous work, we used SINDy to model large-amplitude, nonlinear WEC kinematics from CFD data, the results of which are presented in Appendix D [107]. We saw significant cubic nonlinearities when describing acceleration that could not be adequately captured by a linear model. Separately, Liu *et al.* [108] used SINDy paired with Gaussian process regression to create a probabilistic model to estimate the wave excitation force of a point absorber WEC in irregular waves. They use SINDy to generate nonlinear models to describe kinematics and excitation force for the Wavestar WEC while considering control inputs. The models were trained and validated on data from WEC-Sim, which as previously discussed is primarily a linear modeling tool. While SINDy was successful in describing wave excitation force, there is potential for SINDy to model more realistic, nonlinear WEC data.

5.3 Background

In this section, we outline the SINDy algorithm; the full derivation and code are available in the original paper [89] and the PySINDy open-source software package [109]. However, the formulation of the algorithm in this work varies slightly from the original in that we do not model a system of differential equations, but rather are relating time-varying system variables directly to each other.

Consider a modeled variable, $\zeta(t)$, that can be described as a function of state variables,

$\chi_i(t), i = 1, 2, ..., n$, such that:

$$\zeta(t) = f(\chi_1(t), \chi_2(t), ..., \chi_n(t), t). \tag{5.1}$$

In many dynamical systems, f is "sparse", which means few terms are required to describe $\zeta(t)$, even for complicated dynamics. SINDy aims to exploit this sparsity to build parsimonious models between state variables, $\chi_i(t)$, and modeled variable, $\zeta(t)$, that are generalizable and interpretable. In other words, SINDy uses sparse regression techniques to find the minimum number of terms in f to accurately describe $\zeta(t)$ using a nonlinear function library built from state variables $\chi_i(t)$. By using as few terms as possible, only the important dynamics are retained, reducing the risk of an overfit model.

SINDy takes in snapshot data and creates data matrices $\boldsymbol{Y}$ and $\boldsymbol{Z}$, whose columns correspond to time series measurements of each state and modeled variable, respectively. As a result, data matrices $\boldsymbol{Y}$ and $\boldsymbol{Z}$ become:

$$\boldsymbol{Y} = \begin{bmatrix} | & | & | & | \\ \boldsymbol{\chi}_1 & \boldsymbol{\chi}_2 & ... & \boldsymbol{\chi}_n \\ | & | & | & | \end{bmatrix} \in \mathbb{R}^{m \times n}, \quad \boldsymbol{Z} = \begin{bmatrix} | & | & | & | \\ \boldsymbol{\zeta}_1 & \boldsymbol{\zeta}_2 & ... & \boldsymbol{\zeta}_q \\ | & | & | & | \end{bmatrix} \in \mathbb{R}^{m \times q}, \tag{5.2}$$

where n and q are the total number of state and modeled variables, respectively, and m is the total number of time steps in each signal. Note that $\boldsymbol{Y}$ is the transpose of the data matrix $\boldsymbol{X}$ used in the previous chapter ($\boldsymbol{Y} = \boldsymbol{X}^T$) and $\boldsymbol{\chi}_i$ and $\boldsymbol{\zeta}_i$ are the ith state and modeled variable time series, respectively. SINDy uses the data matrix $\boldsymbol{Y}$ to build a nonlinear function library, $\boldsymbol{\Lambda}(\boldsymbol{Y})$, whose columns represent different functions of the state variable data and act as potential candidates to describe the modeled variables in $\boldsymbol{Z}$. The library $\boldsymbol{\Lambda}(\boldsymbol{Y})$ can include any function of $\boldsymbol{Y}$, such as polynomial, exponential, hyperbolic, etc. However, in this work we only consider polynomial functions of the state variables in the function library, such that:

$$\boldsymbol{\Lambda}(\boldsymbol{Y}) = \begin{bmatrix} | & | & | & & | \\ 1 & \boldsymbol{Y}^{P_1} & \boldsymbol{Y}^{P_2} & ... & \boldsymbol{Y}^{P_N} \\ | & | & | & & | \end{bmatrix} \in \mathbb{R}^{m \times p}, \tag{5.3}$$

where $\boldsymbol{Y}^{P_j}$ is a set of all polynomial combinations of $\boldsymbol{Y}$ of order j. For example, for the data matrix with two states $\boldsymbol{\chi}_1$ and $\boldsymbol{\chi}_2$, $\boldsymbol{Y}^{P_2} = [\boldsymbol{\chi}_1^2 \;\; \boldsymbol{\chi}_1\boldsymbol{\chi}_2 \;\; \boldsymbol{\chi}_2^2]$.

Using the nonlinear polynomial library of order N, $\boldsymbol{\Lambda}^N(\boldsymbol{Y})$, we can set up a regression problem to relate data matrices $\boldsymbol{Y}$ and $\boldsymbol{Z}$:

$$\boldsymbol{Z} = \boldsymbol{\Lambda}^N(\boldsymbol{Y})\boldsymbol{\Xi}. \tag{5.4}$$

SINDy aims to find function weights $\boldsymbol{\Xi} \in \mathbb{R}^{p\times q}$ that result in the fewest number of active terms in $\boldsymbol{\Lambda}^N(\boldsymbol{Y})$ to accurately describe the modeled variables in $\boldsymbol{Z}$. We use a sequential threshold least-squares algorithm to find each set of function weights $\boldsymbol{\xi}_i$ that minimize the following expression [89, 109]:

$$\arg\min_{\boldsymbol{\xi}_i} \left(\left|\left| \boldsymbol{\zeta}_i - \boldsymbol{\Lambda}^N(\boldsymbol{Y})\boldsymbol{\xi}_i \right|\right|_2 + \lambda \left|\left|\boldsymbol{\xi}_i\right|\right|_1 \right), \tag{5.5}$$

where $\boldsymbol{\zeta}_i$ and $\boldsymbol{\xi}_i$ are the ith column of $\boldsymbol{Z}$ and Ξ, respectfully, and λ is a hyperparameter that enforces sparsity by penalizing the sum of nonzero terms in $\boldsymbol{\xi}_i$. λ is tunable and can be chosen such that parsimony is achieved, i.e., the dynamics are accurately represented, but not overfit. We refer to this optimal value as λ_{opt}.

5.4 Methods

In this section, we describe the methods used in this chapter including details on the experiments we conduct and how we apply the SINDy algorithm to the experimental data.

5.4.1 Experiments

We use experimental data taken from the setup described in Section 3.1 as training data for SINDy. We tested the device in regular waves with a wave height of 0.1 m and a period of 1.5 s. As described in Section 3.3, we employed linear damping control, where the motor applied a resistive torque to the flap proportional to flap velocity (Equation 3.3). Figure 5.2 summarizes the performance of the device (capture width ratio, Equation 3.5) as a function of damping coefficient, ν.

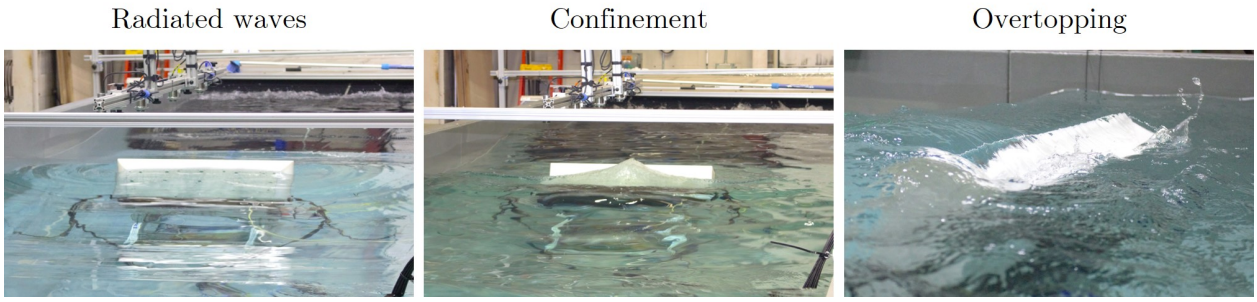

Figure 5.1: Examples of nonlinear experimental phenomena. [Left] Radiated waves around the device. [Middle] Standing wave caused by radiated waves reflecting off the channel walls. [Right] Overtopping caused by limited flap buoyancy and high damping.

Nonlinear phenomena

During testing, we observed nonlinear phenomena that affected flap behavior (Figure 5.1). First, radiated waves from the flap reflected off the wave tank walls in the cross-channel direction, creating a standing wave that peaked at the center of the flap (left and middle panel, Figure 5.1). Additionally, the flap tended to drift away from its vertical position over time, such that the midpoint of the oscillation occurred at increasingly positive flap angles. This flap drift was caused by limited flap buoyancy (further explored in Chapter 6) that did not provide enough restoring torque to return the flap to true neutral position before the next wave crest reached the flap. As a result, waves began to overtop the device for cases with significant flap drift (right panel, Figure 5.1). The severity of the confinement effects, flap drift, and overtopping all increased with higher damping coefficients, ν. Because of this trend, we identify three damping values that result in varying levels of nonlinearity: $\nu =$ 0 Nms (small flap drift, no overtopping), $\nu = 30$ Nms (some flap drift, light overtopping), and $\nu = 70$ Nms (severe flap drift and overtopping). We refer to these tests as Case 1, Case 2, and Case 3, respectively (Figure 5.2). We expect the data from these tests to exhibit nonlinearities from confinement, overtopping, and large-amplitude rotations due to flap drift. However, for simplicity and brevity we commonly refer to the observed nonlinear

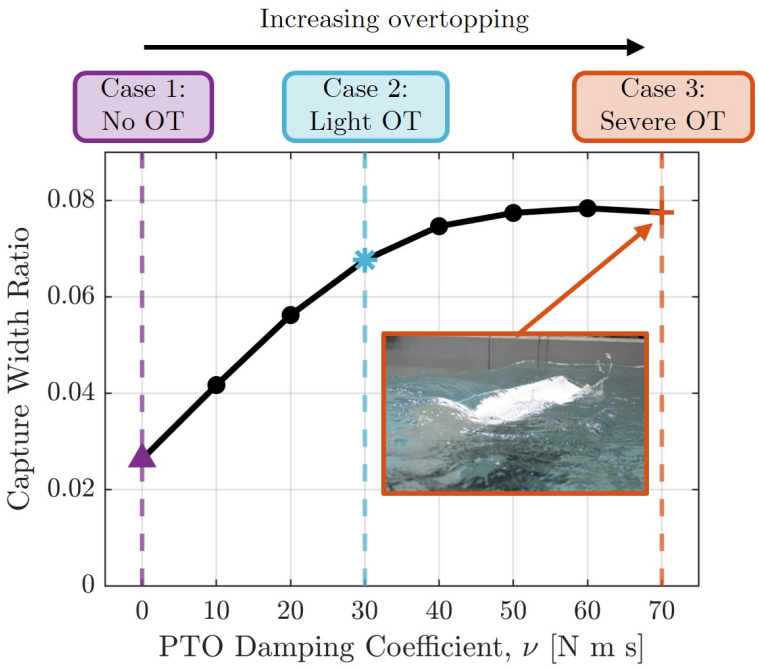

Figure 5.2: OSWEC capture width ratio as a function of damping coefficient for a wave height of 0.1 m and period of 1.5 s. Dashed lines represent experiments with different amounts of overtopping (OT), with purple representing no OT, blue representing light OT, and orange representing severe OT. The inset image shows an example of severe overtopping (taken during a different experiment).

phenomenon as overtopping (OT). Our goal is to build SINDy models that describe the dynamics for all three cases and physically interpret the results to identify how nonlinearity shows up mathematically in the system models. While these phenomena are specific to this experimental setup and device, nonlinear events such as overtopping and flap drift can occur in full-scale devices, therefore a modeling method that can accurately identify severe nonlinearity in WEC data is a promising tool, regardless of the source. However, overtopping would likely be less frequent in irregular seas where flap drift would not be as regular and consistent.

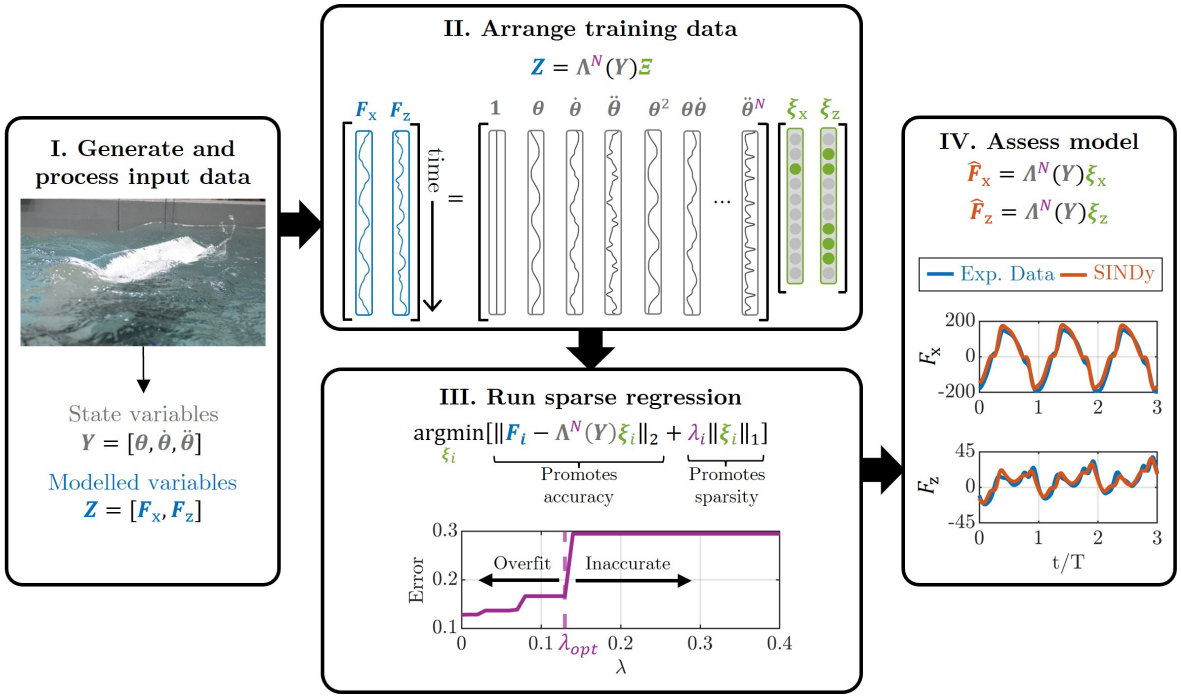

Figure 5.3: SINDy data analysis workflow. (Block I) To start, we collect experimental data including flap kinematics and loads (Section 3.1). (Block II) After processing, we arrange the data for the SINDy algorithm, which creates a library of nonlinear functions of order N, $\boldsymbol{\Lambda}^N(\boldsymbol{Y})$, from data $\boldsymbol{Y}$. (Block III) SINDy then uses a sparse regression algorithm to solve $\boldsymbol{F}_i = \boldsymbol{\Lambda}^N(\boldsymbol{Y})\boldsymbol{\xi}_i$ for coefficients, $\boldsymbol{\xi}_i$. The sparsity-promoting hyperparameter λ is tuned to ensure $\boldsymbol{\xi}_i$ has the minimum number of terms to accurately describe dynamics $\boldsymbol{F}_i$. (Block IV) After choosing λ, we compare the reduced-order model to original data to assess if the dominant dynamics are adequately captured.

5.4.2 Workflow

Figure 5.3 outlines the major steps for applying SINDy to experimental data, which consist of generating and processing the input data, arranging the training data, running the sparse regression, and assessing the reduced-order model. We describe each step in detail in the following sections.

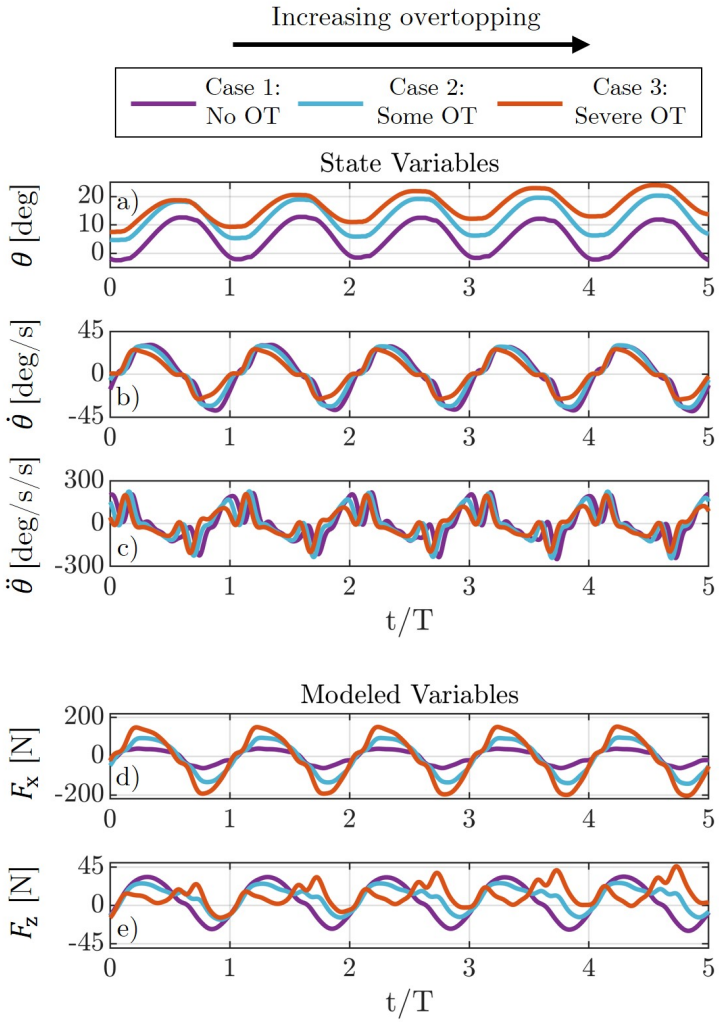

Figure 5.4: Time series of (a-c) state and (d, e) modeled variables with time normalized by wave period, T. Colors correspond to tests with different damping coefficients (Figure 5.2), which correlate to the amount of overtopping (OT) during the test.

Generate and process data

The first step is to generate the data that make up both the state variable matrix, $\boldsymbol{Y}$, and modeled variable matrix, $\boldsymbol{Z}$ (Block I in Figure 5.3). As previously mentioned, we use processed data from experiments described in Section 3.1, however this process similarly

applies to high-fidelity numerical data from mesh-based CFD or SPH simulations.

For this study, we model the surge and heave forces acting at the hinge of the device ($\boldsymbol{F}_{\mathrm{x}}$ and $\boldsymbol{F}_{\mathrm{z}}$ in Equation 3.1, respectively) as a function of flap position measured by the encoder, $\boldsymbol{\theta}$, and numerically calculated velocity, $\dot{\boldsymbol{\theta}}$, and acceleration, $\ddot{\boldsymbol{\theta}}$ (Section 3.2). We chose these state and modeled variables because surge and heave forces are not often measured or investigated when compared to pitch torques in OSWEC studies and we wanted to describe them in terms of commonly measured states such as flap position and velocity. Figure 5.4 presents time series data of the three state variables (position, $\boldsymbol{\theta}$, velocity, $\dot{\boldsymbol{\theta}}$, and acceleration, $\ddot{\boldsymbol{\theta}}$) and two modeled variables (surge force, $\boldsymbol{F}_{\mathrm{x}}$, and heave force, $\boldsymbol{F}_{\mathrm{z}}$) for the three different damping values highlighted in Figure 5.2: $\nu = 0$ Nms (Case 1, no overtopping), 30 Nms (Case 2, light overtopping), and 70 Nms (Case 3, severe overtopping). As previously mentioned, there is evident drift in the position time series that becomes more prominent with increased damping. Although oscillation amplitudes are similar for the three cases, the maximum displacement for Case 3 is approximately double that of Case 1. For velocity, all three cases are relatively sinusoidal with similar amplitudes, however for Case 3, the shape of oscillations is skewed and there is a slight asymmetry in the maximum and minimum of the oscillations. In contrast to the predominantly sinusoidal position and velocity signals, the acceleration time series, though fairly consistent between the damping values, is significantly more complicated due to disturbances at zero velocity, when the flap is changing direction, and slight variations in the shape of the velocity oscillations. Overall, besides the drift in flap position, the general behavior of all three state variables is fairly consistent between damping values/overtopping severity.

For the modeled variables, the structure of surge force, $\boldsymbol{F}_{\mathrm{x}}$, does not vary significantly between the three cases, however the magnitude increases with damping. Heave force, $\boldsymbol{F}_{\mathrm{z}}$, varies drastically with increasing damping, suggesting the overtopping and confinement affect heave force more significantly than surge. At low damping (Case 1), heave force exhibits sinusoidal behavior, however increasing damping (and inherently increasing overtopping severity) results in a more complex structure that evolves throughout the time series. This behavior

is evident in both Cases 2 (light overtopping) and 3 (severe overtopping). Our objective is for SINDy to accurately describe both modeled variables for all three cases, regardless of overtopping severity or complexity.

Arrange training data

Once the experimental data is collected and processed, the next step is to build data matrices, $\boldsymbol{Y}$ and $\boldsymbol{Z}$, and calculate the nonlinear function library, $\boldsymbol{\Lambda}(\boldsymbol{Y})$ (Block II in Figure 5.3). Based on our selection of state and modeled variables, our data matrices take the form:

$$\boldsymbol{Y} = \begin{bmatrix} | & | & | \\ \boldsymbol{\theta} & \dot{\boldsymbol{\theta}} & \ddot{\boldsymbol{\theta}} \\ | & | & | \end{bmatrix} \in \mathbb{R}^{m\times 3}, \quad \boldsymbol{Z} = \begin{bmatrix} | & | \\ \boldsymbol{F}_{\mathrm{x}} & \boldsymbol{F}_{\mathrm{z}} \\ | & | \end{bmatrix} \in \mathbb{R}^{m\times 2}. \tag{5.6}$$

Similar to the data in Chapter 4, each state variable is normalized by its root-mean-square (RMS) value and the modeled variables are normalized by the largest RMS values between them. This scales the relative importance of the modeled variables while reducing significant differences in orders of magnitudes between state and modeled variables.

State variable matrix, $\boldsymbol{Y}$, is then used to generate the nonlinear function library $\boldsymbol{\Lambda}(\boldsymbol{Y})$ (Equation 5.3). Each column of $\boldsymbol{\Lambda}$ contains a polynomial combination of the state variables in $\boldsymbol{Y}$. The function `poolData.m` in the open-source SINDy repository automatically builds $\boldsymbol{\Lambda}$ for polynomial functions of $\boldsymbol{Y}$ for a given polynomial order N. However, $\boldsymbol{\Lambda}$ can include any function of $\boldsymbol{Y}$ outside of those generated in `poolData.m`. Including all the necessary states in $\boldsymbol{Y}$ or the appropriate functions in $\boldsymbol{\Lambda}$ can be difficult, especially for complex systems, but is critical to building an accurate reduced-order model with SINDy. Therefore, multiple iterations of different functions and states may be necessary to create a sufficient library to identify a sparse and accurate system model.

Run sparse regression

Next, we run the SINDy algorithm on the training data (Block III in Figure 5.3) by solving Equation 5.5 individually for both the surge and heave foundation forces. This results in symbolic equations that describe the reduced-order models $\hat{F}_{\mathrm{x}}(t)$ and $\hat{F}_{\mathrm{z}}(t)$:

$$\hat{F}_{\mathrm{x}}(t) = \hat{\boldsymbol{\Lambda}}^{N}(\boldsymbol{\chi})\boldsymbol{\xi}_{\mathrm{x}}, \quad \hat{F}_{\mathrm{z}}(t) = \hat{\boldsymbol{\Lambda}}^{N}(\boldsymbol{\chi})\boldsymbol{\xi}_{\mathrm{z}}, \tag{5.7}$$

where $\hat{\boldsymbol{\Lambda}}^{N}(\boldsymbol{\chi})$ is a library of polynomial functions up to order N composed of symbolic variables $\theta(t)$, $\dot{\theta}(t)$, and $\ddot{\theta}(t)$ (not data), and $\boldsymbol{\xi}_{\mathrm{x}}$ and $\boldsymbol{\xi}_{\mathrm{z}}$ are the coefficients from solving Equation 5.5 with λ_{opt} for surge and heave forces, respectfully. This form of output is advantageous because we can identify drivers of nonlinear behavior by the nonzero terms in $\boldsymbol{\xi}_i$ and, from this, learn more about the underlying dynamics. Additionally, having a closed-form expression for the modeled variable can be critical for important WEC applications such as optimal control, where a fast and accurate system model is needed.

Choosing the maximum polynomial order, N, and sparsity-promoting hyperparameter, λ, is a critical step in building parsimonious models. To choose the optimal parameters that balance model accuracy and complexity, we sweep through a range of λ and N and assess the error of the resulting models. To visualize this tradeoff, Figure 5.5 shows error between experimental heave force data, $\boldsymbol{F}_{\mathrm{z}}$, and the resulting SINDy model as a function of N (top panel) and number of nonzero terms in $\boldsymbol{\xi}_{\mathrm{z}}$ (bottom panel), which directly corresponds to λ, for all three cases. Similar to Chapter 4, we use a normalized error parameter, ϵ, such that:

$$\epsilon_i = \frac{\left|\left|\boldsymbol{\zeta}_i - \boldsymbol{\Lambda}^{N}(\boldsymbol{Y})\boldsymbol{\xi}_i\right|\right|_2}{||\boldsymbol{\zeta}_i||_2}. \tag{5.8}$$

Our goal is to choose the smallest N and fewest terms in $\boldsymbol{\xi}_i$ (corresponding to the largest λ) without accruing significant error. There is a clear drop in error for $N = 2$ for Cases 2 and 3, which implies that quadratic terms are necessary for describing those dynamics. Since error does not significantly decrease for $N > 2$, we choose $N = 2$ for all three SINDy models. Additionally, as λ increases, the number of nonzero terms decreases as sparsity is more heavily weighted in Equation 5.5. By choosing the number of terms to include, we are

implicitly describing λ_{opt}. Following the same logic as choosing N, we select models with the fewest terms that do not result in significant error. This results in two terms for Case 1, three terms for Case 2, and four terms for Case 3. This process was repeated for surge force. Note that even though including more terms decreases the error in the reduced-order model, that would likely lead to overfitting the training data and failing to identify a parsimonious model.

Assess model

Finally, once we have chosen N and λ_{opt}, we compare the modeled variable data in $\boldsymbol{Z}$ to the resulting SINDy model. Additionally, we assess the composition of the SINDy model to interpret the role of nonlinearity as overtopping and confinement effects increase (Sections 5.6.1 and 5.6.2). Finally, we determine how well the SINDy models generalize to data from tests it was not trained on to understand how the dynamics change throughout the control regime (Section 5.6.3).

5.5 Results

In this section, we review the accuracy and composition of the resulting SINDy models from the parameters chosen in Figure 5.5. We consider the reduced-order models for both surge and heave forces for all three cases.

5.5.1 Surge Force

All three SINDy models for surge force contain a single term, regardless of overtopping severity, and take the form $\hat{F}_{\text{x}}(t) = \xi_{\text{x},\dot{\theta}}\dot{\theta}(t)$ (Figure 5.6a). The resulting SINDy models for the three cases are:

$$\begin{aligned} \hat{F}_{\text{x},1}(t) &= 81\ \dot{\theta}(t), \\ \hat{F}_{\text{x},2}(t) &= 213\ \dot{\theta}(t), \\ \hat{F}_{\text{x},3}(t) &= 388\ \dot{\theta}(t). \end{aligned} \tag{5.9}$$

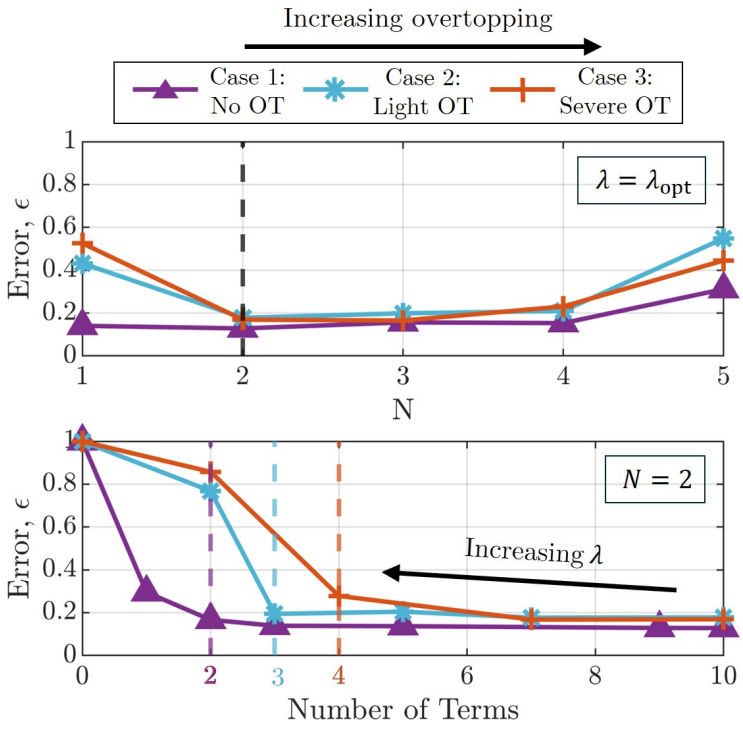

Figure 5.5: [Top] Normalized error as a function of polynomial order, N, for an optimal hyperparameter, λ_{opt} for all three cases. We choose $N = 2$ (dashed line) based on the significant drop in error for Cases 2 and 3 and the lack of accuracy gained when including higher orders. [Bottom] Error as a function of number of terms included in reduced-order model for $N = 2$. Number of terms decrease as λ increases and sparsity is prioritized. For each case, we identify the model with the smallest number of terms that does not result in a drastic increase in error (dashed lines), and that corresponds to the hyperparameter chosen for that SINDy model, λ_{opt}.

This suggests that surge force is in phase with angular velocity and the proportion coefficient, $\xi_{\dot{\theta},\text{x}}$, changes with different levels of damping. This form of $\hat{F}_{\text{x}}$ makes physical sense, as the surge component of the incident wave is the dominant driver of flap motion. Even with a single term, the SINDy fit for all three cases is accurate in both the training and

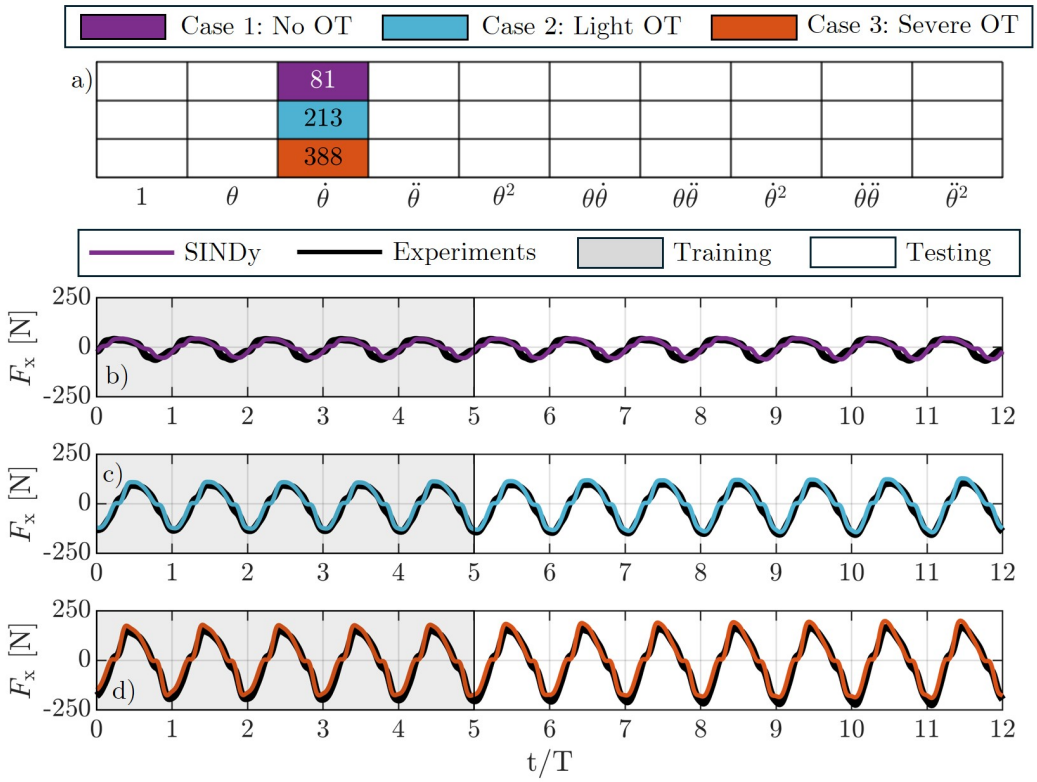

Figure 5.6: a) Activated terms in the surge force SINDy model, $\boldsymbol{\xi}_x$, for Case 1 (no overtopping, OT, purple, top row), Case 2 (light OT, blue, middle row), and Case 3 (severe OT, orange, bottom row). White squares correspond to zero terms in $\boldsymbol{\xi}_x$, while colored squares correspond to nonzero terms with the coefficient magnitude reported inside the square. Time series of experimental surge force (black) and SINDy model for b) Case 1, c) Case 2, and d) Case 3. All SINDy models are trained on the first five periods of data (gray region).

testing region (Figure 5.6b-d), and these trends hold for all other tests in the damping sweep (not shown). Although these dynamics are relatively simple, this highlights the benefits of promoting sparsity to accurately model relevant dynamics without overfitting the data.

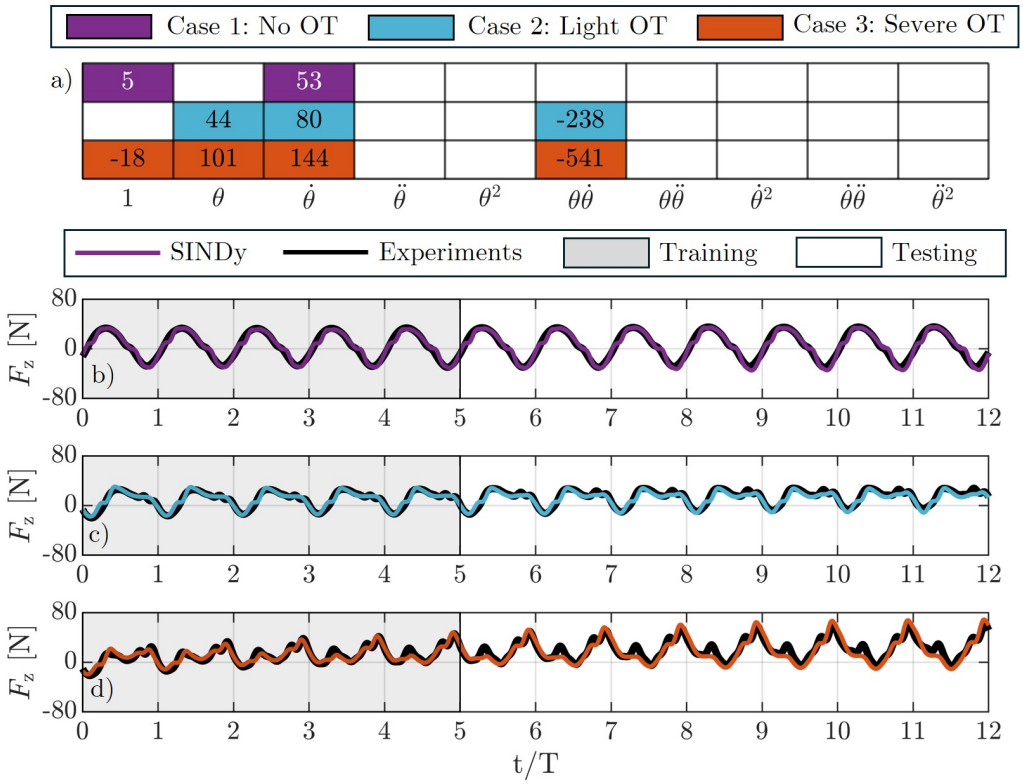

Figure 5.7: a) Activated terms in the heave force SINDy model, $\boldsymbol{\xi}_z$, for Case 1 (no overtopping, OT, purple, top row), Case 2 (light OT, blue, middle row), and Case 3 (severe OT, orange, bottom row). White squares correspond to zero terms in $\boldsymbol{\xi}_z$, while colored squares correspond to nonzero terms with the coefficient magnitude reported inside the square. Time series of experimental heave force (black) and SINDy model for b) Case 1, c) Case 2, and d) Case 3. All SINDy models are trained on the first five periods of data (gray region).

5.5.2 Heave Force

Unlike surge, the heave force dynamics vary drastically with overtopping severity in both structure and magnitude (Figure 5.7). Despite the additional complexities, SINDy generates accurate time series models for all three cases (Figure 5.7b-d), including Case 3 (Figure 5.7d), where the dynamics evolve throughout the training and testing region. The resulting SINDy

models for the three cases are:

$$\begin{aligned}
\hat{F}_{\mathrm{z},1}(t) &= 5 + 53\,\dot{\theta}(t), \\
\hat{F}_{\mathrm{z},2}(t) &= 44\,\theta(t) + 80\,\dot{\theta}(t) - 238\,\theta(t)\,\dot{\theta}(t), \\
\hat{F}_{\mathrm{z},3}(t) &= -18 + 101\,\theta(t) + 144\,\dot{\theta}(t) - 541\,\theta(t)\,\dot{\theta}(t).
\end{aligned} \tag{5.10}$$

The activated terms in the models in Equation 5.10 give insight into the system dynamics. While Case 1 (no overtopping) exhibits similar linear behavior as the surge force and is proportional to angular velocity, as overtopping severity increases additional terms are activated in $\hat{F}_{\mathrm{z},2}$ and $\hat{F}_{\mathrm{z},3}$ including θ and $\theta\dot{\theta}$. We investigate the roles these terms play in modeling heave force dynamics in Section 5.6.2.

5.6 Discussion

Here, we explore the benefits of using SINDy to model WEC dynamics by assessing the importance of model nonlinearity, interpreting the models to gain insight on WEC dynamics, and evaluating how well the models generalize to tests across the damping sweep.

5.6.1 Importance of nonlinearity

SINDy models can include nonlinearity without a significant increase in setup complexity or computational cost. To evaluate the importance of nonlinearity in this system, we compare the SINDy models to a purely linear model using WEC-Sim [58]. This WEC-Sim model is almost identical the one used in Chapter 4, but has updated system parameters and BEM inputs to match the experimental setup and does not include nonlinear FK or buoyancy forces. Details on the BEM used in this model are given in Appendix A. For surge force, F_{x} (Figure 5.8, top row), both WEC-Sim and SINDy accurately model the experimental data. This is unsurprising because the dynamics are predominantly linear. For heave force (Figure 5.8, bottom row), the agreement between the models and experiments vary for the three cases. For Case 1 (no overtopping), WEC-Sim describes the dynamics well, with small error in modeling the peaks of the force. However, for Cases 2 and 3, where overtopping is

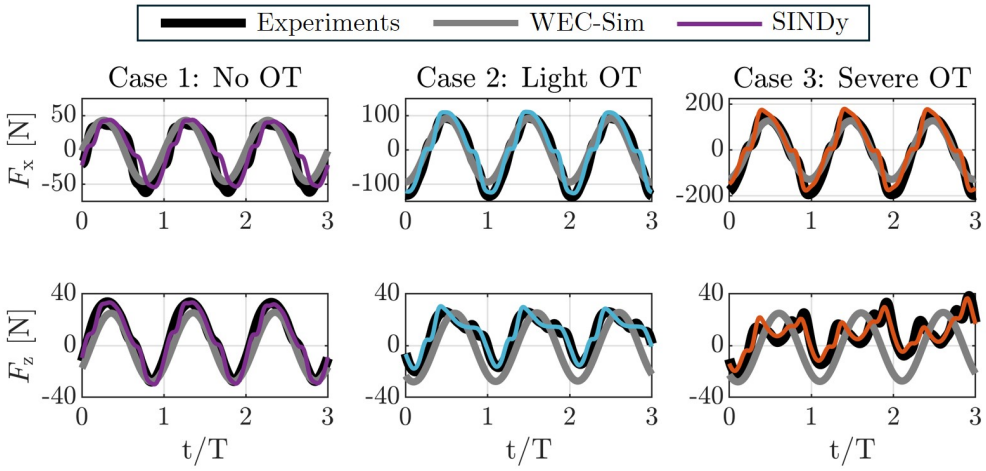

Figure 5.8: Comparison of linear WEC-Sim (gray) and nonlinear SINDy (color) models to experimental data (black) for [top row] surge and [bottom row] heave force for the first three periods of training data. While both WEC-Sim and SINDy provide accurate models for the predominantly linear cases, as nonlinearity increases, the SINDy models are significantly more accurate.

more severe, WEC-Sim loses accuracy because there is no mechanism for a linear model to produce the non-sinusoidal response required to represent the complex behavior. However, the nonlinear terms in the SINDy models for these cases are significantly more accurate. These results demonstrate that SINDy can be an important tool for modeling dynamics with nonlinear events, such as overtopping, that invalidate linear assumptions.

5.6.2 Physical interpretation

One of the benefits of SINDy is using the models to learn more about the system dynamics. Here, we explore each term in Equations 5.9 and 5.10 in order to gain insight on their role in describing $\boldsymbol{F}_x$ and $\boldsymbol{F}_z$.

1. $\dot{\theta}$: This term is the dominant linear term for both surge and heave force that is activated in all models. In the absence of flap drift and overtopping, both surge and heave force

are proportional to angular velocity. It is interesting that two orthogonal cartesian forces can both be described as proportional to a rotational kinematic quantity, such as flap velocity.

2. θ: This term is activated for heave force SINDy models at mid-to-high damping, when flap drift is prevalent. While the heave force for Case 1 is mostly symmetric, for Cases 2 and 3 there is evident drift in the heave force dynamics that is accounted for by the θ term in $\hat{F}_{z,2}$ and $\hat{F}_{z,3}$ (Figure 5.7). In other words, including θ in the heave force models for Cases 2 and 3 provides the drift in the SINDy models that is present in the data.

3. $\theta\dot{\theta}$: This nonlinear term, like θ, is activated for heave force SINDy models at mid-to-high damping levels and is the only second-order term excited in any of the models. Because $\theta\dot{\theta}$ is present in multiple heave force SINDy models, it likely describes fundamental dynamics that are introduced when flap drift and/or overtopping occurs. This nonlinear term could arise from multiple sources, but suggests there is a positional dependence on velocity that increases as the flap drifts and overtopping intensifies. This could arise from positionally-depending hydrodynamic coefficients that become prevalent at large rotations. Specifically, we found that added mass, radiation damping, and excitation force coefficients vary with flap position in the heave direction, but not in surge (Appendix A). The positional dependence of one or more of these coefficients could manifest as a force proportional to velocity that changes with θ. Alternatively, we could interpret $\theta\dot{\theta}$ using quasi-steady rotating plate theory as a projection of the normal force acting on the flap in the heave direction [110]. Due to the complexity of the dynamics and the interplay between different nonlinear phenomenon, it is not possible to state with certainty the source of this term. However, these hypotheses provide interesting avenues of investigating nonlinear WEC behavior in future testing.

Overall, the parsimonious SINDy models in Equations 5.9 and 5.10 contain physically

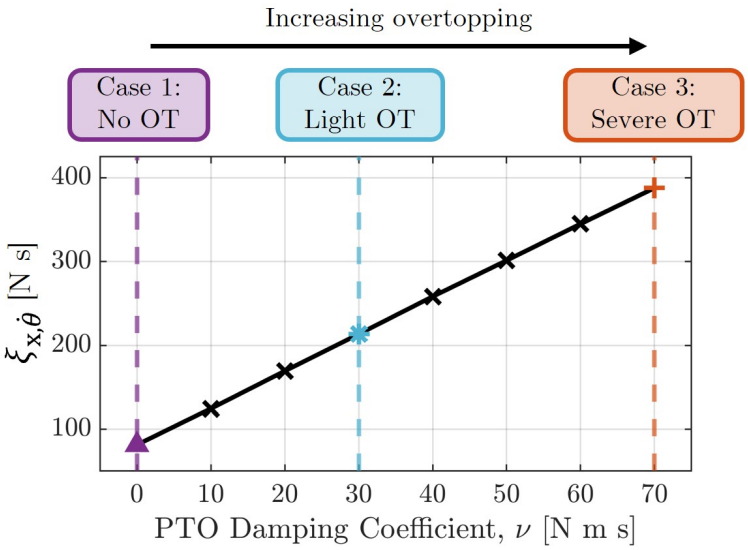

Figure 5.9: SINDy velocity coefficient, $\xi_{x,\dot{\theta}}$, for surge force, $\hat{F}_x$, as a function of damping coefficient, ν.

relevant dynamics that would not be identified using linear modeling techniques.

5.6.3 Generalization

As previously mentioned, with appropriate choice of nonlinear library and hyperparameters, SINDy results in models that are parsimonious, and therefore may be generalizable to other data with similar underlying dynamics. We explore the ability for the SINDy models generated in Section 5.5 to extend to other tests in the damping sweep (Figure 5.2) beyond what the models were trained on. Our goal is to see if a single model is capable of describing the dynamics over the entire control range, or if multiple expressions are needed.

Surge force

The terms included in the SINDy models for surge force do not change for different tests throughout the damping sweep and always take the form: $\hat{F}_x(t) = \xi_{x,\dot{\theta}}\dot{\theta}(t)$, regardless of overtopping severity. Therefore, we can identify how the SINDy coefficient, $\xi_{x,\dot{\theta}}$, changes

with damping coefficient, ν. This relationship is demonstrated in Figure 5.9, where $\xi_{x,\dot{\theta}}$ increases linearly with the damping coefficient ν, such that: $\xi_{x,\dot{\theta}} = 81.5 + 4.4\nu$. We can use this relation to generate a single expression for $\hat{F}_x$ that is valid for all damping values:

$$\hat{F}_x(t) = (81.5 + 4.4\nu)\dot{\theta}(t). \tag{5.11}$$

Equation 5.11 provides a comprehensive expression for $\hat{F}_x$ that is valid over the entire control regime and is only a function of ν and $\dot{\theta}(t)$. This is a powerful model because it relates two important quantities in a simple analytical expression that is easier to evaluate than using a tool like WEC-Sim. In addition, if an estimate for the wave surge force is available, this provides a tool for determining flap velocity under different control inputs.

Heave force

To evaluate how well the SINDy models in Equation 5.10 generalize to other tests, we first apply each model to the data from all other tests in the PTO sweep, the results of which are summarized in Figure 5.10. In general, applying SINDy models to tests with similar levels of damping to the test it was trained on (and therefore similar levels of overtopping) results in low error, while applying the models to tests with drastically different damping results in high error. In other words, the SINDy models only generalize to tests with similar levels of damping, and therefore similar amounts of overtopping. This implies there are fundamental changes in the dynamics as overtopping severity increases that require different models that include varying nonlinearity. As a result, there are three regions throughout the damping range that denote where each model is appropriate to use (shaded regions in Figure 5.10, top panel). Specifically, $\hat{F}_{z,1}$ is the most accurate model for tests with damping up to 10 Nms, $\hat{F}_{z,2}$ is the most accurate model for tests with damping between 10 and 50 Nms, and $\hat{F}_{z,3}$ is the most accurate model for tests with damping above 50 Nms. To visualize how these models perform on other tests, Figure 5.10 also shows time series fits of the three SINDy models ($\hat{F}_{z,1}$, $\hat{F}_{z,2}$, and $\hat{F}_{z,3}$) applied to the data from the three cases. As expected, no single model qualitatively captures the dynamics of all three experiments. This further highlights

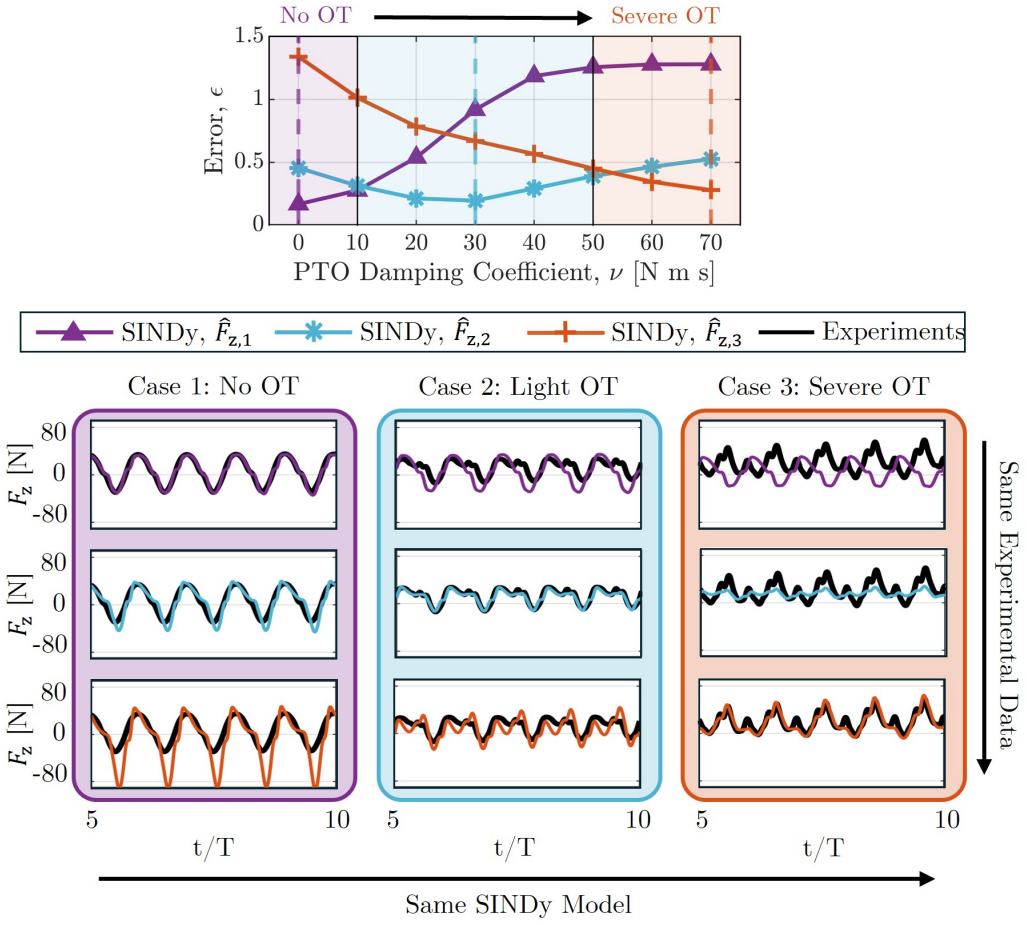

Figure 5.10: [Top] Normalized error, ϵ, as a function of damping coefficient, ν, for the three SINDy models of heave force, $\hat{F}_z$. The color of the solid line corresponds to the test the model was trained on, marked by the dashed vertical line of the same color. The color of the shaded region corresponds to the SINDy model that results in lowest error for those tests. [Bottom] Time series of experimental heave force data data (black) and SINDy fits (color) for the three cases. Each column shows experimental data from the same damping value, as indicated by the shaded region behind each column. Each row shows SINDy fits trained on the test that corresponds to the line color. For example, the purple lines in the top row are from the SINDy fit trained on Case 1, $\hat{F}_{z,1}$, applied to the experimental data for all cases.

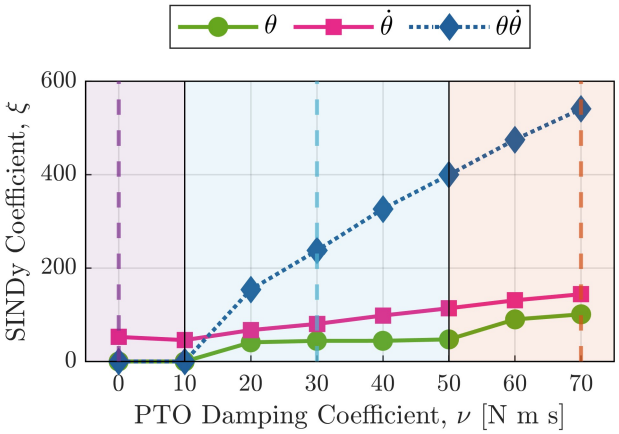

Figure 5.11: SINDy function coefficients for (green circles) θ, (pink squares) $\dot{\theta}$, and (dark blue diamonds) $\theta\dot{\theta}$, for heave force model, $\hat{F}_z$. The color of the shaded region corresponds to the SINDy model that results in lowest error for those tests, with dark purple, blue, and orange representing SINDy models from Case 1, 2, and 3, respectively (Figure 5.10).

limitations of linear modeling techniques that cannot account for changing dynamics or increasing nonlinearities within a damping sweep.

Finally, similar to surge force, we built heave force SINDy models for all tests in the damping sweep to evaluate how the coefficient for each term in the model changes with damping coefficient (Figure 5.11). Similar to Figure 5.10, there are three regions of interest. In region 1 (purple region in Figure 5.11) the only activated term is $\dot{\theta}$ and the function coefficients seem relatively constant as a function of ν, although data is limited. Outside of this region, θ and $\theta\dot{\theta}$ terms are activated and, similarly to surge force, the coefficient for $\dot{\theta}$ increases linearly with ν. The coefficient for $\theta\dot{\theta}$ also increases with ν, indicating nonlinearity becomes more prominent at higher damping, which is expected. Additionally, in each region, there is a distinct jump in the θ coefficient, again suggesting there are fundamental changes in the dynamics that occur throughout the control regime that explain why heave force cannot be described with a single expression. Overall, these coefficients show that SINDy can capture evolving dynamics with varying amounts of nonlinearity and extract common

terms to describe complex behavior.

5.7 Conclusions

In this work, we tested a laboratory-scaled OSWEC and use SINDy to model surge and heave forces using flap kinematics for three levels of damping. We found that increasing damping results in increased overtopping and confinement effects during the experiment, which affects surge and heave forces differently. While surge force maintains a linear relationship with flap velocity throughout the entire control range, heave force shows increasing nonlinearity as overtopping increases. For all three cases, SINDy identifies accurate models with a limited number of terms. Further, the SINDy models contain important nonlinearities that a purely linear model could not replicate. While we identify potential causes for the recurring terms in the SINDy models, we can not determine a definitive interpretation of the nonlinear term due to the coupled nature between the flap drift, overtopping and confinement effects. We also investigate how well the SINDy models generalize to other tests. While the models for surge force can be reduced to a single equation that is valid over the entire control range, heave force requires three regimes where each model is only appropriate to use in the regime in which it was trained. Overall, SINDy is an exciting tool that can expand not only the types of models we can create for WECs, but can assist in learning more about highly nonlinear systems that are not well understood through first principles.

The work in this chapter motivates further testing to expand the applicability of SINDy. We plan to conduct additional testing in a facility with less confinement and limit flap drift (which we explore in Chapter 6). Additionally, we aim to expand this work to other wave conditions to investigate how the resulting SINDy models change with wave height, period, and with an irregular wave spectrum. Once the experimental data is collected, we will run the SINDy algorithm on each test to explore the best-fit model for the different situations and compare the results to understand which terms are activated for the different tests. With these results, we aim to explore the following topics:

- How do the nonlinear terms in the SINDy models connect to hydrodynamic phenomena? By testing in a facility with less confinement, limiting flap drift, and varying the wave parameters, we can systematically isolate the effects of overtopping and large angular displacements to determine the source of nonlinearity in the models.

- How do the SINDy models (particularly the nonlinearity identified from the SINDy models) vary with excitation? Are multiple models required for different wave conditions (both regular and irregular), or can the dynamics be described with a general reduced-order model, regardless of excitation? Additionally, are there other states (such as flap pressure) that allow for equations to be more generalizable? For example, if we include pressure measurements in the SINDy library, is it possible to create a single equation to describe the heave force in this work instead of requiring three different models? This will provide insight into how we train models for control and how often they should be updated.

- Is knowledge of the incident wave required to build SINDy models of OSWEC dynamics in response to irregular waves? Or can the wave field be treated as a latent variable? This again has implications in control and future state prediction.

- Can SINDy be used in realistic wave conditions where hysteresis is present? In other words, can SINDy be used to address the issue of causality in WEC modeling and control? The SINDy models derived in this work are causal, as they only require the state measurements at that time step. However, by including past and future state measurements in the function library, $\mathbf{\Lambda}$, we can explore what is necessary to describe WEC dynamics in response to irregular waves, which is critical to informing future control schemes and modeling/predicting WECs operating in the ocean.

Expanding the analysis to a wider range of wave inputs is a step towards implementing SINDy models in an optimal control scheme to optimize WEC performance.

Chapter 6

INVESTIGATION INTO EXPERIMENTAL SCALE EFFECTS OF AN OSCILLATING SURGE WAVE ENERGY CONVERTER

The work in this chapter is co-authored by Greg Talpey, Ama Hartman, Corey Crisp, Gemma Calandra, Charles Candon, Rebecca Fao, and Dr. Brian Polagye.

6.1 Motivation

Scaled experimental modeling is an important step for WEC development, however, as previously mentioned, there are limitations when the scaled device does not achieve exact similarity to the full-scale system. Differences between model- and full-scale devices, referred to as "scale effects", can cause inaccurate predictions of full-scale dynamics and performance [56], therefore quantifying these differences and assessing their importance is critical to accurately inform full-scale design [37].

In this chapter, we identify, characterize, and mitigate scale effects in the experimental setup described in Chapter 3 and used in Chapter 5. Specifically, for a range of control parameters in regular waves, we characterize the effects of reduced flap buoyancy from pressure sensor weight and frictional torque from the waterproof driveline seals. From this, we develop control strategies to mitigate these effects and demonstrate their relative importance when interpreting scale-model experimental results. The remainder of this chapter is laid out as follows. Section 6.3 describes methods used in this chapter. Section 6.4 describes the results of this study, focusing on kinematic and performance implications of flap buoyancy and friction. Finally, Section 6.5 summarizes conclusions and recommended future work.

6.2 Review of prior work

Despite the body of work dedicated to experimentally investigating OSWEC hydrodynamics and performance, few of these studies have directly addressed limitations of the scale models beyond the general appropriateness of Froude scaling (e.g., Schmitt and Elsäßer [111]). However, characterizing and addressing the effect of friction in marine energy experiments has been previously explored. Asmuth *et al.* [112] identified frictional losses in the drivetrain for free decay experiments of a scaled OSWEC, but did not account for their effect on the hydrodynamics or quantify how the losses would affect full-scale performance predictions. Additionally, Lee *et al.* [113] characterized the nonlinear static and dynamic friction losses for a vortex vibration harvester and used a motor to offset the effect of those losses during testing. Similarly, DeVries *et al.* [114] characterized and compensated for the friction present in a two degree-of-freedom gantry system used to control a submerged cylinder in both surge and heave. Both studies used a motor with feedback control to apply a friction correction in addition to a spring and/or damping force. However, neither study directly measures the frictional losses in real-time or investigates the effect friction had on system behavior or performance, both of which we demonstrate here.

6.3 Methods

In this chapter, we use the same experimental setup described in Chapter 3, but employ control strategies in addition to linear damping to address limited flap buoyancy and driveline friction. Although the results presented here and in Chapter 5 came from the experimental same setup, the data were collected separately from different experiments.

6.3.1 Motor Control Strategies

To evaluate the significance of limited flap buoyancy due to sensor weight and friction introduced by driveline, we developed control strategies using real-time sensor feedback that mitigate these scale-model effects on the flap dynamics. Specifically, for each test there are up

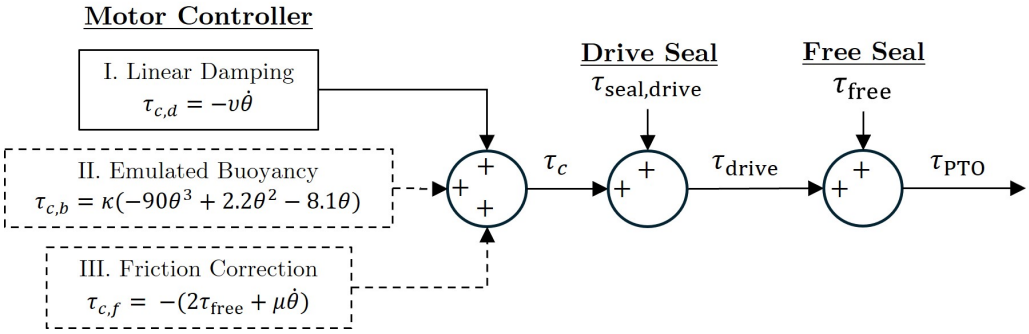

Figure 6.1: Diagram of OSWEC driveline torques including the control torque applied by the motor, τ_c, the torque measured by the drive load cell, τ_{drive}, the torque measured by the free load cell, τ_{free}, and the total PTO torque applied to the flap, τ_{PTO}. The control torque, τ_c, is composed of a linear damping torque, $\tau_{c,d}$, an optional torque to emulate additional flap buoyancy, $\tau_{c,b}$, and an optional torque to offset the role of friction in the driveline, $\tau_{c,f}$ (optional torques denoted by dashed lines).

to three control schemes applied by the motor: linear damping control, emulated buoyancy, and a friction correction (Blocks I, II, and III in Figure 6.1, respectively).

Emulated buoyancy

As described in Section 5.4.1, for tests with high damping, the flap tended to drift away from its neutral vertical position in time. This dynamic response occurred because the restoring torque was insufficient to return the flap to vertical before the next wave crest, likely due to the relative increase in flap density (and therefore decrease in flap buoyancy) from the pressure sensors (Section 3.1.2). To understand the implications of this buoyancy decrease on flap dynamics and performance, we use the motor to emulate a more buoyant flap by providing an additional restoring torque (Block II in Figure 6.1). We first characterize the restoring torque profile by measuring the torque required to hold the flap at different positions. As illustrated by Figure 6.2, there is a strong cubic relationship between net

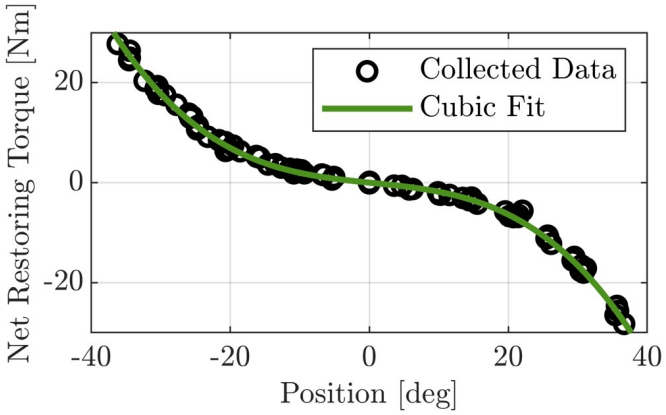

Figure 6.2: Net restoring torque as a function of position. Data (black circles) is well represented by a cubic fit (green line) for positions between ±40°.

restoring torque and flap position. Although the restoring torque is a combination of flap weight and buoyancy, for simplicity we refer to this torque as "buoyancy" torque, τ_b, which can be described by a best-fit cubic polynomial as:

$$\tau_b = -90.0\,\theta^3 + 2.2\,\theta^2 - 8.1\,\theta, \tag{6.1}$$

where θ is flap position in radians. Because of the cubic relationship and the relatively small linear coefficient in Equation 6.1, substantial buoyancy torque only occurs for larger oscillations. For example, for oscillation amplitudes less than 25 degrees, the buoyancy torque is smaller than the frictional losses in the driveline (about 10 Nm, Section 6.3.1 and Figure 6.3), meaning at small oscillation angles, buoyancy is insufficient overcome the friction in the driveline to return the flap to vertical.

To emulate additional flap buoyancy, we apply a control torque, $\tau_{c,b}$, that takes the same form as Equation 6.1:

$$\tau_{c,b} = \kappa\tau_b, \tag{6.2}$$

where κ is a "buoyancy coefficient" that describes the factor by which we are increasing flap

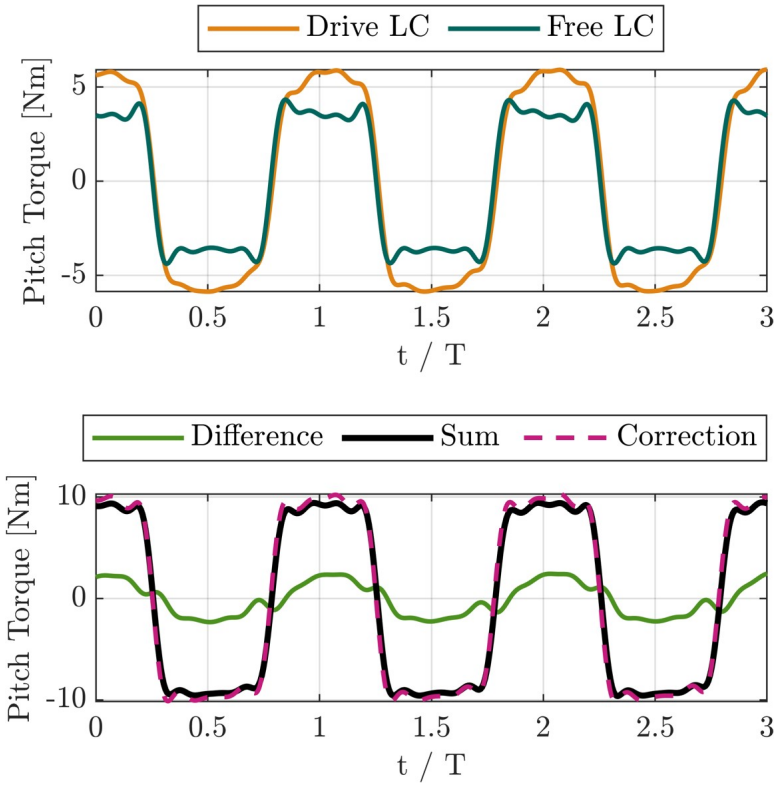

Figure 6.3: [Top] Pitch torque measurements from the drive (yellow) and free (teal) load cell for the case with no control torque ($\nu = 0$ Nms). In this case, the load cells are only measuring the losses on each side of the driveline. [Bottom] The sum (black) and difference (green) between the drive and free load cell measurement. The sum represents the total torque due to friction, while the difference highlights asymmetries in the drive and free ends of the driveline. The pink dashed line shows the friction correction, $\tau_{c,f}$ for this test.

buoyancy. For example, when $\kappa = 1$, we are doubling the total flap buoyancy.

Friction compensation

As previously mentioned, while the driveline seals are imperative to waterproof the housing, they cause appreciable friction torque on the driveline (Section 3.1.2, Figure 3.3). To determine the form of this torque, we performed an uncontrolled experiment (i.e., $\tau_c = 0$ Nm)

such that the load cells only measure friction on either end of the driveline. Seal friction produces an oscillatory torque with amplitude around 5 Nm (Figure 6.3) that is in phase with negative flap velocity, similar to a linear damper (Equation 3.3). We expect the slight difference in peak torque between the drive and free end to be a consequence of additional friction internal to the motor. Combined, the system experiences a friction torque of up to 10 Nm, which is appreciable in comparison to the imposed control torque.

To offset the effect of seal friction and understand its effect on flap dynamics and performance, we command a control torque, $\tau_{c,f}$, that is equal and opposite to the driveline friction (Block I in Figure 6.1). Because of potential changes to the form of seal friction under different operating conditions, this correction uses real-time torque feedback from the load cells. However, the friction torque on the drive side cannot be exactly isolated from the control torque for cases where $\tau_c \neq 0$, therefore we estimate the friction torque from only the free load cell and apply a correction torque, $\tau_{c,f}$, of the form

$$\tau_{c,f} \approx -2\tau_{\text{free}} - \mu\dot{\theta}, \tag{6.3}$$

where μ is a tuning parameter that accounts for the asymmetry in friction between the drive and free side. We found an μ value of 4.6 to best approximate the difference between the drive and free load cell measurements for the case of no control torque ($\tau_c = 0$ Nms). $\tau_{c,f}$ is an accurate fit for the friction in the uncontrolled test (pink line, Figure 6.3), however this derivation does assume that these measurements are representative of the friction asymmetry under all test conditions. By applying this correction with the servomotor and using the real-time torque measurements, we account for the significant friction in the driveline and offset its effect.

6.3.2 Control Strategy Evaluation

To assess the efficacy of these mitigation procedures, we consider six damping sweeps and compare their kinematics and performance (Table 6.1). First, to evaluate the role of buoyancy

Table 6.1: Summary of damping sweeps with varying amounts of added buoyancy ($\kappa = 0, 5, 7.5, 10$) without the friction correction and two sweeps with $\kappa = 0, 10$ with the friction correction.

Sweep Number	**Buoyancy Coeff., κ**	**Friction Correction**	**Marker**
1	0	No	
2	5	No	
3	7.5	No	
4	10	No	
5	0	Yes	
6	10	Yes	

on OSWEC dynamics, we consider four damping sweeps with different amounts of additional buoyancy ($\kappa = 0, 5, 7.5$, and 10) and no friction correction. Next, to evaluate the role of driveline friction, we consider two additional damping sweeps with the friction correction applied and buoyancy coefficients of $\kappa = 0$ and 10. The latter highlights any coupled effects between buoyancy and friction. Similar to Chapter 5, all tests occurred in monochromatic waves with a wave height of 0.1 m and a period of 1.5 seconds.

We compare kinematics between experiments including mean flap position, position range, velocity magnitude, and velocity magnitude range (Figure 6.4). To exclude transient behavior at the beginning of the experiment, mean flap position and position range are calculated over the last nine flap oscillations, which we approximate as steady-state conditions (teal shaded region, Figure 6.4). We calculate velocity magnitude using the fast-Fourier transform (FFT) of the full time series. However, the velocity magnitude can vary slightly throughout the test. Therefore, we calculate the FFT on a sliding window containing four periods of data to account for the temporal variation (pink shaded region, Figure 6.4). We

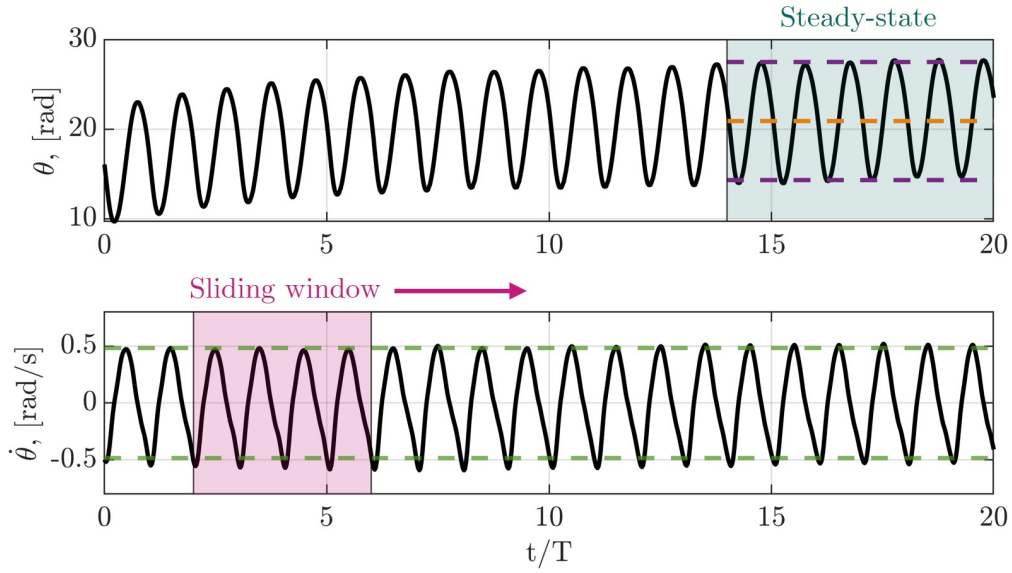

Figure 6.4: [Top] Flap position time series. Due to the transient drift in position, the last nine oscillations represent the steady-state region (teal shading), where we calculate the mean flap position (yellow dashed line) and position range (purple dashed lines). [Bottom] Velocity time series. Green dashed lines represent velocity magnitude calculated from the FFT of the full signal. To quantify temporal variations in magnitude, we calculate the FFT on a sliding window (pink shaded region) containing four periods of data. Both time series are from a test in Sweep 1 with $\nu = 50$ Nms, but the trends and procedures are similar for all tests in Sweeps 1-6.

consider the maximum and minimum amplitude of all the windows as the velocity magnitude range. Finally, to align time series from different tests, we shift one data set such that their wave gauge signals are in phase. This allows for a single time shift to compare position, velocity, and power time series and provides accurate comparisons relative to their respective incident wave.

To compare parameters across control schemes, we define an "effective" damping coefficient, $\hat{\nu}$. This takes advantage of the observation that driveline friction is primarily a function

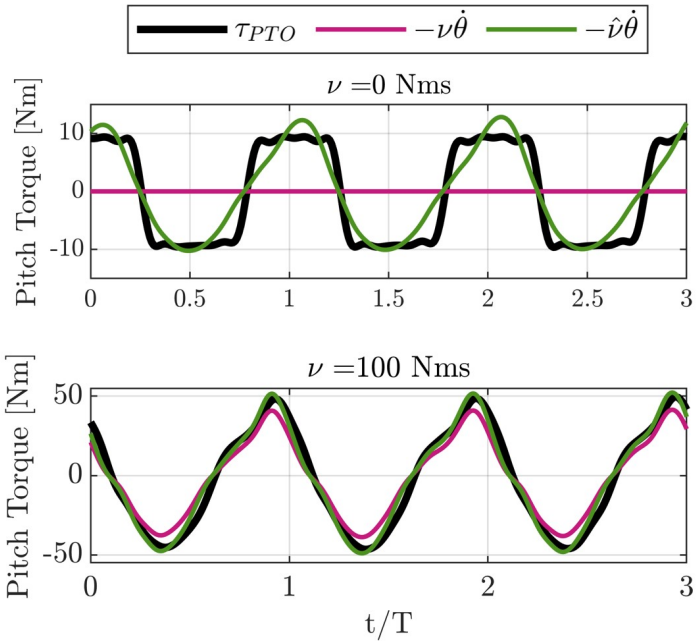

Figure 6.5: Pitch torque as a function of time for two experiments in Sweep 1 ($\kappa = 0$ and no friction correction): [top] $\nu = 0$ Nms and [bottom] $\nu = 100$ Nms. Black lines represent measured PTO torque, τ_{PTO}, pink lines represent the commanded linear damping control, $\tau_{c,d} = -\nu\dot{\theta}$, and the green lines represent effective damping torque, $-\hat{\nu}\dot{\theta}$.

of velocity, such that:

$$\tau_{\text{PTO}} = \tau_{\text{drive}} + \tau_{\text{free}} \approx -\hat{\nu}\dot{\theta}. \tag{6.4}$$

where $-\hat{\nu}\dot{\theta}$ is the effective damping torque. As opposed to ν, which is the nominal damping coefficient sent to the motor, $\hat{\nu}$ considers the additional torques in the driveline and therefore includes the impact of friction in τ_{PTO}. This enables consistent comparison between PTO sweeps with and without the friction correction. For each experiment, we estimate $\hat{\nu}$ through linear regression ($\hat{\nu}$ minimizes the L_2 error between τ_{PTO} and -$\hat{\nu}\dot{\theta}$). As shown in Figure 6.5, while -$\hat{\nu}\dot{\theta}$ is not identically equal to τ_{PTO}, it is a better descriptor than -$\nu\dot{\theta}$.

6.4 Results and discussion

In this section, we present experimental results from the PTO Sweeps summarized in Table 6.1. We discuss how emulating flap buoyancy and applying a friction correction affects flap kinematics and performance, with goals of assessing how well the control schemes mitigate the scale effects, as well as evaluating their impact on behavior and performance.

6.4.1 Buoyancy

First, we evaluate the effect of flap buoyancy on kinematics (flap position and velocity) and performance. For these results, we consider four PTO sweeps with different amounts of added buoyancy: $\kappa = 0, 5, 7.5$, and 10 (Sweeps 1-4 in Table 6.1, respectively) and compare the time-average flap position, velocity, and CWR. Additionally, for each parameter, we consider time-resolved dynamics at two effective damping coefficients: $\hat{\nu} = 18$ Nms, the case with the smallest damping, and $\hat{\nu} = 71$ Nms, the case near optimal damping that maximizes CWR.

Kinematics

Figure 6.6 summarizes the effect of buoyancy on flap position. We consider the mean flap position (i.e., the angle the flap oscillates about, with $\theta = 0$ degrees corresponding to vertical) and the position range (maximum and minimum flap position) for each experiment in Sweeps 1-4. Without additional buoyancy ($\kappa = 0$), the mean flap position increases significantly with the effective damping coefficient, $\hat{\nu}$. While this trend is present for all damping sweeps regardless of flap buoyancy, it is most pronounced for $\kappa = 0$. Increasing buoyancy ($\kappa \geq 5$) keeps oscillations centered close to vertical ($\theta = 0$) and increases oscillation amplitude (wider position range). However, increasing the buoyancy beyond $\kappa = 5$ has limited improvement on symmetry, especially for lower damping levels. With these results, we demonstrate that this mitigation technique is effective for emulating flap buoyancy by keeping the oscillation axis closer to vertical and increasing the total position range of the flap.

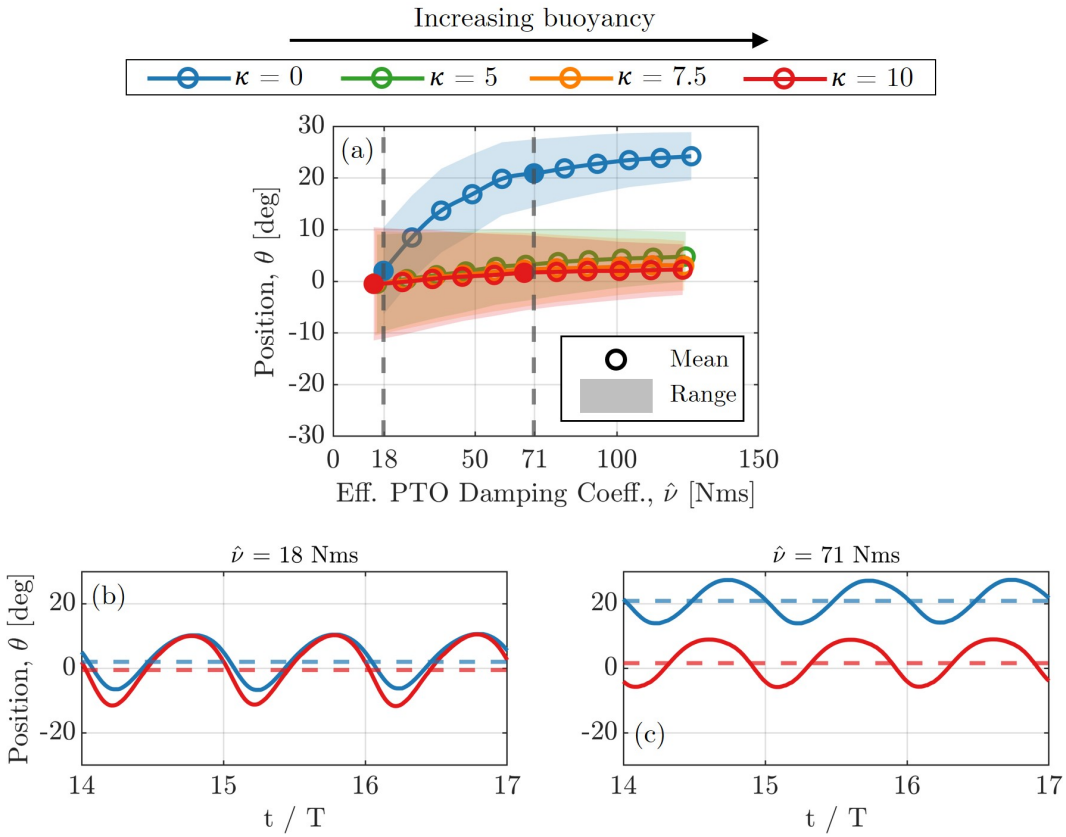

Figure 6.6: (a) Mean flap position (solid lines) and range (shaded region) as a function of effective damping coefficient, $\hat{\nu}$ for PTO Sweeps 1-4. Color corresponds to added buoyancy, with blue representing no added buoyancy ($\kappa = 0$) and red representing an additional 10 times the nominal buoyancy ($\kappa = 10$), described by Equation 6.1. Position time series for (c) $\hat{\nu} = 18$ Nms and (d) $\hat{\nu} = 71$ Nms for the first three periods of steady-state data. Dashed horizontal lines represent the time-average, steady state flap position for each test, as reported in (a). Dashed vertical lines in (a) correspond to the tests represented in (b) and (c).

Next, we explore how buoyancy affects flap velocity (Figure 6.7). Velocity magnitude increases with buoyancy for almost all damping values (Figure 6.7a). Given that oscillation period is constant, this is consistent with observed increases in the flap position range with

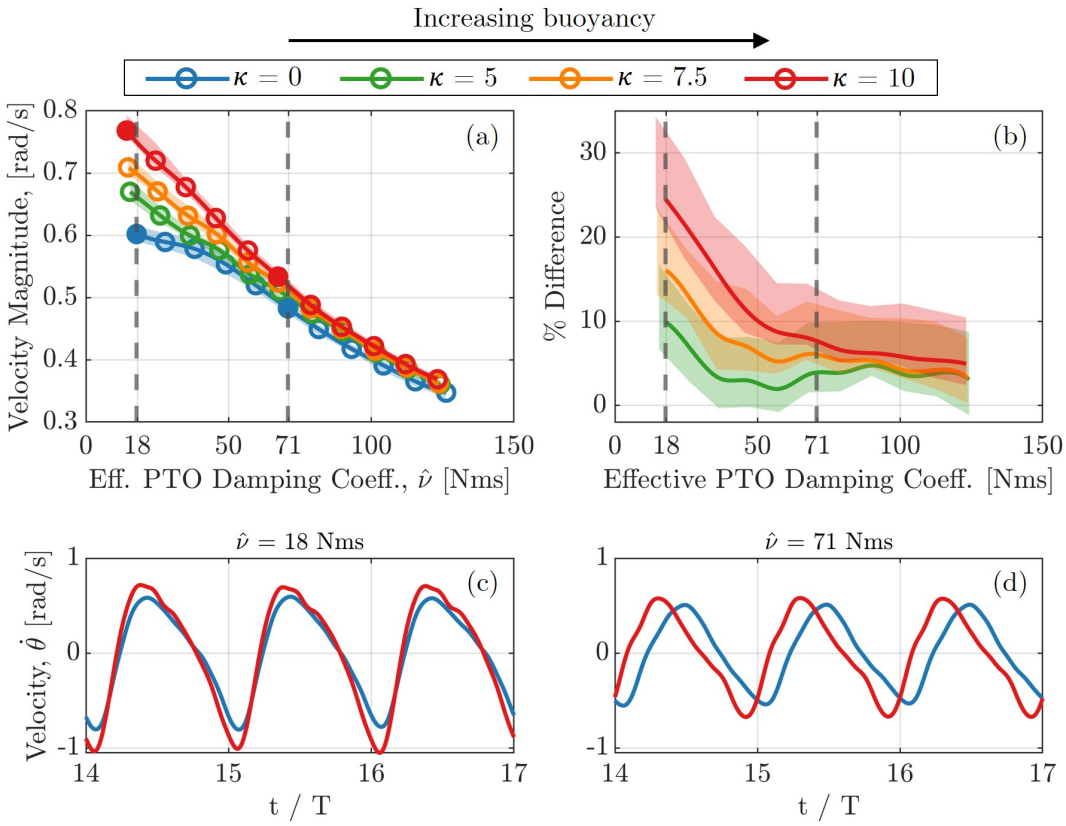

Figure 6.7: (a) Velocity magnitude (solid lines) and range (shaded region) as a function of effective damping coefficient, $\hat{\nu}$ for PTO Sweeps 1-4. Color corresponds to the amount of added buoyancy. (b) Percent difference in velocity magnitude for PTO Sweeps 2-4 compared to Sweep 1. Shaded region uses the velocity magnitude range in (a) to calculate the largest range in the percent difference for each Sweep. Velocity time series for (c) $\hat{\nu} = 18$ Nms and (d) $\hat{\nu} = 71$ Nms over the first three periods in the steady-state region.

buoyancy. However, velocity magnitude is less affected by buoyancy as damping increases, becoming nearly independent of buoyancy at the highest damping (Figure 6.7b). This is in contrast to mean flap position, which is strongly affected by buoyancy at high effective damping. Finally, we note that the range of velocity amplitude is relatively small in comparison to the average and decreases as damping increases, indicating that these results are

relatively insensitive to variations.

Velocity time series (Figure 6.7c) show that changing flap buoyancy can affect both velocity magnitude and phase, depending on effective damping. With relatively low damping, increasing buoyancy increases velocity magnitude, but does not affect phase. In contrast, near optimal damping, the velocity magnitude is relatively insensitive to buoyancy, but velocity phase is altered. Because the time series are aligned by their respective incident waves, emulating buoyancy changes the phase of velocity relative to the incident wave, essentially acting as a form of phase control, which we explore in the following section.

Performance

Figure 6.8 summarizes OSWEC performance changes as a function of buoyancy. Fewer tests are shown for in Sweeps 2-4 than Sweep 1 due to torque measurements exceeding the calibrated cell range. Unsurprisingly, because $P_a = \tau_{\mathrm{PTO}}\dot{\theta} \approx \hat{\nu}\dot{\theta}^2$, the overall trends follow those of velocity magnitude, but are exaggerated because power scales with the square of velocity. For all damping values, increasing flap buoyancy consistently increases CWR (Figure 6.8a), though the marginal benefit decreases as the effective damping coefficient increases (Figure 6.8b).

The additional buoyant torque provided by the PTO is most obvious in the time series with the lowest effective damping (Figure 6.8c), where the PTO periodically injects power into the emulated spring (negative values for $\kappa = 10$). Because the additional buoyant torque is conservative (i.e., does not do net work over an oscillation), this does not inherently alter CWR. Rather, the increase in CWR is caused by changes to the flap dynamics from buoyancy. As for flap velocity, increasing buoyancy near optimal damping changes the phase of absorbed power (Figure 6.8d). In other words, emulating additional flap buoyancy using the PTO is a form of phase control, which seeks to increase CWR by aligning the phase of flap velocity and excitation torque such that the WEC is in resonance [5]. To investigate this further, we consider the natural period of the flap, T_n,

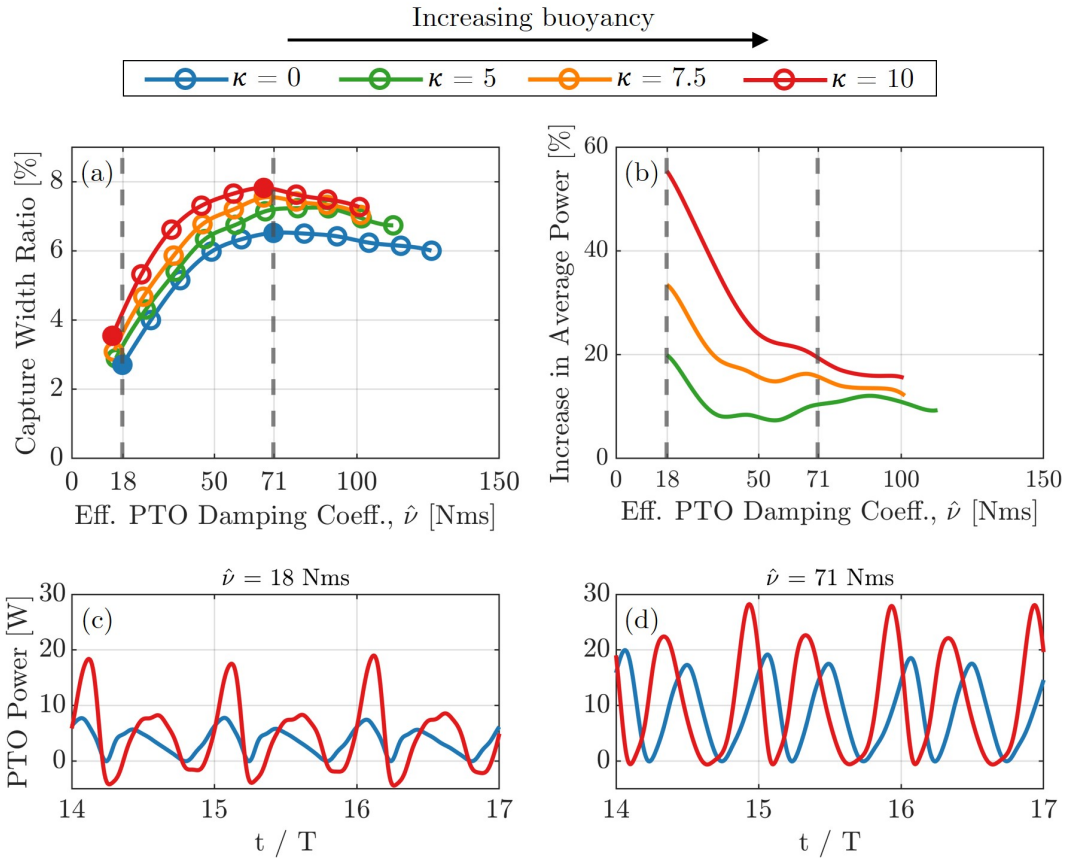

Figure 6.8: (a) Capture width ratio (CWR) as a function of effective damping coefficient for Sweeps 1-4. Color corresponds to the amount of added buoyancy. (b) Percent increase in CWR for Sweeps 2-4 relative to Sweep 1. Time series of absorbed PTO power for (c) $\hat{\nu} = 18$ Nms and (d) $\hat{\nu} = 71$ Nms over the first three periods in the steady-state region.

$$T_n = \frac{1}{2\pi}\sqrt{\frac{I + I_a}{\kappa C}}, \tag{6.5}$$

where C is the linear flap spring stiffness (linear term in Equation 6.1), I is the flap moment of inertia, and I_a is the flap added inertia. Table 6.2 gives the natural period of the flap for all four buoyancy coefficients: $\kappa = 0, 5, 7.5$, and 10. The moment of inertia of the flap is 3.43 kg m^2 (calculated from in-air forced oscillation with negligible fluid forces), and the added inertia

Table 6.2: OSWEC natural period as a function of buoyancy coefficient, κ

κ	Natural period, T_n (s)
0	10.8
5	4.4
7.5	3.7
10	3.3

is 20.6 kg m^2 (calculated using WAMIT [115], Appendix A). Adding 10 times the nominal flap buoyancy ($\kappa = 10$) reduces the natural period of the flap from 10.8 to 3.3 seconds. While this is still greater than the 1.5 second wave period, the flap is operating closer to resonance. Similarly, the influence of resonance on CWR decreases with damping, consistent with the declining marginal benefit from additional buoyancy as damping increases.

Overall, emulating higher flap buoyancy using the PTO reduces flap drift and increases velocity magnitude and CWR. This is because the flap is operating closer to resonance due to an increase in the natural period of the flap. This suggests that sensor weight in laboratory experiments can reduce performance relative to a full-scale system and should be considered and addressed in experimental setups. Emulating buoyancy is an effective way to address scale effects, as well as explore the effect of buoyancy on device performance without physical alterations.

6.4.2 Friction

We begin our evaluation of the role of driveline friction by demonstrating the effectiveness of the correction term (Equation 6.3). Figure 6.9 compares the effective and nominal damping coefficient for PTO Sweeps 1 (no friction correction) and 5 (friction correction). When friction is present in the system, the effective damping coefficient is consistently higher than the nominal. However, when the correction is applied, the nominal and effective damping

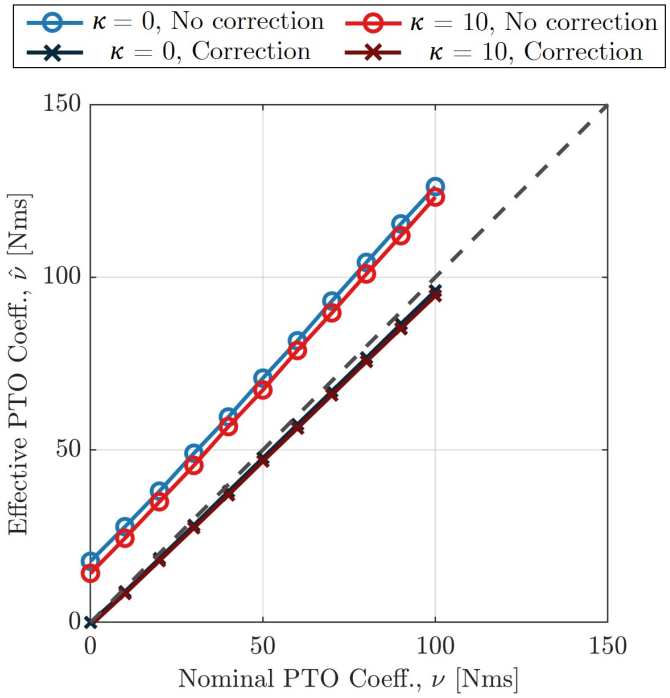

Figure 6.9: Nominal vs. effective damping coefficient. The light blue and red lines with circle markers represent PTO Sweeps 1 and 4, respectively (no friction correction applied), while the dark blue and red lines with cross markers represent PTO Sweeps 5 and 6, respectively (friction correction applied). Dashed line denotes parity between the effective and nominal coefficients.

coefficients are approximately equal, indicating that the contribution from friction has been offset.

Kinematics

Figure 6.10 summarizes the effect of applying the friction correction on flap position. We consider the same parameters as the buoyancy analysis, including mean flap position and position range (Figure 6.6). However, because there are fewer tests that resulted in similar effective damping coefficients for all four cases, we only compare time series at a near-optimal

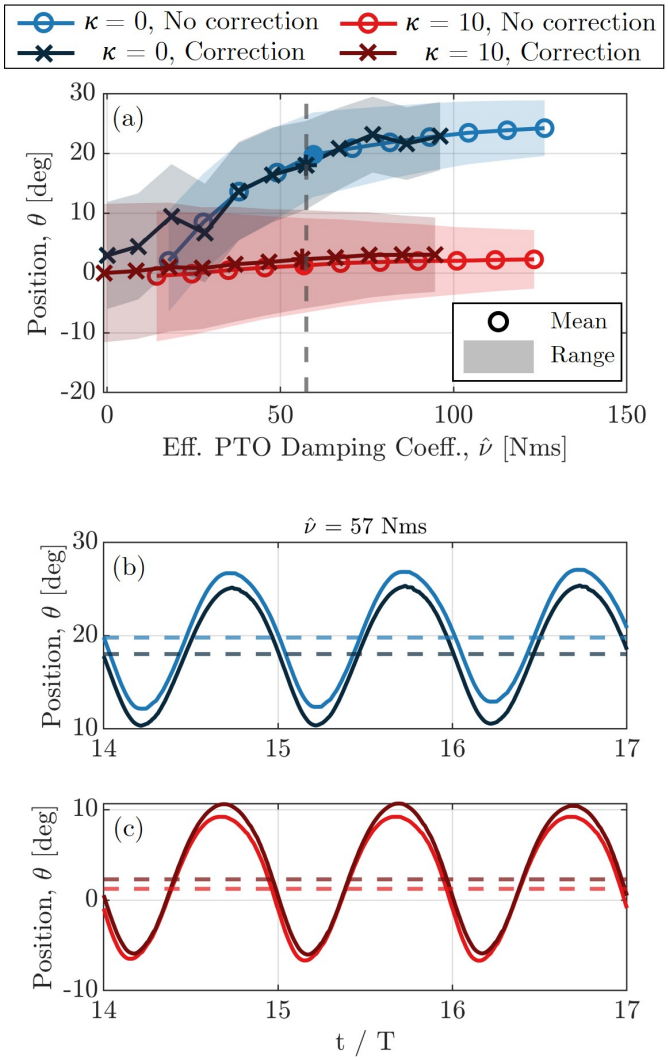

Figure 6.10: (a) Mean flap position (solid lines) and position range (shaded region) as a function of effective damping coefficient, $\hat{\nu}$ for PTO Sweeps 1, 4, 5, and 6. Position time series for $\hat{\nu} = 57$ Nms for tests with (b) $\kappa = 0$ and (c) $\kappa = 10$ over the first three periods in the steady-state region. Dashed horizontal lines represent the steady state mean flap position for each test as reported in (a). Dashed vertical line in (a) corresponds to the tests represented in (b) and (c).

damping coefficient: $\hat{\nu} = 57$ Nms. The friction correction does not significantly change either mean position or position range (Figure 6.10a). Time series of flap position near optimal damping show similar invariance. At the baseline buoyancy ($\kappa = 0$), the friction correction slightly reduces the offset from vertical (Figure 6.10b), but this is consistent with the slight difference in effective damping coefficient between the two tests.

Performance

Finally, Figure 6.12 summarizes flap performance with and without the friction correction. Similar to velocity, the friction correction does not significantly affect flap performance. For $\kappa = 10$, there is an increase in CWR for higher damping coefficients for tests with the friction correction over those without (Figure 6.12a), which mirrors the slight increase in velocity magnitude in Figure 6.11a). Beyond this, there is minimal difference between tests with and without the correction at the same effective damping coefficient, either in a time-average or time-resolved sense (Figure 6.12b).

Overall, driveline friction does not significantly affect flap kinematics or performance, as long as it is accounted for through an effective damping coefficient. Consequently, this scale effect should not reduce the applicability of laboratory-scale results to understand full-scale systems.

6.5 Conclusions

In this chapter, we identify scale effects in our experimental setup including artificially low flap buoyancy due to sensor weight and driveline friction. To mitigate these scale effects, we first characterize the net restoring torque as a function of position and the friction profile of the driveline. Based on these characterizations, we use a motor to emulate additional flap buoyancy and offset driveline friction using real-time position and torque measurements. By comparing experiments with and without these corrections, we asses the impact of these scale effects on flap kinematics and OSWEC performance.

We find that increasing flap buoyancy results in larger oscillation amplitudes that are

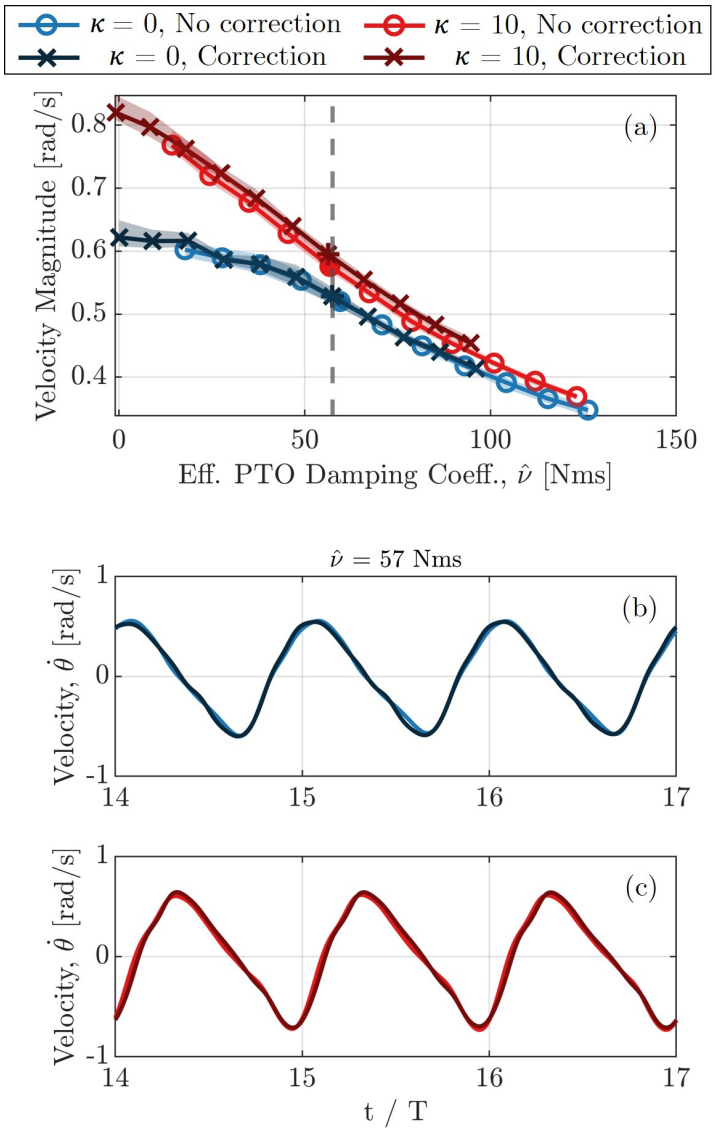

Figure 6.11: (a) Velocity magnitude (solid lines) range (shaded region) as a function of effective damping coefficient, $\hat{\nu}$ for PTO Sweeps 1, 4, 5, and 6. Velocity time series for $\hat{\nu} = 57$ Nms for tests with (b) $\kappa = 0$ and (c) $\kappa = 10$ over the first three periods in the steady-state region. Dashed vertical line in (a) corresponds to the tests represented in (b) and (c).

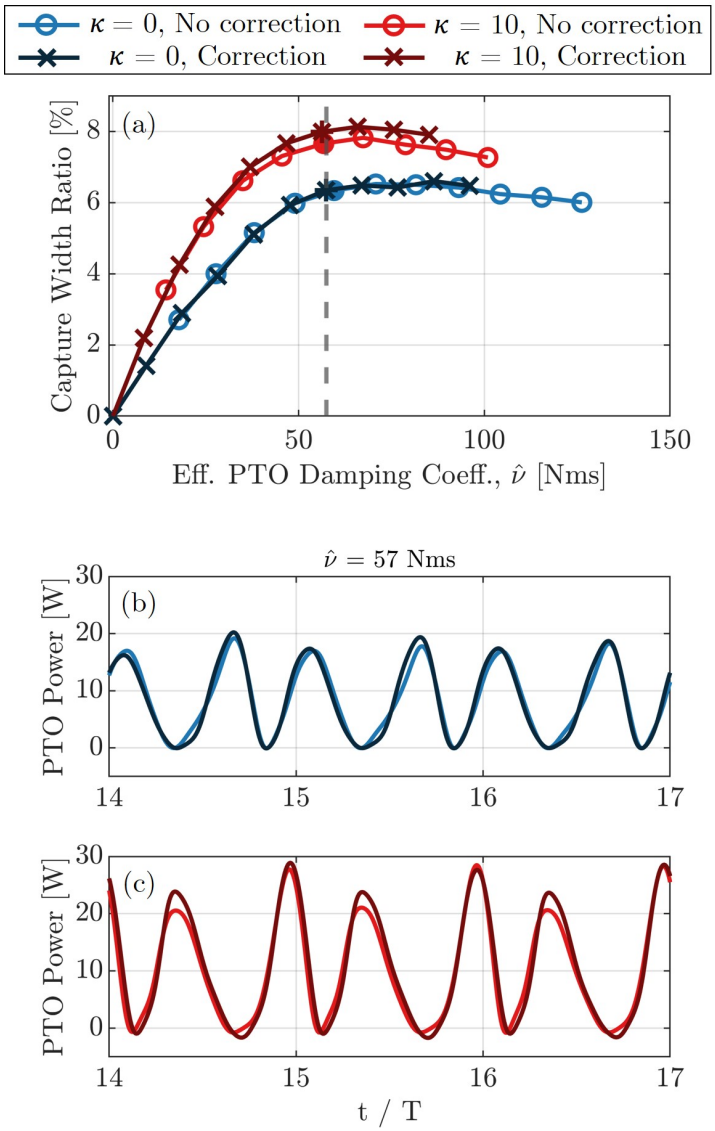

Figure 6.12: (a) Capture width ratio as a function of effective damping coefficient, $\hat{\nu}$, for PTO Sweeps 1, 4, 5, and 6. PTO power time series for $\hat{\nu} = 57$ Nms for tests with (b) $\kappa = 0$ and (c) $\kappa = 10$ over the first three periods in the steady-state region. Dashed vertical line in (a) corresponds to the tests represented in (b) and (c).

symmetric about the vertical position. Consequently, velocity magnitude and capture width ratio also increase with flap buoyancy. However, these changes in flap kinematics and perfor-

mance are most prominent at low damping and decrease at higher damping because buoyancy torque plays a relatively smaller role in flap dynamics. Increasing flap buoyancy with the PTO essentially acts as a form of phase control by reducing the natural period of the flap such that the device operates closer to resonance. We found that emulating flap buoyancy using a PTO is an effective way of adjusting mass parameters to offset the role of sensor weight with software rather than hardware. This enables rapid data collection without physical changes (e.g., fabricating a new flap, removing sensors), resulting in a low-cost method that not only addresses scale effects, but also allows for exploration of design spaces to optimize performance. In future work, we plan to further investigate the phase shift between the velocity and incident wave as a function of buoyancy coefficient and effective damping. Because phase control and resonance estimates use linear representations of device dynamics, such experiments would also assess the validity of these assumptions using non-ideal experimental data.

Additionally, we found that friction does not have a significant effect on flap kinematics and performance, as long as it is accounted for in the effective damping coefficient. This shows that friction does not necessarily limit the applicability of experiments to interpreting full-scale systems. However, this result only strictly holds for a linear damping control scheme and driveline friction that behaves similarly to a linear damper. If this is not the case, a more advanced controller would be required to offset the role of friction, though this could still be implemented through real-time measurements of friction.

Chapter 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

Creating accurate and computationally efficient models of WEC behavior is an important step in advancing their development. However, the tradeoffs between accuracy and computational efficiency for linear theory and high-fidelity simulations limit the application of many modeling techniques. Because of this, experiments can be an important alternative to simulation, but "scale effects" such as friction and sensor weight can affect the accuracy of small-scale testing. In this work, we address these limitations by introducing novel data-driven modeling techniques that accurately describe OSWEC dynamics and demonstrating experimental methods that address scale effects using control.

First, in Chapter 4, we used dynamic mode decomposition (DMD), a linear data-driven algorithm, to model OSWEC dynamics when considering three common modeling barriers: noisy signals, weak nonlinearity, and irregular wave forcing. The DMD algorithm was trained on linear WEC-Sim data and included states such as position, velocity, surge force, hydrodynamic torque, and wave pressure. We employed an extension of DMD to address each modeling barrier and compared the results to that of the unmodified algorithm (referred to as exact DMD). First, we investigated how signal noise affects DMD accuracy when modeling OSWEC dynamics in a polychromatic sea state. DMD accuracy decreases with increasing signal noise, causing a spurious temporal decay that cannot be remedied with additional data. We compared exact DMD to total-least-squares DMD (TLS DMD), which generated accurate system models, even in the presence of noise. TLS DMD did exhibit appreciable error for tests with more severe signal noise (SNR $\leq$ 30), particularly in the testing region. However, unlike the errors from the exact DMD model, this can be mitigated by including

more training data in the data matrices. The next barrier we considered was modeling weakly nonlinear dynamics. We generated the training data by including nonlinear Froude-Krylov and buoyancy torques in the WEC-Sim model and compared the performance of exact DMD and DMD with a time delay (i.e., including a time-shifted copy of the measurements in the data matrices). Including a time delay significantly improved model accuracy by providing additional phase information and increasing the available rank of the data matrices. The final modeling barrier we considered was irregular wave forcing. Because of the complexity of the dynamics, we used DMD to model the spectrotemporal behavior of the state variables, rather than their time-resolved dynamics. We compared exact DMD with optimized DMD (optDMD), which uses variable projection techniques to choose optimal DMD eigenvalues and eigenvectors using the full data matrix, rather than the first snapshot. While optDMD significantly outperforms exact DMD in the training region, neither model was able to predict the spectrotemporal behavior in the testing region. Overall, DMD is a promising tool for WEC modeling, however this study considered primarily linear dynamics, which limits its application in more energetic sea states.

In Chapter 5, we address this limitation by modeling experimental data generated from testing a laboratory-scale OSWEC in the SWEL wave tank at NREL. This data exhibited nonlinearities from multiple sources, including large rotations due to limited flap buoyancy, confinement effects, and wave overtopping, the severity of which varied throughout the control regime. We used sparse identification of nonlinear dynamics (SINDy) to generate models for surge and heave force as a function of flap kinematics. SINDy generated accurate models for surge and heave force for all tests, regardless of hydrodynamic complexity. While the SINDy models for surge force all only included a single term proportional to flap velocity, the heave models varied in structure and included a second-order term for tests with demonstrable nonlinearity. Based on the identified models, there are fundamental changes in the heave force dynamics when overtopping and confinement effects occur. As expected, both linear WEC-Sim and SINDy are accurate at low damping where nonlinearity is minimal. However, SINDy outperforms WEC-Sim when power takeoff damping is relatively high and

higher-order hydrodynamics are present. Although the resulting SINDy models provide some insight into how the identified terms relate to the physical phenomena, no definitive conclusions could be drawn due to the coupled nonlinearities present in the experiments. Finally, we explored how well the SINDy models can generalize to data from tests they were not trained on. For surge force, we were able to accurately describe all tests with a single expression that includes flap velocity and the PTO damping coefficient. However, heave force requires multiple models to describe the dynamics accurately over the control regime, reinforcing that there are changes in the heave dynamics that arise from the nonlinear phenomena.

Finally, in Chapter 6, we use the same experimental setup to characterize, mitigate, and assess the impact of experimental scale effects. Specifically, we address the reduction in flap buoyancy caused by pressure sensor weight and seal friction in the driveline. Rather than minimizing these effects with physical changes (i.e., removing sensors or altering the driveline), we applied a control torque that emulates additional flap buoyancy and offsets the seal friction over a range of control parameters. To address the reduced buoyancy, we first characterized flap restoring torque profile as a function of position. Then, to emulate buoyancy during wave testing, we applied a control torque of the same profile using real-time position and torque feedback. Increasing flap buoyancy significantly improved the symmetry of flap oscillation about its vertical axis, as well as increased the velocity magnitude and capture width ratio (CWR), especially at lower effective damping values. Beyond addressing scale effects, using a controller to increase flap buoyancy is also a form of phase control, because it reduces the natural period of the device such that it operates closer to resonance. Based on these results, the change in mass properties due to sensors should be considered in scaled experiments to ensure device performance is properly characterized. To offset the seal friction, we first characterize the friction profile as a function of the free load cell measurements for an uncontrolled test. The motor then applied a torque to offset the friction using real-time torque and kinematic feedback. Unlike buoyancy, friction does not significantly affect flap kinematics or performance, as long as it is accounted for in the effective damping coefficient. This suggests that friction does not necessarily limit small-scale experiments, but demon-

strates the impact of real-time sensing and feedback control to understand its role. Overall, using software to address scale effects can be an effective method to efficiently collect a wide range of data without the difficulties and expense of changing the device hardware.

Modeling WEC behavior is complicated and nuanced. All modeling methods come with limitations that are important to address when considering their use and application. With this work, we expand the established methods of OSWEC modeling with novel data-driven and experimental techniques that provide an accurate and efficient way to describe complex WEC dynamics.

7.2 Future Work

There are multiple avenues of future research that can expand the applications of this work, including:

- **Improving the OSWEC controller.** As shown in Chapter 6, driveline friction is complex and may change with control parameters, but can be offset if modeled correctly. Therefore, we aim to implement a PID controller that minimizes the real-time error between the commanded torque and the torque that is being applied to the flap (i.e., torque measured by both load cells). This will expand our control capabilities and ensure that future models can accurately incorporate a wider range of control inputs.

- **Expanding the experimental campaign.** The experimental results presented in Chapters 5 and 6 are limited to a single regular wave and linear damping control. However, to fully characterize the system and systematically study the linear and nonlinear hydrodynamics, we aim to perform experiments with a wider range of wave and control parameters. Ideally, these experiments will be conducted in a wider wave tank to minimize confinement artifacts.

- **Building DMD and SINDy models with control inputs.** (i.e., DMD with con-

trol [65] and SINDy with control [66]). The procedure for building these models will be similar to those performed in Chapters 4 and 5, however the data matrices in each algorithm will include control inputs in addition to state variables. This is is a critical step for implementing the models in optimal control. Additionally, we plan to use SINDy to build full differential equations to allow for future time prediction, rather than just state estimation, as is the case in Chapter 5. This does not require significant changes to the procedure, but will involve including numerical derivatives of the state variables as modeled variables in Equation D.2. In doing so, we can numerically integrate the system to predict the dynamics past the training region (demonstrated in Appendix D). This step is critical to implement the models in optimal control, and will also allow us to learn more about the underlying system and investigate the nonlinearity explored in Chapter 5.

- **Incorporating data-driven models into an optimal controller.** Finally, we aim to implement data-driven models into a model predictive controller (MPC) to optimize device performance. This requires defining a cost function and constraints that can be described using DMDc or SINDyc models. In MPC, the model will predict the system dynamics so that the optimizer can choose the control torque that maximizes power absorption while upholding the constraints. The requirements of the model will depend on the design of the controller, but Fasel *et al.* [67] and Kaiser *et al.* [64] both demonstrated data-driven optimal control with DMDc and SINDyc for other dynamical systems. We will first aim to implement this controller in WEC-Sim, similar to the work in Guerrero-Fernández *et al.* [116], before testing the controller experimentally. Depending on the DMDc or SINDyc model, future information of the incident wave field may be required for accurate state prediction, which has been a common complication in past WEC optimal control studies [94]. However, recent progress by Fisher *et al.* [117] demonstrated promising results in deterministic irregular wave prediction using an array of SWIFT buoys, which we plan to investigate for future control applications.

This is neither a step-by-step guide nor an exhaustive list of the requirements to build a data-driven optimal controller. Instead, this is a starting point, and accomplishing these goals will require iteration between these steps and adjustments based on the results. For example, the control schemes we test experimentally will depend on the design of the model predictive controller and the results from the DMDc and SINDyc models. Overall, we believe the results presented here, in addition to the proposed future work, can help promote WEC development and advance the technology toward successful, long-term deployments.

Appendix A

BEM CALCULATIONS WITH WAMIT

WAMIT [115] is a commercial program that uses boundary element methods (BEM) to calculate hydrodynamic coefficients including added mass/inertia, radiation damping, and excitation force/torque of wave energy converters and other marine structures. After meshing the hydrodynamic body, WAMIT calculates the coefficients using potential flow and linear wave theory. We use WAMIT to determine the hydrodynamic coefficients for the experimental device described in Section 3.1.2 and used in Chapters 5 and 6. We use these results primarily as input to WEC-Sim to model linear time-resolved WEC behavior and compare to nonlinear SINDy models in Chapter 5 (5.8), and to determine the added inertia coefficient to calculate the natural period of the flap in Chapter 6 (Equation 6.5).

To model the experimental device described in Section 3.1.2, we built a two-body simulation in WAMIT including the flap and driveline, but not the base structure. However,

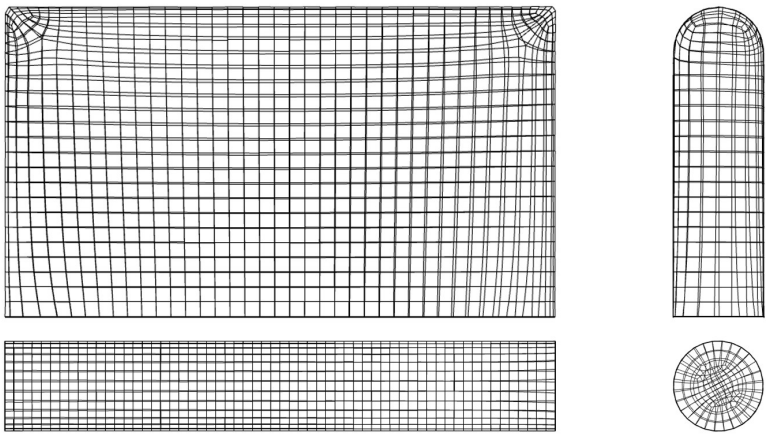

Figure A.1: Mesh of [top] flap and [bottom] driveline used in WAMIT simulations.

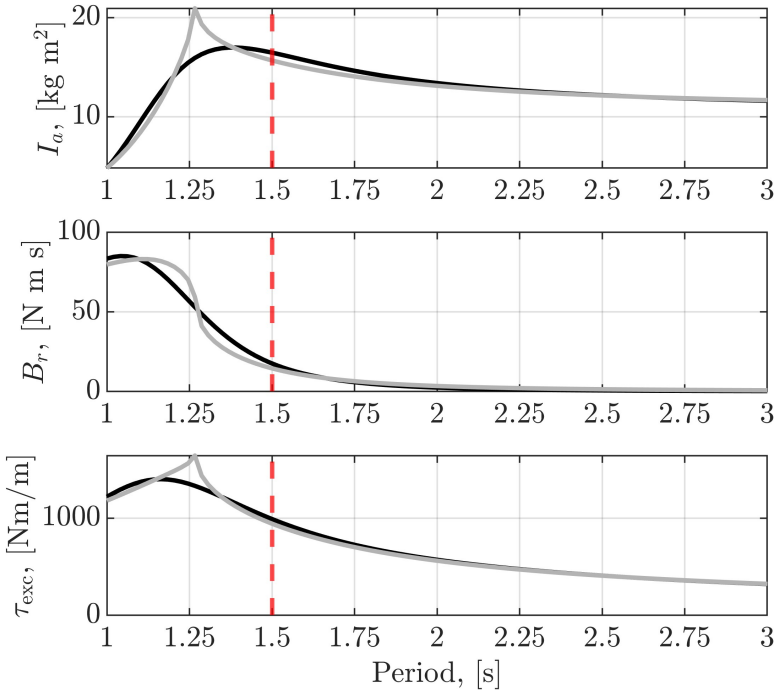

Figure A.2: [Top] Added inertia (I_a), [middle] radiation damping (B_r), and [bottom] excitation torque (τ_{exc}) coefficients calculated from WAMIT as a function of period. Gray and black and lines represent results with and without channel walls, respectively. Red dashed lines denote the wave period used in Chapters 5 and 6.

because the structure is low in the water column relative to wave orbital decay, we expect this simplification to have limited effects on the hydrodynamic coefficients. The flap and driveline mesh were generated from CAD models using Rhino 8 and followed guidelines in Kass [118]. The meshes, shown in Figure A.1, included 1633 quadrilateral elements for the flap and 1922 for the driveline. We corrected for irregular wave frequencies and included channel walls of the same width as the wavetank (2.5 m) in the simulation using methods described in Chapter 10 and 12 in the WAMIT User Manual, respectively [115]. WAMIT calculated the hydrodynamic coefficients for 100 periods evenly spaced between 1 and 3 seconds. Figure A.2 shows the WAMIT hydrodynamic coefficients as a function of period calculated with and without channel walls.

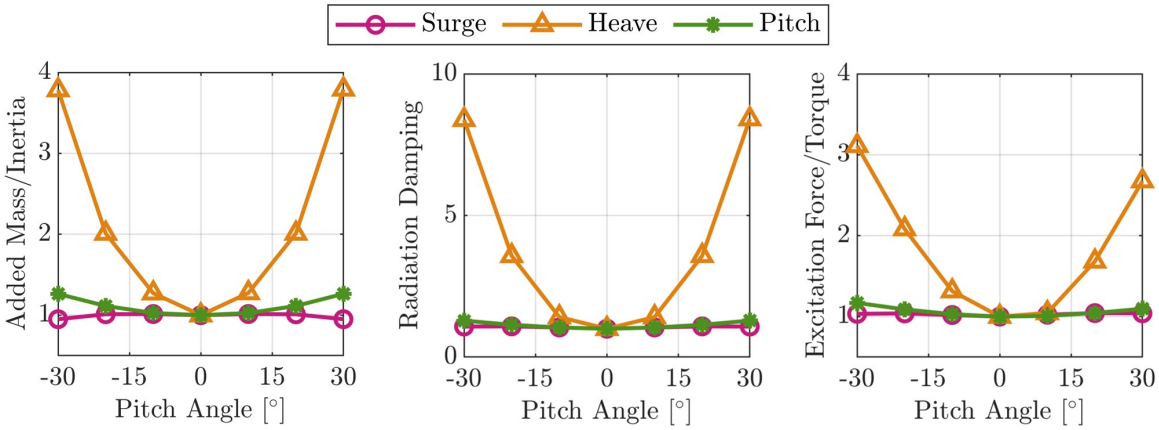

Figure A.3: [Left] Added mass/inertia, [middle] radiation damping, and [right] excitation force/torque coefficients as a function of flap pitch angle for (pink) surge, (yellow) heave, and (green) pitch. All curves are normalized by their value for a vertical flap (0°).

A.1 Angled flap BEM

Because the flap experienced large-amplitude oscillations during experimental testing (Section 5.4.1), we expect that the hydrodynamic coefficients could change as the flap rotates through large angles. Therefore, we built BEM simulations where the flap is rotated off vertical at six angles between ±30° and calculated the hydrodynamic coefficients for surge, heave, and pitch (Figure A.3). By far, heave is most sensitive to changes in pitch angle for all three coefficients. This supports the experimental results in Chapter 5, where heave forces are more sensitive to flap drift than surge. This position-dependence of hydrodynamic parameters could explain the nonlinear term in the heave force SINDy models when flap drift is severe, and inspires future work to investigate the underlying hydrodynamics causing this dependence.

Appendix B

DYNAMIC MODE DECOMPOSITION AND VARIANTS

Here, we provide derivations of the exact DMD algorithm as well as the extensions used in Chapter 4.

B.1 Exact DMD

This appendix includes a detailed derivation of the exact dynamic mode decomposition (DMD) algorithm, as defined in Tu [73]. This algorithm is also described in detail in Kutz *et al.* [74], which includes sample code and data that is available at the accompanying website [98].

Consider a generic linear dynamical system with states $\boldsymbol{x}(t)$:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{\mathcal{A}}\boldsymbol{x}, \tag{B.1}$$

where $\mathcal{A}$ describes the system dynamics in continuous time. To model this system in discrete time, we use the linear operator $\boldsymbol{A} = \exp(\boldsymbol{\mathcal{A}}\Delta t)$ to describe the linear dynamics that shift the system forward from time t_k to t_{k+1}, with $t_k = k\Delta t$:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k, \tag{B.2}$$

where $\boldsymbol{x}_k$ is a column vector of all state measurements at time t_k. We can model Equation B.2 using snapshot data at discrete times arranged into data matrices $\boldsymbol{X}$ and $\boldsymbol{X}^{'}$, such that:

$$\boldsymbol{X}^{'} \approx \boldsymbol{A}\boldsymbol{X}, \tag{B.3}$$

where the data matrices are of the form:

$$\boldsymbol{X} = \begin{bmatrix} | & | & & | \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_m \\ | & | & & | \end{bmatrix}, \quad \boldsymbol{X}^{'} = \begin{bmatrix} | & | & & | \\ \boldsymbol{x}_2 & \boldsymbol{x}_3 & \dots & \boldsymbol{x}_{m+1} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n\times m}. \tag{B.4}$$

The solution to the least-squares regression problem laid out in Equation B.3 can be approximated by finding a matrix $\boldsymbol{A}$ that minimizes $||\boldsymbol{X}^{'} - \boldsymbol{A}\boldsymbol{X}||_F$, where $||\cdot||_F$ is the Frobenius norm. This can be done using the Moore-Penrose pseudoinverse of data matrix $\boldsymbol{X}$, denoted $\boldsymbol{X}^{\dagger}$:

$$\boldsymbol{A} = \boldsymbol{X}^{'}\boldsymbol{X}^{\dagger}. \tag{B.5}$$

We calculate $\boldsymbol{X}^{\dagger}$ using the singular value decomposition (SVD) of data matrix $\boldsymbol{X}$:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{*}, \tag{B.6}$$

where the columns of $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{V} \in \mathbb{R}^{m\times n}$ are the left and right singular vectors of $\boldsymbol{X}$, respectively, and the diagonal of $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$ contains the associated singular values. For large systems, it may be necessary to truncate these matrices and only take the first r singular values and modes to approximate a high-dimensional data matrix, such that:

$$\boldsymbol{X} \approx \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^{*}, \tag{B.7}$$

where $\boldsymbol{U}_r \in \mathbb{R}^{n\times r}$, $\boldsymbol{\Sigma}_r \in \mathbb{R}^{r\times r}$, and $\boldsymbol{V}_r \in \mathbb{R}^{m\times r}$. Because $\boldsymbol{X}$ is real, $\boldsymbol{U}_r$ and $\boldsymbol{V}_r$ are orthogonal, which means the psuedoinverse is easily calculated as $\boldsymbol{X}^{\dagger} = \boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-1}\boldsymbol{U}_r^{*}$:

$$\boldsymbol{A} = \boldsymbol{X}^{'}\boldsymbol{X}^{\dagger} = \boldsymbol{X}^{'}\boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-1}\boldsymbol{U}_r^{*} \in \mathbb{R}^{n\times n}. \tag{B.8}$$

The eigenvectors of $\boldsymbol{A}$ are the DMD modes, $\boldsymbol{\phi}_j$, with corresponding eigenvalues λ_j that represent their growth/decay rates and frequencies. For large systems, working with (n $\times$ n) matrix $\boldsymbol{A}$ is too computationally expensive. To ease the computational burden, we project $\boldsymbol{A}$ onto the POD modes described by $\boldsymbol{U}_r$:

$$\boldsymbol{A}_p \doteq \boldsymbol{U}_r^{*}\boldsymbol{A}\boldsymbol{U}_r = \boldsymbol{U}_r^{*}\boldsymbol{X}^{'}\boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-1} \in \mathbb{R}^{r\times r}, \tag{B.9}$$

where $\boldsymbol{A}_p$ is the projected linear operator in the POD basis. To approximate the DMD modes in the POD basis, we set up an eigenvalue problem with $\boldsymbol{A}_p$:

$$\boldsymbol{A}_p\boldsymbol{W} = \boldsymbol{W}\boldsymbol{D}, \tag{B.10}$$

where the columns of $\boldsymbol{W}$ are the eigenvectors of $\boldsymbol{A}_p$, and $\boldsymbol{D}$ contains the eigenvalues of both $\boldsymbol{A}_p$ and $\boldsymbol{A}$. As described by [73], the exact DMD modes are the columns of $\boldsymbol{\Phi}$, where:

$$\boldsymbol{\Phi} = \boldsymbol{X}^{'}\boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-1}\boldsymbol{W}. \tag{B.11}$$

With the DMD eigenvectors and eigenvalues, we can now describe the system dynamics:

$$\boldsymbol{x}(t) \approx \sum_{j=1}^{r} \boldsymbol{\phi_j}\exp(\gamma_j t)b_j = \boldsymbol{\Phi}\exp(\boldsymbol{\Gamma}t)\boldsymbol{b}, \tag{B.12}$$

where $\gamma_j = \ln(\psi_j)/\Delta t$ are the continuous-time DMD eigenvalues and $\boldsymbol{b} = \boldsymbol{\Phi}^{\dagger}\boldsymbol{x}_1$ is the initial time vector translated to the DMD eigenbasis. In practice, it is likely the rank of the system cannot be known a priori, so the number of SVD modes to include is a free variable that may need adjusting to optimize accuracy and run time for the algorithm. With Equation B.12, we can now describe and predict system state dynamics using only snapshot data.

B.2 Total-least-squares DMD

This appendix includes details of the total-least-squares DMD algorithm introduced in Section 4.3.3 and used in Section 4.4.1. We refer the reader to Hemati *et al.* [87] for a full derivation, and to Kutz *et al.* [74, 98] for example code and data.

Consider a new combined data matrix $\boldsymbol{X}_{TLS}$ and its SVD:

$$\boldsymbol{X}_{TLS} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{X}^{'} \end{bmatrix} = \boldsymbol{U}_{TLS}\boldsymbol{\Sigma}_{TLS}\boldsymbol{V}_{TLS}^{*}. \tag{B.13}$$

To account for noise in both $\boldsymbol{X}$ and $\boldsymbol{X}^{'}$, we project the data matrices onto a basis created by the right singular vectors of $\boldsymbol{X}_{TLS}$, $\boldsymbol{V}_{TLS}$:

$$\breve{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{V}_{TLS}\boldsymbol{V}_{TLS}^{*}, \quad \breve{\boldsymbol{X}}^{'} = \boldsymbol{X}^{'}\boldsymbol{V}_{TLS}\boldsymbol{V}_{TLS}^{*}. \tag{B.14}$$

We now have a modified regression problem:

$$\breve{\boldsymbol{X}}^{'} = \breve{\boldsymbol{A}}\breve{\boldsymbol{X}}, \tag{B.15}$$

and can calculate the DMD matrix $\breve{\boldsymbol{A}}$ using the SVD of $\breve{\boldsymbol{X}}$:

$$\breve{\boldsymbol{A}} = \breve{\boldsymbol{U}}^{*}\breve{\boldsymbol{X}}'\breve{\boldsymbol{V}}\breve{\boldsymbol{\Sigma}}^{-1}, \tag{B.16}$$

where $\breve{\boldsymbol{U}}$ and $\breve{\boldsymbol{V}}$ and right singular vectors of $\breve{\boldsymbol{X}}$, respectively, and $\breve{\boldsymbol{\Sigma}}$ contains the singular values of $\breve{\boldsymbol{X}}$, such that $\breve{\boldsymbol{X}} = \breve{\boldsymbol{U}}\breve{\boldsymbol{\Sigma}}\breve{\boldsymbol{V}}^{*}$. We can now follow the original DMD algorithm, replacing the original linear operator $\boldsymbol{A}$ with $\breve{\boldsymbol{A}}$.

B.3 Time delays

This appendix explains how to modify the data matrices using time delays to increase the rank of the data matrices and gain important phase information [73].

By stacking matrices that are shifted by a time step, we obtain new data matrices:

$$\boldsymbol{X}_{TD} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{X}' \end{bmatrix}, \quad \boldsymbol{X}'_{TD} = \begin{bmatrix} \boldsymbol{X}' \\ \boldsymbol{X}'' \end{bmatrix}, \tag{B.17}$$

where $\boldsymbol{X}''$ follows the same pattern as $\boldsymbol{X}'$, but is shifted two time steps rather than one. $\boldsymbol{X}_{TD}$ and $\boldsymbol{X}'_{TD}$ replace $\boldsymbol{X}$ and $\boldsymbol{X}'$ in the exact DMD algorithm in Section 4.3.3, respectively. Since the maximum rank we can consider when reconstructing $\boldsymbol{x}(t)$ is the length of the column vectors in the data matrices, time delays are particularly useful when higher rank is required for fidelity, but the number of sensors is limited. For example, a single time delay doubles the length of the column vectors in the data matrices, which doubles the number of SVD modes we can use to model the dynamics. Adding a shifted time matrix also provides more phase information, which can improve reconstruction accuracy, especially for the case of standing waves [73]. However, there are diminishing returns when using time delays that occur when adding more time-shifted copies only increases the number of columns in $\boldsymbol{X}$ without increasing its rank.

B.4 Optimized DMD

This appendix provides a brief derivation of the main ideas of optDMD that is introduced in Section 4.3.3 and used in Section 4.4.3. For a complete derivation as well as open-source

code, we refer the reader to Askham and Kutz [103].

To begin, we assume our data $\boldsymbol{X}$ is a solution of a linear system of r differential equations, where r is the chosen rank, which means we can express $\boldsymbol{X}$ as a combination of exponentials,

$$\boldsymbol{X}^T \approx \boldsymbol{\Omega}(\boldsymbol{\alpha})\boldsymbol{B}, \tag{B.18}$$

where $\boldsymbol{\Omega} \in \mathbb{C}^{m\times r}$ is a matrix of exponentials, such that $\Omega(\boldsymbol{\alpha})_{i,j} = \exp(\alpha_j t_i)$, $\boldsymbol{\alpha}$ are the equivalent of continuous-time DMD eigenvalues, and $\boldsymbol{B} \in \mathbb{C}^{r\times n}$ are the weights of the exponential functions. To find the optimal values of $\boldsymbol{\alpha}$ and $\boldsymbol{B}$ that best approximates our data matrix $\boldsymbol{X}$, we want to solve the following nonlinear least squares problem:

$$\min_{\boldsymbol{\alpha},\,\boldsymbol{B}} \left|\left| \boldsymbol{X}^T - \boldsymbol{\Omega}(\boldsymbol{\alpha})\boldsymbol{B} \right|\right|_F, \tag{B.19}$$

which is achieved using variable projection methods. Due to the complexity of solving this optimization problem, we will not outline the process for solving for $\boldsymbol{\alpha}$ and $\boldsymbol{B}$ and refer the reader to Askham and Kutz [103] for the full derivation. In Eq. B.19, F denotes the Frobenius norm. Once we have $\boldsymbol{\alpha}_{opt}$ and $\boldsymbol{B}_{opt}$ that minimizes Equation B.19, we can calculate the DMD eigenvalues and eigenvectors:

$$\gamma_j = \boldsymbol{\alpha}_{opt,j}, \quad \boldsymbol{\phi}_j = \frac{1}{\left|\left|\boldsymbol{\beta}_{opt,j}\right|\right|_2}\boldsymbol{\beta}_{opt,j}, \tag{B.20}$$

and estimate system dynamics:

$$\boldsymbol{x}(t) = \sum_{j=1}^{r} \left|\left|\boldsymbol{\beta}_{opt,j}\right|\right|_2 \exp(\gamma_j t)\boldsymbol{\phi}_j, \tag{B.21}$$

where $\boldsymbol{\beta}_{opt,j}$ is the jth column of $\boldsymbol{B}_{opt}^T$.

Appendix C

EXPERIMENTAL POST-PROCESSING PROCEDURES

Here, we describe details on the data collection and processing for the encoder and load cells used in the experiments from Chapters 5 and 6. These procedures were primarily developed by Ama Hartman and Greg Talpey.

C.0.1 Encoder data

Before testing, we take an initial tare to set the vertical flap position as $\theta = 0°$. To process kinematic data, we perform the following steps:

1. Apply the tare value (in counts) to enforce $\theta = 0°$ as vertical.

2. Convert the data from encoder counts to flap position in radians using the encoder resolution and motor gearbox ratio.

3. Calculate velocity and acceleration using instantaneous derivatives of position:

$$\dot{\theta}_n = \frac{\theta_{n+1} - \theta_n}{\Delta t}, \tag{C.1}$$

$$\ddot{\theta}_n = \frac{\dot{\theta}_n - \dot{\theta}_{n-1}}{\Delta t}, \tag{C.2}$$

   where Δt is the time step between samples. As a result of these expressions, the calculated velocity and acceleration time series are sampled at slightly different times than the original position signal and are different lengths. This is rectified in the final step of post-processing.

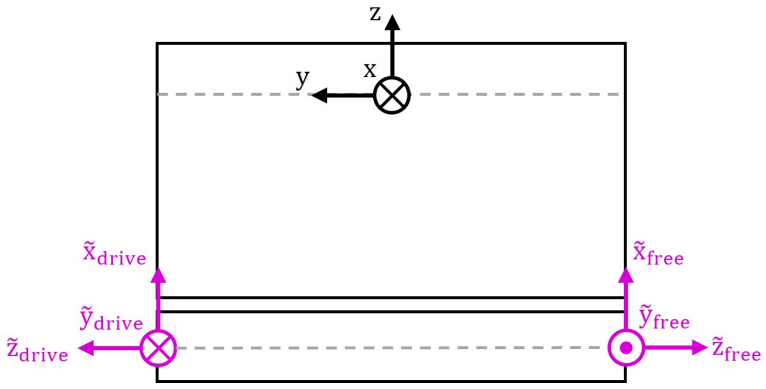

Figure C.1: Local load cell (pink) and global (black) coordinate systems. Circles with crosses denote a direction into the page, while circles with dots denote a direction out of page. The direction of wave propagation is into the page, in the +x global direction.

4. Apply a low-pass filter to reduce noise in the velocity and acceleration estimates. We use an infinite impulse response (IIR) low-pass filter with a passband frequency of 5 Hz and a stopband frequency of 15 Hz, which are substantially higher frequency than the fundamental flap motion.

5. Linearly interpolate the filtered velocity and acceleration signals to align data points at original time steps.

C.0.2 Load cell data

The raw load cell data ranges from ±10 V. We took periodic tares where we collect load cell data in still water with a vertical flap to provide a "zero" value for the load cells and account for drift over time. To process torque data, we perform the following steps:

1. Identify any tests that exceed the load cell pitch torque calibration limit of 60 Nm. These tests are marked and removed from any analysis that requires pitch torque measurements (e.g., evaluating the capture width ratio).

2. Calculate and subtract an average of the tares taken directly before and after the experiment from the data (in volts).

3. Convert the readings from volts to N/Nm using calibration matrices provided by the manufacturer.

4. Apply a rotation matrix to transform the measurements from the local load cell coordinate system to the global coordinate system described in Figure 3.3. The transformed forces and torques are still located at the load cell positions, just in the global coordinate system orientation. The following equations describe the transformation of forces and torques in the local reference from (denoted by $\tilde{}$) to the global reference frame:

$$\begin{bmatrix} F_{\mathrm{x}} \\ F_{\mathrm{y}} \\ F_{\mathrm{z}} \\ \tau_{\mathrm{x}} \\ \tau_{\mathrm{y}} \\ \tau_{\mathrm{z}} \end{bmatrix}_{\mathrm{drive}} = \begin{bmatrix} \sin(\theta) & \cos(\theta) & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \cos(\theta) & -\sin(\theta) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \cos(\theta) & -\sin(\theta) & 0 \end{bmatrix} \begin{bmatrix} \tilde{F}_{\mathrm{x}} \\ \tilde{F}_{\mathrm{y}} \\ \tilde{F}_{\mathrm{z}} \\ \tilde{\tau}_{\mathrm{x}} \\ \tilde{\tau}_{\mathrm{y}} \\ \tilde{\tau}_{\mathrm{z}} \end{bmatrix}_{\mathrm{drive}} \tag{C.3}$$

$$\begin{bmatrix} F_{x} \\ F_{y} \\ F_{z} \\ \tau_{x} \\ \tau_{y} \\ \tau_{z} \end{bmatrix}_{\mathrm{free}} = \begin{bmatrix} \sin(\theta) & -\cos(\theta) & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ \cos(\theta) & \sin(\theta) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sin(\theta) & -\cos(\theta) & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & \cos(\theta) & \sin(\theta) & 0 \end{bmatrix} \begin{bmatrix} \tilde{F}_{x} \\ \tilde{F}_{y} \\ \tilde{F}_{z} \\ \tilde{\tau}_{x} \\ \tilde{\tau}_{y} \\ \tilde{\tau}_{z} \end{bmatrix}_{\mathrm{free}} \tag{C.4}$$

5. Apply the same low-pass filter as for encoder data to reduce noise in the measured torque signal.

Appendix D

NONLINEAR WEC MODELING USING SPARSE IDENTIFICATION OF NONLINEAR DYNAMICS (SINDY)

D.1 Motivation

In Chapter 5, we use SINDy to describe experimental data with different degrees of nonlinear behavior. However, SINDy can also be used to model high-fidelity CFD data. Here, we present work published in the proceedings of 15th European Wave and Tidal Energy Conference in 2023 [107], where we use SINDy to model OSWEC kinematics from CFD data generated using OpenFOAM. Although the data presented here is too limited to publish beyond a conference proceeding, we find SINDy can accurately model nonlinear acceleration using position and velocity data. Additionally, unlike the work in Chapter 5, here, we model an ordinary differential equation describing OSWEC kinematics, allowing for future state prediction using numerical integration. This work further demonstrates that SINDy is a promising tool for nonlinear WEC modeling and is an important step to integrate SINDy models into model predictive control schemes. Sections D.2, D.3, and D.4 provide the methods, results, and conclusions of this work, respectively.

D.2 Methods

The methods of this work closely follow those described in Section 5.4 and Figure 5.3, with two main exceptions: the source of the training/testing data and the state/modeled variables used in the SINDy model.

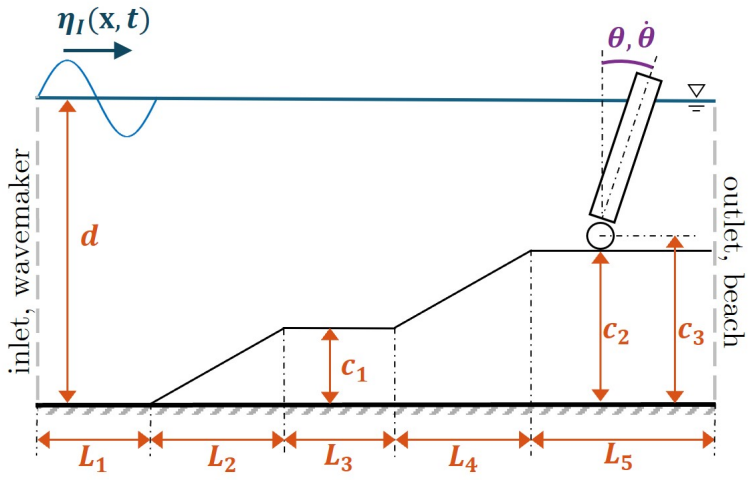

Figure D.1: Schematic of the OpenFOAM simulation setup [119], based on experiments in [15]. The incident wave input is shown in blue, with measured system states in green and dimensions in purple. The values of the dimensions are give in Table D.1.

D.2.1 Data generation

Instead of using experimental data as input to the SINDy model, we generate input data using an OpenFOAM v2012 model of a laboratory-scale OSWEC described in [119]. A schematic of the simulation is given in Figure D.1 and is based on experiments run at Queen's University in Belfast [15]. The wave tank is 4.58 meters wide and 20 meters long with a progressive sloped bathymetry with two flat sections connected by linear ramps. There is a wavemaker that generates regular, linear waves with a period of 2.0625 seconds and a height of 0.05 meters. Although Figure D.1 shows the OSWEC composed of a base and flap, the OpenFOAM simulation only models the flap with a constraint that it oscillates at a hinge oriented with the cross-dimension of the tank and located 0.476 m from the bottom of the tank. The OSWEC flap is 0.65 meters wide and 0.34 meters tall. Relevant system parameters are summarized in Table D.1 and we refer the reader to [119] for further simulation details on wave tank, OSWEC, and wave parameters.

Table D.1: Relevant values for OpenFOAM simulation setup.

Symbol	Quantity	Value
a	Flap thickness	0.1 m
h	Flap height	0.341 m
w	Flap width	0.65 m
d	Still water depth	0.71 m
H	Wave height	0.05 m
T	Wave period	2.065 s
I	Flap moment of inertia	0.1750 kg m^2
m_f	Flap mass	10.77 kg
L_1	Beach dimensions	4.82 m
L_2		1.30 m
L_3		2.40 m
L_4		3.70 m
L_5		6.20 m
c_1		0.150 m
c_2		0.356 m
c_3	Hinge height	0.476 m

OpenFOAM is a finite volume fluid solver that can be combined with a rigid body solver to model complex fluid-structure interaction. Because we aim to model nonlinear OSWEC behavior, we want the OSWEC to experience large rotations. However, this can complicate the meshing process in the numerical simulation, as large rotations can cause the mesh to degrade and become skewed [15, 120]. To overcome this, this OpenFOAM model uses an overset grid, where the simulation domain contains two meshes: a background mesh of the wave tank and a overset body-fitted mesh around the flap [121]. Using this overset grid

method, neither mesh is deformed throughout the simulation. Instead, the overset mesh moves with the rigid flap and flow field values are interpolated from the boundary of the background and overset mesh. There have been several recent studies have utilized this meshing method to model numerical wave tanks [122] and WEC hydrodynamics [123–128]. Further details on meshing and simulation details are provided in Riddle [119].

D.2.2 State and modeled variables

In this work, we aim to describe angular acceleration as a function of flap position and velocity. In other words, we aim to find a function f such that:

$$\ddot{\boldsymbol{\theta}} = f(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}). \tag{D.1}$$

We extract flap position, $\boldsymbol{\theta}$, and angular velocity, $\dot{\boldsymbol{\theta}}$, from the OpenFOAM simulation and calculate $\ddot{\boldsymbol{\theta}}$ by applying a fourth-order numerical differentiation scheme to $\dot{\boldsymbol{\theta}}$. We trim the time series into a six second window (almost three oscillation periods) to use as training data for SINDy. Our data matrices, $\boldsymbol{Y}$ and $\boldsymbol{Z}$ become:

$$\boldsymbol{Y} = \begin{bmatrix} | & | \\ \boldsymbol{\theta} & \dot{\boldsymbol{\theta}} \\ | & | \end{bmatrix} \in \mathbb{R}^{m\times 2}, \quad \boldsymbol{Z} = \begin{bmatrix} | \\ \ddot{\boldsymbol{\theta}} \\ | \end{bmatrix} \in \mathbb{R}^{m\times 1}. \tag{D.2}$$

This setup differs from Chapter 5 because we are finding a second-order ordinary differential equation that we can numerically integrate to get position and velocity estimates without instantaneous measurement data. Because $P_a = \tau_{\text{PTO}}\dot{\theta}$, a model that accurately predicts velocity is critical for MPC implementation to optimize performance.

Figure D.2 shows the state $(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ and modeled $(\ddot{\boldsymbol{\theta}})$ variables used to train the SINDy model. These kinematics demonstrate complex, nonlinear behavior that can not be captured by linear potential flow models. Linear models often assume that the flap oscillates as a perfect sinusoid with an oscillation frequency equal to that of the incident wave, such that:

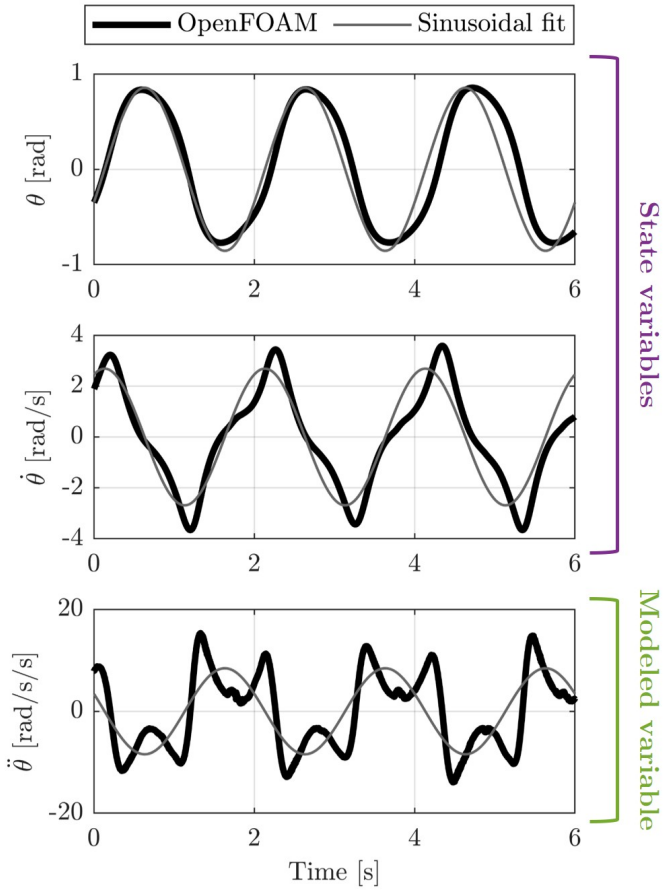

Figure D.2: Output from OpenFOAM (black) of [top] angular position, $\boldsymbol{\theta}$, [middle] velocity, $\dot{\boldsymbol{\theta}}$, and [bottom] acceleration, $\ddot{\boldsymbol{\theta}}$. Gray curves represent a sinusoid with the same initial conditions as the OpenFOAM data and an oscillation period equal to the wave period, as is the form of kinematic behavior from linear modeling techniques. The OpenFOAM output exhibits nonlinear behavior including time-varying oscillation frequency and evident cubic nonlinearity in $\dot{\boldsymbol{\theta}}$.

$\theta_{\text{linear}}(t) = A\sin(\omega t + \upsilon)$, where $\omega = 2\pi/T$, and T is the incident wave period [5, 129, 130]. We compare the OpenFOAM output against this linear representation, with parameters A and υ determined from the state initial condition and maximum amplitude of the OpenFOAM model, respectively. Similarly, we calculate the derivatives of the linear representation to

compare to $\dot{\boldsymbol{\theta}}$ and $\ddot{\boldsymbol{\theta}}$. Comparing these two curves highlights the nonlinearities present in the state variables. First, the OpenFOAM kinematics oscillate at a slightly different frequency than the incident wave. In addition, the shape of the oscillations for the kinematics vary from a simple sinusoid. The shape of $\boldsymbol{\theta}$ is mostly sinusoidal, with some asymmetry about the peak, but in taking the derivatives the nonlinearity is more evident, and it is clear that higher order functions are required to describe the behavior. For example, there is evident cubic behavior in the oscillations for angular velocity. And finally, the peaks in amplitude of the OpenFOAM angular velocity and acceleration exceed that of the linear approximation. It is important to model these peaks well to ensure structural integrity and power conversion efficiency for the device.

With these data matrices, we use SINDy to find a parsimonious model for angular acceleration, $\ddot{\boldsymbol{\theta}}$ (Equation 5.4). Similar to Chapter 5, we only include polynomial functions in the nonlinear function library, $\boldsymbol{\Lambda}(\boldsymbol{Y})$, and solve Equation 5.5 for function weights, $\boldsymbol{\xi}$, using sequential threshold least-squares algorithm [89, 109].

D.2.3 Choosing SINDy parameters

As in Chapter 5, to find a parsimonious model, we run the SINDy algorithm over a range of the two user-defined parameters: maximum polynomial order, N, and the sparsity-promoting hyperparameter, λ. To assess how well the SINDy model describes the nonlinear dynamics, we define error as:

$$\text{error} = ||\ddot{\boldsymbol{\theta}} - \ddot{\boldsymbol{\theta}}_{\text{SINDy}}||_2. \tag{D.3}$$

We look at the error value over the tested range of model parameters to determine what values result in a parsimonious model (i.e., the best-fit for the nonlinear dynamics using the fewest nonzero terms in $\boldsymbol{\xi}$).

Figure D.3 summarizes the results of iterating the SINDy algorithm over a range of maximum polynomial ordesr, N, and sparsity-promoting hyperparameters, λ. To choose the maximum polynomial order, N, we look at error as a function of N for $\lambda = 0$ case (no

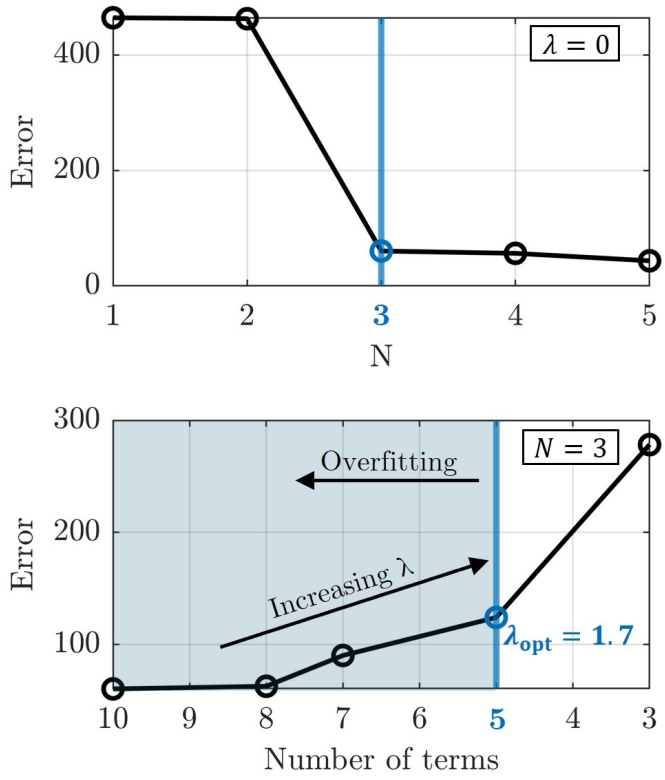

Figure D.3: [Top] Error as a function of maximum polynomial order, N, for $\lambda = 0$, i.e., including all terms, no sparsity enforced. Error drops significantly when considering cubic nonlinearity, with little accuracy gained when considering higher orders. This behavior also occurs for all other λ values. We choose $N = 3$ based on this drastic decrease in error. [Bottom] Error as a function of number of activated terms in $\boldsymbol{\xi}$ for $N = 3$. Number of terms decrease as λ increases and we further enforce sparsity. Error increases drastically when the number of terms drops below five, therefore we choose $\lambda_{\text{opt}} = 1.7$ to obtain a parsimonious model that is accurate but not overfit.

sparsity enforced), shown in Figure D.3. Although we are only showing the case for $\lambda = 0$, there is a similar pattern for all λ values tested. There is a drastic drop in error for $N > 2$,

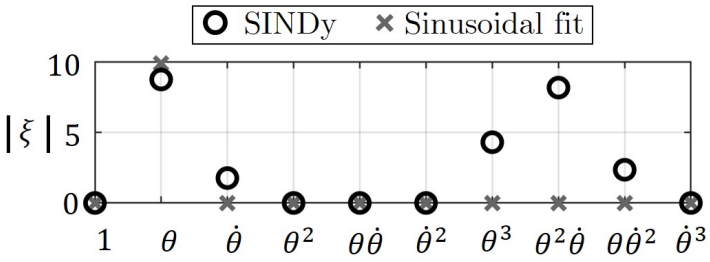

Figure D.4: Magnitude of function weights, $\boldsymbol{\xi}$, for SINDy model of angular acceleration, $\ddot{\boldsymbol{\theta}}$ with optimal conditions $N = 3$ and $\lambda_{\text{opt}} = 1.7$ (black circles) and function weights of a purely sinusoidal, linear dynamics, $\ddot{\boldsymbol{\theta}} = -\omega^2\boldsymbol{\theta}$ (gray markers). Although both models have a large magnitude coefficient for $\boldsymbol{\theta}$, $\boldsymbol{\xi}$ has other nonzero coefficients for nonlinear terms, particularly cubic functions.

which implies that the cubic terms are vital to describing system dynamics. Since error does not significantly decrease for $N > 3$, we choose $N = 3$ as indicative of the lowest polynomial order that gives an accurate representation of the dynamics. After choosing N, we choose hyperparameter λ by considering error as a function of number of terms in the reduced order model for $\ddot{\boldsymbol{\theta}}$ (Figure D.3). As previously mentioned, as λ increases, the number of terms decreases as sparsity is more heavily weighted in Equation 5.5. Although error increases as number of terms decreases (as expected), until the number of terms decreases below five terms the error is relatively stable. This means that five terms are the minimum number of terms necessary to describe the dynamics, so we choose $\lambda_{\text{opt}} = 1.7$ as our hyperparameter. Note that even though including more terms decreases the error in the SINDy model, including all of those terms means we are likely overfitting the data and failing to identify the most parsimonious model.

D.3 Results and discussion

After choosing the maximum polynomial order, N, and sparsity-promoting hyperparameter, λ_{opt}, we have an expression for angular acceleration with five remaining terms:

$$\ddot{\boldsymbol{\theta}} = -8.7\boldsymbol{\theta} + 1.8\dot{\boldsymbol{\theta}} + 4.3\boldsymbol{\theta}^3 - 8.2\boldsymbol{\theta}^2\dot{\boldsymbol{\theta}} - 2.4\boldsymbol{\theta}\dot{\boldsymbol{\theta}}^2. \tag{D.4}$$

To visualize the relative importance of each term in Equation D.4, Figure D.4 shows the magnitude of function weights $\boldsymbol{\xi}$ for each candidate function in $\boldsymbol{\Lambda}(\boldsymbol{Y})$. It is important to note that if this was a purely linear system and $\boldsymbol{\theta}$ could be represented by a sinusoid (such that $\ddot{\boldsymbol{\theta}} = -\omega^2\boldsymbol{\theta}$), the only nonzero component in $\boldsymbol{\xi}$ would be for the $\boldsymbol{\theta}$ function. Although the magnitude of the $\boldsymbol{\theta}$ function coefficient is the largest of $\boldsymbol{\xi}$, and of similar value of what it would be for the sinusoidal oscillation case, we see that there are several other active terms required for accurate modeling that deviate from the standard sinusoidal case. For example, both $\dot{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$ are needed to accurately describe $\ddot{\boldsymbol{\theta}}$, not just $\boldsymbol{\theta}$. Also, there are several higher order terms contributing to the dynamics of $\ddot{\boldsymbol{\theta}}$. In particular, the cubic terms contribute significantly to the model, including cross terms $\boldsymbol{\theta}^2\dot{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}\dot{\boldsymbol{\theta}}^2$. In addition, all the quadratic terms are zeroed out, further emphasizing that there are dominant cubic nonlinearities represented in the dynamics that are necessary for accurate reconstruction. Note that even though only the odd orders of the library are activated, the active terms do not follow a traditional sine expansion of the state variables, which are already oscillatory in nature, due to the cross terms in Equation D.4. This type of analysis is beneficial for multiple reasons. First, it informs the tools required for accurate modeling of OSWEC behavior in other systems. For example, if we were attempting to model a different OSWEC in response to a different incident wave, we would now know that for large rotations we will likely need to consider higher order terms up to the third order in our analysis. Without that knowledge, we would likely miss important dynamics that couldn't be captured with traditional linear methods. Second, the characteristics and order of the nonlinearity identified by SINDy can give important information on the nature of the system and how to control it. For example, it has been shown other oscillators with higher-order nonlinearities have a shift in eigenvalues

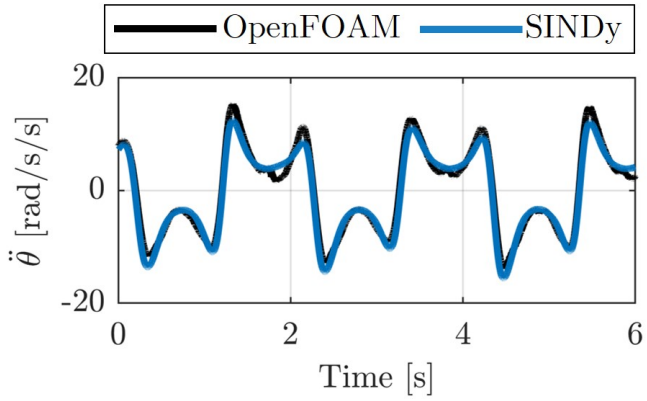

Figure D.5: OpenFOAM (black) and SINDy (blue) model of $\ddot{\boldsymbol{\theta}}$ in the training region.

of the system, which in term shift the oscillation frequency and amplitude, (both of which are evident in Figure D.2) [131]. This has important implications in resonant behavior as well as system control.

Figure D.5 compares the SINDy model for $\ddot{\boldsymbol{\theta}}$ to the output from OpenFOAM in the training region. The SINDy model describes the dynamics almost perfectly, which is what we expect since the model was trained on this data. There is slight underestimation on the positive peaks, but the nonlinear oscillation shape and frequency are well captured, unlike the case of sinusoidal motion (Figure D.2). This suggests that the model describes the significant nonlinearities well in the training region. This is a promising result, but to confirm that Equation D.4 describes the true dynamics of the full system, we need to look beyond the training region to see if the model can still accurately describe data it was not trained on.

Figure D.6 compares the OpenFOAM and SINDy outputs for $\ddot{\boldsymbol{\theta}}$ over the full time series of OpenFOAM output. The time series includes three regions: the training region, a transient region before the training region where the wavemaker is ramping up, and a steady-state region after the training region. To be confident our model represents the true dynamics of the system, it is critical that it can accurately describe the system behavior in all three

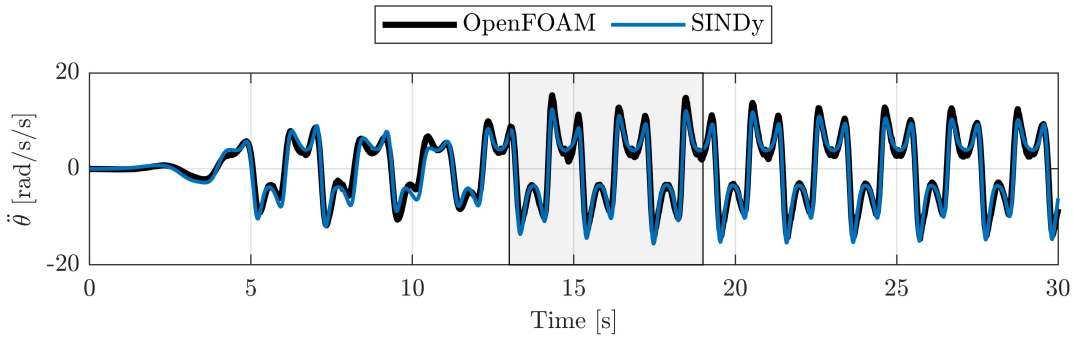

Figure D.6: OpenFOAM output (black) and SINDy output (blue) of angular acceleration, $\ddot{\boldsymbol{\theta}}$, in the training region (gray) and testing region (white). The SINDy model captures the dominating nonlinear features of the dynamics throughout both the training and testing region. In particular, the SINDy model is accurate even in the transient region before the training region, ensuring the model is describing the true system dynamics.

regions. Although we only trained the SINDy model on a small section of steady-state data, it is able to capture the dynamics well in all three regions. It is particularly promising that SINDy can describe the transient region because even though there are other reduced order modeling techniques that may be able to capture the higher order periodic behavior observed in the training region (such as a Fourier analysis or dynamic mode decomposition with time delays), this transient behavior would not be accurately represented unless it was included in the data. Although we may not be interested in this exact type of transient behavior, this shows that SINDy is able to capture transient kinematics, even when being trained on only steady-state data.

Now that we are confident that Equation D.4 accurately describes the angular acceleration of the system, $\ddot{\boldsymbol{\theta}}$, we can integrate Equation D.4 to evaluate the time evolution of state variables $\dot{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$. Figure D.7 shows the results of this integration using the same initial conditions as the OpenFOAM data. There is an evident transient and steady-state region and the shape of the oscillations is similar between the OpenFOAM output and SINDy model,

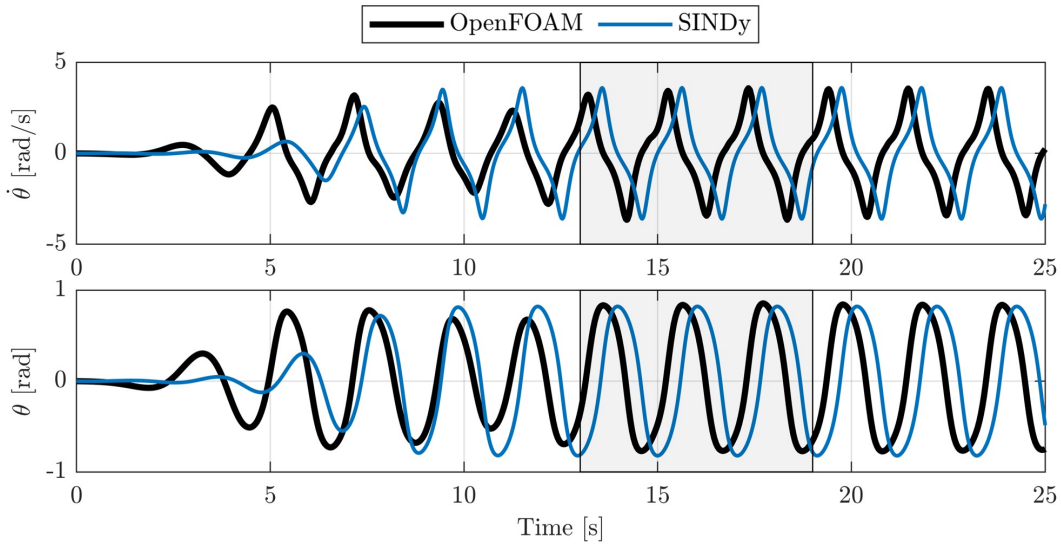

Figure D.7: OpenFOAM output (black) and integrated SINDy output (blue) of [top] angular velocity, $\dot{\boldsymbol{\theta}}$, and [bottom] angular position, $\boldsymbol{\theta}$. The training region is shown in gray. The SINDy model captures the dominating nonlinear features of the dynamics, but there is an evident phase shift between the two time series.

but there is a clear phase difference between the two. This can be a common result from integrating SINDy models, and we hypothesize that this is a consequence of a mismatch of excitations of the system. Since Equation D.4 has no external excitation term, the nonzero initial conditions act as excitation to the system, while the OpenFOAM model is driven externally from the incident waves. Therefore, if the initial conditions of the OpenFOAM model do not excite the SINDy model in the same way as the external wave forcing, there will be a mismatch in phase as the transient region varies between the two models. We plan to investigate this further in future work. However, besides the evident phase shift, the two models both have a transient region, and steady state region, and oscillate with similar amplitude, structure, and frequency.

To more clearly visualize the differences in the integrated state variables from the SINDy model, Figure D.8 includes a 2.5 second phase shift that aligns the SINDy integration with

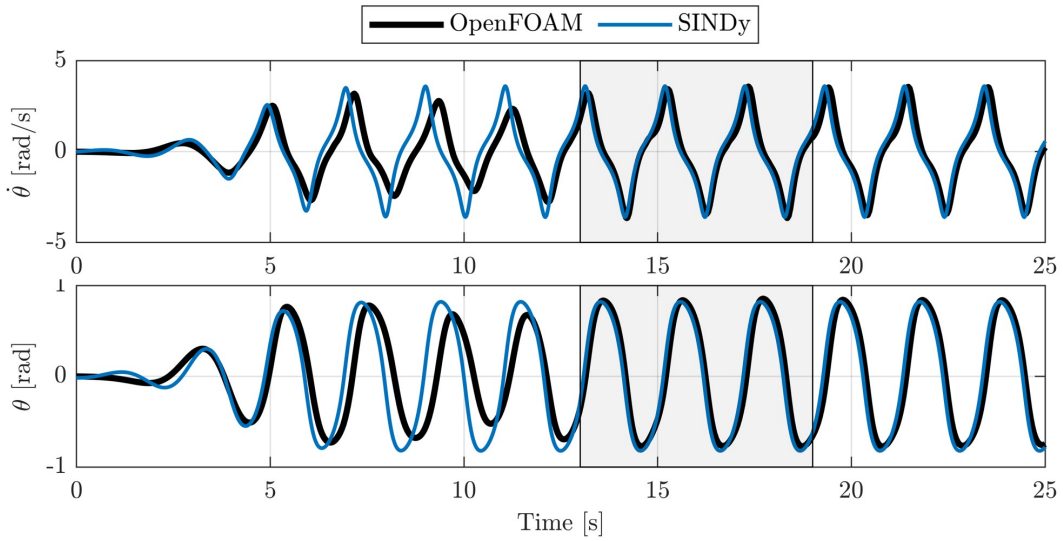

Figure D.8: OpenFOAM output (black) and integrated SINDy output (blue) of [top] angular velocity, $\dot{\boldsymbol{\theta}}$, and [bottom] angular position, $\boldsymbol{\theta}$. The SINDy time series is time-shifted to better match the phase of the OpenFOAM data. The training region is shown in gray. The two models are almost identical in the steady-state region, but there is some discrepancy in the transient region before the training data.

the steady-state period. From this representation, it is clear that the SINDy model captures the steady-state region for both variables well in both the training and testing region. For the transient region, the SINDy model starts modeling the behavior accurately, but then there is period in which the SINDy model does not match the OpenFOAM data. This could be due to a numerical artifact from the OpenFOAM model, or some underlying dynamics that are lost when integrating the SINDy model, even though angular acceleration is well-described throughout (Figure D.6).

Finally, Figure D.9 shows angular velocity $\dot{\boldsymbol{\theta}}$ as a function of angular position, $\boldsymbol{\theta}$, i.e., a phase-space diagram of the state variables. Representing the variables this way allows us to compare the two models without comparing the two time series directly. Both models have a transient region that approaches a stable "limit cycle". The shape of the limit cycles

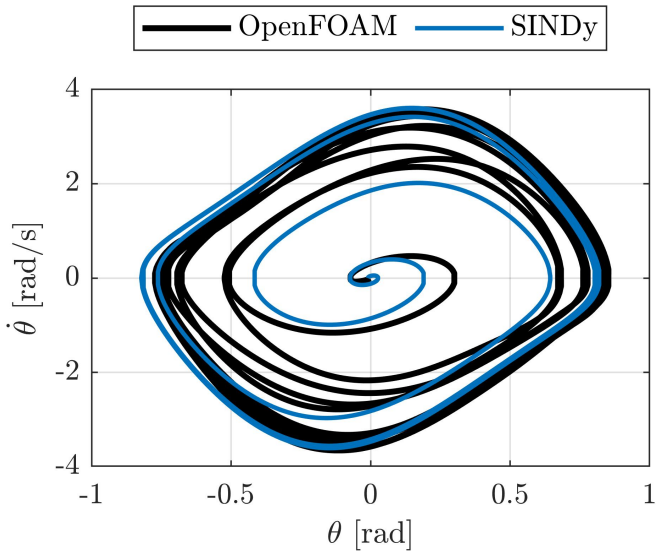

Figure D.9: Phase-space diagram of the two state variables: angular velocity, $\dot{\boldsymbol{\theta}}$, and angular position, $\boldsymbol{\theta}$, for both OpenFOAM (black) and SINDy model (blue). Both models have a transient region that leads to a "limit cycle". The SINDy model captures the initial transient behavior and the final limit cycle well, but there is discrepancy in the region between the initial transient and the limit cycle that is also evident in the time domain representations in Figure D.8.

for the OpenFOAM and SINDy models match well. The SINDy model also models the initial transient behavior well, but, as was evident in Figure D.8, there is discrepancy in the two models in the late transient region before the limit cycle. Again, this may be due to a numerical artifact or hidden dynamics not well described when integrated the Equation D.4. Interestingly, we observe that the behavior of this system is similar to that of a Hopf bifurcation [132], with an unstable fixed point at the center of a stable limit cycle. Because this behavior has been well studied in other dynamical systems, existing work in modeling and controlling similar systems may be able to inform further research in modeling and controlling similar WEC systems.

D.4 Conclusions

In this study, we use a nonlinear data-driven algorithm, SINDy, to generate accurate models of nonlinear OSWEC kinematics using a limited number of terms. We used kinematic data from a high-fidelity CFD simulation of an OSWEC in response to large amplitude regular waves, and showed the data to be highly nonlinear, with time-varying oscillation frequency and nonlinear oscillation profile. Using SINDy, we developed an accurate model for angular acceleration as a sum of nonlinear functions built from kinematic data (i.e., angular position and velocity). This model was not only able to describe angular acceleration in the training data, but also accurately captured early transient behavior before the training region and steady-state behavior after. When we numerically integrate the model to obtain time series of angular velocity and position, there was a notable phase shift between the data and the integrated SINDy model, as well as amplitude and frequency deviations in part of the transient region. Agreement between input kinematics and integrated quantities is, however, excellent in the training and steady-state region.

By adjusting the maximum polynomial order in the model as well as the sparsity-promoting hyperparameter, we were able to ensure we had a parsimonious model that we could interpret to gain insight into the underlying nonlinear behavior. We found that to accurately describe the nonlinearity in the angular acceleration we must include cubic terms, and quadratic terms did not contribute to the dynamics, suggesting that future models need to consider terms up to the third order to be able to describe the dynamics well. Using this model, we can efficiently and accurately model and predict nonlinear OSWEC kinematics using only existing data. Overall, this work supports the conclusion of Chapter 5 that SINDy is a promising tool to generate reduced-order models of nonlinear WEC behavior and can provide insights into the underlying dynamics of these complex systems.

BIBLIOGRAPHY

[1] A. F. d. O. Falcão, "Wave energy utilization: A review of the technologies," *Renewable and sustainable energy reviews*, vol. 14, no. 3, pp. 899–918, 2010.

[2] D. Clemente, P. Rosa-Santos, and F. Taveira-Pinto, "On the potential synergies and applications of wave energy converters: A review," *Renewable and Sustainable Energy Reviews*, vol. 135, p. 110 162, 2021.

[3] M. Folley, T. Whittaker, M. Osterried, *et al.*, "The oscillating wave surge converter," in *The Fourteenth International Offshore and Polar Engineering Conference*, International Society of Offshore and Polar Engineers, 2004.

[4] T. Whittaker, D. Collier, M. Folley, M. Osterried, A. Henry, and M. Crowley, "The development of oyster—a shallow water surging wave energy converter," in *Proceedings of the 7th European wave and tidal energy conference*, 2007, pp. 11–14.

[5] J. Falnes, "Ocean waves and oscillating systems," *Ocean Waves and Oscillating Systems*, p. 286, 2002.

[6] M. Folley and T. Whittaker, "The cost of water from an autonomous wave-powered desalination plant," *Renewable Energy*, vol. 34, no. 1, pp. 75–81, 2009.

[7] G. Giorgi, M. Penalba, and J. Ringwood, "Nonlinear hydrodynamic force relevance for heaving point absorbers and oscillating surge converters," in *Proceedings of the AWTEC Asian Wave and Tidal Energy Conference, Singapore*, 2016.

[8] A. Babarit, J. Hals, M. J. Muliawan, A. Kurniawan, T. Moan, and J. Krokstad, "Numerical benchmarking study of a selection of wave energy converters," *Renewable energy*, vol. 41, pp. 44–63, 2012.

[9] A. Henry, O. Kimmoun, J. Nicholson, G. Dupont, Y. Wei, F. Dias, *et al.*, "A two dimensional experimental investigation of slamming of an oscillating wave surge converter," in *The Twenty-fourth International Ocean and Polar Engineering Conference*, International Society of Offshore and Polar Engineers, 2014.

[10] M. Penalba and J. V. Ringwood, "A review of wave-to-wire models for wave energy converters," *Energies*, vol. 9, no. 7, p. 506, 2016.

[11] J. Davidson and R. Costello, "Efficient nonlinear hydrodynamic models for wave energy converter design—a scoping study," *Journal of Marine Science and Engineering*, vol. 8, no. 1, p. 35, 2020.

[12] M. Folley, A. Henry, and T. Whittaker, "Contrasting the hydrodynamics of heaving and surging wave energy converters," in *Proceedings of the 11th European Wave and Tidal Enerrgy Conference, Nantes, France*, 2015, pp. 6–11.

[13] Y. Wei and F. Dias, "Numerical study of three dimensional effects of wave impact on an oscillating wave surge converter," in *ASME 2015 34th International Conference on Ocean, Offshore and Arctic Engineering*, American Society of Mechanical Engineers Digital Collection, 2015.

[14] Y. Wei, T. Abadie, A. Henry, and F. Dias, "Wave interaction with an oscillating wave surge converter. part ii: Slamming," *Ocean Engineering*, vol. 113, pp. 319–334, 2016.

[15] P. Schmitt and B. Elsaesser, "On the use of openfoam to model oscillating wave surge converters," *Ocean Engineering*, vol. 108, pp. 98–104, 2015.

[16] Y. Wei, A. Rafiee, B. Elsaesser, and F. Dias, "Numerical simulation of an oscillating wave surge converter," in *ASME 2013 32nd International Conference on Ocean, Offshore and Arctic Engineering*, American Society of Mechanical Engineers Digital Collection, 2013.

[17] D.-W. Chen *et al.*, "Numerical modeling of wave-induced rotations of a bottom-hinged flapper with a sph model," *Journal of Marine Science and Technology*, vol. 22, no. 3, pp. 372–380, 2014.

[18] S. Yeylaghi, B. Moa, P. Oshkai, B. Buckham, and C. Crawford, "Isph modelling of an oscillating wave surge converter using an openmp-based parallel approach," *Journal of Ocean Engineering and Marine Energy*, vol. 2, no. 3, pp. 301–312, 2016.

[19] D. Zhang, Y. Shi, C. Huang, Y. Si, B. Huang, and W. Li, "Sph method with applications of oscillating wave surge converter," *Ocean Engineering*, vol. 152, pp. 273–285, 2018.

[20] Z. Wei, B. L. Edge, R. A. Dalrymple, and A. Hérault, "Modeling of wave energy converters by gpusph and project chrono," *Ocean Engineering*, vol. 183, pp. 332–349, 2019.

[21] M. Brito *et al.*, "A numerical tool for modelling oscillating wave surge converter with nonlinear mechanical constraints," *Renewable Energy*, vol. 146, pp. 2024–2043, 2020.

[22] B. R. Noack, M. Morzynski, and G. Tadmor, *Reduced-order modelling for flow control*. Springer Science & Business Media, 2011, vol. 528.

[23] P. Benner, S. Gugercin, and K. Willcox, "A survey of projection-based model reduction methods for parametric dynamical systems," *SIAM review*, vol. 57, no. 4, pp. 483–531, 2015.

[24] K. Duraisamy, G. Iaccarino, and H. Xiao, "Turbulence modeling in the age of data," *Annual review of fluid mechanics*, vol. 51, no. 1, pp. 357–377, 2019.

[25] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," *Annual review of fluid mechanics*, vol. 52, no. 1, pp. 477–508, 2020.

[26] S. Giorgi, J. Davidson, M. Jakobsen, M. Kramer, and J. V. Ringwood, "Identification of dynamic models for a wave energy converter from experimental data," *Ocean Engineering*, vol. 183, pp. 426–436, 2019.

[27] M. Rosati, T. Kelly, D. G. Violini, and J. Ringwood, "Data-based hydrodynamic modelling of a fixed owc wave energy converter," in *Proceedings of the 14th European Wave and Tidal Energy Conference, Plymouth, UK*, 2021.

[28] M. Rosati, T. Kelly, and J. V. Ringwood, "Nonlinear data-based hydrodynamic modeling of a fixed oscillating water column wave energy device," *IEEE Access*, vol. 9, pp. 149 756–149 765, 2021.

[29] M. Rosati, J. Ringwood, H. Bingham, B. Joensen, and K. Nielsen, "A data-based modelling approach for a vented oscillating water column wave energy converter," *Trends in Renewable Energies Offshore*, pp. 339–347, 2022.

[30] X. Wang, D. Liang, M. Li, P. Stansby, and L. Zhang, "Data-driven system identification modelling for multi-float m4 wave energy converter with elastic bed-buoy-bow float mooring," in *Proceedings of the 15th European Wave and Tidal Energy Conference*, 2023.

[31] S. Giorgi, J. Davidson, and J. Ringwood, "Identification of nonlinear excitation force kernels using numerical wave tank experiments," in *Proceedings of the 11th European Wave and Tidal Energy Conference*, European Wave and Tidal Energy Conference 2015, 2015.

[32] J. Davidson, S. Giorgi, and J. V. Ringwood, "Numerical wave tank identification of nonlinear discrete time hydrodynamic models," *Renewable Energies Offshore*, p. 279, 2015.

[33] S. Giorgi, J. Davidson, and J. V. Ringwood, "Identification of wave energy device models from numerical wave tank data—part 2: Data-based model determination," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 3, pp. 1020–1027, 2016.

[34] D. Valério, M. J. Mendes, P. Beirão, and J. S. da Costa, "Identification and control of the aws using neural network models," *Applied Ocean Research*, vol. 30, no. 3, pp. 178–188, 2008.

[35] C. Ni, X. Ma, and J. Wang, "Integrated deep learning model for predicting electrical power generation from wave energy converter," in *2019 25th International Conference on Automation and Computing (ICAC)*, IEEE, 2019, pp. 1–6.

[36] E. Katsidoniotaki, S. Guth, M. Göteman, and T. P. Sapsis, "Reduced order modeling of wave energy systems via sequential bayesian experimental design and machine learning," 2023.

[37] S. A. Hughes, *Physical models and laboratory techniques in coastal engineering*. World Scientific, 1993, vol. 7.

[38] A. Henry, "The hydrodynamics of small seabed mounted bottom hinged wave energy conerverters in shallow water," Ph.D. dissertation, Queen's University Belfast, 2009.

[39] P. Schmitt *et al.*, "Hydrodynamic loading on a bottom hinged oscillating wave surge converter," in *The Twenty-second International Offshore and Polar Engineering Conference*, International Society of Offshore and Polar Engineers, 2012.

[40] D. Ning, C. Liu, C. Zhang, M. Goteman, H. Zhao, and B. Teng, "Hydrodynamic performance of an oscillating wave surge converter in regular and irregular waves: An experimental study," *Journal of Marine Science and Technology*, vol. 25, no. 5, pp. 520–530, 2017.

[41] M. Brito, R. M. Ferreira, L. Teixeira, M. G. Neves, and L. Gil, "Experimental investigation of the flow field in the vicinity of an oscillating wave surge converter," *Journal of Marine Science and Engineering*, vol. 8, no. 12, p. 976, 2020.

[42] M. Brito, R. M. Ferreira, L. Teixeira, M. G. Neves, and L. Gil, "Experimental investigation of the flow field in the vicinity of an oscillating wave surge converter," *Journal of Marine Science and Engineering*, vol. 8, no. 12, p. 976, 2020.

[43] A. Henry, P. Schmitt, T. Whittaker, A. Rafiee, F. Dias, *et al.*, "The characteristics of wave impacts on an oscillating wave surge converter," in *The Twenty-third Interna-*

tional Offshore and Polar Engineering Conference, International Society of Offshore and Polar Engineers, 2013.

[44] A. Henry, T. Abadie, J. Nicholson, A. McKinley, O. Kimmoun, and F. Dias, "The vertical distribution and evolution of slam pressure on an oscillating wave surge converter," in *International Conference on Offshore Mechanics and Arctic Engineering*, American Society of Mechanical Engineers, vol. 56475, 2015, V001T01A034.

[45] P. Lamont-Kane, A. McKinley, A. Henry, J. Nicholson, M. Folley, and B. Elsaesser, "Investigating extreme loads on an oscillating wave surge converter," in *Proceedings of the 11th European Wave and Tidal Energy Conference*, 2015.

[46] N. Tom, M. Lawson, Y.-H. Yu, and A. Wright, "Development of a nearshore oscillating surge wave energy converter with variable geometry," *Renewable Energy*, vol. 96, pp. 410–424, 2016.

[47] N. M. Tom, Y.-H. Yu, A. D. Wright, and M. J. Lawson, "Pseudo-spectral control of a novel oscillating surge wave energy converter in regular waves for power optimization including load reduction," *Ocean Engineering*, vol. 137, pp. 352–366, 2017.

[48] M. A. Choiniere, N. M. Tom, and K. P. Thiagarajan, "Load shedding characteristics of an oscillating surge wave energy converter with variable geometry," *Ocean Engineering*, vol. 186, p. 105 982, 2019.

[49] M. Choiniere, J. Davis, N. Nguyen, N. Tom, M. Fowler, and K. Thiagarajan, "Hydrodynamics and load shedding behavior of a variable-geometry oscillating surge wave energy converter (oswec)," *Renewable Energy*, vol. 194, pp. 875–884, 2022.

[50] D. Sarkar, K. Doherty, and F. Dias, "The modular concept of the oscillating wave surge converter," *Renewable Energy*, vol. 85, pp. 484–497, 2016.

[51] L. Wilkinson, T. Whittaker, P. Thies, S. Day, and D. Ingram, "The power-capture of a nearshore, modular, flap-type wave energy converter in regular waves," *Ocean Engineering*, vol. 137, pp. 394–403, 2017.

[52] G. Bacelli and J. V. Ringwood, "Nonlinear optimal wave energy converter control with application to a flap-type device," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 7696–7701, 2014.

[53] G. Bacelli, R. Genest, and J. V. Ringwood, "Nonlinear control of flap-type wave energy converter with a non-ideal power take-off system," *Annual Reviews in Control*, vol. 40, pp. 116–126, 2015.

[54] V. Heller, "Scale effects in physical hydraulic engineering models," *Journal of Hydraulic Research*, vol. 49, no. 3, pp. 293–306, 2011.

[55] W. Sheng, R. Alcorn, and T. Lewis, "Physical modelling of wave energy converters," *Ocean Engineering*, vol. 84, pp. 29–36, 2014.

[56] L. O'Boyle, K. Doherty, J. van't Hoff, and J. Skelton, "The value of full scale prototype data-testing oyster 800 at emec, orkney," in *Proceedings of the 11th European wave and tidal energy conference (EWTEC), Nantes, France*, 2015, pp. 6–11.

[57] B. Lydon, B. Polagye, and S. Brunton, "Data-driven modeling of an oscillating surge wave energy converter using dynamic mode decomposition," *Journal of Renewable and Sustainable Energy*, vol. 17, no. 2, 2025.

[58] K. Ruehl *et al.*, *Wec-sim v5.0.1*, version v5.0.1, 2022. DOI: 10.5281/zenodo.7121186. [Online]. Available: https://zenodo.org/badge/latestdoi/20451353.

[59] M. Lawson, Y.-H. Yu, K. Ruehl, and C. Michelen, "Development and demonstration of the wec-sim wave energy converter simulation tool," 2014.

[60] J. N. Kutz, *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013.

[61] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[62] M. Folley, *Numerical modelling of wave energy converters: state-of-the-art techniques for single devices and arrays*. Academic Press, 2016.

[63] J. van't Hoff, M. Folley, and T. Whittaker, "Numerical modelling of an oscillating wave surge converter using volterra theory," in *Proceedings of the 11th European Wave and Tidal Energy Conference, Nantes, France*, 2015, pp. 6–11.

[64] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Sparse identification of nonlinear dynamics for model predictive control in the low-data limit," *Proceedings of the Royal Society of London A*, vol. 474, no. 2219, 2018.

[65] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition with control," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.

[66] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Sparse identification of nonlinear dynamics with control (sindyc)," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 710–715, 2016.

[67] U. Fasel, E. Kaiser, J. N. Kutz, B. W. Brunton, and S. L. Brunton, "Sindy with control: A tutorial," in *2021 60th IEEE conference on decision and control (CDC)*, IEEE, 2021, pp. 16–21.

[68] P. Andersen, T. S. Pedersen, K. M. Nielsen, and E. Vidal, "Model predictive control of a wave energy converter," in *2015 IEEE Conference on Control Applications (CCA)*, IEEE, 2015, pp. 1540–1545.

[69] M. Richter, M. E. Magaña, O. Sawodny, and T. K. Brekken, "Power optimisation of a point absorber wave energy converter by means of linear model predictive control," *IET Renewable Power Generation*, vol. 8, no. 2, pp. 203–215, 2014.

[70] A. Mérigaud and J. V. Ringwood, "Optimal trajectories, nonlinear models and constraints in wave energy device control," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15 645–15 650, 2017.

[71] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.

[72] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of fluid mechanics*, vol. 641, pp. 115–127, 2009.

[73] J. H. Tu, "Dynamic mode decomposition: Theory and applications," Ph.D. dissertation, Princeton University, 2013.

[74] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

[75] P. J. Schmid, "Application of the dynamic mode decomposition to experimental data," *Experiments in fluids*, vol. 50, no. 4, pp. 1123–1130, 2011.

[76] I. Mezić, "Analysis of fluid flows via spectral properties of the koopman operator," *Annual Review of Fluid Mechanics*, vol. 45, pp. 357–378, 2013.

[77] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[78] B. Sharan, A. Dittmer, and H. Werner, "Real-time model predictive control for wind farms: A koopman dynamic mode decomposition approach," in *2022 European Control Conference (ECC)*, IEEE, 2022, pp. 1006–1011.

[79] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space," *Proceedings of the National Academy of Sciences*, vol. 17, no. 5, pp. 315–318, 1931.

[80] I. Mezić, "Spectral properties of dynamical systems, model reduction and decompositions," *Nonlinear Dynamics*, vol. 41, no. 1-3, pp. 309–325, 2005.

[81] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz, "Chaos as an intermittently forced linear system," *Nature Communications*, vol. 8, no. 19, pp. 1–9, 2017.

[82] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, "Modern Koopman theory for dynamical systems," *SIAM Review*, vol. 64, no. 2, pp. 229–340, 2022.

[83] M. J. Colbrook, "The mpedmd algorithm for data-driven computations of measure-preserving dynamical systems," *arXiv preprint arXiv:2209.02244*, 2022.

[84] M. J. Colbrook, L. J. Ayton, and M. Szőke, "Residual dynamic mode decomposition: Robust and verified koopmanism," *Journal of Fluid Mechanics*, vol. 955, A21, 2023.

[85] K. K. Chen, J. H. Tu, and C. W. Rowley, "Variants of dynamic mode decomposition: Boundary condition, koopman, and fourier analyses," *Journal of nonlinear science*, vol. 22, no. 6, pp. 887–915, 2012.

[86] P. J. Schmid, "Dynamic mode decomposition and its variants," *Annual Review of Fluid Mechanics*, vol. 54, pp. 225–254, 2022.

[87] M. S. Hemati, C. W. Rowley, E. A. Deem, and L. N. Cattafesta, "De-biasing the dynamic mode decomposition for applied koopman spectral analysis of noisy datasets," *Theoretical and Computational Fluid Dynamics*, vol. 31, no. 4, pp. 349–368, 2017.

[88] S. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley, "Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition," *Experiments in Fluids*, vol. 57, no. 3, pp. 1–19, 2016.

[89] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the national academy of sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[90] A. R. Hartman, *Development of a Laboratory Oscillating Surge Wave Energy Converter*. University of Washington, 2022.

[91] A. Serani, P. Dragone, F. Stern, and M. Diez, "On the use of dynamic mode decomposition for time-series forecasting of ships operating in waves," *Ocean engineering*, vol. 267, p. 113 235, 2023.

[92] M. Diez, A. Serani, E. F. Campana, and F. Stern, "Time-series forecasting of ships maneuvering in waves via dynamic mode decomposition," *Journal of Ocean Engineering and Marine Energy*, pp. 1–8, 2022.

[93] Y. Jia, J. Sun, Z. Xu, C. Sun, K. Meng, and Z. Y. Dong, "Data-driven economic mpc of a point absorber wave energy converter," *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*, 2024.

[94] N. Faedo, S. Olaya, and J. V. Ringwood, "Optimal control, mpc and mpc-like algorithms for wave energy systems: An overview," *IFAC Journal of Systems and Control*, vol. 1, pp. 37–56, 2017.

[95] M. Lawson, Y.-H. Yu, A. Nelessen, K. Ruehl, and C. Michelen, "Implementing nonlinear buoyancy and excitation forces in the wec-sim wave energy converter modeling tool," in *International Conference on Offshore Mechanics and Arctic Engineering*, American Society of Mechanical Engineers, vol. 45547, 2014, V09BT09A043.

[96] J. L. Proctor and P. A. Eckhoff, "Discovering dynamic patterns from infectious disease data using dynamic mode decomposition," *International health*, vol. 7, no. 2, pp. 139–145, 2015.

[97] N. Mohan, K. Soman, and S. S. Kumar, "A data-driven strategy for short-term electric load forecasting using dynamic mode decomposition model," *Applied energy*, vol. 232, pp. 229–244, 2018.

[98] J. N. Kutz, S. L. Brunton, B. Brunton, and J. L. Proctor, *Dynamic mode decomposition*, 2016. [Online]. Available: http://dmdbook.com/.

[99] J. H. Tu, C. W. Rowley, J. N. Kutz, and J. K. Shang, "Spectral analysis of fluid flows using sub-nyquist-rate piv data," *Experiments in Fluids*, vol. 55, no. 9, pp. 1–13, 2014.

[100] D. Duke, J. Soria, and D. Honnery, "An error analysis of the dynamic mode decomposition," *Experiments in fluids*, vol. 52, no. 2, pp. 529–542, 2012.

[101] S. Bagheri, "Effects of weak noise on oscillating flows: Linking quality factor, floquet modes, and koopman spectrum," *Physics of Fluids*, vol. 26, no. 9, p. 094 104, 2014.

[102] J. Juang and R. Pappa, "An eigensystem realization algorithm (era) for modal parameter identification and model reduction," in *JPL Proc. of the Workshop on Identification and Control of Flexible Space Struct., Vol. 3*, 1985.

[103] T. Askham and J. N. Kutz, "Variable projection methods for an optimized dynamic mode decomposition," *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 1, pp. 380–416, 2018.

[104] J. Thomson, "Wave breaking dissipation observed with "swift" drifters," *Journal of Atmospheric and Oceanic Technology*, vol. 29, no. 12, pp. 1866–1882, 2012.

[105] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data–driven approximation of the koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, pp. 1307–1346, 2015.

[106] B. M. de Silva, K. Champion, M. Quade, J.-C. Loiseau, J. N. Kutz, and S. L. Brunton, "Pysindy: A python package for the sparse identification of nonlinear dynamics from data," *arXiv preprint arXiv:2004.08424*, 2020.

[107] B. Lydon, B. Polagye, and S. Brunton, "Nonlinear wec modeling using sparse identification of nonlinear dynamics (sindy)," in *Proceedings of the European Wave and Tidal Energy Conference*, vol. 15, 2023.

[108] Y. Liu, S. Shi, Y. Fang, and A. Fu, "Robust data-driven wave excitation force estimation for wave energy converters with nonlinear probabilistic modelling," *Ocean Engineering*, vol. 310, p. 118 726, 2024.

[109] A. A. Kaptanoglu *et al.*, "Pysindy: A comprehensive python package for robust sparse system identification," *arXiv preprint arXiv:2111.08481*, 2021.

[110] G. J. Leishman, *Principles of helicopter aerodynamics with CD extra*. Cambridge university press, 2006.

[111] P. Schmitt and B. Elsäßer, "The application of froude scaling to model tests of oscillating wave surge converters," *Ocean Engineering*, vol. 141, pp. 108–115, 2017.

[112] H. Asmuth, P. Schmitt, A. Henry, and B. Elsaesser, "Determination of non-linear damping coefficients of bottom-hinged oscillating wave surge converters using numerical free decay tests," in *Renew 2014*, 2014.

[113] J. Lee, N. Xiros, and M. Bernitsas, "Virtual damper–spring system for viv experiments and hydrokinetic energy conversion," *Ocean Engineering*, vol. 38, no. 5-6, pp. 732–747, 2011.

[114] L. D. DeVries, L. Luznik, and S. Malek, "Modeling, system identification, and friction compensation of an actuator system for heave and surge wave energy converter experiments," *Journal of Ocean Engineering and Marine Energy*, pp. 1–20, 2024.

[115] *WAMIT User Manual Version 7.5*. Massachusetts Institute of Technology, 2013.

[116] J. L. Guerrero-Fernández, N. M. Tom, and J. A. Rossiter, "Nonlinear model predictive control based on real-time iteration scheme for wave energy converters using wec-sim," in *International Conference on Offshore Mechanics and Arctic Engineering*, American Society of Mechanical Engineers, vol. 85932, 2022, V008T09A076.

[117] A. Fisher, J. Thomson, and M. Schwendeman, "Rapid deterministic wave prediction using a sparse array of buoys," *Ocean Engineering*, vol. 228, p. 108 871, 2021.

[118] S. Kass, "Exploring the limits of boundary element methods for wave energy converter hydrodynamics," M.S. thesis, University of Washington, 2024.

[119] M. Riddle, "Cfd modeling of an oscillating wave surge converter using the overset grid method," M.S. thesis, University of Washington, 2022.

[120] A. O. Winter and M. R. Motley, "Development of a fluid-structure interaction model of an oscillating wave surge converter using openfoam," in *International Conference on Offshore Mechanics and Arctic Engineering*, American Society of Mechanical Engineers, vol. 84416, 2020, V009T09A034.

[121] O. Ltd. "Openfoam: User guide v2112." (2019), [Online]. Available: https://www.openfoam.com/documentation/guides/latest/doc/guide-overset.html.

[122] H. Chen *et al.*, "Application of an overset mesh based numerical wave tank for modelling realistic free-surface hydrodynamic problems," *Ocean Engineering*, vol. 176, pp. 97–117, 2019.

[123] Z. Lin, H. Chen, L. Qian, Z. Ma, D. Causon, and C. Mingham, "Simulating focused wave impacts on point absorber wave energy converters," *Proceedings of the Institution of Civil Engineers-Engineering and Computational Mechanics*, vol. 174, no. 1, pp. 19–31, 2021.

[124] E. Katsidoniotaki and M. Göteman, "Comparison of dynamic mesh methods in openfoam for a wec in extreme waves," in *Developments in Renewable Energies Offshore*, CRC Press, 2020, pp. 214–222.

[125] C. Windt, J. Davidson, B. Akram, and J. V. Ringwood, "Performance assessment of the overset grid method for numerical wave tank experiments in the openfoam environment," in *International Conference on Offshore Mechanics and Arctic Engineering*, American Society of Mechanical Engineers, vol. 51319, 2018, V010T09A006.

[126] C. Windt, J. Davidson, D. D. Chandar, N. Faedo, and J. V. Ringwood, "Evaluation of the overset grid method for control studies of wave energy converters in openfoam numerical wave tanks," *Journal of Ocean Engineering and Marine Energy*, vol. 6, no. 1, pp. 55–70, 2020.

[127] D. P. Coiro, G. Troise, G. Calise, and N. Bizzarrini, "Wave energy conversion through a point pivoted absorber: Numerical and experimental tests on a scaled model," *Renewable Energy*, vol. 87, pp. 317–325, 2016.

[128] J. van Rij, Y.-H. Yu, Y. Guo, and R. G. Coe, "A wave energy converter design load case study," *Journal of Marine Science and Engineering*, vol. 7, no. 8, p. 250, 2019.

[129] E. Renzi and F. Dias, "Hydrodynamics of the oscillating wave surge converter in the open ocean," *European Journal of Mechanics-B/Fluids*, vol. 41, pp. 1–10, 2013.

[130] E. Renzi and F. Dias, "Resonant behaviour of an oscillating wave energy converter in a channel," *Journal of Fluid Mechanics*, vol. 701, pp. 482–510, 2012.

[131] P. B. Kahn and Y. Zarmi, *Nonlinear dynamics: exploration through normal forms*. Courier Corporation, 2014.

[132] J. E. Marsden and M. McCracken, *The Hopf bifurcation and its applications*. Springer Science & Business Media, 2012, vol. 19.