AN ABSTRACT OF THE THESIS OF

Mitchell Colby for the degree of Master of Science in Mechanical Engineering presented on March 12, 2012.

Title:

Optimizing Ballast Design of Wave Energy Converters Using Evolutionary Algorithms

Abstract approved: ____

Kagan Tumer

Wave Energy Converters (WECs) promise to be a viable alternative to current electrical generation methods. However, these WECs must become more efficient before wide-scale industrial use can become cost-effective. The efficiency of a WEC is primarily dependent upon its geometry and ballast configuration which are both difficult to evaluate, due to slow computation time and high computation cost of current models. In this thesis, we use evolutionary algorithms to optimize the ballast geometry of a wave energy generator using a two step process. First, we generate a function approximator (neural network) to predict wave energy converter power output with respect to key geometric design variables. This is a critical step as the computation time of using a full model (e.g., AQWA) to predict energy output prohibits the use of an evolutionary algorithm for design optimization. The function approximator reduced the

computation time by over 99% while having an average error of only 3.5%. The evolutionary algorithm optimized the weight distribution of a WEC, resulting in an 84% improvement in power output over a ballast-free WEC.

Optimizing Ballast Design of Wave Energy Converters Using Evolutionary Algorithms

by

Mitchell Colby

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented March 12, 2012
Commencement June 2012

Master of Science thesis of Mitchell Colby presented on March 12, 2012.

APPROVED:

Major Professor, representing Mechanical Engineering

Head of the School of Mechanical, Industrial, and Manufacturing Engineering

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Mitchell Colby, Author

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Kagan Tumer, for his continual support and guidance throughout this project. I would also like to thank everyone at Columbia Power, specifically Ken Rhinefrank, Al Schacher, and Stephen Zhang for their help with the AQWA simulator. I also thank my father, Robert Colby, for his constant advice on coding. Finally, I would like to thank Whitney O'Dell, for her continued patience with me as I have spent so much time on this project.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

Chapter 1 – Introduction

Due to the increase in population, demand for electricity, and the desire for economical stability, there is a high demand for renewable energy sources that are both environmentally friendly and economically viable. Waves provide an energy source that satisfies both the political and economical demands, and can reduce the environmental impact of power generation [15, 17]. Wave energy belongs to a group of energy sources which are easy to access, but require a complex process to capture.

The issues of harnessing waves as a viable source of energy include the complicated nature of predicting the wave climate, as well as the difficulties associated with optimizing a Wave Energy Converter (WEC) for a particular wave climate. System parameters including inertia, center of gravity position, system draft, and submerged volume all affect the power output of a WEC. Mass parameters such as inertia and center of gravity position can be manipulated during operation by utilizing ballast, thus affecting system output. Because of the complex simulations required, testing different design parameters is very time consuming, which limits the number of designs which may be considered.

The physics governing the interactions between a WEC and the sea prohibit developing and solving analytical expressions for mapping a WEC design to its power output. Although the equations of motion may be developed analytically, they are nonlinear partial differential equations which cannot be solved analytically. In order to judge the effectiveness of a WEC, a numerical solution must be developed. Hydrodynamic simulators, such as Ansys AQWA, are capable of simulating different WEC designs. However, these simulators are extremely time consuming, due to the fact that energy and momentum equations are solved thousands, or millions, of times during one simulation [1]. The computational cost of numerical simulations severely limits the number of designs which may be considered, prohibiting the use of search algorithms to optimize the WEC design. In general, human experts guide the design process, which imparts bias into the design. Ideally, the computational time required to analyze a WEC should be reduced, in order to utilize an unbiased search algorithm to optimize the design.

In order to reduce computational time, we developed a function approximation that maps the design configuration of a WEC to its power output. A new time domain simulator is created using this function approximation, and is orders of magnitude faster than the simulation in Ansys AQWA. Evolutionary algorithms search through the design space to find an optimal ballast configuration, using the time domain simulator to assign fitness to individual solutions. An optimal ballast configuration for the WEC is found in much less time than using traditional hydrodynamic analysis methods. Optimizing the ballast design for a WEC is an essential component of the design process in order to ensure that the WEC is cost effective.

The geometry of the WEC being analyzed has been defined and was developed by Columbia Power Technologies. With this defined geometry, parameters such as system draft and submerged volume are set and cannot be changed. However, the mass parameters may be changed during operation in order to affect power output. The power output of this particular WEC has been shown in our simulations to be most sensitive to the inertia and center of gravity positions of each of its components. Installing ballast chambers in the WEC allows us to modify those values during the course of operation, which alters the power output of the device. However, there are countless possibilities for the ballast chamber configuration, so some type of search must be completed to find an optimal ballast configuration which maximizes the WEC power output. An evolutionary search over ballast chamber configurations yields a WEC design which is far superior to a ballast free model.

In this thesis, we develop a neural network function approximator that maps the mass parameters of each component of the WEC to its power output. Next, a time domain simulator is created with this function approximator, which predicts the annual energy output of a WEC given its ballast configuration and wave climate it operates in. Finally, an evolutionary algorithm intelligently searches through the set of potential ballast configurations, using the time domain simulator to assign a fitness to each design. This ultimately leads to the optimal ballast configuration within the WEC.

In order to optimize the ballast configuration of the WEC to maximize energy output for a specific wave climate, we complete the following steps:

- Find the appropriate geometry for the WEC

- Simulate that geometry with different mass parameters using Ansys AQWA
- Use the simulated data to train the neural network function approximator
- Create a time domain simulator based on the neural network function approximator
- Search through the different ballast configurations using an evolutionary algorithm, where the fitness of each member is found using the time domain simulator

1.1 Contributions

The contribution of this paper involves the following elements:

1. Approximating the power output of a WEC as a function of design parameters and the sea state
2. Performing active data collection for generating an accurate function approximation
3. Creating a time based control algorithm of ballast chambers for optimizing energy capture
4. Creating a computationally efficient time-domain simulator to determine the energy capture of a specific WEC geometry while varying ballast configurations

5. Using an evolutionary algorithm to optimize the ballast configuration of a WEC

The remainder of this paper is organized as follows. Chapter 2 discusses background material and related work. Chapter 3 discusses the ballast configuration of the WEC and the function approximation based on AQWA data. Chapter 4 explains the time domain simulator used to replace AQWA. Chapter 5 details the search algorithms used to optimize the ballast configuration of the WEC. Chapter 6 gives the experiments and results from this research. Finally, Chapter 7 discusses the results and presents opportunities for future work.

Chapter 2 – Background

The following sections give background information on wave energy, simulation software, neural networks, and evolutionary algorithms. Related work is also discussed.

2.1 Wave Energy

WECs have been investigated for decades, and designs from the 1970s such as Salter's Duck are still in use [14]. Due to the complexities involved in WEC design, coupled with the fact that hydrodynamic simulators are very time-consuming, most WEC design processes are ad-hoc in which existing designs are slightly modified and tested in specific wave climates [13]. Due to the nature of the ad-hoc designs, formal algorithms for optimizing WECs do not exist (to the knowledge of the author). Often, new WEC designs are simply modified models of existing wave energy converters [6].

While our research does concentrate on optimizing one WEC geometry, we do not implement an ad-hoc algorithm. The use of an evolutionary algorithm in the design process removes the need for expert domain knowledge, because the algorithm automatically searches the design space in an intelligent manner. Thus, there are two key differences between the traditional WEC design process and the process we implement. First, the design is guided by computational evolution, rather than a human expert. Secondly, we use function approximators to replace simulations, drastically increasing the number of designs which may be considered. The function approximation is the key to allowing the evolutionary algorithm to be computationally feasible. In order to create the function approximator, simulation data from a hydrodynamic simulator is needed to create a data set for training.

2.2 Simulating Wave Energy Converters Power Output

Traditionally, hydrodynamic simulators are generally used to determine the effectiveness of a WEC during the design process. Although these simulators are effective for judging a single design, they are computationally expensive and limit the number of designs which may be considered. In order to analyze more designs, a faster simulator is required. The following sections introduce ANSYS AQWA, as well as a methodology for speeding up WEC simulations.

2.2.1 Ansys AQWA

Ansys AQWA is a powerful software package which simulates bodies floating in the ocean. Unlike traditional computational fluid dynamics packages, which utilize finite volume analysis, AQWA utilizes panel analysis to determine the hydrodynamic response of offshore and marine structures. Running a simulation in AQWA consists of generating a surface mesh, defining the mass properties of the body, and defining the wave climate [1]. The AQWA simulation outputs the hyrodynamic response of the defined body in a specified wave climate.

Although AQWA is an extremely powerful tool, computation time is very slow, often taking over 24 hours to run a single simulation to evaluate a WEC's power output. In order to test multiple WEC configurations, weeks or months are required. The computational expense of AQWA limits the number of WEC configurations that may be considered in an analysis, which severely limits the design process. When using a simulator such as AQWA, the limit on the number of designs which may be considered necessitates a focused search in the design space, meaning that only a small fraction of potential designs may be considered.

2.2.2 Approximate Simulations with Neural Networks

Simulators such as AQWA are too computationally expensive to analyze many different WEC designs. However, if certain parameters of the WEC are set, then the power output of the WEC can be approximated with minimal use of AQWA. The geometry of the WEC being analyzed is set, and only the mass parameters are altered. A WEC with a constant geometry can be simulated in AQWA while varying mass parameters, resulting in a dataset which can be used to train a function approximator. Once the function approximator is trained, it can be used to evaluate the power output of a WEC without the computational expense associated with AQWA. In this thesis, an artificial neural network is used to approximate the power output of the WEC.

An artificial neural network, or neural network, is a mathematical model which is inspired by biological neural networks in the brain. A neural network represents a nonlinear mapping from a set of inputs to a set of outputs. In a neural network, multiple layers of nodes are connected with sets of weights.

Consider a neural network with n_i input nodes, n_h hidden nodes, and n_o output nodes. Each node is connected with a link which has a weight $w_{i,j}$, which is the weight from node i to node j. The algorithm for determining the output $\vec{y}$ given an input $\vec{x}$ is detailed in Algorithm 1.

Algorithm 1: Neural network mapping from input $\vec{x}$ to output $\vec{y}$

input: $\vec{x} \in \mathbb{R}^{n_i}$;
for $i = 1$ *to* n_i **do**
  $\vec{z}_{1,i} = \sigma(\vec{x}_i)$;
end
for $j = 1$ *to* n_h **do**
  $\vec{z}_{2,j} = \sum_{i=1}^{n_h} w_{i,j} \vec{z}_{1,i}$;
end
for $i = 1$ *to* n_h **do**
  $\vec{z}_{3,j} = \sigma(\vec{z}_{2,j})$;
end
for $j = 1$ *to* n_o **do**
  $\vec{z}_{4,j} = \sum_{i=1}^{n_o} w_{i,j} \vec{z}_{3,i}$;
end
for $i = 1$ *to* n_o **do**
  $\vec{y}_i = \sigma(\vec{z}_{4,i})$;
end
return: $\vec{y}$;

The key step in developing a neural network is to learn the weights $w_{i,j}$. A common method to determine the weights is backpropogation [8]. The general methodology for backpropogation is to determine the gradient of the neural network error with respect to the weights, and to use gradient descent to adjust the weights in order to minimize the error.

Neural networks have been used in many applications, including control, system identification, function approximation, and nonlinear signal processing [4, 10, 12, 16]. Neural networks are excellent tools for finding maps from inputs to outputs from datasets, and have been shown to be capable of finding these mappings in highly nonlinear datasets [4]. A key advantage to using neural networks as function approximators is that once trained, a neural network can be evaluated to make predictions almost instantaneously. This computational speed is a crucial component when performing a population based search such as an evolutionary algorithm.

2.3 Time Domain Simulations

Once the power output of a WEC is determined as a function of the WEC design and the sea state, we can determine the energy captured by the WEC over a given interval of time. After defining the power output of the WEC as a function of certain device parameters, a time-domain simulation is completed by simulating the WEC operating in the sea and keeping track of instantaneous power rates. The power output information in AQWA can be used to find the energy capture of the WEC given a controller and the sea state the WEC operates in. The following sections describe the approach to time domain simulations used in AQWA, as well as how the function approximation developed in Section 2.2.2 is used to create a time domain simulator.

2.3.1 Traditional Simulators

AQWA may be used to run a time domain simulation once the power output of the device as a function of the sea state is determined. Once the WEC power output $\dot{W}_{out}$ is known as a function of the dominant wave frequency ω_D, the energy captured $E_{\Delta t}$ at any time step Δt is:

$$E_{\Delta t} = \dot{W}_{out} \cdot \Delta t \tag{2.1}$$

In a time-domain simulation with discrete time steps, the energy captured at each time step is totaled and the result is the total energy capture over the specified time interval. The time domain simulations take an extremely long time in AQWA, because the energy and momentum equations solved in the frequency domain, and the power rates are combined with a controller and specified sea state in order to determine the total energy capture. In Section 2.2.2, we showed that a neural network could learn the frequency domain power output. This neural network function approximator replaces the power calculations in AQWA and allows for the creation of a fast time domain simulator.

2.3.2 Function Approximation Based Simulators

Given a neural network which defines the WEC design to power output mapping, a time domain simulation may be quickly completed. The general time domain simulator algorithm is given in Algorithm 2

Algorithm 2: General time domain simulation algorithm

```
time t = 0 ;
energy E = 0 ;
while t ⩽ t_max do
    Power Calc: find P with neural network ;
    Energy Update: E = E + P · Δt ;
    t = t + Δt
end
return: E ;
```

This time domain simulation follows the same algorithm used in AQWA. However, the frequency domain calculations in AQWA are replaced by the neural network function approximation, which drastically decreases computational time. With the faster time domain simulation, more WEC configurations may be analyzed.

2.4 Optimization of a Wave Energy Converter

The following sections describe traditional methods for optimizing WEC designs, as well as learning based methods to optimize WEC designs.

2.4.1 Traditional Optimization

The physics governing a WEC's interactions with a sea environment cannot be derived and solved analytically. For this reason, traditional optimization techniques such as Lagrange multipliers are not viable options for optimizing the design of a WEC. Traditional techniques to optimize the design of WECs involve a human expert driving the design process [17]. In this case, the general design of the WEC is known prior to the optimization process, and parameters are slightly modified to increase the performance of the WEC. Although these methods typically produce acceptable WEC designs, the search space of potential designs is severely limited; only WECs from a small subset of the possible design space are considered. This is due to two factors. First, the time associated with simulating different designs limits the number of designs which may be considered, driving the design process to local searches. Secondly, the human experts impart bias to the search, using their experience to drive the design process. The limitation of the search space is the key limitation to the traditional optimization process. In order to find an optimal design, the search should not be limited to a subset of the design space. In order for such an optimization process to be completed, a method for speeding up simulations must be developed, and the search must be unbiased by human factors.

2.4.2 Optimization with Evolutionary Algorithms

Evolutionary algorithms are stochastic population based searches inspired by biological evolution. An evolutionary algorithm starts with a population of candidate solutions. Each solution in the population creates a mutated copy of itself. The value of each solution is determined through a fitness function, and the individuals with the highest fitness are selected to proceed to the next generation. The general evolutionary algorithm is detailed in Algorithm 3.

Algorithm 3: General evolutionary algorithm

Randomly initialize a population of n candidate solutions ;
generation $G = 1$;
while $G \leqslant G_{max}$ **do**
    Create successor solutions from population ;
    Mutate successor solutions ;
    Calculate fitness of each solution ;
    Select n solutions to succeed to next generation ;
end

Evolutionary algorithms have been successfully implemented to optimize design processes, including autonomously designing antennas, finding optimal gaits of quadruped robots, solving offset assignment problems for embedded processors, and developing distributed control policies in multi-component systems [2, 7, 9, 11]. Evolutionary algorithms have proven to be useful in design in many different domains. These designs indicate that evolutionary algorithms guiding design processes offer many advantages, including an autonomous process and the fact that no domain expert is necessary. Often, as design problems become more complex, a domain expert may impart bias into the design process, greatly limiting the potential designs which may be considered. With an evolutionary algorithm, designs which a domain expert would never consider have the potential to be considered. Often, in these more complex domains, the evolutionary algorithms produce exceptional results when compared to traditional design methods [2, 5, 20]. In the context of WEC design, an evolutionary algorithm is preferable to a traditional optimization process because it searches the entire design space and does not impart human bias into the search. However, for an evolutionary algorithm to be a computationally feasible method for designing a WEC, the process of simulating WECs must be sped up considerably. This can be attained through function approximation as described in Section 2.2.2.

Chapter 3 – Wave Energy Converter Power Output

The specific geometry of the WEC for analysis was developed by Columbia Power Incorporated. The WEC is a heaving and pitching design, which is based on the concept of the components resonating with the dominant wave frequency experienced in a given wave climate [13]. The WEC has three components as shown in Figure 3.1. The components are the spar, the forward float, and the aft float.

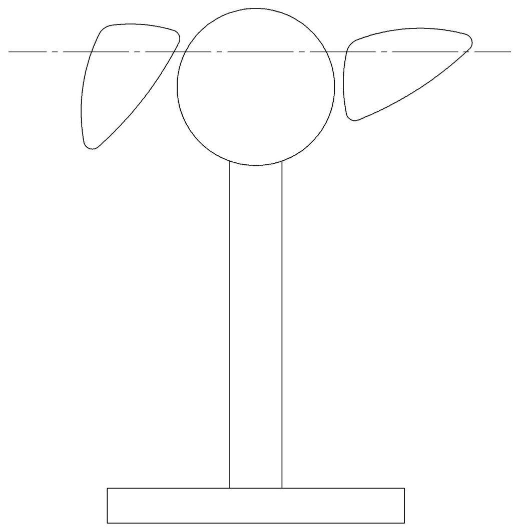

Figure 3.1: Columbia Power Manta WEC, which has three sections: the spar (the middle rode), aft float (the top right float), and forward float (the top left float). The dashed line indicates the level-sea waterline. Waves approach the WEC from the right of the figure.

The floats utilized on the WEC are variations of Salter's Duck design, which has been shown to be capable of capturing 90% of a single wave's energy in idealized climates [14]. Although Salter's Duck float design is extremely efficient in an ideal climate, this ideal climate consists of constant frequency waves moving in one direction. This is not a realistic model for operation in the sea. In a true sea climate, the efficiency of the Duck design decreases significantly. It is impractical to change the WEC geometry during operation, but ballast can easily be altered in order to change mass parameters and alter the power output. For a given geometry, it is essential that an optimal ballast configuration is found, in order to maximize energy extraction from a given sea state.

3.1 Ballast Configurations

In order to determine the effect of ballast on WEC mass parameters, a relationship between ballast and mass parameters was developed. The center of gravity position of each component may be determined if the ballast configuration of the WEC is known as follows:

$$CG_z = \frac{1}{M} \int_V \rho(r) r dV \tag{3.1}$$

where M is the total mass of the component, r is the position of an infinitesimal volume dV, and $\rho(r)$ is the density of the infinitesimal volume. Thus, defining the ballast configuration of a device allows for the center of gravity to be computed directly. The inertia of a component about its center of mass is defined as:

$$I_{yy} = \int_M r^2 dm \tag{3.2}$$

where M is the total mass of the component, r is the distance from a point to the center of mass, and dm is the infinitesimal mass at that point. Given a ballast configuration and mass of water in each ballast chamber, the center of mass of a component may be determined directly, and the inertia about the center of mass may then be computed. Thus, there is a simple and direct mapping from the ballast configuration of a WEC to its mass parameters.

The entire volume of the WEC is not available for use for ballast chambers. Practical constraints, such as placement of the batteries, the generator, electrical equipment, and mass dampers limit the portion of the WEC which may be used for ballast chamber placement. Regions available for use as ballast chambers are shown in Figure 3.2. The main stem of the spar is available for use as a ballast chamber, as are the fore and aft float bodies. The objective of this work is to split these volumes into separate ballast chambers such that the annual power output of the WEC is optimized, given a set number of total ballast cuts. The reason that the number of ballast cuts is not a design variable is that if included as a design variable, the number of ballast chambers would tend to infinity during the optimization process. An infinite number of infinitesimal chambers which can instantly fill and empty would allow for instantaneous large changes of the inertia and center of gravity values, which is mathematically ideal to optimize the power output but is impractical from a manufacturing or cost standpoint.

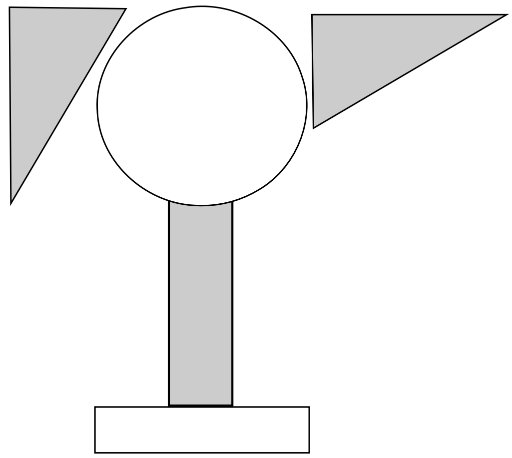

Figure 3.2: Shaded areas indicate ballast chamber locations

The volumes available for use as ballast are cut with planes to create the separate ballast chambers, shown in Figure 3.3. The variables which define these cuts are the position at which the cuts intercept the overall ballast chamber volume. Given a set of cuts, a mapping from the mass of seawater in each chamber to the overall component I_{yy} and CG_z values is found using the methods explained previously. Thus, for a given set of cuts, the inertia and center of mass in each component is simply a function of the ballast mass in each chamber

3.2 Function Approximation

The following sections describe the process of developing the neural network function approximator which maps WEC design parameters to a power output. Simulations in AQWA are run in order to create an initial data set, and active data collection is used in order to minimize neural network error.

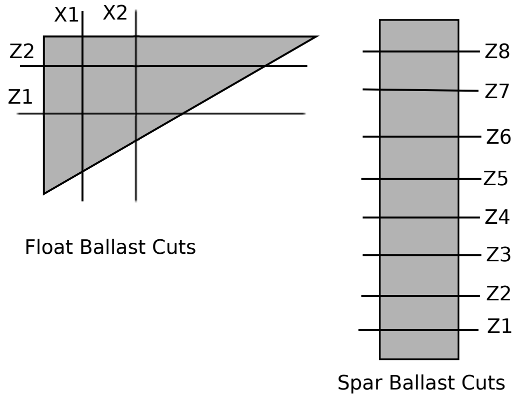

Figure 3.3: Cuts which create ballast chambers

3.2.1 Neural Network Training

The power output of this particular WEC has been shown to be most sensitive to the y-moment of inertia (I_{yy}) and vertical center of gravity position (CG_z) of each of the device components. The inertia parameters are important because the WEC design produces power when the components rotate about the y-axis, and the center of gravity positions are important because they affect hydrodynamic stability [19]. After defining the feasible bounds for each of the mass parameters (I_{yy} and CG_z for each component), the input space was discretized into 250 points, and each of these inputs were simulated in AQWA. The simulations produced a data set relating the power output of the WEC to the mass parameter configuration, as well as the dominant wave frequency of the sea (assuming a specified wave climate including dominant wave frequencies).

In order to perform a large search through different designs, a function approximator is necessary to reduce computational time associated with the simulations.

AQWA is capable of finding the power output of the WEC, based upon the I_{yy} and CG_z parameters of each component. However, the simulation process is extremely slow, taking up to 24 hours to complet a single simulation. In order to speed up the simulation process, a neural network is trained as a function approximator of the power output of the AQWA simulations, for the particular set of geometries characteristic of this particular WEC.

A neural network was used because the data set mapping mass parameters to power outputs is highly nonlinear, and neural networks are excellent choices for finding patterns in nonlinear datasets [10, 16, 18]. Simpler function approximation techniques, including linear and polynomial regression, were unable to produce accurate mappings of the dataset in our experiments.

The neural network maps the mass parameters (I_{yy} and CG_z) and dominant wave frequency to the WEC power output. It is important to note that this approximator is a substitute for the part of AQWA that simulates power output based on WEC mass parameters. In addition, it is only suited for the geometry of this particular WEC, so we are not replacing the entire AQWA simulator. This function approximation was designed to reduce the simulation time for a specific WEC geometry and wave climate, which is over 99% reduction from the original AQWA simulation run times.

We used the AQWA simulator to generate a set of training data for the neural network. First, we determined the manufacturability bounds on I_{yy} and CG_z values (determined by Columbia Power Technologies) in order to keep the training data in a realistic portion of the input space. The input space was discretized into 250 sets of data points (I_{yy},CG_z), and each set was simulated in AQWA to find the resultant power output. Given the data set, we needed to determine the appropriate number of hidden units to use in the neural network. The training data was fed into three separate neural networks with 10, 15, and 20 hidden units. The validation error on all three hidden units had an average error of at least 15% per data point, with maximum errors reaching up to 43%. The initial results for neural network training are listed in Table 3.1.

Data Points	Hidden Units	Max Error	Mean Error	Portion of Data Set with less than 5% Error
250	10	$42.3 \pm 0.05\%$	$24.3 \pm 0.02\%$	$22 \pm 0.02\%$
250	15	$36.5 \pm 0.03\%$	$18.2 \pm 0.06\%$	$36 \pm 0.04\%$
250	20	$34.4 \pm 0.06\%$	$15.3 \pm 0.05\%$	$42 \pm 0.03\%$

Table 3.1: Validation error for the neural network function approximator from initial data set; the best network has a maximum error of 34.4% and mean error of 15.3%. Reported errors are generated from 10-fold cross validation

3.2.2 Active Data Collection

As seen in Table 3.1, the function approximator is a poor approximation for the data set using 250 training points. In order to reduce this error, an active data collection is completed. In the regions of the input space with highest error, 150 new data points were introduced, resulting in 400 total data points. The 150 new points were simulated in AQWA, and the neural network was retrained with the new data set. The networks with 10 and 15 hidden units still had high error, but the 20 hidden unit neural network had an average error of 1.5% and a maximum error of 6.12%. This accuracy is sufficient to utilize the function approximator to replace AQWA in order to make quantitative distinctions between different WEC configurations. The function approximator allows for an exhaustive search through the entire input space, allowing for near instantaneous mapping of a WEC configuration to its power output. The neural network can produce an output in less than one second, which is over $86,000$ times faster than the AQWA simulations. Further, once the dataset is created, training the neural network takes only minutes. Thus, given a ballast configuration, we can determine the mass parameters of the WEC and find the associated power output in near real time. The general algorithm for neural network training with active data collection is shown in Figure 3.4, and the results of neural network training are shown in Table 3.2.

Data Points	Hidden Units	Max Error	Mean Error	Portion of Data Set with less than 5% Error
250	10	$42.3 \pm 0.05\%$	$24.3 \pm 0.02\%$	$22 \pm 0.02\%$
250	15	$36.5 \pm 0.03\%$	$18.2 \pm 0.06\%$	$36 \pm 0.04\%$
250	20	$34.4 \pm 0.06\%$	$15.3 \pm 0.05\%$	$42 \pm 0.03\%$
400	10	$14.2 \pm 0.02\%$	$8.6 \pm 0.02\%$	$89 \pm 0.06\%$
400	15	$9.3 \pm 0.05\%$	$5.3 \pm 0.06\%$	$92 \pm 0.02\%$
400	20	$6.1 \pm 0.02\%$	$1.5 \pm 0.02\%$	$98 \pm 0.01\%$

Table 3.2: Validation error for the neural network function approximator after active data collection; the best network has a maximum error of 6.2% and mean error of 1.5%. Reported errors are generated from 10-fold cross validation

The neural network allows us to determine the optimal I_{yy} and CG_z values for each dominant wave frequency. For a given wave frequency, we search through the combinations of I_{yy} and CG_z, in order to find the mass parameters which maximize the power output of the WEC given a dominant wave frequency. The combination of mass parameters which yields the highest energy output is the optimal set of I_{yy} and CG_z for a given wave frequency.

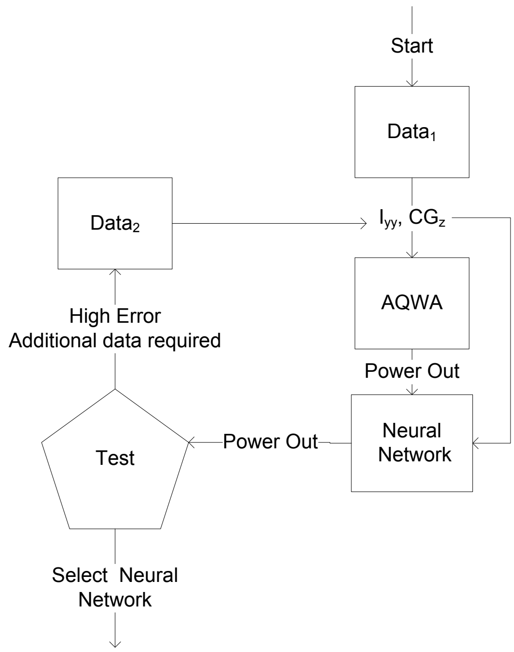

Figure 3.4: Neural Network Training: 250 data points (Data 1) were created uniformly across input space. These points were simulated in AQWA, giving the power output of each point. The neural network trained from this data was inadequate, so 150 new data points were simulated in AQWA (Data 2), in the region of the input space with highest uncertainty. The new dataset produced an accurate neural network function approximation

Chapter 4 – Time Domain Simulation

The neural network described in Chapter 3 calculates the power output of a WEC based on its design parameters without the use of AQWA. In order to judge the design of a WEC, it must be simulated in the time domain to calculate energy capture over some interval of time. The time domain simulation must take into account the control algorithm being utilized, as well as the wave climate in which the WEC operates. A time domain simulation was created using the neural network function approximator, in order to simulate the WEC over the course of a year of deployment. This chapter goes over the wave climate, the control algorithm, and the time domain simulator based off of this wave climate and controller.

4.1 Wave Occurence Prediction

Wave energy converters must be optimized for specific wave climates. There are several reasons for tailoring the design of a WEC to the climate it will operate in. First, the WEC should resonate at the frequencies seen in the wave climate which it will operate in. Furthermore, predicting the future wave climate is necessary for proper control of the WEC ballast chambers. It is impractical to have the ballast pumps constantly activated during operation, so the WEC ballast settings should be determined based on predictions of the wave climate for some future window of time, rather than react to current wave climates. Without an accurate prediction of the wave climate, the ballast settings will be suboptimal and power output will suffer. By designing a WEC for a specific climate, the ballast chamber geometry can be optimized for the waves expected in that climate, and the controller can set ballast weights based on accurate predictions.

The WEC is being optimized for use off of the Oregon coast. The wave climate in this region is defined by a wave occurence table, which gives the total number of hours per year that each dominant wave frequency is present. From the wave occurance table, a smooth and rough sea model are created for use in the time domain simulator. The smooth sea is the realistic wave climate based on the wave occurance table, and the rough sea is a random permutation of the smooth sea. Simulations in the smooth sea give insight to the normal operating performance of a WEC, and simulations in the rough sea give insight to the level or robustness of WEC performance under extreme conditions. Furthermore, the rough sea model helps us differentiate between designs which perform similarly in the smooth sea. The random sea forces transient operation, which is what the ballast geometry choices have the greatest effect on. At steady state, different ballast configurations perform equally well, assuming that the ballast configurations for the ideal I_{yy} and CG_z to be physically realizable. Ultimately, the smooth sea gives an idea of the expected annual power output of the WEC, while the random sea gives an idea of transient WEC performance and the relative efficacy of each ballast configuration analyzed.

4.2 Time Domain Simulation

For a given wave frequency, the optimal mass parameter values are known (Chapter 3). For a given set of ballast cuts, the mass of seawater in each ballast chamber which yields these mass parameters can be directly found (Chapter 3). So, for a given sea state and ballast configuration, we can directly determine the mass of seawater in each ballast chamber which will optimize the power output of the WEC. The time domain simulator takes the ballast chamber configurations as inputs, and simulates control and operation of the WEC for one year off of the Oregon coast.

4.2.1 Control of the Ballast

At any given point in time, each of the n ballast chambers has seawater in it, with a mass of m_i. The mass of seawater in each ballast chamber is:

$$\vec{m} = \{m_1, m_2, m_3, \ldots, m_n\} \tag{4.1}$$

Given the wave frequency of the sea, the optimal I_{yy} and CG_z values are known, and the optimal mass of seawater in each ballast chamber is also known. We define the optimal mass of seawater in each ballast chamber as:

$$\vec{m}^* = \{m_1^*, m_2^*, m_3^*, \ldots, m_n^*\} \tag{4.2}$$

The difference between the optimal masses and actual masses in the ballast chambers is:

$$\Delta\vec{m} = \{m_1^* - m_1, m_2^* - m_2, m_3^* - m_3, \ldots, m_n^* - m_n\} \tag{4.3}$$

Each ballast chamber has a water pump and PID controller, and we assume that the WEC is able to predict the future state of the sea. The *control loop time* is defined as the amount of time that the controller sets a single ballast configuration. For example, if the control loop time is 1 hour, then the controller finds the average dominant wave frequency ω_{avg} for that hour, then finds the optimal mass parameters CG_z^* and I_{yy}^* for each WEC component. The optimal mass parameters $\vec{m}^*$ are then calculated, and used as setpoints for the masses of ballast in each chamber. A PID controller then uses those setpoints to control the pump rates of ballast into and out of each chamber using the values $\Delta\vec{m}$, $\int\Delta\vec{m}$, and $\Delta\dot{\vec{m}}$. After the hour is complete, the process is repeated to reset the ballast mass in each chamber. The ballast control is shown in Algorithm 4.

4.2.2 Simulator Algorithm

With the sea state, ballast to mass parameter mapping, and control strategy defined, we can develop the time domain simulator for the WEC. Given a WEC ballast configuration, the inputs to the time domain simulator include the control loop time, the time discretization, and the sea type (random or smooth). The simulator tracks the power output of the WEC over a year of operation. At each time step in the simulation, the mass of the ballast in each chamber is used to de-

Algorithm 4: Ballast Control Algorithm

input: control loop time T_c, time step Δt ;
time $t = 0$;
Find $\vec{m}$ based on current WEC state ;
$\omega_{avg} = \frac{1}{T_c}\sum_{\tau=0}^{T_c} \omega_\tau$;
Find CG_z^* and I_{yy}^* using neural network approximator ;
Find $\vec{m}^*$ based on CG_z^* and I_{yy}^* ;
while $t \leqslant T_c$ **do**
    $\Delta\vec{m} = \vec{m}^* - \vec{m}$;
    if $\Delta\vec{m} \neq \vec{0}$ **then** use PID control to set pump rates $\vec{p}$;
    else $\vec{p} = \vec{0}$;
    $\vec{m} = \vec{m} + \Delta t \cdot \vec{p}$;
    $t = t + \Delta t$;
end

termine the mass parameters of each component of the WEC. The neural network then maps those mass parameters and the current dominant wave frequency to the instantaneous power output of the WEC. The power output is then multiplied by the time discretization term to yield the energy produced at that time step. The energy production at each time step is summed over the course of a year of operation, yielding the total annual energy capture. Throughout the entire simulation, the ballast control algorithm (Algorithm 4) sets the optimal ballast configuration in order to maximize energy capture. The time domain simulator algorithm is detailed in Algorithm 5.

A single simulation which has the WEC run for a year of operation takes about 45 seconds to run, compared to over 3 days using Ansys AQWA and a similar control strategy. This is a massive reduction in computational time, which allows for many more ballast configurations to be considered. With this new simulator, a large search through ballast configurations may be considered. This simulator is a crucial part of this work, because without a fast simulator, a search such as an evolutionary algorithm would be computationally intractable.

Algorithm 5: Time Domain Simulator Algorithm

input: control loop time T_c, time step Δt, sea type (random or smooth) ;
number of control loops $n_{loops} = \frac{1\ year}{T_c}$;
current control loop $l_c = 1$;
total energy $E_{tot} = 0$;
while $l_c \leqslant n_{loops}$ **do**
    run ballast control algorithm (Algorithm 4) ;
    foreach *time step* Δt *in control loop* **do**
        find Power $P_{\Delta t}$ using neural network ;
        $E_{tot} = E_{tot} + \Delta t \cdot P_t$;
    end
    $l_c = l_c + 1$;
end
return: E_{tot} ;

In order to test the accuracy of the time domain simulator created using the neural network function approximation, a set of 50 arbitrary WEC designs were simulated in both AQWA and our time domain simulator. The results from AQWA and our time domain simulator are consistent with each other, validating the accuracy of the time domain simulation using the neural network approximator. The results from this validation are shown in Table 4.1. A total of 50 randomly generated designs were tested, and 5 are shown in the table. The maximum error between AQWA and our function approximation based simulator was 4.65% in the designs tested. We conclude that the time domain simulator developed using the neural network function approximation is an accurate simulator to judge different

WEC Design	Average Power (AQWA)	Average Power (NN-based simulator)	Error
Design 1	115.27kW	116.01kW	0.64%
Design 2	422.94kW	422.12kW	0.19%
Design 3	354.19kW	355.68kW	0.42%
Design 4	272.64kW	264.41kW	3.01%
Design 5	194.38kW	203.42kW	4.65%

Table 4.1: Validation of time domain simulator. 50 arbitrary WEC designs were simulated in AQWA and our function approximation based time domain simulator. Our simulator had a maximum error of 4.65% in the designs tested

WEC designs.

Chapter 5 – Optimization

In order to optimize the design of the ballast configuration in the WEC, a search through the design space must be completed. In order to find the optimal ballast configuration, a search algorithm must consider the entire design space, as well as be free of human bias. With the neural network function approximator and the time domain simulator completed, an evolutionary algorithm is implemented to search through potential ballast designs in order to find an optimal ballast configuration. Because the neural network saves so much computational time in the simulation stage, a population based search such as an evolutionary algorithm is computationally feasible. Further, the evolutionary algorithm allows for an unbiased search across the entire feasible design space. The following sections explain the specific attributes of the evolutionary algorithm used to optimize the ballast configuration.

5.1 Population Representation

Each member of the population in the evolutionary algorithm is a collection of cut positions defining a specific set of ballast chambers. One population member is a vector with 16 cut positions; there are two $x-$ and $z-$ cut positions for each float, and eight $z-$cut positions for the spar. A single population member s_i is defined as:

$$s_i = \{x_{1,f}, x_{2,f}, z_{1,f}, z_{2,f}, x_{1,a}, x_{2,a}, z_{1,a}, z_{2,a}, \\ z_{1,s}, z_{2,s}, z_{3,s}, z_{4,s}, z_{5,s}, z_{6,s}, z_{7,s}, z_{8,s}\} \tag{5.1}$$

The goal of the optimization problem is to find an optimal s_i such that annual energy capture of the WEC is optimized.

5.2 Mutation Operator

The mutation operator has two stages. First, the number of mutations (between one and three) is randomly chosen. Random indices are chosen to determine which element(s) of the population member will be mutated. Once the elements of the population member which will be mutated are chosen, a random variable drawn from a Gaussian curve is added to those elements. The Gaussian curve has a mean of 0.0 meters and a standard deviation of 0.33 meters. The only restriction on the mutation operator is that cut positions are not allowed to leave the physical bounds of the device. As an example, consider the population member defined in Equation 5.1. If cut positions $x_{1,f}$, $z_{2,f}$, and $z_{4,s}$ were mutated by δ_1, δ_2, and δ_3, respectively, then the resultant solution would be:

$$s_i = \{x'_{1,f}, x_{2,f}, z_{1,f}, z'_{2,f}, x_{1,a}, x_{2,a}, z_{1,a}, z_{2,a},$$
$$z_{1,s}, z_{2,s}, z_{3,s}, z'_{4,s}, z_{5,s}, z_{6,s}, z_{7,s}, z_{8,s}\} \tag{5.2}$$

Thus, the mutation operator slightly alters a single population member, resulting in a new ballast chamber configuration. With the population members and mutation operator defined, all that is needed to complete an evolutionary algorithm is a fitness assignment operator and selection operator. The mutation operator is depicted in Figure 5.1.

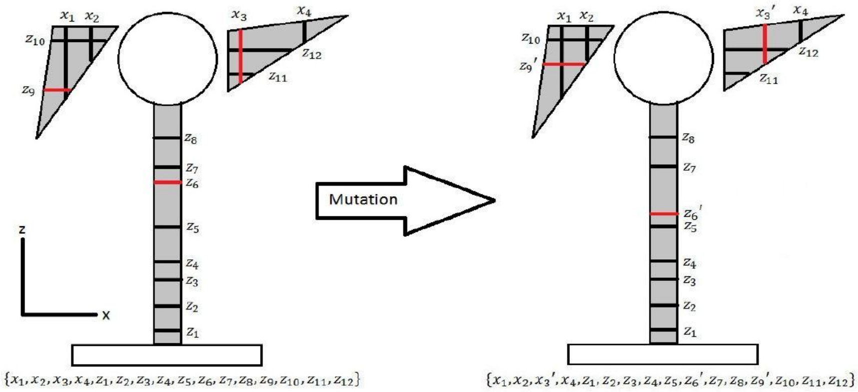

Figure 5.1: The mutation operator. The WEC on the left shows the initial ballast design, and the ballast cuts indicated in red are altered to produce a new WEC ballast design

5.3 Fitness Assignment and Selection

The fitness of each population member s_i as defined in Equation 5.1 must be determined in order to perform an intelligent search through the solution space. In order to assign a fitness to each population member, the time domain simulation (Chapter 4) is used. The fitness of each member is simply the energy captured in one year by that WEC configuration. So, the fitness of each population member is determined via the time domain simulator to gauge the relative quality of each ballast chamber configuration being considered. Once the fitness of each population member is defined, a method to select which population members will survive to the next generation is necessary. For member selection, an ϵ-greedy policy was used. A parameter $\epsilon \in [0, 1]$ is selected. The members with the highest fitness are selected with a probability of ϵ, and random members are selected with a probability of $(1 - \epsilon)$. It is important to occasionally select random ballast configurations, or the evolutionary algorithm would tend to get stuck in local optima. By adding a measure of stochasticity to the search, the probability of the algorithm getting stuck in a locally optimal solution is decreased. This is a key advantage of evolutionary algorithms when compared to gradient-descent or hillclimbing methods, which generally find the local optimum closest to the starting point of the search. We have now defined the representation of each population member, the mutation operator, the fitness assignment operator, and the selection operator. An evolutionary algorithm on the ballast designs can now be accomplished.

The evolutionary algorithm starts by randomly initializing a population of ballast configurations. At each generation, each member of the population is copied and mutated, resulting in a doubling of the population size. Each population member is then assigned a fitness using the time domain simulator. Then, the population is down-selected to its original size using ϵ-greedy selection based on the fitness values. The algorithm then proceeds to the next generation. The evolutionary algorithm is run for a set number of generations, and the final output of the algorithm is the population member with the highest fitness. This member corresponds to the optimal ballast configuration found by the algorithm. The evolutionary algorithm is detailed in Algorithm 6.

5.4 Hill Climbing Algorithm with Random Restarts

Although evolutionary algorithms are excellent tools for optimization in nonlinear and complex search spaces, a key drawback is that they are computationally expensive when compared to other optimization techniques such as gradient descent or hill climbing. A random restart hill climbing algorithm is also tested on the ballast optimization problem, in order to determine if the comutational complexity of an evolutionary algorithm is required, or if a simpler algorithm can be used to get the same results.

A hillclimbing optimization algorithm is a simple algorithm in which a solution in the search spaced is used to seed other solutions near its location, and the solution with the highest value is taken as the new solution. This process is repeated until a maximum value is found [3]. The key benefit to such an approach is that

Algorithm 6: Evolutionary algorithm to determine optimal ballast configuration

```
input: number of members n, ϵ, max generations G_max, control loop time T_c, time step Δt, sea type ;
for i = 1 to n do
    randomly initialize population member s_i using values drawn from uniform distribution ;
end
generation G = 1 ;
while G ⩽ G_max do
    create a copy of each member s_i, resulting in 2n members ;
    for i = n + 1 to 2n do
        mutate each member s_i ;
    end
    for i = 1 to 2n do
        calculate fitness f_i of member s_i using time domain simulator ;
    end
    select n members to survive to next generation using ϵ-greedy and f_i values ;
    G = G + 1 ;
end
return: population member s_i with highest fitness ;
```

the algorithm is computationally inexpensive. However, hill climbing algorithms tend to find the closest local optimum to the original solution, and are ill-equipped to find globally optimal solutions. Depending on the gradients associated with the search space, this drawback may prove to cause hill climbing algorithms to find severely suboptimal solutions.

In the case of the ballast optimization, the hill climbing algorithm is designed as follows. A random solution is defined in the same manner a random population member of the evolutionary algorithm is defined (Equation 5.1). For each parameter in the solution, two neighbor solutions are created. The first neighbor solution is created by adding a small value to the parameter, and the second neighbor solution is created by subtracting a small value from the parameter. For example, if a solution is defined as $\{x, y\}$, then the neighbor solutions $s_{n,i}$ are:

$$s_{n,1} = \{x + \delta, y\} \tag{5.3}$$

$$s_{n,2} = \{x - \delta, y\} \tag{5.4}$$

$$s_{n,3} = \{x, y + \delta\} \tag{5.5}$$

$$s_{n,4} = \{x, y - \delta\} \tag{5.6}$$

So, if a solution is defined by n parameters, then $2n$ neighbor solutions are created. Each solution is then tested to determine which solution is best, and that solution is then used to start a new local hill climbing search. The hill climbing algorithm for searching for an optimal ballast configuration is shown in Algorithm 7.

Algorithm 7: Hill climbing algorithm for ballast optimization

input: initial solution s_1, step size δ, max steps N_{max}, number of parameters n_p;
step $n = 1$;
while $n \leqslant N_{max}$ **do**
    for $j = 1$ *to* n_p **do**
        create neighbor solutions $s_{n,j}$ based on s_n and δ ;
    end
    find value $v_{n,j}$ of each solution using time domain simulator ;
    select best solution s_n^* based on $v_{n,j}$ values ;
    $s_{n+1} = s_n^*$;
    $n = n + 1$;
end
return: solution $s_{N_{max}}$;

If the ballast configuration search space is not monotonic with respect to power output, then the hill climbing algorithm will return suboptimal ballast configurations. However, the nature of the search space is not explicitly known, so it is important to determine how a search algorithm which is simpler than the evolutionary algorithm will perform. If the search space is not too complex, the simpler hill climbing algorithm may give adequate results. However, as the search space grows in complexity, the hill climbing algorithm will perform worse. The hill climbing algorithm serves as a benchmark to compare with the evolutionary algorithm, in order to determine if the additional computation cost of evolution is justified by the final solution found.

Chapter 6 – Experiments and Results

A series of three experiments was carried out in order to optimize the ballast chamber cuts. First, a WEC with ballast chambers found with the evolutionary algorithm was compared to a WEC without ballast chambers. In this case, the control loop time was zero. This experiment gives insight to the additional performance that ballast chambers give to the WEC. Next, using multiple control loops, ballast configurations found from the evolutionary algorithm were compared to ballast configurations found with the hill climbing algorithm. This experiment gives insight to the additional performance that ballast control provides, as well as allowing us to compare the relative efficacy of evolutionary algorithms and hill climbing algorithms. Next, an experiment was run where the number of ballast cuts was varied and the ballast configurations were optimized via evolution. This gives insight to how the addition of ballast chambers affects the power output of the WEC. Finally, a comparison of the simulation time using the neural network function approximation versus the Ansys AQWA simulator was completed. This experiment shows the importance of developing the function approximator in order to complete the evolutionary algorithm. The experiments are outlined below.

1. With a control loop time of zero, ballast configuration from evolution is compared against a ballast-free WEC

2. Utilizing multiple control loops, ballast configuration from evolution is compared to ballast configuration from hill climbing

3. The number of ballast cuts was varied and ballast configurations are optimized with evolution

4. The computational costs of the neural network function approximator and Ansys AQWA are compared

For each evolutionary experiment, a population size of 500 members is used, and the algorithm progresses for 1000 generations. For each set of results, the error in the mean ($\sigma/\sqrt{N}$) is reported, where σ is the sample standard deviation and $N = 100$ is the number of statistical runs.

6.1 Ballast-Free WEC vs. Ballasted WEC

Ballast chamber cut positions were evolved in order to maximize annual WEC energy capture, with a control loop time of zero (Figure 6.1). A control loop time of zero corresponds to a completely reactive controller. Rather than predicting future sea states and adjusting ballast setpoints accordingly, ballast setpoints are constantly reset based on current sea conditions. A population of ballast configurations is randomly initialized, and the fitness of each population member was based on how much energy was captured in one year in the time domain simulation. For comparison with the ballasted WEC, a ballast free WEC was designed. Although there are no ballast chambers in this WEC, the mass parameters I_{yy} and CG_z

are set to the optimal values associated the the average dominant wave frequency encountered throughout the entire year. This comparison gives an indication of the gains in power output that can be obtained by the addition of ballast chambers alone.

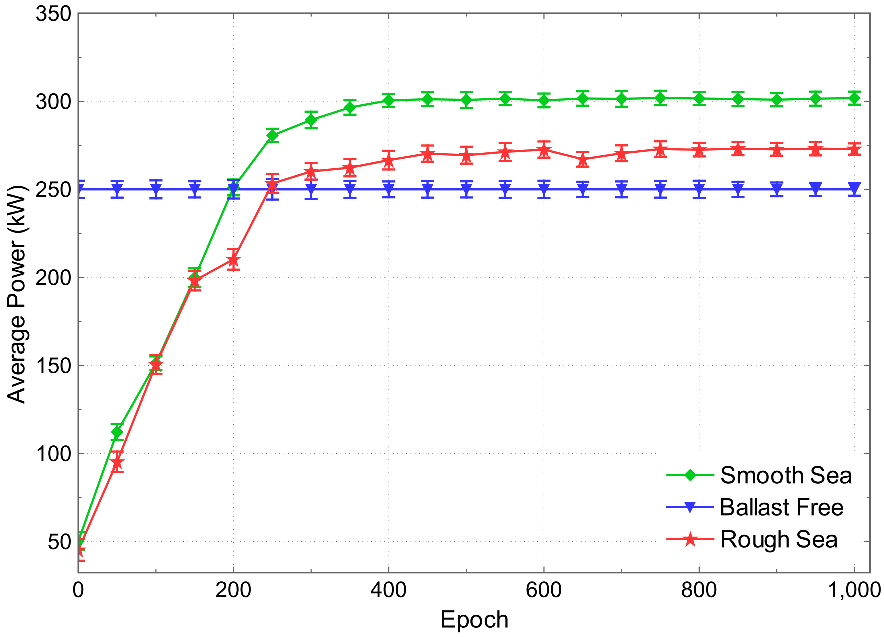

Figure 6.1: The evolutionary algorithm performance in two environments are compared against a ballast-free WEC which retains a static configuration during operation. The results shows that ballast chambers makes the average power increase up to 17% when compared to the ballast-free WEC. The smooth sea is a realistic sea climate based on the Oregon coast, and the rough sea is a random permutation of this sea climate. (Note: as the ballast-free model remains static throughout operation, it performs equally well in the rough and smooth seas, because they are permutations of each other)

A few interesting observations can be made from Figure 6.1. First, the WEC produces more power when operating in a smooth sea than when operating in a random sea. This is due to the fact that the WEC is producing the most power when operating in steady state. In the random sea, the time spent in transient operation increases, reducing the power output. As the difference in performance between two ballast configurations is primarily seen in transient operation, the random sea is selected as the primary testing domain. Two ballast configurations which are capable of attaining the same mass parameters perform identically in steady state operation, but have different power outputs when operating in transient operation. Thus, transient operation is what ultimately differentiates one ballast configuration from another. The evolved WEC design performed better than the ballast-free WEC whether operating in a smooth or random sea. This result shows that the addition of ballast chambers improves WEC energy capture rates, and that an evolutionary algorithm can effectively guide the design of the ballast configuration in order to increase energy capture.

6.2 Comparison of Control Loop Times

We have shown that ballast chambers can effectively increase the power output of a WEC, but still do not know the extent of the power increase if the ballast controller is also optimized. We now test the performance of evolving ballast configurations while varying the control loop time in the ballast control algorithm (Algorithm 4). The WEC can alter its mass properties at any time by pumping water into and out of the ballast chambers. In the time domain simulator, the controller considers a length of time (defined by the control loop time) with which to set the desired ballast masses. The maximum power produced (as found from evolution) as a function of the control loop time is shown in Figure 6.2.

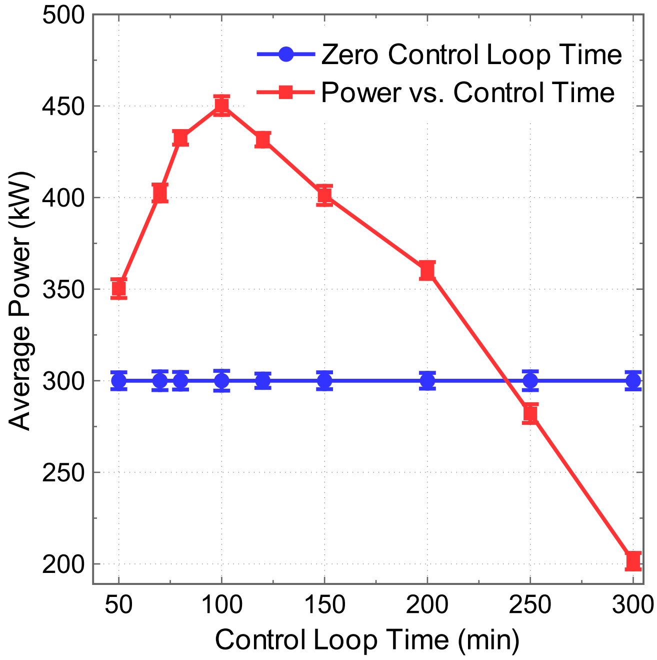

Figure 6.2: Average power output as a function of the control loop time. The constant line is the power output obtained when the control loop time was zero

From Figure 6.2, we see that the choice of the control loop time is critical to maximizing WEC power output. If the WEC has a short control loop time, it is constantly in transient operation, and power output decreases. If the WEC has too long of a control loop time, the prediction of the expected wave climate becomes inaccurate, also reducing the power output. There exists an optimal control loop time, which is long enough to prevent constant transient operation, but short enough to give accurate wave predictions. For our experiments, the optimal control loop time is about 100 minutes in the random sea climate.

Next, we compare the maximum WEC energy output found from evolution to the energy output of a WEC produced through hill climbing (Figure 6.3). For the hill climbing approach, 500 points in the solution space are randomly initialized. Each solution is used to seed the hill climbing algorithm (Algorithm 7), where the neighbor states were created with $\delta = 1$mm. The solution with the highest total power output from all hill climbing experiments is then chosen for comparison.

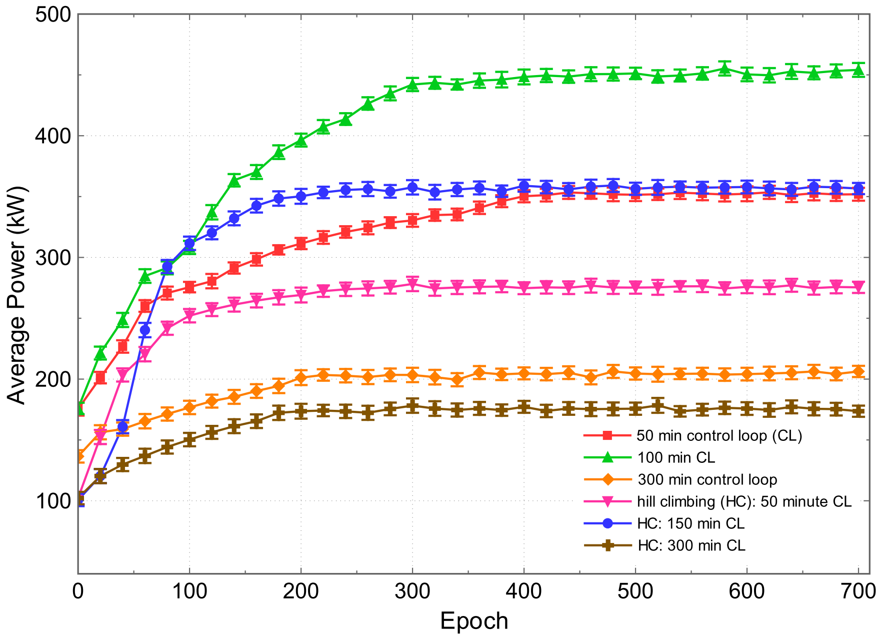

Figure 6.3: Comparison of evolution of ballast cuts while varying control loop times (50, 100, and 300 minutes) versus the ballast cuts produced from hill-climbing

The results in Figure 6.3 are consistent with the results from Figure 6.2. The WEC with the highest energy capture was found when the control loop time is 100 minutes. When the control loop time is higher or lower than 100 minutes, the WEC performance suffers. The ballast configuration found with hill climbing captures about 22% less power than the ballast configuration from evolution. This shows that the extra computational cost associated the evolutionary algorithm is a worthwhile expense, because the final design configuration is significantly better than that found from hill climbing. The results for ballast configurations found from hill climbing with control loop times other than 100 minutes are omitted, as they produce less power and are thus not of interest. When comparing Figures 6.3 and 6.1, we see that the optimization of the controller and ballast configuration produces much better results than optimizing solely the ballast configuration. When both the controller and ballast configuration are optimized, the WEC produces 84% more power than the ballast-free WEC. We have shown that an evolutionary algorithm is capable of finding an excellent design for the ballast configuration of the WEC, resulting in almost twice the power output of a ballast free model. We now investigate the effects of varying the number of ballast chambers in the WEC.

6.3 Varying Number of Ballast Chambers

Next, we varied the amount of ballast cuts in order to determine how the number of ballast chambers affects the power output. Intuitively, increasing the number of ballast chambers should increase the power output. This is because more ballast chambers corresponds to smaller ballast chambers, which can fill and empty faster. This corresponds to a decreased transient response. However, the addition of ballast chambers also means that more pumps and materials are required to produce the WECs. So, if the gains in energy aquisition are offset by the increased cost of manufacturing the device, then adding more ballast chambers is not a useful way of increasing power output of the WEC. For this experiment, each of the floats has n ballast cuts, and the spar has $2n$ ballast cuts. The evolutionary algorithm is then used to optimize the position of the ballast cuts, using the time domain simulator with a control loop time of 100 minutes to assign fitness values. The results of this experiment are shown in Figure 6.4.

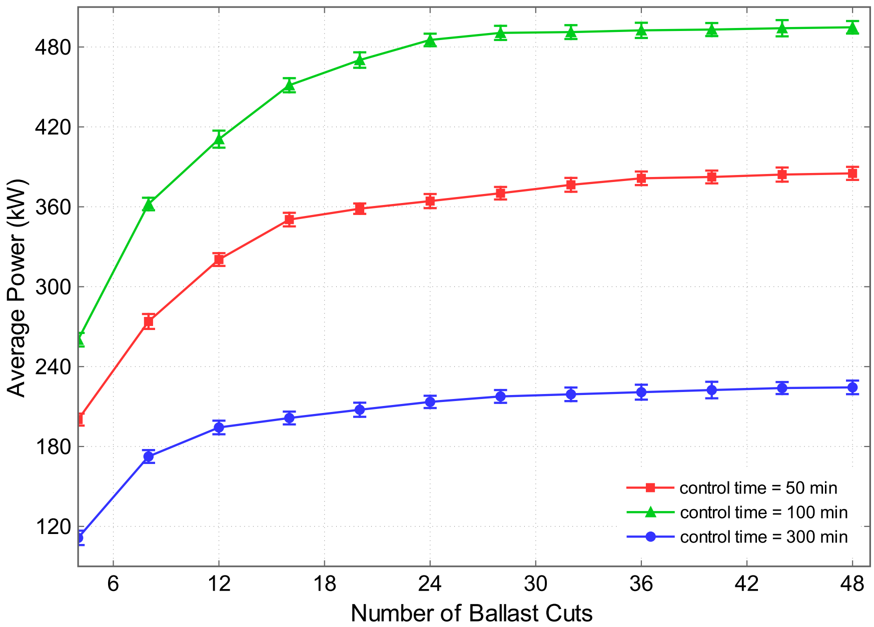

Figure 6.4: Optimized WEC power output as a function of the number of ballast cuts. As the number of ballast chambers increases, the power output of the device increases. However, each additional ballast cut yields smaller gains in performance

From Figure 6.4, we see that increasing the number of ballast chambers corresponds to an increase in the power output of the WEC. However, each additional ballast chamber yields a smaller increase in performance. When there are few ballast chambers, the addition of extra ballast chambers improves performance significantly, but once there are many ballast chambers, adding more does little to impact performance. It is not explicitly known how the number of ballast cuts affects the manufacturing cost of the WEC, so these results cannot be used to definitively determine what the optimal number of ballast cuts is. However, these results in conjunction with estimates of the cost of each ballast configuration would allow us to decide which number of ballast chambers is best when comparing the cost of manufacturing the device with its power output.

6.4 Computation Cost of Function Approximation

Completing the evolutionary algorithm was only possible because of the speed of the neural network function approximator. Running the algorithm using AQWA would be prohibitively slow, because of the number of fitness calculations required during the course of evolution. At each time step in the time domain simulation, the neural network is called to calculate the power output. The time domain simulation is embedded in the evolutionary algorithm, and runs for each population member at each generation. Evaluating the neural network once takes less than one second, while one AQWA simulation takes around 24 hours. A comparison of the time needed to run some experiments using the neural network or AQWA is shown in Table 6.1.

In our experiments with evolving the ballast geometry, a population size of 500 run for 1000 generations takes around 6 hours to complete. If an AQWA simulation was run every time the nerual network was called, the evolutionary algorithm would take up to two million hours (over 200 years) to complete. This shows how

Experiment	Population Size	Generations	NN (hours)	AQWA (hours)
1 Power Calculation	n/a	n/a	$1 \cdot 10^{-3}$	24
EA optimization	100	1000	4	$400 \cdot 10^{3}$
EA optimization	500	1000	6	$2 \cdot 10^{6}$

Table 6.1: Comparison of time needed to run experiments using neural network function approximation vs. using AQWA. AQWA is prohibitively slow, with evolutionary algorithms taking 6 hours using the neural network taking over 200 years using AQWA

important the function approximation is to the evolutionary search. Without some type of function approximation to replace the AQWA simulations, completing an evolutionary algorithm would be impossible. As population size or the number of generations increases to any realistic amount for running an evolutionary algorithm, AQWA ceases to be a viable method for finding power output. Thus, the key to completing this work was in fact developing an approximation to find the power outputs of the WEC quickly and efficiently.

Chapter 7 – Discussion

The traditional approach to optimizing WECs has some key drawbacks, including the time associated with simulations and the bias imparted in the search by human experts. This thesis alleviates these problems by developing a faster simulation process, as well as using an unbiased automated search algorithm. Specifically, we approximated the power output of a WEC based with an artificial neural network. Active data collection is used to minimize the error of this function approximator. A time domain simulation is developed using the neural network; this simulator is orders of magnitude faster than simulations in AQWA. Finally, an evolutionary algorithm optimized the ballast configuration of the WEC, using the time domain simulator to assign fitness levels to each design.

We used an evolutionary algorithm to successfully design the ballast configuration of a WEC, given a specific WEC geometry. The key to running this evolutionary algorithm was the neural network function approximator, which drastically reduced the computational time of calculating power output. The key difference between our work and traditional WEC design processes is the implementation of this function approximator. By reducing the time needed to analyze one specific design, we were able to consider many more candidate designs than in a normal WEC design process.

Another important difference between this research and the traditional WEC design process is the use of evolution. WEC design typically involves a human expert, who often imparts bias into the analysis (intentionally or unintentionally). By automating the design process with an evolutionary algorithm, the need for in-depth domain knowledge is eliminated, because more designs may be considered and the designs are automatically searched through in an intelligent manner. Further, the evolutionary algorithm lacks the bias associated with human designers, which allows for solutions to be found may be overlooked in the traditional design process.

The algorithm presented in this paper could potentially be extended to many other design problems, especially similar problems in wave energy. We optimized the ballast configuration for a specific WEC geometry, but there is no apparent reason that this algorithm would not extend to other geometries. For any arbitrary WEC geometry, if a function approximation for power can be developed using simulation data, then that function approximator could be used in a time domain simulation similar to the one implemented in this research, to predict annual power output of a WEC as a function of chosen design parameters. More generally, any problem which could be solved with a population based search but requires time-consuming simulations could potentially be solved using an algorithm similar to the one presented in this paper. Future research of this algorithm in other domains could potentially demonstrate that substituting a simulation with a faster function approximation could result in a greatly improved design process.

Bibliography

[1] Ansys AQWA: Elucidative programming, 2010.

[2] A.K. Agogino and K. Tumer. Efficient evaluation functions for evolving coordination. *Evolutionary Computation*, pages 257–288, 2008.

[3] Justin Boyan and Andrew Moore. Learning evaluation functions to improve optimization by local search. *Journal of Machine Learning*, pages 77–112, November 2000.

[4] Tianping Chen and Robert Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, pages 911–917, 1995.

[5] Kalyanmoy Deb. Evolutionary algorithms for multi-criterion optimization in engineering design. *Proceedings of Evolutionary Algorithms in Engineering and Computer Science*, pages 135–161, 1999.

[6] F.J.M. Farley and R.C.T. Rainey. Radical design options for wave-profiling wave energy converters. *21st International Workshop on Water Waves and Floating Bodies*, April 2006.

[7] Al Globus, Greg Hornby, Derek Linden, and Jason Lohn. Automated antenna design with evolutionary algorithms. *AAIA Space 2006 Conference*, (7242), 2006.

[8] R. Hecht-Neilsen. Theory of the backpropogation neural network. *International Joint Conference on Neural Networks*, pages 593–605, June 1989.

[9] Gregory S. Hornby, Seichi Takamura, Takashi Yamamoto, and Masahiro Fujita. Autonomous evolution of dynamic gaits with two quadruped robots. *IEEE Transactions on Robotics*, pages 402–410, 2005.

[10] J.L. Hudson, M. Kube, R.A. Adomaitis, I.G. Kevrekidis, A. Lapedes, and R. Farbar. Nonlinear signal processing using neural networks: Prediction and system modeling. In *Los Alamos National Laboratory Theoretical Division*, pages 2075–2081, July 1987.

[11] Rainer Leupers and Fabian David. A uniform optimization technique for offset assignment problems. In *Code Optimization Techniques for Embedded Processors: Methods, Algorithms, and Tools*, pages 3–8. 2000.

[12] Peter X. Liu, Ming J. Zuo, and Max Q.-H. Meng. Using neural network function approximation for optimal design of continuous-state parallel-series systems. *Computers & Operations research*, pages 339–352, July 2001.

[13] M. McCormick. *Ocean Wave Energy Conversion*. John Wiley and Sons, New York, 1981.

[14] Denis Mollison, O.P. Buneman, and S.H. Salter. Wave power availability in the ne atlantic. *Letters to Nature*, pages 223–226, 1976.

[15] M. Mueller. Electrical generators for direct drive wave energy converters. In *IEE Proceedings*, pages 446–456. IET, February 2002.

[16] K. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, pages 4–27, March 1990.

[17] D Ross. *Power from the Waves*. Oxford University Press, New York, 1995.

[18] W. Sarle. Neural networks and statistical models. In *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, pages 84–89, April 1994.

[19] Edward J. Shaughnessy, Ira M. Katz, and James P. Schaffer. *Introduction to Fluid Mechanics*. Oxford University Press, New York, 2005.

[20] J. Wieczorek, O. Gol, and Z. Michalewicz. An evolutionary algorithm for the optimal design of induction motors. *IEEE Transactions on Magnetics*, pages 3882–3887, November 1998.